Privacy Pass Ecosystem

Drafts, Key Management, APIs, Implementations

IETF 107 – Virtual – 2020-03

Steven Valdez - svaldez@google.com
Privacy Pass Drafts
pp-protocol

- Protocol for Client/Server Issuance and Redemption
- Specification of cryptographic requirements
- Defines Extension Policy for the protocol
pp-architecture

- Interfaces that Client/Server should expose
- Interface for configuration/key management
- Privacy Considerations
  - User Segregation
  - Tracking/Identity Leakage
- Security Considerations
  - Key Rotation
  - Token Exhaustion
- Defines Extension Policy for the architecture
pp-http-api

- [GitHub Repository](https://github.com/alxdavids/privacy-pass-ietf/blob/master/draft-svaldez-pp-http-api.md)
- HTTP API Extension of pp-architecture
- HTTP Wrapping for Protocol Messages/Interfaces
- Key Management for HTTP Clients (Commitment Registry)
- Delegated Redemption
Key Management Options
Key Management Requirements

- Consistent Key Commitments across Clients
  - User Segregation
  - Issuance-Time Fingerprinting
- Key Rotation
  - Compromised Keys
  - Lost Keys
- Auditability
Option: Issuer Configuration

- Clients fetch latest Key Commitments directly from Issuer
  - Anonymous *separate* Connection (reduce the fingerprint an Issuer can get)
  - Fetched at issuance and redemption time to verify Issuer isn't quickly rotating keys
- Some form of "auditor" is required to defend against split-view.
- Auditing is only a partial mitigation.
Option: Proxied Configuration Fetching

- Proxy fetches key configurations.
- Clients fetch key configuration from proxy.
- Auditors verify proxy is seeing consistent views of key commitments.
- Proxy has no client state to segregate key configurations.
Option: Commitment Registry

- Append-Only Log
- Auditors verify issuers aren't rotating key commitments too frequently
  - Client Policy for rate of key rotation is allowable.
  - Client or Third-Party Auditors responsible for detecting violations.
  - Log may prevent additions to the log on violations.
- Clients fetch latest Key Commitments
  - Directly from Log
  - From Token Issuer (with some form of inclusion proof)
Open Questions

● Strategy for double-spending?
  ○ Eventually consistent
  ○ Global Registry

● Protocol for detecting/reporting malicious servers?
  ○ Client Policy
  ○ Gossip Protocol

● Acceptable key rotation windows?
  ○ Key Compromise vs First Party State

● Recommended key management strategy?
  ○ Proxied Fetching vs Append-only Log

● Balancing number of issuers versus privacy considerations?
  ○ Consolidation, Limiting Redemption vs Issuance
Current APIs/Implementations

- **Challenge Bypass Extension** ([https://github.com/privacypass/challenge-bypass-extension](https://github.com/privacypass/challenge-bypass-extension))
  - Cloudflare
  - Go/JS

  - Brave
  - Rust/JS

- **Trust Token** ([https://groups.google.com/a/chromium.org/g/blink-dev/c/X9sF2uLe9rA/](https://groups.google.com/a/chromium.org/g/blink-dev/c/X9sF2uLe9rA/))
  - Chrome
  - C/JS
Privacy Pass Ecosystem

Drafts, Key Management, APIs, Implementations

IETF 107 – Virtual – 2020-03

Steven Valdez - svaldez@google.com