SCT AUDITING
IN CERTIFICATE TRANSPARENCY

SARAH MEIKLEJOHN, JOE DEBLASIO, DEVON O’BRIEN, CHRIS THOMPSON, KEVIN YEO, AND EMILY STARK (GOOGLE)
CERTIFICATE ISSUANCE

- verifies the certificate
CERTIFICATE TRANSPARENCY

- promises to include (SCT)
- appends entry to log
- provides inclusion proofs

this is done according to the client’s policy (how many SCTs, which log operators, etc.)

- verifies the certificate
- verifies the SCTs
CERTIFICATE TRANSPARENCY

CA

domain operator

client

log operator

- verifies the certificate
- verifies the SCTs

- promises to include (SCT)
- appends entry to log
- provides inclusion proofs

monitor can notify domain operators about newly issued certificates

Inspect

d1

d2

d3

d4

h1

h2

h3

h4

h_{1234}

h_{12}

h_{34}

h1_{234}

+ SCTs

+ SCTs

Monitor
PROMISE VS. REALITY

CA

log operator

log

domain operator

+ SCTs

+ SCTs

client

- verifies the certificate
- verifies the SCTs

inspect

monitor

- promises to include (SCT)
- appends entry to log
- provides inclusion proofs
PROMISE VS. REALITY

CA + SCTs \rightarrow domain operator + SCTs \rightarrow client

- verifies the certificate
- verifies the SCTs

log operator

- promises to include (SCT)
- appends entry to log
- provides inclusion proofs

inspect \rightarrow monitor

HOW DO WE CHECK THAT CERTIFICATES SEEN BY CLIENTS ARE REALLY IN THE Log?
A NAIVE APPROACH

- CA
  - + SCTs
  - Log
  - log operator
    - promises to include (SCT)
    - appends entry to log
    - provides inclusion proofs

- domain operator
  - + SCTs
  - monitor

- client
  - verifies the certificate
  - verifies the SCTs

Inspect

fetch inclusion proof after the SCT deadline, report if it fails to verify
A NAIVE APPROACH

client

log operator

- verifies the certificate
- verifies the SCTs

- promises to include (SCT)
- appends entry to log
- provides inclusion proofs

revealing a certificate means revealing your browsing history

fetch inclusion proof after the SCT deadline, report if it fails to verify

clients are not in a position to report!
AUDITING AND REPORTING

- CA
  - + SCTs
- domain operator
  - + SCTs
- client
  - verifies the certificate
  - verifies the SCTs

- log operator
  - promises to include (SCT)
  - appends entry to log
  - provides inclusion proofs

- auditor
  - punishes misbehaving logs
  - doesn’t get SCTs directly

- monitor

Inspect
HOW DO WE CHECK THAT CERTIFICATES SEEN BY CLIENTS ARE REALLY IN THE LOG, AND INFORM AN AUDITOR IF NOT?
GATHERING REQUIREMENTS

Consider **querying** and **reporting** phases

1. **Functionality**: Does it work?
GATHERING REQUIREMENTS

Consider querying and reporting phases

1. **Functionality**: Does it work?
2. **Privacy**: What information do which parties learn?
   1. No privacy
   2. k-anonymity
   3. Provable unlinkability
GATHERING REQUIREMENTS

Consider querying and reporting phases

1. **Functionality**: Does it work?
2. **Privacy**: What information do which parties learn?
3. **Client-side performance**
   1. Bandwidth
   2. Computation
   3. Storage
GATHERING REQUIREMENTS

Consider querying and reporting phases

1. Functionality: Does it work?
2. Privacy: What information do which parties learn?
3. Client-side performance
4. Issuance latency: How much longer does it take to issue a cert?
GATHERING REQUIREMENTS

Consider querying and reporting phases

1. **Functionality**: Does it work?
2. **Privacy**: What information do which parties learn?
3. **Client-side performance**
4. **Issuance latency**: How much longer does it take to issue a cert?
5. **Server-side performance**: What would it cost to run this?
GATHERING REQUIREMENTS

Consider querying and reporting phases

1. **Functionality**: Does it work?
2. **Privacy**: What information do which parties learn?
3. **Client-side performance**
4. **Issuance latency**: How much longer does it take to issue a cert?
5. **Server-side performance**: What would it cost to run this?
6. **Threat model**: What trust assumptions are needed?
GATHERING REQUIREMENTS

Consider querying and reporting phases

1. **Functionality**: Does it work?
2. **Privacy**: What information do which parties learn?
3. **Client-side performance**
4. **Issuance latency**: How much longer does it take to issue a cert?
5. **Server-side performance**: What would it cost to run this?
6. **Threat model**: What trust assumptions are needed?
7. **Near-term deployability**: Could this be deployed in 2-3 years?
GATHERING PROPOSALS

Identified proposals that were compatible with Certificate Transparency as it exists today in publications, experimental deployments, and posts on mailing lists.

Also considered proposals for related problems of:
- Safe Browsing
- Checking for certificate revocation
- Checking for compromised credentials
OUTCOMES

Existing proposals don’t take into account all the dimensions of this problem:

- Certificates cannot contain sequence number (rapid issuance)
- Clients operate constrained devices
- Privacy degrades if each query achieves only k-anonymity
- Web servers are slow to change
- Clients need to report to an auditor in a private way

THANKS!
ANY QUESTIONS?