background

CFRG interim meeting
Oblivious Pseudorandom Functions (OPRF)

PRF(k, x)

unblind

PRF(k, x)

blinded
Verifiable OPRF (VOPRF)

\[ \text{C} \xrightarrow{\text{PRF}(k,x)} \text{S} \]

Verify proof && unblind

\[ \text{PRF}(k,x) \xrightarrow{\text{proof}(k)} \]

blinded
Prime-order groups

$x, \, pk=kG$

$P = r \cdot H_2G(x)$

$Q = k \cdot P$

$H(x, (1/r) \cdot Q)$

proof $(DL(pk) == DL(Q))$

$k, \, pk=kG$

verify proof

optional (VOPRF)
Related applications

Privacy Pass: datatracker.ietf.org/wg/privacypass/

OPAQUE: draft-krawczyk-cfrg-opaque

Trust Token API: github.com/WICG/trust-token-api
draft-04 updates

CFRG interim meeting
Major API changes

Client & Server global contexts [Section 4.4]

- modeVerifiable
- function instantiations:
  - Server: KeyGen, Evaluate, VerifyFinalize
  - Client: Blind, Unblind, Finalize

Remove batching as standard

- Present as an efficiency improvement
Ciphersuites

Including support for 128-bit security ciphersuites [Section 5]:

- P-256
- curve25519

Consolidate all hash functions into SHA-512
Ciphersuites

All supported ciphersuites:

- OPRF(curve25519, SHA-512) [Section 5.1]
- OPRF(curve448, SHA-512) [Section 5.2]
- OPRF(P-256, SHA-512) [Section 5.3]
- OPRF(P-384, SHA-512) [Section 5.4]
- OPRF(P-521, SHA-512) [Section 5.5]

Hoping to include ristretto/decaf support in the future
Prime-order group API

Concrete API for instantiating prime-order group [Section 3.1]

Group elements no longer exposed in public API

Instructions for implementing prime-order groups for all ECC ciphersuites [Section 5]
Other updates

Proof-of-concept sage implementation

Improved DST usage to remove HMAC/HKDF for ROM

Updates for latest hash-to-curve draft
concluding remarks
TODOs

Update PoC implementations in go, rust, BoringSSL (github.com/alxdavids/voprf-poc)

ristretto/decaf ciphersuites?

Test vectors in draft
Questions?

We think API should remain stable now

Thoughts on state of document?

Plan for RGLC?