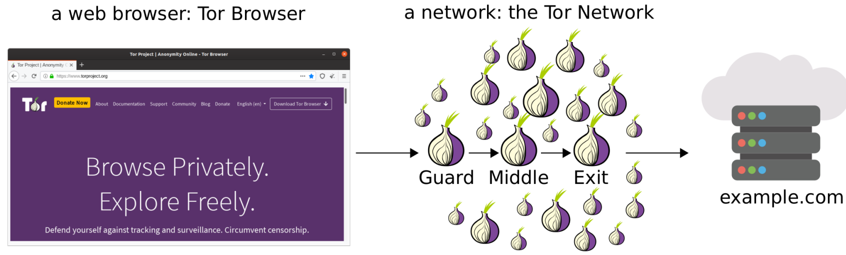
Certificate Transparency in Tor and Sigsum Logging

October 15, 2024

Rasmus Dahlberg

CT in Tor · · · Halftime · · · Sigsum Logging

Tor crash course


a web browser: Tor Browser
a network: the Tor Network
Tor Project | Anonymity Online - Tor Browser
Tor Project | Anonymity O
https://www.torproject.org
Tor
Donate Now
About
Documentation
Support
Community
Blog
Donate
English (en)
Download Tor Browser
Guard
Middle
Exit
example.com

Browse Privately.
Explore Freely.

Defend yourself against tracking and surveillance. Circumvent censorship.

[1] Credit: figure created by Tobias Pulls

[2] Design: https://murdoch.is/papers/tor14design.pdf

Tor Browser

- Firefox derivative
- Route all traffic through Tor
- Prevent user activity on one site from being linked to activity on another
- Do not write any state to disk
- ...

[1] Credit: Tom Ritter, see https://ritter.vg/p/tor-v1.6.pdf

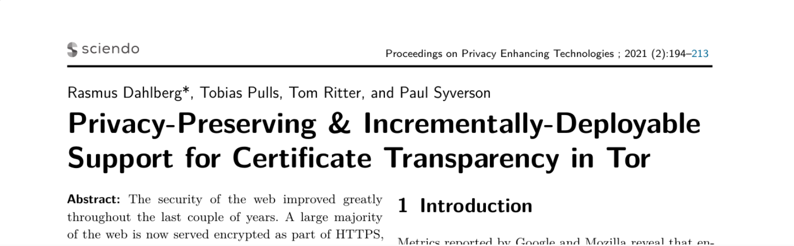
[2] Design: https://2019.www.torproject.org/projects/torbrowser/design/

Tor Browser

- Firefox derivative
- Route all traffic through Tor
- Prevent user activity on one site from being linked to activity on another
- Do not write any state to disk
- ...

[1] Credit: Tom Ritter, see https://ritter.vg/p/tor-v1.6.pdf

[2] Design: https://2019.www.torproject.org/projects/torbrowser/design/



sciendo Proceedings on Privacy Enhancing Technologies ; 2021 (2):194–213

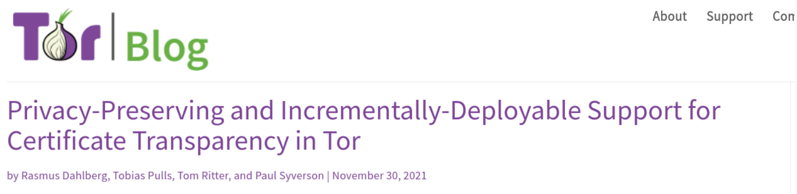
Rasmus Dahlberg*, Tobias Pulls, Tom Ritter, and Paul Syverson

Privacy-Preserving & Incrementally-Deployable Support for Certificate Transparency in Tor

Abstract: The security of the web improved greatly throughout the last couple of years. A large majority of the web is now served encrypted as part of HTTPS,

1 Introduction

Metrics reported by Google and Mozilla reveal that en-



Privacy-Preserving and Incrementally-Deployable Support for Certificate Transparency in Tor

by Rasmus Dahlberg, Tobias Pulls, Tom Ritter, and Paul Syverson | November 30, 2021

Problem statement

- Tor Browser does not enforce CT
- Guard against prominent threats
  - DigiNotar style attacks
  - Interception to deanonymize
- Go beyond “just CT compliance”

Problem statement

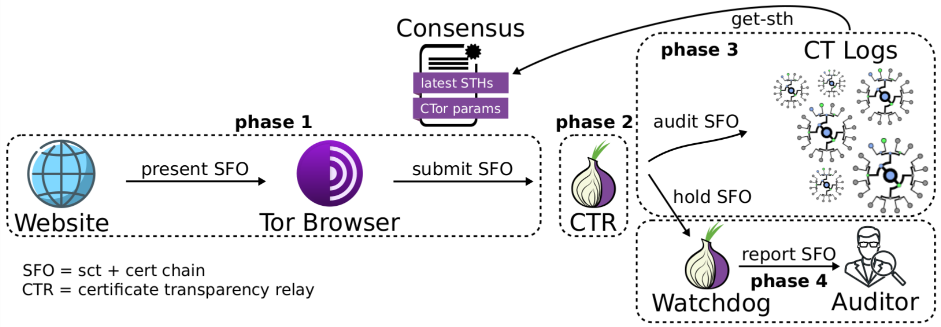
- Tor Browser does not enforce CT
- Guard against prominent threats
  - DigiNotar style attacks
  - Interception to deanonymize
- Go beyond “just CT compliance”

Attacker in Tor’s threat model + controls a CA and two CT logs

Incremental deployment

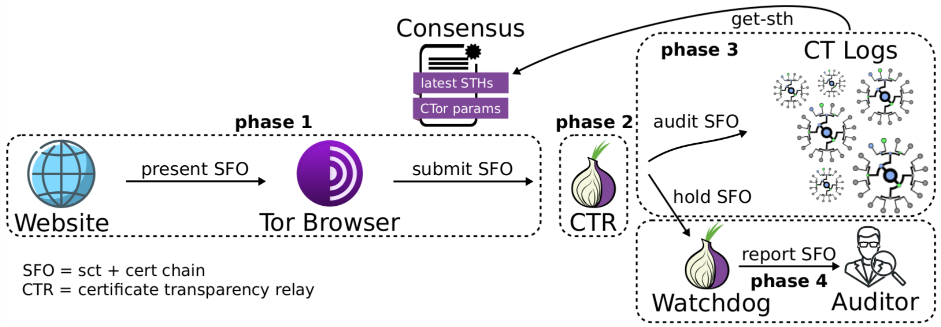
1. Catch up with CT compliant browsers	*pairs of logs* are trusted blindly
2. Steps towards decentralized verification	*some log* is trusted blindly
3. Fully decentralized verification	*no log* is trusted blindly

Full design



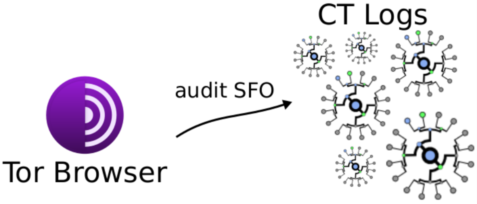
SFO = sct + cert chain
CTR = certificate transparency relay

Full design

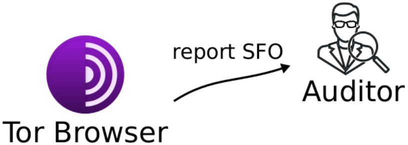


Security? Difficult to interfere without detection in any phase

Why not just...?

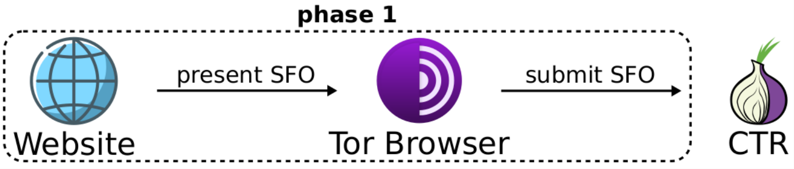


Fetch an inclusion proof



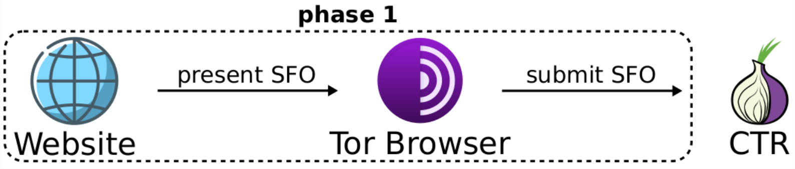
Rely on a centralized party

Phase 1: Submission



1. Probabilistic submit
2. Random CTR

Phase 1: Submission

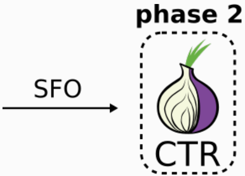


1. Probabilistic submit
2. Random CTR

Best attack: quickly take control over Tor Browser

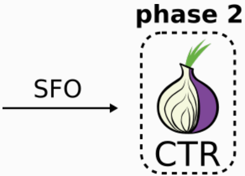
Phase 2: Buffering

1. Buffer until logging is required
2. Add a random delay to leak less
3. Cache audited SFOs to leak less



Phase 2: Buffering

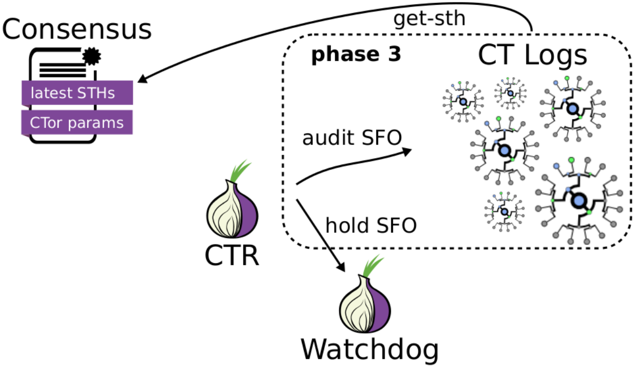
1. Buffer until logging is required
2. Add a random delay to leak less
3. Cache audited SFOs to leak less



Best attack: network-wide flush

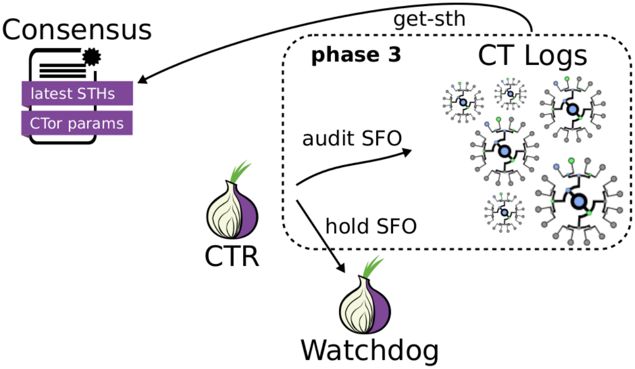
Phase 3: Auditing

1. Fetch inclusion proof
2. STH from Tor's consensus
3. Collaborate with a watchdog
    - CTR identification
    - "Tagging"



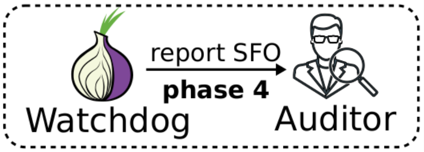
Phase 3: Auditing

1. Fetch inclusion proof
2. STH from Tor's consensus
3. Collaborate with a watchdog
   - CTR identification
   - "Tagging"



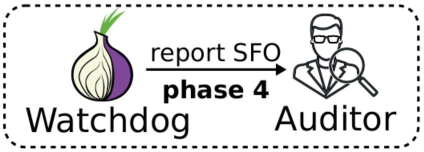
Best attack: quickly take control over CTR

Phase 4: Reporting



1. Report SFO on timeout

Phase 4: Reporting

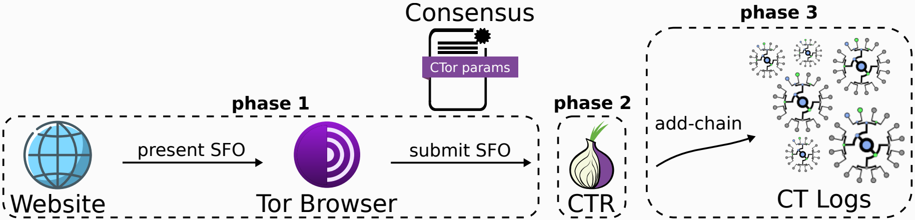


1. Report SFO on timeout

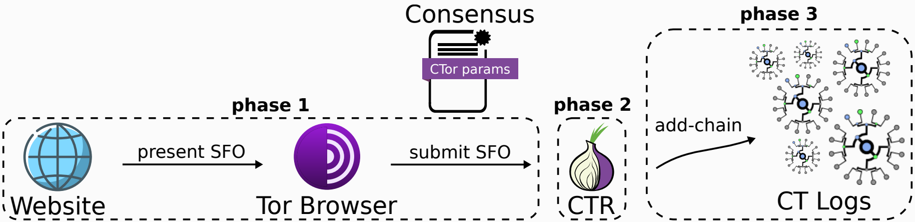
Best attack: n/a

This is quite the leap from “just CT compliance”

Incremental design



Incremental design



Use the log ecosystem against the attacker

Conclusion

- Tor's setting is quite different
- Delegated audiding is key here
- Roadmap from start to finnish

Resources

- PETS paper[1]
- PETS talk[2]
- Tor blog post[3]

Next steps

- Torspec proposal(s)
- Browser implementation
- Relay implementation

[1] https://petsymposium.org/2021/files/papers/issue2/popets-2021-0024.pdf

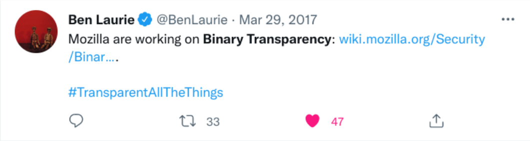
[2] https://www.youtube.com/watch?v=f7yDJOd6g3U

[3] https://blog.torproject.org/tor-certificate-transparency/

Halftime

Tweets you can probably relate to

Tweets you can probably relate to

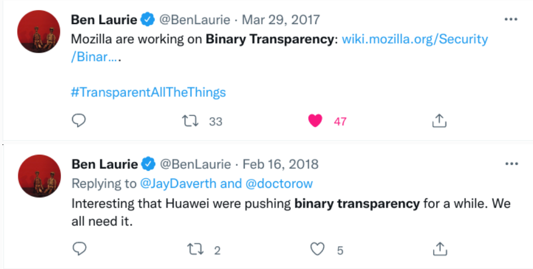


Ben Laurie @BenLaurie · Mar 29, 2017

Mozilla are working on **Binary Transparency**: wiki.mozilla.org/Security/Binar….

#TransparentAllTheThings

Tweets you can probably relate to



Ben Laurie @BenLaurie · Mar 29, 2017

Mozilla are working on **Binary Transparency**: wiki.mozilla.org/Security/Binar… .

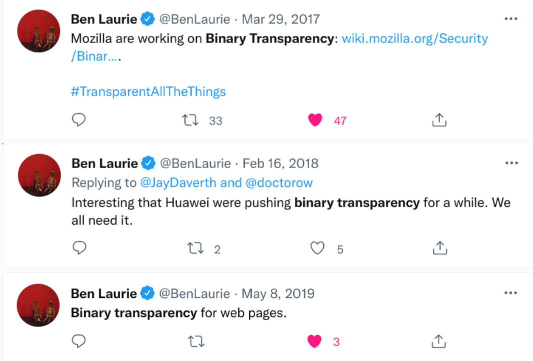
#TransparentAllTheThings

Ben Laurie @BenLaurie · Feb 16, 2018

Replying to @JayDaverth and @doctorow

Interesting that Huawei were pushing **binary transparency** for a while. We all need it.

Tweets you can probably relate to



Ben Laurie @BenLaurie · Mar 29, 2017

Mozilla are working on **Binary Transparency**: wiki.mozilla.org/Security /Binar... .

#TransparentAllTheThings

33 ♥ 47

Ben Laurie @BenLaurie · Feb 16, 2018

Replying to @JayDaverth and @doctorow

Interesting that Huawei were pushing **binary transparency** for a while. We all need it.

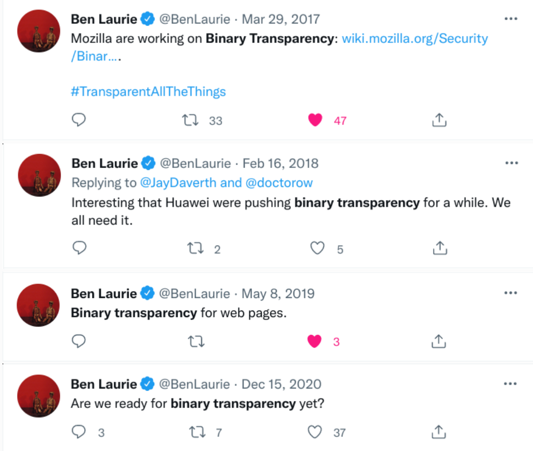
2 ♡ 5

Ben Laurie @BenLaurie · May 8, 2019

Binary transparency for web pages.

♥ 3

Tweets you can probably relate to



Ben Laurie @BenLaurie · Mar 29, 2017

Mozilla are working on **Binary Transparency**: wiki.mozilla.org/Security/Binar….

#TransparentAllTheThings

33 47

Ben Laurie @BenLaurie · Feb 16, 2018

Replying to @JayDaverth and @doctorow

Interesting that Huawei were pushing **binary transparency** for a while. We all need it.

2 5

Ben Laurie @BenLaurie · May 8, 2019

Binary transparency for web pages.

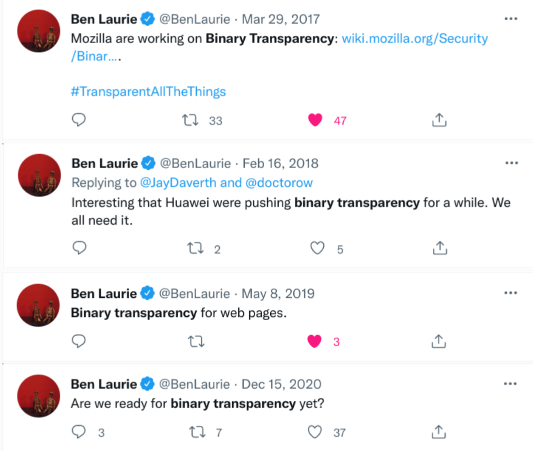
3

Ben Laurie @BenLaurie · Dec 15, 2020

Are we ready for **binary transparency** yet?

3 7 37

Tweets you can probably relate to



Ben Laurie @BenLaurie · Mar 29, 2017

Mozilla are working on **Binary Transparency**: wiki.mozilla.org/Security/Binar….

#TransparentAllTheThings

Ben Laurie @BenLaurie · Feb 16, 2018

Replying to @JayDaverth and @doctorow

Interesting that Huawei were pushing **binary transparency** for a while. We all need it.

Ben Laurie @BenLaurie · May 8, 2019

Binary transparency for web pages.

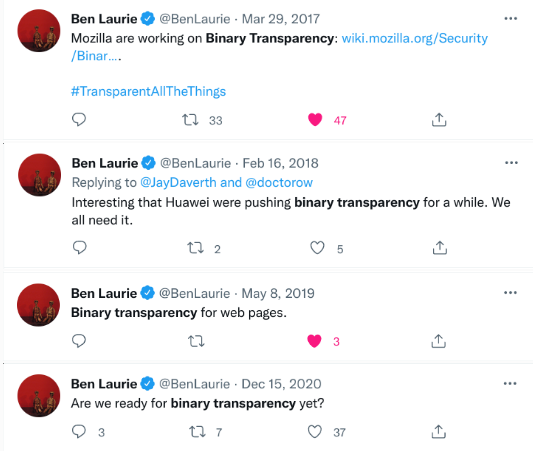
Ben Laurie @BenLaurie · Dec 15, 2020

Are we ready for **binary transparency** yet?

2022?

More initatives than can be counted on two hands

Tweets you can probably relate to



Ben Laurie @BenLaurie · Mar 29, 2017

Mozilla are working on **Binary Transparency**: wiki.mozilla.org/Security/Binar… .

#TransparentAllTheThings

Ben Laurie @BenLaurie · Feb 16, 2018

Replying to @JayDaverth and @doctorow

Interesting that Huawei were pushing **binary transparency** for a while. We all need it.

Ben Laurie @BenLaurie · May 8, 2019

Binary transparency for web pages.

Ben Laurie @BenLaurie · Dec 15, 2020

Are we ready for **binary transparency** yet?

2022?

More initatives than can be counted on two hands

https://binary.transparency.dev

...

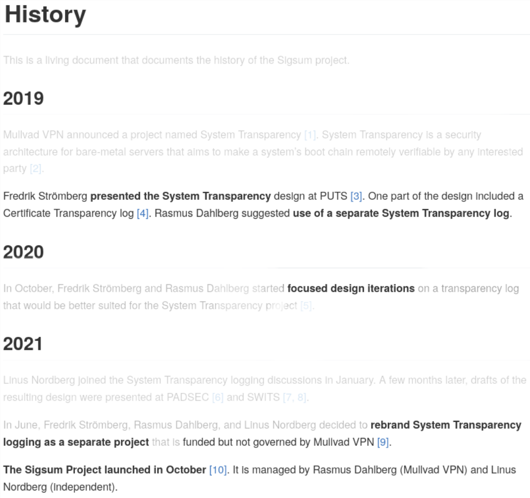
Common denominator?

Certificates
Executable binaries
Source code
TPM quotes
Onion address rulesets
Official documents
...

Where is the low-hanging fruit?

Meet the Sigsum project

- FOSS
- Signed checksums
- Enforcement of logging
- Minimal building block
- “Transparent key-usage”



History

This is a living document that documents the history of the Sigsum project.

2019

Mullvad VPN announced a project named System Transparency [1]. System Transparency is a security architecture for bare-metal servers that aims to make a system's boot chain remotely verifiable by any interested party [2].

Fredrik Strömberg **presented the System Transparency** design at PUTS [3]. One part of the design included a Certificate Transparency log [4]. Rasmus Dahlberg suggested **use of a separate System Transparency log**.

2020

In October, Fredrik Strömberg and Rasmus Dahlberg started **focused design iterations** on a transparency log that would be better suited for the System Transparency project [5].

2021

Linus Nordberg joined the System Transparency logging discussions in January. A few months later, drafts of the resulting design were presented at PADSEC [6] and SWITS [7, 8].

In June, Fredrik Strömberg, Rasmus Dahlberg, and Linus Nordberg decided to **rebrand System Transparency logging as a separate project** that is funded but not governed by Mullvad VPN [9].

The Sigsum Project launched in October [10]. It is managed by Rasmus Dahlberg (Mullvad VPN) and Linus Nordberg (independent).

https://git.sigsum.org/sigsum/tree/doc/history.md

Use-case - Signature Transparency

“Oh, a new signature was created. That’s weird. I’m at the gym.”

Use-case - Binary Transparency

“Oh, that’s the key binaries are signed with”
“By policy binaries are located at releases.example.com/$CHECKSUM”

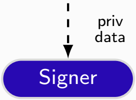
s/binary/something else/

Many answers and trade-offs

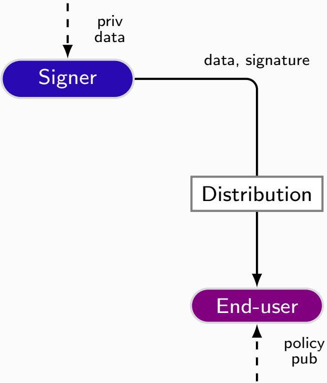
- Purpose of logging
- What is (not) logged
- Auditing, SCTs
- Gossip
- Anti-poison
- Anti-spam
- Sharding
- Privacy
- Simple API

Accept latency, no rich metadata, no complicated protocols and parsers

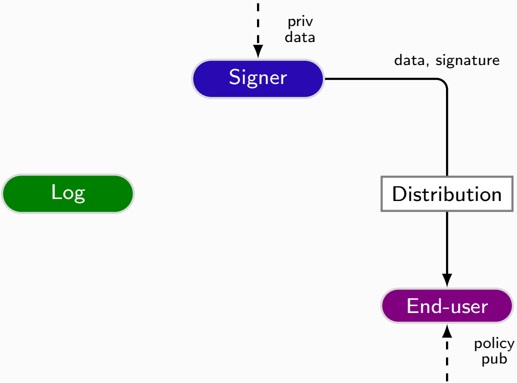
System overview



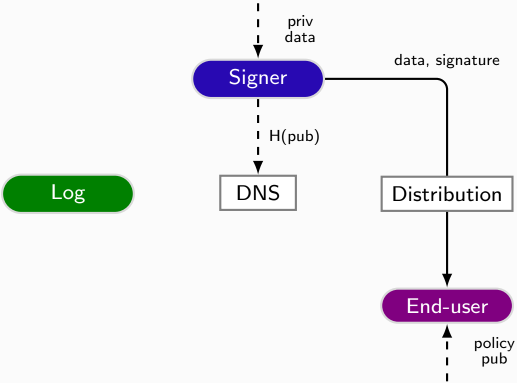
System overview



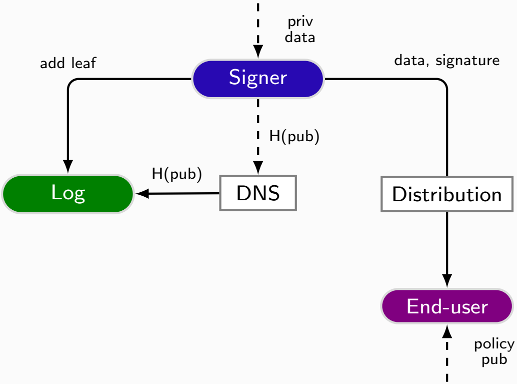
System overview



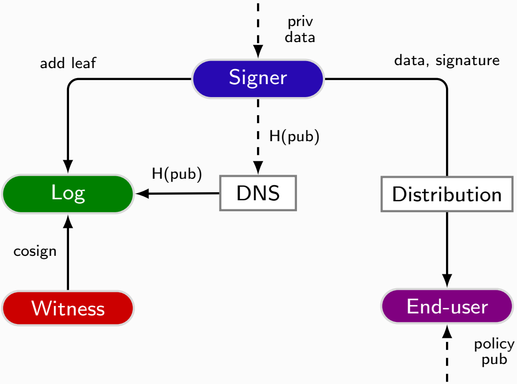
System overview



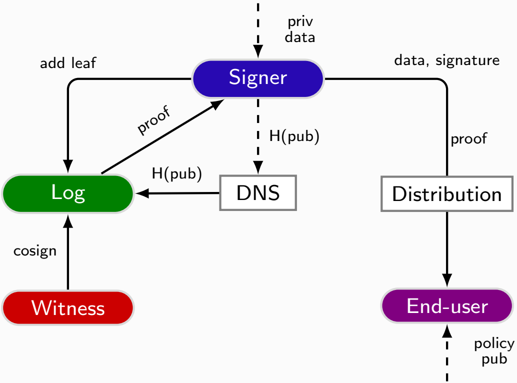
System overview



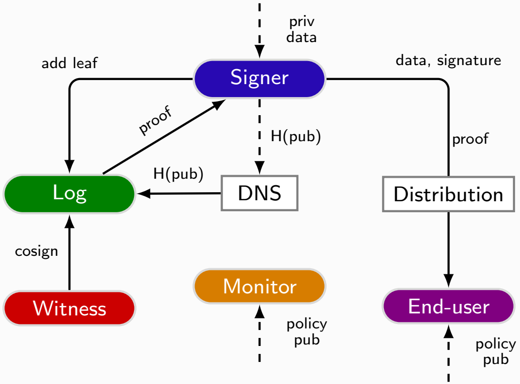
System overview



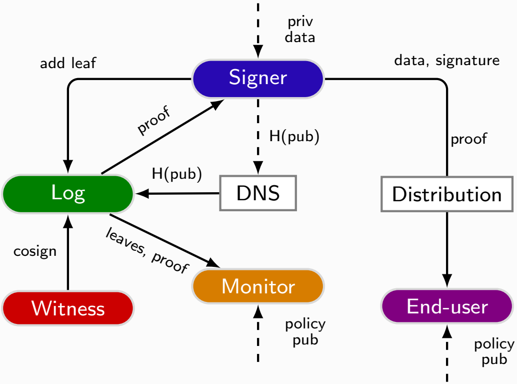
System overview



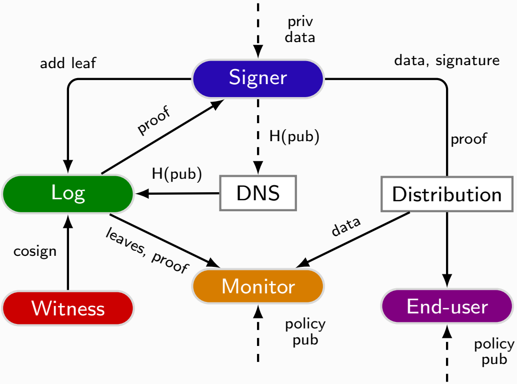
System overview



System overview

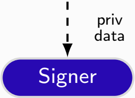


System overview

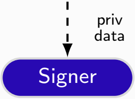


A step-by-step breakdown

Signing



Signing



priv data

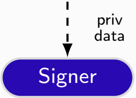
Signer

```
#define MAGIC_PREAMBLE "SSHSIG"

byte[6] MAGIC_PREAMBLE
string  namespace
string  reserved
string  hash_algorithm
string  H(message)
```

https://github.com/openssh/openssh-portable/blob/master/PROTOCOL.sshsig#L81

Signing



```
#define MAGIC_PREAMBLE "SSHSIG"

byte[6] MAGIC_PREAMBLE
string  namespace
string  reserved
string  hash_algorithm
string  H(message)
```

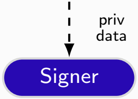
https://github.com/openssh/openssh-portable/blob/master/PROTOCOL.sshsig#L81

```
Values used by Sigsum (only Ed25519)


"tree_leaf:v0:<shard_hint>@sigsum.org"
""
"sha256"
message = H(data)
```

https://git.sigsum.org/sigsum/tree/doc/proposals/2021-11-ssh-signature-format.md

Signing



```
#define MAGIC_PREAMBLE "SSHSIG"

byte[6] MAGIC_PREAMBLE
string  namespace
string  reserved
string  hash_algorithm
string  H(message)
```

https://github.com/openssh/openssh-portable/blob/master/PROTOCOL.sshsig#L81

```
Values used by Sigsum (only Ed25519)


"tree_leaf:v0:<shard_hint>@sigsum.org"
""
"sha256"
message = H(data)
```

https://git.sigsum.org/sigsum/tree/doc/proposals/2021-11-ssh-signature-format.md

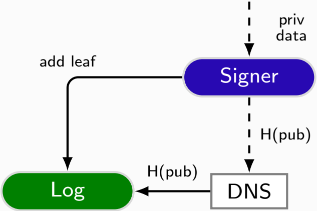
`ssh-keygen -Y` `signify` `minisign`

Why not support more signing formats and tools?

Submission

HTTP POST ASCII

- Shard hint
  - $\in$ [shard_start, now()]
- Message
- Signature
- Public key
- Domain hint
  - `_sigsum_v0.*` $\rightarrow$ H(pub)



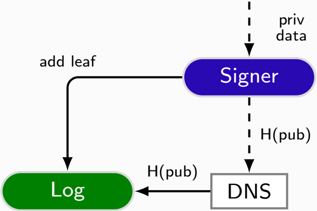
Submission

HTTP POST ASCII

- Shard hint
  - $\in$ [shard_start, now()]
- Message
- Signature
- Public key
- Domain hint
  - `_sigsum_v0.*` → H(pub)

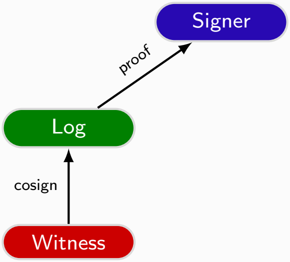
Stored leaf (136 bytes)

- Shard hint
- Checksum
- Signature
- Key hash



Bundling

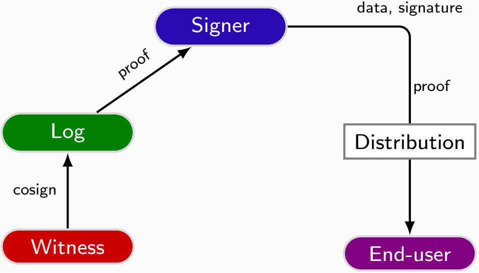
- Signer must **wait** for witnessing[1]
  - Append-only
  - Freshness
  - Some simplifications
- Proof of logging
  - Cosigned tree head
  - Inclusion proof



[1] Originally proposed by Syta et al.: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7546521

Bundling and Distribution

- Signer must **wait** for witnessing[1]
  - Append-only
  - Freshness
  - Some simplifications
- Proof of logging
  - Cosigned tree head
  - Inclusion proof

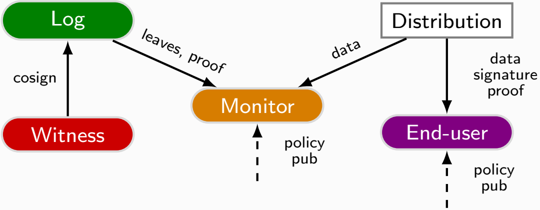


[1] Originally proposed by Syta et al.: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7546521

Verification

Example policy

- Known logs
- Known witnesses
- M-of-N (co)signatures



No reactive gossip/audit, offline verification by end-users (!)

Current status

- Solid foundation, hopefully(!)
- V0 design[1] and API[2] is pretty stable
- Public prototypes, log and witness
- Tooling? Kind of “pipe into `curl`”
- https://git.sigsum.org

https://bygg.se/valj-ratt-husgrund-till-din-villa/

[1] https://git.sigsum.org/sigsum/tree/doc/design.md

[2] https://git.sigsum.org/sigsum/tree/doc/api.md

Current status

- Solid foundation, hopefully(!)
- V0 design[1] and API[2] is pretty stable
- Public prototypes, log and witness
- Tooling? Kind of "pipe into `curl`"
- https://git.sigsum.org

https://bygg.se/valj-ratt-husgrund-till-din-villa/

Next steps: more feedback, tooling, mature code, SLA for a v0 log, eventually v1 spec

[1] https://git.sigsum.org/sigsum/tree/doc/design.md

[2] https://git.sigsum.org/sigsum/tree/doc/api.md

Take away

- Minimal building block
- Log a signed checksum
- Offline end-user verification
- Many potential use-cases
- Reach out to get involved[1]

[1] irc, matrix, email list, etc., are linked from https://www.sigsum.org

Q/A