SCT Auditing: Revisited

Lena Heimberger
Research Intern
Cloudflare, Inc.

Signed Certificate Timestamps (SCTs)

Promise of (eventual) public logging

Allows user to check they got a certificate that a log has seen

Auditing leaks browsing history
→ SCT auditing problem

State of the Art : Auditing by Browsers

(some)
proxying ++

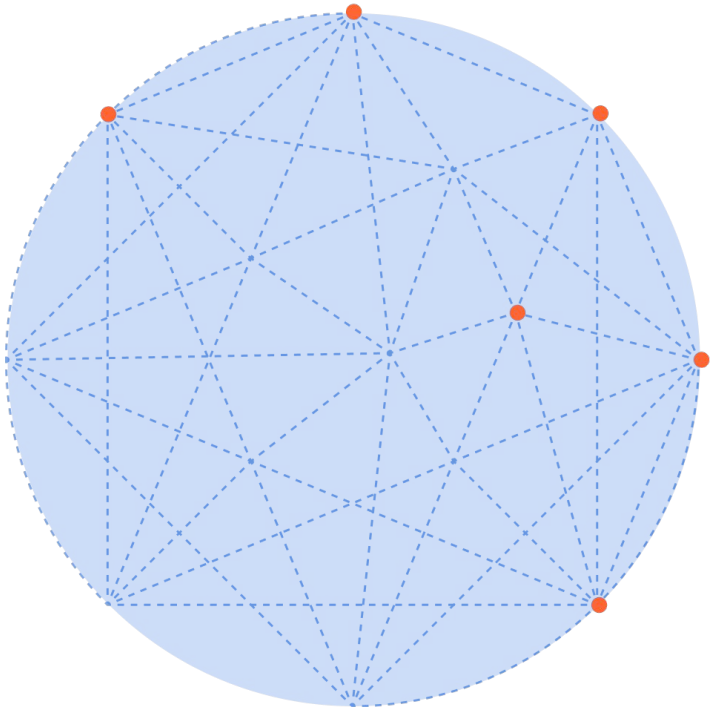
Safe Browsing API
Proxying

No Auditing

The Problem with Proxying

Traffic Analysis

Non-collusion assumption



Alternate Auditing Proposals

Sarah Meiklejohn, Joe DeBlasio, Devon O'Brien, Chris Thompson, Kevin Yeo, and Emily Stark

SoK: SCT Auditing in Certificate Transparency

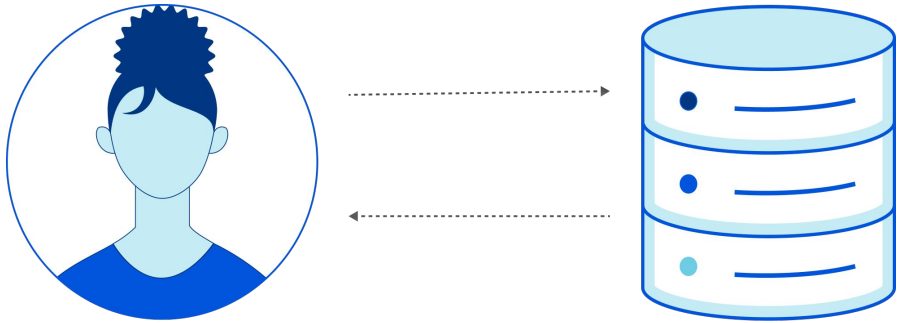
Abstract: The Web public key infrastructure is essential to providing secure communication on the Internet today, and certificate authorities play a crucial role in this ecosystem by issuing certificates. These authorities may misissue certificates or suffer misuse attacks, however, which has given rise to the Certificate Transparency (CT) project. The goal of CT is to store all issued certificates in public logs, which can then be valid by ensuring they are signed by, or have a signature chain rooted in, a trusted CA. If a CA is compromised, it can be used to issue false certificates that in turn would allow an attacker to eavesdrop on the communication between clients and a website. Furthermore, CAs may simply fail to fully verify a domain owner's identity and misissue a certificate. Both of these scenarios have happened [67], which illustrates the need for additional

Private Information Retrieval: Revisited

Natural solution, cryptographic security

New PIR proposals since SoK

Lookup by sequencing numbers
(added by static CT)



Fewer audits when combined with slow embedding proposals!

Slow embedding against fast quantum algorithms

Quantum-safe certificates are huge

Slow Embedding Certificates

Smaller, faster, quantum-secure!

Can’t get new certificate immediately

When is slow embedding too slow?

Register New Domain

Overlooked Certificate Renewal

Unplanned Domain Move

CLOUDFLARE

0.01-0.1%

chance of randomly browsing a website
that would need immediate issuance

Requirements for PIR in SCT Auditing

Criteria	Ideal Case	Tolerable Case
Preprocessing	None	No per-client preprocessing
Computations dependent on database updates	No	In under 10 minutes
Leakage	None (information-theoretic security)	Better differential privacy than current deployments
Audit timing	Immediate auditing	Batched to 1 audit/day to save bandwidth

Batching the lookup

Several audits per day

Having the sum of hashes is enough

PIR schemes support batching natively

State of the Art: Private Information Retrieval

SEAL

- 3 MB public parameters
  80-328 KB query
- ⊕ Constant communication size!
- ⊖ Per-client storage when reusing public parameters

Hintless/Frodo

- 3 MB query
- ⊕ No state
- ⊖ Large query size
- ⊖ Database preprocessing, updating unclear
- ⊖ Communications dependent on database size

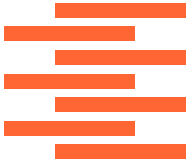
Spiral

- 8 MB public parameters,
  37 KB query
- ⊕ Good tradeoffs in parameters
- ⊖ Per-client storage (public parameters)
- ⊖ Communication dependent on database size

Conclusion

Current auditing

Very small sample, proxied communication!



Static CT

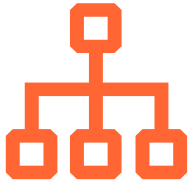
Enable lookup by sequence number instead of by hash

New PIR protocols

Very active research field.

Leveraging Slow Embedding

Potentially reduce need for lookups by a very large factor!



New insights

Only one PIR call necessary to get batched auditing!

Better than generic

Less communication than trivial download