Adding public metadata to Privacy Pass

Sofía Celi

https://github.com/ietf-wg-privacypass/base-drafts/pull/78

- Proposal: **A Fast and Simple Partially Oblivious PRF, with Applications** [TCRSTW21] (https://eprint.iacr.org/2021/864.pdf) by Tyagi, Celi, Ristenpart, Sullivan, Tessaro and Wood
- Other proposals:
    - In some applications, it is needed to add metadata (public, private) to (V)OPRFs:
        - Private metadata bit [KLOR2020] (https://eprint.iacr.org/2020/072.pdf)
        - Attribute-based VOPRFS [Facebook21] (https://research.fb.com/wp-content/uploads/2021/04/DIT-De-Identified-Authenticated-Telemetry-at-Scale_final.pdf)

Why should we add metadata?

- Useful against certain attacks:
  - *Hoarding attacks*: individual users (or groups of users) gather tokens over a long period of time and redeem them all at once in an attempt to overwhelm a service
  - Possible metadata values to prevent it:
    - Epoch (linked to key rotation)
    - Geographic localization

Underlying primitive used

- VOPRF with PO-PRF:

$$y = F(x, sk) \rightarrow \mathbf{y = F(x, t, sk)}$$

x: private client input

sk: private server key

t: public client or server metadata

y: output of the PRF function

From a Privacy Pass perspective (Issuance):

```
Client(pkS, m, info, clientMetadata, serverMetadata)      Server(skS, pkS, clientMetadata, serverMetadata)
-----------------------------------------------------------------------------------------------------------

commit_req = Prepare(info)

                                      commit_req
                                   ------------------>

                                       commit_resp = Commit(skS, pkS, commit_req)

                                      commit_resp
                                   <------------------

cInput = Generate(m, commit_resp)
req = cInput.req

                                        req
                                   ------------------>

                                        issueResp = Issue(pkS, skS, req, serverMetadata, clientMetadata)

                                     serverResp
                                   <------------------

tokens = Process(pkS, cInput, serverResp, serverMetadata, clientMetadata)
store[server.id].push(tokens)
```

From a Privacy Pass perspective (Redemption):

```
Client(info, clientMetadata, serverMetadata)            Server(skS, pkS, serverMetadata, clientMetadata)
-----------------------------------------------------------------------------------------------------------
token = store[server.id].pop()
req = Redeem(token, info, serverMetadata, clientMetadata)

                                        req
                                  ------------------>

                                        if (dsIdx.includes(req.data)) {
                                          raise ERR_DOUBLE_SPEND
                                        }
                                        resp = Verify(pkS, skS, req, serverMetadata, clientMetadata)
                                        if (resp.success) {
                                          dsIdx.push(req.data)
                                        }

                                         resp
                                  <------------------
Output resp
```

Notes on metadata

- Every metadata diminishes the anonymity set
- PO-PRF does not bound the metadata length, but applications using metadata should bound it to balance privacy and utility

https://github.com/cfrg/draft-irtf-cfrg-voprf/pull/258

https://github.com/ietf-wg-privacypass/base-drafts/pull/78

Thank you!

@claucece

https://github.com/ietf-wg-privacypass/base-drafts/pull/78
https://eprint.iacr.org/2021/864