# Concrete Key Consistency

Ben Schwartz, PRIVACYPASS @ IETF 116

*“Describing verification mechanisms for trusting Issuers and their corresponding keying material. Such mechanisms should prevent Issuers from presenting any key material that could be used to deanonymize clients.”*

# draft-ietf-privacypass-key-consistency

- Gives us the landscape of Key Consistency protocol architectures
- Enumerates a few notable architecture-categories
  - “Direct Discovery”
  - “Trusted Proxy Discovery”
  - “Shared Proxy with Key Confirmation”
  - “Multi-Proxy Discovery”
  - “Database Discovery”
- Does not describe implementation details of these architectures
- All architectures depend on some additional parties that are assumed to be non-malicious.
  - Differences include how many parties are involved, how much they are trusted, and which way information flows between them.

# Private State Tokens (f.k.a. “Trust Tokens”)

W3C WICG proposal for exposing Privacy Pass to the Web

PST assumes that we are going to solve this problem:

> *“...the Privacy Pass protocol should ensure that issuers publish a public key commitment list, verifies it is small (e.g. max 3 keys), and verifies consistency between issuance and redemption.”*

but we have not!

# What happens if we don’t solve it?

Section 4 of Private State Tokens contains a sort of “stopgap” solution:

> *“It is recommended that each user agent fetches the key commitments from issuers at a regular cadence and through trusted infrastructure, and then sends the concatenated map of issuers and key commitments to the client to ensure consistency between the keys different user agent instances use.”*

-> Only Issuers selected by the User Agent are allowed, and every Issuer’s keys must be broadcast to, and stored by, every client -> **Limited number of Issuers.**

-> Every User Agent must operate its own* high-reliability key distribution service and curate a list of included Issuers -> **High barrier to participation for User Agents.**

**Strongly consolidating for both Issuers and User Agents**

# Is there a hint about how to solve this?

> *“Section 10.2.1. Potential Attack: Side Channel Fingerprinting*
>
> *Unlinkability is lost if the issuer is able to use network-level fingerprinting … and can associate the user agent at redemption time with the user agent at token issuance time”*

In other words, we have to assume the availability of a network proxy service – a third party whom the client trusts to protect its anonymity.

# Possible Consistency Protocol Design Goals

- Allow User Agents to provide network unlinkability and key consistency for Privacy Pass without having to design or operate their own services.
  - Requires defining a concrete Key Consistency protocol.
- Allow Clients to use any and all Issuers, without requiring the User Agent to know about the Issuer in advance.
  - Excludes most broadcast architectures.
- Allow Issuers to serve any and all User Agents, without requiring the Issuer to know about the User Agent in advance.
  - Excludes many “Database Discovery” architectures.
- Avoid adding trust assumptions that are not already needed for network unlinkability.
  - Excludes the “Trusted Proxy” architecture.
- [YOUR GOAL HERE]

# Paths Forward

- Ignore the problem
- Solve the problem later
- Solve the problem now
  - draft-schwartz-ohai-consistency-doublecheck
  - Design Team
  - Free-for-all
  - ???