Oblivious Pseudorandom Functions (OPRFs) using Prime-Order Groups

Alex Davidson, Armando Faz Hernández, Nick Sullivan, Christopher Wood
draft-irtf-cfrg-voprf@ietf.org

IETF - 111 - CFRG

https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-voprf

OPRF: Oblivious Pseudorandom Function

Two-party 1-RRT protocol between a server and a client

Client holds some input x

Server holds a private key k

y = PRF(k, x)

Oblivious

Client learns y,
but nothing about k.

Server does not learn
anything about x or y.

Verifiable

Client verifies proof that
PRF was computed with k.

Server commits to the
key k, and gives a proof.

What is new?

Blinding mechanism:
- Verifiable mode uses Additive blinding (secure as the key is committed)
- Base mode uses Multiplicative blinding

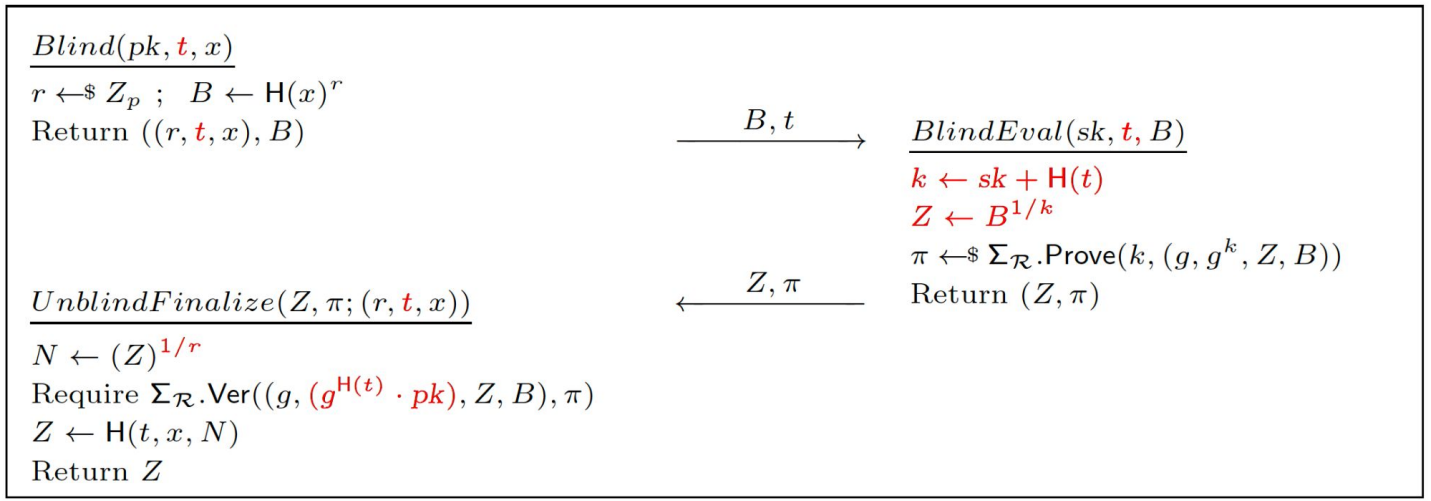
Explicit errors:
- Deserialization of objects
- Documentation and API considerations

Adopt SHAKE-256 for decaf448 ciphersuite

Generalize DLEQ proof functionality
- Use domain separation tags for its use in other protocols

Update test vectors and editorial changes

Adding public metadata proposal



Blind evaluation for our 3H PO-PRF construction. All three algorithms have implicit input the parameters $pp = (p, g, \mathbb{G})$ that describe the group used. The NIZK uses relation $\mathcal{R} = \{(g, U, V, W), (\alpha) : U = g^\alpha \wedge W = V^\alpha\}$.

From: Partially oblivious PRFs (PO-PRFs) [TCRSTW21] (https://eprint.iacr.org/2021/864)

Protocol:

Client	Server
`ctx ⇇ SetupClient(suiteId)`	`ctx,sk ⇇ ServerSetup(suiteId)`
`r, blinded ⇇ Blind(input)`	
	`eval ⇇ Evaluate(sk,blinded, serverMetadata, clientMetadata)`
`unblinded ⇇ Unblind(r, eval)`	
`output ⇇ Finalize(input, Unblinded, serverMetadata, clientMetadata)`	

Considerations

- Metadata can be of any length
- Metadata can be added by server and client
- Does not work with additive blinding
- Works with well with batching

Questions?

Sources: github.com/cfrg/draft-irtf-cfrg-voprf

Data Tracker: datatracker.ietf.org/doc/draft-irtf-cfrg-voprf

Issues: github.com/cfrg/draft-irtf-cfrg-voprf/issues