

Trusted Third Parties (and where to find them)

Tjaden Hess
tjaden.hess@trailofbits.com

here!



Two flavors of auditor



Cryptographic Protocol

Participants



AKV auditor SIGNAL KEY TRANSPARENCY
Signal, Cloudflare and Trail of Bits



Cloudflare Plexi WHATSAPP KEY TRANSPARENCY (AKD)



crt.sh / SSLMate CERTIFICATE TRANSPARENCY



Witness (cosigner) SIGSUM / TRANSPARENCY.DEV



Rekor Monitor SIGSTORE REKOR



GO Client auditor GO CHECKSUM DATABASE

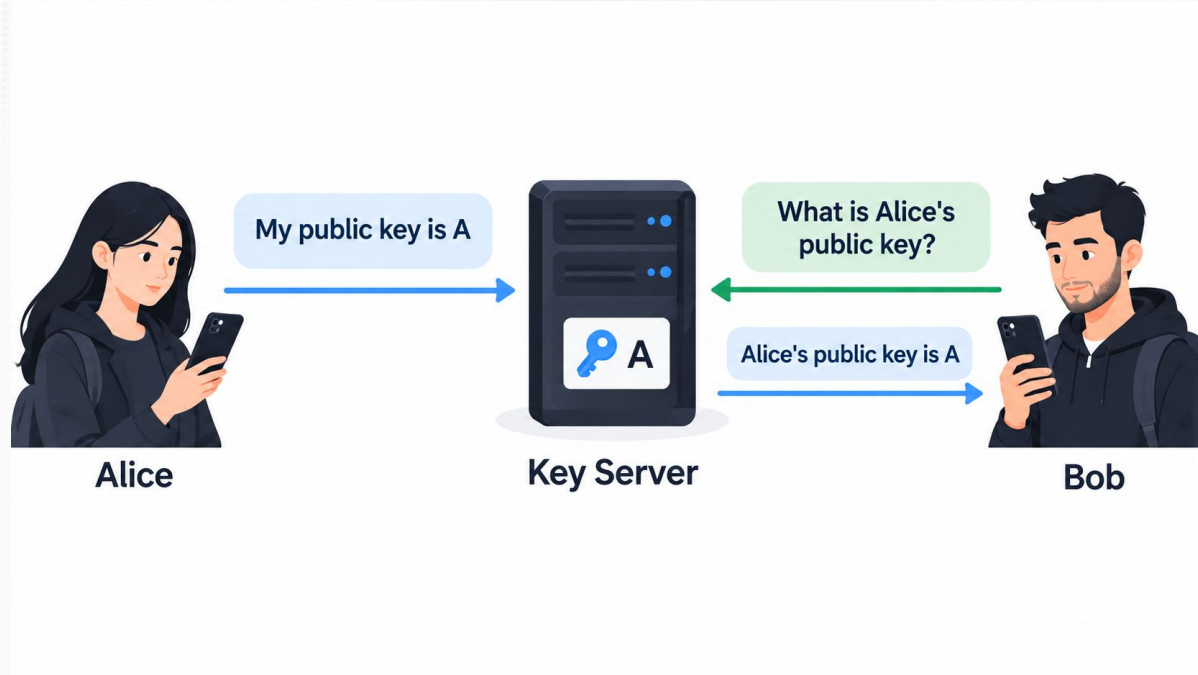
Skilled Humans



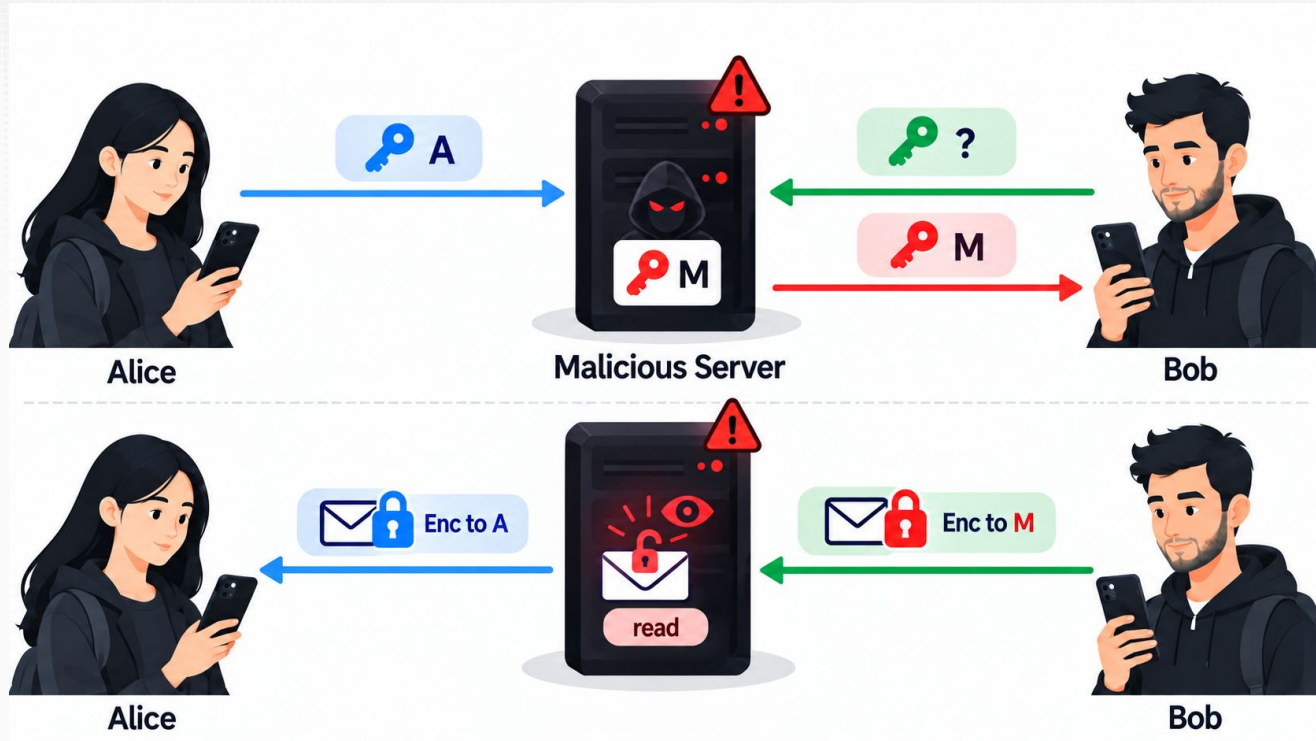
Why not both?

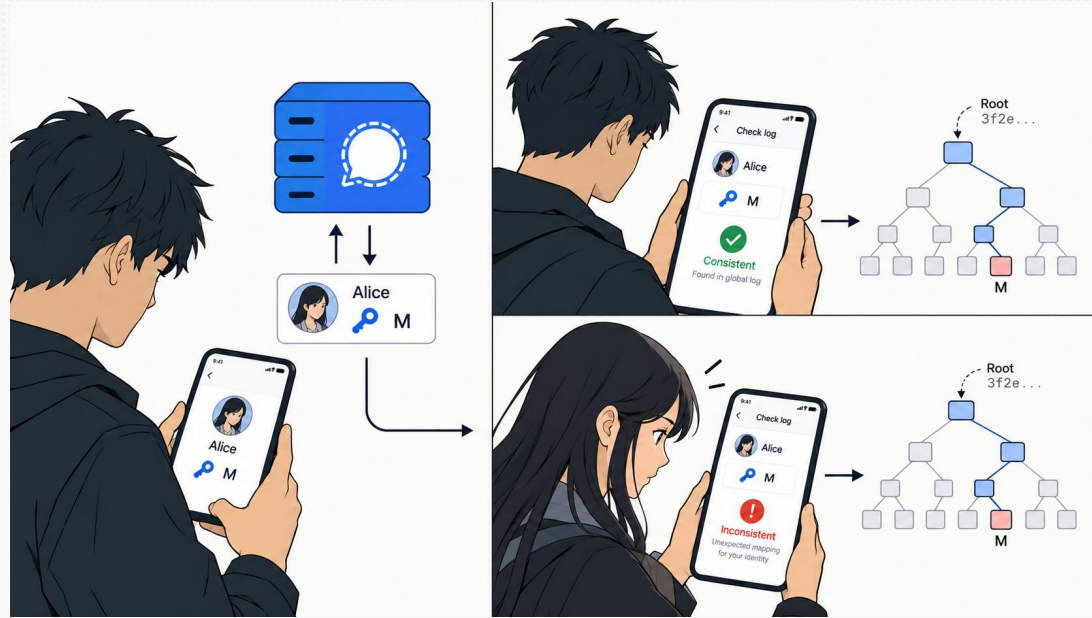
- Your code reviewers already understand the system and its guarantees
- Customers already rely on security review reports
- Auditing is a business of reputation – the product we sell is independence

An Example - Key Transparency

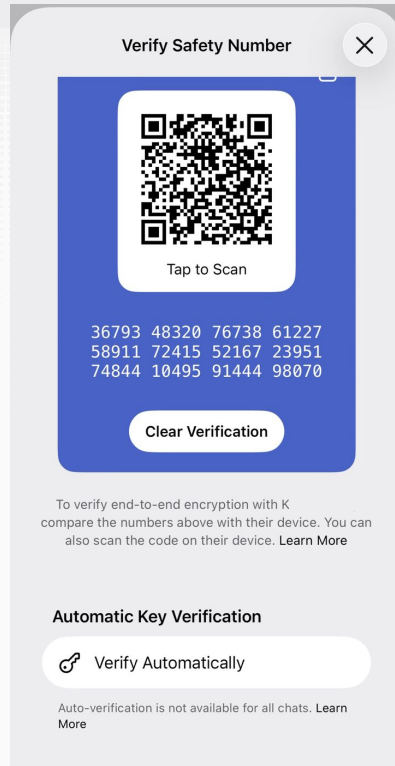
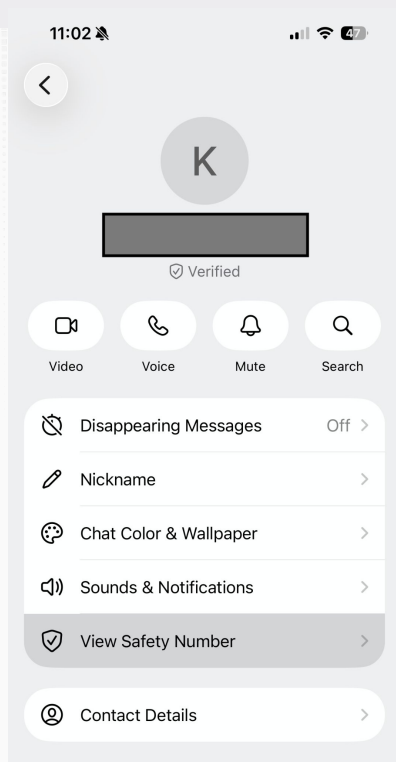


Key Transparency





Signal's Automatic Key Verification



Querying the log

- **POST /v1/key-transparency/search**
 - Look up a key's current mapping and get proofs
- **POST /v1/key-transparency/monitor**
 - Re-check a key already looked up, against newer log state
- **GET /v1/key-transparency/distinguished**
 - Fetch a recent "distinguished" head

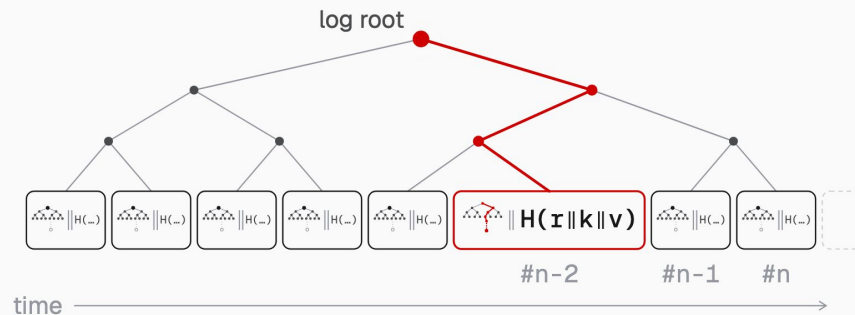
Merkle^2

A top level tree acts as the linear “log”, where leaves are log entries

The entries of the log are $K \rightarrow V$ mapping updates

Each log entry also comes with a prefix tree, which maps a search key to a struct:

`{firstIndex, updateCount}`



Search process

1. Look up the search key in the latest prefix tree
 - a. Returns a (`firstIndex`, `updateCount`) pair
2. Binary search in the linear stream of updates, starting with `firstIndex`, until you find the entry which transitions from (`updateCount-1`) to `updateCount`
3. Check all the inclusion proofs

Types of mappings

- Username → ACI
 - Phone number → ACI
 - ACI → Public Key
-
- Username lookups not currently supported for contacts

Fake entries

- We would like to avoid publicly leaking information about update timing
- The log periodically generates “fake” updates which act as normal entries
- The Auditor knows which entries are real vs fake

Why Auditing

Auditors ensure the log is:

- **Well formed:** Alice wants to check that no unknown keys are present – version numbers must be monotonic
- **Consistent:** Alice and Bob need to see the same log view

How it works

- Auditors stream log entries from the KT server
- An auditor maintains enough state to append new leaves to the Merkle tree
- No Merkle proofs needed from the KT server - the auditor just builds the tree themselves
- Periodically, the auditor signs a tree head and sends it to the server
 - We do this every 5 minutes
- Clients check lookup consistency against a recent auditor-signed checkpoint, no more than 7 days in the past

Alternatives

- Contact monitoring
 - When Bob looks up Alice's key, he continues to monitor until the next distinguished log entry
 - Prevents hiding key versions, assuming Bob stays online
- Gossip
 - Users gossip STHs in existing E2EE pipes
 - Prevents split views, probabilistically
- TEEs / hardware devices

Privacy Implications

- Auditors do not see:
 - Phone numbers
 - Usernames
 - Account Identifiers
 - Lookup behavior
- Auditors Do See
 - Per-account key update timing
 - New registration volume

Auditor Burden

- One GCP e2-small
- A few KB of object storage
- 7-days of leeway for downtime
- 90 days notice for opt-out or key rotation

We've been operating for 1 year:

- 2 incidents
- ~15 hours of maintenance effort

<https://github.com/trailofbits/signal-auditor>

Finding an Auditor

- Reliable
- Technically skilled
- Independent
- Trusted

As we work to achieve operational maturity and resolve bugs at this tremendous scale, we will be launching Key Transparency with an internal auditing service across our production KT trees. We plan to share more details about our public auditing strategy in 2024.

- iMessage KT

Finding an Auditor

- <https://witness-network.org/>
 - If you can use the witness protocol, you should
- Otherwise:



What is a “Trail of Bits”?



Home



NEW

Patch the Planet



Explore



Services



Company



Open Source

Security work that outlasts the engagement

Every engagement leaves something behind on purpose: open-source tools, published research, and defenses that keep working long after the work is done. We are here to raise the security bar for the whole field, not just the client in front of us.



What is a “Trail of Bits”?

- Founded in 2012
- ~140 people
- Privately owned, for profit - no external shareholders
- ~70/30 billable / research+OSS split
- <https://blog.trailofbits.com/>

Independence

- We volunteered to run Signal's KT stack for free
 - Talk to me if you have OSS systems
- For paying customers, we have NDAs
- But, we own all content and have editorial freedom
- Clients can prevent publication, but cannot edit publications
- This makes a canary model compelling
- No single client is so large that we could not lose them

Don't *only* trust us

- Security auditors are a compelling source of trusted third parties for many types of system
- There are many firms which are trusted and qualified to help - don't just pick your favorite and assume they'll be around forever
- Involve the public where you can

What does all of this buy us

Does this affect my privacy?

No. Your messages remain end-to-end encrypted.

This feature only helps confirm that you and your contact are using the correct encryption keys. A missing or changing verification status does not indicate a compromised chat.

Reminder: It [does not verify the real-world identity](#) of the person using that account. Someone can still pretend to be another person, brand, or service.

What does all of this buy us

- Split views, etc can be maintained for up to 7 days
- In general, KT security is retrospective - so we need to be able to tell when it fails
- Who does the client call when KT verification fails?
- How do we distinguish between a benign failure and a malicious one?
- In CT, we have ct-policy@chromium.org
- But iMessage, WhatsApp, Signal KT failures just become bug reports

TEE Transparency

The hot new thing is TEEs

- Apple PCC
- WhatsApp Private Processing
- Google Private AI Compute
- Amazon Nitro + Bee
- Meta Confidential Muse
- [Confer.to](#), [Tinfoil.sh](#), [Privatemode.ai](#)

TEE Transparency

- Many of these systems have closed-source components
- Some don't even release the binaries, just hashes
- For many, the public can't audit append-only behavior of the log

Right now, the best way to see what these systems are doing is to read our reports

<https://github.com/trailofbits/publications>

Retrospective Security

- What happens when a binary is found to have a vulnerability?
- What if operational failures require resetting the log?
- There is no purely technical solution - in the end, it's trust

Thank you!



- Katherine Yen - Building and deploying the Signal KT system
 - <https://signal.org/blog/automatic-key-verification/>
 - Brendan McMillion - Standardizing KT in the IETF
 - <https://datatracker.ietf.org/doc/draft-ietf-keytrans-architecture/>
 - Cloudflare - <https://developers.cloudflare.com/key-transparency/>
-
- ToB Blog
 - <https://blog.trailofbits.com/2026/08/11/how-trail-of-bits-helps-verify-the-integrity-of-your-signal-chats>
 - Signal KT Server
 - <https://github.com/signalapp/key-transparency-server>
 - ToB Auditor
 - <https://github.com/trailofbits/signal-auditor>

Email me!
tjaden.hess@trailofbits.com