Key Consistency and Discovery

draft-wood-key-consistency
Motivation

Background

Emerging privacy-focused protocols require a mechanism for clients to discover server public keys

- Privacy Pass: Issuer verification key
- ODoH and OHTTP: Target public encryption key
- Tor: Relay public keys

Common requirements:

1. Unlinkability: Servers cannot link usage of a key to specific users
2. Authenticity: Clients use an authentic key for the intended server
Motivation

Unlinkability
Motivation
Unlinkability

Sent from one of $n$ clients...
Motivation

Unlinkability

\[ C_1 \rightarrow S \]

\[ K_1 \]

\[ C_2 \]

\[ C_n \]

\[ K_n \]
Motivation
Unlinkability

$C_1$ $K_1$ $f(K_1)$ $S$

Sent from $C_1$!
Motivation
Authenticity

$C_1 \rightarrow C_2 \rightarrow C_n \rightarrow \text{Adv} \rightarrow S$

$K'$

$K$
Motivation

Authenticity

$C_1 \rightarrow K' \rightarrow f(K') \rightarrow Adv \rightarrow f(K') \rightarrow Adv\rightarrow f(K) \rightarrow S$

Adv learns something meant for $S$
Unlinkability and authenticity means that all clients in the same anonymity set have a consistent view of the server’s intended key, and that view is correct.
Consistency and Correctness

In practice

A key consistency and correctness system (KCCS) is something that provides consistency and correctness for clients.

KCCS varies in practice based on:

- Threat model
- Cryptographic dependencies
- Trust model and PKI
- Operational complexity
- External dependencies
Consistency and Correctness

Design space

Vary the topology and trust model:

• Fetch through a trusted proxy
• Fetch through multiple less-trusted proxies
• Outsource to an audited or verified data store

Vary the cryptography:

• Classical signatures vs other primitives
Trusted Proxy Discovery

\[ C_1, C_2, \ldots, C_n \]

\[ P \rightarrow S \]

\[ K \]
Trusted Proxy Discovery

$C_i, i \in [1, \ldots, n]$ get the same $K$
Multi-Proxy Discovery

\[ C_1 \rightarrow C_i \rightarrow R_1 \]
\[ C_2 \rightarrow R_2 \]
\[ C_n \rightarrow R_m \]

\[ S \]
Multi-Proxy Discovery
Multi-Proxy Discovery

$C_1$  
$C_2$  
$C_n$

$C_i$

$R_1$

$R_2$

$R_m$

$S$

$C_i$ detects a problem!

$S$ tries to lie
External Database Discovery

$C_1$, $C_2$, ..., $C_n$ connected to $D$, $K$ to $S$.
External Database Discovery

Append-only audited log, nodes running consensus protocol, ...
Wrapping Up
Status and questions for the group

Currently *not meant to be published as an RFC*

Most schemes can be deployed without any new technology

Questions for the group:

• Is this useful?

• How can this document help build reliable key consistency solutions and protocols?