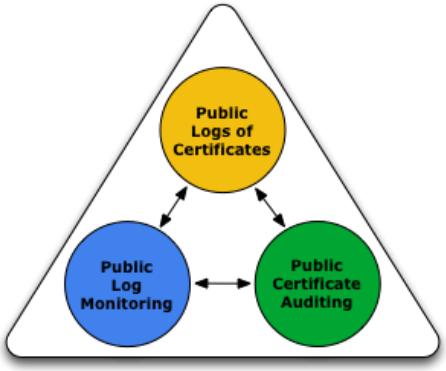
October 15, 2024

Verifiable Light-Weight Monitoring for Certificate Transparency Logs

Rasmus Dahlberg and Tobias Pulls

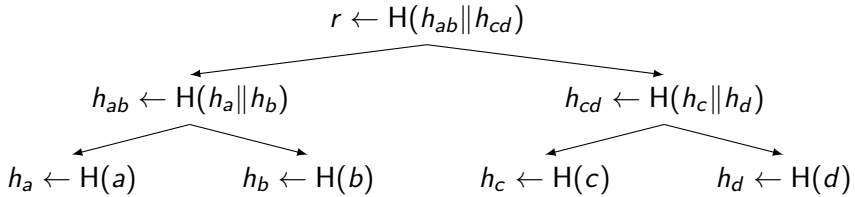
Certificate Transparency (CT)

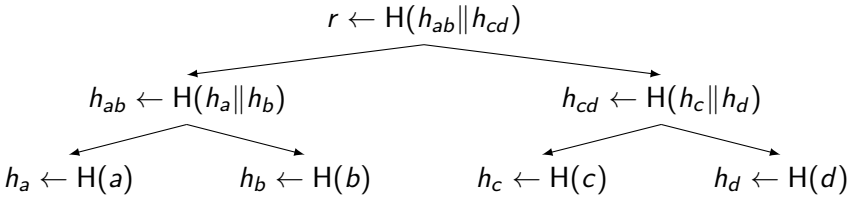
- Add transparency to CA ecosystem
- Publicly log all certificates
- No need[1] to trust the log
  - Membership proofs
  - Append-only proofs



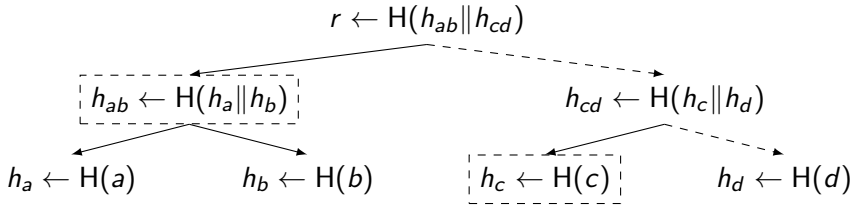
http://www.certificate-transparency.org/what-is-ct

[1] As deployed right now we do trust the logs tho ☹


$r \leftarrow \mathrm{H}(h_{ab} \| h_{cd})$
$h_{ab} \leftarrow \mathrm{H}(h_a \| h_b)$
$h_{cd} \leftarrow \mathrm{H}(h_c \| h_d)$
$h_a \leftarrow \mathrm{H}(a)$
$h_b \leftarrow \mathrm{H}(b)$
$h_c \leftarrow \mathrm{H}(c)$
$h_d \leftarrow \mathrm{H}(d)$



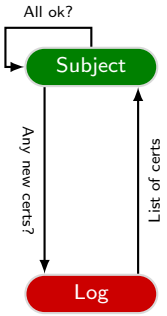
- Append new certificates in batches
- Sign tree head every hour ➠ STH



- Traverse tree from root to leaf
- Grab all sibling hashes on the way

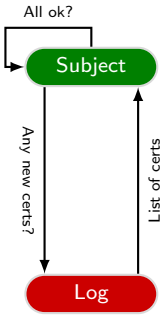
Two approaches towards monitoring a CT log

Self-monitoring



Two approaches towards monitoring a CT log

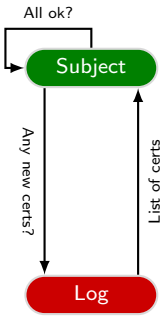
Self-monitoring



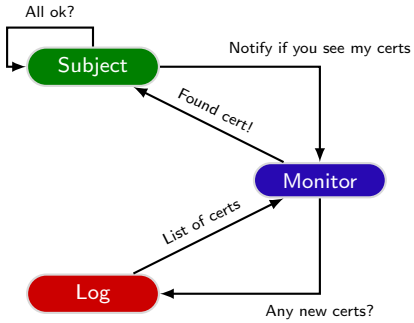
- ☹ Continuous uptime
- ☹ Download everything

Two approaches towards monitoring a CT log

Self-monitoring



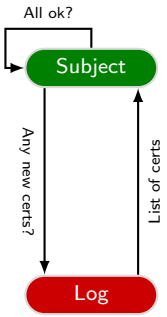
Monitoring-as-a-service



- ☹ Continuous uptime
- ☹ Download everything

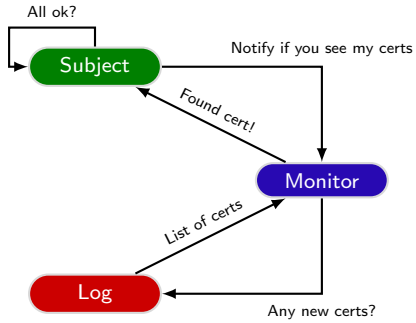
Two approaches towards monitoring a CT log

Self-monitoring



- ☹ Continuous uptime
- ☹ Download everything

Monitoring-as-a-service



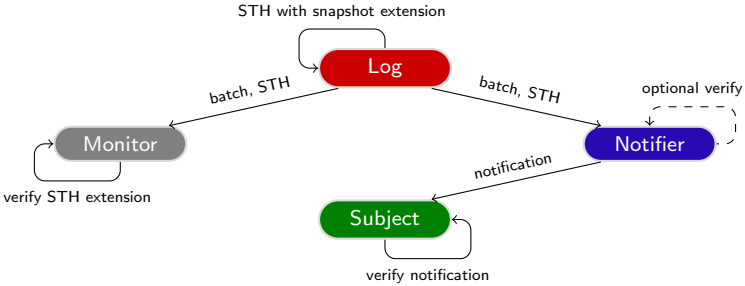
- ☺ Ezpz
- ☹ Trusted 3rd party

Notion of what we would like to achieve

'Wish list'

- CT/bis backwards compatibility
- Piggy-back on gossip-audit model
- Self-monitor wildcards w/o full download
- Reduced 3rd party monitoring trust

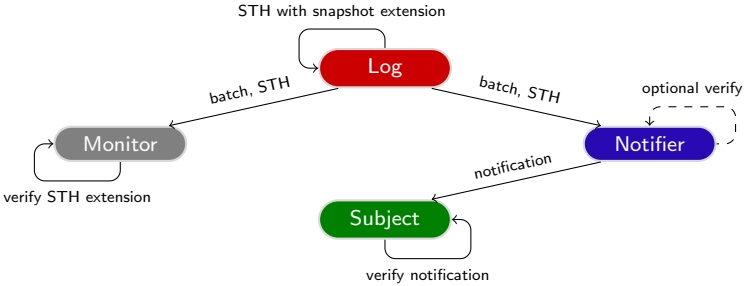
An overview of light-weight monitoring



- A new Merkle tree for each batch
- Add snapshot to STH as extension

One wildcard (non-)membership notification per STH

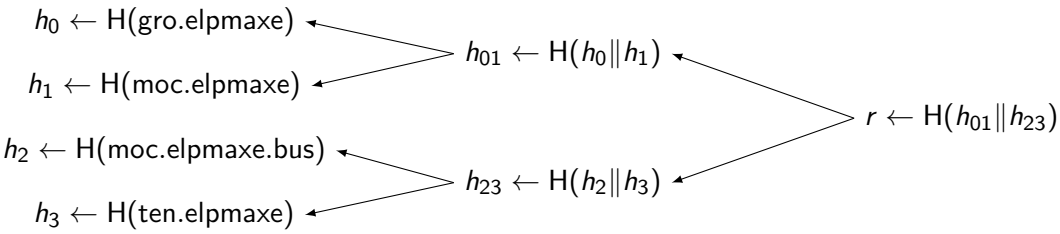
An overview of light-weight monitoring



- A new Merkle tree for each batch
- Add snapshot to STH as extension

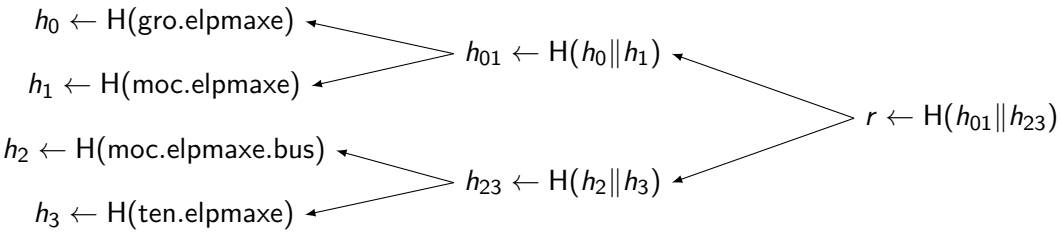
One wildcard (non-)membership notification per STH
How do you know if you got all notifications ➠ index extension

Wildcard notifications



- Merkleize reverse-sorted list
- Wildcard proof ➠ at most two audit paths

Wildcard notifications



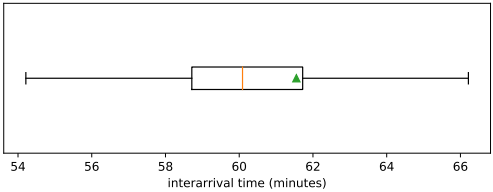
- Merkleize reverse-sorted list
- Wildcard proof ➠ at most two audit paths

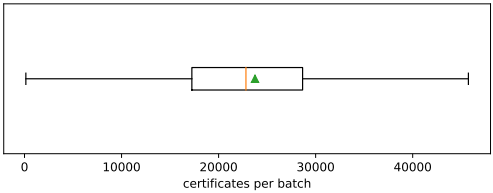
Security of this data structure? It is still just a Merkle tree...

- PoC: 351 lines of Go[2]
- Interesting metrics
  - Snapshot creation time
  - Proof generation time
  - Proof verification time
  - Bandwidth overhead
- Two log characteristics that matter
  - STH frequency
  - Batch size

We observed all Chrome-included logs for eight months to determine these characteristics

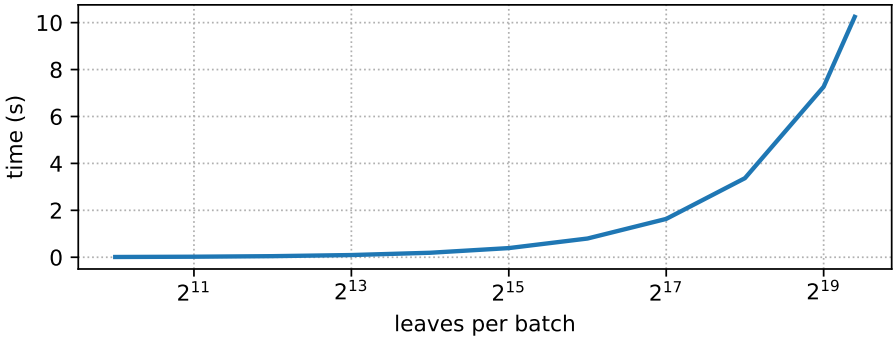
Google's Icarus Log





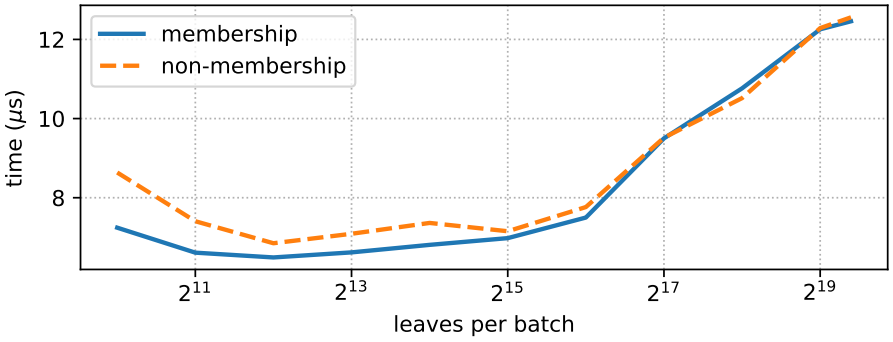
[2] https://github.com/rgdd/lwm

Snapshot creation time



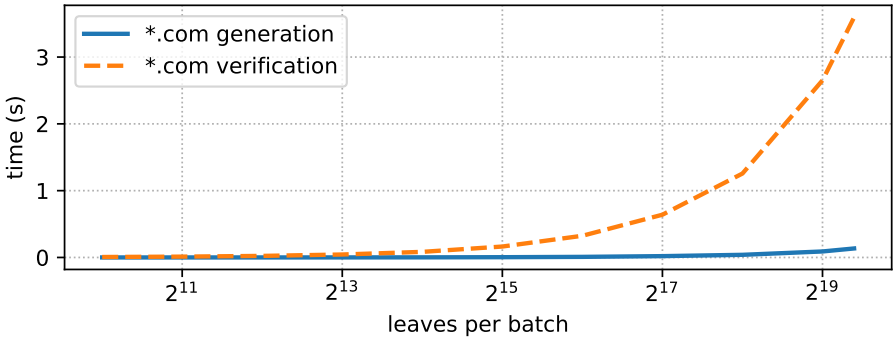
Negligible in comparison to STH issuance rate (1h)

Proof generation time



At least 288M non-membership proofs per hour on a single core

Proof generation and verification for *.com



352k matches in max batch ➠ 29k proofs per hour on a single core

Bandwidth overhead

Audit paths max batch size ➠ 1 KB

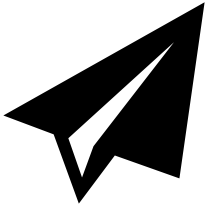
Self-monitor compare to median batch size of 32.6 MB

Notifier 288M audit paths per hour ➠ 640 Mbps

http://blog.coviam.com/wp-content/uploads/2016/07/Performance-Evaluation-Process-z.jpg

Take away

- Unfortunate if CT monitoring relies on trusted parties
- Light-weight monitoring
  - One verifiable wildcard notification per batch
  - Untrusted notification component with push/pull model
  - Untrusted log ⇛ rely on one honest monitor
  - Trusted log ⇛ no need to also trust monitor



Any questions?