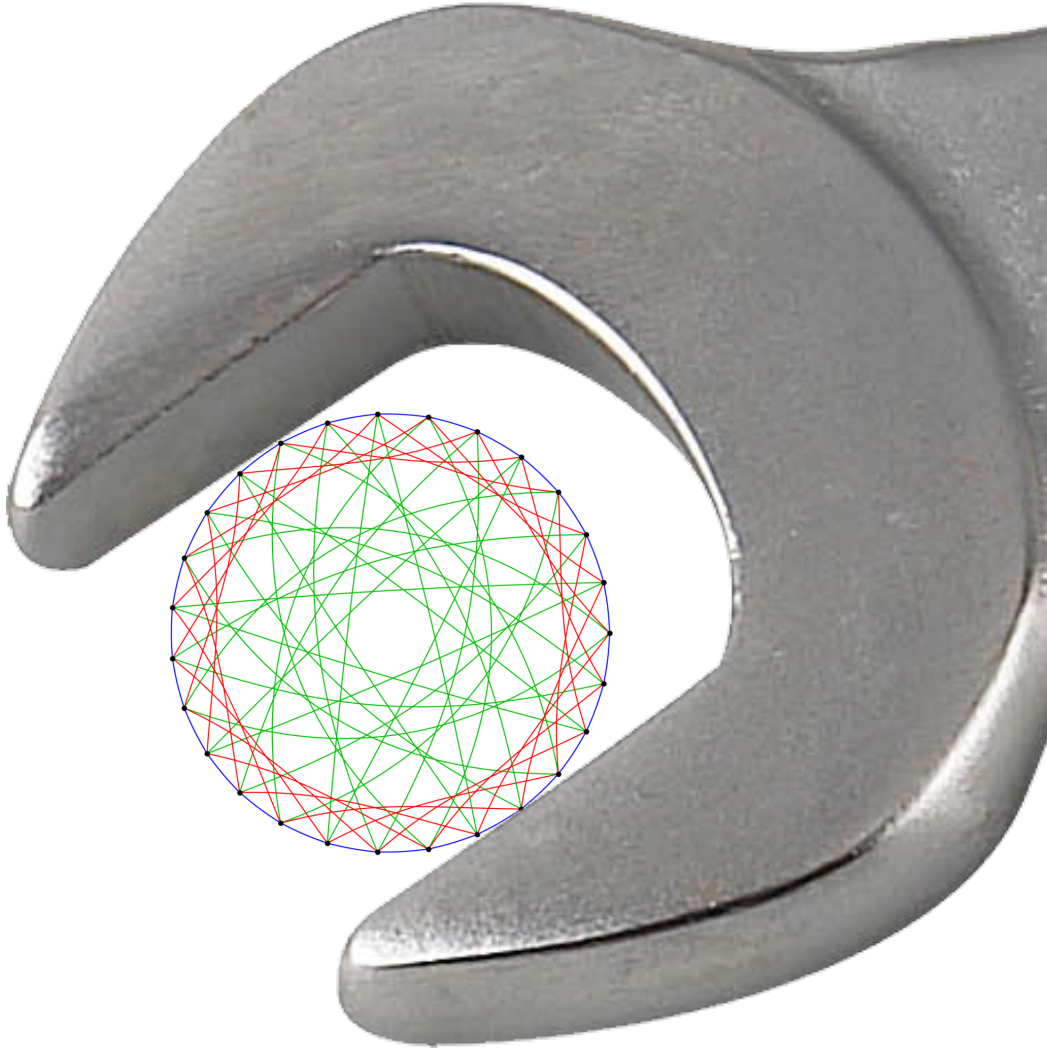
A new toolbox for CSIDH



Cyprien Delpech de Saint Guilhem
Robi Pedersen

Group Action Setting

Group G acting freely and transitively on a set $\mathcal{E}$ by the action

$$\star : G \times \mathcal{E} \to \mathcal{E}$$
$$(\mathfrak{a}, E) \mapsto E' = \mathfrak{a} \star E$$
$$\mathfrak{a} \star (\mathfrak{b} \star E) = (\mathfrak{a}\mathfrak{b}) \star E$$

Assume $\langle \mathfrak{g} \rangle = G$

$$[\,] : \mathcal{G} \times \mathcal{E} \to \mathcal{E}$$
$$(a, E) \mapsto E' = [a]E = \mathfrak{g}^a \star E$$
$$[a]([b]E) = [a+b]E$$

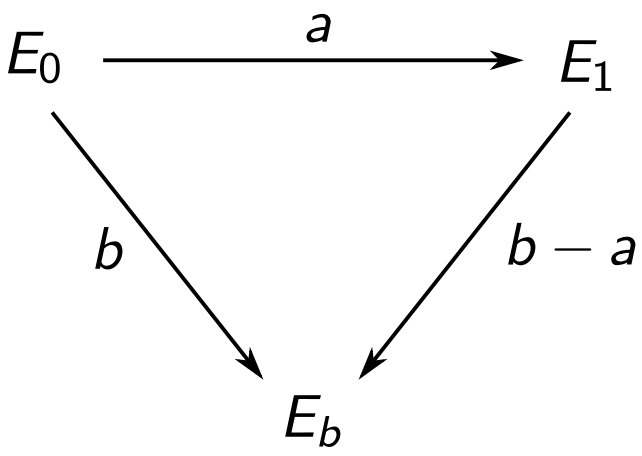
Exponent group $\mathcal{G} = \mathbb{Z}/|G|\mathbb{Z}$

$[ab]E = \mathfrak{g}^{ab} \star E$
(Scalar) Multiplication

$[a^e]E = \mathfrak{g}^{a^e} \star E$
Exponentiation

$[f(a)]E = \mathfrak{g}^{f(a)} \star E$
Polynomial Evaluation

ID protocol



$\mathcal{P}$ $\mathcal{V}$

$b \leftarrow \mathcal{G}$

$E_b = [b]E_0$ $\xrightarrow{E_b}$

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

$\xleftarrow{c}$

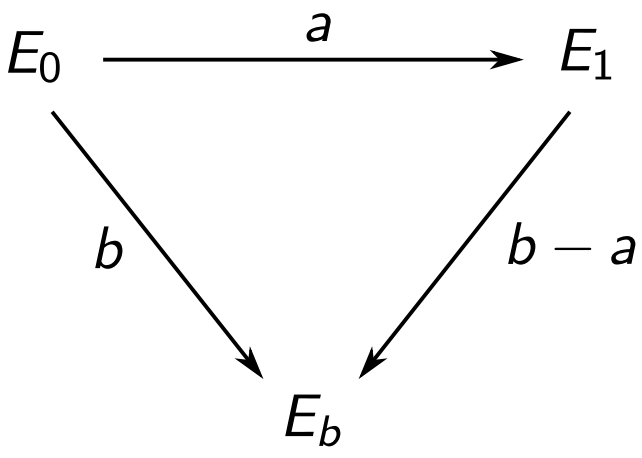
$r = b - ca$

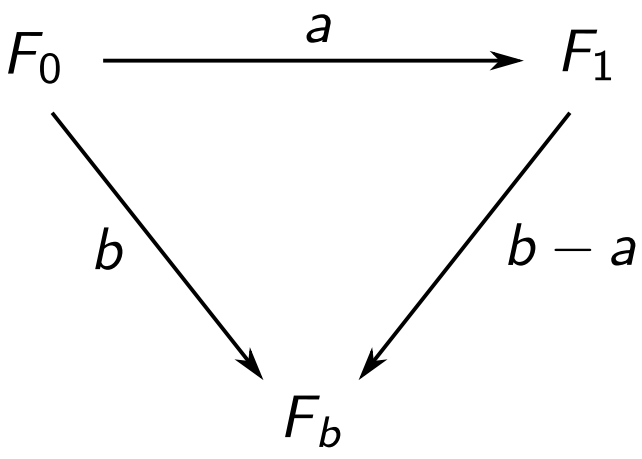
$\xrightarrow{r}$

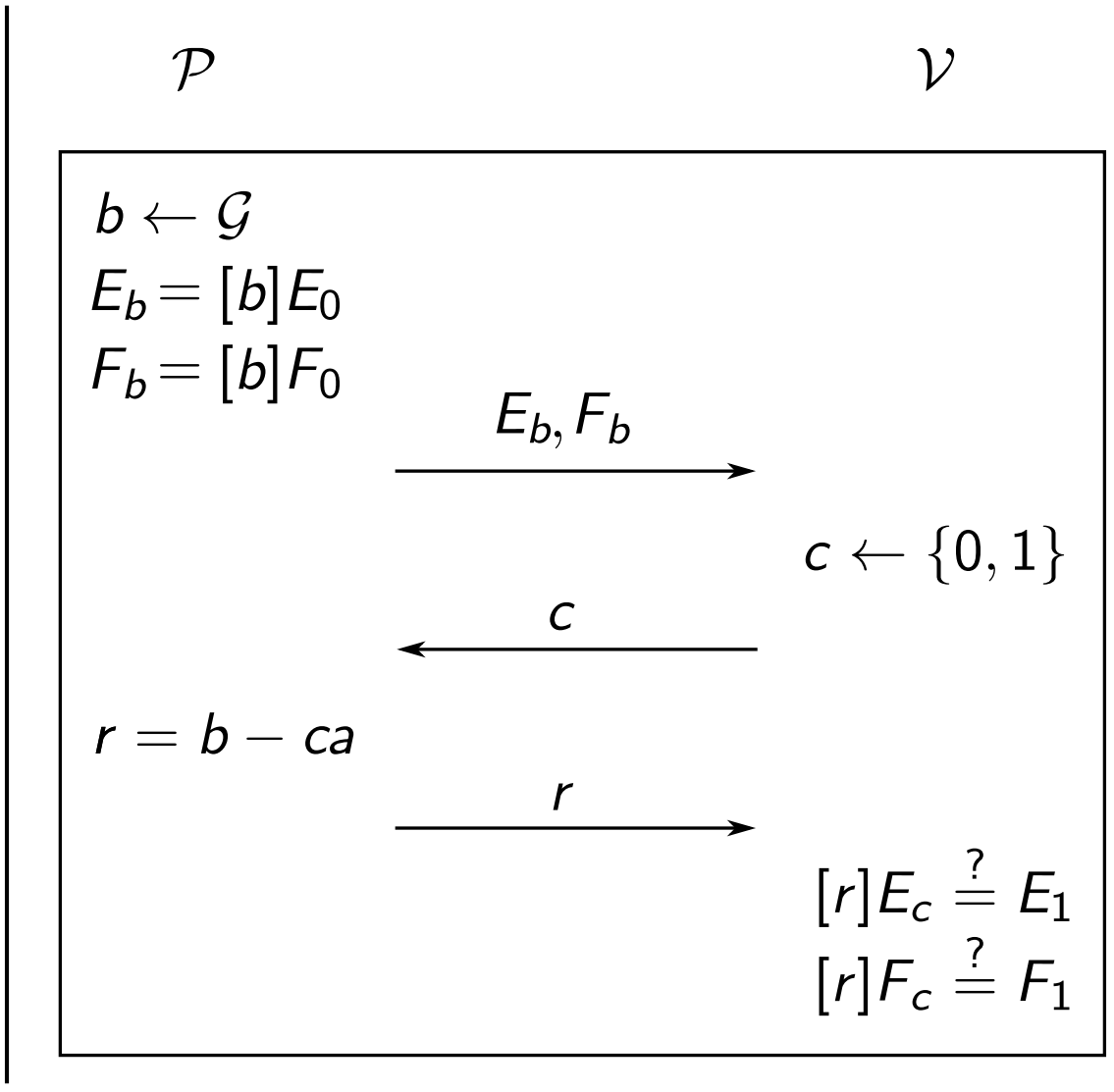
$[r]E_c \stackrel{?}{=} E_1$

$$\mathcal{L} = \left\{ ((E_0, E_1), a) : E_1 = [a]E_0 \right\}$$

Addition

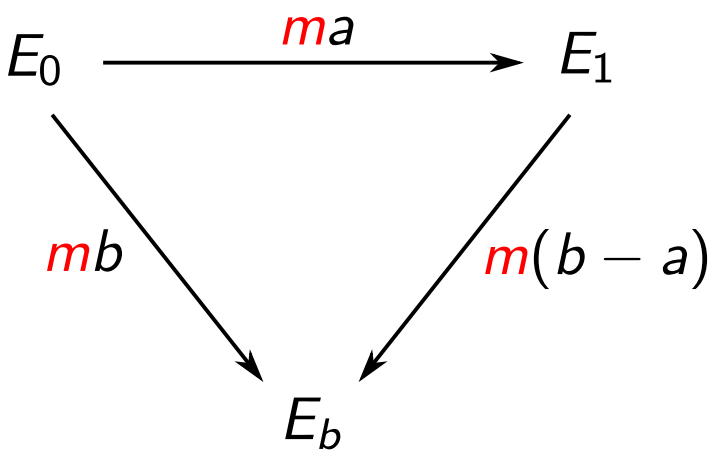


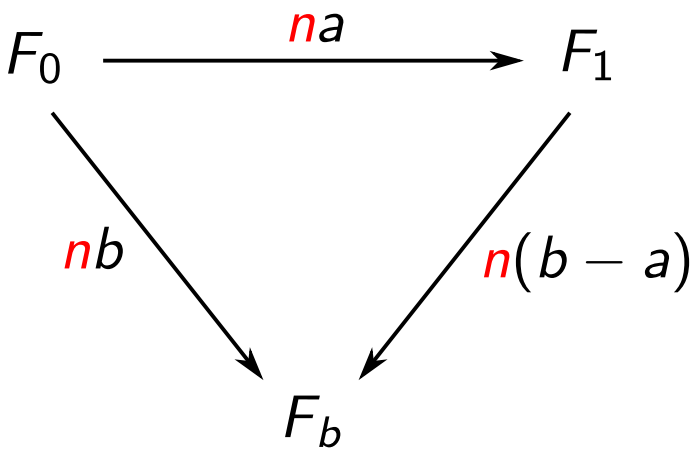


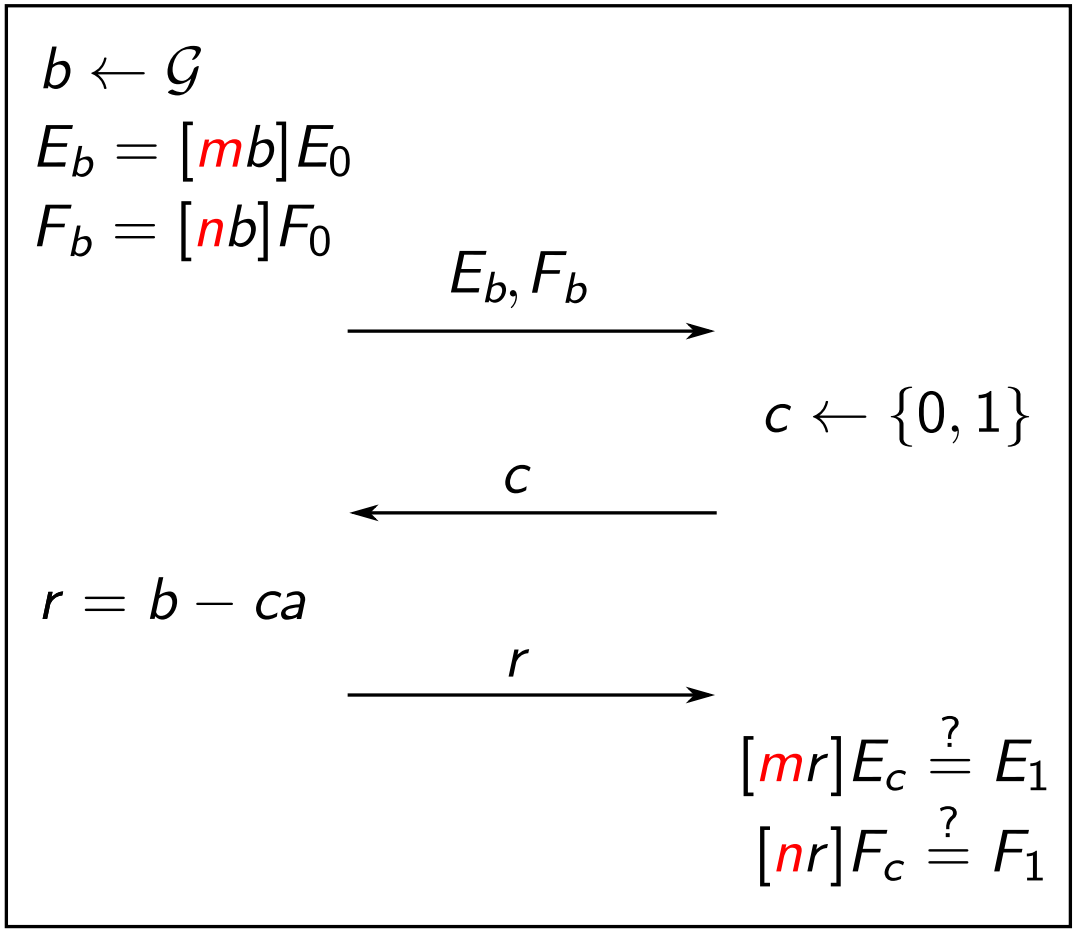


$$\mathcal{L}_2^{\mathsf{Add}} = \left\{ ((E_0, E_1), (F_0, F_1), a) : E_1 = [a]E_0 \wedge F_1 = [a]F_0 \right\}$$

Scalar Multiplication







$$\mathcal{L}_2^{\text{Scal}} = \left\{ ((E_0, E_1, m), (F_0, F_1, n), a) : E_1 = [ma]E_0 \wedge F_1 = [na]F_0 \right\}$$

Languages

$$\mathcal{L}_k^{\mathsf{Add}} = \left\{ \left((E_i, E_i')_{i=1,\ldots,k}, a \right) : \bigwedge_{i=1}^{k} E_i' = [a]E_i \right\}$$

$$\mathcal{L}_k^{\mathsf{Scal}} = \left\{ \left((E_i, E_i', c_i)_{i=1,\ldots,k}, a \right) : \bigwedge_{i=1}^{k} E_i' = [c_i a]E_i \right\}$$

$$\mathcal{L}^{\mathsf{Exp}} = \left\{ \left(([a]E, E', e), a \right) : E' = [a^e]E \right\}$$

$$\mathcal{L}^{\mathsf{Mult}} = \left\{ \left(([a]E, [b]E, E'), (a, b) \right) : E' = [ab]E \right\}$$

Addition $\equiv$ Scalar Mult. $\leq$ Exponentiation $\equiv$ Multiplication $\leq$ Key Recovery

A proof for $\mathcal{L}^{\mathsf{Mult}}$

$x, y \leftarrow \mathcal{G}$

$\alpha = a - x$

$\beta = b - y$

public

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

$$\pi_1 : \Big((E_0, [y]E_0, 1), (F, [\alpha y]F, \alpha), y\Big) \in \mathcal{L}_2^{\mathsf{Scal}}$$

$$\pi_2 : \Big((E_0, [x]E_0, 1), (F', [\beta x]F', \beta), x\Big) \in \mathcal{L}_2^{\mathsf{Scal}}$$

A proof for $\mathcal{L}^{\mathsf{Mult}}$

$x, y \leftarrow \mathcal{G}$

$\alpha = a - x$

$\beta = b - y$

public

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

$$\pi_1 : \Big((E_0,\ M_y\ ,1), (F, [\alpha y]F, \alpha), y\Big) \in \mathcal{L}_2^{\mathsf{Scal}}$$

$$\pi_2 : \Big((E_0,\ M_x\ ,1), (F', [\beta x]F', \beta), x\Big) \in \mathcal{L}_2^{\mathsf{Scal}}$$

$(\pi_1, \pi_2, \alpha, \beta)$

$$[\alpha]M_x \stackrel{?}{=} E_a \wedge [\beta]M_y \stackrel{?}{=} E_b \wedge \mathsf{Verify}(\pi_1) \wedge \mathsf{Verify}(\pi_2)$$

Trusted Setup $\mathcal{T} : (x, y, M_x = [x]E_0, M_y = [y]E_0, M_z = [xy]E_0)$

A proof for $\mathcal{L}^{\text{Mult}}$ using MPC-in-the-head

$$\mathcal{T} : \Big((x_i, y_i, z_i)_{i\in\{1,\ldots,n\}}, M_x, M_y\Big)$$

$$\sum_{i=1}^{n} x_i = x, \quad \sum_{i=1}^{n} y_i = y, \quad \sum_{i=1}^{n} z_i = xy$$

$$M_x = [x]E_0, \quad M_y = [y]E_0$$

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

E_0

$\mathcal{P}_1 : (\alpha, \beta, x_1, y_1, z_1)$	$E_1 = [\beta x_1 + \alpha y_1 + z_1]E_0$
$\mathcal{P}_2 : (\alpha, \beta, x_2, y_2, z_2)$	$E_2 = [\beta x_2 + \alpha y_2 + z_2]E_1$
$\mathcal{P}_3 : (\alpha, \beta, x_3, y_3, z_3)$	$E_3 = [\beta x_3 + \alpha y_3 + z_3]E_2$
$\ldots$	$\ldots$
$\mathcal{P}_{n-1} : (\alpha, \beta, x_{n-1}, y_{n-1}, z_{n-1})$	$E_{n-1} = [\beta x_{n-1} + \alpha y_{n-1} + z_2]E_{n-2}$
$\mathcal{P}_n : (\alpha, \beta, x_n, y_n, z_n)$	$E_n = [\alpha\beta + \beta x_n + \alpha y_n + z_n]E_{n-1}$

A proof for $\mathcal{L}^{\text{Mult}}$ using MPC-in-the-head

$T_1 : \left((x_i, y_i, z_i)_{i\in\{1,\dots,n\}}, M_x, M_y\right)_1$ $\quad$ $T_2 : \left((x_i, y_i, z_i)_{i\in\{1,\dots,n\}}, M_x, M_y\right)_2$

$T_3 : \left((x_i, y_i, z_i)_{i\in\{1,\dots,n\}}, M_x, M_y\right)_3$ $\quad$ $\dots$

$T_{m-1} : \left((x_i, y_i, z_i)_{i\in\{1,\dots,n\}}, M_x, M_y\right)_{m-1}$ $\quad$ $T_m : \left((x_i, y_i, z_i)_{i\in\{1,\dots,n\}}, M_x, M_y\right)_m$

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

$\mathcal{P}_1 : (\alpha, \beta, x_1, y_1, z_1)$ $\quad$ E_0

$\mathcal{P}_2 : (\alpha, \beta, x_2, y_2, z_2)$ $\quad$ $E_1 = [\beta x_1 + \alpha y_1 + z_1]E_0$

$\mathcal{P}_3 : (\alpha, \beta, x_3, y_3, z_3)$ $\quad$ $E_2 = [\beta x_2 + \alpha y_2 + z_2]E_1$

$\dots$ $\quad$ $E_3 = [\beta x_3 + \alpha y_3 + z_3]E_2$

$\mathcal{P}_{n-1} : (\alpha, \beta, x_{n-1}, y_{n-1}, z_{n-1})$ $\quad$ $\dots$

$\mathcal{P}_n : (\alpha, \beta, x_n, y_n, z_n)$ $\quad$ $E_{n-1} = [\beta x_{n-1} + \alpha y_{n-1} + z_2]E_{n-2}$

$E_n = [\alpha\beta + \beta x_n + \alpha y_n + z_n]E_{n-1}$

Costs

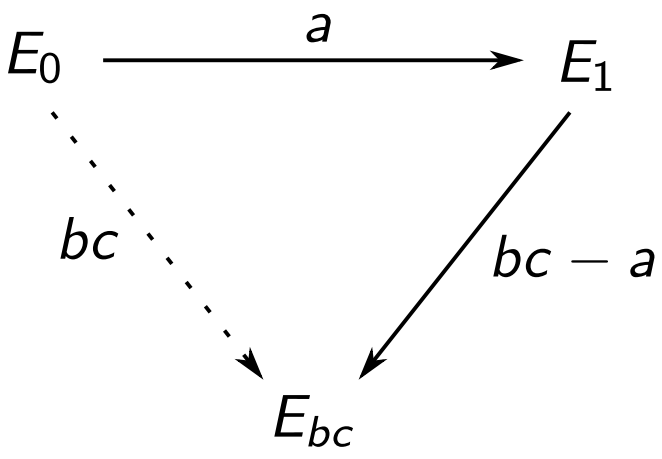
	Prover cost	Verifier cost	Trusted Party	Statement type
Scal.ZK$_{k=1}$	λ	λ	No	$(E, [s]E)$
Scal.ZK$_{k=2}$	2λ	2λ	No	$(E, [a]E, [b]E, [a+b]E)$
Mult.ZK	4λ	4λ	Yes	$(E, [a]E, [b]E, [ab]E)$
MultITH$_{\mathcal{T}}$	1.89λ	2.52λ	Yes	
MultITH$_{\text{cnc}}$	4.96λ	4.25λ	No	
Exp.ZK	$\lfloor \log_2 e \rfloor 6\lambda$	$\lfloor \log_2 e \rfloor 6\lambda$	Yes	$(e, [a]E, [a^e]E)$
ExpITH$_{\mathcal{T}}$	1.89λ	1.89λ	Yes	
ExpITH$_{cnc}$	3.52λ	3.52λ	No	

ZK : standard zero-knowledge proof

ITH : MPC-in-the-Head proof

$\mathcal{T}$: Trusted Setup , *cnc* : Cut-and-Choose

Applications: ID protocol



$\mathcal{P}$ $\mathcal{V}$

$b \leftarrow \mathcal{G}$ $c \leftarrow \mathcal{G}$

Interactive protocol

$\Downarrow$

$E_{bc} = [bc]E_0$

c

$r = bc - a$

r

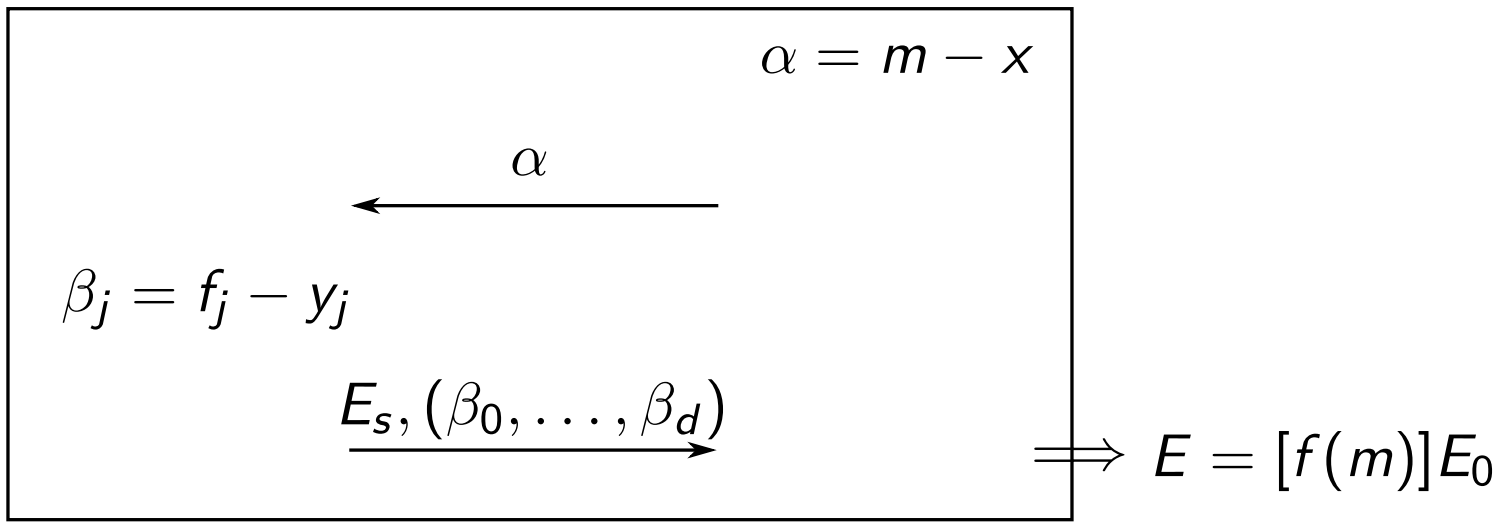
$[r]E_{bc} \stackrel{?}{=} E_1$

$$\mathcal{L} = \left\{ ((E_0, E_1), a) : E_1 = [a]E_0 \right\}$$

Applications: (Verifiable) oblivious pseudo-random function

$\mathcal{S}$ key $k = (f_0, f_1, \ldots, f_d) \in \mathcal{G}^{d+1}$

$\mathcal{C}$ input $m \in \mathcal{G}$



- $\deg f \geq 2$
- f has uniform output
- f is collision-resistant
- Given oracle access to $[f(m)]E_0 \leftarrow \mathcal{O}(m)$, find $(m^*, [f(m^*)]E_0)$ with m^* not yet queried.

Applications: (Verifiable) oblivious pseudo-random function

Source	Malicious Client	Verifiable	Number of Rounds	Approx. Computational cost	Approx. Communication cost	No Trusted Setup	Without Class Group
[1]	X	X	$2\lambda + 2$	3λ	$3\lambda \log p$	✓	✓
[2]	X	X	2	5λ	$2\lambda(\log p + \lambda)$	X	X
	✓	X	4	11λ	$5\lambda(\log p + \lambda)$	X	X
$d = 2$	✓	X	2	2	$6 \log p$	X*	X
$d = 3$	✓	X	2	2	$9 \log p$	X*	X
$d = 2$	✓	✓	2	8λ	$2\lambda \log p$	X	X
$d = 3$	✓	✓	2	14λ	$3.5\lambda \log p$	X	X

*: Trusted setup can be removed for one-time cost of 5λ.

[1]: Lena Heimberger, Tobias Hennerbichler, Fredrik Meisingseth, Sebastian Ramacher, Christian Rechberger: *OPRFs from Isogenies: Designs and Analysis*. Cryptology ePrint Archive, Paper 2023/369 (2023).

[2]: Dan Boneh, Dmitry Kogan, Katharine Woo: *Oblivious Pseudorandom Functions from Isogenies*. In Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part II. LNCS, vol. 12492, pp. 520-550. Springer, Heidelberg (Dec 2020).

Thank you!

