CS 350S: Privacy-Preserving Systems

Certificate Transparency

Outline

1. Overview of certificate transparency
2. Client auditing
3. Log monitoring
4. Rollout and remaining challenges

Motivation

- Hacker compromised DigiNotar and issued at least 531 bad certificates
  - Included: Google, Mozilla, Yahoo, Skype, Facebook, Twitter, Tor, CIA, Israel’s Mossad, UK’s MI6, …
- Detected via Chrome certificate pinning
- Google, Mozilla, and Microsoft removed DigiNotar CA from list of trusted CAs
- Enabled man-in-the-middle attacks against >300,000 unique IP addresses in Iran over a period of potentially over a month
- Attacker released statement claiming that he was an Iranian helping the government to monitor communications

https://www.eff.org/deeplinks/2011/09/post-mortem-iranian-diginotar-attack

Motivation

- Many hundreds of CAs
- Attacker only needs to compromise ONE to start issuing fraudulent certificates
- DigiNotar had maintained all of servers for certificate issuance on one Windows domain with a weak password

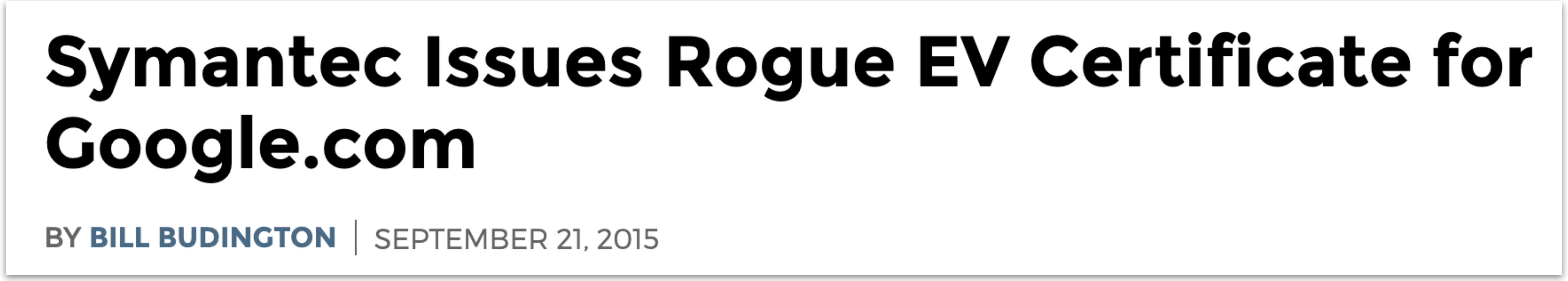
https://www.eff.org/deeplinks/2011/09/post-mortem-iranian-diginotar-attack

High-level goal

- Goal: Make it possible to quickly *detect* misissued certificates
- Non-goal: *Prevent* misissued certificates

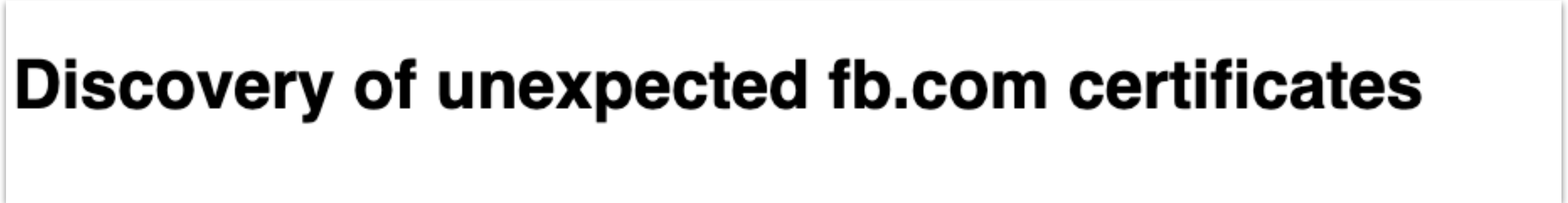
Challenge: accommodating many requirements from web infrastructure

CT has helped catch misissued certificates



Symantec Issues Rogue EV Certificate for Google.com

BY BILL BUDINGTON | SEPTEMBER 21, 2015



Discovery of unexpected fb.com certificates

System goals

- Migration path
- Scales to many parties and certificates
- Avoid placing trust in a single entity in the system
- No TLS handshake external dependencies (no page-load latency increase)
- Should not require user decisions (certificate warnings are confusing)

How does CT meet these goals?

- Migration path
- Scales to many parties and certificates
- Avoid placing trust in a single entity in the system
- No TLS handshake external dependencies (no page-load latency increase)
- Should not require user decisions (certificate warnings are confusing)

How does CT meet these goals?

- **Migration path**
  - Certificates issued and revoked similarly to before
  - Security benefits even if only some clients perform auditing
- Scales to many parties and certificates
- Avoid placing trust in a single entity in the system
- No TLS handshake external dependencies (no page-load latency increase)
- Should not require user decisions (certificate warnings are confusing)

How does CT meet these goals?

- Migration path
- **Scales to many parties and certificates**
  - Every certificate should be added to log
  - Log is not making any “judgment” about the certificate
- Avoid placing trust in a single entity in the system
- No TLS handshake external dependencies (no page-load latency increase)
- Should not require user decisions (certificate warnings are confusing)

How does CT meet these goals?

- Migration path
- Scales to many parties and certificates
- **Avoid placing trust in a single entity in the system**
    - Anyone can check if the log is behaving correctly
- No TLS handshake external dependencies (no page-load latency increase)
- Should not require user decisions (certificate warnings are confusing)

How does CT meet these goals?

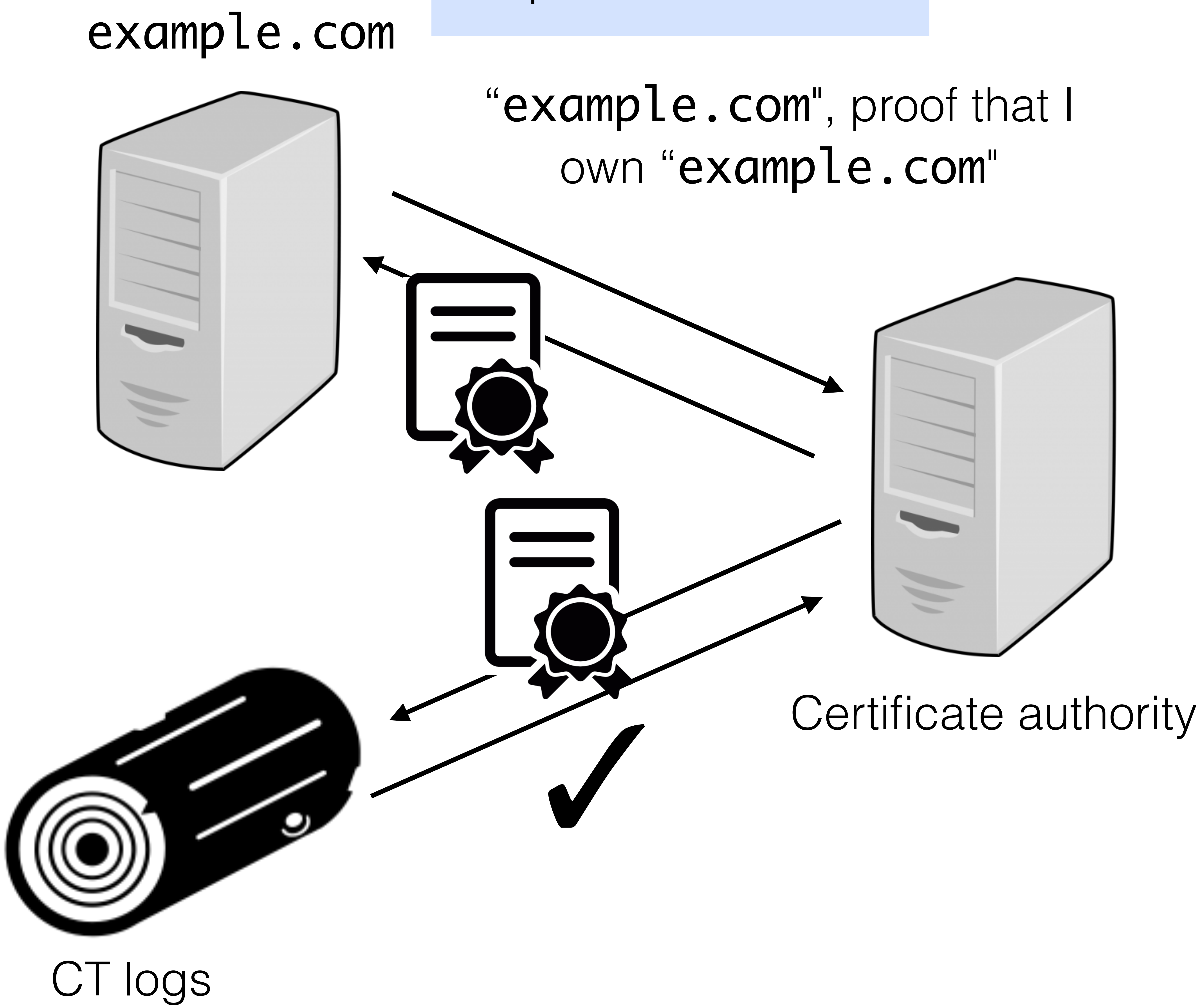
- Migration path
- Scales to many parties and certificates
- Avoid placing trust in a single entity in the system
- **No TLS handshake external dependencies (no page-load latency increase)**
  - Client only needs to fetch and verify SCT from the server
  - No communication with log or CA at page-load time
- Should not require user decisions (certificate warnings are confusing)

How does CT meet these goals?

- Migration path
- Scales to many parties and certificates
- Avoid placing trust in a single entity in the system
- No TLS handshake external dependencies (no page-load latency increase)
- **Should not require user decisions (certificate warnings are confusing)**
  - Domain owners monitor CT logs for misissued certificates
  - SCT auditing error reports can be sent automatically, without requiring user decisions

High-level overview of CT

Step 1: Get certificate



`example.com`

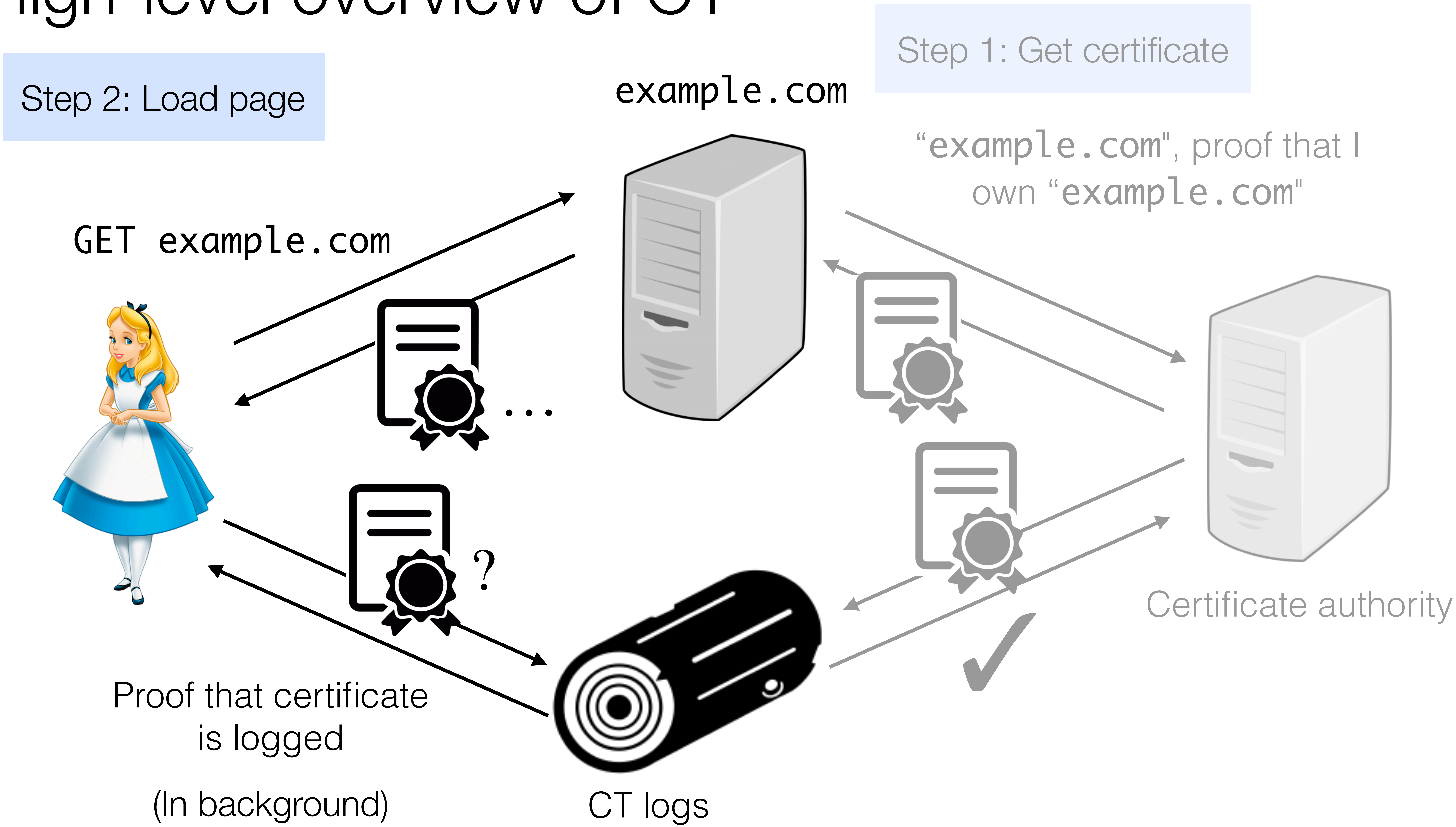
"`example.com`", proof that I own "`example.com`"

✓

Certificate authority

CT logs

High-level overview of CT



Step 2: Load page

example.com

Step 1: Get certificate

“example.com", proof that I own “example.com"

GET example.com

…

?

✓

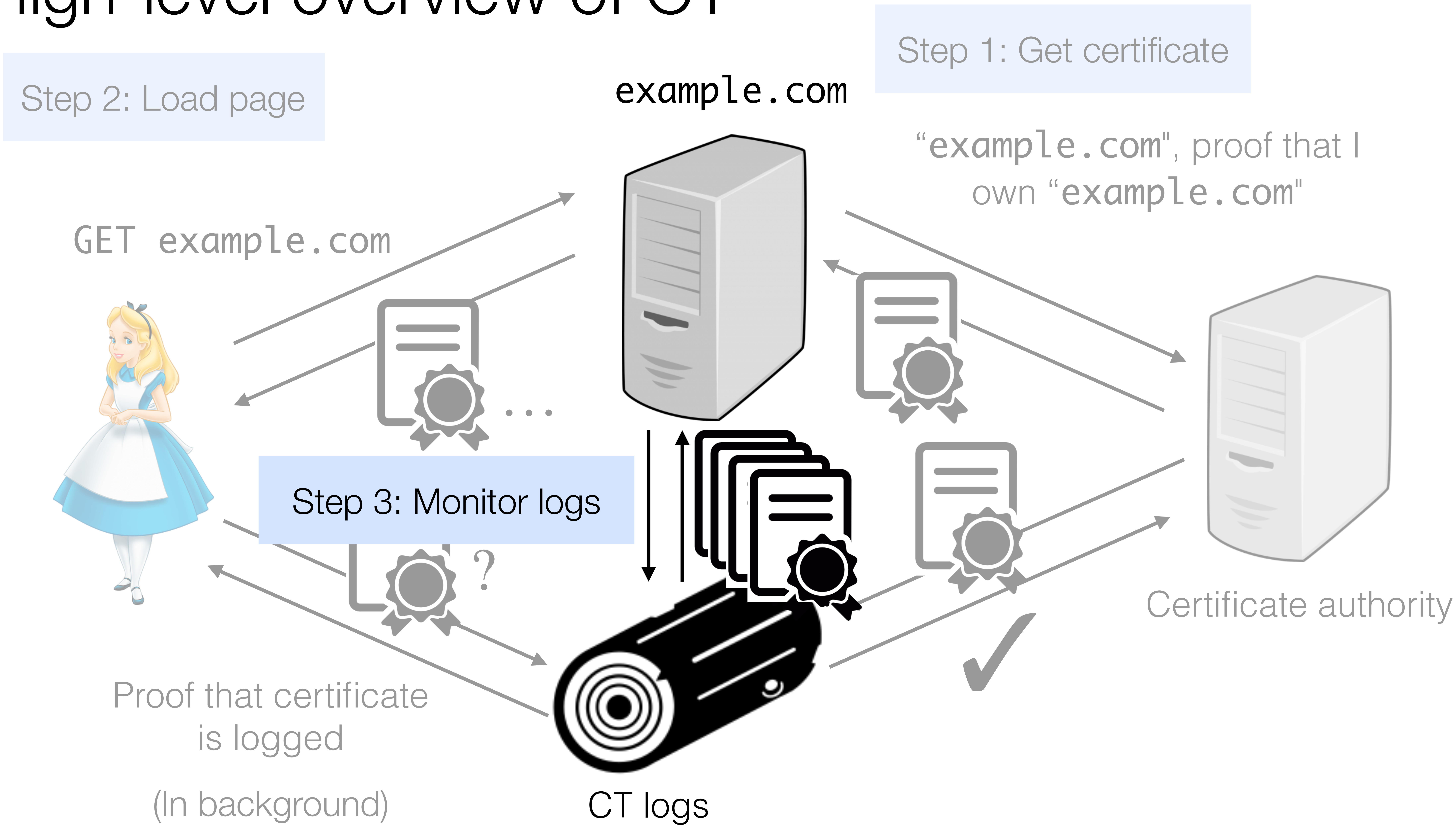
Certificate authority

Proof that certificate is logged

(In background)

CT logs

High-level overview of CT



Step 1: Get certificate

Step 2: Load page

Step 3: Monitor logs

example.com
“example.com”, proof that I own “example.com”
GET example.com
…
?
✓
Certificate authority
Proof that certificate is logged
(In background)
CT logs

Outline

1. Overview of certificate transparency
2. **Client auditing**
3. Log monitoring
4. Rollout and remaining challenges

Client auditing properties

- Client needs to check that certificate has been included in log
- Client cannot download entirety of CT log

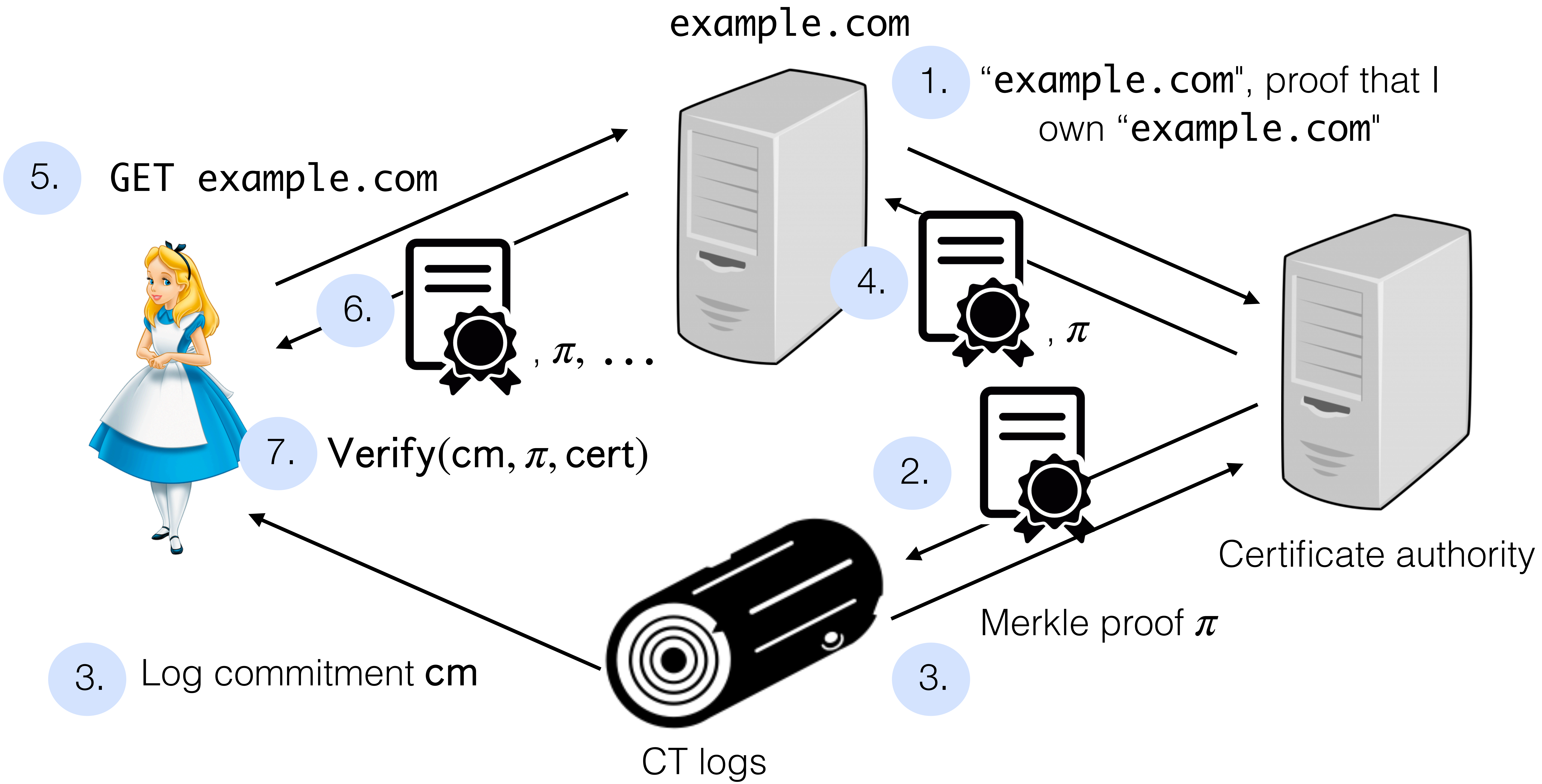
Tool: Merkle proof

- Compact proof of inclusion
- Client only needs a short (32B) commitment to the state

What goes wrong without client auditing?

- No guarantee that the certificates in the CT logs correspond to the ones that clients see on the web
- A misbehaving CT log may choose to not append a certificate to the log
- Client auditing makes it possible to detect CT log misbehavior

A simple client auditing solution (not deployed)



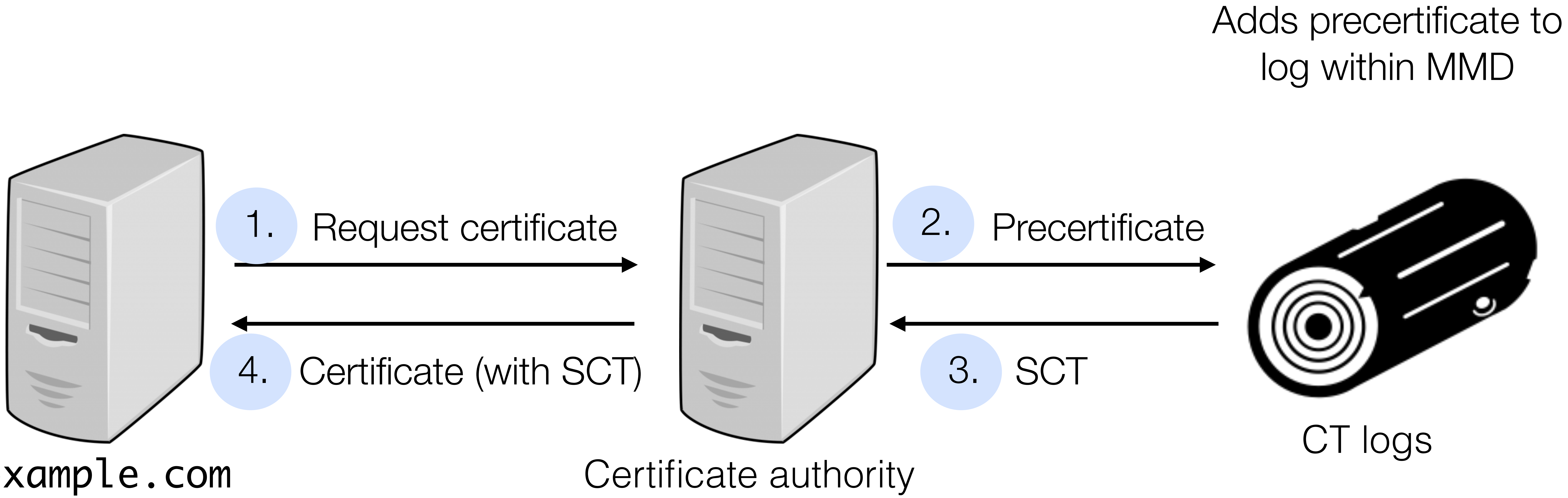
System requirement challenges

- Requirement: low latency for certificate issuance
- Delay for ingesting data in log can be hours
  - Cannot wait for data to be ingested and create Merkle proof to issue a certificate!
  - Problem both for new sites, but also existing sites with expired certificates

Solution: Signed Certificate Timestamps (SCTs)

- SCT: signed promise from a CT log that it will include a certificate within some time period (maximum merge delay, or MMD, e.g., 24 hours)

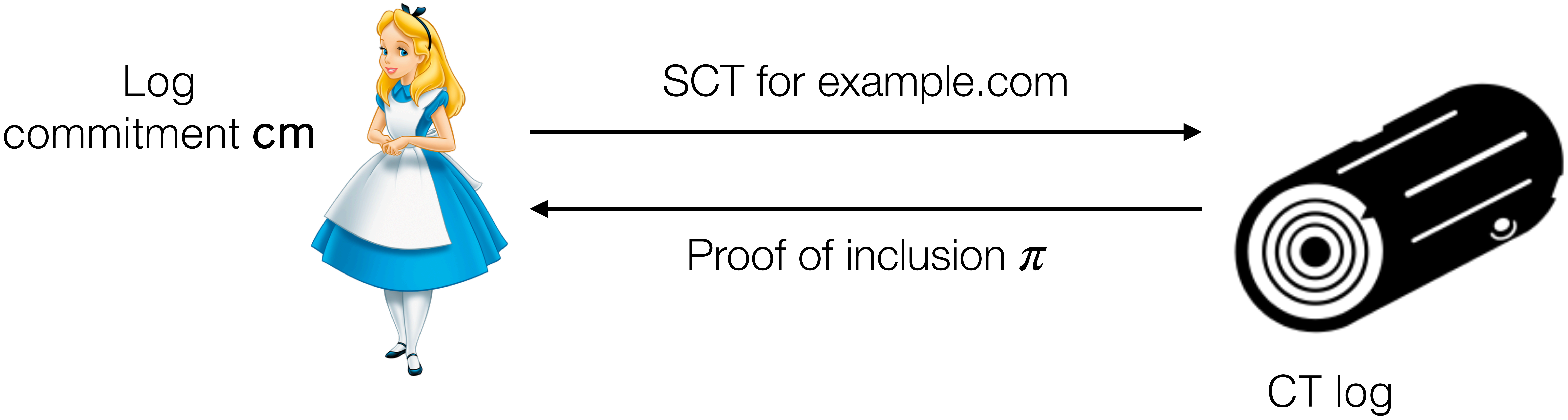
Certificate issuance with SCTs



Advantages?

- No delay issuing certificates
- SCT is part of certificate (helps with incremental deployment)
- Domain owner doesn't need to think about CT

Starting point for SCT auditing



Log commitment **cm**

SCT for example.com

Proof of inclusion π

CT log

Drawback?

Privacy: CT log learns which websites Alice visited

Background: K-anonymity

K-anonymity: For each released record, there are at least (k-1) other records with the same identifiable (i.e., externally linkable) fields

Idea: minimize damage of identifying data being associated with sensitive data

Non-sensitive			Sensitive
Gender	Age	Zipcode	Medical condition
M	40	94305	Heart disease
M	40	94305	Diabetes
F	40	94305	Cancer
F	40	94305	High blood pressure

K-anonymity: Homogeneity attack

Alice’s friend is a 40-year old, female in ZIP code 94305.

Non-sensitive			Sensitive
Gender	Age	Zipcode	Medical condition
M	40	94305	Heart disease
M	40	94305	Diabetes
F	40	94305	Cancer
F	40	94305	Cancer

Any of the 40-year old females in 94305 in the dataset have cancer: Alice’s friend has cancer

Machanavajjhala, Ashwin, et al. "l-diversity: Privacy beyond k-anonymity."

K-anonymity: Background knowledge attack

Bob's friend is a 40-year old, Japanese male in ZIP code 94305.

Non-sensitive			Sensitive
Gender	Age	Zipcode	Medical condition
M	40	94305	Heart disease
M	40	94305	Diabetes
F	40	94305	Cancer
F	40	94305	Cancer

Bob's friend has diabetes with high probability

Background knowledge: Extremely low incidence of heart disease among Japanese

Machanavajjhala, Ashwin, et al. "l-diversity: Privacy beyond k-anonymity."

K-anonymity

Limited privacy properties for releasing data

Later in this class: differential privacy for publishing aggregate statistics

SCT auditing in Chrome

Every 1 in 1,000 connections, the client:

- Hashes the SCT
- Computes the 20-bit prefix of the hash
- Waits the maximum merge delay (MMD) + ingestion time
- Fetches all of the certificates with the same 20-bit hash prefix

SCT auditing in Chrome

How does requesting a 20-bit hash provide k-anonymity-style privacy?

- Assume a minimum of 2.8B non-expired SCTs
- Hashing uniformly distributes SCTs across 256-bit space
- 32-bit prefix is enough to uniquely identify a certificate with good probability
- Use a 20-bit prefix to, with good probability, sample a set of size >= 1,000
- Server sees the 20-bit prefix, but the request could be for any of the >=1,000 SCTs

SCT auditing in Chrome

How is SCT auditing affected by the fact that some websites are visited much more frequently than others?

- If every 1 in 1000 connections are sampled, then popular websites will be audited frequently, while less popular ones will not
- Chrome’s solution: preload popular SCTs and don’t count them towards “budget” of 1,000 connections
- Privacy challenge: distribution of website popularity is essentially “background knowledge” — can use to narrow down the anonymity set
  [Lehmkuhl, Henzinger, Corrigan-Gibbs]

Cryptographically private SCT auditing (not deployed)

Keyword private information retrieval (PIR): Look up a key from a key-value store without revealing the key to the server

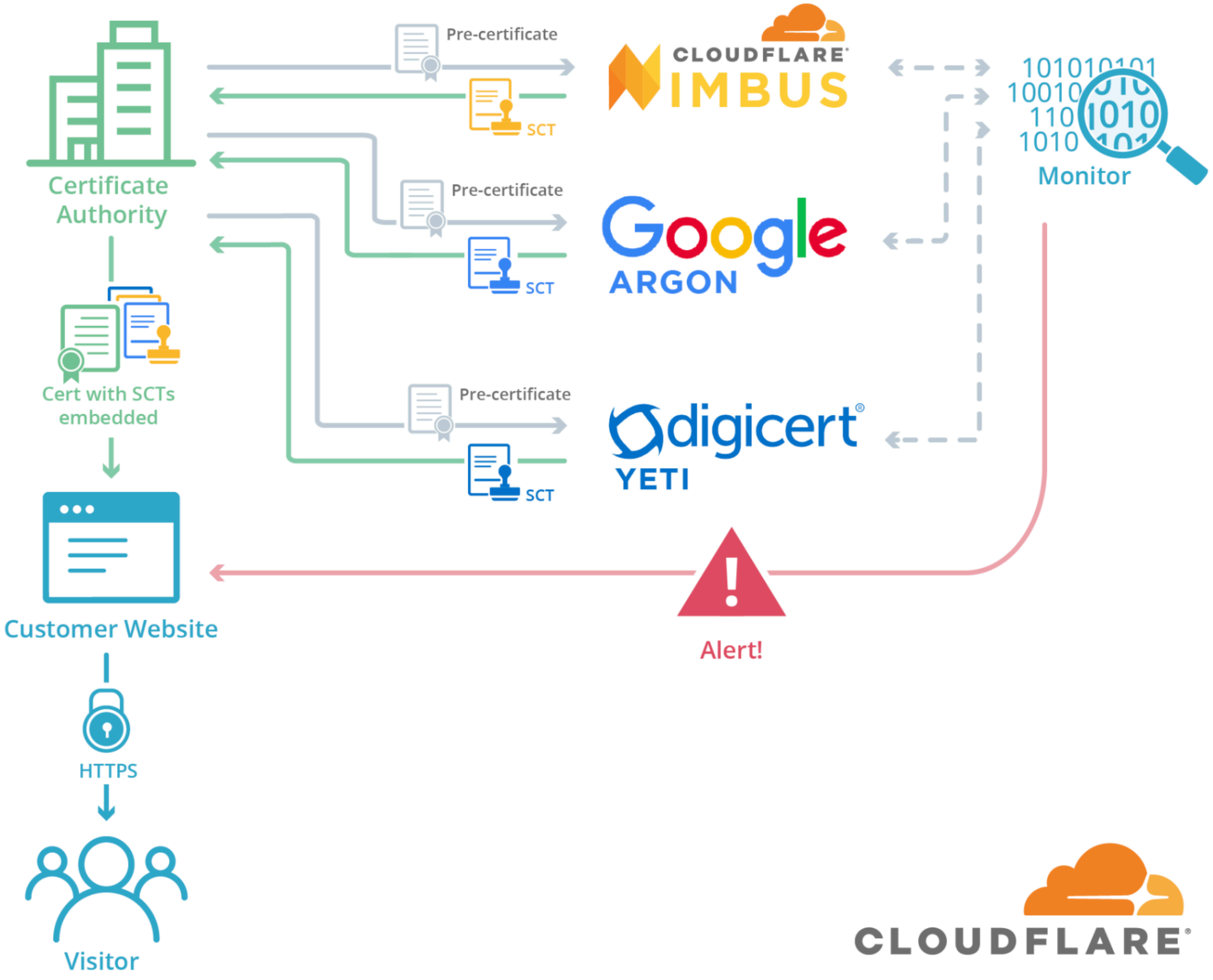
Using PIR for SCT auditing:

- Keys are SCTs
- Values are SCT inclusion proofs
- Client uses PIR to fetch the inclusion proof for a SCT without revealing the SCT to the CT log

Outline

1. Overview of certificate transparency
2. Client auditing
3. **Log monitoring**
4. Rollout and remaining challenges

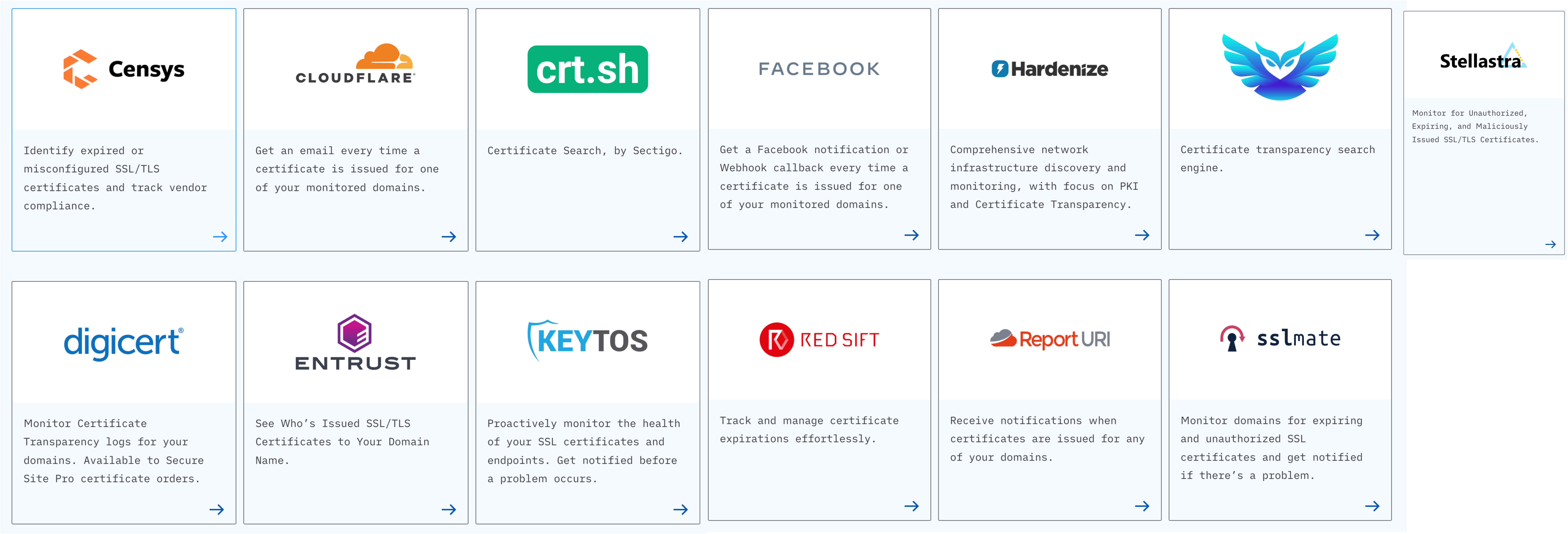
CT monitors



Monitors can help alert website owners to misissued certificates

https://blog.cloudflare.com/introducing-certificate-transparency-and-nimbus/

CT monitors



Censys

Identify expired or misconfigured SSL/TLS certificates and track vendor compliance.

CLOUDFLARE

Get an email every time a certificate is issued for one of your monitored domains.

crt.sh

Certificate Search, by Sectigo.

FACEBOOK

Get a Facebook notification or Webhook callback every time a certificate is issued for one of your monitored domains.

Hardenize

Comprehensive network infrastructure discovery and monitoring, with focus on PKI and Certificate Transparency.

Certificate transparency search engine.

Stellastra

Monitor for Unauthorized, Expiring, and Maliciously Issued SSL/TLS Certificates.

digicert

Monitor Certificate Transparency logs for your domains. Available to Secure Site Pro certificate orders.

ENTRUST

See Who's Issued SSL/TLS Certificates to Your Domain Name.

KEYTOS

Proactively monitor the health of your SSL certificates and endpoints. Get notified before a problem occurs.

RED SIFT

Track and manage certificate expirations effortlessly.

Report URI

Receive notifications when certificates are issued for any of your domains.

sslmate

Monitor domains for expiring and unauthorized SSL certificates and get notified if there's a problem.

https://certificate.transparency.dev/monitors/

What happens if a misissuance is detected?

- If operational error:
  - Domain owner communicates with CA owner to ensure certificate is revoked
  - If multiple operational errors from a single CA, may lead to distrust over time across browser (e.g., Symantec misissuances)
- If CA has been compromised:
  - Browsers may remove CA from “trusted CAs” list

CT monitors

- Some monitors also audit CT logs to ensure correct behavior
- Chrome audits CT logs to see if they meet Chrome’s requirements
  - Incorporate a certificate with an issued SCT within the MDD
  - Maintain high log availability (>=99%)
  - Logs are append-only (certificates never deleted)
  - Never present different views of state at different time / to different parties
  - …

`https://github.com/GoogleChrome/CertificateTransparency/blob/main/log_policy.md`

What happens if CT log misbehavior is detected?

- Multiple CT log failures since CT was rolled out
  - A few reasons for failure: excessive downtime, not including submitted certificates, compromised or reused private key, data corruption
- Log is marked as “retired” (read-only)
- Chrome and Apple require (at a high level)
  1. At least one SCT from a log that is approved at time of cert validation
  2. At least 2 SCTs from logs that were approve at time of cert issuance (for certs with lifetime <= 180 days)
- Retired logs count towards second requirement (but not first)

CT logs can be challenging to maintain

June 2023

- Single bit flip detected in DigiCert Yeti2022 CT log
- Error is not recoverable: already released signed tree root, so fixing would requiring finding the SHA256 pre-image
- No evidence that it was due to any error; could be due to hardware fault or cosmic ray
- Log was retired

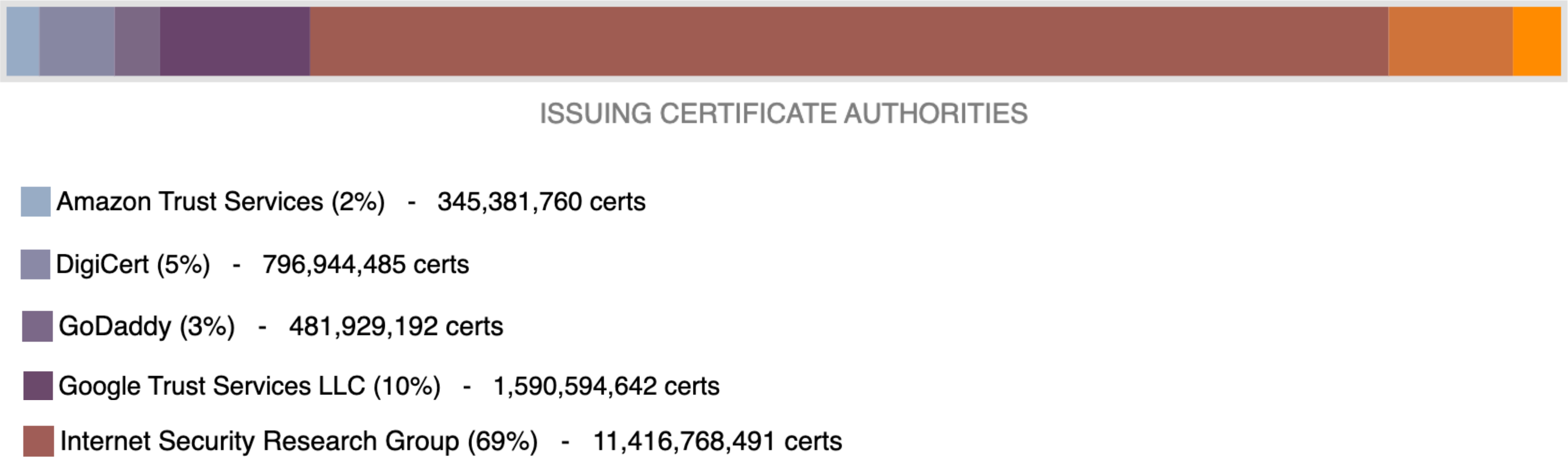
Very strong integrity and availability requirements for CT logs

- Drawback: limits the organizations operating logs

Outline

1. Overview of certificate transparency
2. Client auditing
3. Log monitoring
4. **Rollout and remaining challenges**

How many certificates CAs issue

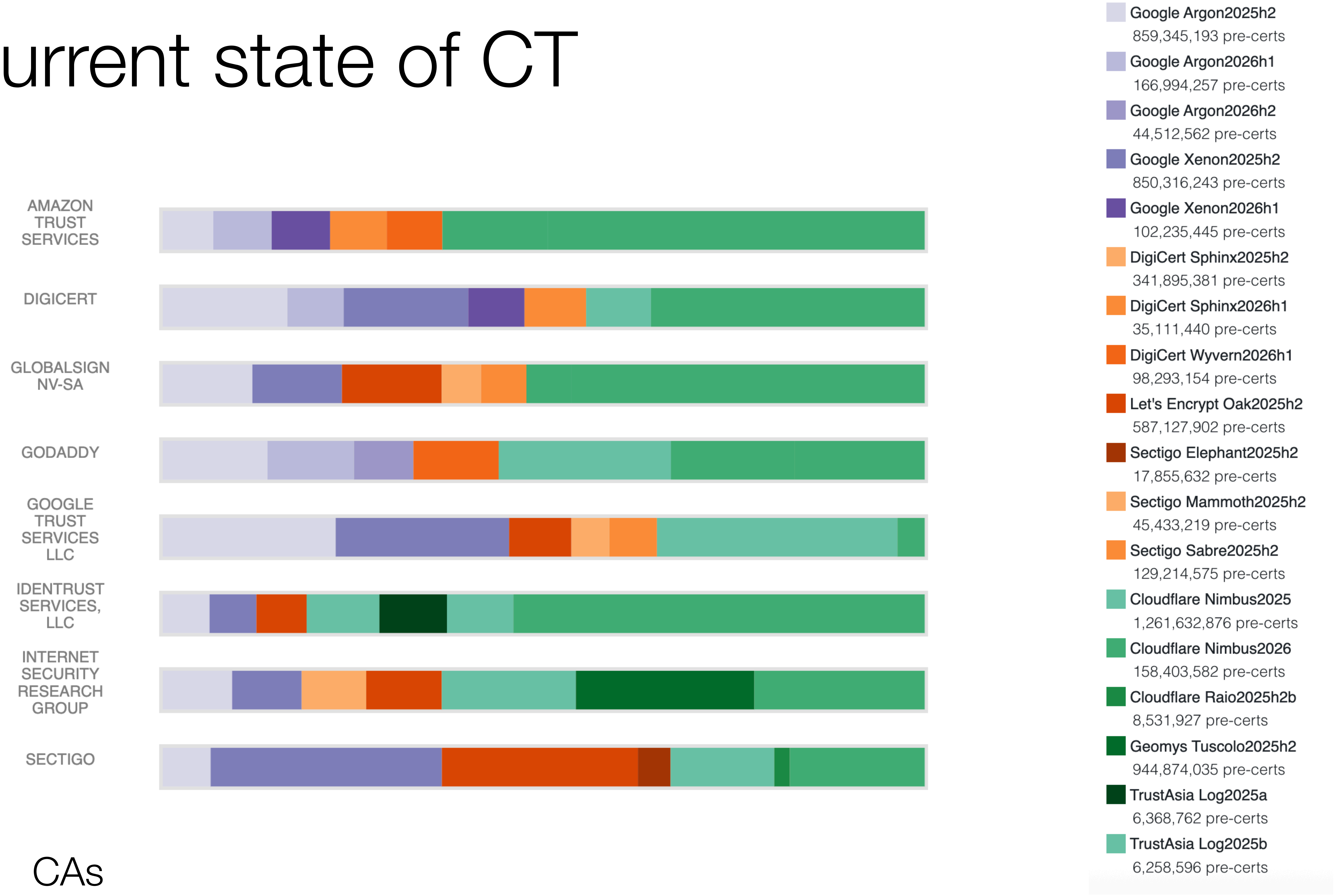


ISSUING CERTIFICATE AUTHORITIES

Amazon Trust Services (2%)	-	345,381,760 certs
DigiCert (5%)	-	796,944,485 certs
GoDaddy (3%)	-	481,929,192 certs
Google Trust Services LLC (10%)	-	1,590,594,642 certs
Internet Security Research Group (69%)	-	11,416,768,491 certs

https://ct.cloudflare.com/

Current state of CT



Google Argon2025h2	859,345,193 pre-certs
Google Argon2026h1	166,994,257 pre-certs
Google Argon2026h2	44,512,562 pre-certs
Google Xenon2025h2	850,316,243 pre-certs
Google Xenon2026h1	102,235,445 pre-certs
DigiCert Sphinx2025h2	341,895,381 pre-certs
DigiCert Sphinx2026h1	35,111,440 pre-certs
DigiCert Wyvern2026h1	98,293,154 pre-certs
Let's Encrypt Oak2025h2	587,127,902 pre-certs
Sectigo Elephant2025h2	17,855,632 pre-certs
Sectigo Mammoth2025h2	45,433,219 pre-certs
Sectigo Sabre2025h2	129,214,575 pre-certs
Cloudflare Nimbus2025	1,261,632,876 pre-certs
Cloudflare Nimbus2026	158,403,582 pre-certs
Cloudflare Raio2025h2b	8,531,927 pre-certs
Geomys Tuscolo2025h2	944,874,035 pre-certs
TrustAsia Log2025a	6,368,762 pre-certs
TrustAsia Log2025b	6,258,596 pre-certs

CAs

CT logs

https://ct.cloudflare.com/

Open question: does the client get the right root hash?

- To detect a log giving different root hashes to different clients, client should gossip
- Why is this hard?
  - Direct gossiping between clients doesn't fit well into existing Internet model
- What are some options for addressing this problem?
  - Servers help clients exchange a few root hashes (requires server updates)
  - Trusting the browser vendor to include the correct root hash

Other transparency log application: binary transparency

- Supply chain attack: Attacker compromises software shipped to users
- How can I be sure that I'm retrieving the right software binary?
- Use a transparency log to keep a publicly auditable list of software binaries
- Users can verify that their device is running the correct software (not a malicious binary)

Pixel Binary Transparency to Better Protect Pixel Owners Against Supply Chain Attacks

https://www.bitdefender.com/en-us/blog/hotforsecurity/pixel-binary-transparency-to-better-protect-pixel-owners-against-supply-chain-attacks

https://binary.transparency.dev/

Other transparency log application: key transparency

- How does Alice know that she’s sending a message to Bob?
- Messaging service (e.g., WhatsApp) could give Alice the attacker’s key instead of Bob’s key in order to launch a man-in-the-middle attack
- Key transparency: help Alice get the right public key to message Bob
- Next class!
- *Guest speaker: Kevin Lewi (Meta)*

Group project presentations

- Presentation should be 15 minutes, 5 minutes after for questions
- When preparing the presentation, think about:
  - How does the paper connect to the class topic for the day?
  - What is the motivation for this work?
  - What is the main contribution of the work, and what are any limitations?
  - What is the core technical insight?
  - How does the paper build on existing work?
- Goal is not to present every technical detail in the paper (although you should be ready for questions), but to explain what’s interesting about the paper

References

Machanavajjhala, A., Kifer, D., Gehrke, J., & Venkitasubramaniam, M. (2007). l-diversity: Privacy beyond k-anonymity. *Acm transactions on knowledge discovery from data (tkdd).*

Meiklejohn, S., DeBlasio, J., O'Brien, D., Thompson, C., Yeo, K., & Stark, E. (2022). SoK: SCT auditing in certificate transparency. *arXiv preprint arXiv:2203.01661*.

Stark, Emily, Joe DeBlasio, and Devon O'Brien. "Certificate transparency in google chrome: Past, present, and future." *IEEE Security & Privacy* 19.6 (2021): 112-118.

L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5), 2002; 571- 588.

https://www.bitdefender.com/en-us/blog/hotforsecurity/pixel-binary-transparency-to-better-protect-pixel-owners-against-supply-chain-attacks

https://binary.transparency.dev/

https://ct.cloudflare.com/

https://www.eff.org/deeplinks/2011/09/post-mortem-iranian-diginotar-attack

https://www.wired.com/2011/09/diginotar-hacker/

https://blog.torproject.org/diginotar-debacle-and-what-you-should-do-about-it/

https://www.agwa.name/blog/post/how_ct_logs_fail

https://groups.google.com/a/chromium.org/g/ct-policy/c/hxNohyZncfQ/m/_ak9BQssAAAJ

https://support.apple.com/en-us/103214

https://blog.cloudflare.com/introducing-certificate-transparency-and-nimbus/

https://docs.google.com/document/d/16G-Q7iN3kB46GSW5b-sfH5MO3nKSYyEb77YsM7TMZGE/edit?tab=t.0#heading=h.bat9awopsp53

https://educatedguesswork.org/posts/transparency-part-1/

https://educatedguesswork.org/posts/transparency-part-2/#fn1

https://queue.acm.org/detail.cfm?id=2668154