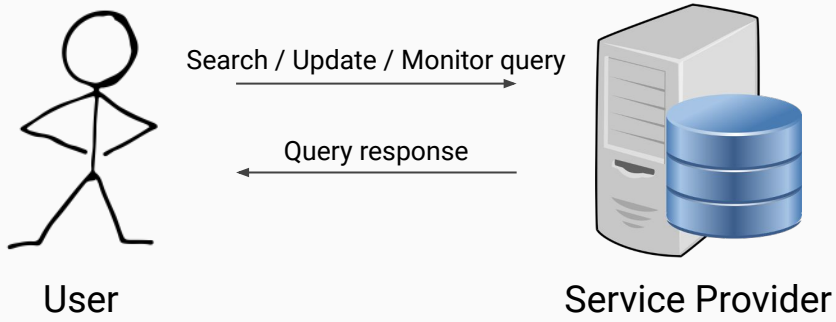
draft-mcmillion-key-transparency

Brendan McMillion
IETF 116 / March 29, 2023

Basic Model

1. **Search:** What's the value of this key?
2. **Update:** Here's a new value for this key!
3. **Monitor:** What's new with my keys?

- Looks like a key-value database
- Service Provider enforces access control rules by simply rejecting queries that aren't allowed
- User (generally) only needs direct communication with the service provider



Design Goals

Boring / non-controversial:

- Efficient verification processes and small state
- Avoid cryptographic algorithms that don't have a straightforward path to being post-quantum secure

Maybe interesting:

- New entries should be added to the log immediately

Interesting:

- **Can still be secure without third-party auditing**
- Metadata privacy (not addressed)

Important trade-off: Efficiency vs Third-party Assistance

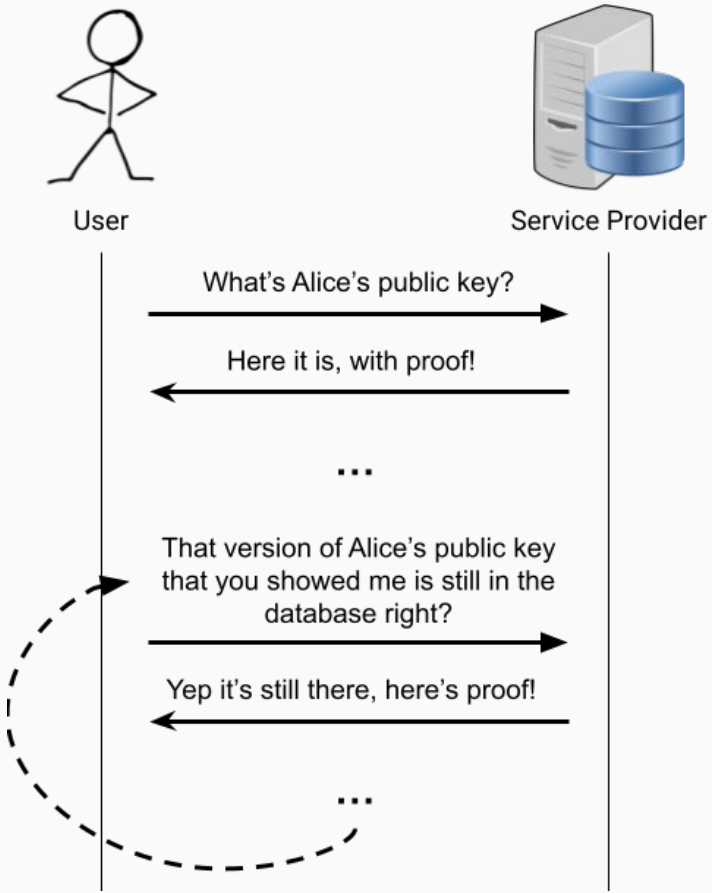
Intermission

Big idea: Take a KT construction that works for single-party deployments and then define ways to optimize it with a trusted third party, AKA:

Deployment Modes

1. Contact Monitoring
2. Third-party Auditing
3. Third-party Management

1. Contact Monitoring



2. Third-party Auditing

Many users

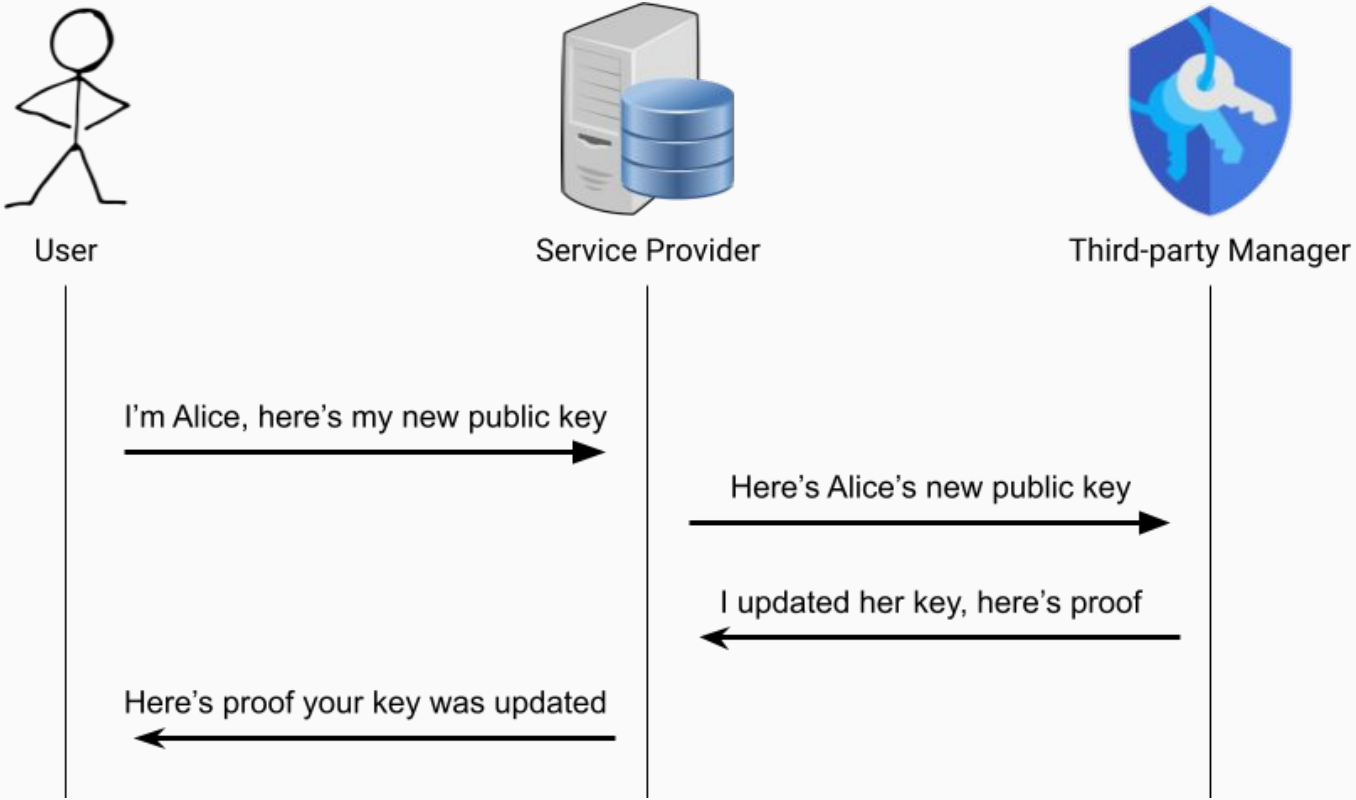
Service Provider

Third-party Auditor

Here's a list of all the changes I made to the database today!

Looks like you did everything right, here's a signature saying that

3. Third-party Management



Notably missing: anonymous third-party auditors

Some constructions allow public auditing:

- Service Provider exposes a public endpoint where anyone can download a log's content and check that basic invariants hold
- Similar to Certificate Transparency

Decided to omit from this proposal:

- Assumes out-of-band communication (trying to avoid)
- This type of endpoint tends to be expensive to support, and an easy target for abusers

Questions?
Thoughts?