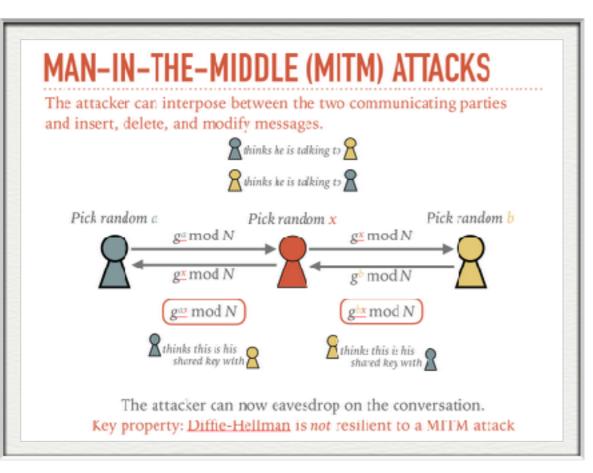
# PROVING WHO YOU ARE TLS & THE PKI

#### **CMSC 414** MAR 29 2018



#### **RECALL OUR PROBLEM WITH DIFFIE-HELLMAN**



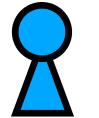
The two communicating parties thought, *but did not confirm*, that they were talking to one another.

Therefore, they were vulnerable to MITM attacks.

Certificates allow us to verify with whom we are communicating.

We will solve this by incorporating public key cryptography

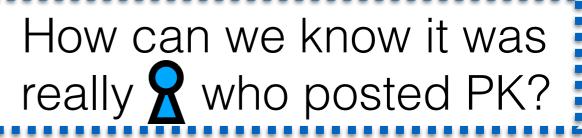
How can we know it was really **2** who posted PK?

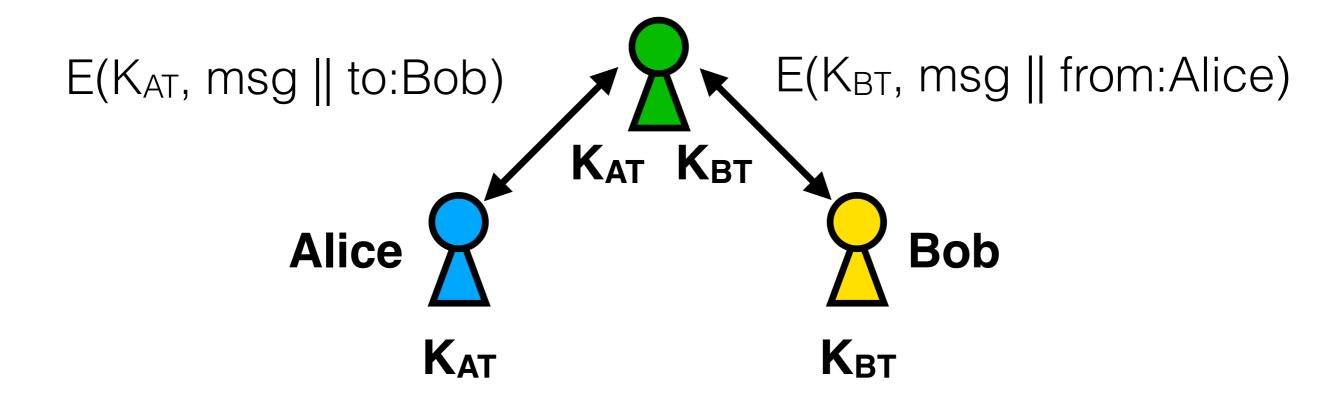




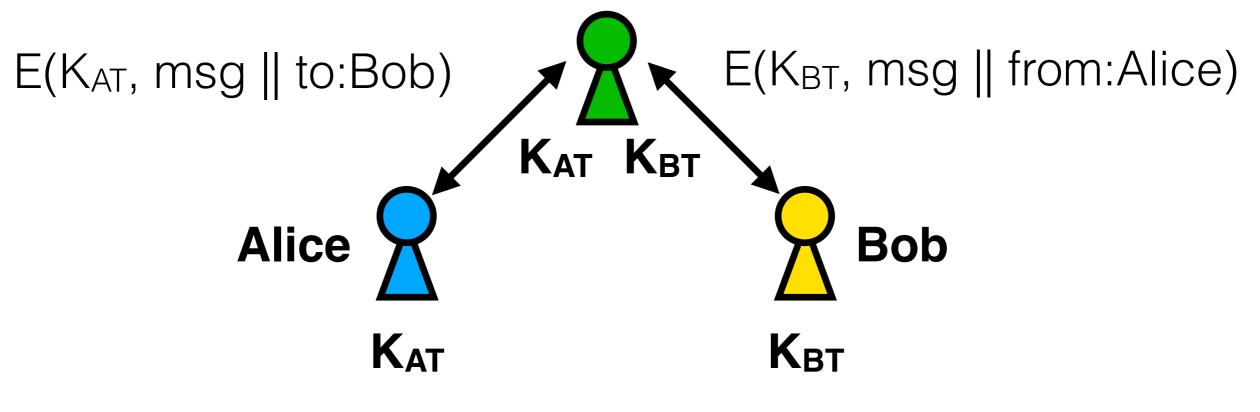


Generate public/private key How can we know it was pair (PK,SK); publicize PK really **2** who posted PK?





Generate public/private key How can we know it was pair (PK,SK); publicize PK really **2** who posted PK?



Can we achieve authentication without Trent in the middle of *every message*?

Trent **(PK<sub>T</sub>, SK<sub>T</sub>)** 

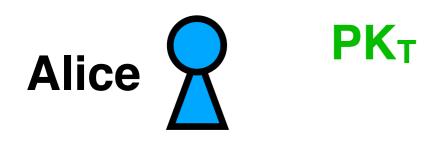
1. Trent's public key is widely disseminated (pre-installed in browsers/operating systems)

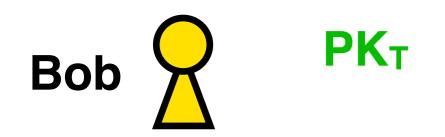




Trent **(PK<sub>T</sub>, SK<sub>T</sub>)** 

1. Trent's public key is widely disseminated (pre-installed in browsers/operating systems)



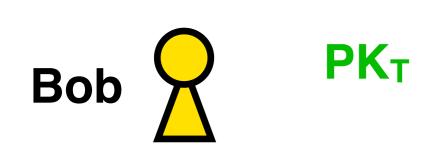


Trent **(PK<sub>T</sub>, SK<sub>T</sub>)** 

Alice  $\begin{array}{c} \mathsf{PK}_{\mathsf{T}} \\ \mathsf{PK}_{\mathsf{A}}, \, \mathsf{SK}_{\mathsf{A}} \end{array}$ 

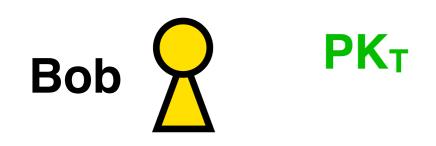
1. Trent's public key is widely disseminated (pre-installed in browsers/operating systems)

2. Alice generates a public/private key pair and asks Trent to bind her
 PKA to her identity



Trent (PK<sub>T</sub>, SK<sub>T</sub>) Trent vets Alice Alice PK<sub>T</sub> (PK<sub>A</sub>, SK<sub>A</sub>) 1. Trent's public key is widely disseminated (pre-installed in browsers/operating systems)

2. Alice generates a public/private key pair and asks Trent to bind her **PKA** to her identity

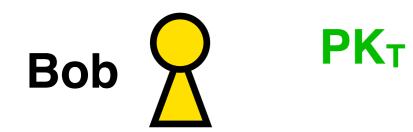


Trent (PKT, SKT) Trent vets Alice 1. Trent's public key is widely disseminated (pre-installed in browsers/operating systems)

2. Alice generates a public/private key pair and asks Trent to bind her **PKA** to her identity

3. Trent *signs* a message (with **SK<sub>T</sub>**):

"The owner of the secret key corresponding to **PKA** is Alice"



Trent (PK<sub>T</sub>, SK<sub>T</sub>) Trent vets Alice Alice PK<sub>T</sub> (PK<sub>A</sub>, SK<sub>A</sub>)

**PK**<sub>T</sub>

Bob  $\heartsuit$ 

1. Trent's public key is widely disseminated (pre-installed in browsers/operating systems)

2. Alice generates a public/private key pair and asks Trent to bind her PKA to her identity

3. Trent *signs* a message (with **SK<sub>T</sub>**):

"The owner of the secret key corresponding to **PKA** is Alice"

This message + sig = Certificate

Trent **(PK<sub>T</sub>, SK<sub>T</sub>)** Trent vets Alice PK<sub>T</sub> (PK<sub>A</sub>, SK<sub>Δ</sub>) Alice Alice =  $PK_A$ ΡΚτ Bob

1. Trent's public key is widely disseminated (pre-installed in browsers/operating systems)

Alice generates a public/private key pair and asks Trent to bind her
 PKA to her identity

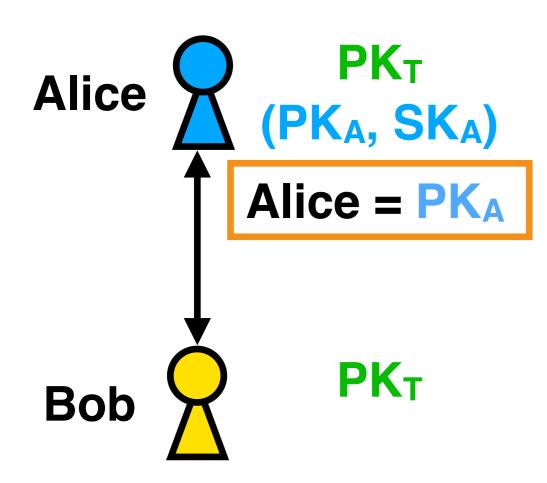
3. Trent *signs* a message (with **SK<sub>T</sub>**):

"The owner of the secret key corresponding to **PKA** is Alice"

This message + sig = Certificate

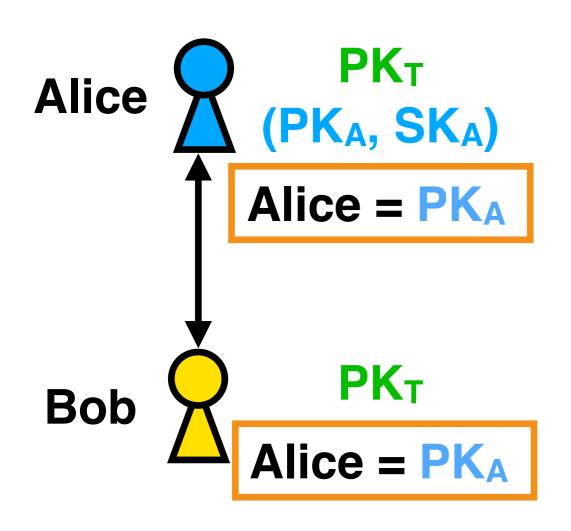


4. Alice makes her certificate publicly available
(or Bob simply asks for it)





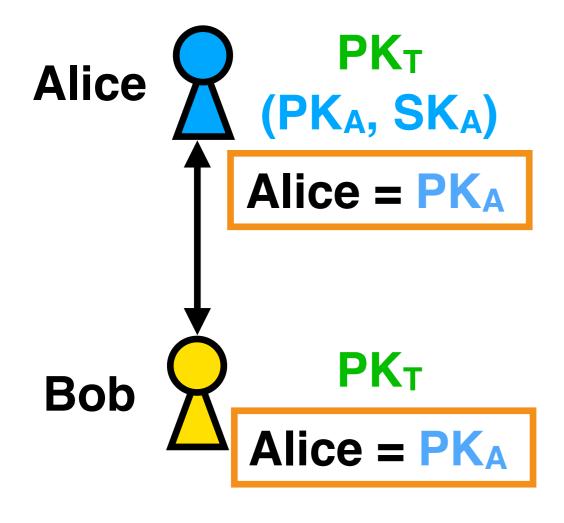
4. Alice makes her certificate publicly available
 (or Bob simply asks for it)





4. Alice makes her certificate publicly available
(or Bob simply asks for it)

5. Bob verifies the **certificate** using **PK**<sub>T</sub>

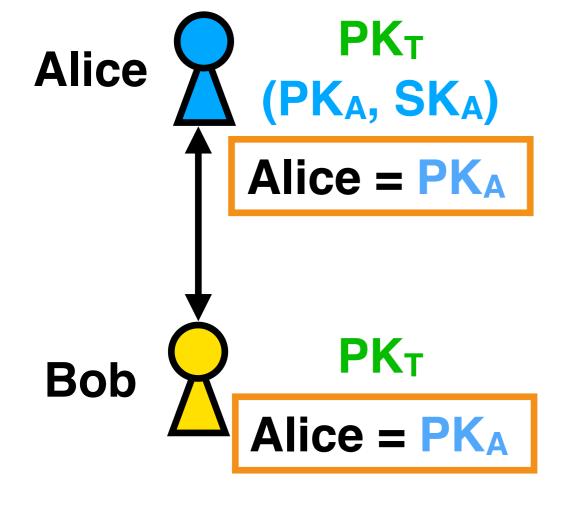


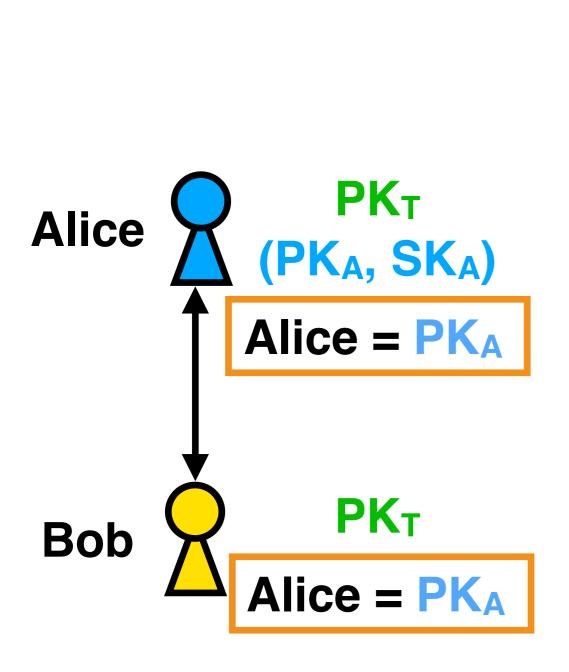


4. Alice makes her certificate publicly available
(or Bob simply asks for it)

 Bob verifies the certificate using PKT

If Bob trusts Trent, then Bob trusts that he properly vetted Alice, and thus that her public key is **PK**A





Trent **(PK<sub>T</sub>, SK<sub>T</sub>)** 

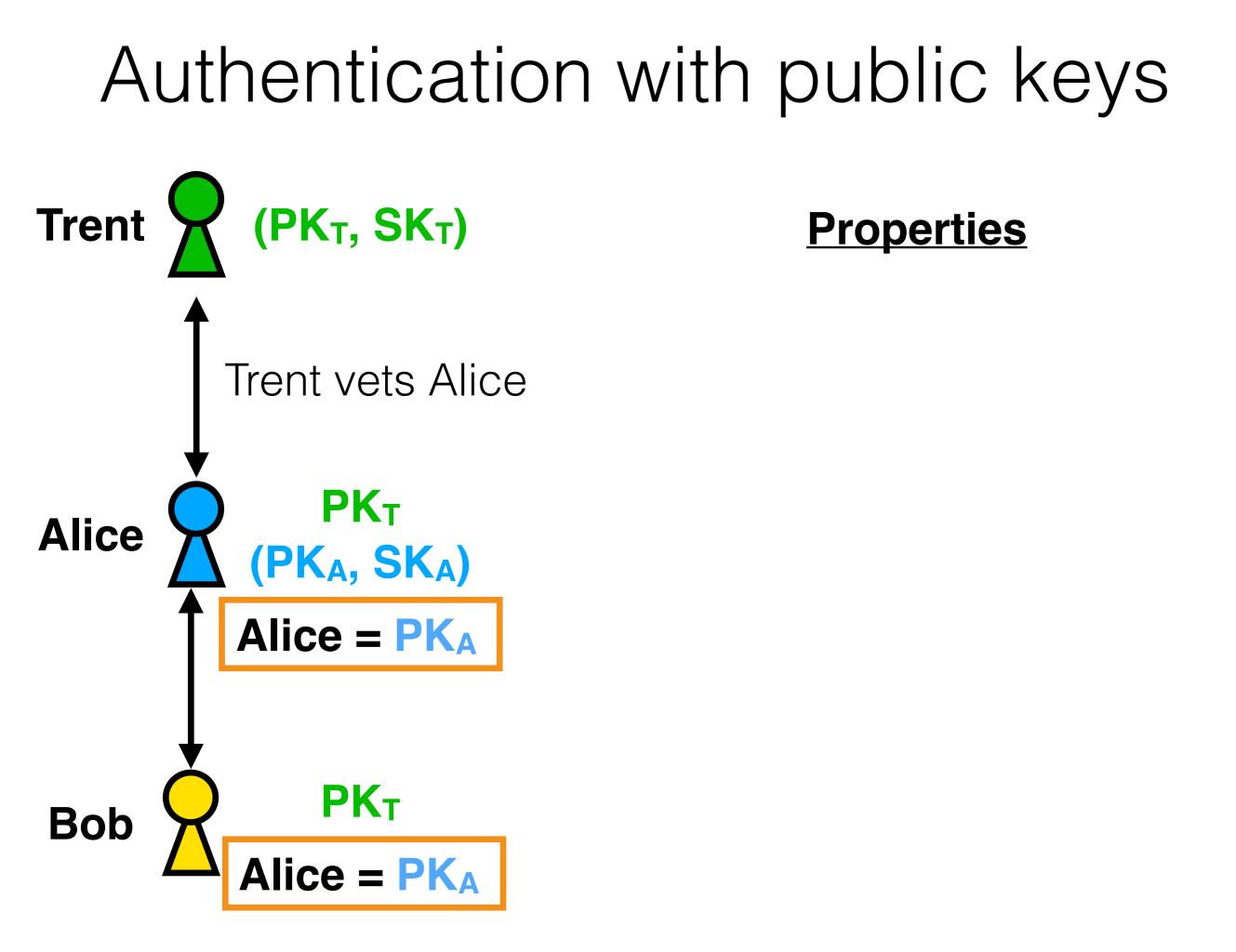
4. Alice makes her certificate publicly available
(or Bob simply asks for it)

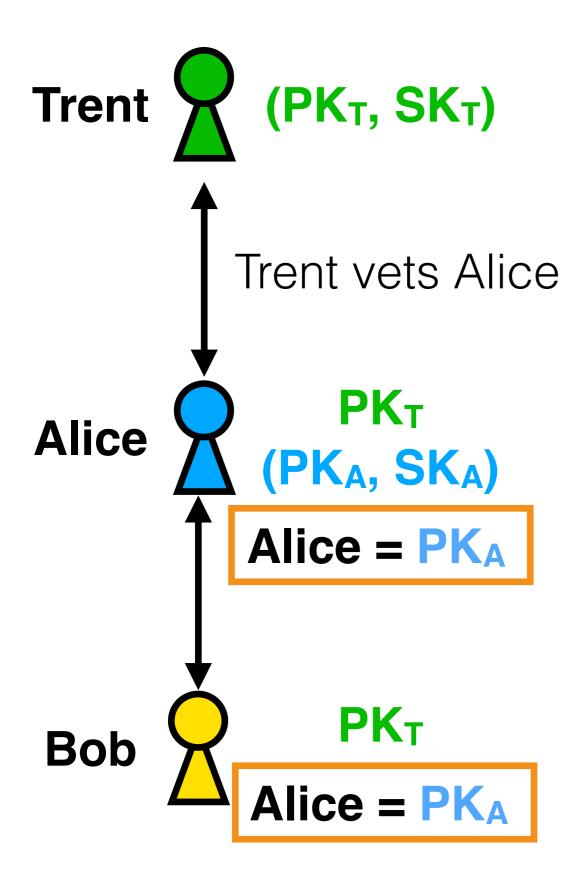
5. Bob verifies the **certificate** using **PK<sub>T</sub>** 

If Bob trusts Trent, then Bob trusts that he properly vetted Alice, and thus

that her public key is PKA

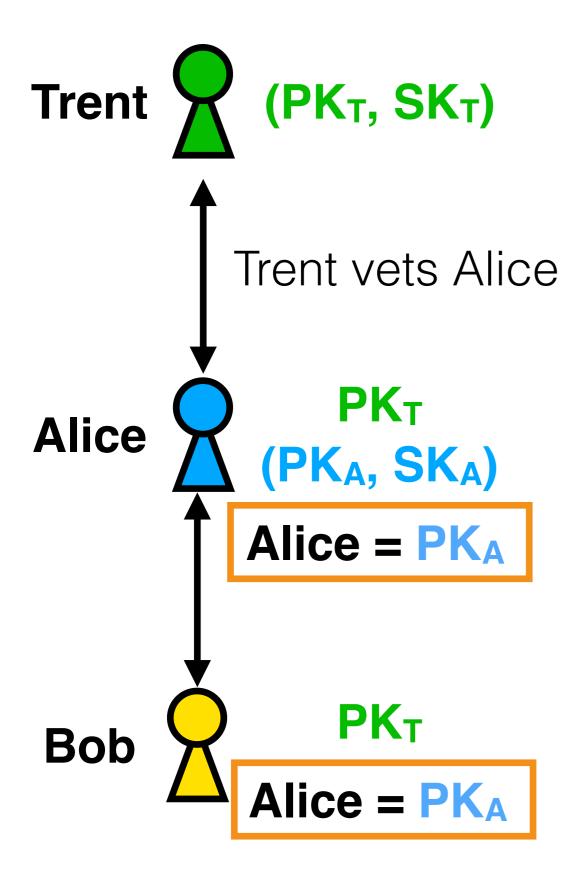
6. Bob (via hybrid encryption) sends a message to Alice using her public key PKA





#### **Properties**

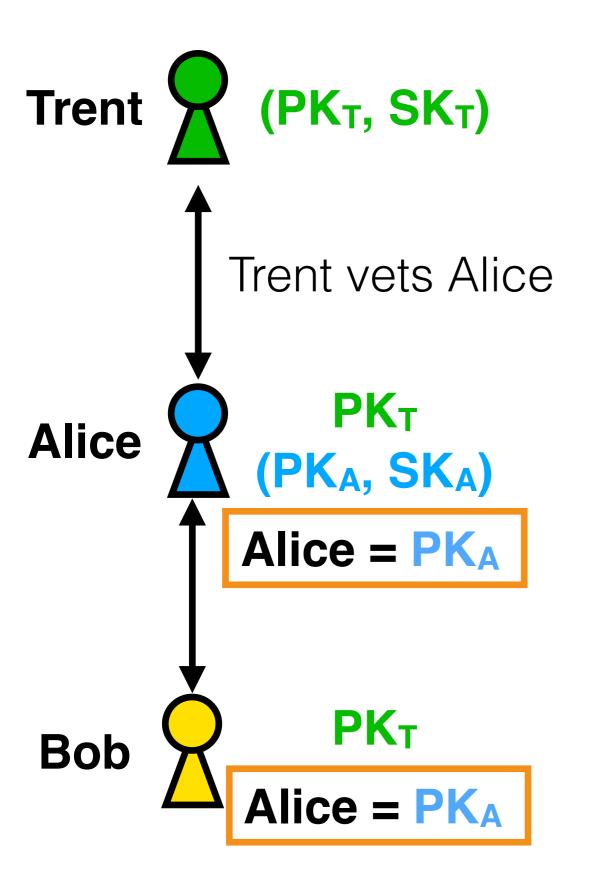
Trent need be online only when giving out **certificates**, not any time users want to communicate with one another



#### **Properties**

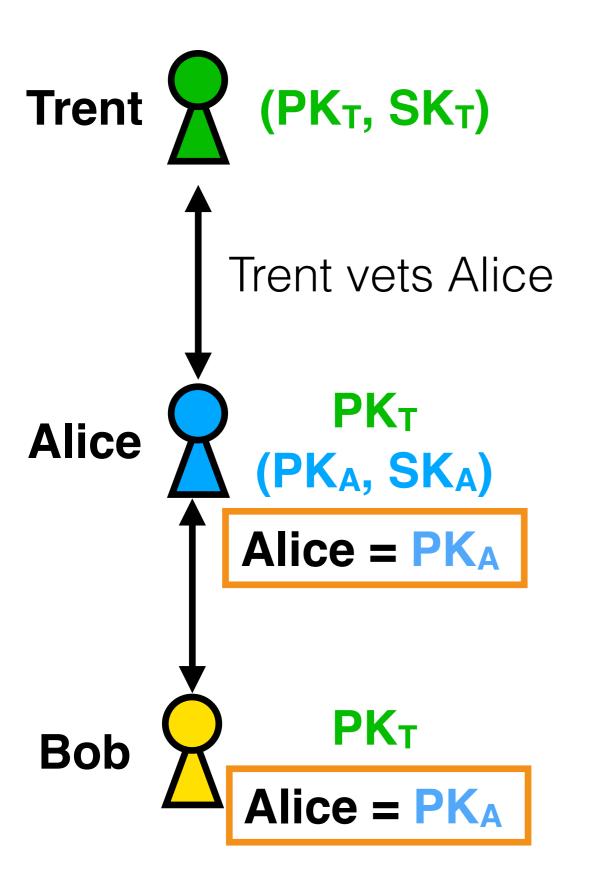
Trent need be online only when giving out **certificates**, not any time users want to communicate with one another

Alice and Bob can communicate in an authenticated manner without having to go through Trent



Trust assumptions from our symmetric key protocol:

Do not *read* messages
 Do not *alter* messages
 Do not *forge* messages
 Do not *forge* messages



Trust assumptions from our symmetric key protocol:

Do not *read* messages
 Do not *alter* messages
 Do not *forge* messages
 Do not *forge* messages

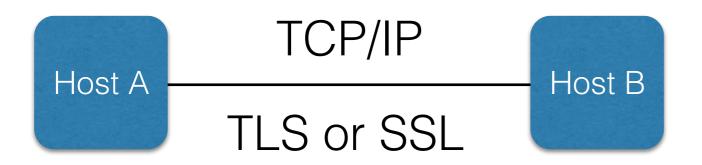
Trust assumptions in this public key protocol:

**1. Correctly vet users** 

(Some more in practice...)

# TLS/SSL

- TLS (Transport Layer Security)
  - A suite of protocols to provide secure communication
    - Confidentiality by applying block & stream ciphers
    - Integrity with MACs
    - Authenticity with certificates
  - Predecessor: SSL (secure sockets layer)
    - TLS was proposed as an upgrade
    - All versions of SSL are considered insecure (recently, the POODLE—padding oracle—attack)



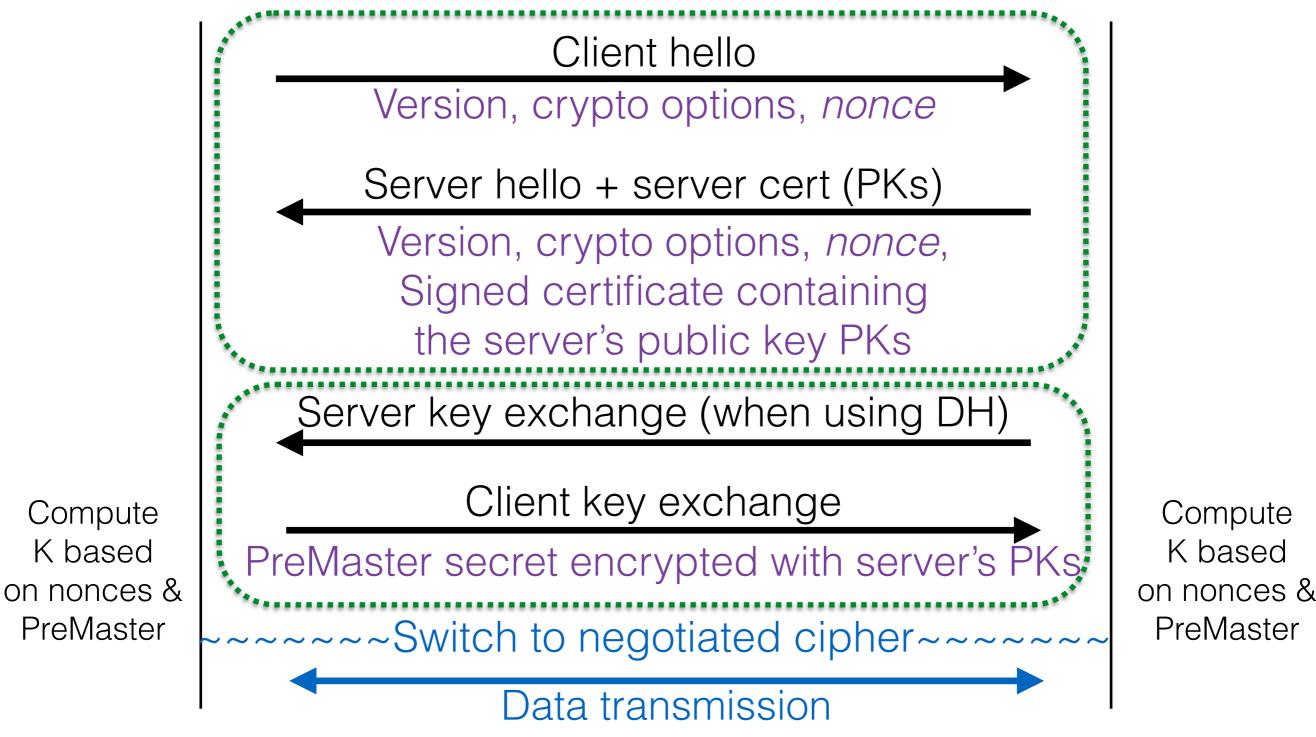
TCP/IP: Host A and B can send packets to one another

TLS/SSL: operate "over" TCP/IP to ensure security/authenticity

## TLS/SSL protocol (high level)

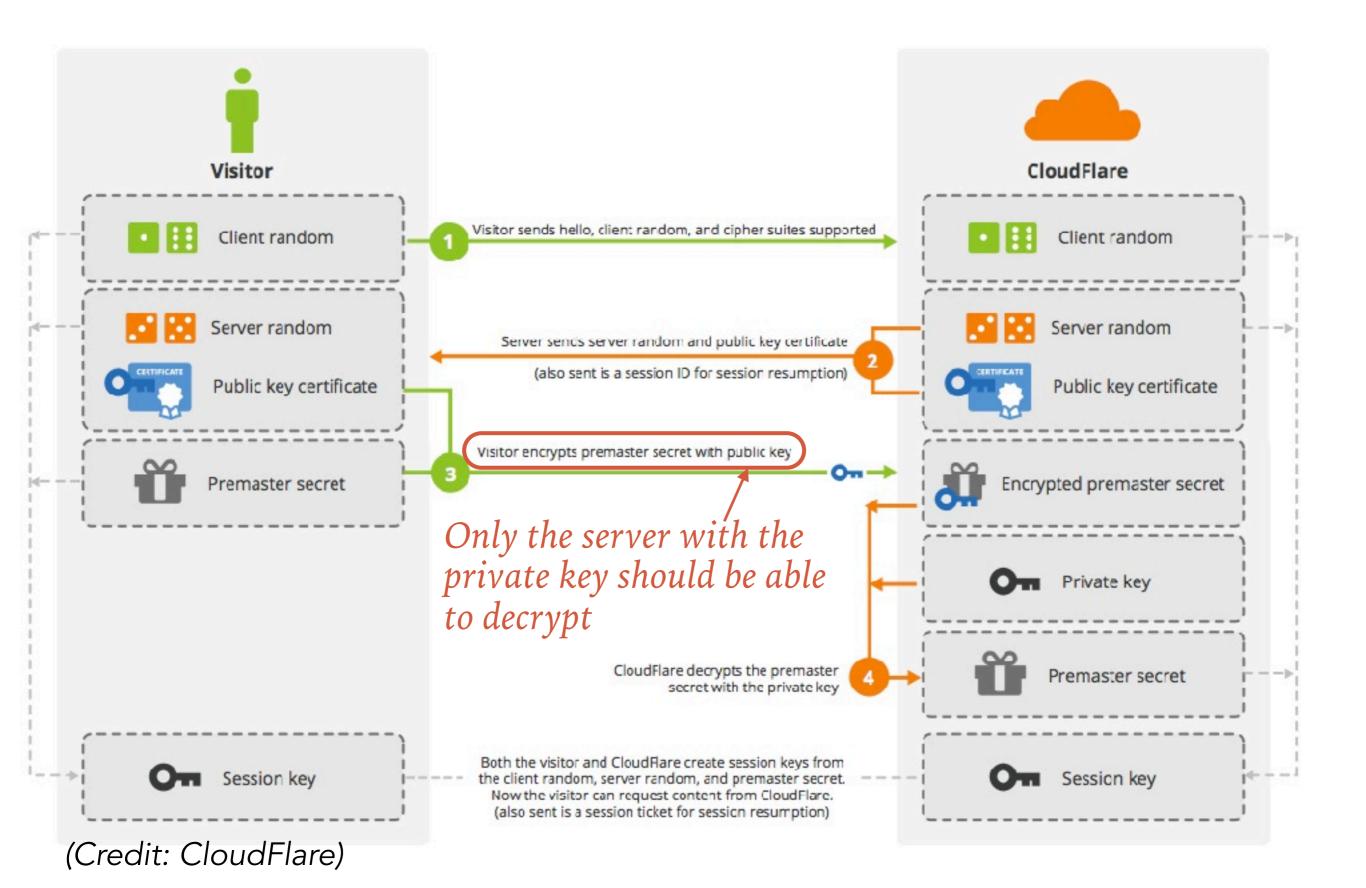
Browser (initiates connection)

Server (authenticates itself)



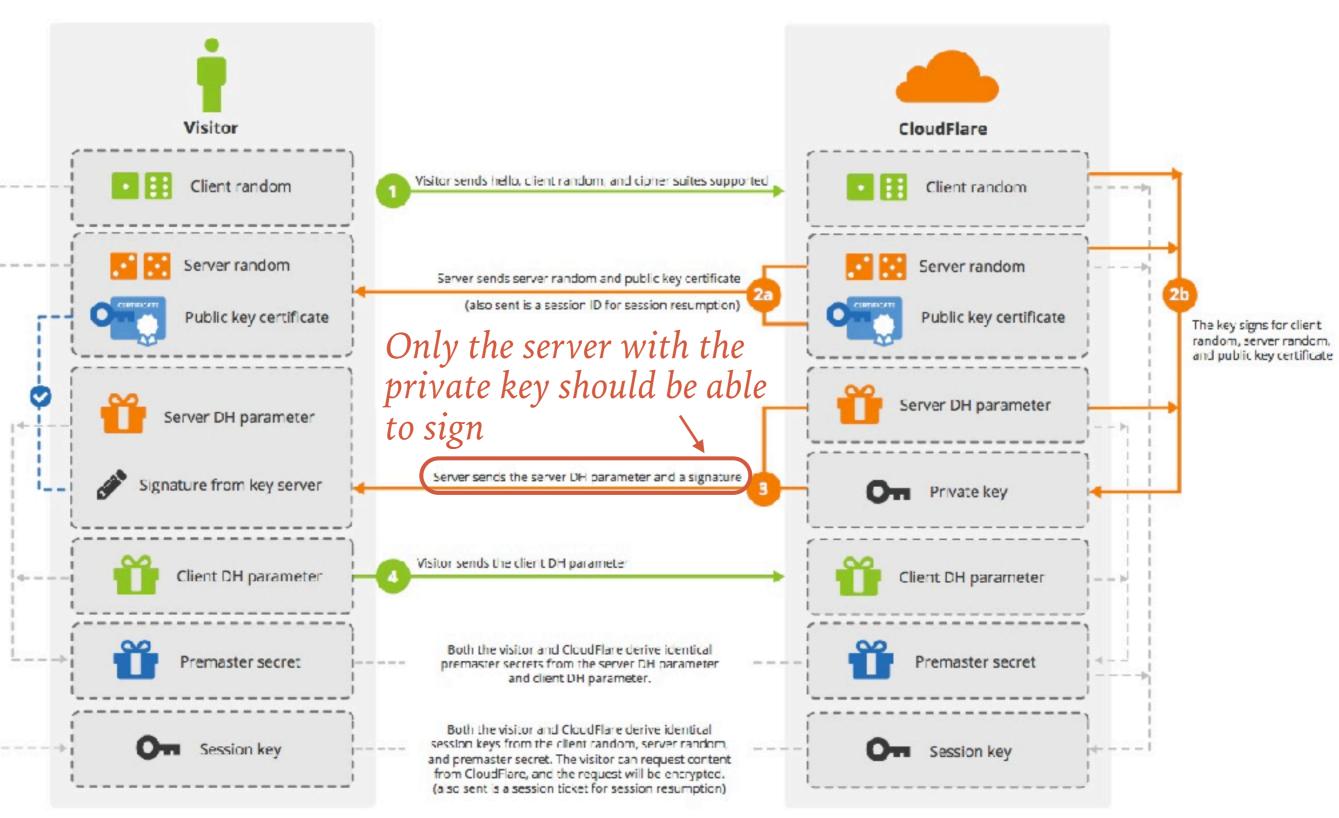
#### SSL Handshake (RSA)

Handshake



#### SSL Handshake (Diffie-Hellman)

Handshake



(Credit: CloudFlare)

### AUTHENTICATED DIFFIE-HELLMAN

Visitor encrypts premaster secret with public key Only the server with the private key should be able to decrypt
Only the server DH parameter and a signature 3
Only the server DH parameter and a signature 3
Only the server DH parameter and a signature 3
Only the server DH parameter and a signature 3
Only the server DH parameter and a signature 3
Only the server DH parameter and a signature

Both of these serve as a "challenge/response" protocol:

The client is "challenging" the server to prove that it knows the secret key corresponding to the public key in the certificate

The server is providing a "zero-knowledge proof":

The server proves that it knows the secret key without having to reveal the secret key itself

The key property that makes this work: The only person who knows the secret key is the entity in the certificate

# Certificate revocation

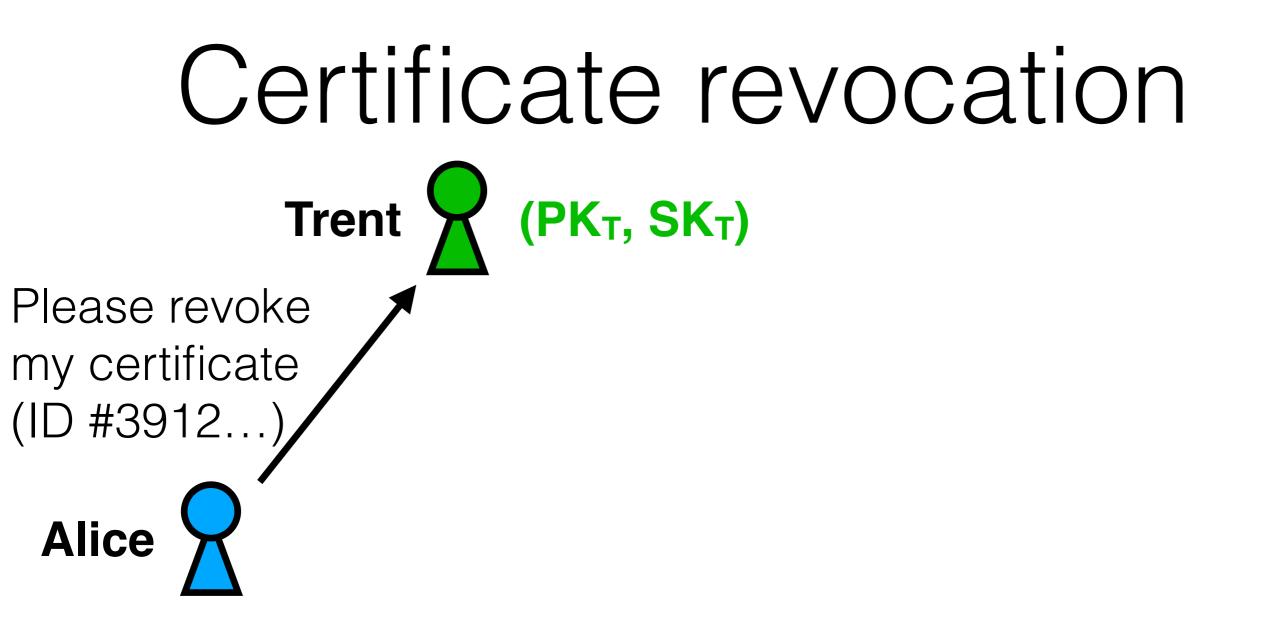
3. Trent *signs* a message (with **SK<sub>T</sub>**):

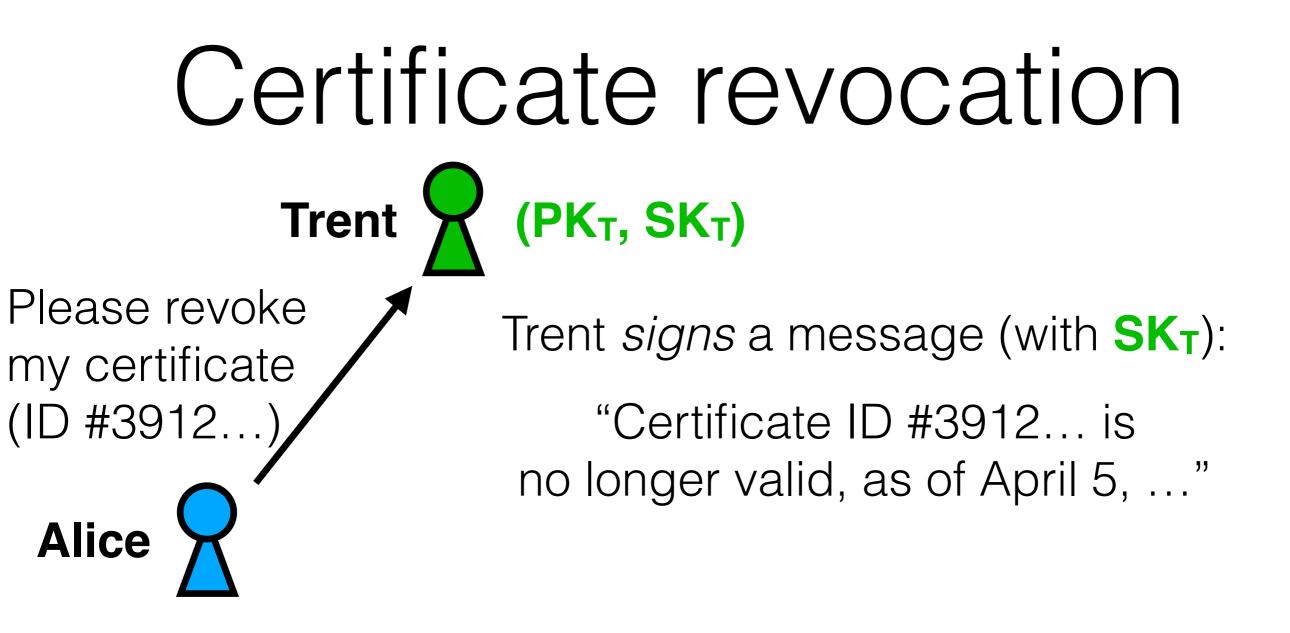
"The owner of the secret key corresponding to PKA is Alice"

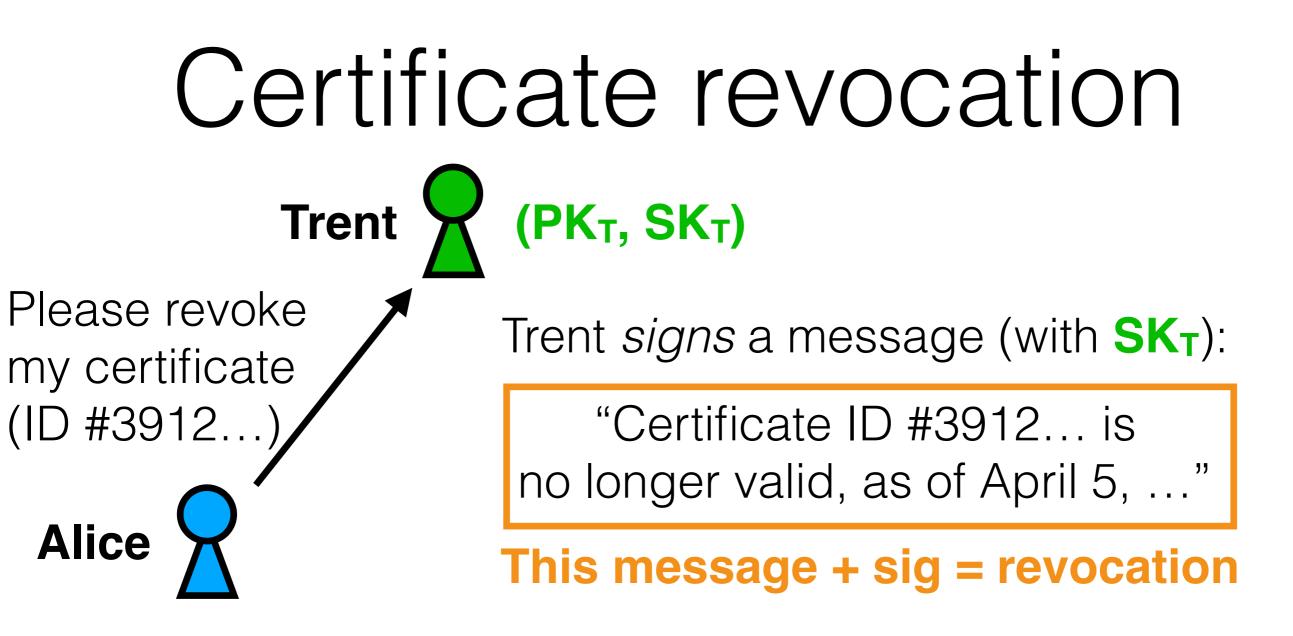
This message + sig = Certificate

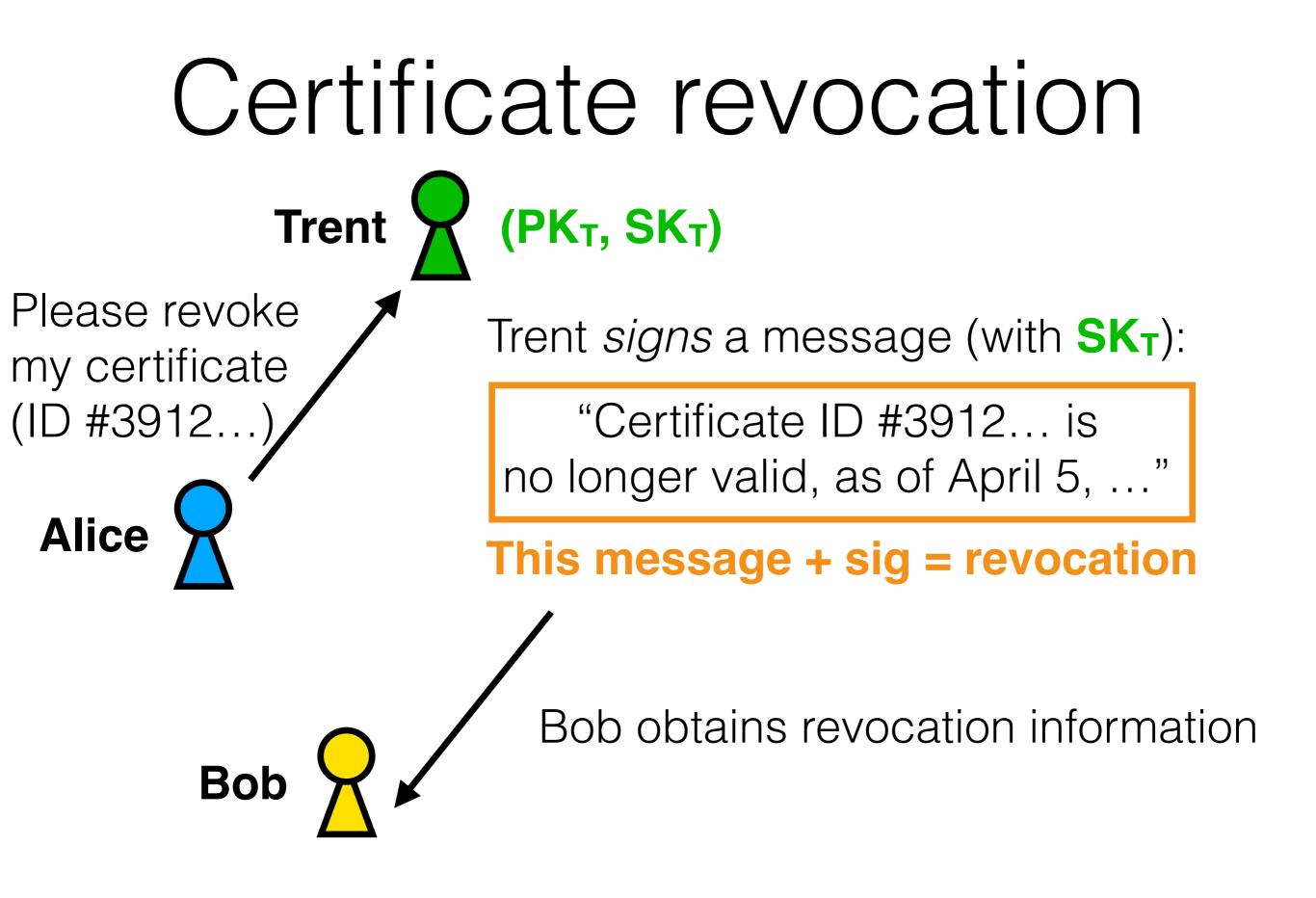
Put another way: "The only person who knows **SK**<sub>A</sub> is Alice"

What happens if Alice's key gets compromised? (Stolen, accidentally revealed, ...)









### Obtaining revocation data Certificate Revocation Lists (CRLs)

A (often large) signed list of revocations



"Certificate ID #3912... is no longer valid, as of April 5, ..."

### Obtaining revocation data Certificate Revocation Lists (CRLs)

A (often large) signed list of revocations



"Certificate ID #3912... is no longer valid, as of April 5, ..."

Browsers and OSes occasionally download CRLs



### Obtaining revocation data Certificate Revocation Lists (CRLs)

A (often large) signed list of revocations



"Certificate ID #3912... is no longer valid, as of April 5, ..."

Browsers and OSes occasionally download CRLs



**Disincentive**: CRLs can be large, so it takes time & bandwidth

# Obtaining revocation data Certificate Revocation Lists (CRLs)

A (often large) signed list of revocations



"Certificate ID #3912... is no longer valid, as of April 5, ..."

Browsers and OSes occasionally download CRLs



**Disincentive**: CRLs can be large, so it takes time & bandwidth

Result: delayed days/weeks/forever

# Obtaining revocation data Online Certificate Status Protocol (OCSP)

Browsers and OSes perform OCSP checks on-demand (when verifying the certificate)





# Obtaining revocation data Online Certificate Status Protocol (OCSP)

Browsers and OSes perform OCSP checks on-demand (when verifying the certificate)



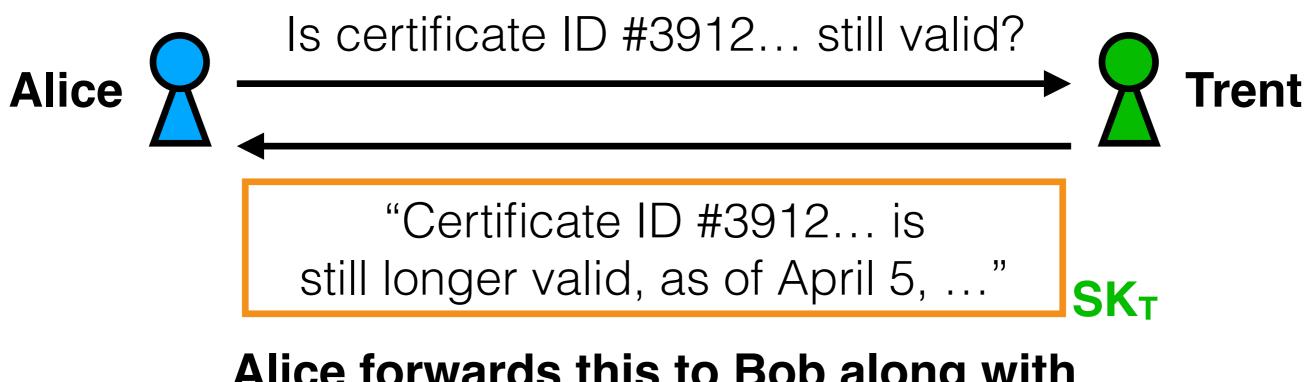
### Obtaining revocation data **Online Certificate Status Protocol (OCSP)** Browsers and OSes perform OCSP checks on-demand (when verifying the certificate) Is certificate ID #3912... still valid? Bob Trent "Certificate ID #3912... is still longer valid, as of April 5, ..." **SK**<sub>T</sub>

### Obtaining revocation data **Online Certificate Status Protocol (OCSP)** Browsers and OSes perform OCSP checks on-demand (when verifying the certificate) Is certificate ID #3912... still valid? Bob Trent "Certificate ID #3912... is still longer valid, as of April 5, ..." **SK**<sub>T</sub>

**Disincentive**: Still delays the initial validation of the certificate (can increase webpage load time)

# Obtaining revocation data ocsp Stapling

Websites issue OCSP requests, include responses in initial handshake



Alice forwards this to Bob along with the certificate when they first start to communicate

### Certificate revocation responsibilities



Alice's responsibility: Request revocations





**Bob's responsibility:** Check for revocations

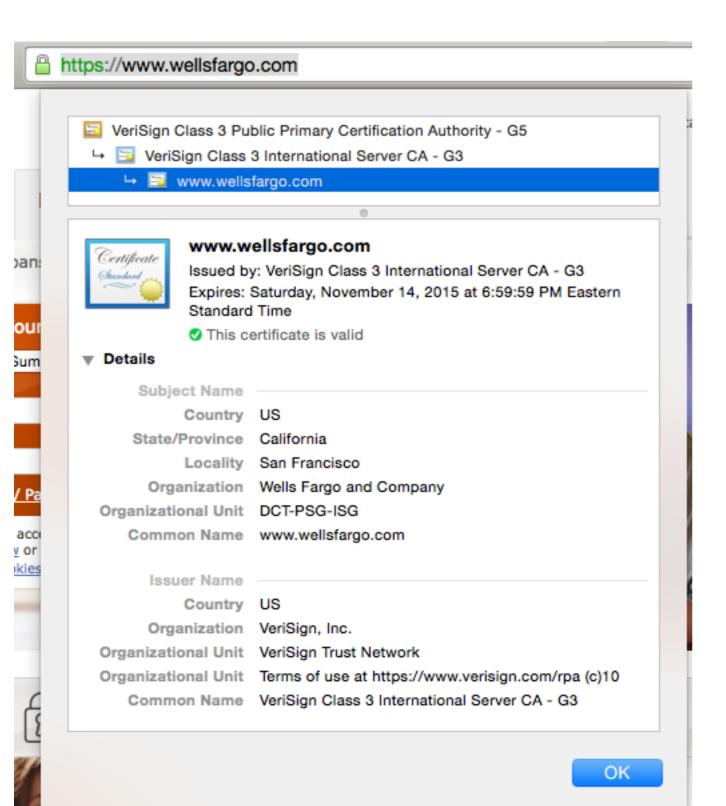
# Certificates in the wild

The lock icon indicates that the browser was able to authenticate the other end, i.e., validate its certificate



Bank of America Corporation [US] https://www.bankofamerica.com

	wellsfargo.com ty verified	5
Perr	missions Connection	
	The identity of this website has been verified by VeriSign Class 3 International Server CA - G3 but does not have public audit records. Certificate Information Your connection to www.wellsfargo.com is encrypted with obsolete cryptography. The connection uses TLS 1.2. The connection is encrypted using RC4_128, with SHA1 for message authentication and RSA as the key exchange mechanism.	5
	Site information	



### **Certificate chain**

**Subject** (who owns the public key)

**Common name:** the URL of the subject

**Issuer** (who verified the identity and signed this certificate)

Dalik wherever me takes you

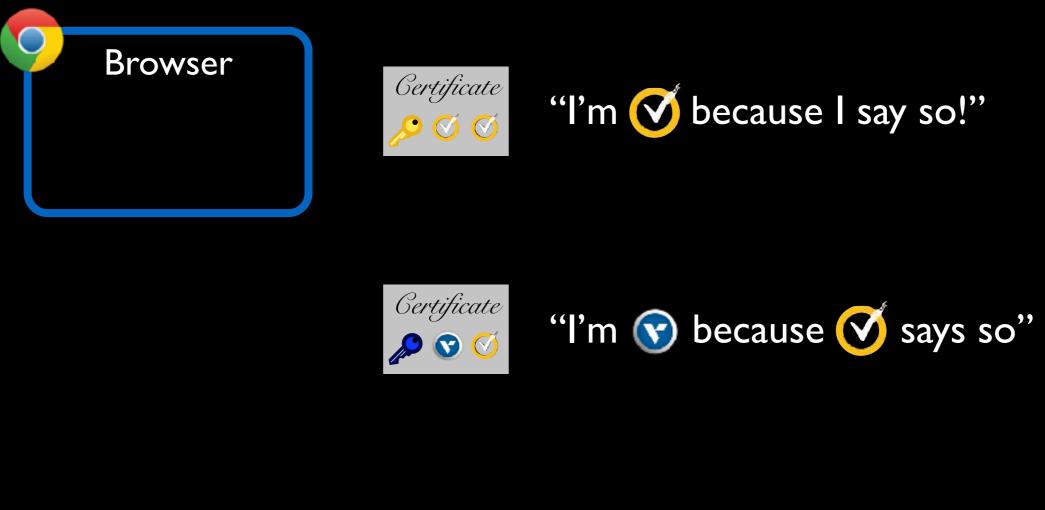








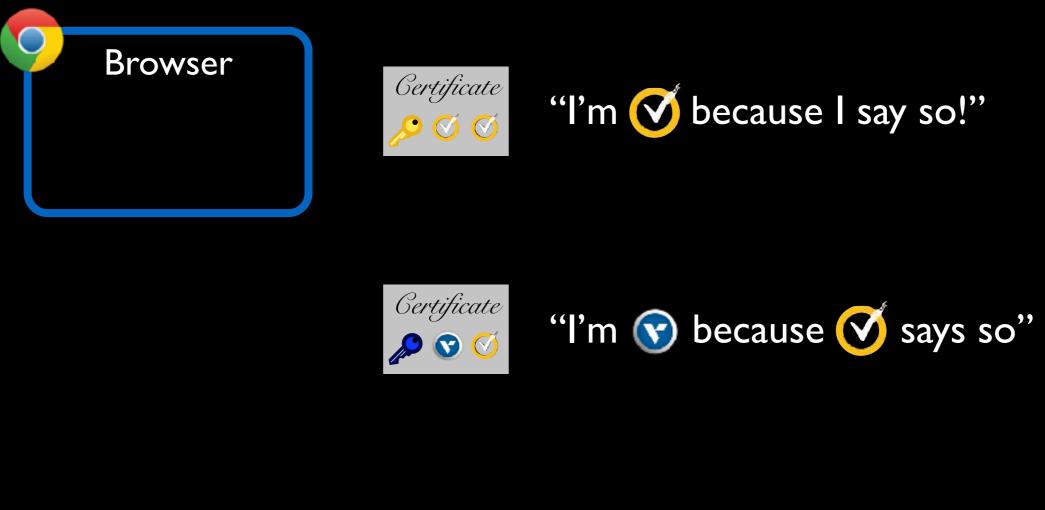




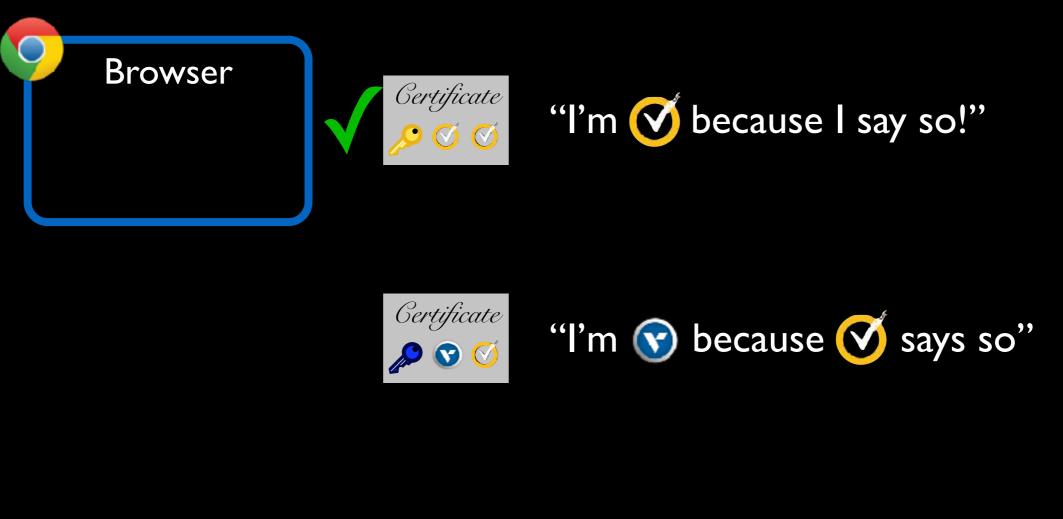


•		Keychain Access				
$\cap$	Click to unlock the	he System Roote keychain				Q Search
		unlock the System Roots keychain.				Q Cealon
~	Keychains					
	login	Certificate	Symantec Class 1 Public Primary Certificat	tion Authority - G4		
吊	iCloud	Red may	Root certificate authority	Circulard Time		
- 🗎	System		Expires: Monday, January 18, 2038 at 6:59:59 PM Es	actern Standard Time		
	System Roots		This certificate is valid			
		Name		^ Kind	Expires	Keychain
		Starfield	Class 2 Certification Authority	certificate	Jun 29, 2034, 1:39:16 PM	A System Hoots
		📷 Starfield	Root Certificate Authority - G2	certificate	Dec 31, 2037, 6:59:59 Pt	M System Roots
		📷 Starfield	Services Root Certificate Authority - G2	certificate	Dec 31, 2037, 6:59:59 Pt	M System Foots
		📴 StartCon	n Certification Authority	certificate	Sep 17, 2036, 3:46:36 PM	M System Roots
		📴 StartCon	n Certification Authority	certificate	Sep 17, 2036, 3:46:36 PM	M System Roots
		📴 StartCon	n Certification Authority G2	certificate	Dec 31, 2039, 6:59:01 PI	M System Foots
	Category	😂 Swissco	m Root CA 1	certificate	Aug 18, 2025, 6:06:20 PM	M System Roots
2	All Items	📴 Swissco	m Root CA 2	certificate	Jun 25, 2031, 3:38:14 AM	A System Roots
1	Passwords	Swissco	m Root EV CA 2	certificate	Jun 25, 2031, 4:45:08 AM	A System Foots
/		📴 SwissSig	gn CA (RSA IK May 6 1999 18:00:58)	certificate	Nov 26, 2031, 6:27:41 P	M System Roots
	Secure Notes	📷 SwissSig	an Gold CA - G2	certificate	Oct 25, 2036, 4:30:35 AM	A System Roots
	My Certificates	😋 SwissSig	gn Platinum CA - G2	certificate	Oct 25, 2036, 4:36:00 AM	A System Roots
Ŷ	Keys	📷 SwissSig	an Silver CA - G2	certificate	Oct 25, 2036, 4:32:46 AM	A System Foots
	Certificates	📉 📴 Symante	c Class 1 Public Primary Certification Authority - G4	certificate	Jan 18, 2038, 6:59:59 PM	A System Foots
		📴 Symante	c Class 1 Public Primary Certification Authority - G6	certificate	Dec 1, 2037, 5:59:59 PM	System Foots
		📴 Symante	c Class 2 Public Primary Certification Authority - G4	certificate	Jan 18, 2038, 6:59:59 PM	A System Roots
		📴 Symante	c Class 2 Public Primary Certification Authority - G6	certificate	Dec 1, 2037, 5:59:59 PM	System Foots
		📴 Symante	c Class 3 Public Primary Certification Authority - G4	certificate	Dec 1, 2037, 5:59:59 PM	System Roots
		📴 Symante	c Class 3 Public Primary Certification Authority - G6	certificate	Dec 1, 2037, 5:59:59 PM	System Roots
		SZAFIR	ROOT CA	certificate	Dec 6, 2031, 5:10:57 AM	System Roots
		🔚 T-TeleSe	c GlobalRoot Class 2	certificate	Oct 1, 2033, 7:59:59 PM	System Roots
		🛅 T-TeleSe	c GlobalRoot Class 3	certificate	Oct 1, 2033, 7:59:59 PM	System Roots
		📴 TC Trust	Center Class 2 CA II	certificate	Dec 31, 2025, 5:59:59 PM	M System Roots
		📷 TC Trust	Center Class 3 CA II	certificate	Dec 31, 2025, 5:59:59 PM	M System Roots
		TC Trust	Center Class 4 CA II	certificate	Dec 31, 2025, 5:59:59 Pt	M System Roots
		TC Trust	Center Universal CA I	certificate	Dec 31, 2025, 5:59:59 Pt	M System Roots
		🛅 TC Trust	Center Universal CA II	certificate	Dec 31, 2030, 5:59:59 PM	M System Roots
			Center Universal CA III	certificate	Dec 31 2029 6:59:59 P	M System Boots
		+ i Cop	Ŷ	210 items		

Keychain Access				
Click to unlock the	System Roots keychain.			Q Search
Keychains Iogin ICloud System System Roots	Certificate Configurate Configurate Certificate Control Class 1 Public Primary Certificate Root certificate authority Expires: Monday, January 18, 2038 at 6:59:59 PM Eac This certificate is valid	-		
	Name Starfield Class 2 Certification Authority - G2 Starfield Root Certificate Authority - G2 Starfield Services Root Certificate Authority - G2 StartCom Certification Authority StartCom Certification Authority StartCom Certification Authority G2	vice has c	Dec 31, 2037, 6:59:59 PM	Keychain System Roots System Roots System Roots System Roots System Roots System Roots
Category All Items Category All Items Secure Notes My Certificates Keys	Swisscom Root CA 1 Swisscom Root CA 2 Swisscom Root EV CA 2 SwissSign CA (RSA IK May 6 1999 malicious SwissSign Gold CA - G2 SwissSign Platinum CA - G2 SwissSign Silver CA - G2	certificate certificate certificate	JUN 20, 2001, 4:40:00 AM	System Roots System Roots System Roots System Roots System Roots System Roots System Roots
Certificates	Symantee Cass 1 Public Primary Certification Authority - G4         Symantee Class 1 Public Primary Certification Authority - G6         Symantee Class 2 Public Primary Certification Authority - G4         Symantee Class 2 Public Primary Certification Authority - G6         Symantee Class 3 Public Primary Certification Authority - G6         Symantee Class 3 Public Primary Certification Authority - G6         Symantee Class 3 Public Primary Certification Authority - G6         Symantee Class 3 Public Primary Certification Authority - G6         Symantee Class 3 Public Primary Certification Authority - G6         Symantee Class 3 Public Primary Certification Authority - G6         Symantee Class 3 Public Primary Certification Authority - G6         Symantee Class 3 Public Primary Certification Authority - G6         Symantee Class 3 Public Primary Certification Authority - G6         Symantee Class 3 Public Primary Certification Authority - G6         Symantee Class 3 Public Primary Certification Authority - G6         T-TeleSec GlobalRoot Class 2         T-TeleSec GlobalRoot Class 3         TC TrustCenter Class 2 CA II         TC TrustCenter Class 3 CA II         TC TrustCenter Universal CA II         TC TrustCenter Universal CA II         TC TrustCenter Universal CA II	certificate         certificate	Jan 18, 2038, 6:59:59 PM Dec 1, 2037, 6:59:59 PM Jan 18, 2038, 6:59:59 PM Dec 1, 2037, 5:59:59 PM Dec 1, 2037, 5:59:59 PM Dec 1, 2037, 6:59:59 PM Dec 1, 2037, 6:59:59 PM Dec 6, 2031, 6:10:57 AM Oct 1, 2033, 7:59:59 PM Oct 1, 2033, 7:59:59 PM Dec 31, 2025, 5:59:59 PM	System FootsSystem RootsSystem Roots
	TC TrustCenter Universal CA III + j Copy	210 items	Dec 31 2029 6:59:59 PM	System Boots













































https://www.wellsfargo.com

ban

oui

Sum

acci v or

kies

27	VeriSign	Class	3 Public	Primary	Certification	Authority - (	G5
----	----------	-------	----------	---------	---------------	---------------	----

- VeriSign Class 3 International Server CA G3

Serial Number Version	6E A3 6F 06 1A 46 2C 78 9D 10 DE A2 22 15 C9 E4 3
Signature Algorithm Parameters	SHA-1 with RSA Encryption (1.2.840.113549.1.1.5) none
Not Valid Before	Wednesday, November 12, 2014 at 7:00:00 PM Eastern Standard Time
Not Valid After	Saturday, November 14, 2015 at 6:59:59 PM Eastern Standard Time
Public Key Info	
Algorithm	RSA Encryption (1.2.840.113549.1.1.1)
Parameters	none
Public Key	256 bytes : E1 9D 53 21 ED 6A DD 67
Exponent	65537
Key Size	2048 bits
Key Usage	Encrypt, Verify, Wrap, Derive
Signature	256 bytes : 55 0F FF 63 93 EA 76 AA

Serial number: Uniquely identifies this cert with respect to the issuer (look for this in CRLs)

## Signature algorithm: How the issuer will sign parts of the cert

Not valid before/after: When to start and stop believing this cert (start & expiration dates)

The public key: And the issuer's signature of the public key

ank wherever me takes you

OK

#### https://www.wellsfargo.com

an

วนเ

um

acco or kies VeriSign Class 3 Public Primary Certification Authority - G5

	0
Extension	Subject Alternative Name (2.5.29.17)
Critical	NO
DNS Name	www.wellsfargo.com
Extension	Certificate Policies (2.5.29.32)
Critical	NO
Policy ID #1	(2.16.840.1.113733.1.7.54)
Qualifier ID #1	Certification Practice Statement (1.3.6.1.5.5.7.2.1)
CPS URI	https://d.symcb.com/cps
Qualifier ID #2	User Notice ( 1.3.6.1.5.5.7.2.2 )
User Notice	https://d.symcb.com/rpa
Extension	CRL Distribution Points (2.5.29.31)
Critical	NO
URI	http://se.symcb.com/se.crl
Extension	Certificate Authority Information Access (1.3.6.1.5.5.7.1.1)
Critical	NO
Method #1	
URI	http://se.symcd.com
Method #2	,
URI	http://se.symcb.com/se.crt
Fingerprints	
	CA 7B 01 AF B9 DC 1F B5 E7 3C 4A 50 0C 79 3E 74
ULA I	10 E2 44 FF
MD5	70 8B C2 CB 22 06 65 C2 37 B7 C2 E5 90 F7 FA 5C
	ОК

### **Subject Alternate Names:**

Other URLs for which this cert should be considered valid. (<u>wellsfargo.com</u> is not the same as <u>www.wellsfargo.com</u>)

Can include wildcards, e.g., \*.google.com

#### Inttps://www.chick-fil-a.com

G

#### 📴 GlobalSign Root CA

- → 🔄 GlobalSign CloudSSL CA SHA256 G3
  - 🛏 🛅 incapsula.com



#### incapsula.com

Issued by: GlobalSign CloudSSL CA - SHA256 - G3 Expires: Wednesday, August 1, 2018 at 6:10:26 AM Eastern Daylight Time This certificate is valid

#### Details

Subject Name Country US State/Province Delaware Locality Dover Organization Incapsula Inc Common Name incapsula.com **Issuer Name** Country BE Organization GlobalSign nv-sa Common Name GlobalSign CloudSSL CA - SHA256 - G3 Serial Number 10 7D 89 FE DA 24 BD ED 35 83 B7 29 Version 3 Signature Algorithm SHA-256 with RSA Encryption (1.2.840.113549.1.1.11) Parameters none Not Valid Before Wednesday, August 23, 2017 at 4:56:15 PM Eastern Daylight Time Not Valid After Wednesday, August 1, 2018 at 6:10:26 AM Eastern Daylight Time Public Key Info Algorithm RSA Encryption (1.2.840.113549.1.1.1) Parameters none Public Key 256 bytes : CF 70 70 52 92 AB 2E 36 ... Exponent 65537

### **Subject Alternate Names:**

The *spirit* is that it represents different domain names of the same entity (google.com, google.co.uk, youtube.com, ...)

The *letter* of the rule doesn't say that they need to be the same company—or really have anything in common

#### <u>ج</u> 2 ☆

#### 📴 GlobalSign Root CA

- → 
  GlobalSign CloudSSL CA SHA256 G3
  - 🛏 🛅 incapsula.com

Extension	Subject Alternative Name ( 2.5.29.17 )
Critical	NO
DNS Name	incapsula.com
DNS Name	*.70trades.net
DNS Name	*.acquapanna.com
DNS Name	*.afrique360.com
DNS Name	*.alghanim.com
DNS Name	*.ashbree.com.au
DNS Name	*.atacadao.com.br
DNS Name	*.beautysolutions.com.au
DNS Name	*.biglegalminds.com
DNS Name	*.borsapro.com
DNS Name	*.bsp2012.iata.or.kr
DNS Name	*.cescomarketing.com
DNS Name	*.cgt.it
DNS Name	*.chick-fil-a.com
DNS Name	*.coleccion.caixaforum.com
DNS Name	*.corporateaffairs.tv
DNS Name	*.critical-intelligence.com
DNS Name	*.cultivatenext.org
DNS Name	*.dogchow.ca
DNS Name	*.electronicthecorporatecounsel.com
DNS Name	*.enrich.malaysiaairlines.com
DNS Name	*.exporters.mod.gov.il
DNS Name	*.financikatrade.ae
DNS Name	*.financikatrade.net
DNS Name	*.girona.cat
DNS Name	*.greennghetto.org
DNS Name	*.hatio.com
DNS Name	.meettherealme.org

### **Subject Alternate Names:**

The *spirit* is that it represents different domain names of the same entity (google.com, google.co.uk, youtube.com, ...)

The *letter* of the rule doesn't say that they need to be the same company—or really have anything in common

#### https://www.wellsfargo.com

an

วน

um

aco

r or

kies

VeriSign Class 3 Public Primary Certification Authority - G5

- VeriSign Class 3 International Server CA G3

Subject Alternative Name (2.5.29.17) Extension Critical NO DNS Name www.wellsfargo.com Extension Certificate Policies (2.5.29.32) Critical NO Policy ID #1 (2.16.840.1.113733.1.7.54) Qualifier ID #1 Certification Practice Statement (1.3.6.1.5.5.7.2.1) CPS URI https://d.symcb.com/cps Qualifier ID #2 User Notice (1.3.6.1.5.5.7.2.2) User Notice https://d.symcb.com/rpa Extension CRL Distribution Points (2.5.29.31) Critical NO http://se.symcb.com/se.crl URI Extension Certificate Authority Information Access (1.3.6.1.5.5.7.1.1)Critical NO Method #1 Online Certificate Status Protocol (1.3.6.1.5.5.7.48.1) URI http://se.symcd.com CA Issuers (1.3.6.1.5.5.7.48.2) Method #2 http://se.symcb.com/se.crt URI Fingerprints SHA1 CA 7B 01 AF B9 DC 1F B5 E7 3C 4A 50 0C 79 3E 74 10 E2 44 FF MD5 70 8B C2 CB 22 06 65 C2 37 B7 C2 E5 90 F7 FA 5C OK

### **Subject Alternate Names:**

Other URLs for which this cert should be considered valid. (<u>wellsfargo.com</u> is not the same as <u>www.wellsfargo.com</u>)

Can include wildcards, e.g., \*.google.com

### CRL & OCSP:

Where to go to check if this certificate has been revoked

### Non-cryptographic checksums

# Certificate types

Certificates can be classified in two broad ways

What the certificate can be used for

Signing (root and intermediate certs) Encrypting (leaf certs)

The type of vetting process used

### **DV (Domain validation)**

Prove administrative access to the domain, e.g., by uploading a file

### **OV (Organization validation)**

Prove ownership of the organization that owns the domain

**EV (Extended validation)** More extensive validation (\$\$)

# Certificate types

### Why are these different?

https://www.wellsfargo.com

Bank of America Corporation [US] https://www.bankofamerica.com

# Certificate types

### Why are these different?

https://www.wellsfargo.com	Bank of America Corporation [US] https://www.bankofamerica.com
https://www.wellsfargo.com	Bank of America Corporation [US] https://www.bankofamerica.com
<ul> <li>VeriSign Class 3 Public Primary Certification Authority - G5</li> <li>VeriSign Class 3 International Server CA - G3</li> <li>Www.wellsfargo.com</li> </ul> Anticipation of the server can be served as a server can be server can be served as a server can be served as a server can be server can be served as a server can be server can	A VeriSign Class 3 Public Primary Certification Authority - G5 → Symantec Class 3 EV SSL CA - G3 →  Symantec Class 3 EV SSL CA - G3 Intermediate certificate authority Expires: Monday, October 30, 2023 at 7:59:59 PM Eastern Dayight Time → This certificate is valid → Details
OK	т ds···· creait cara — кеwaras···· creait cara — вајалсе

This is an EV (extended validation) certificate; browsers show the full name for these kinds of certs

## Proper reaction to Heartbleed

1. Patch the software

2. "Reissue" a new key (get a new one and load it onto your servers)

3. Revoke the old key

# Proper reaction to Heartbleed

1. Patch the software

2. "Reissue" a new key (get a new one and load it onto your servers)

3. Revoke the old key

Order matters!

If we reissued and then patched, then our new key would be compromised, too.

If we revoked first, we'd be offline.



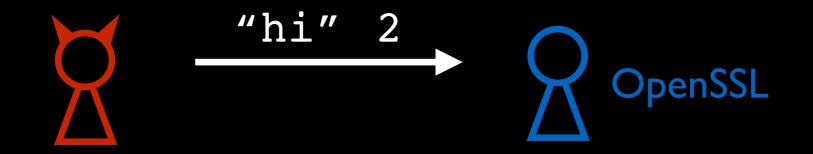






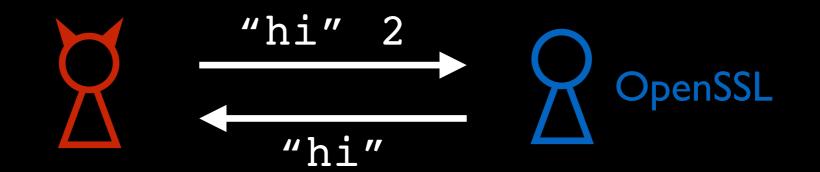








Heartbleed











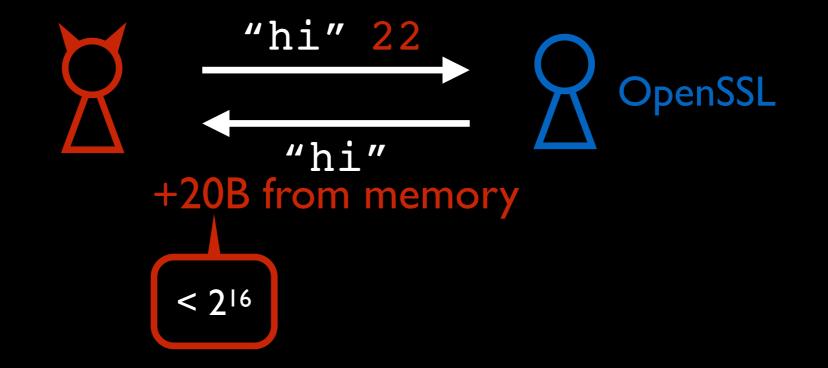






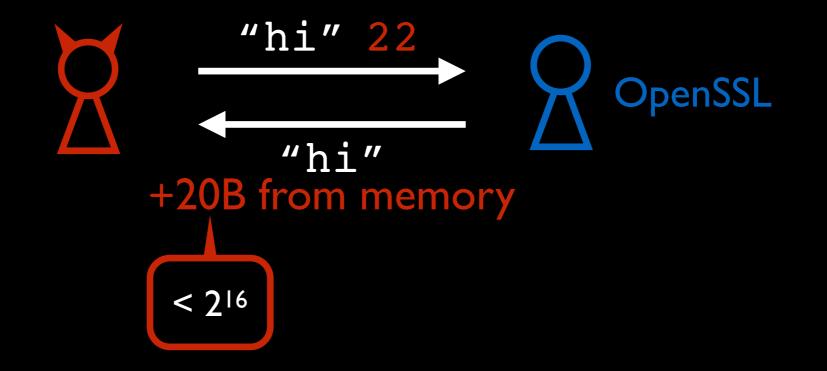










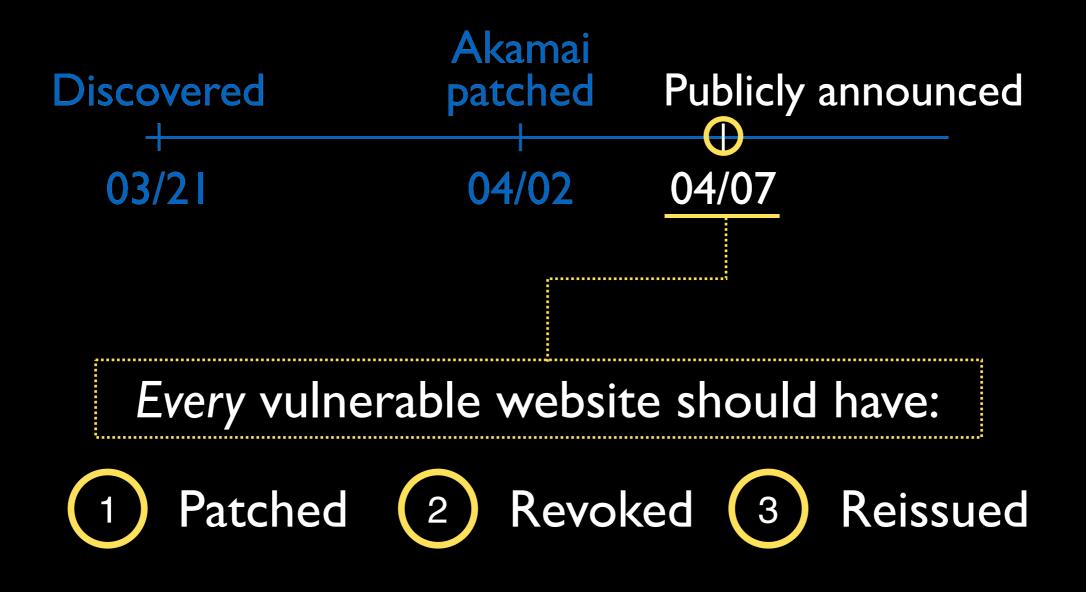


Potentially reveals user data and private keys Heartbleed exploits were undetectable

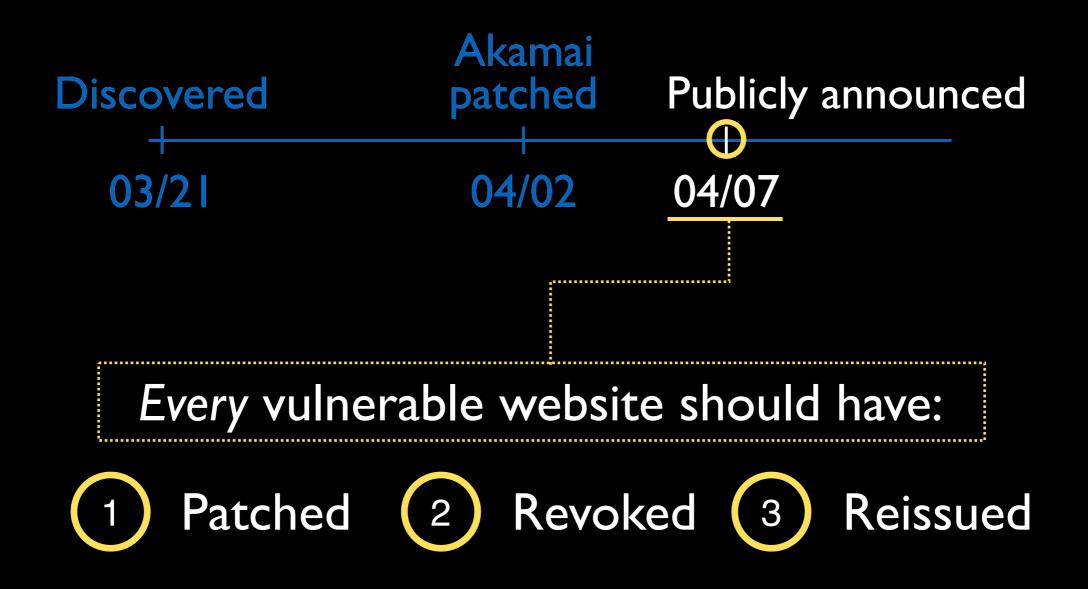
## Why study Heartbleed?



## Why study Heartbleed?

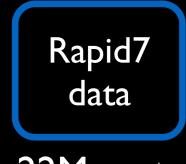


## Why study Heartbleed?

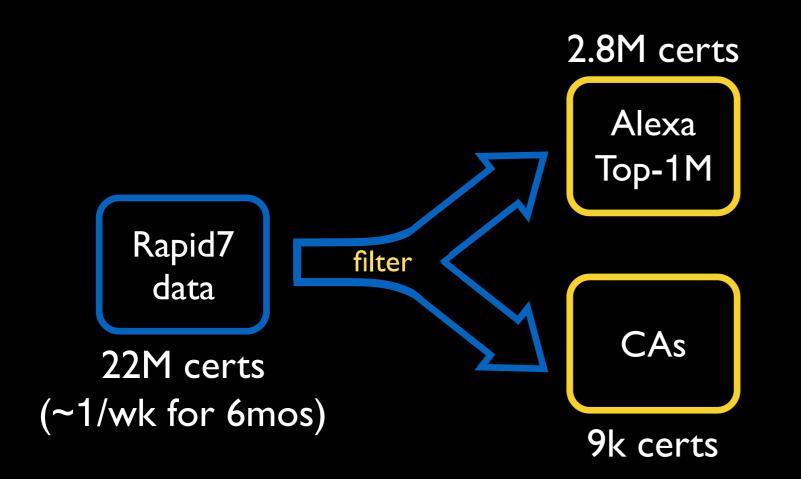


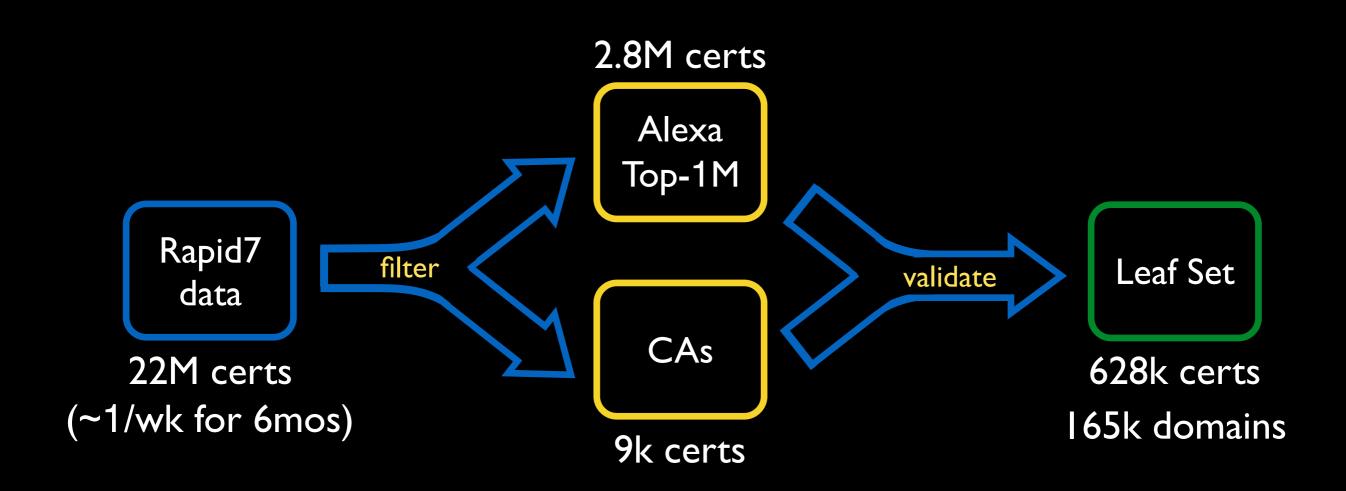
Heartbleed is a natural experiment: How quickly and thoroughly do administrators act?

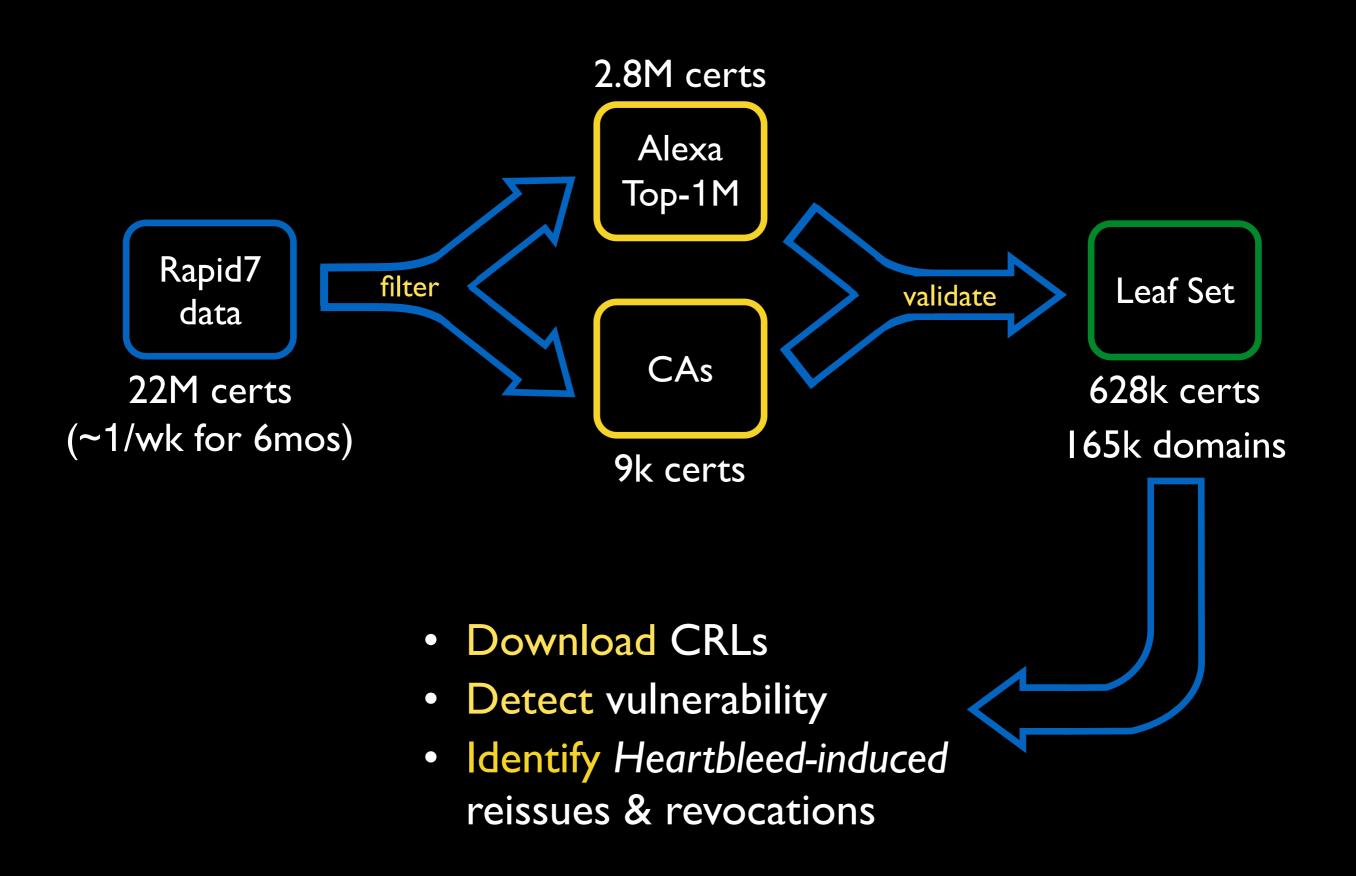


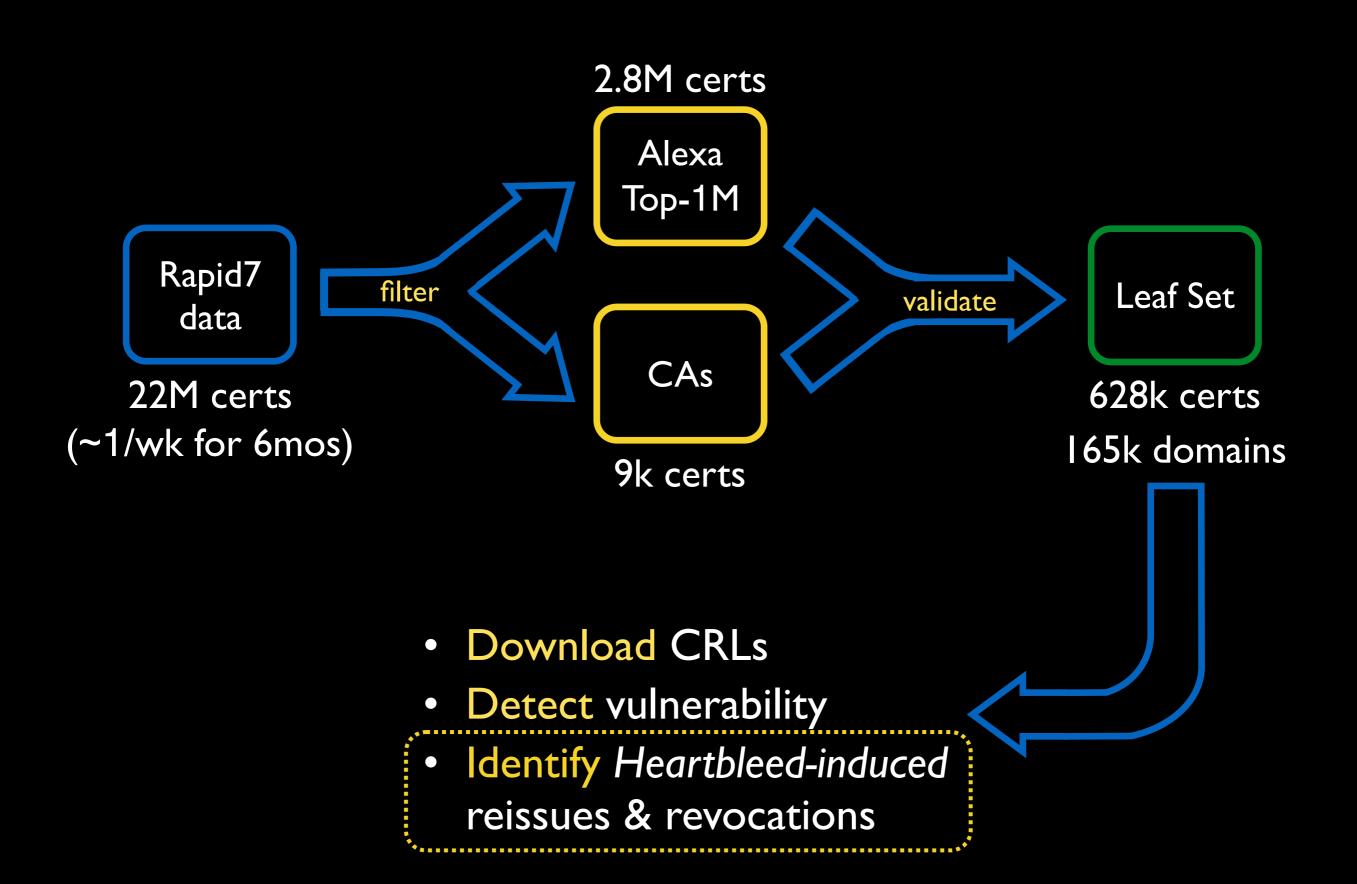


#### 22M certs (~1/wk for 6mos)

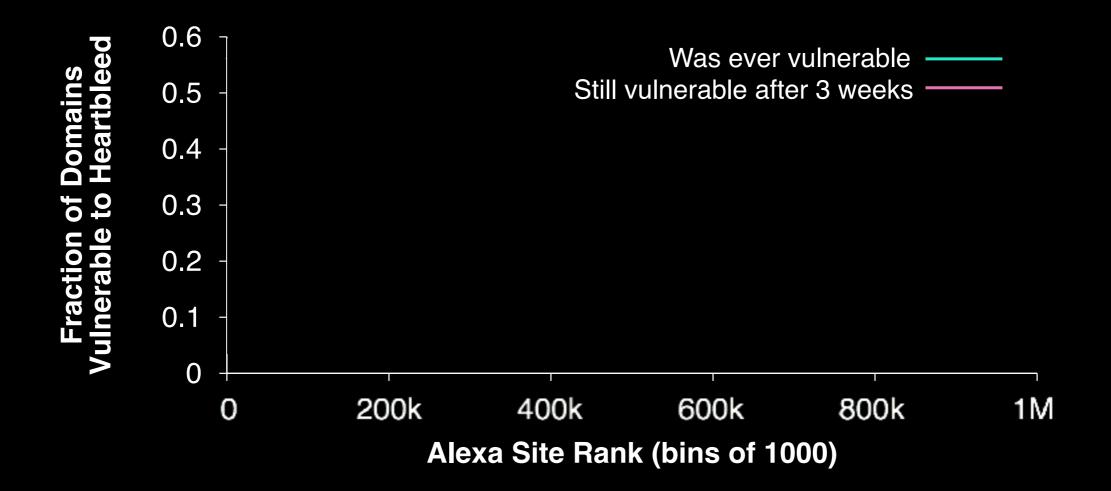




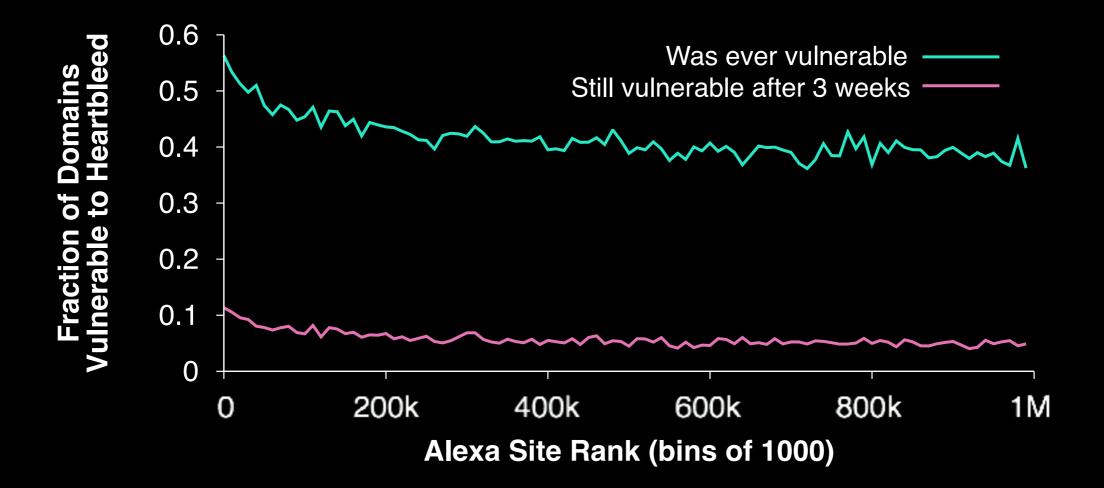




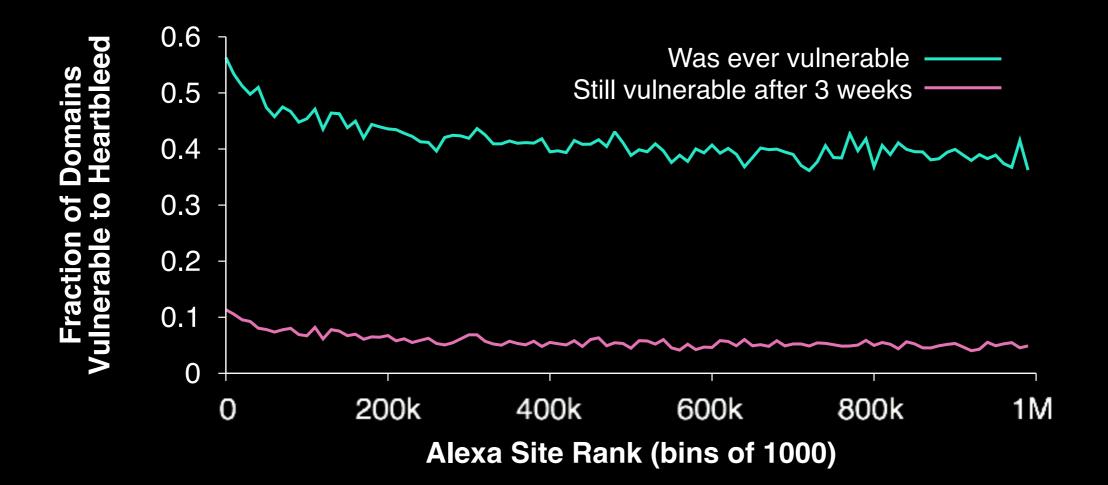
### Prevalence and patch rates



### Prevalence and patch rates



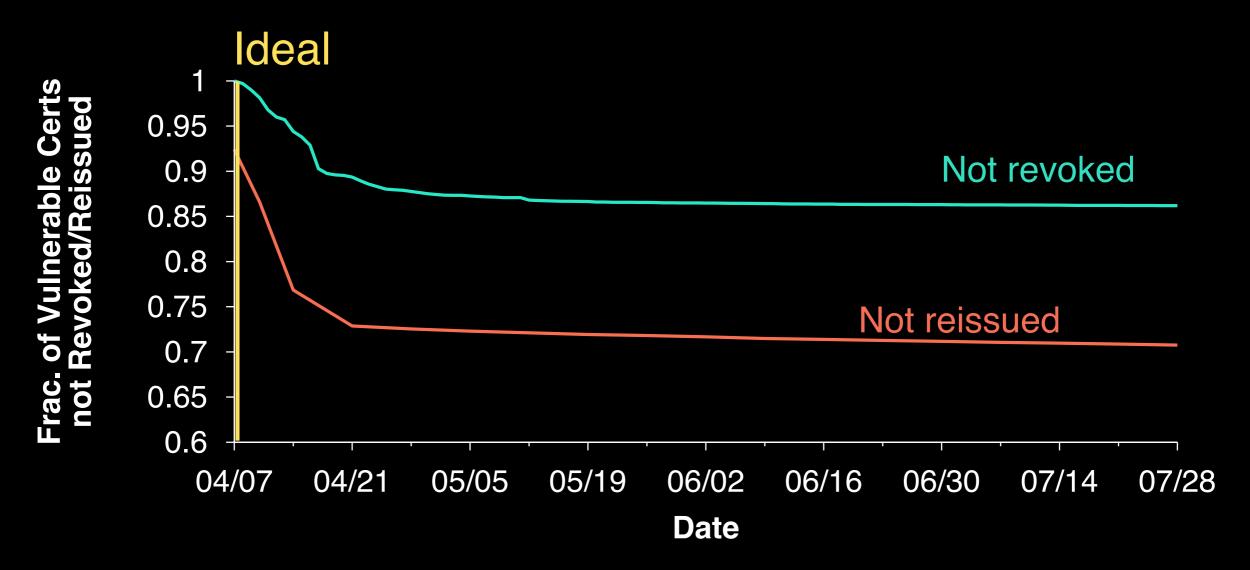
### Prevalence and patch rates

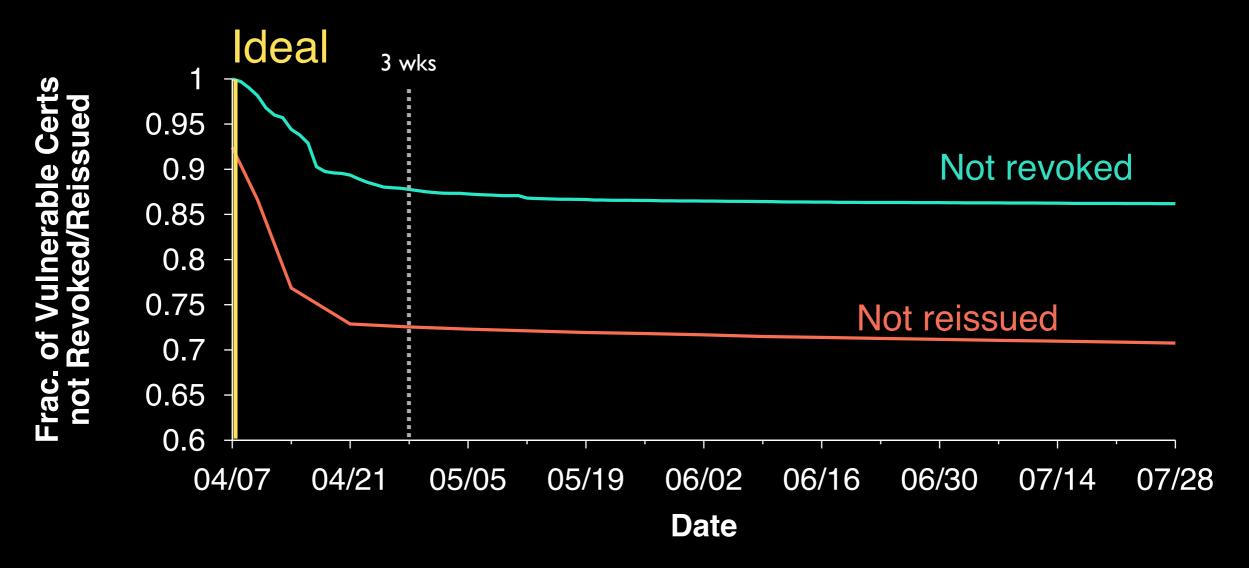


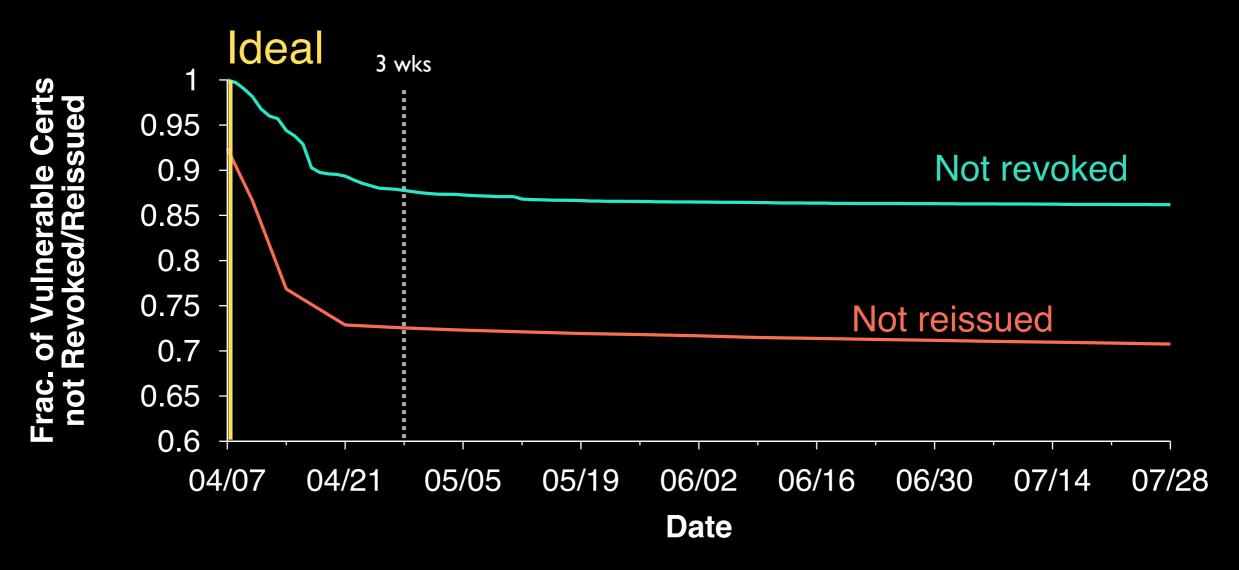
Patching rates are mostly positive Only ~7% had not patched within 3 weeks









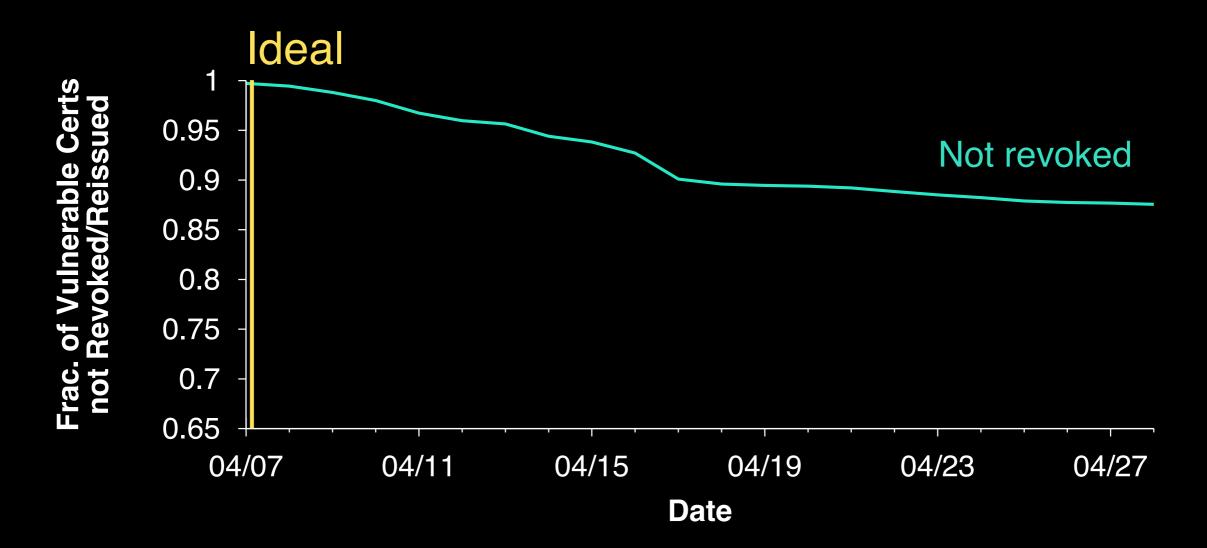


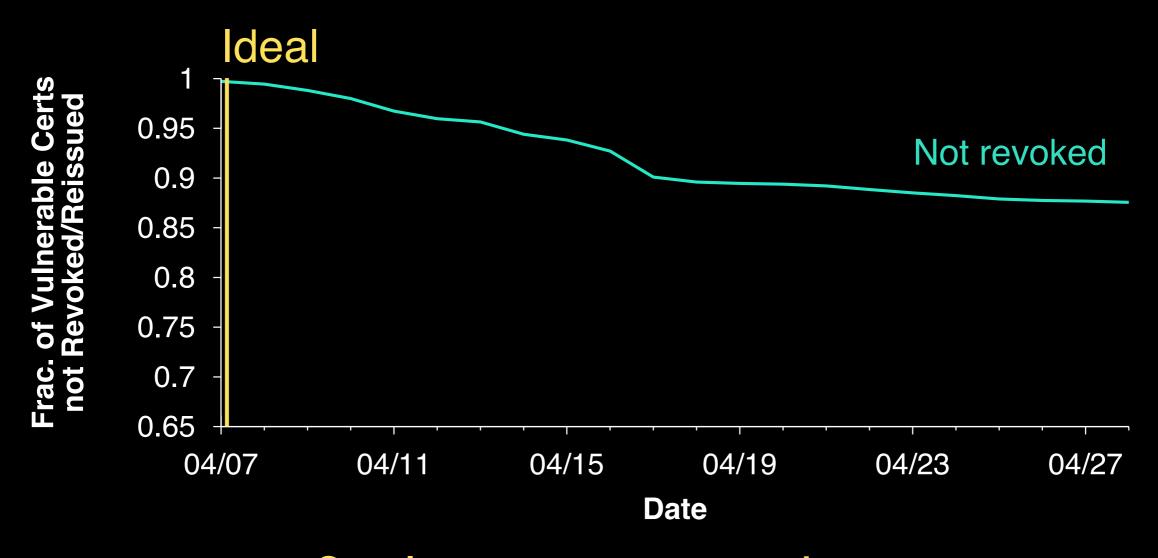
#### Similar pattern to patches: Exponential drop-off, then levels out



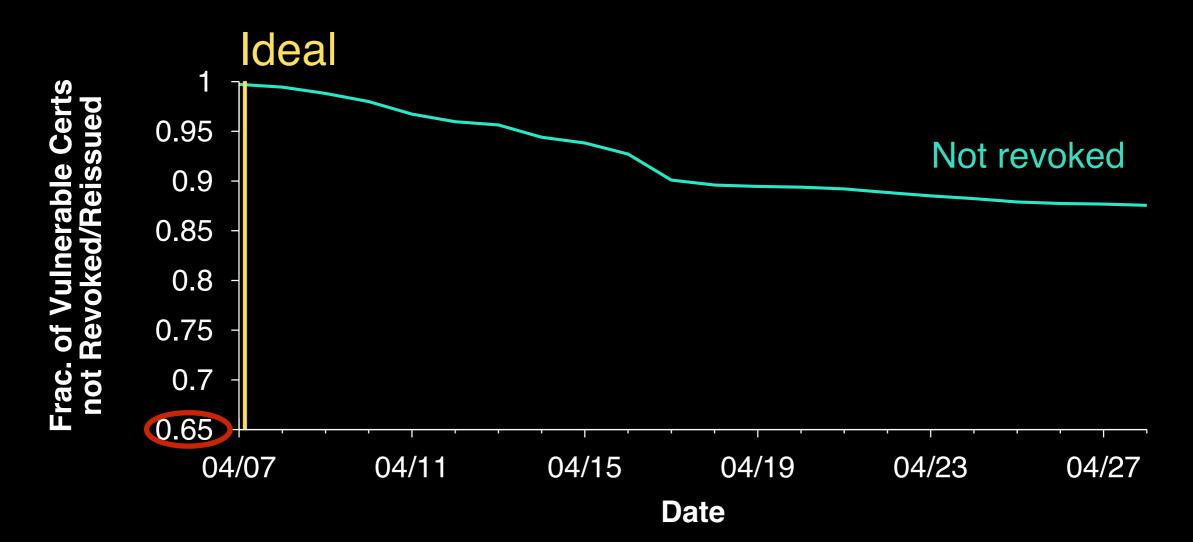






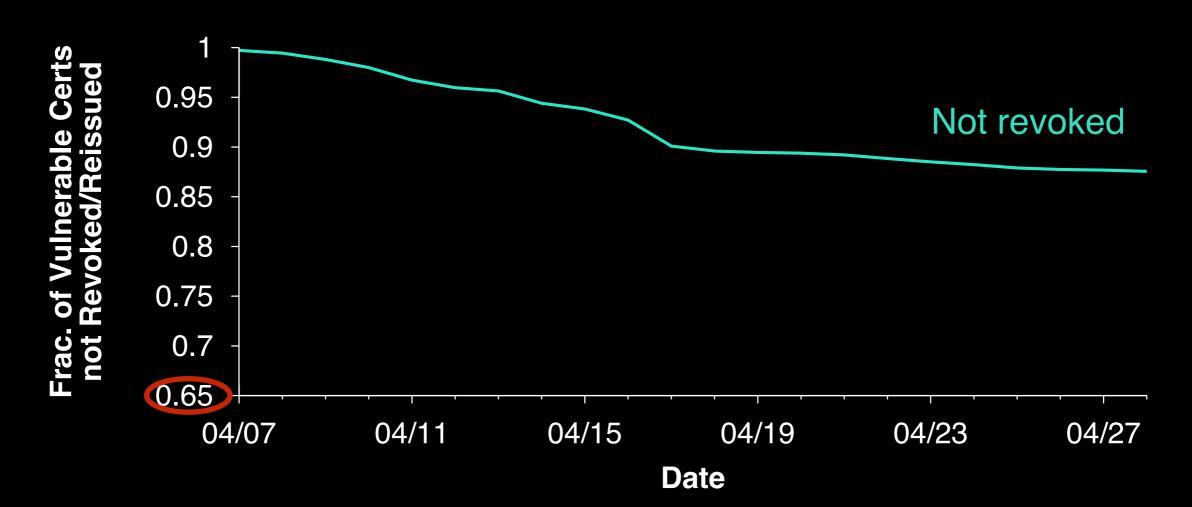


Similar pattern to patches: Exponential drop-off, then levels out



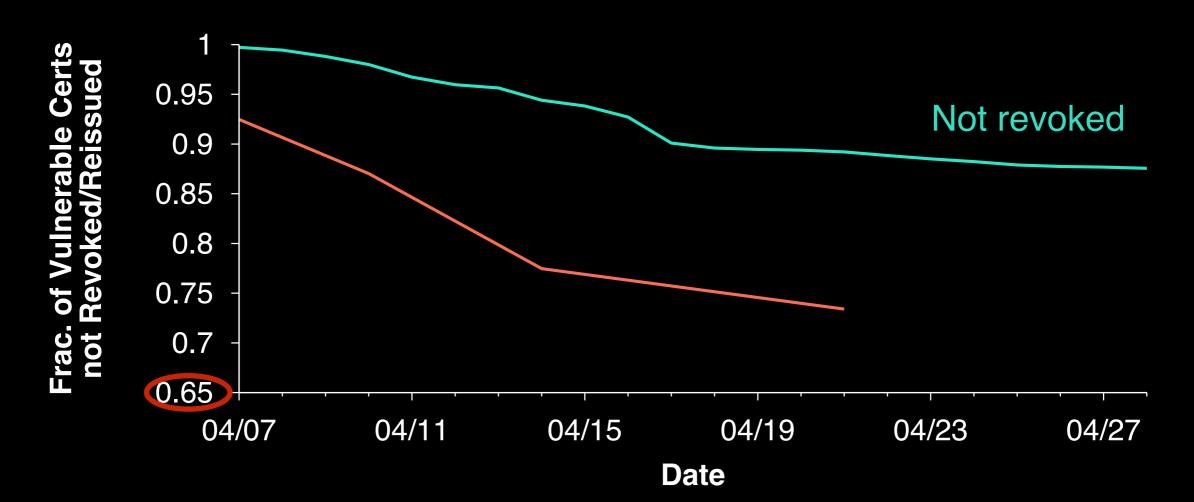
### Similar pattern to patches: Exponential drop-off, then levels out





### Similar pattern to patches: Exponential drop-off, then levels out



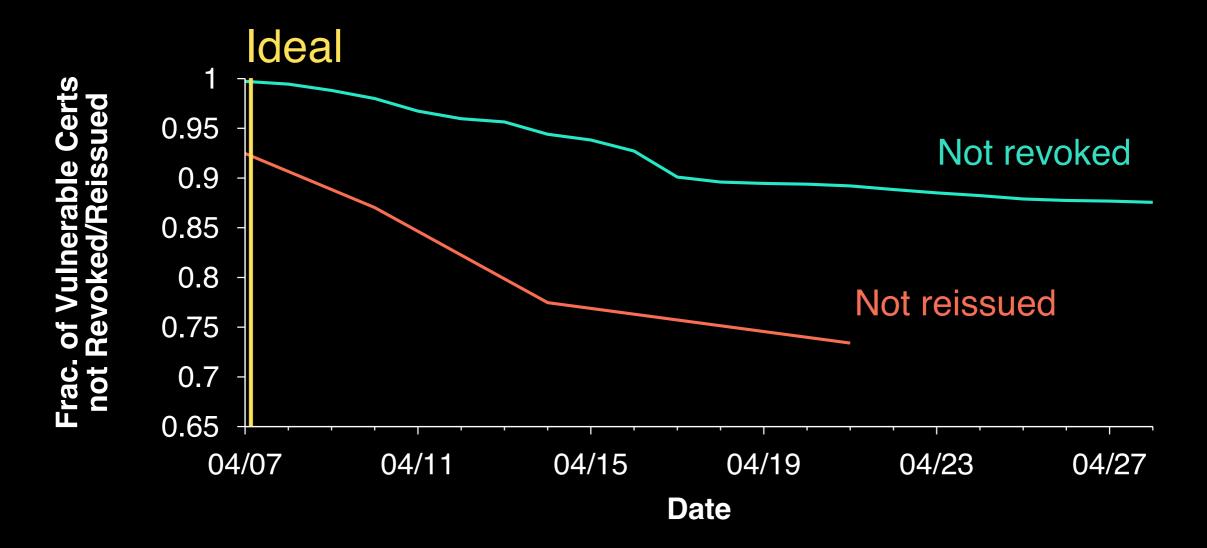


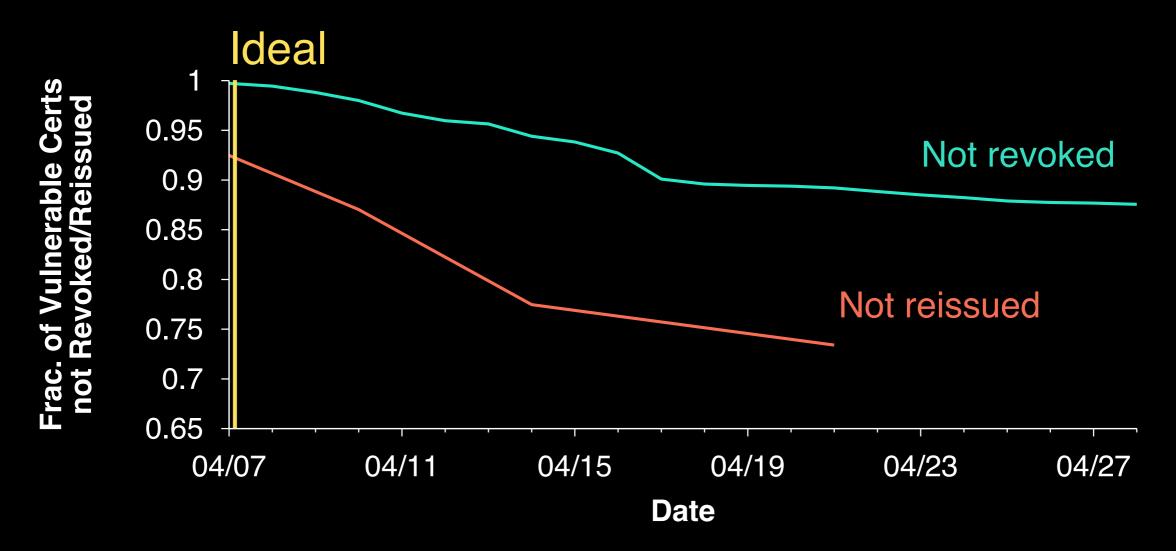
#### Similar pattern to patches: Exponential drop-off, then levels out



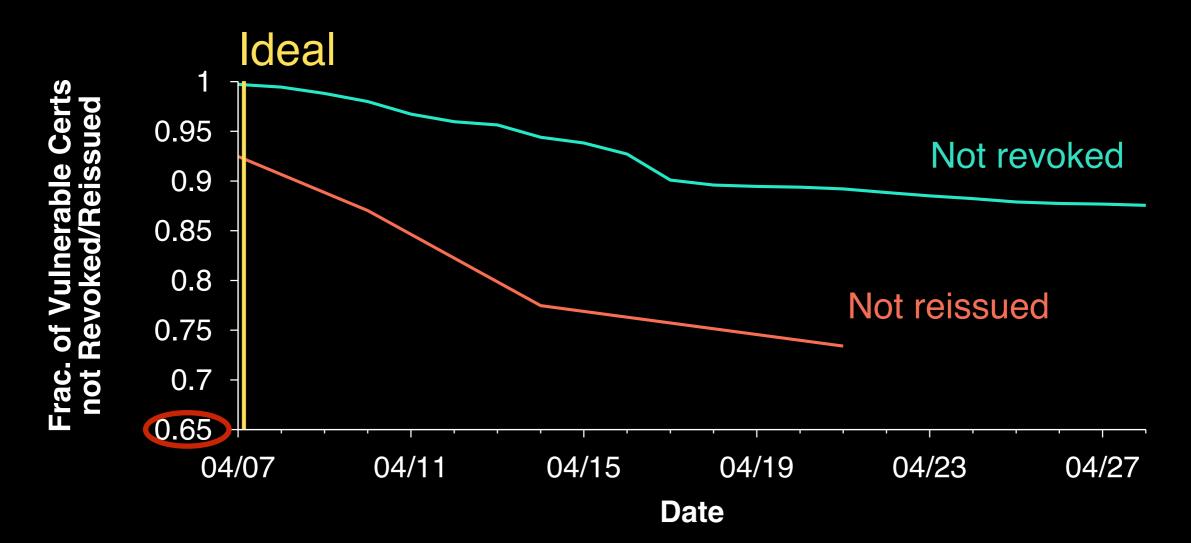








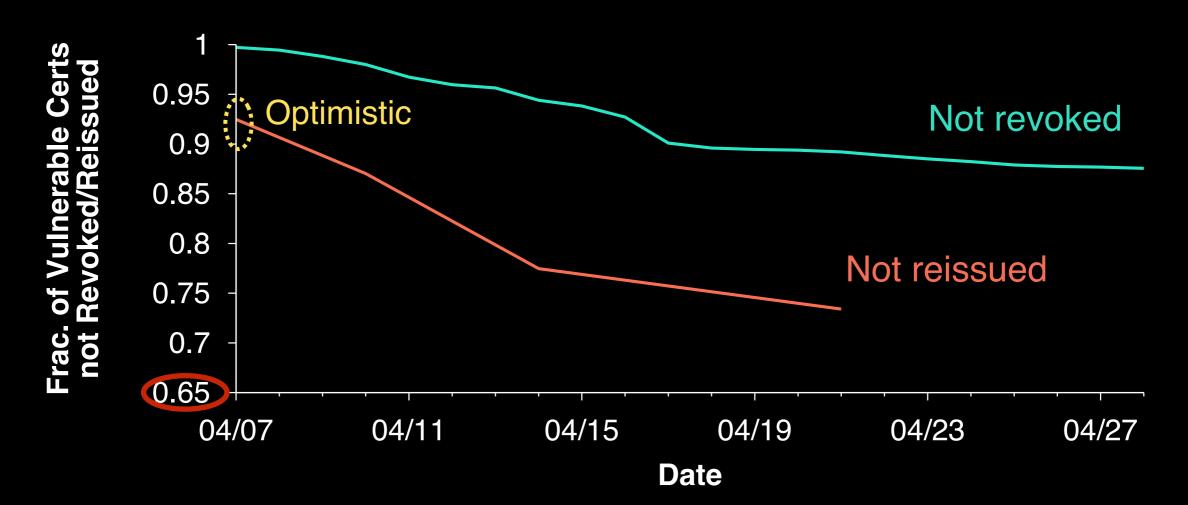
#### Similar pattern to patches: Exponential drop-off, then levels out



### Similar pattern to patches: Exponential drop-off, then levels out



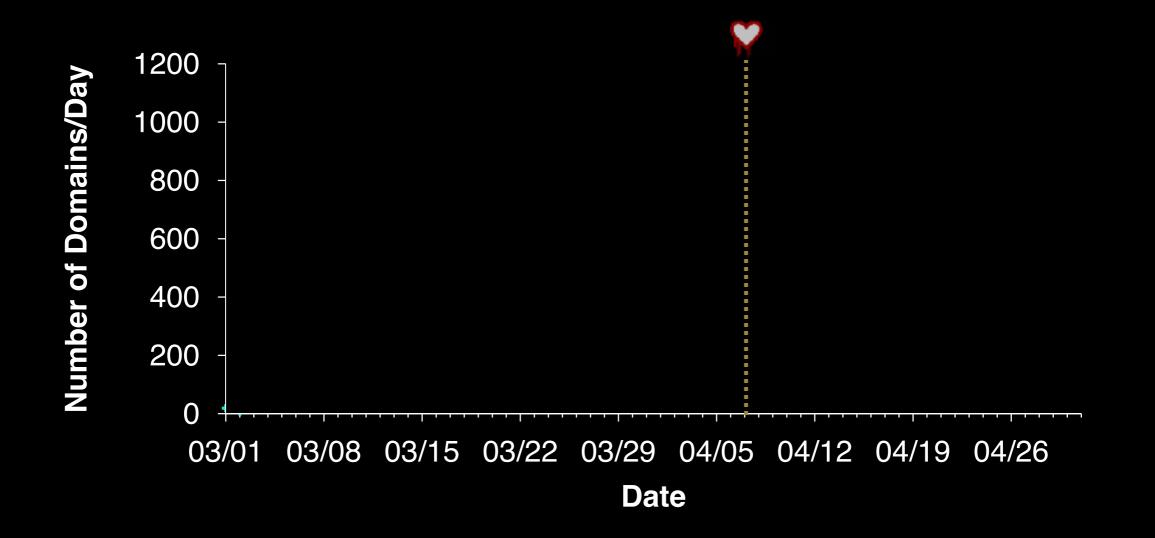


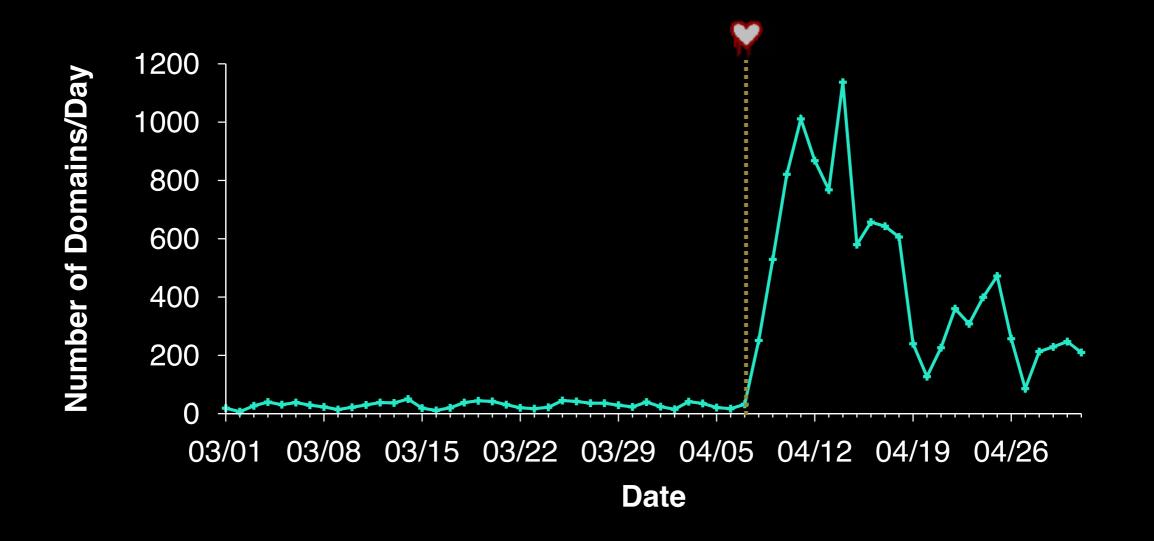


### Similar pattern to patches: Exponential drop-off, then levels out

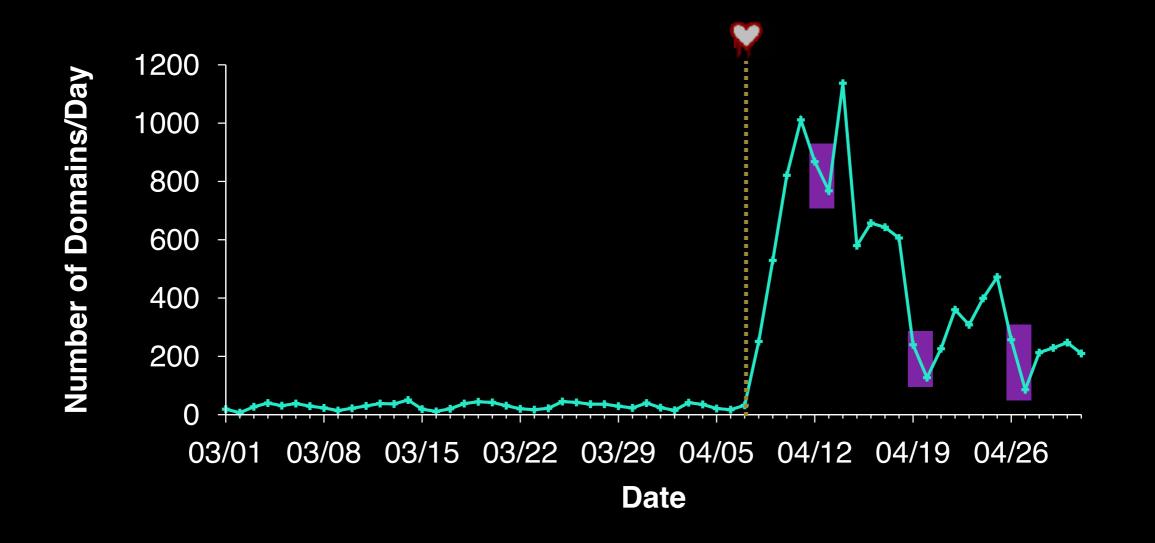




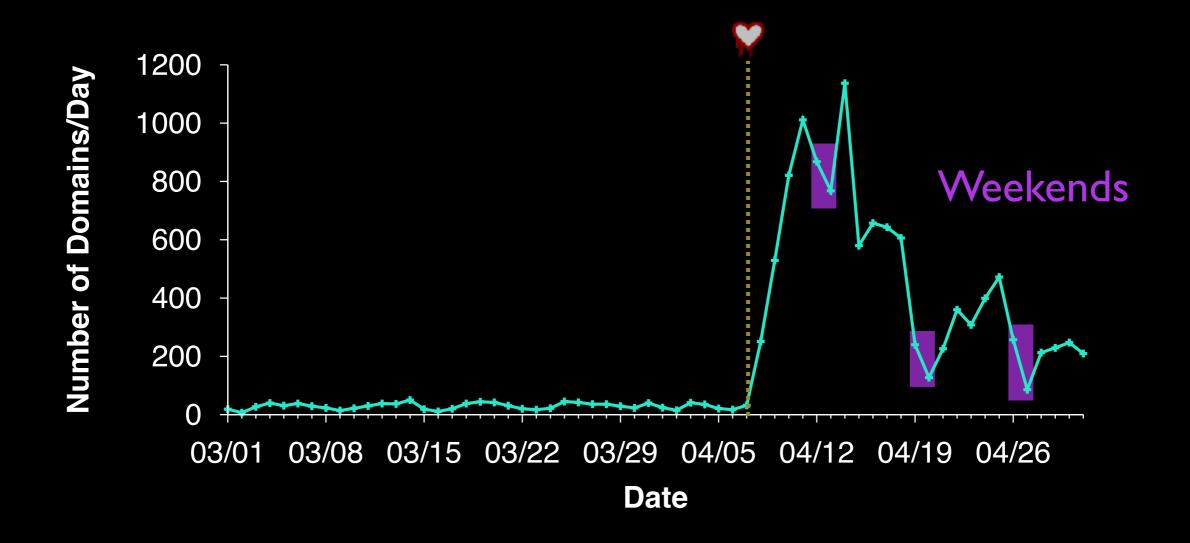




#### Reaction ramps up quickly

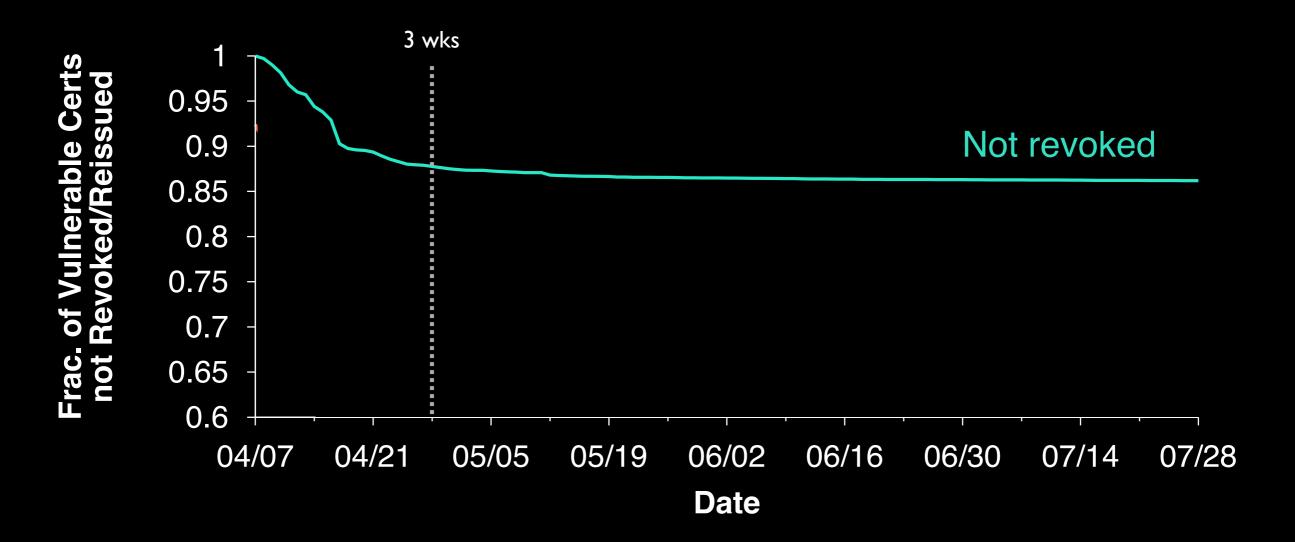


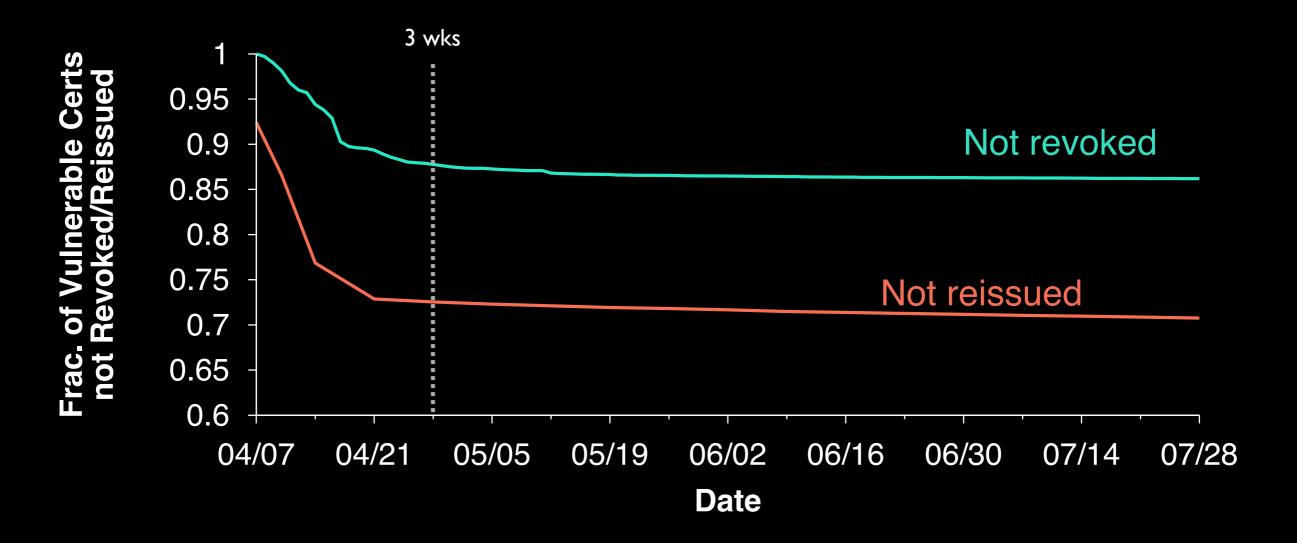
#### Reaction ramps up quickly

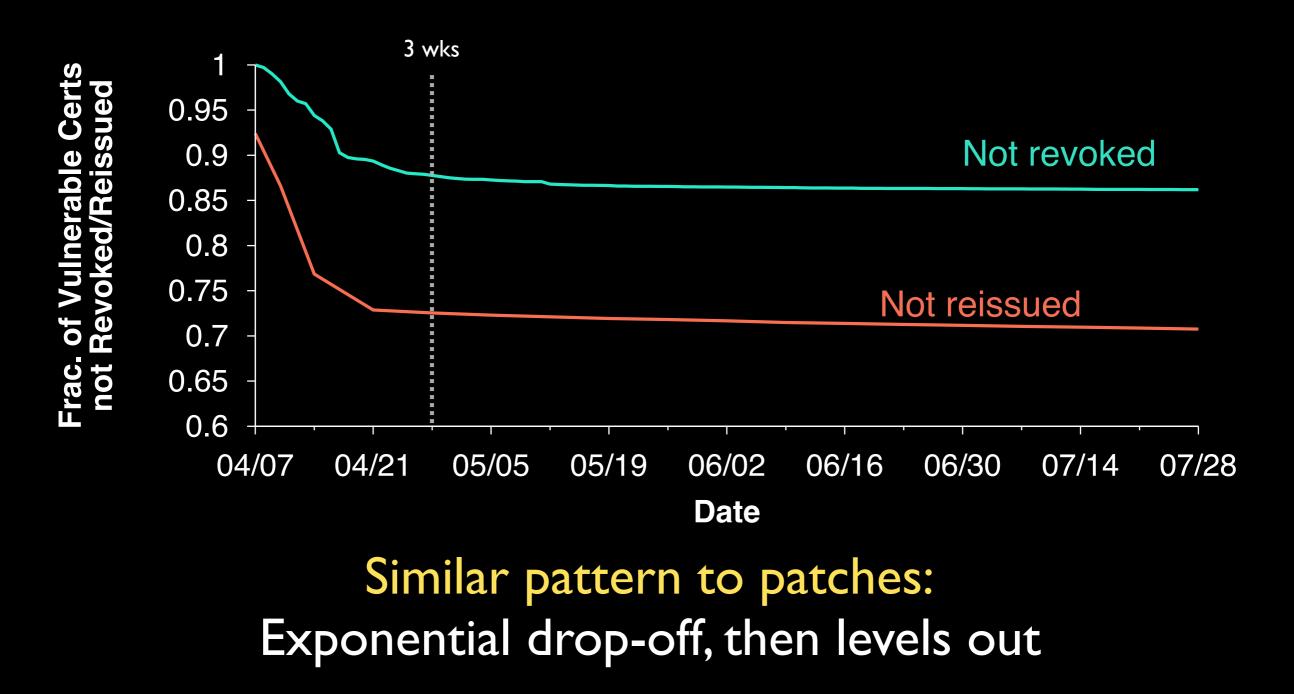


Reaction ramps up quickly

Security takes the weekends off



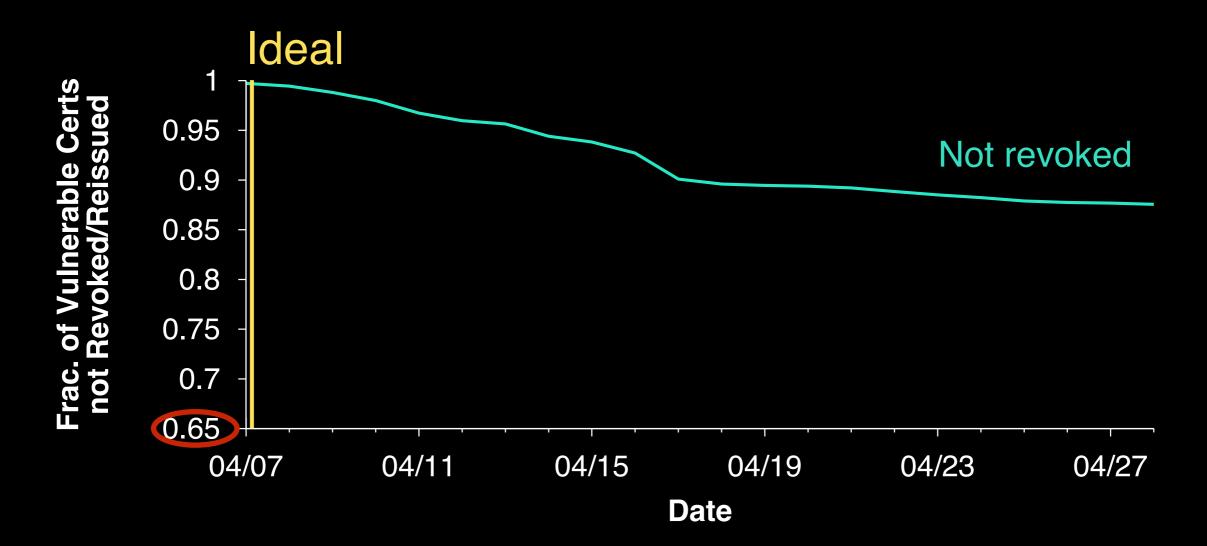


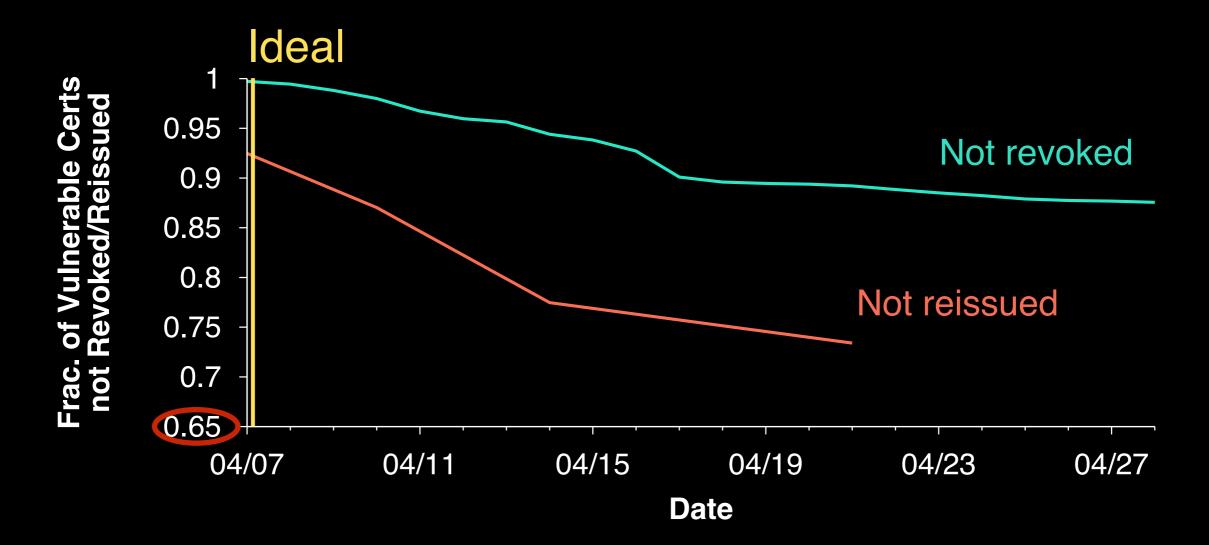


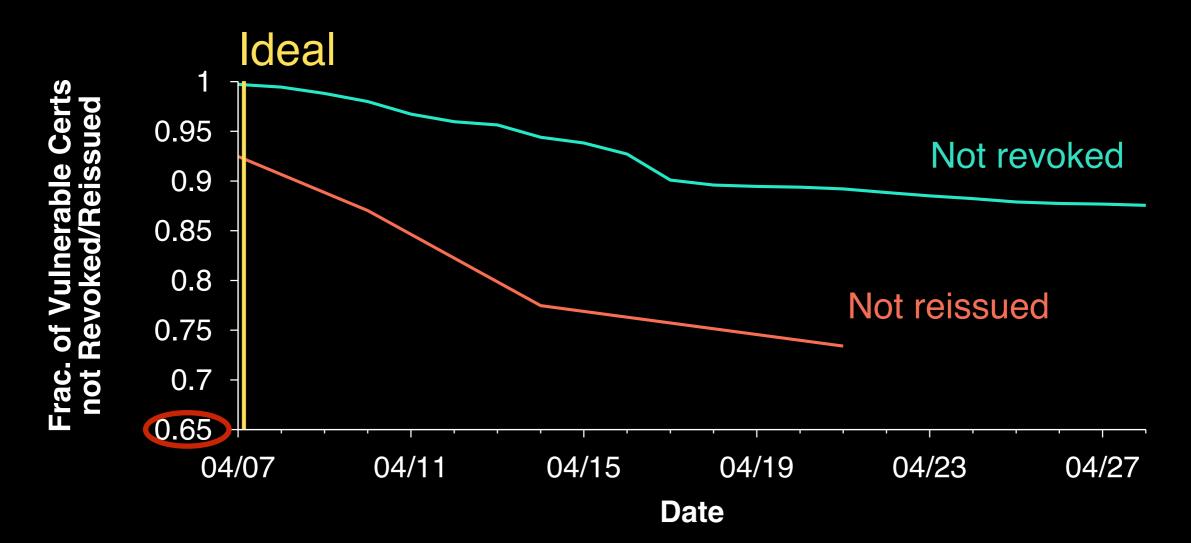
After 3 weeks:









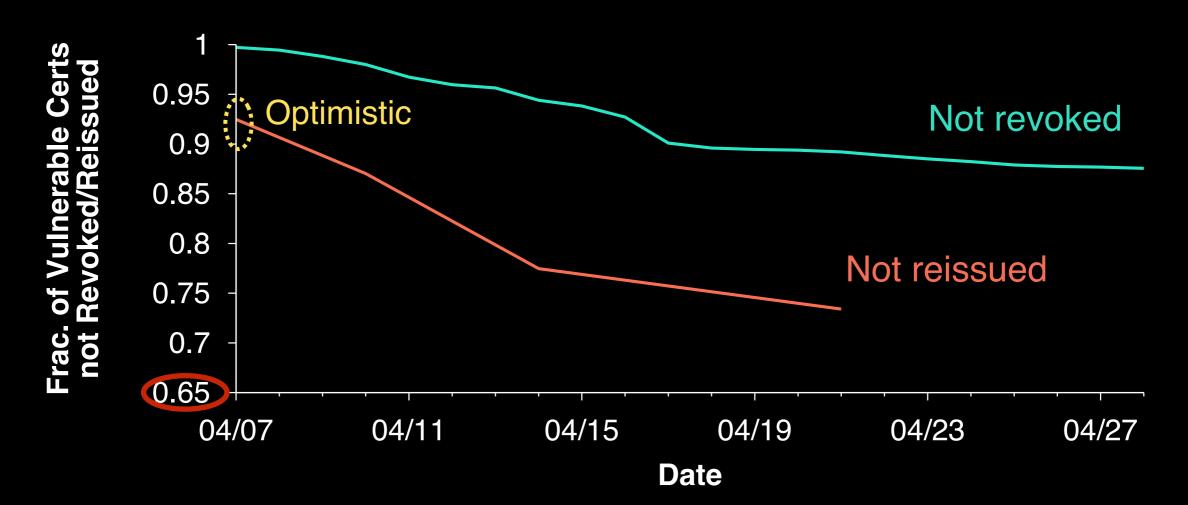


#### Similar pattern to patches: Exponential drop-off, then levels out

After 3 weeks:







#### Similar pattern to patches: Exponential drop-off, then levels out

After 3 weeks:

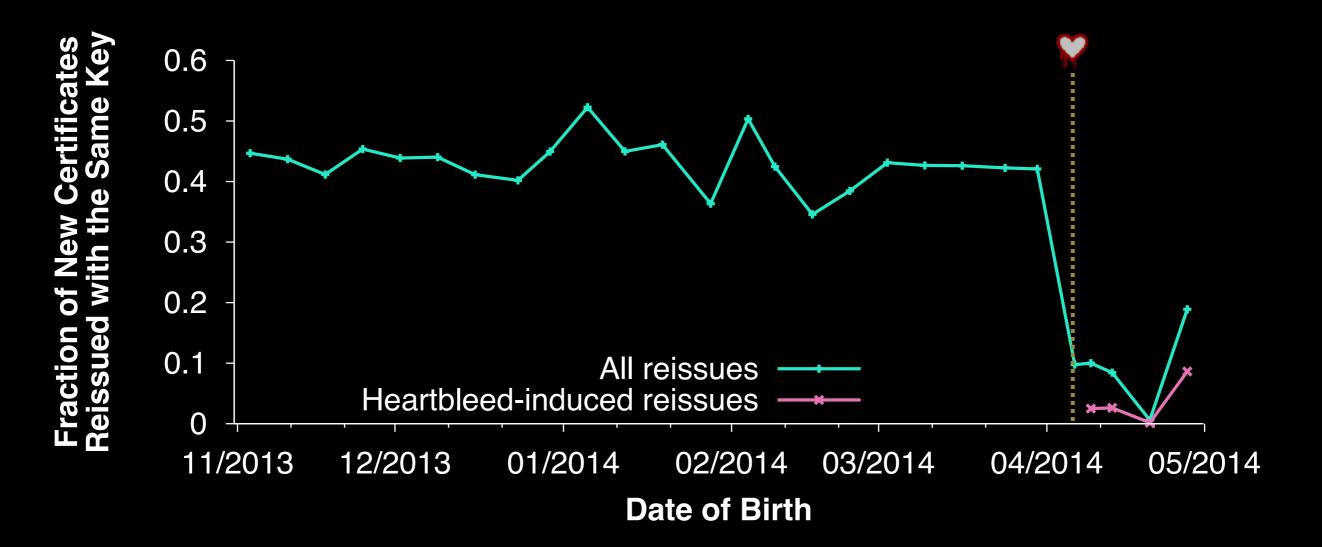




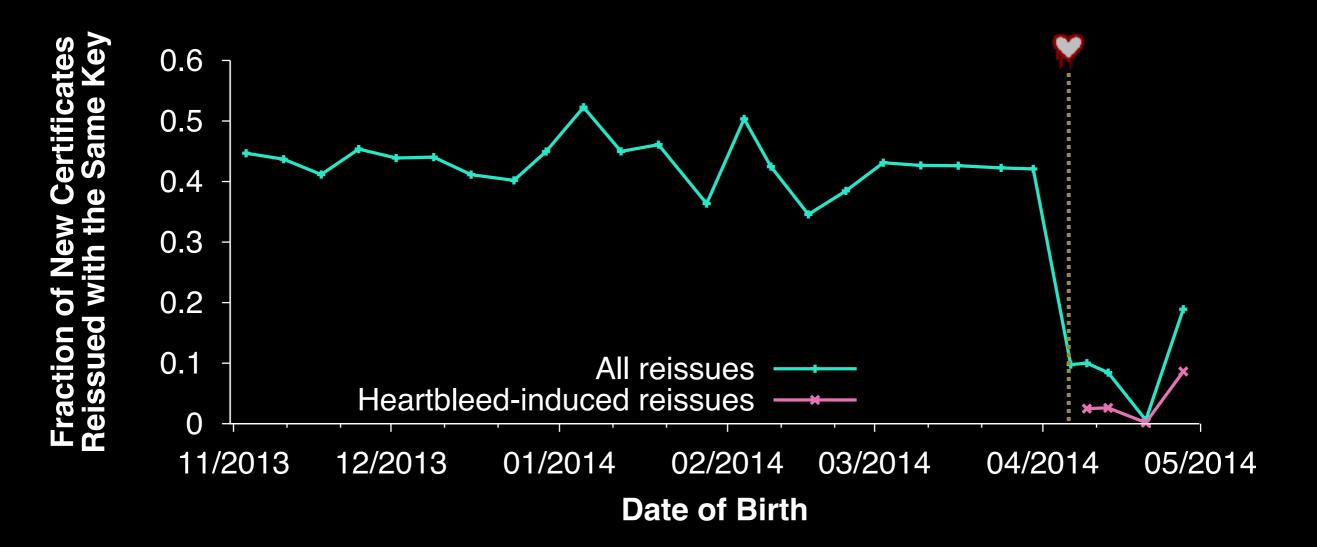
#### Reissue $\Rightarrow$ New key?



#### Reissue $\Rightarrow$ New key?



#### Reissue $\Rightarrow$ New key?



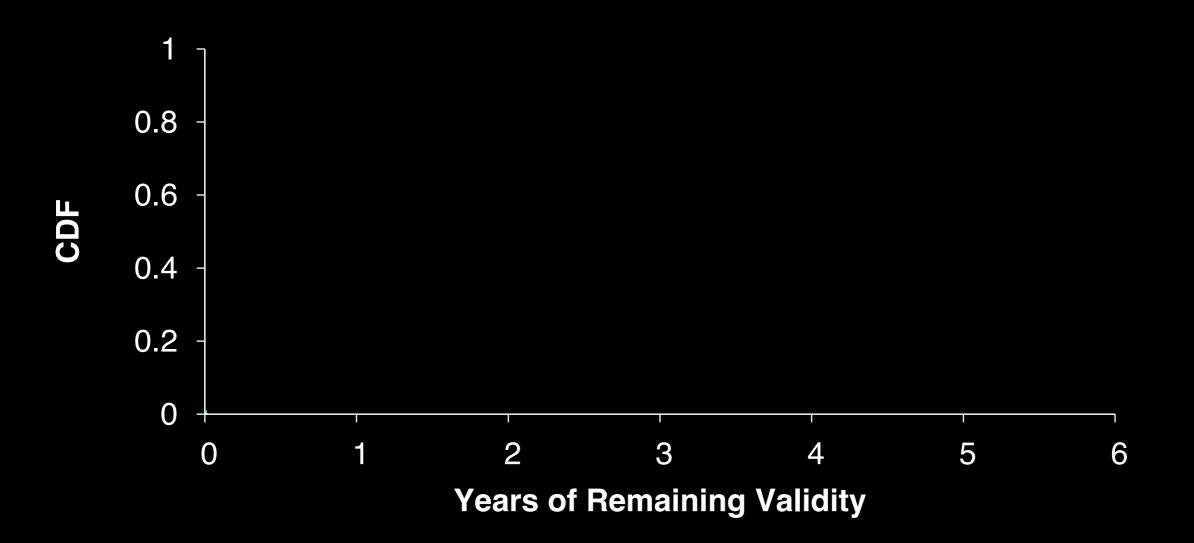
Reissuing the same key is common practice 4.1% Heartbleed-induced

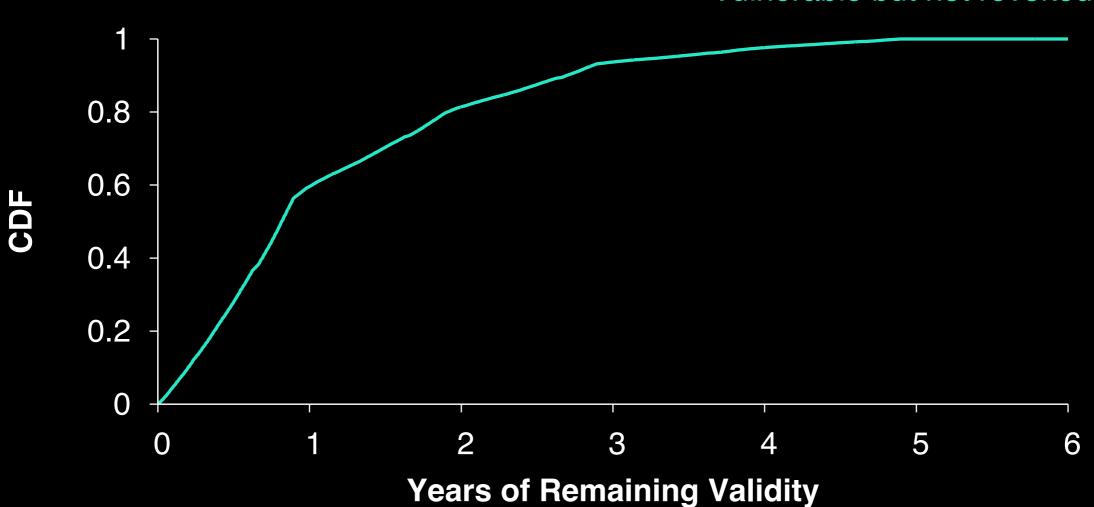
## The ugly truth of revocations



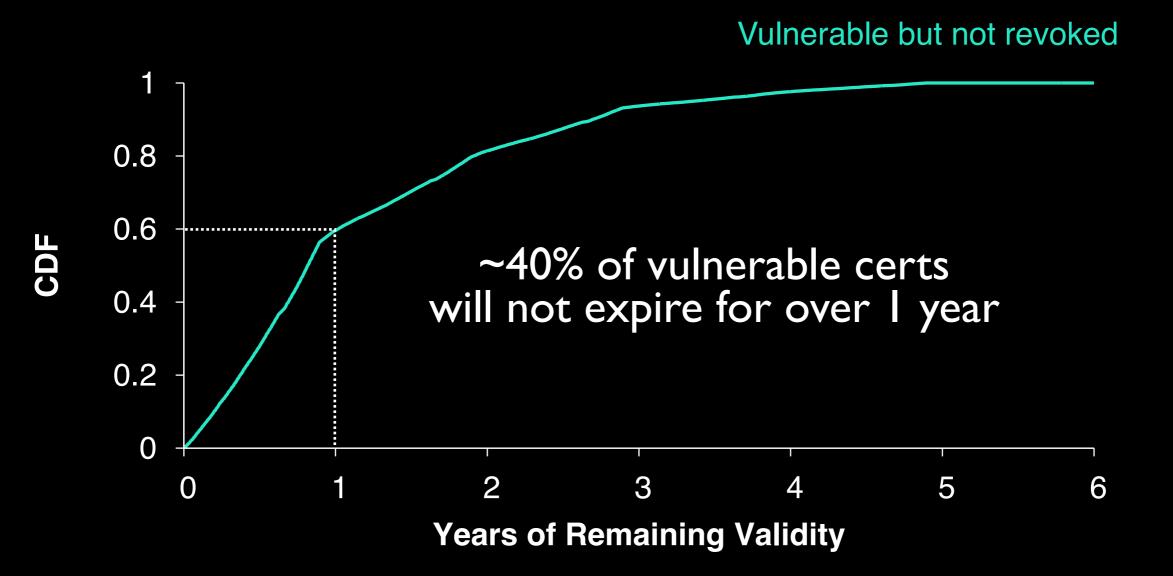
#### Security is supposed to be a fundamental design goal, but

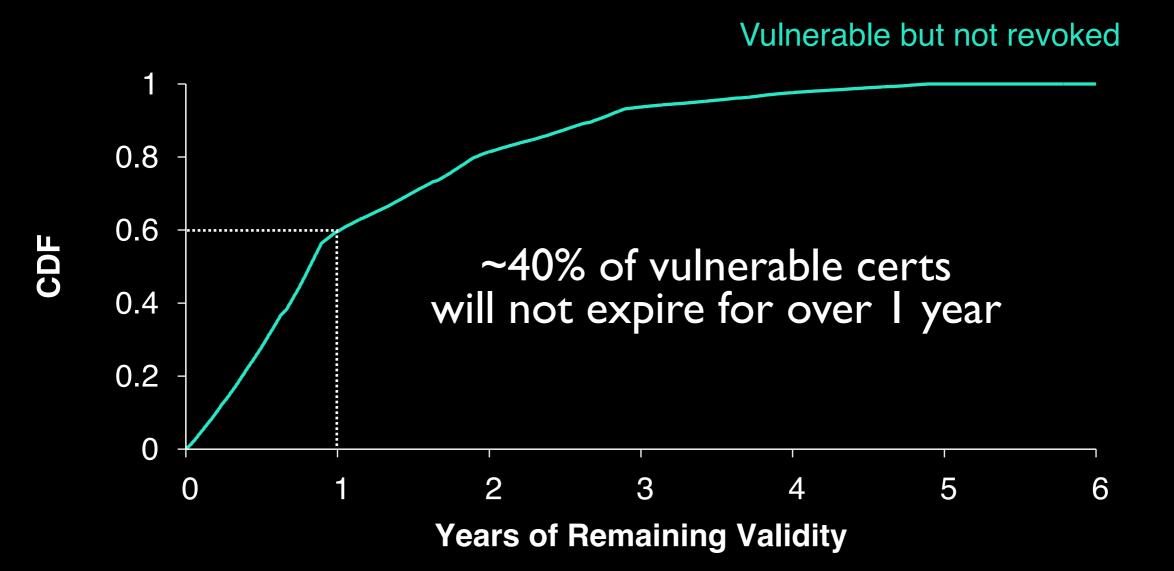
- Administrators trade off security for ease of maintenance/cost
- Certificate authorities trade off security for profit





Vulnerable but not revoked

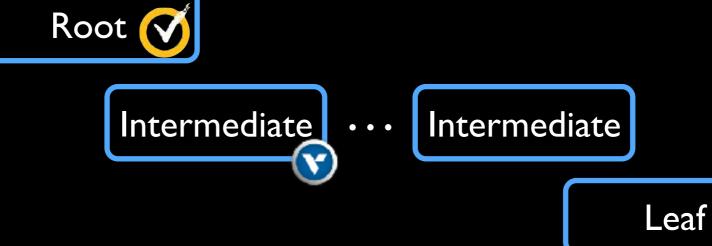




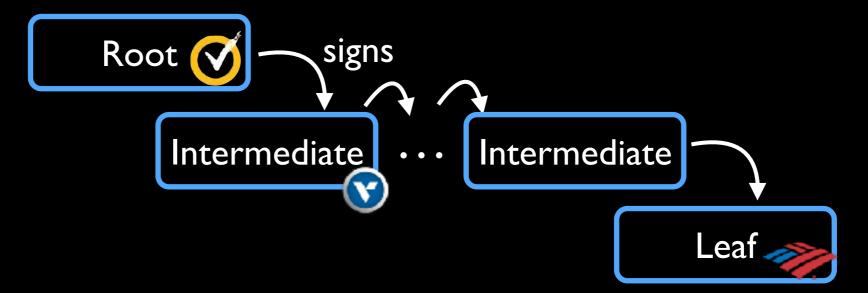
We may be dealing with Heartbleed for years

Browsers should support all major protocols Revocation protocols CRLs, OCSP, OCSP stapling lacksquare• Browsers should reject certs they cannot check Availability of revocation info E.g., because the OCSP server is down Browsers should reject a cert if any on the chain fail  $\bullet$ Chain lengths Leaf, intermediate(s), root ullet

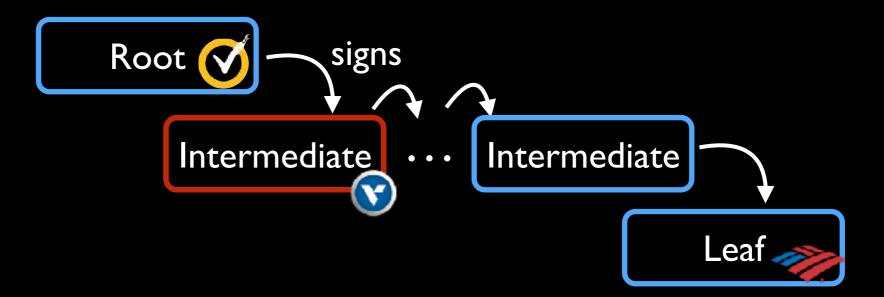
Browsers should support all major protocols Revocation protocols CRLs, OCSP, OCSP stapling  $\bullet$ Browsers should reject certs they cannot check Availability of revocation info E.g., because the OCSP server is down Browsers should reject a cert if any on the chain fail  $\bullet$ Chain lengths Leaf, intermediate(s), root ullet



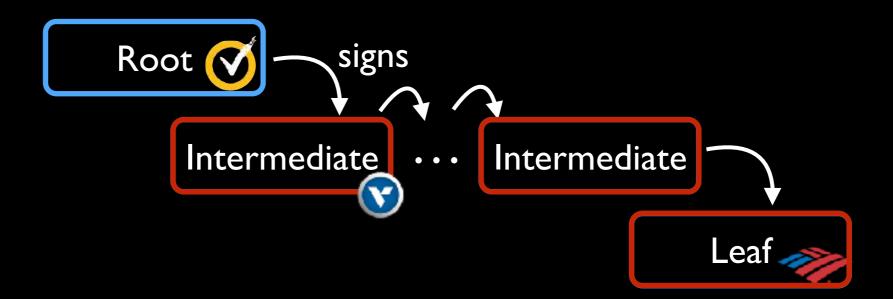
Browsers should support all major protocols Revocation protocols CRLs, OCSP, OCSP stapling  $\bullet$ Browsers should reject certs they cannot check Availability of revocation info E.g., because the OCSP server is down Browsers should reject a cert if any on the chain fail  $\bullet$ Chain lengths Leaf, intermediate(s), root ullet



Browsers should support all major protocols Revocation protocols CRLs, OCSP, OCSP stapling  $\bullet$ Browsers should reject certs they cannot check Availability of revocation info E.g., because the OCSP server is down Browsers should reject a cert if any on the chain fail  $\bullet$ Chain lengths Leaf, intermediate(s), root ullet



Browsers should support all major protocols Revocation protocols CRLs, OCSP, OCSP stapling  $\bullet$ Browsers should reject certs they cannot check Availability of revocation info E.g., because the OCSP server is down Browsers should reject a cert if any on the chain fail ulletChain lengths Leaf, intermediate(s), root ullet



		Desktop Browsers					Mobile Browsers							
		C	hrome	42	Firefox	Ope	era	Safari	] ]	12	iOS	Andr	4.1-5.1	IE
		OS X	Win.	Linux	35-37	12.17	28.0	6-8	7–9	10 - 11	6-8	Stock	Chrome	8.0
	CRL													
Int 1	Revoked	EV	1	EV	×	1	1	1	1	1	×	×	×	×
Int. 1	Unavailable	EV	1	-	×	×	1	1	1	1	×	×	×	X
Test Or	Revoked	EV	EV	EV	×	1	1	1	1	1	X	X	×	X
Int. 2+	Unavailable	×	×	_	×	×	×	×	×	×	×	×	×	×
Taaf	Revoked	EV	EV	EV	×	1	1	1	1	1	×	×	×	×
Leaf	Unavailable	×	×	_	×	×	x	×	×	А	x	×	×	×
	OCSP													
Int. 1	Revoked	EV	EV	EV	EV	×	1	1	1	1	X	X	×	X
1110. 1	Unavailable	×	×	_	×	×	L/W	×	1	1	×	×	×	×
Int. 2+	Revoked	EV	EV	EV	EV	×	1	1	1	1	×	×	×	×
III0. 2+	Unavailable	×	×	-	×	×	×	×	×	×	×	×	×	×
Last	Revoked	EV	EV	EV	1	1	1	1	1	1	×	X	×	X
Leaf	Unavailable	×	×	_	×	×	×	×	×	А	×	×	×	×
OC	SP Stapling													
Request OCSP Staple		1	<ul> <li>Image: A second s</li></ul>	1	1	1	1	×	1	1	×	I	I	×
Respect Revoked Staple		×	1	-	1	1	l/w	-	1	1	-	-	-	-

Passes testFails test

EV Passes for EV certsI Ignores OCSP Staple

		С	hrome	42
		OS X	Win.	Linux
Test 1	CRL Revoked	EV	1	EV
Int. 1	Unavailable	EV	1	-
Int. 2+	Revoked Unavailable	EV X	EV X	EV
Leaf	Revoked Unavailable	ev X	ev X	EV -
	OCSP			
Int. 1	Revoked Unavailable	EV X	EV X	EV
Int. 2+	Revoked Unavailable	EV X	EV	EV
Leaf	Revoked Unavailable	ev X	EV X	EV -
OCSP Stapling Request OCSP Staple Respect Revoked Staple		×	1	× -



Generally, only checks for EV certs ~3% of all certs

Allows if revocation info unavailable

Supports OCSP stapling

Passes testFails test

EV Passes for EV certsI Ignores OCSP Staple

		Deskte Firefox
		35-37
	CRL	
Int. 1	Revoked	×
1116. 1	Unavailable	×
Int. 2+	Revoked	×
Int. 2+	Unavailable	×
Leaf	Revoked	×
Lear	Unavailable	×
	OCSP	
Int 1	Revoked	EV
Int. 1	Unavailable	×
Int. 2+	Revoked	EV
Int. 27	Unavailable	×
1 6	Revoked	1
Leaf	Unavailable	×
ocs	SP Stapling	
Reques	1	
Respect 1	Revoked Staple	1



Never checks CRLs Only checks intermediates for EV certs

Allows if revocation info unavailable

Supports OCSP stapling

Passes testFails test

EV Passes for EV certsI Ignores OCSP Staple

		Safari
		6-8
	CRL	
T	Revoked	1
Int. 1	Unavailable	1
T . O.	Revoked	1
Int. 2+	Unavailable	×
Leaf	Revoked	1
Lear	Unavailable	×
	OCSP	
	Revoked	1
Int. 1	Unavailable	×
		<u>^</u>
Int. 2+	Revoked	<ul> <li>Image: A set of the set of the</li></ul>
	Unavailable	×
1	Revoked	1
Leaf	Unavailable	×
OCS		
Reques Bornort I	×	
Respect I	_	



Checks CRLs and OCSP

Allows if revocation info unavailable Except for first intermediate, for CRLs

Does not support OCSP stapling

Passes testFails test

EV Passes for EV certsI Ignores OCSP Staple

		IE	
		7–9	10-11
	CRL		
Int. 1	Revoked	1	1
1116. 1	Unavailable	1	1
Test Or	Revoked	1	1
Int. 2+	Unavailable	×	×
T 6	Revoked	1	1
Leaf	Unavailable	×	Α
	OCSP		
Int. 1	Revoked	1	1
1116. 1	Unavailable	1	1
Test Or	Revoked	1	1
Int. 2+	Unavailable	×	×
	Revoked	1	1
Leaf	Unavailable	×	Α
OCS	SP Stapling		
	t OCSP Staple	1	1
-	Revoked Staple	1	1



Checks CRLs and OCSP

Often rejects if revocation info unavailable Pops up alert for leaf in IE 10+

Supports OCSP stapling

Passes testFails test

EV Passes for EV certsI Ignores OCSP Staple

		Mobile Browsers				
		iOS	IE			
		6-8	Stock	Chrome	8.0	
	CRL					
Int. 1	Revoked	×	×	×	×	
Inc. 1	Unavailable	×	×	×	×	
Int. 2+	Revoked	×	X	×	X	
Int. 2+	Unavailable	×	×	×	×	
Leaf	Revoked	×	×	×	×	
Lear	Unavailable	×	×	×	×	
	OCSP					
	Revoked				~	
Int. 1		0	×	×.	×	
	Unavailable	<u>^</u>	×	×	×	
Int. 2+	Revoked	×	×	×	×	
IIIC. 2+	Unavailable	×	×	×	×	
Lcaf	Revoked	×	×	×	X	
Lear	Unavailable	×	×	×	×	
OCSP Stapling Request OCSP Staple Respect Revoked Staple						
		5				
		×	I	I	×	
		-	-	-	-	



Uniformly never check

Android browsers request Staple ....and promptly ignore it

Passes testFails test

EV Passes for EV certsI Ignores OCSP Staple

The PKI is how we know with whom we are communicating online

The PKI's job is to bind human-understandable identities (domain names) to cryptographic keys (public keys)

The central mechanism for this is **certificates**: digital signatures from trusted entities that tie domain names and public keys together

TLS along with Diffie-Hellman leverages public key crypto to arrive at ephemeral *session keys* (symmetric keys)

There is significant *mismanagement* in today's PKI:

- Websites don't revoke or get new certs ("reissue") when they should
- Browsers don't check for revocations when they should
- Websites share their private keys with their hosting providers

Improving the web's PKI is an active area of research (securepki.org)