Privacy Pass Extensions

Scott Hendrickson, Christopher Wood, Ghous Amjad, Kevin Yeo, Steven Valdez

IETF 120 - Privacy Pass
Two drafts adopted:

- [draft-ietf-privacypass-auth-scheme-extensions](#)
  - Adds an `extensions` parameter of PrivateToken:
    Authorization: PrivateToken token="abc..." extensions="def..."

- [draft-ietf-privacypass-public-metadata-issuance](#)
  - Defines privately (POPRF) and publicly (PBRSA) verifiable public metadata mechanisms
Status

RSA Signatures with Public Metadata

- Chrome IP Protection Authentication (including Expiration Extension)
- Pixel VPN Authentication
- Third-party audit of Cryptography
Extension Flow

+--------+                                  +--------+
| Client |                                  | Origin |
+---+----+                                  +---+----+
|                                           |
|------ Request --------------------------->+  (RFC 9577)
+<--- TokenChallenge{token_type: 0xDA7A} ---|
|                                           |
|-- Request(Authorization: PrivateToken     |
|  token="abc...", extensions="def..."!)--->|
|                                           |
|                                           |
- **token_type** defines cryptographic protocol used
  - standardized in draft-ietf-privacypass-public-metadata-issuance
- **extensions="..."**
  - content is not yet standardized (example: Expiration Extension)
  - negotiation is not yet standardized
    - consider a TokenChallenge extension?
  - see auth-scheme-extensions sec 4 discussion
Proposal: Extension Negotiation

- Image starts from draft-ietf-privacypass-auth-scheme-extensions
- adds a set of extensions requested by origin
- client can choose to present a subset
- reminder: the extensions presented add more entropy
Path to WGLC

- **Partially Blind RSA** adoption
  - Presenting again at CFRG (Thursday session)
  - Prior adoption call requested privacypass draft usage
- Negotiate extensions, or not (previous slide)

+ Bonus: Adopt **Expiration Extension** in Privacy Pass?
Thank you!

IETF 120 - Privacy Pass
ExpirationTimestamp

- Expiration and key rotation are interchangeable
- Key rotation is difficult to perform quickly
- draft spec

```c
struct ExpirationTimestamp {
    uint64_t timestamp_precision;
    uint64_t timestamp;
}
```

all readers should verify safe rounding