ISOGENIES AND MPC

a marriage of love

Robi Pedersen
DTU Copenhagen

THE isogeny club

25th March 2025

Part I - Distributed Key Generation

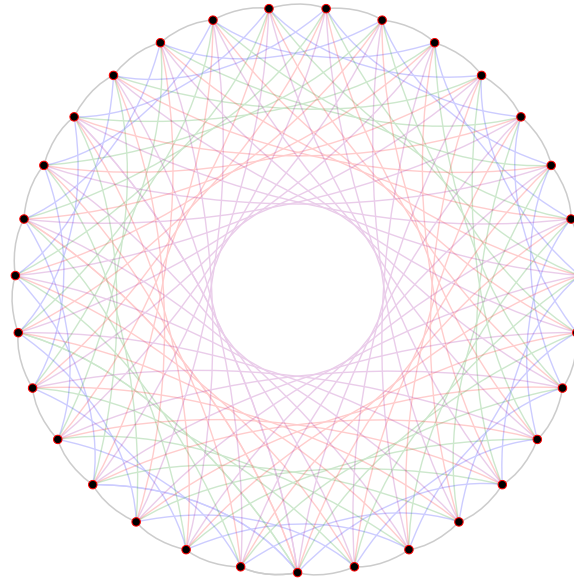
Part II - MPC Toolbox For Isogenies

Part III - Threshold Pseudorandom Protocols

THE

EFFECTIVE GROUP ACTION

SETTING



Effective group actions

commutative, additive, cyclic group

set

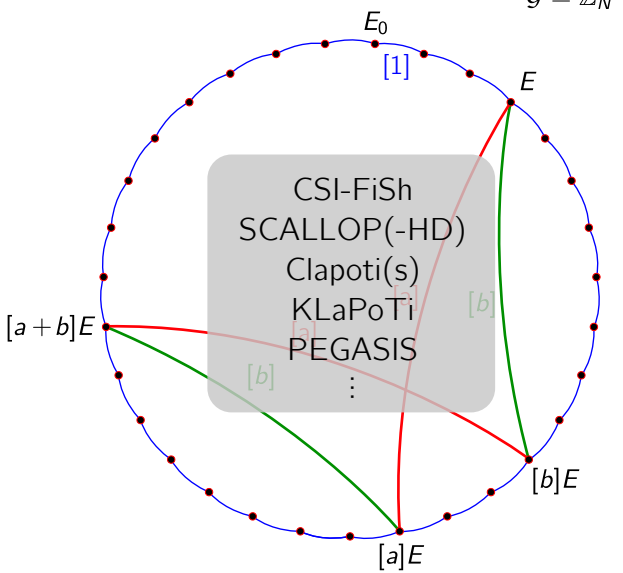
$$[\,] : (\mathcal{G}, +) \times \mathcal{E} \to \mathcal{E}$$

1. Efficient group operations in $\mathcal{G}$.
2. Efficient membership testing and canonical representation in $\mathcal{E}$, starting curve $E_0 \in \mathcal{E}$.
3. Efficient group action computation
$$(a, E) \mapsto [a]E.$$

Hard problems

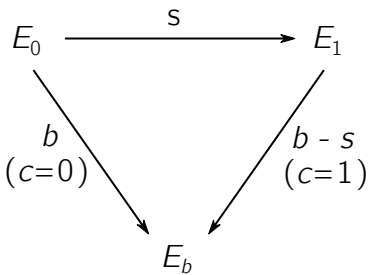
Vectorization: $(E, [a]E) \mapsto a$

Parallelization: $(E, [a]E, [b]E) \mapsto [a+b]E$



Basic ID-protocol

$\mathcal{R}_{ID} = \{(E_0, E_1), s \mid E_1 = [s]E_0\} \subset \mathcal{E}^2 \times \mathbb{Z}_N$



Commit	$b \leftarrow \mathbb{Z}_N$ $E_b = [b]E_0$
Challenge	$c \leftarrow \{0, 1\}$
Response	$r = b - cs$
Verification	$[r]E_c \stackrel{?}{=} E_b$

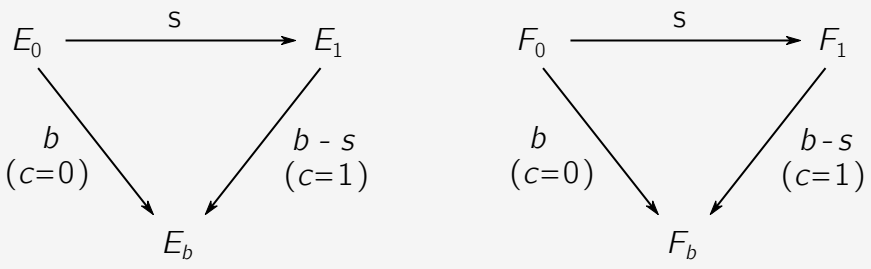
Hard problems

Vectorization: $(E, [a]E) \mapsto a$

Parallelization: $(E, [a]E, [b]E) \mapsto [a+b]E$

AND-proof

$$\mathcal{R}_{AND} = \{(E_0, E_1, F_0, F_1), s \mid E_1 = [s]E_0 \wedge F_1 = [s]F_0\} \subset \mathcal{E}^4 \times \mathbb{Z}_N$$



Hard problems

Vectorization: $(E, [a]E) \mapsto a$

Parallelization: $(E, [a]E, [b]E) \mapsto [a+b]E$

Commit

$b \leftarrow \mathbb{Z}_N$

$E_b = [b]E_0$

$F_b = [b]F_0$

Challenge

$c \leftarrow \{0, 1\}$

Response

$r = b - cs$

Verification

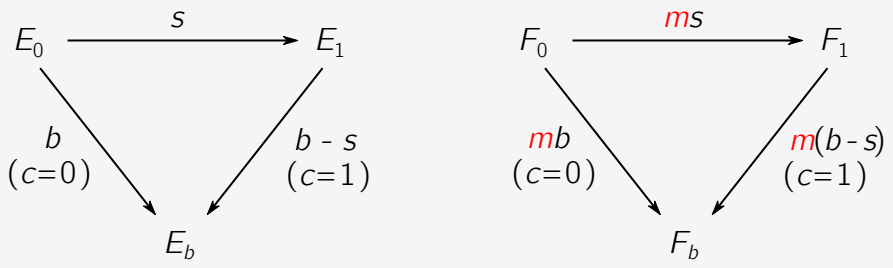
$[r]E_c \stackrel{?}{=} E_b$

$\wedge\ [r]F_c \stackrel{?}{=} F_b$

D. Cozzo and N. Smart. Sashimi: Cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol

SCAL-proof

$$\mathcal{R}_{SCAL} = \{(E_0, E_1, F_0, F_1, m), s \mid E_1 = [s]E_0 \wedge F_1 = [ms]F_0\} \subset \mathcal{E}^4 \times \mathbb{Z}_N \times \mathbb{Z}_N$$



Hard problems

Vectorization: $(E, [a]E) \mapsto a$

Parallelization: $(E, [a]E, [b]E) \mapsto [a+b]E$

Commit

$b \leftarrow \mathbb{Z}_N$

$E_b = [b]E_0$

$F_b = [mb]F_0$

Challenge

$c \leftarrow \{0, 1\}$

Response

$r = b - cs$

Verification

$[r]E_c \stackrel{?}{=} E_b \wedge$

$[mr]F_c \stackrel{?}{=} F_b$

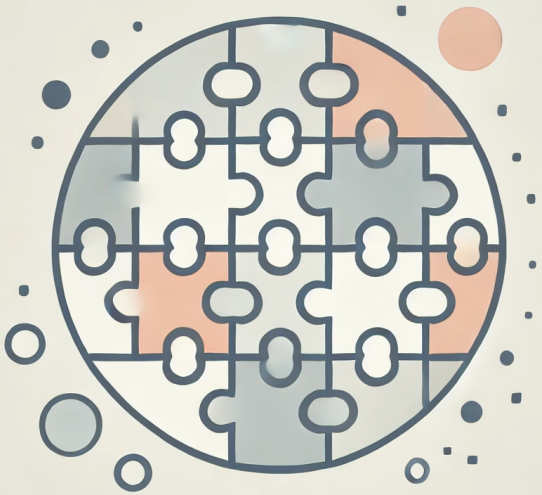
K. Baghery, D. Cozzo, R. Pedersen (2021). An Isogeny-Based ID Protocol Using Structured Public Keys.

Has this ever happened to you???

Protect against theft and loss now!!!

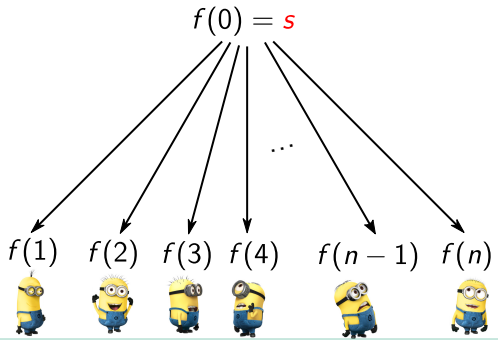
Distributed Schemes

for Isogenies



Shamir secret sharing

$$f(X) = s + f_1 X + f_2 X^2 + \cdots + f_t X^t$$



Needs all differences of i,j to be invertible mod N:
exceptional set Ξ_N (note: q>n for smallest q|N)

Adi Shamir (1979). How to share a secret.

(t,n)-Shamir secret sharing

degree t polynomial: $f(X) = s + \sum_{i=1}^{t} f_i X^i$

n shares: $f(1), f(2), \ldots, f(n)$

- Any polynomial of degree t is uniquely defined by t+1 evaluations
- t or less evaluations define an underdetermined set of solutions

Efficient reconstruction for any subset $S \subseteq \{1, \ldots, n\}$ of size at least t+1

$$f(x) = \sum_{i \in S} \ell_i^S(x) f(i)$$

$$\ell_i^S(X) = \prod_{j \in S \setminus \{i\}} \frac{X - j}{i - j}$$

Lagrange interpolation polynomials

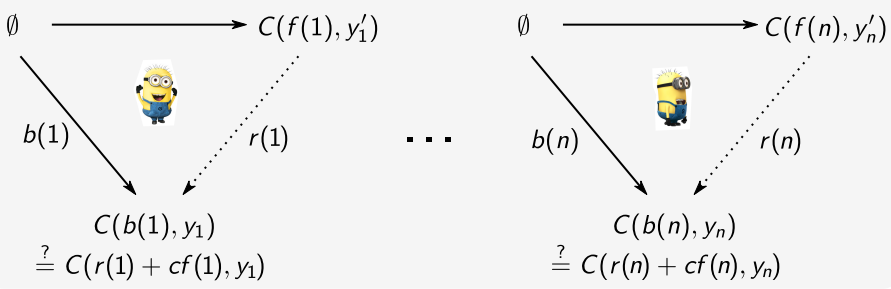
Verifiable Shamir secret sharing

Honest-majority setting

Correct if **t+1 honest parties** agree

Secure if at most **t parties corrupt**

$$\deg f = t < \frac{n}{2}$$



Commit

$b(X) \leftarrow \mathbb{Z}_N[X]_t$

$\{B_j = C(b(j), y_j)\}_j$

Challenge

$c \leftarrow \Xi_N$

Response

$r(X) = b(X) - cf(X)$

Verification

$r(X) \stackrel{?}{\in} \mathbb{Z}_N[X]_t$

$\{B_j \stackrel{?}{=} C(r(j) + cf(j), y_j)\}_j$

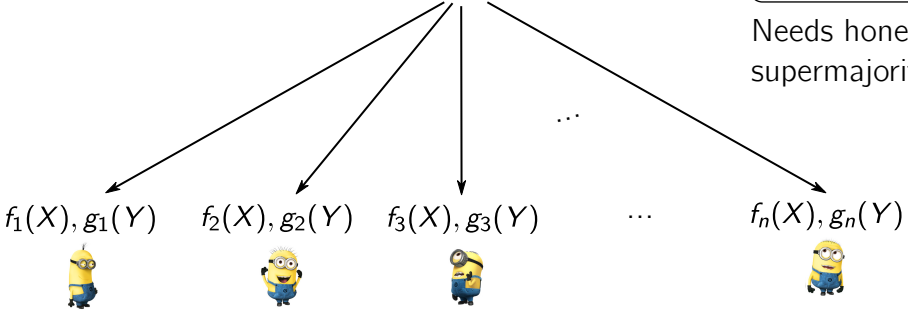
S. Atapoor, K. Baghery, D. Cozzo, R. Pedersen (2023). VSS from distributed ZK proofs and applications.

Bivariate polynomials

$$S(X, Y) \leftarrow \mathbb{Z}_N[X, Y]_{(t,t)}$$
$$s = S(0, 0)$$
$$f_i(X) = S(X, i) \quad g_i(Y) = S(i, Y)$$


$f_i(j) \stackrel{?}{=} g_j(i)$
$g_i(j) \stackrel{?}{=} f_j(i)$

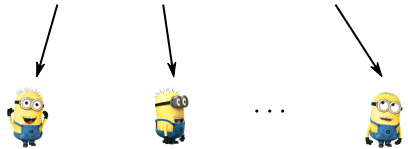

Needs honest supermajority: $t < \frac{n}{3}$



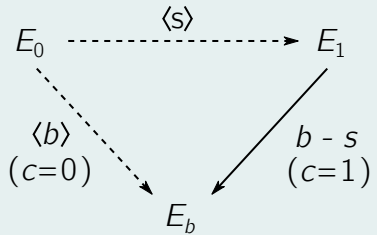
Reconstruct via $s = \sum_{i \in S} \ell_i^S(0) f_i(0)$

Distributed group action

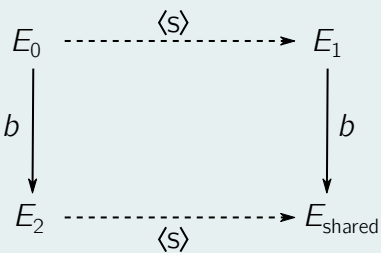
$$[s]E_0 = [f(0)]E_0 = [\ell_1^S(0)f(1)][\ell_2^S(0)f(2)]\cdots[\ell_{t+1}^S(0)f(t+1)]E_0$$



Threshold signature



Threshold decryption



L. De Feo, M. Meyer (2020). Threshold Schemes from Isogeny Assumptions.

Verifiable distributed group action computation

$$\mathcal{R}_{SPVP} = \{(E_0, E_1, a, \{x_i\}_{P_i}), f(X) \mid E_1 = [af(0)]E_0 \wedge x_i = f(i)\}$$

$$\subseteq \mathcal{E}^2 \times \mathbb{Z}_N^{1+n} \times \mathbb{Z}_N[X]_t$$

(structured piecewise verifiable proofs)

$$E_0 \xrightarrow{af(0)} E_1 \qquad \emptyset \longrightarrow \{C(f(j), y'_j)\}_j$$

$$\bigwedge_j$$

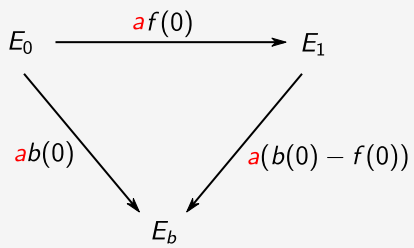
S. Atapoor, K. Baghery, D. Cozzo, R. Pedersen (2023). Practical robust DKG protocols for CSIDH.

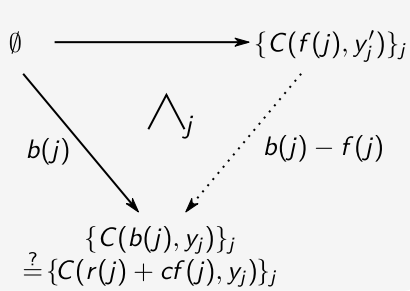
Verifiable distributed group action computation

$$\mathcal{R}_{SPVP} = \{(E_0, E_1, a, \{x_i\}_{P_i}), f(X) \mid E_1 = [af(0)]E_0 \wedge x_i = f(i)\}$$

(structured piecewise verifiable proofs)

$$\subseteq \mathcal{E}^2 \times \mathbb{Z}_N^{1+n} \times \mathbb{Z}_N[X]_t$$





$\Longrightarrow$ Honest-majority setting

Commit

$b(X) \leftarrow \mathbb{Z}_N[X]_t$

$E_b = [ab(0)]E_0$

$\{B_j = C(b(j), y_j)\}_j$

Challenge

$c \leftarrow \{0, 1\}$

Response

$r(X) = b(X) - cf(X)$

Verification

$r(X) \stackrel{?}{\in} \mathbb{Z}_N[X]_t$

$E_b \stackrel{?}{=} [ar(0)]E_c$

$\{B_j \stackrel{?}{=} C(r(j) + cf(j), y_j)\}_j$

S. Atapoor, K. Baghery, D. Cozzo, R. Pedersen (2023). Practical robust DKG protocols for CSIDH.

Distributed key generation protocol

Verifiable secret sharing

$\{f^{(i)}(X) \leftarrow \mathrm{VSS}_i(1^\lambda, t, n)\}_i$

+ proof of Shamir shares

Define $f(X) = \sum_{i \in S} f^{(i)}(X)$

secret $s = f(0)$ share of P_j is $f(j)$

$\langle s \rangle \leftarrow \mathrm{VSS}(1^\lambda, t, n)$

ROBUST

Distributed group action computation

$[\ell_1^T f(1)] \cdots [\ell_\tau^T f(\tau)] E_0$

+ structured PVPs

$E = [\langle s \rangle] E_0$

ROBUST

Direct reconstruction

$s = \sum_{i \in S} \ell_i^S(0) f(i)$ $s \leftarrow \mathrm{Rec}(\langle s \rangle)$

ROBUST

Linear reconstruction

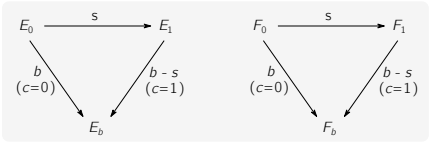
$\langle b - cs \rangle = \langle b \rangle - c\langle s \rangle$

Shamir ~~ROBUST~~ Bivariate **ROBUST**

More Complex Arithmetic from Group Actions

Proofs of arithmetic operations

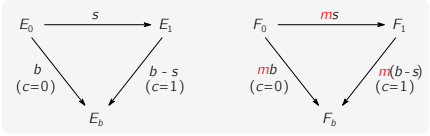
AND-proof



$$\mathcal{R}_{AND} = \{(E_0, E_1, F_0, F_1), s \mid E_1 = [s]E_0 \wedge F_1 = [s]F_0\}$$

Addition: $(E_0, [s]E_0, [f]E_0, [f+s]E_0)$

SCAL-proof



$$\mathcal{R}_{SCAL} = \{(E_0, E_1, F_0, F_1, m), s \mid E_1 = [s]E_0 \wedge F_1 = [ms]F_0\}$$

$$\mathcal{R}_{SCAL}^{F_0=E_0} = \{(E_0, E_1, F_1, m), s \mid E_1 = [s]E_0 \wedge F_1 = [ms]E_0\}$$

Scalar multiplication: $(E_0, m, [s]E_0, [ms]E_0)$

Exponentiation

$(E_0, e, [s]E_0, [s^e]E_0)$

Multiplication

$(E_0, [s]E_0, [f]E_0, [fs]E_0)$

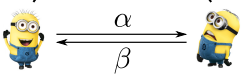
Polynomial evaluation

$(E_0, f(X), [s]E_0, [f(s)]E_0)$

$(E_0, [f_1]E_0, \ldots, [f_t]E_0, [s]E_0, [f(s)]E_0)$

Multi-Party Multiplication

$$[ab]E_0 = [(\alpha + x)(\beta + y)]E_0 = [\alpha\beta][\beta x][\alpha y][xy]E_0$$
$$= [\alpha\beta][\beta x + z_1][\alpha y + z_2]E_0$$



$\alpha = a - x$ $\quad \alpha \rightleftarrows \beta \quad$ $\beta = b - y$

(x, z_1) $\quad$ (y, z_2)

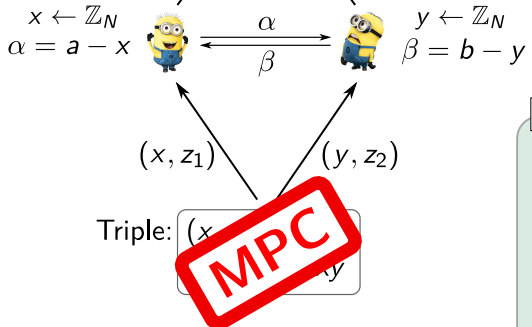
Triple: $(x, y, (z_1, z_2))$ with $z_1 + z_2 = xy$

C. Delpech de Saint Guilhem, R. Pedersen (2024). New proof systems and an OPRF from CSIDH.

Multi-Party Multiplication (in the head)

$$[ab]E_0 = [(\alpha + x)(\beta + y)]E_0 = [\alpha\beta][\beta x][\alpha y][xy]E_0$$

$$= [\alpha\beta][\beta x + z_1][\alpha y + z_2]E_0$$



OT protocols + symmetric OT extension

MPC-in-the-head proof system

$([a]E_0, [b]E_0, [ab]E_0)$

Triples: (x_i, y_i, z_i)

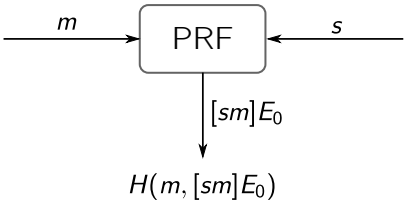
$$[ab]E_0 = \Big[\alpha\beta\Big]\Big[\sum_{i=1}^{n} \beta x_i + \alpha y_i + z_i\Big]E_0$$

C. Delpech de Saint Guilhem, R. Pedersen (2024). New proof systems and an OPRF from CSIDH.

Compact Pseudorandom Functions

from Group Actions

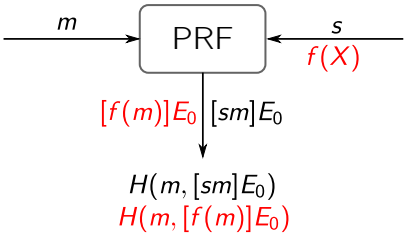
New pseudorandom function



One-more assumption

Given oracle access to $[sm]E_0 \leftarrow \mathcal{O}_s(m)$, produce $(m^*, [sm^*]E_0)$ for m^* not queried.

New pseudorandom function



Concern 1: Quantum access to PRF allows recovery of the key s [1]

Concern 2: Reduced quantum security w.r.t. Vectorization [2]

Concern 3: Security reduced to largest prime divisor of class group size N [3]

KLaPoTi and PEGASIS give us choice over N

Prevent quantum access to PRF and increase parameter sizes **or**
use polynomial evaluations instead of multiplication

One-more assumption

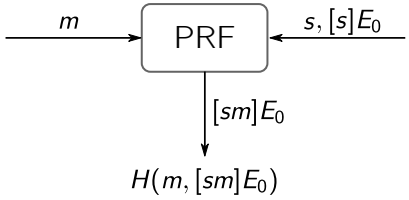
Given oracle access to $[sm]E_0 \leftarrow \mathcal{O}_s(m)$, produce $(m^*, [sm^*]E_0)$ for m^* not queried.

[1] **S. Galbraith, L. Panny, B. Smith, F. Vercauteren (2018).** Quantum Equivalence of the DLP and CDHP for Group Actions

[2] **P. Frixons, V. Gilchrist, P. Kutas, S. Merz, C. Petit (2025).** Another Look at the Quantum Security of the Vectorization Problem with Shifted Inputs

[3] **Y. Lai (2024).** A Note on Isogeny Group Action-Based Pseudorandom Functions.

Verifiable and oblivious pseudorandom functions



Verifiable random function

⟶ prove that $[sm]E_0$ is correct

$$\mathcal{R}^*_{SCAL} = \{(E_0, E_1, F, m), s \mid E_1 = [s]E_0 \wedge F = [ms]E_0\}$$

~4x less communication

~400x less group actions

Oblivious pseudorandom function

$$\mathcal{R}^*_{SCAL} = \{(E_0, E_1, F, m), s \mid E_1 = [s]E_0 \wedge F = [ms]E_0\}$$

⟶ input m and output $[sm]E_0$ are hidden from server

$[ms]E_0 = [\beta m + z_1]\ [\alpha y + z_2]\ E_0$

Triple:

$(x, y, (z_1, z_2))$

$z_1 + z_2 = xy$

$\alpha = m - x$

(x, z_1)

α

β

$\beta = s - y$

(y, z_2)

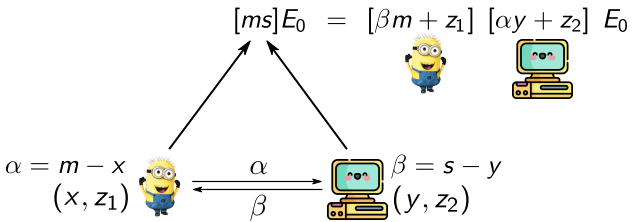
~100x less communication

~200x less group actions

Verifiable oblivious pseudorandom function

Triple:

$(x, y, (z_1, z_2))$
$z_1 + z_2 = xy$



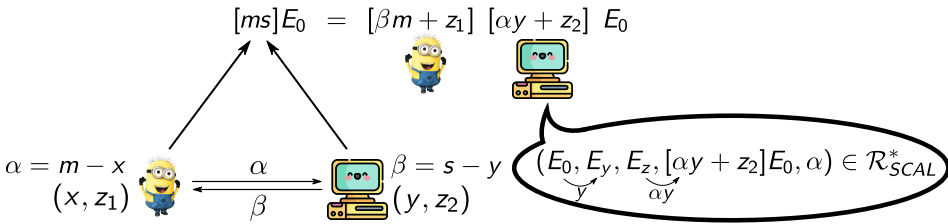
$$\mathcal{R}_{SCAL} = \{(E_0, E_1, F_0, F_1, m), s \mid E_1 = [s]E_0 \wedge F_1 = [ms]F_0\} \subset \mathcal{E}^4 \times \mathbb{Z}_N \times \mathbb{Z}_N$$

Verifiable oblivious pseudorandom function

Triple:

$(x, y, (z_1, z_2))$
$z_1 + z_2 = xy$

$E_y = [y]E_0,$
$E_z = [z_2]E_0$

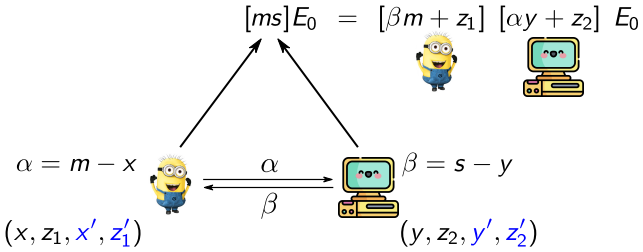


- much heavier setup
- the extra curves prevent standard OT extension techniques
- security issue for multiplication (not for polynomial evaluation)

Verifiable oblivious pseudorandom function

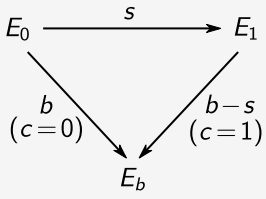
Triple:

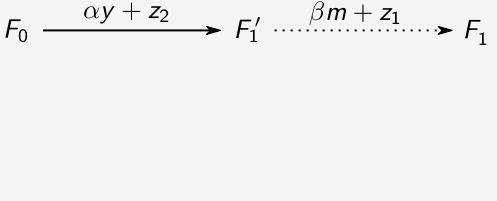
$(x, y, (z_1, z_2))$
$z_1 + z_2 = xy$
$(x', y', (z_1', z_2'))$
$z_1' + z_2' = x'y'$



Commit
$b \leftarrow \mathbb{Z}_N$
$E_b = [b]E_0$

$$\mathcal{R}_{SCAL} = \{(E_0, E_1, F_0, F_1, m), s \mid E_1 = [s]E_0 \wedge F_1 = [ms]F_0\} \subset \mathcal{E}^4 \times \mathbb{Z}_N \times \mathbb{Z}_N$$

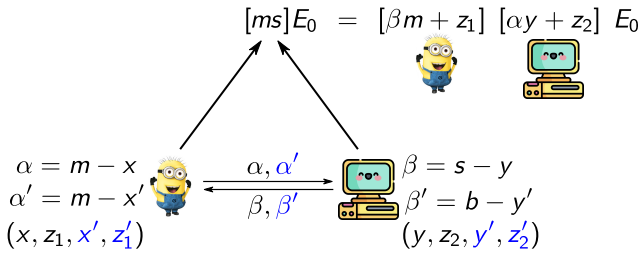




Verifiable oblivious pseudorandom function

Triple:

$$(x, y, (z_1, z_2))$$
$$z_1 + z_2 = xy$$
$$(x', y', (z_1', z_2'))$$
$$z_1' + z_2' = x'y'$$



Commit

$b \leftarrow \mathbb{Z}_N$

$E_b = [b]E_0$

$F_b' = [\alpha' y' + z_2']F_0$

Challenge

$c \leftarrow \{0, 1\}$

Response

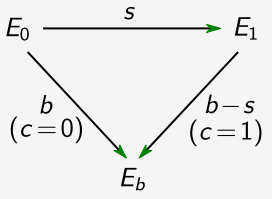
$r = b - cs$

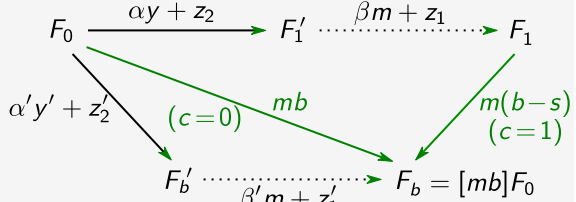
Verification

$[r]E_c \stackrel{?}{=} E_b \wedge$

$[mr]F_c \stackrel{?}{=} F_b$

$\mathcal{R}_{SCAL} = \{(E_0, E_1, F_0, F_1, m), s \mid E_1 = [s]E_0 \wedge F_1 = [ms]F_0\} \subset \mathcal{E}^4 \times \mathbb{Z}_N \times \mathbb{Z}_N$





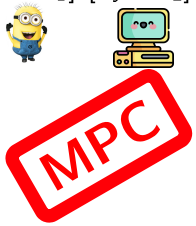
Verifiable oblivious pseudorandom function

$$[ms]E_0 \;=\; [\beta m + z_1]\;[\alpha y + z_2]\;E_0$$

Triple:

$$(x, y, (z_1, z_2))$$
$$z_1 + z_2 = xy$$
$$(x', y', (z_1', z_2'))$$
$$z_1' + z_2' = x'y'$$

$\Rightarrow$ λ tuples $(x^{(\lambda)}, y^{(\lambda)}, (z_1^{(\lambda)}, z_2^{(\lambda)}))$



Commit

$b \leftarrow \mathbb{Z}_N$

$E_b = [b]E_0$

$F_b' = [\alpha' y' + z_2']F_0$

Challenge

$c \leftarrow \{0, 1\}$

Response

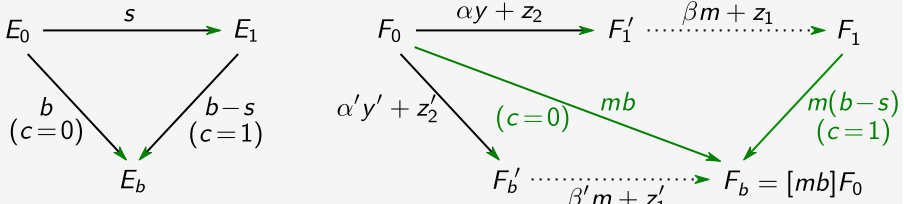
$r = b - cs$

Verification

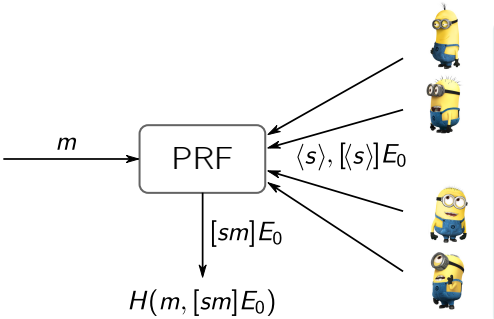
$[r]E_c \stackrel{?}{=} E_b \wedge$

$[mr]F_c \stackrel{?}{=} F_b$

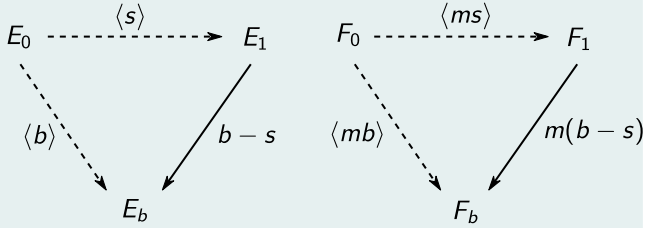
$\mathcal{R}_{SCAL} = \{(E_0, E_1, F_0, F_1, m), s \mid E_1 = [s]E_0 \wedge F_1 = [ms]F_0\} \subset \mathcal{E}^4 \times \mathbb{Z}_N \times \mathbb{Z}_N$



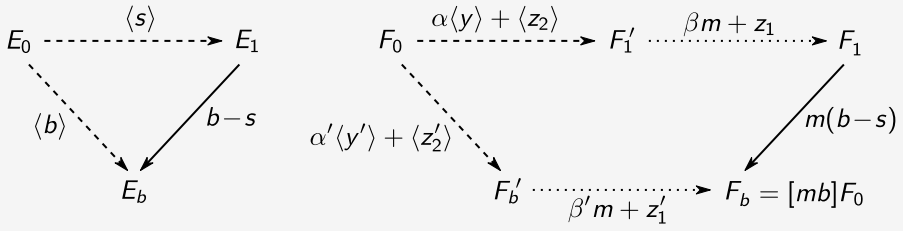
Thresholdizing pseudorandom functions



Threshold scalar proof ⇒ **Threshold VRF**



Threshold multiplication proof ⇓ **Threshold VOPRF**



Comparison of threshold VOPRF (malicious setting)

	Assumptions	Sharing	Server-side rounds	Verifier cost	Verifiability	Robustness
Ours	Isogenies (Group action)	bivariate (t<n/3)	n (round-robin)	O(1)	Yes	Yes
[1]	Legendre PRF	replicated (t<n/3) and full	1-2 (client-side aggregation)	O(n)	Yes(*)	No (abort)
[2]	Lattices (RingLWE)	full and replicated	1 (client-side aggregation)	O(n)	Yes	No (abort)

[1] **N. Cheng, N. Kaluđerović, K. Mitrokotsa (2024).** A post-quantum Distributed OPRF from the Legendre PRF

[2] **M.Albrech, K. Gur (2024).** Verifiable Oblivious Pseudorandom Functions from Lattices: Practical-ish and Thresholdisable.


THANK
YOU!