Trust Token

IETF 108 – Virtual – 2020-07

Steven Valdez - svaldez@google.com

Outline

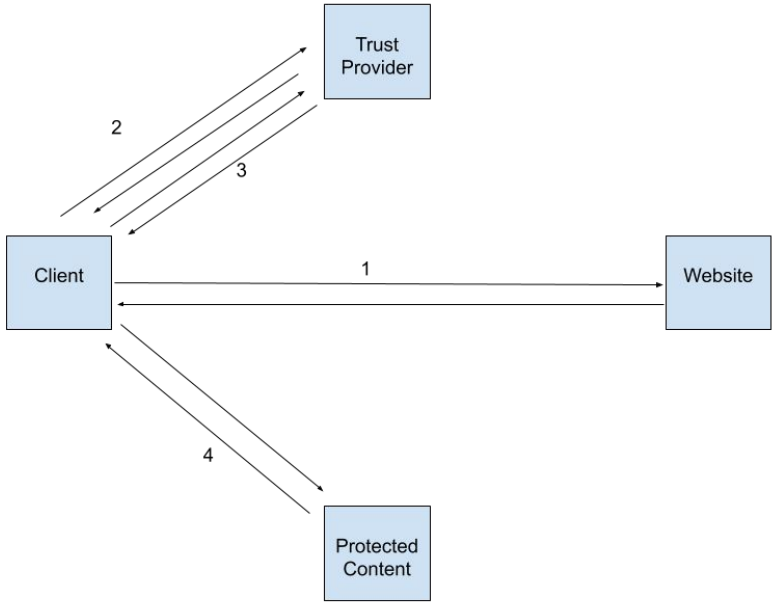
- Problem
- Old Way
- New Way (Privacy Pass)
- Trust Token

Problem

- Protection against:
  - (D)DoS protection
  - Bots
  - Spam
- Avoid pain for legitimate users.
- Avoid relying on cross-site tracking/fingerprinting.

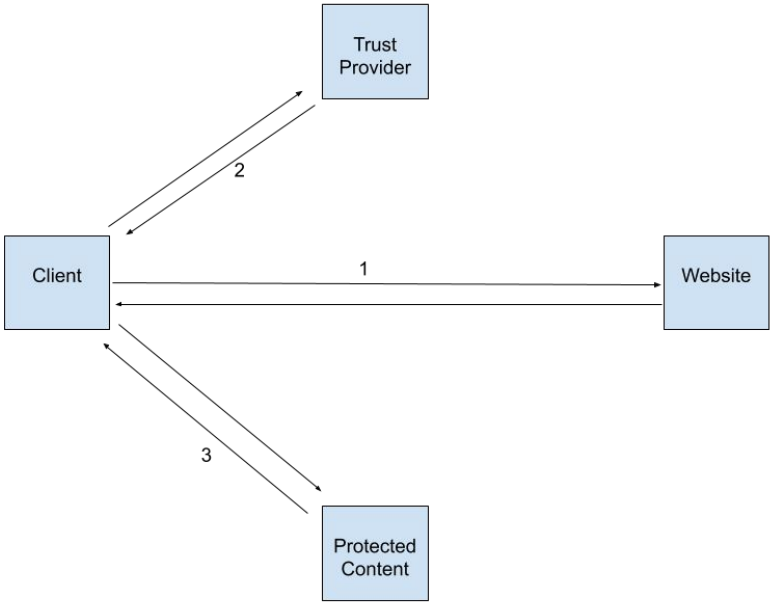
Old Way (1)

1. Request content from website, get asked to complete a CAPTCHA.
2. Ask the Trust Provider for a CAPTCHA challenge.
3. Send the response and receive a challenge verification and some state the client keeps.
4. Send the challenge verification with the request to the Protected content.



Old Way (2)

1. Request content from website, get asked to complete a CAPTCHA.
2. Ask the Trust Provider for a CAPTCHA challenge including your **3P state**. The Trust Provider skips (or simplifies) the challenge/response and returns a challenge verification.
3. Send the challenge verification with the request to the Protected content.



New Way

- Instead of generalized **3P State**, issue some sort of pass/token that only contains the exact information needed to carry the trust attestation from the Trust Provider (Issuer) and can later be redeemed.
- A raw signature allows an Issuer to track the client between issuance and redemption which introduces another cross-site tracking vector.

Token Properties

- Unforgeable - Client can't make more tokens.
- Non-malleable - Client can't alter the token.
- Unlinkable - Issuer can't correlate an issuance of a token with its redemption.
- Efficient - Can be used at scale.
- Verifiable - The token can only contain the amount of information allowed by the client.

Privacy Pass with VOPRFs

- Unforgeable
- Non-malleable
- Unlinkable
- Efficient
- Verifiable

Trustworthy-ish Signal

- Trust Provider **3P State** and challenge verification can represent a spectrum of trustworthiness from trustworthy to untrustworthy.
- Allows a Trust Provider to propagate trust/distrust without immediate feedback to the client.
- Prevent reverse engineering of the bot detection algorithms via instant feedback of whether the issuer believes you are trustworthy (if they issued a token).

New Token Property

- Private Metadata - A limited amount (1 bit) of information about the issuance that isn't visible to the client, but is provably limited to only the specified amount of information

Attempt 1: Two Keys and a DLEQOR

- PrivacyPass effectively uses a DLEQ proof to prove that a token was signed with a specific key.
- Instead, Issuer uses one of two keys to sign the Privacy Pass token and sends a DLEQOR proof to prove it used one of those two keys.
- Attacks where upon redemption, the validity of the token (whether the issuer accepts the signature) allows you to compare whether two tokens were signed with the same or different keys.

Attempt 2: PMBTokens

- Issuer Key consists of:
  - KeyA/KeyA - Key to sign 'A' or 'B' tokens.
- On issuance, the Issuer signs the token as:
  - Sig(T, KeyS) + DLEQ - A signature using the validity signing key and a proof showing that the token was signed with that key.
  - Sig(T, KeyN) + DLEQOR - A signature using either KeyA or KeyB (based on whether the issuer sets the private metadata to be A or B, and a proof showing one of those keys was used.
- On redemption, the Issuer verifies the validity signature first, and if it succeeds, proceeds to use which key the private metadata was signed with to determine the private metadata value.

Trust Token

Underlying Crypto Protocol

- PMBTokens Crypto Scheme
- Uses P-384
  - Concerns about application of Cheon/Brown-Gallant attacks (Diffie-Hellman Oracle)
- Multiple Tokens in a batch
- DLEQ(OR) Batching

Redemption Records

- Sites with many embedded resources that need some trust attestation:
  - Multiple comment boards
  - Advertisements
  - Social Media buttons
  - Heavy resources
  - …
- Client redeems a Trust Token and receives a redemption record valid for the current time and top-level website, and can send that redemption record along with all the resource requests.
- Allows downstream consumers of Trust Token without requiring them to have to handle the QPS of redeeming a token against the Trust Provider for every request.

Key Management

- Each key that an Issuer uses can divide the anonymity set of Trust Token users.
- Avoid issuers from providing per-user/region keysets.
- Proxied Configuration Fetching
  - Proxy fetches all the issuer key commitments and then sends them to clients.
- Extensions/Alternatives
  - Public append-only log (similar to CT) for key commitments
  - Auditing parties that verify issuers aren't changing their keys too frequently.
  - Key rotation policies/restrictions.

Trust Token (Δ from Privacy Pass)

- PMBTokens
  - Working with authors to bring crypto primitive to IETF
  - Privacy Pass work to support other underlying crypto primitives.
- Redemption Records
  - Potentially useful for wider Privacy Pass use cases.
- Key Management
  - Depending on the ecosystem, moving to an append-only log/commitment registry.

Next Steps

- Privacy Pass IETF Standardization
  - First WG Session: Friday Session III
- Experiments
  - Verify value of Trust Token signal.
  - Verify ergonomics of Trust Token API.
- Trust Token Ecosystem
- Web API W3C Standardization
  - Privacy Pass and/or Trust Token

Links

- Privacy Pass (https://datatracker.ietf.org/wg/privacypass/about/)
- PMBTokens (https://eprint.iacr.org/2020/072)
- Trust Token (https://github.com/WICG/trust-token-api)

(First Privacy Pass WG Session: Friday Session III)

Trust Token

IETF 108 – Virtual – 2020-07

(First Privacy Pass WG Session: Friday Session III)

Steven Valdez - svaldez@google.com