



TACEO

OPRFs in the Wild

A Survey of Real-World Deployments

Daniel Escudero, Lena Heimberger, Daniel Kales, Christian Rechberger, Verena Schröppel, Roman Walch

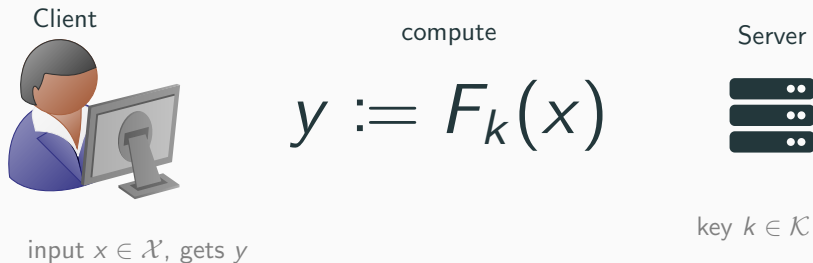
RWMPC 2026, Taipei

7th of March, 2026

Have you ever wanted to...

- ... send a hash over a wire?
- ... intersect a set privately?
- ... generate unlinkable tokens?
- ... generate MPC-friendly deterministic identifiers?
- ... take your low-entropy input and get a high-entropy output?

You may want to consider an OPRF!



Pseudorandom Functions [GGM84, GGM86]

- deterministic and polynomial-time function
- hard to distinguish output y from a randomly chosen element from \mathcal{Y} .
- relatively weak guarantees!

Simple PRF over elliptic curves

Sample k and compute PRF as $y = \text{hash_to_curve}(x) * k$
Remove algebraic structure with $H(y)$

Blind-evaluate-unblind: 2HashDH

Client(x)

Sample r

$$y = r^{-1} * c$$

$$= r^{-1} * k * b$$

$$= \text{hash_to_group}(x) * k$$

output $H(y)$

Server (k)

$$b := \text{hash_to_group}(x) * r$$

$$c := k * b$$

What about other constructions?



A word cloud of cryptographic terms is displayed on a computer monitor graphic. The terms are arranged in various sizes and orientations, with some being significantly larger than others. The colors of the text include yellow, blue, red, and black. The largest word is 'bhashhsahi' in blue, oriented diagonally. Other prominent words include 'quadratic-residue', 'batching', 'preprocessing', 'lattice-based', 'lattices', 'naor-reingold', 'rounded-subset-product', 'lwr', 'isogenies', 'legendre-symbol', and '2-hashdh'.

quadratic-residue batching
preprocessing
lattice-based
lattices
bhashhsahi
naor-reingold
rounded-subset-product
lwr
isogenies
legendre-symbol
2-hashdh

What about other constructions?

quadratic-residue batching

**In practice, only
2HashDH is deployed!***

naor rounded legendre-symbol
2-hashdh

*as far as we know

- OPRFs presented at NIST's 2023 Multi-party Threshold Scheme workshop
- IETF: only completed OPRF standard

OPRF 2HashDH

VOPRF $\text{OPRF} + \pi(F_k(x), k)$

POPRF Partial OPRF, the client submits a tag t in addition to the input x and gets the evaluation $y := F_k(t, x)$. t contains additional information, e.g. a validity period.

Could I use a blind signature instead of a VOPRF?

Protocol phase	action	blind RSA	VOPRF
Issuance	Client Blind	63 μ s 256 bytes	54 μ s 32 bytes
	Server Evaluate	2.69 ms 256 bytes	260 μ s 96 bytes
	Client Finalize	37 μ s 256 bytes	376 μ s 64 bytes
Redemption	Client	- 300 bytes	- 96 bytes
	Server	37 μ s -	57 μ s -

Table 1: Comparing runtimes between blind signatures and VOPRFs, Source: <https://blog.cloudflare.com/private-rate-limiting/#how-much-does-this-all-cost>

Also, OPRFs are amortizable!

Overview of deployed primitives

Category	Name	Application type
unlinkability	CAPTCHAs	rate limiting
	Kagi	unlinkable search
	MoQ	authentication
protect low-entropy secrets	Whatsapp	authentication
PSI	Google CCC	set intersect.
	Microsoft CCC	set intersect.
	MIGP	set intersect.
	Ad-to-purchase	set intersect.
deterministic identifiers	Callisto	matching
	human.world	identifier
	TACEO:OPRF	identity
	MetaMask	identity

Getting Unlinkability

Application: PrivacyPass

Client

Server (k)

1. Gather:

Choose $\{x_1, \dots, x_{30}\}$

Receive $\{F_k(x_i)\}_{i \in [30]}$

OPRF



2. Redeem:

$x_i, F_k(x_i)$



- Reduces number of CAPTCHAs by factor 30 [CDFH⁺22]
- Authentication for Media over Quic (MoQ) [NJM25]
- Kagi [doca]: anonymous identity for unlinkable search



Verifying...



Deterministic high-entropy output from low-entropy secrets

Private Set Intersection (PSI)

OPRFs are useful for **unbalanced** sets, where one party, typically the server in client-server protocols, has a significantly larger set than the other party.

Client (x_1, \dots, x_m)

Server $(y_1, \dots, y_n), k$

$\{F_k(x_i)\}_{i \in [m]}$

VOPRF



$\{z_j = F_k(y_j)\}_{j \in [n]}$



$F_k(x_i) = z_j$. Otherwise $F_k(y_j)$ is pseudorandom.

VOPRF: per-client keys may cause segmentation when output is revealed

Applications using OPRFs for unbalanced set intersection

- Google advertisement view to purchase conversion [IKN⁺20]
- Google [TPY⁺19]/Microsoft CCC [LKLCM21]: check if credential appeared in breach
- Cloudflare Might I Get Pwned (MIGP) [VRW21]: checks if credential or *related* credential have been leaked

No schemes use verifiability.

POPRFs are not useful.

OPRFs for identification: OPAQUE and identifiers

OPAQUE: Authenticate without revealing password

Client (pwd)

Server (k)

1. Registration:

$k' = F_k(pwd)$

OPRF



Sample (pk, sk)

$pk, ct = Enc_{k'}(sk)$



Store (pk, ct)

2. Login:

ct



$k' = F_k(pwd)$

OPRF



$sk = Dec_{k'}(ct)$

AKE using sk



- IETF RFC 9807 [BKLR25]
- Whatsapp: avoid revealing backup PIN [wha21]

Identities and Wallet keys from low-entropy outputs in a Threshold setting

Threshold A TOPRF enables a valid evaluation from a m -out-of- n servers.

- Callisto [RQA⁺18] to detect serial perpetrators of sexual misconduct while preserving victim privacy
- identities from OPRF input:
 - MetaMask [docb]
 - TACEO OPRF [KW]
 - human.world [hn26]
- value MPC-friendliness!

MPC-friendly OPRFs

Generate ZKP for evaluation with correct key.

Stronger and more flexible than VOPRF:

- anyone can verify ZKP tying input and evaluation together
- can prove *without* revealing input (dlog equivalent cannot do this)

Optimizing for fewer constraints

- each constraint requires additional computation
- R1CS: each equality constraint has one multiplication and unlimited additions
 - use a cipher with low multiplicative depth
- Example: commit to the challenge with Poseidon2 instead of SHA2
 - 2 orders of magnitude fewer constraints
- BabyJubJub as a curve:
 - base field is the same as the scalar field as prove system
 - directly operate on the (x, y) coordinate representation

Recursive Verification

- Trivial threshold setting: verify m proofs per OPRF
- Recursive Verification: threshold verification can be aggregated into a single proof
- responses are gathered by an accumulator
- 2 challenge/response interactions
- no communication between the servers

More expensive to generate proof, easier verification!

Table 2: Compute cost microbenchmarks from TACEO deployment: query(**Q.**), answer (**A.**) and finalization (**Fin.**), as well as overall communication complexity(**comm.**). Publicly verifiable Groth16 proof is given in the **Prov.** column.

Setting	Client Q. [μs]	Server A. [μs]	Client Fin. [μs]	Rounds	Server Comm. [bytes]	Client Comm. [bytes]	Prov. [ms]
Single-server OPRF	128	90	99	2	32	32	-
Single-server VOPRF	128	468	974	2	32	96	32
TOPRF ($t = 3$)	128	90	105	2	32	32	-
TOPRF ($t = 5$)	128	468	105	2	32	32	-
TOPRF ($t = 30$)	128	468	108	2	32	32	-
TVOPRF ($t = 3$)	128	90	2730	2	32	96	55
TVOPRF ($t = 5$)	128	468	4480	2	32	96	97
TVOPRF ($t = 30$)	128	468	26358	2	32	96	712
Public TVOPRF (all t)	147	653	1175	4	192	192	32

Conclusion

- wide applications for lightweight primitives
- extensions give rise to a number of protocols
- MPC+ZK gives efficient OPRFs:
 - use zk-friendly curves
 - use zk-friendly hash functions
 - **constant** verification cost for the client

Thank you!



Name	Application type	no reused x	used curve	VOPRF	POPRF	TOPRF
CAPTCHAs	rate limiting	✓	☕, ☕, P-256, P-384, P-512	✓	✓	✗
MoQ	authentication	✓	☕, ☕, P-256, P-384, P-512	✓	✓	✗
Kagi	unlinkable search	✓	☕	✓	✗	✗
Whatsapp	authentication	✗	?	✗	✓	✗
Google CCC	set intersect.	✗	🔒	✗	✗	✗
Microsoft CCC	set intersect.	✗	?	✗	✗	✗
MIGP	set intersect.	✓ [†]	🔒	✗	✗	✗
Ad-to-purchase	set intersect.	✗	?	✗	✗	✗
Callisto	matching	✗	P-384, ☕	✓	✗	✗
human.world	identifier	✗	☕, 🔒	✓	✗	✓
TACEO:OPRF	identity	✓	☕	✓ [‡]	✗	✓
MetaMask	login	✗	P-256, P-384	✓	✗	✓

Table 3: Protocols using OPRFs in practice. All implementations use the 2HashDH OPRF construction. [†] Input x is not repeated, but related. [‡] Threshold proofs are aggregable into a single constant-size proof with efficient verification. ☕ Ristretto255, ☕ Decaf488, ☕ BabyJubJub, 🔒 Sekp256k1, 🔒' Sekp224r1.

What about PQ?

work	assumption	rounds	Comm. cost	security(C-S)	preprocessing	trusted setup	verifiable	available
ICRSTM22	3-Mash SMH (not PQ)	2	766 bits	malicious-malicious	-	YES	YES	YES
ADDS21	(R)LWE+SIS	2	2 MB	semihonest-semihonest	-	YES	NO	YES
ADDS21	(R)LWE+SIS	2	128 GB	malicious-malicious	-	YES	YES	NO
AG24	(R)LWE+SIS	2	222 kb'	malicious-malicious	316 KB	YES	YES	NO
DOT25	binary LWR	2	11.9 KB	semihonest-semihonest	1.5 MB	YES	NO	YES
ADDS23	mod(2,3)+lattices	2	10 KB	malicious-semihonest	2.5 MB	YES	NO	NO
ADDS23	mod(2,3)+lattices	2	160 KB	malicious-malicious	2.5 MB	YES	YES	NO
HKL+25	heuristic LWR	6	23 KB	semihonest-semihonest	793 KB	YES	NO	YES
ESTX24	1MLWE-RU-R+MLWE+SIS	2	159 kb'	malicious-malicious	20 KB	YES	YES	NO
APRR24	mod(2,3)	2	916 bits	malicious-semihonest	38 bits	YES	YES	NO
DGH+21	mod(2,3)	2	641 bits	semihonest-semihonest	1836 bits	NO	NO	NO
SHB23	Legendre PRF	3	13 KB	semihonest-semihonest	?	YES	YES	NO
KCM24	Legendre PRF	2	~144 KB	semihonest-semihonest	~432 KB	YES	YES	YES
BDFH24	Legendre PRF as 2HashDH	9	356 KB	malicious-malicious	392 KB	YES	YES	YES
YBHKR24	generalized power residue (Legendre) PRF	3	774 KB	malicious-semihonest	-	YES	YES	NO
YBHKR24	generalized power residue (Legendre) PRF	3	978 KB	malicious-malicious	-	YES	YES	NO
FOO23	AES+Garbled Circuits	2	6.79MB	semihonest-semihonest	-	YES	NO	YES
HKL524	Minicrypt	?	22 bytes'	malicious-malicious	-	YES	NO	NO
Basso23	Isogenies F_p^2	2	3.8 MB	malicious-malicious	-	NO	NO	NO
Basso23	Isogenies F_p^2	2	8.7 MB	malicious-malicious	-	NO	YES	NO
BKW20	Isogenies F_p + lattices	2	20.54 KB	semihonest-semihonest	-	NO	NO	NO

References i

-  Daniel Bourdrez, Hugo Krawczyk, Kevin Lewi, and Christopher A. Wood.
The OPAQUE Augmented Password-Authenticated Key Exchange (aPAKE) Protocol.
RFC 9807, July 2025.
-  S. Celi, A. Davidson, A. Faz-Hernandez, S. Valdes, and C. A. Wood.
The privacy pass protocol.
Internet Draft, April 2022.
-  Alex Davidson, Armando Faz-Hernandez, Nick Sullivan, and Christopher A. Wood.
Oblivious Pseudorandom Functions (OPRFs) Using Prime-Order Groups.
RFC 9497, December 2023.
-  Kagi docs.
How does privacy pass work?

References ii



MetaMask documentation.

How does social login with metamask work?

https:

[//support.metamask.io/configure/wallet/social-login/](https://support.metamask.io/configure/wallet/social-login/).



Oded Goldreich, Shafi Goldwasser, and Silvio Micali.

On the cryptographic applications of random functions.

pages 276–288, 1984.



Oded Goldreich, Shafi Goldwasser, and Silvio Micali.

How to construct random functions.

Journal of the ACM, 33(4):792–807, October 1986.






human network.

Reintroducing human network's voprf, 1 2026.

https:

[//human.tech/blog/reintroducing-human-network-s-voprf](https://human.tech/blog/reintroducing-human-network-s-voprf).

References iii

-  Mihaela Ion, Ben Kreuter, Ahmet Erhan Nergiz, Sarvar Patel, Shobhit Saxena, Karn Seth, Mariana Raykova, David Shanahan, and Moti Yung.
On deploying secure computing: Private intersection-sum-with-cardinality.
pages 370–389, 2020.
-  Daniel Kales and Roman Walch.
A nullifier protocol based on a verifiable, threshold oprf.
<https://github.com/TaceoLabs/oprf-service/blob/df4de8e938222f93cd9af3a7272a84df6807e273/docs/oprf.pdf>.
-  Kristin Lauter, Sreekanth Kannepalli, Kim Laine, and Radames Cruz Moreno.
Password monitor: Safeguarding passwords in microsoft edge, 1 2021.
<https://www.microsoft.com/en-us/research/blog/password-monitor-safeguarding-passwords-in-microsoft-edge/>.



Suhas Nandakumar, Cullen Fluffy Jennings, and Thibault Meunier. Privacy Pass Authentication for Media over QUIC (MoQ). Internet-Draft draft-ietf-moq-privacy-pass-auth-01, Internet Engineering Task Force, October 2025.
Work in Progress.



Anjana Rajan, Lucy Qin, David W. Archer, Dan Boneh, Tancrede Lepoint, and Mayank Varia.
Callisto: A cryptographic approach to detecting serial perpetrators of sexual misconduct.
In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, COMPASS '18, New York, NY, USA, 2018. Association for Computing Machinery.

References v

-  Kurt Thomas, Jennifer Pullman, Kevin Yeo, Ananth Raghunathan, Patrick Gage Kelley, Luca Invernizzi, Borbala Benko, Tadek Pietraszek, Sarvar Patel, Dan Boneh, and Elie Bursztein. Protecting accounts from credential stuffing with password breach alerting. pages 1556–1571, 2019.
-  Luke Valenta, Cefan Daniel Rubin, and Christopher Wood. Privacy-preserving compromised credential checking, 10 2021. <https://blog.cloudflare.com/privacy-preserving-compromised-credential-checking/>.
-  Security of end-to-end encrypted backups, 09 2021. Version 1, https://www.whatsapp.com/security/WhatsApp_Security_Encrypted_Backups_Whitepaper.pdf.