KU LEUVEN

CSI-RAShi:
Distributed key generation for CSIDH

Ward Beullens[1] Lucas Disson[2]

Robi Pedersen[1] Frederik Vercauteren[1]

[1] imec-COSIC, ESAT, KU Leuven, Belgium
(firstname.lastname@esat.kuleuven.be)

[2] ENS, Lyon, France
(lucas.disson@ens-lyon.fr)

CSI-RAShi

Commutative Supersingular Isogeny

Robust and Actively secure distributed Shamir secret sharing

CSI-RAShi

Commutative Supersingular Isogeny
Robust and Actively secure distributed Shamir secret sharing

CSI-RAShi

Commutative Supersingular Isogeny

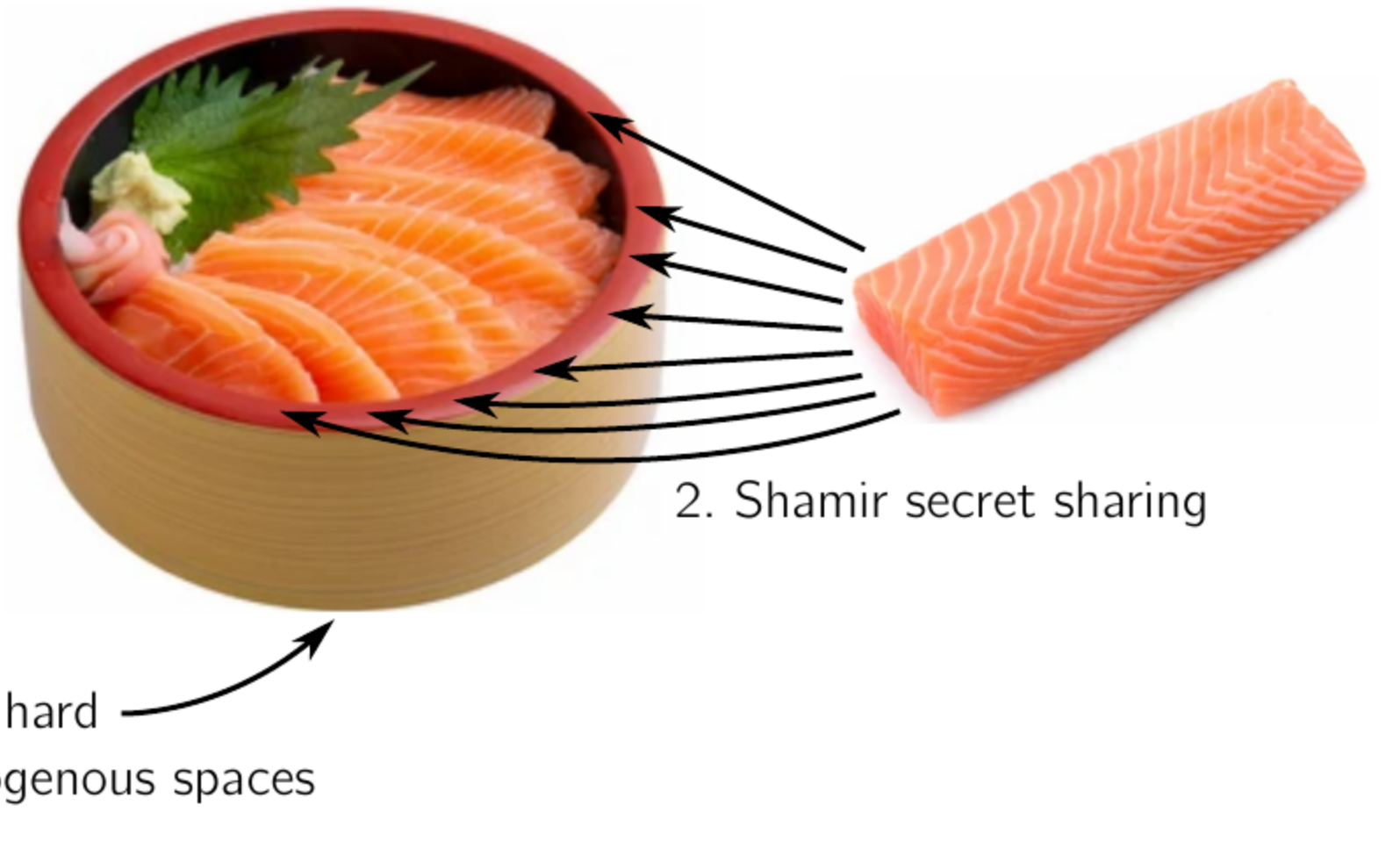
Robust and Actively secure distributed Shamir secret sharing

1. Very hard homogenous spaces

CSI-RAShi

Commutative Supersingular Isogeny

Robust and Actively secure **distributed Shamir secret sharing**

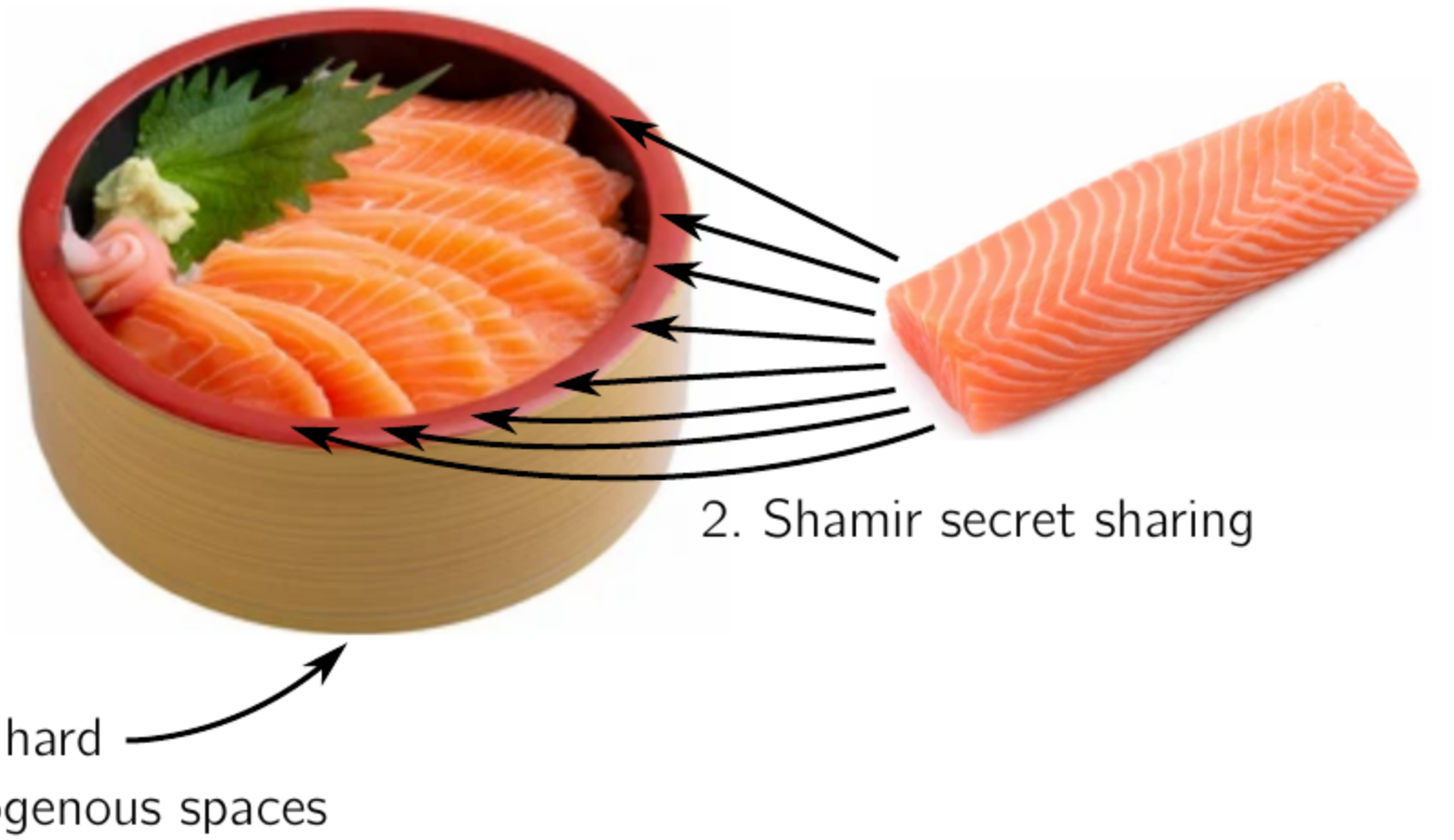


CSI-RAShi

Commutative Supersingular Isogeny

Robust and Actively secure distributed Shamir secret sharing

3. Piecewise verifiable proofs



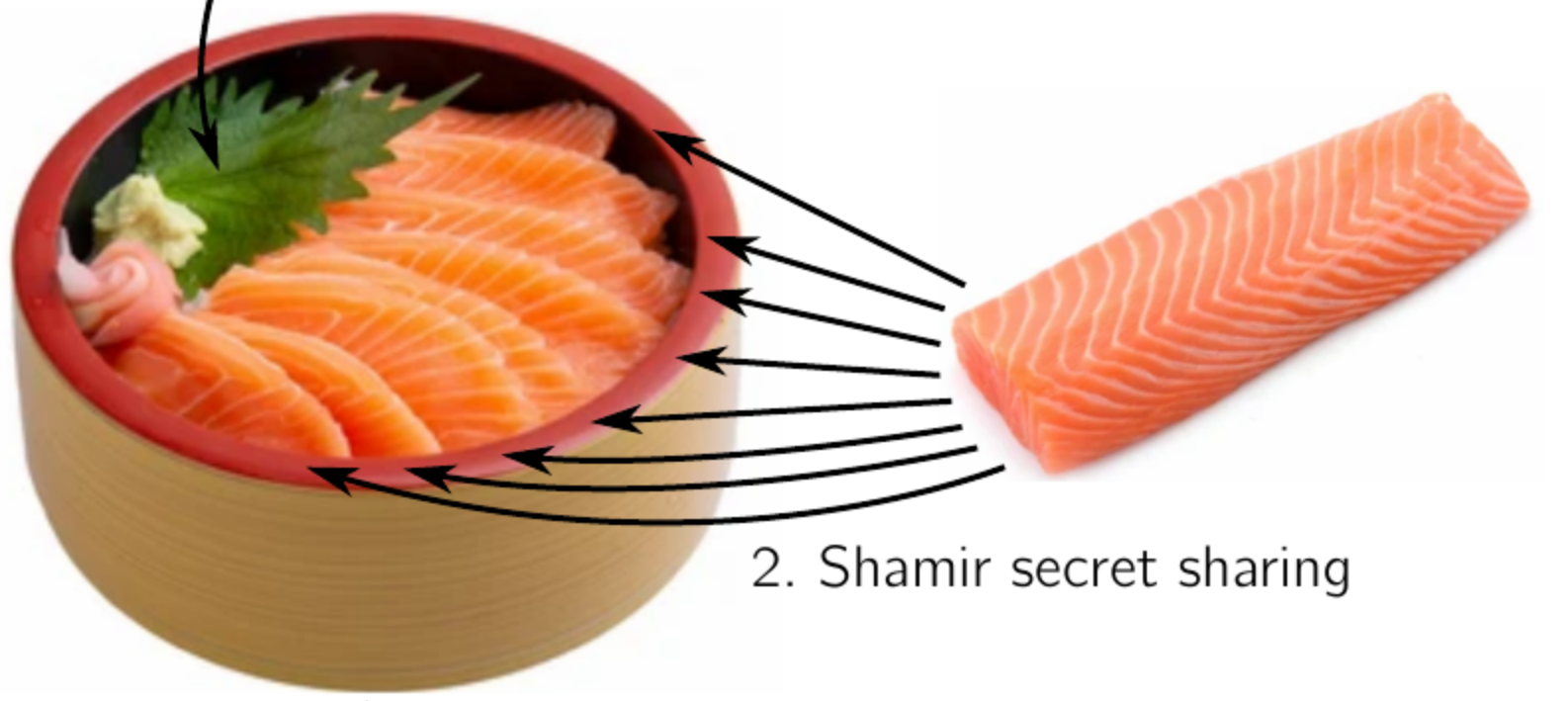
CSI-RAShi

Commutative Supersingular Isogeny

Robust and Actively secure distributed Shamir secret sharing

3. Piecewise verifiable proofs

4. Zero-knowledge proofs



2. Shamir secret sharing

1. Very hard homogenous spaces

Elliptic curve points	Very hard homogeneous spaces
Ring $\mathbb{Z}_N$, Group $\mathcal{P}$	
$[\ ] : \mathbb{Z}_N \times \mathcal{P} \to \mathcal{P}$	
$(a, P) \mapsto [a]P = \underbrace{P + \cdots + P}_{a \text{ times}}$	

Elliptic curve points	Very hard homogeneous spaces
Ring $\mathbb{Z}_N$, Group $\mathcal{P}$	Group $\mathcal{G}$, Set $\mathcal{E}$
$[\ ] : \mathbb{Z}_N \times \mathcal{P} \to \mathcal{P}$	$\star : \mathcal{G} \times \mathcal{E} \to \mathcal{E}$
$(a, P) \mapsto [a]P = \underbrace{P + \cdots + P}_{a \text{ times}}$	

Elliptic curve points	Very hard homogeneous spaces
Ring $\mathbb{Z}_N$, Group $\mathcal{P}$	Group $\mathcal{G} = \langle \mathfrak{g} \rangle$, Set $\mathcal{E}$
$[\ ] : \mathbb{Z}_N \times \mathcal{P} \to \mathcal{P}$	$\star : \mathcal{G} \times \mathcal{E} \to \mathcal{E}$
$(a, P) \mapsto [a]P = \underbrace{P + \cdots + P}_{a \text{ times}}$	

Elliptic curve points	Very hard homogeneous spaces
Ring $\mathbb{Z}_N$, Group $\mathcal{P}$	Group $\mathcal{G} = \langle \mathfrak{g} \rangle$, Set $\mathcal{E}$
$[\,] : \mathbb{Z}_N \times \mathcal{P} \to \mathcal{P}$	$\star : \mathcal{G} \times \mathcal{E} \to \mathcal{E}$ $[\,] : \mathbb{Z}_N \times \mathcal{E} \to \mathcal{E}$
$(a, P) \mapsto [a]P = \underbrace{P + \cdots + P}_{a \text{ times}}$	$(a, E) \mapsto [a]E = \mathfrak{g}^a \star E$

Elliptic curve points	Very hard homogeneous spaces
Ring $\mathbb{Z}_N$, Group $\mathcal{P}$	Group $\mathcal{G} = \langle \mathfrak{g} \rangle$, Set $\mathcal{E}$
$[\,] : \mathbb{Z}_N \times \mathcal{P} \to \mathcal{P}$	$\star : \mathcal{G} \times \mathcal{E} \to \mathcal{E}$ $[\,] : \mathbb{Z}_N \times \mathcal{E} \to \mathcal{E}$
$(a, P) \mapsto [a]P = \underbrace{P + \cdots + P}_{a \text{ times}}$	$(a, E) \mapsto [a]E = \mathfrak{g}^a \star E$
$[a]P + [b]P = [a+b]P$ $[a]([b]P) = [ab]P$ $e([a]P, [b]Q) = e(P, Q)^{ab}$	

Elliptic curve points	Very hard homogeneous spaces
Ring $\mathbb{Z}_N$, Group $\mathcal{P}$	Group $\mathcal{G} = \langle \mathfrak{g} \rangle$, Set $\mathcal{E}$
$[\,] : \mathbb{Z}_N \times \mathcal{P} \to \mathcal{P}$	$\star : \mathcal{G} \times \mathcal{E} \to \mathcal{E}$ $[\,] : \mathbb{Z}_N \times \mathcal{E} \to \mathcal{E}$
$(a, P) \mapsto [a]P = \underbrace{P + \cdots + P}_{a \text{ times}}$	$(a, E) \mapsto [a]E = \mathfrak{g}^a \star E$
$[a]P + [b]P = [a+b]P$ $[a]([b]P) = [ab]P$ $e([a]P, [b]Q) = e(P, Q)^{ab}$	No addition! $[a]([b]E) = [a+b]E$ No pairings!

Elliptic curve points	Very hard homogeneous spaces
Ring $\mathbb{Z}_N$, Group $\mathcal{P}$	Group $\mathcal{G} = \langle \mathfrak{g} \rangle$, Set $\mathcal{E}$
$[\,] : \mathbb{Z}_N \times \mathcal{P} \to \mathcal{P}$	$\star : \mathcal{G} \times \mathcal{E} \to \mathcal{E}$ $[\,] : \mathbb{Z}_N \times \mathcal{E} \to \mathcal{E}$
$(a, P) \mapsto [a]P = \underbrace{P + \cdots + P}_{a \text{ times}}$	$(a, E) \mapsto [a]E = \mathfrak{g}^a \star E$
$[a]P + [b]P = [a+b]P$ $[a]([b]P) = [ab]P$ $e([a]P, [b]Q) = e(P, Q)^{ab}$	No addition! $[a]([b]E) = [a+b]E$ No pairings!

Vectorization Problem (DLOG Problem): Given $[a]E$, find a.

Parallelization Problem (CDH Problem): Given $E, [a]E, F$, find $[a]F$.

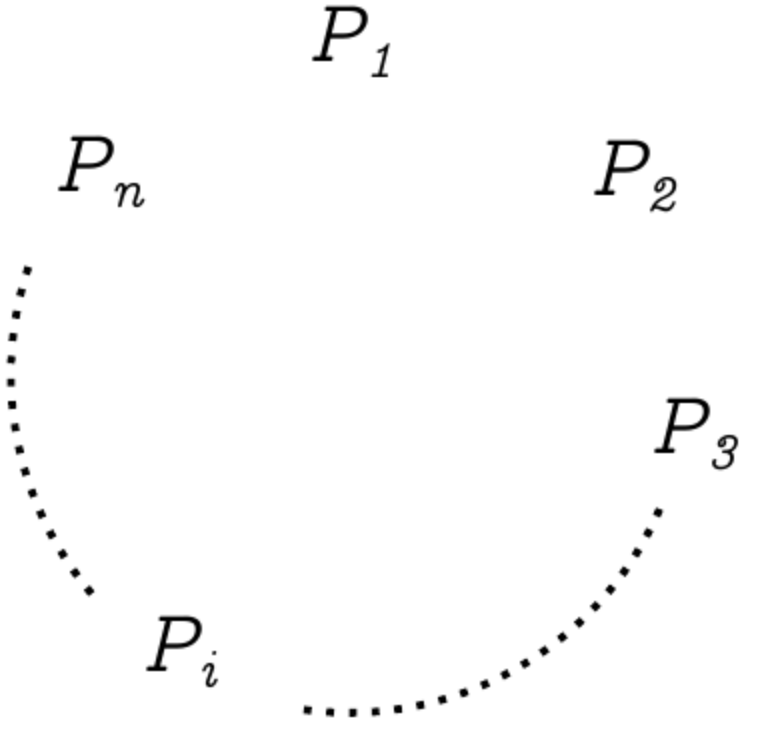
Decisional Parallelization Problem (*DDH Problem*):
Distinguish $([a]E, [b]E, [a+b]E)$ from $([a]E, [b]E, [c]E)$.

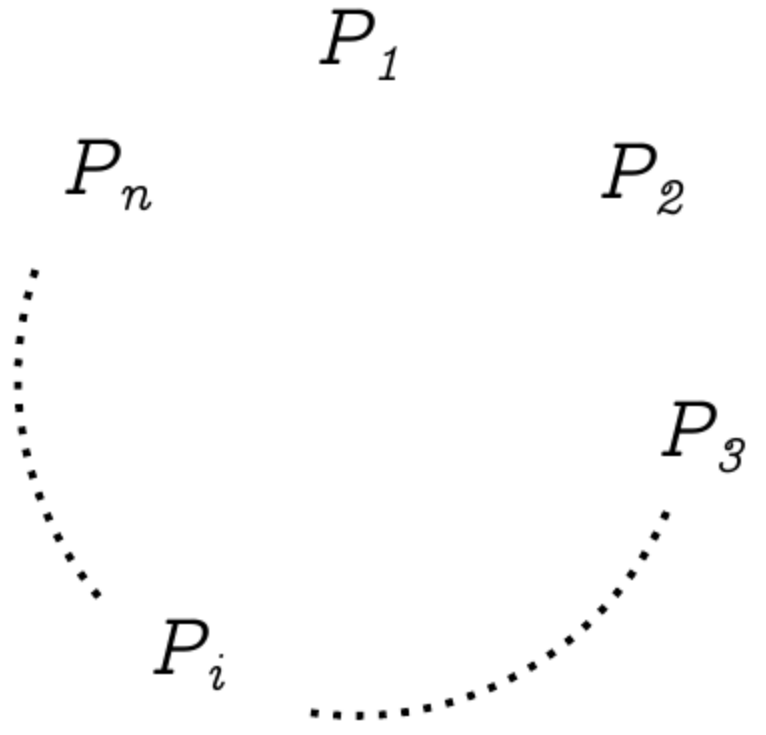
Shamir secret sharing

(Animations)

Shamir secret sharing

- k distinct points in a graph uniquely define a polynomial $f(x)$ of degree $k-1$
- $k-1$ points give no information about any further points of $f(x)$

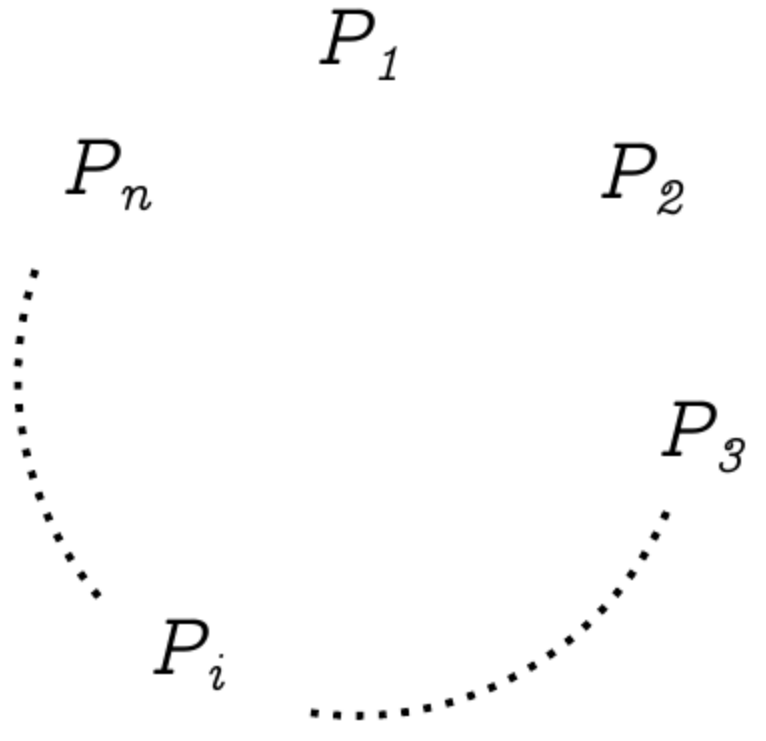

P_1
P_n
P_2
P_3
P_i



Let $\deg f = k - 1$:

For each $S \subseteq \{1, \ldots, n\}$ with $|S| \geq k$,

$$f(0) = \sum_{i \in S} L_i f(i)$$

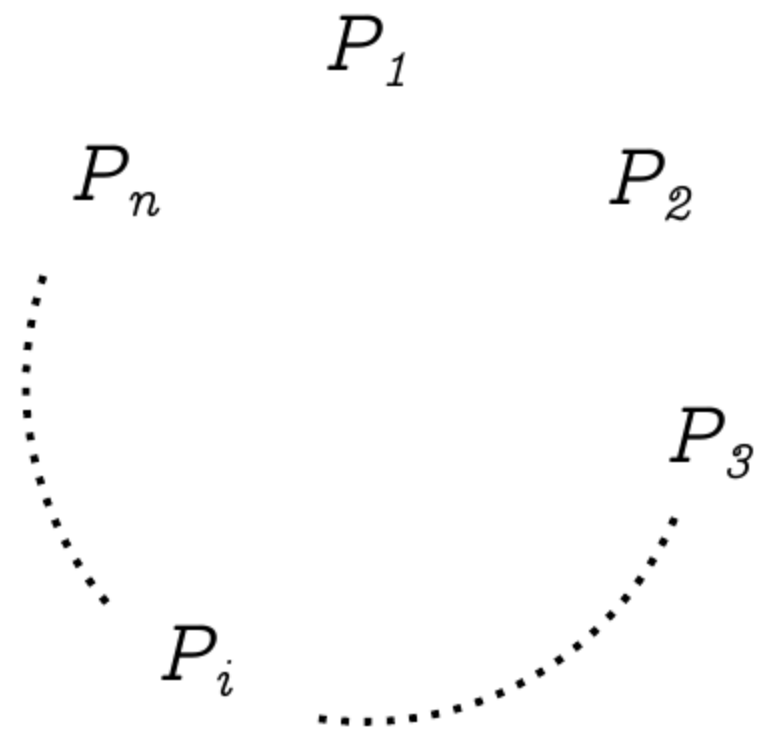


Let $\deg f = k - 1$:

For each $S \subseteq \{1, \ldots, n\}$ with $|S| \geq k$,

$$f(0) = \sum_{i \in S} L_i f(i)$$

Problem: Any set of k malicious players can reconstruct the secret $f(0)$



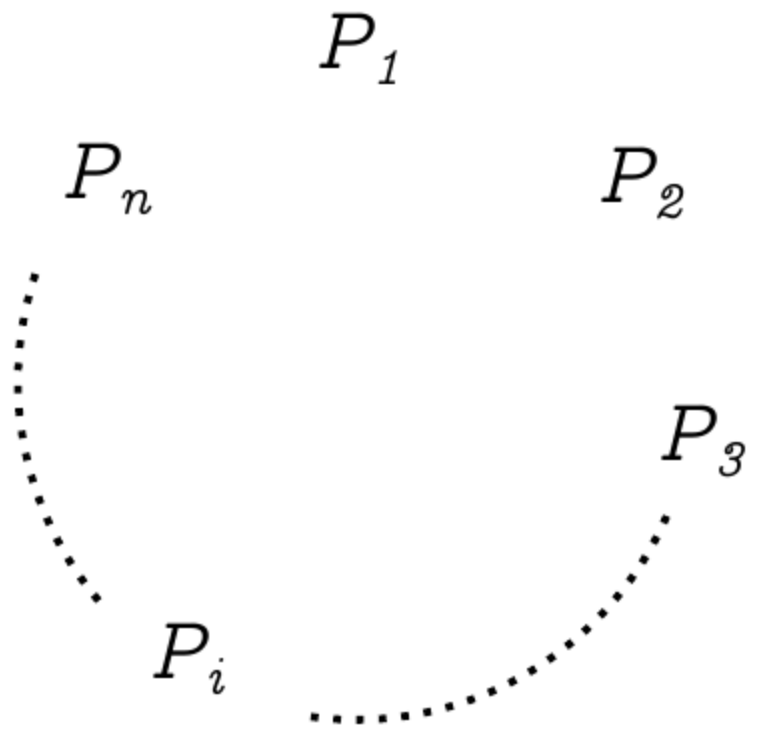
Let $\deg f = k - 1$:

For each $S \subseteq \{1, \ldots, n\}$ with $|S| \geq k$,

$$f(0) = \sum_{i \in S} L_i f(i)$$

Problem: Any set of k malicious players can reconstruct the secret $f(0)$

Assumption 1: There are at most $k - 1$ malicious players



Let $\deg f = k - 1$:

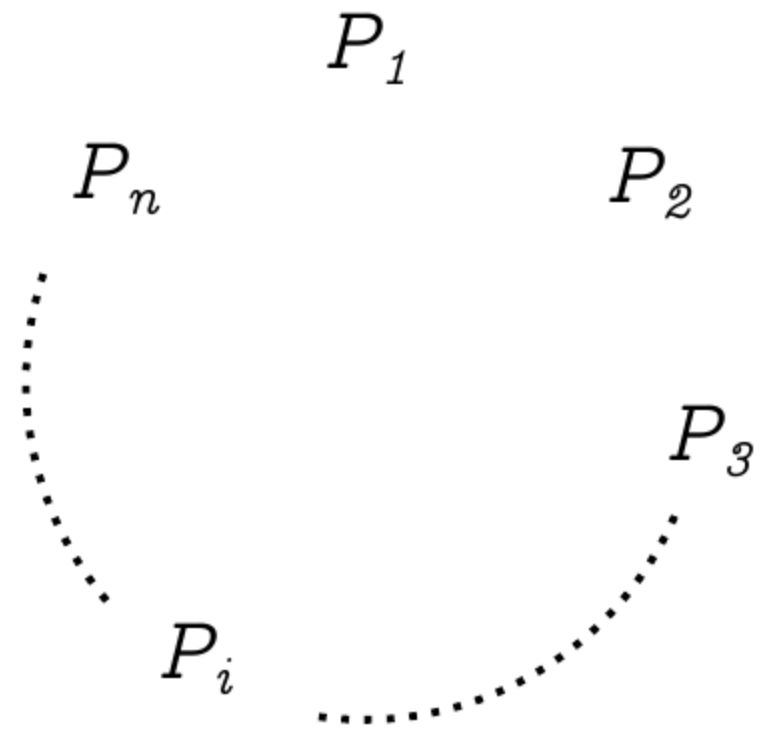
For each $S \subseteq \{1, \ldots, n\}$ with $|S| \geq k$,

$$f(0) = \sum_{i \in S} L_i f(i)$$

Problem: Any set of k malicious players can reconstruct the secret $f(0)$

Assumption 1: There are at most $k - 1$ malicious players

Assumption 2: There are at least k honest players



Let $\deg f = k - 1$:

For each $S \subseteq \{1, \ldots, n\}$ with $|S| \geq k$,

$$f(0) = \sum_{i \in S} L_i f(i)$$

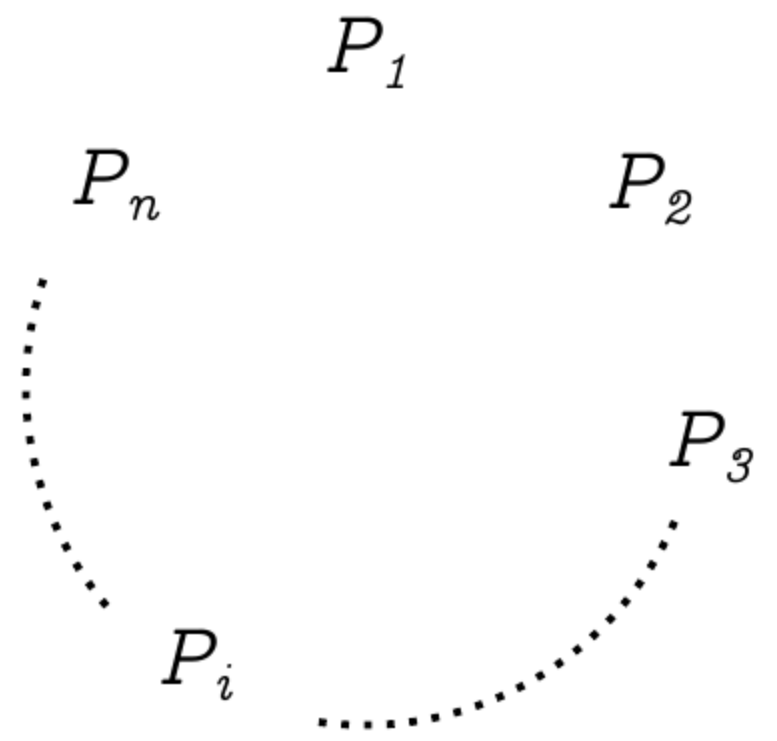
Problem: Any set of k malicious players can reconstruct the secret $f(0)$

Assumption 1: There are at most $k - 1$ malicious players

Assumption 2: There are at least k honest players

[en.wikipedia.org/wiki/File:20140808%EB%8B%A4%EB%AC%B8%ED%99%94%EA%B0%80%EC%A1%B1%EA%B3%BC_%ED%95%A8%EA%BB%98119%EC%84%9C%EC%9A%B8%ED%88%AC%EC%96%B498.jpg]

[commons.wikimedia.org/wiki/File:BTC_Logo.svg]



Let $\deg f = k - 1$:

For each $S \subseteq \{1, \ldots, n\}$ with $|S| \geq k$,

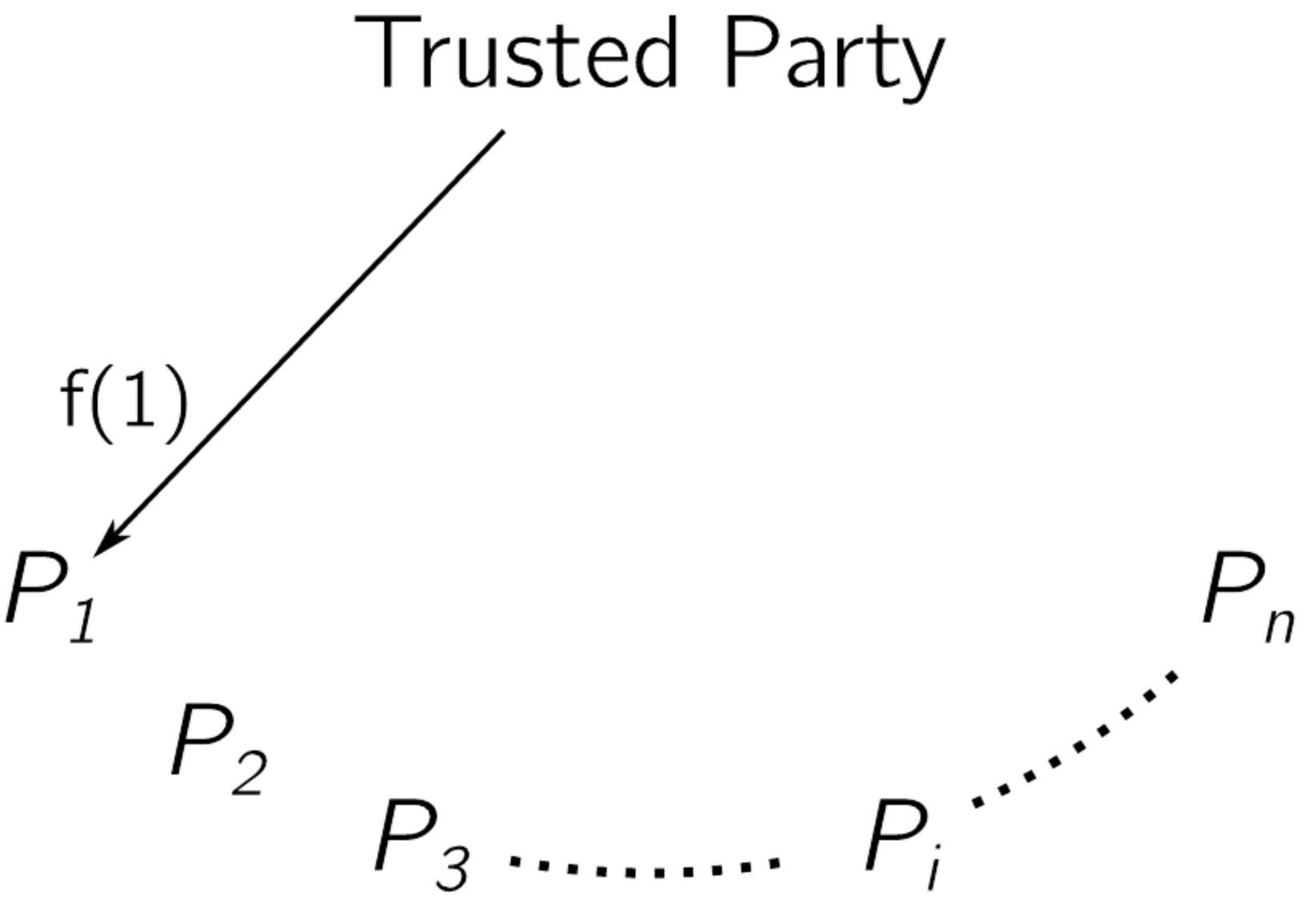
$$f(0) = \sum_{i \in S} L_i f(i)$$

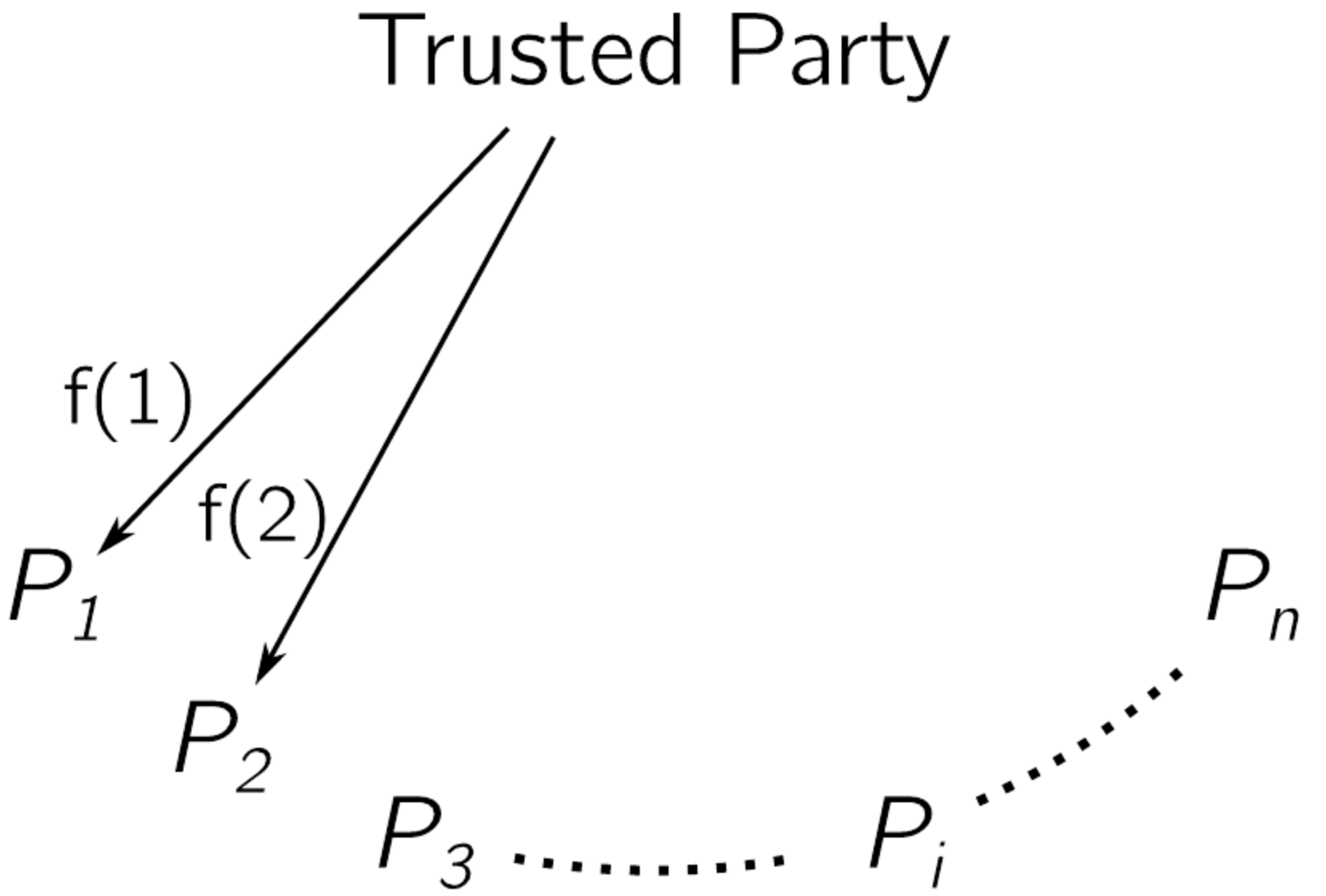
Problem: Any set of k malicious players can reconstruct the secret $f(0)$

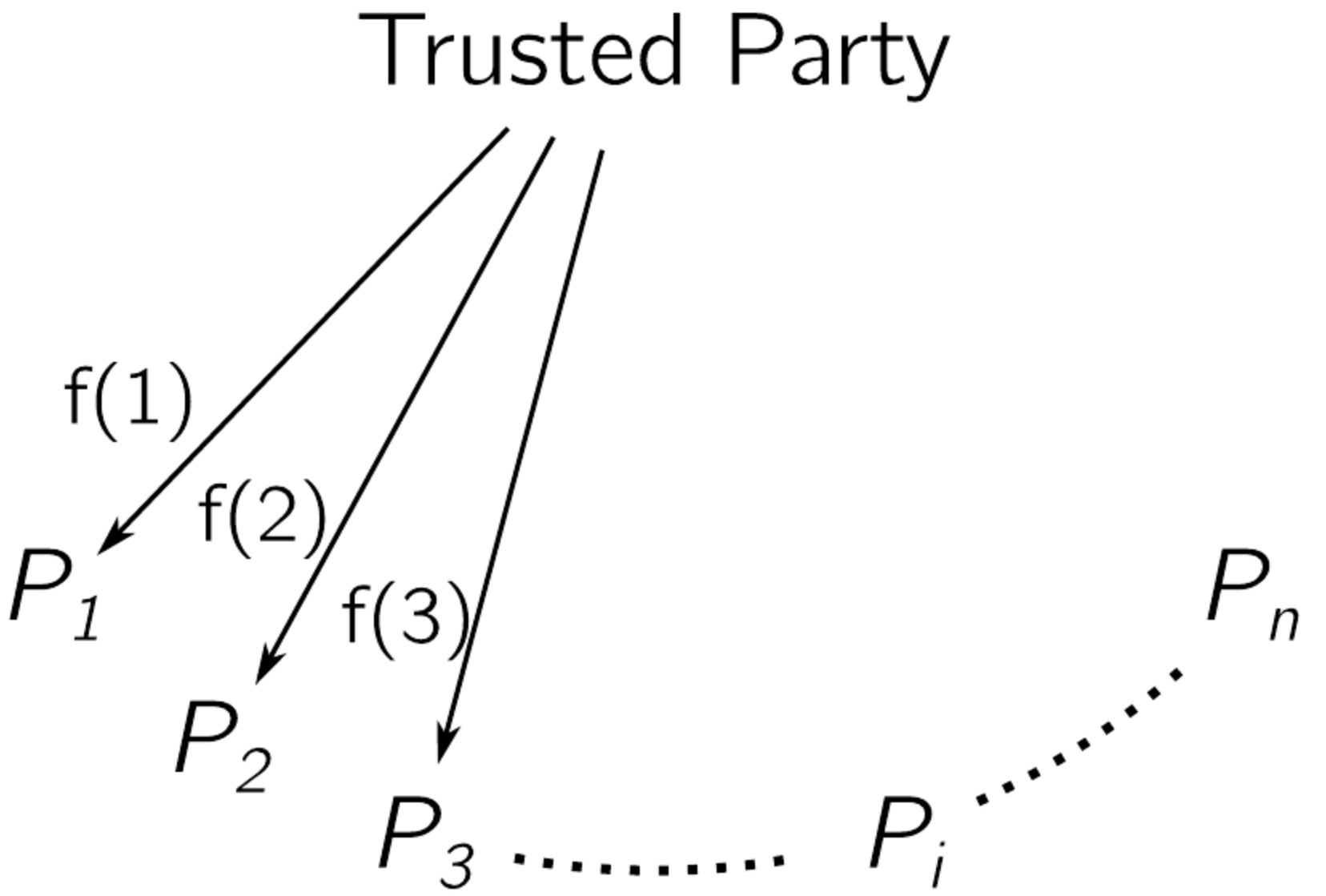
Assumption 1: There are at most $k - 1$ malicious players
Assumption 2: There are at least k honest players

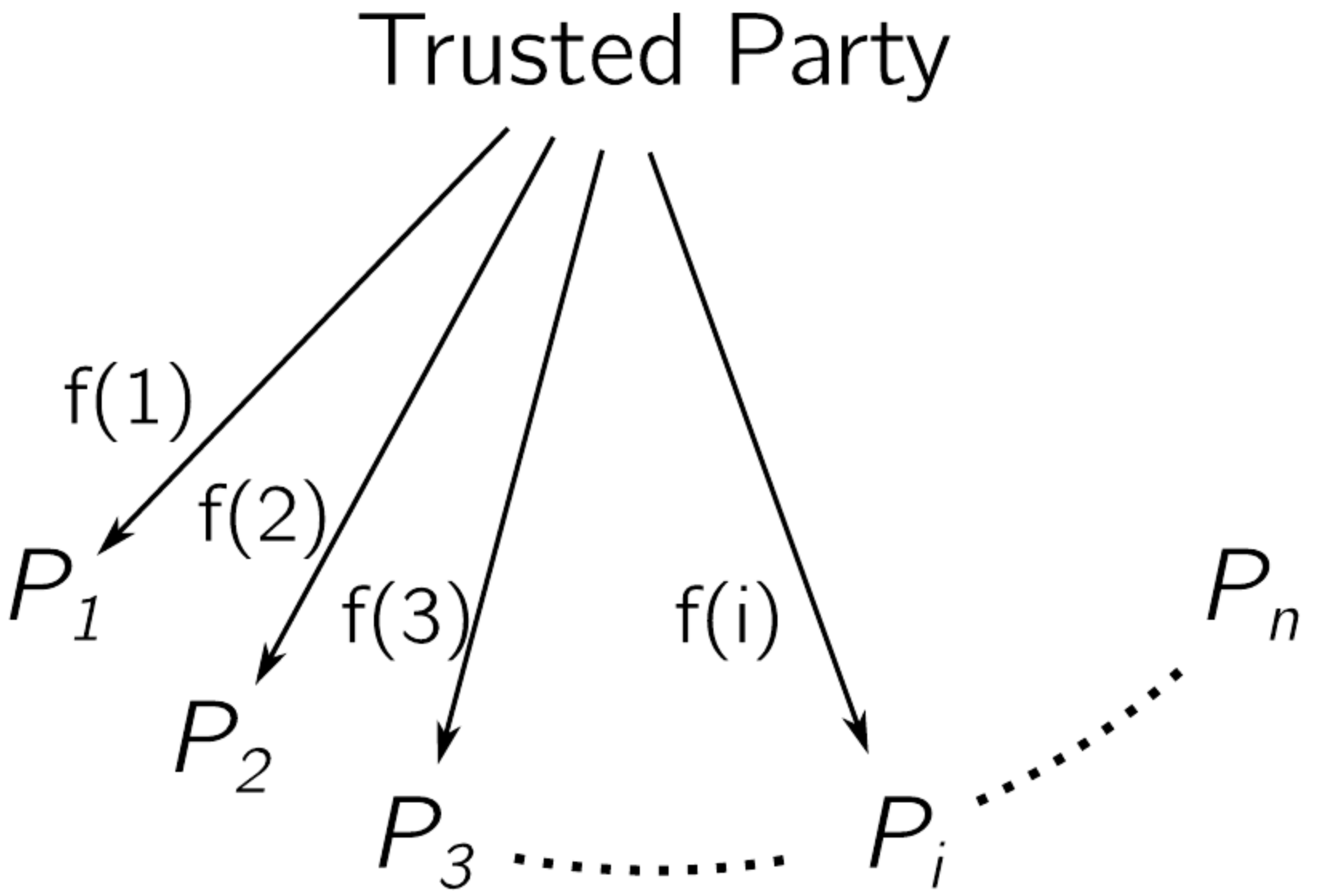
honest majority (k,n)-threshold scheme

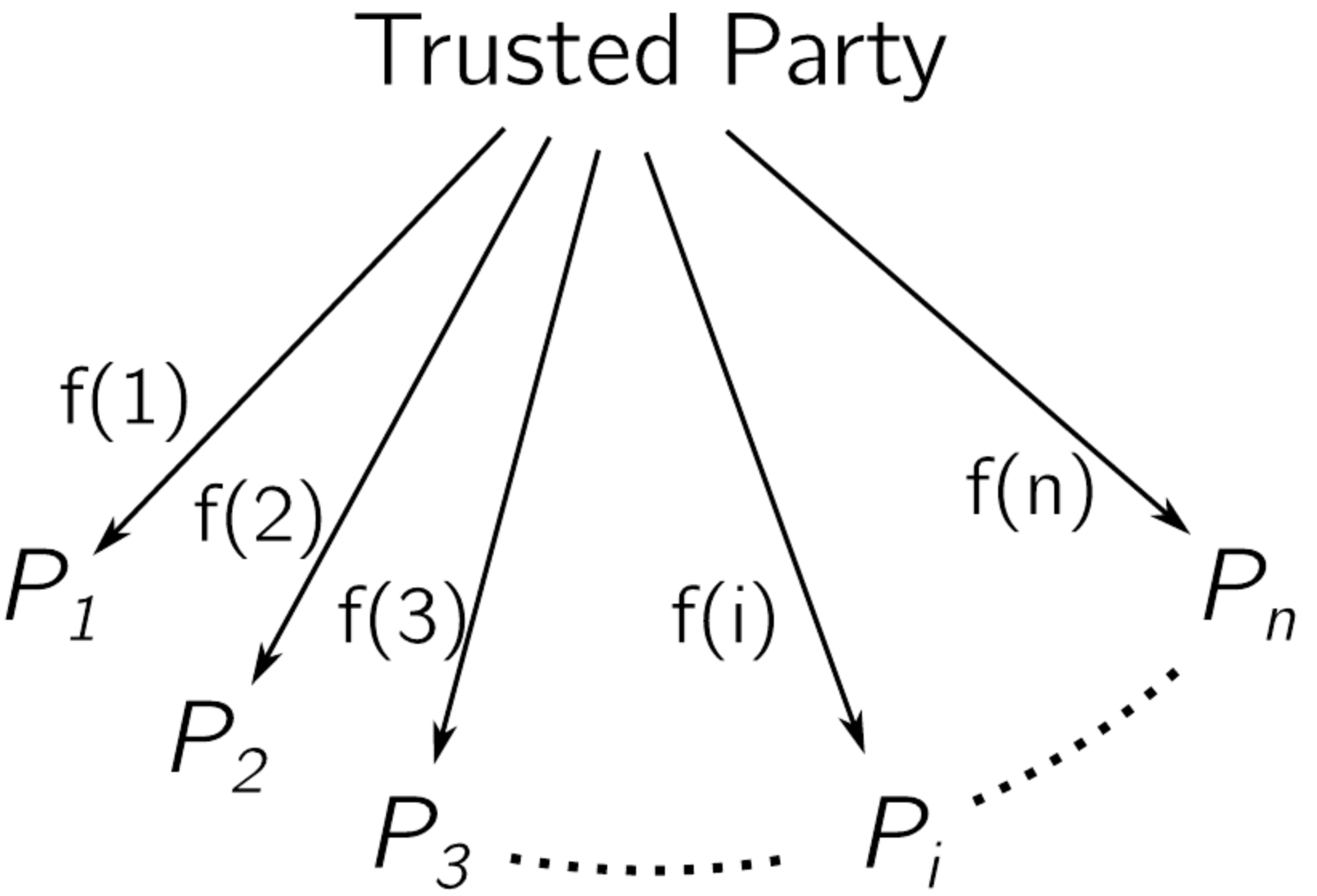
Trusted Party

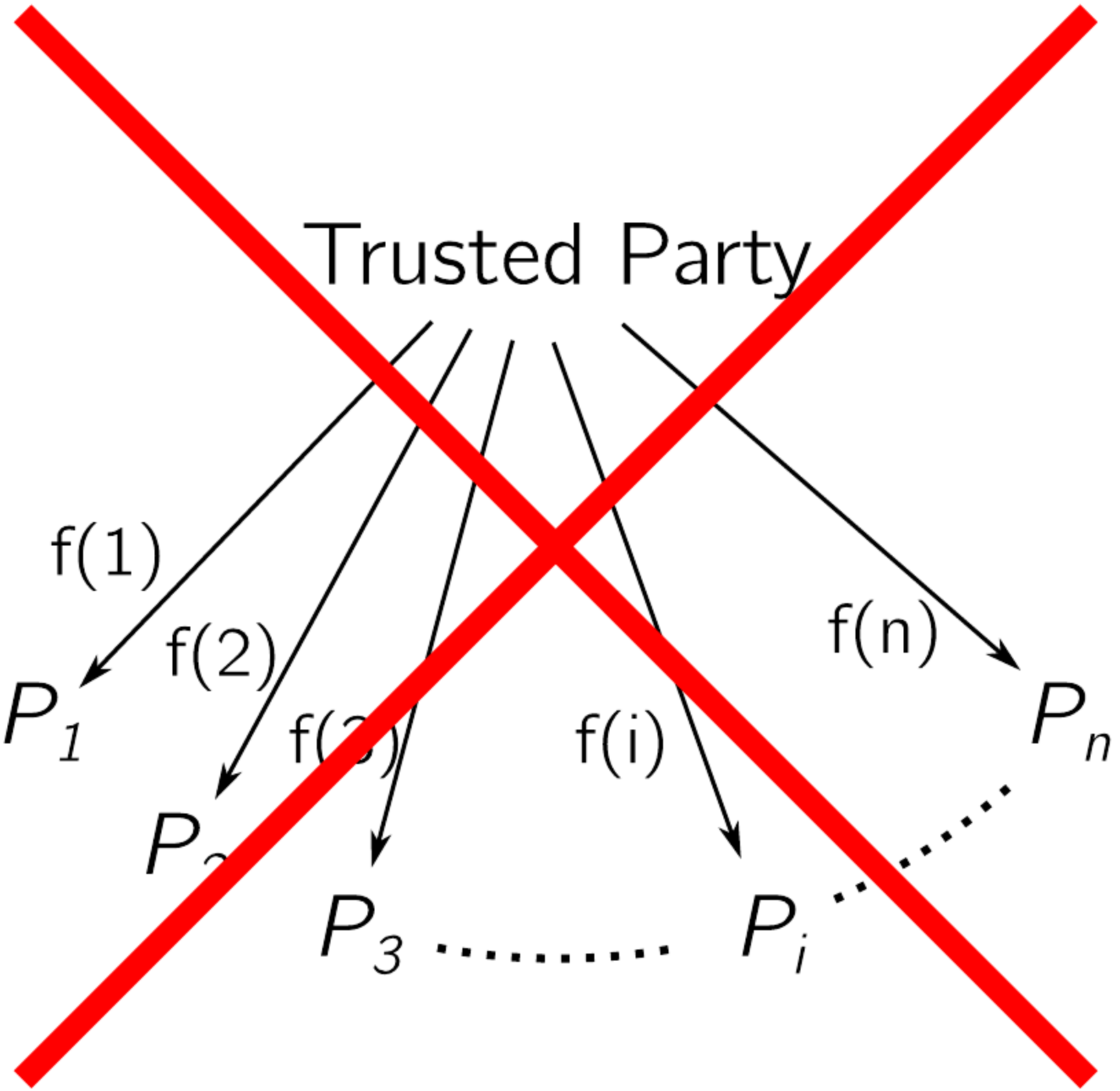

Trusted Party
f(1)
P_1
P_2
P_3
P_i
P_n


Trusted Party
f(1)
f(2)
P_1
P_2
P_3
P_i
P_n


Trusted Party
f(1)
f(2)
f(3)
P_1
P_2
P_3
P_i
P_n


Trusted Party
f(1)
f(2)
f(3)
f(i)
P_1
P_2
P_3
P_i
P_n


Trusted Party
f(1)
f(2)
f(3)
f(i)
f(n)
P_1
P_2
P_3
P_i
P_n


Trusted Party
f(1)
f(2)
f(3)
f(i)
f(n)
P_1
P_2
P_3
P_i
P_n

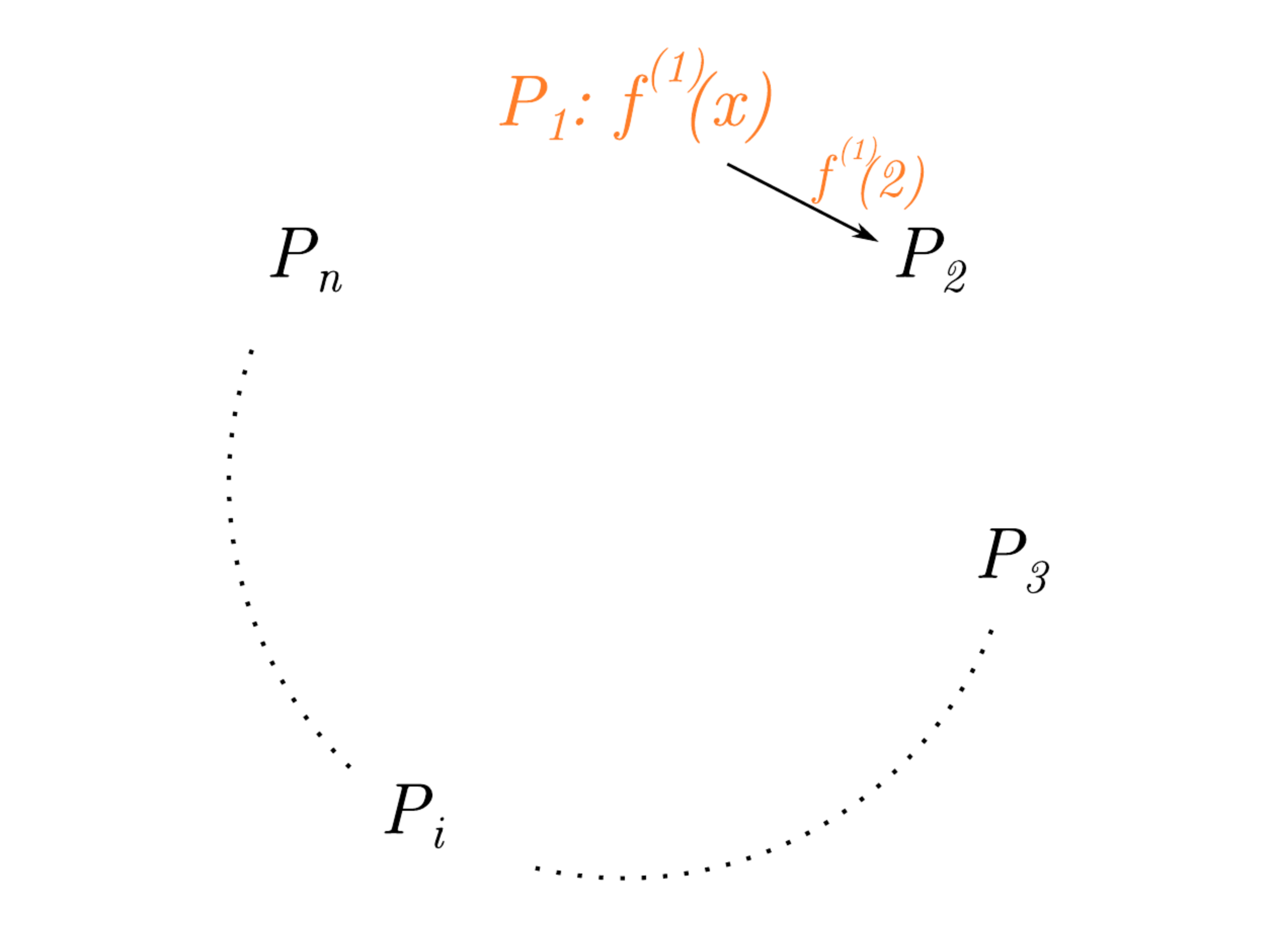
P_1: $f^{(1)}(x)$

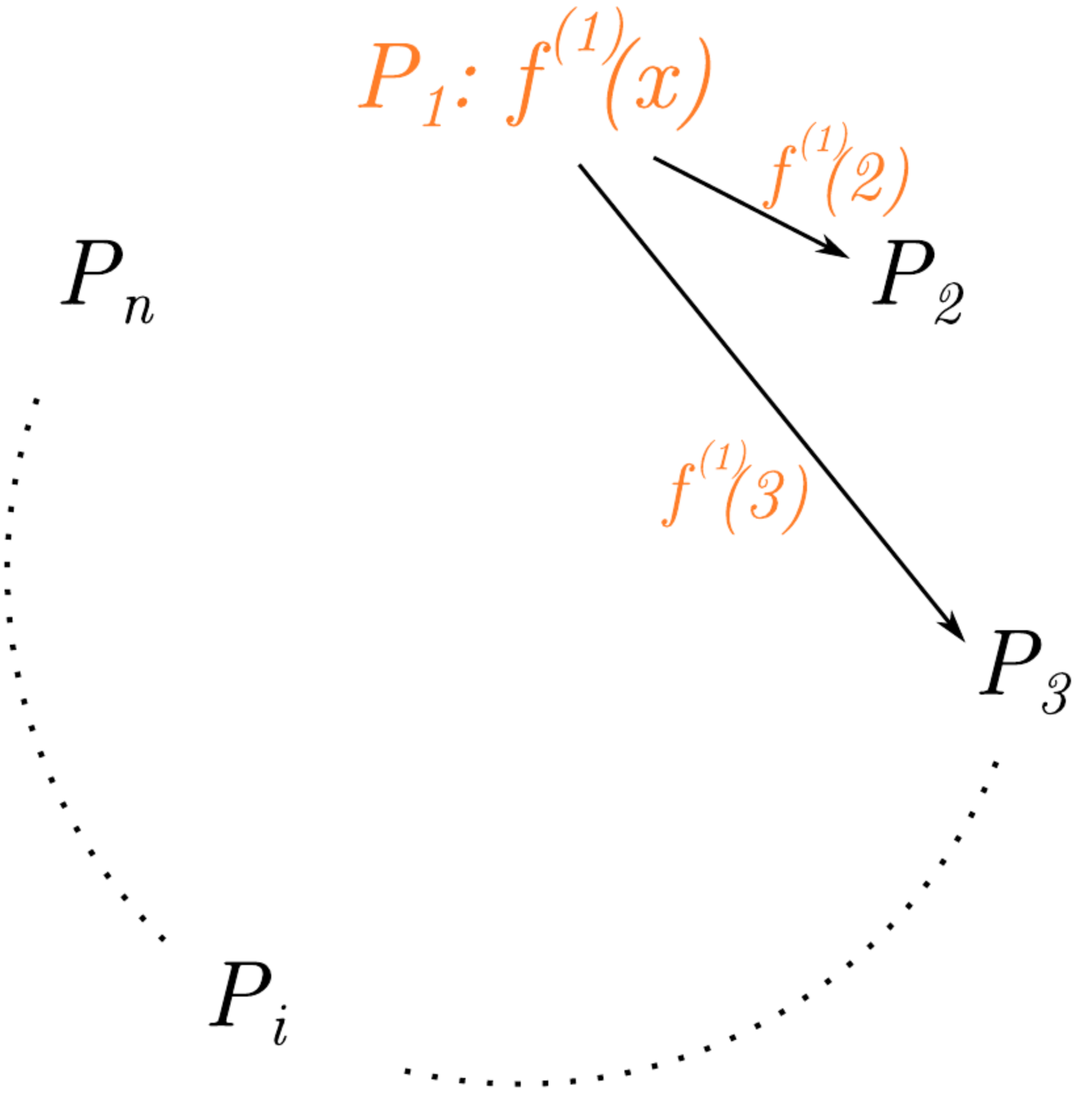
P_n

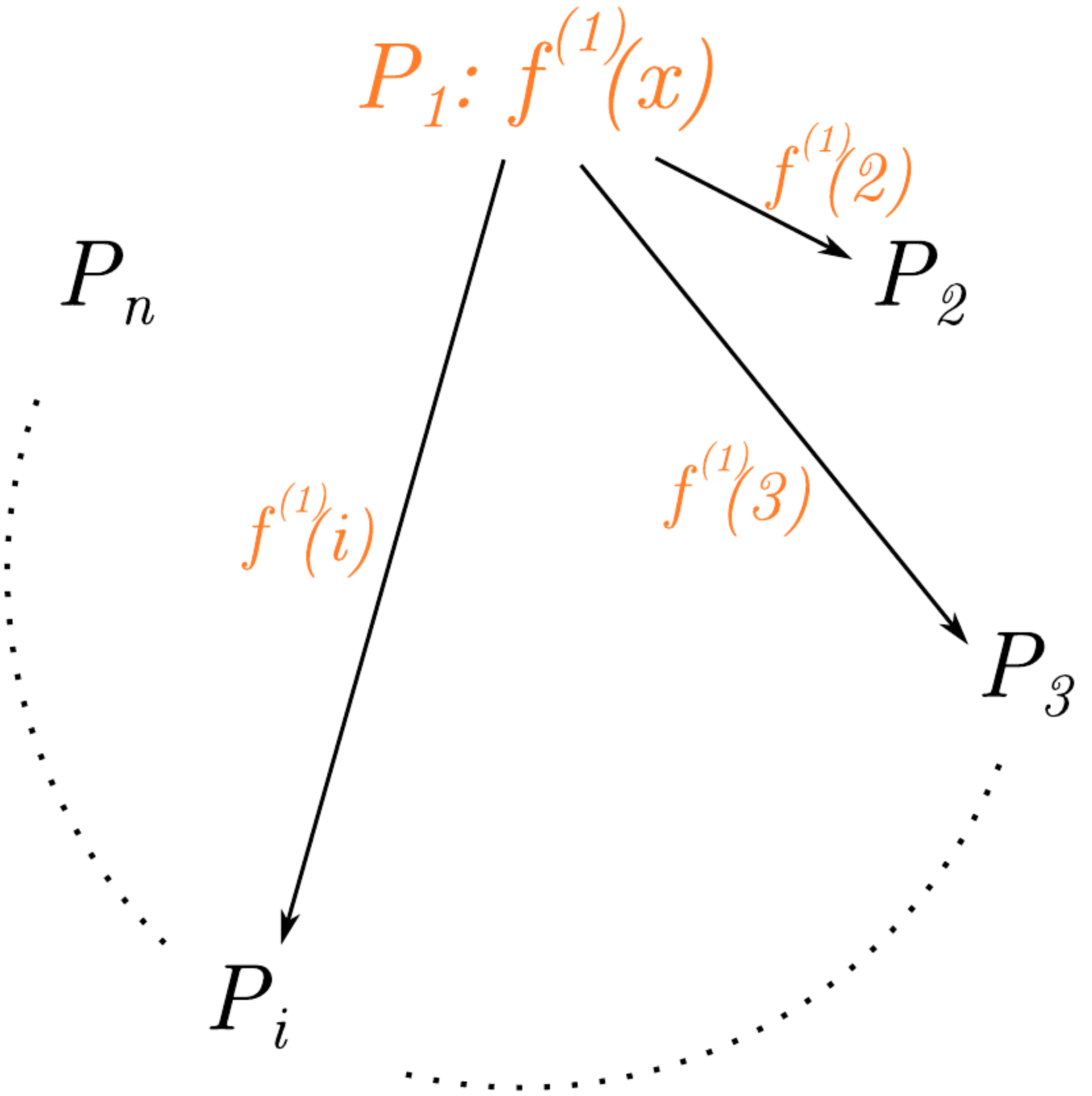
P_2

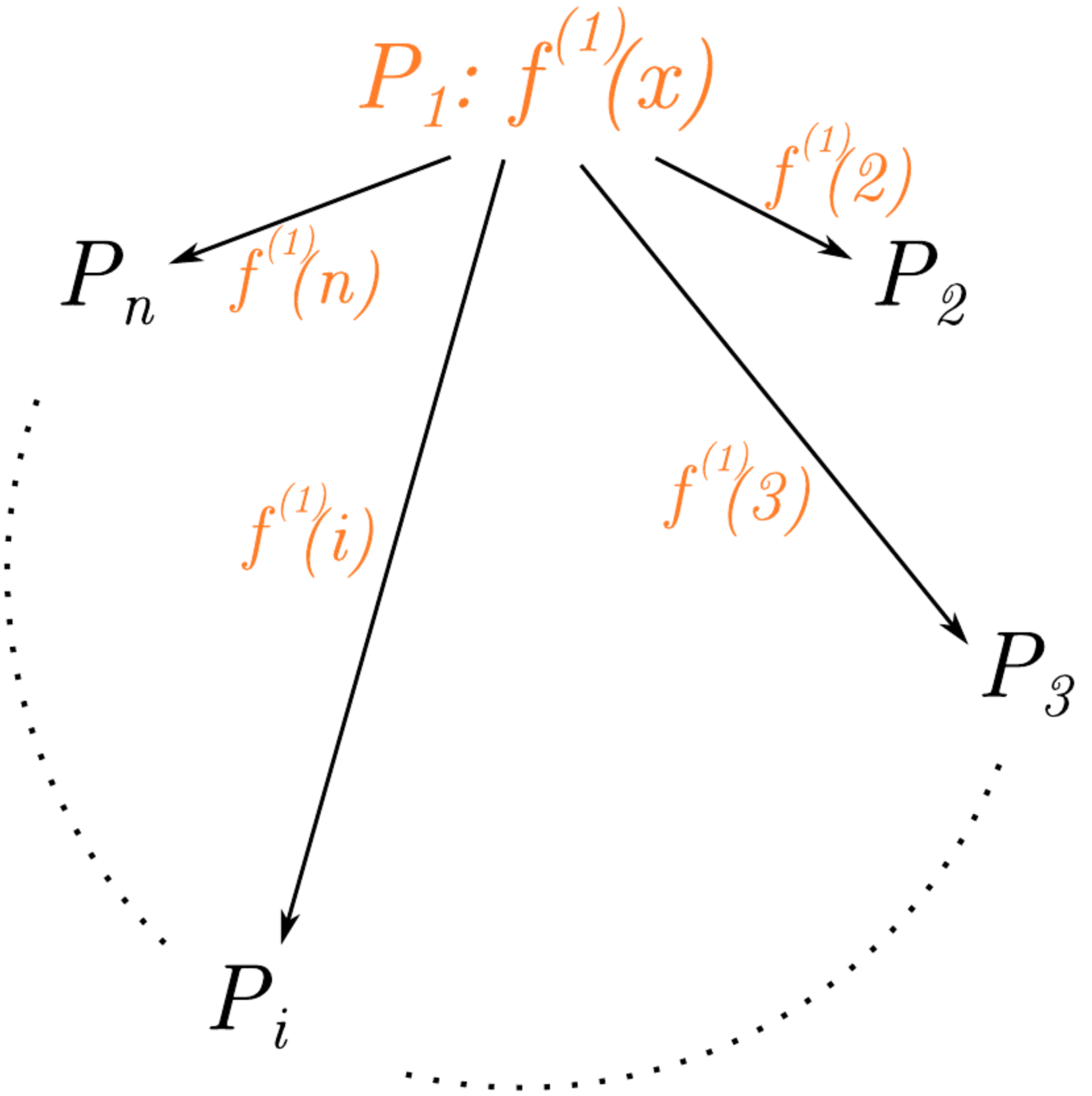
P_3

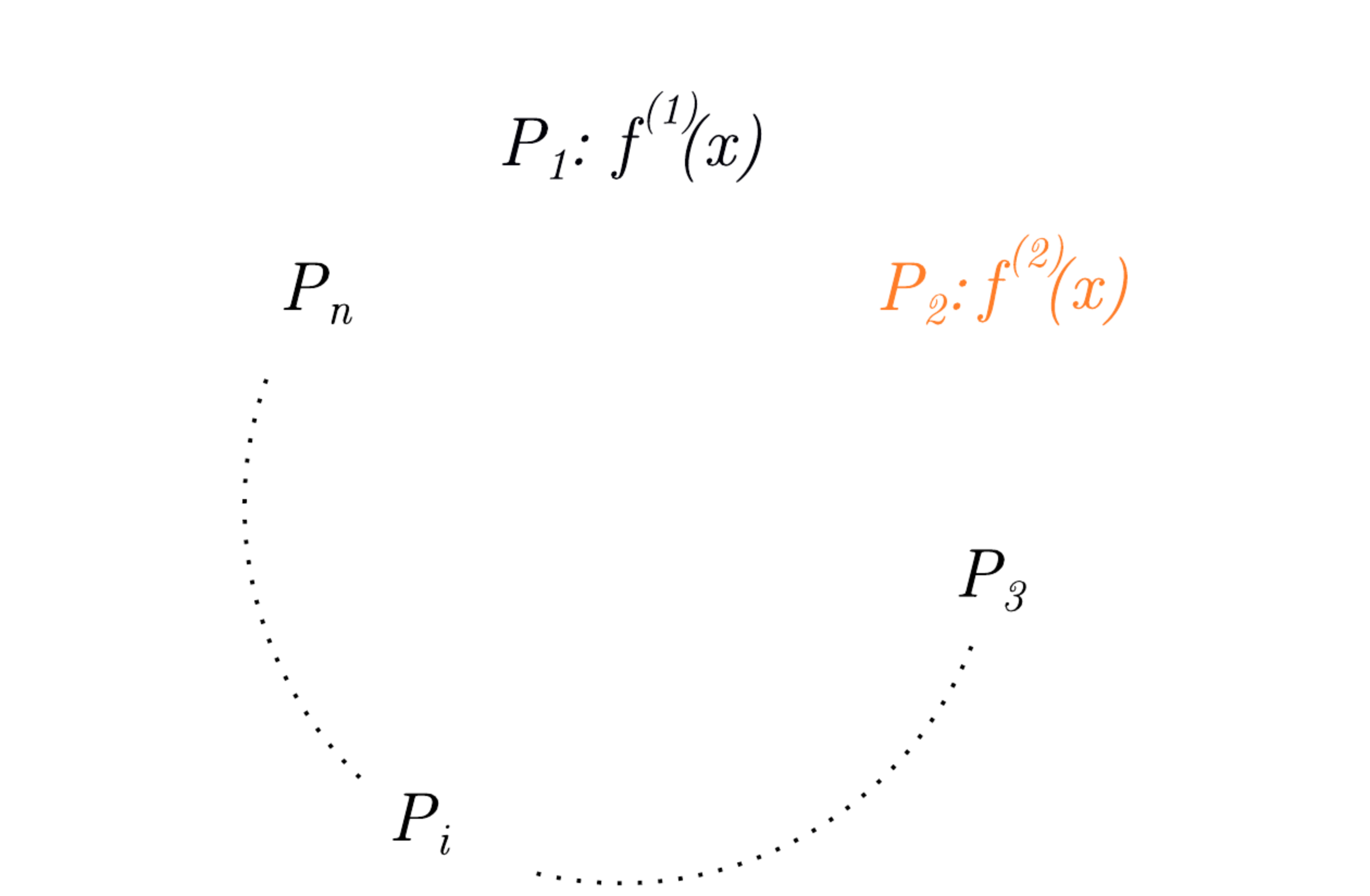
P_i

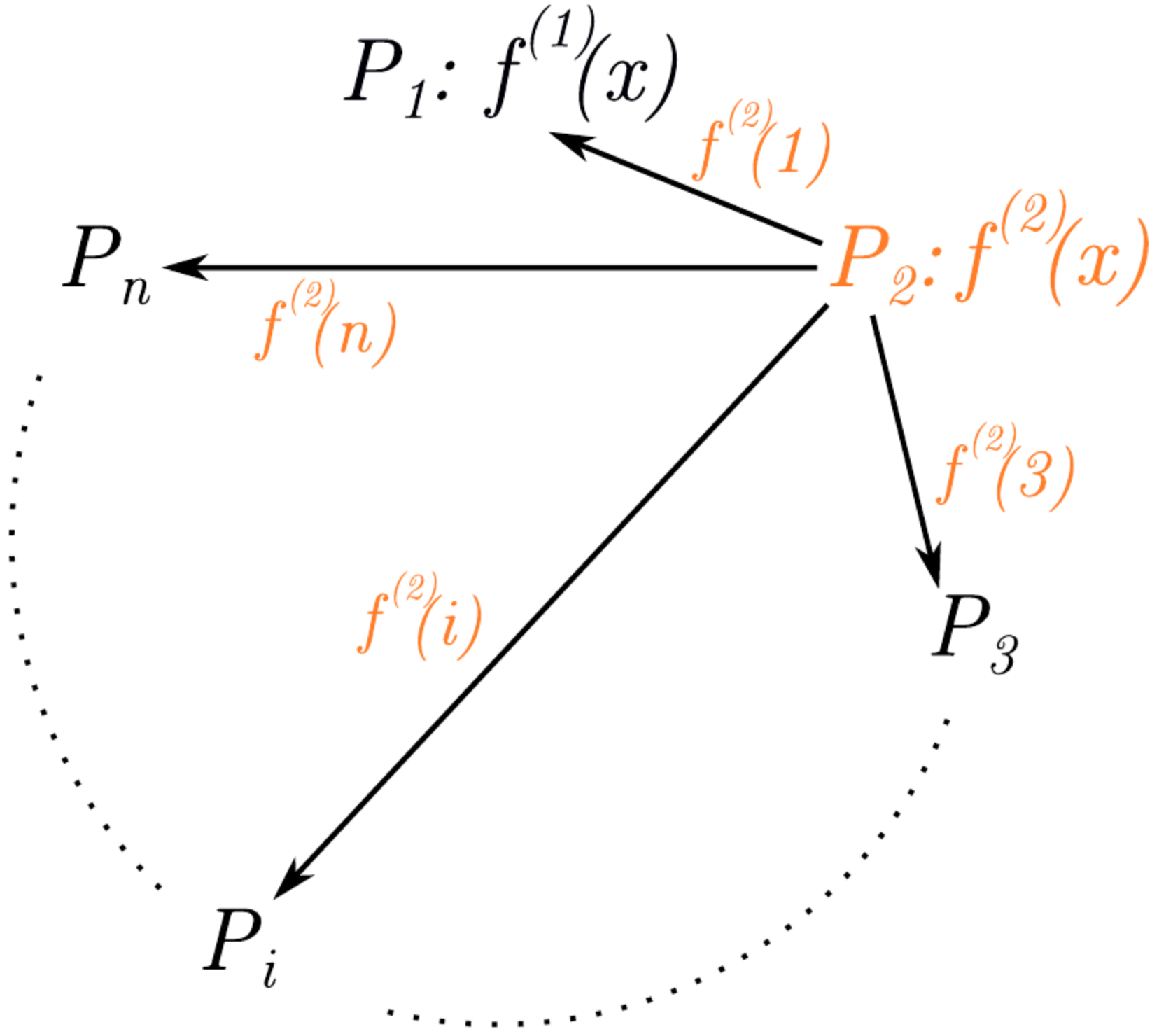

P_1: $f^{(1)}(x)$
$f^{(1)}(2)$
P_2
P_3
P_n
P_i

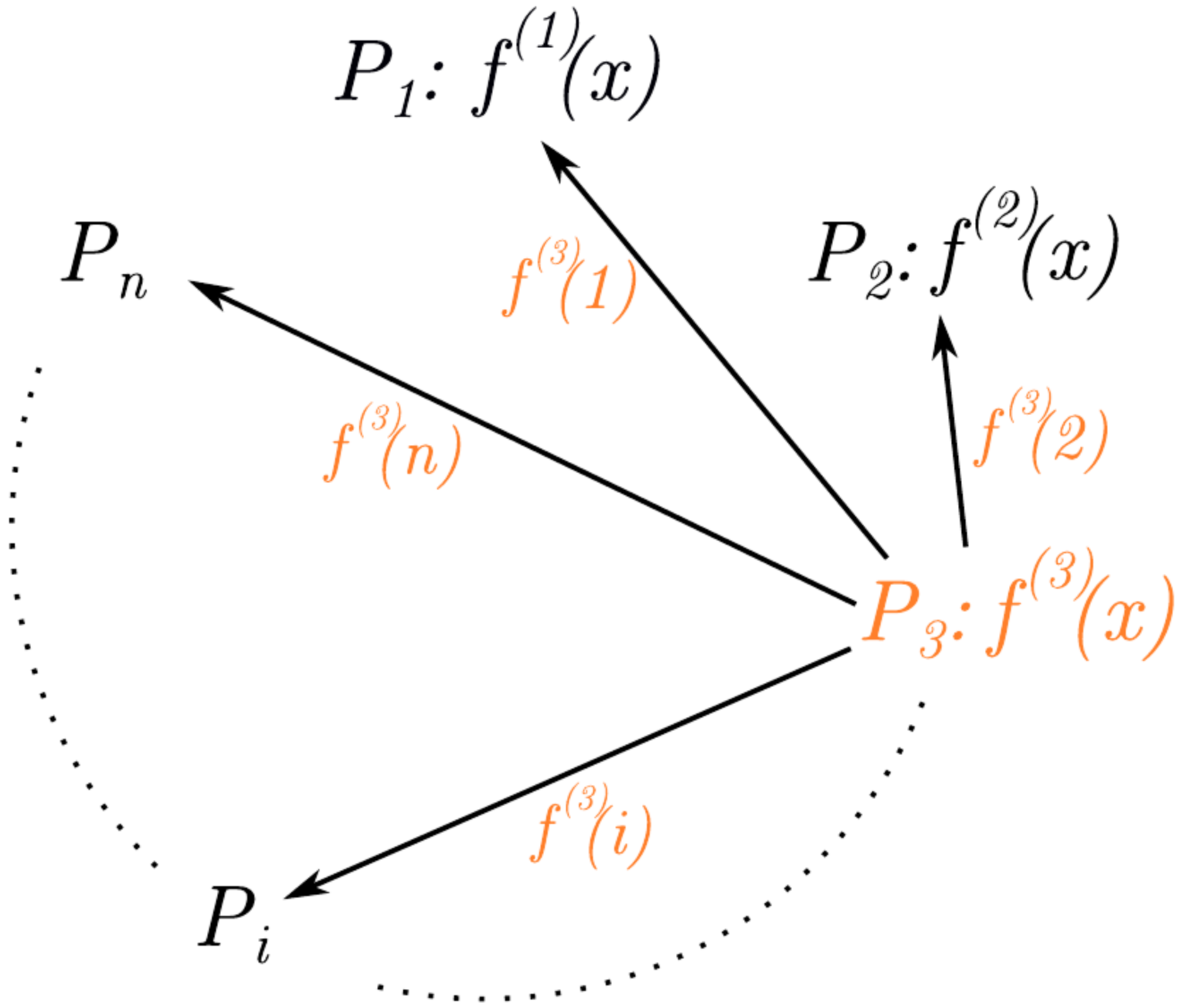

P_1: $f^{(1)}(x)$
$f^{(1)}(2)$
$f^{(1)}(3)$
P_2
P_3
P_n
P_i

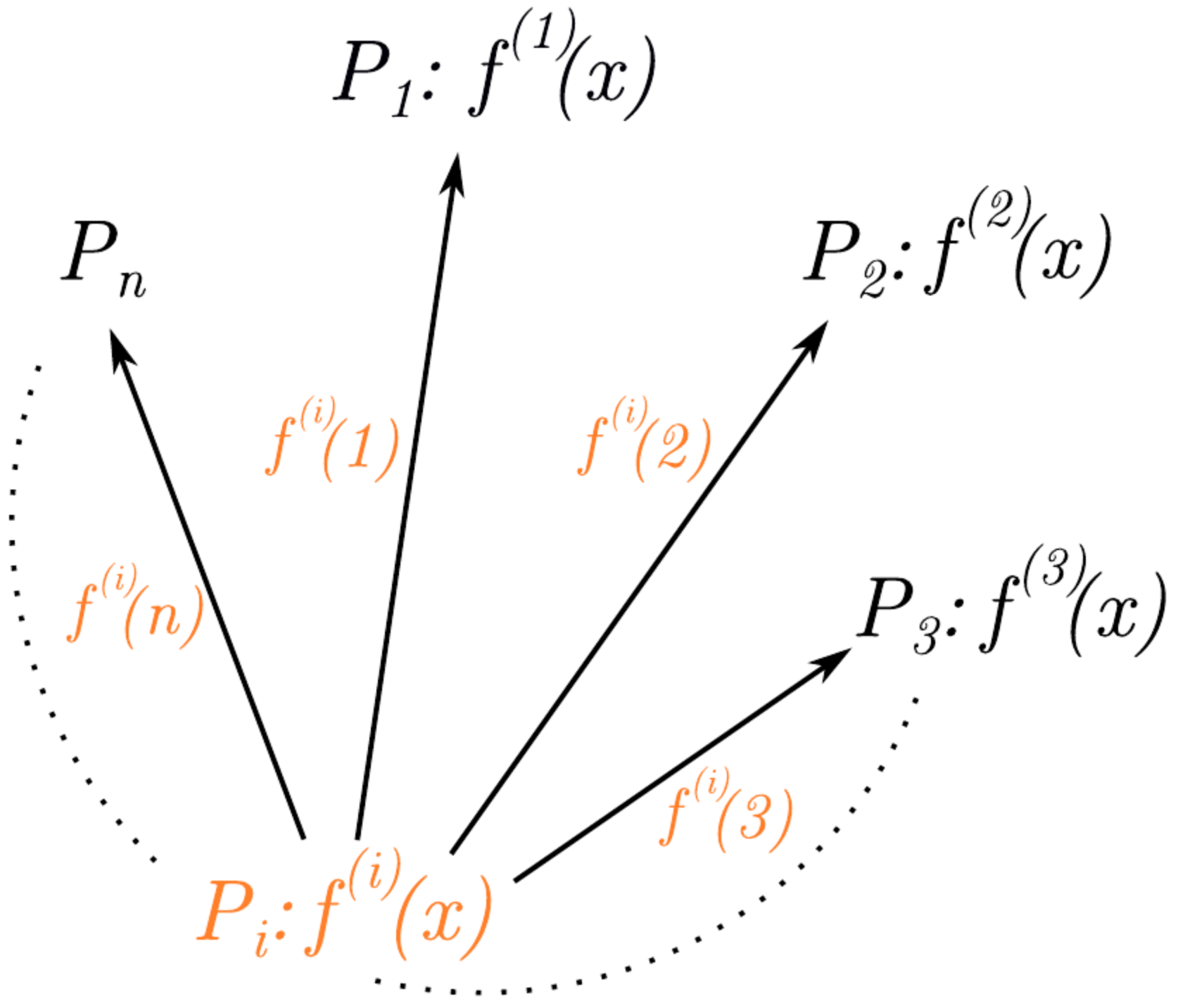

P_1: $f^{(1)}(x)$
$f^{(1)}(2)$
P_2
P_n
$f^{(1)}(i)$
$f^{(1)}(3)$
P_3
P_i

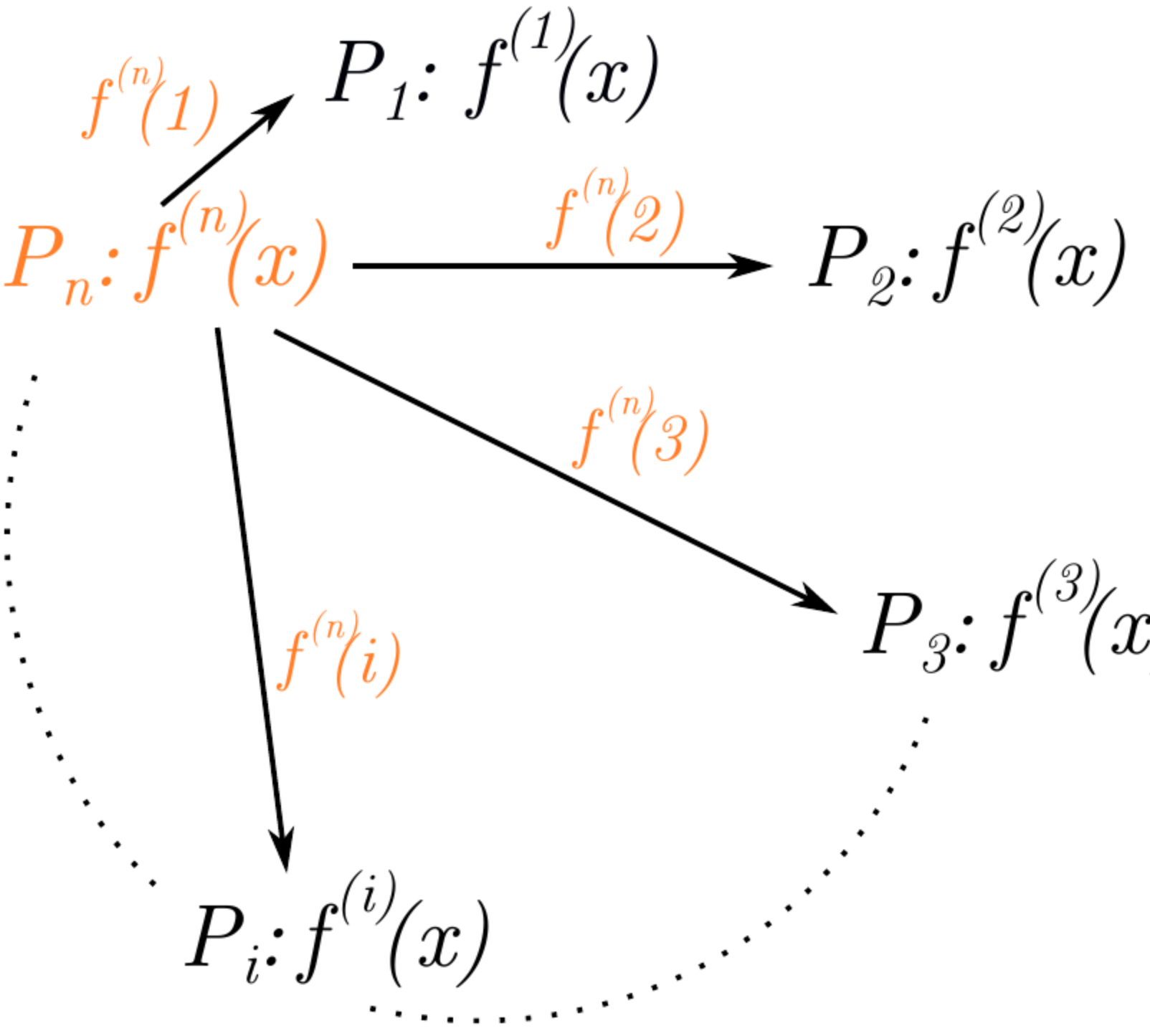

P_1: $f^{(1)}(x)$
$f^{(1)}(2)$
P_n
$f^{(1)}(n)$
P_2
$f^{(1)}(i)$
$f^{(1)}(3)$
P_3
P_i


P_1: $f^{(1)}(x)$
P_2: $f^{(2)}(x)$
P_3
P_i
P_n


P_1: $f^{(1)}(x)$
$f^{(2)}(1)$
P_n
$f^{(2)}(n)$
P_2: $f^{(2)}(x)$
$f^{(2)}(3)$
$f^{(2)}(i)$
P_3
P_i


P_1: $f^{(1)}(x)$
P_2: $f^{(2)}(x)$
P_3: $f^{(3)}(x)$
P_n
P_i
$f^{(3)}(1)$
$f^{(3)}(2)$
$f^{(3)}(n)$
$f^{(3)}(i)$


P_1: $f^{(1)}(x)$
P_n
P_2: $f^{(2)}(x)$
$f^{(i)}(1)$
$f^{(i)}(2)$
$f^{(i)}(n)$
P_3: $f^{(3)}(x)$
$f^{(i)}(3)$
P_i: $f^{(i)}(x)$


$P_1: f^{(1)}(x)$
$f^{(n)}(1)$
$P_n: f^{(n)}(x)$
$f^{(n)}(2)$
$P_2: f^{(2)}(x)$
$f^{(n)}(3)$
$P_3: f^{(3)}(x)$
$f^{(n)}(i)$
$P_i: f^{(i)}(x)$

P_1 holds $f^{(1)}(1), \ldots, f^{(n)}(1)$
P_2 holds $f^{(1)}(2), \ldots, f^{(n)}(2)$
P_3 holds $f^{(1)}(3), \ldots, f^{(n)}(3)$
$\vdots$
P_i holds $f^{(1)}(i), \ldots, f^{(n)}(i)$
$\vdots$
P_n holds $f^{(1)}(n), \ldots, f^{(n)}(n)$

Define $f(x) := \sum_{i=1}^{n} f^{(i)}(x)$

P_1 holds $f^{(1)}(1), \ldots, f^{(n)}(1)$
P_2 holds $f^{(1)}(2), \ldots, f^{(n)}(2)$
P_3 holds $f^{(1)}(3), \ldots, f^{(n)}(3)$
$\vdots$
P_i holds $f^{(1)}(i), \ldots, f^{(n)}(i)$
$\vdots$
P_n holds $f^{(1)}(n), \ldots, f^{(n)}(n)$

(Animations)

$$\text{Define} \quad f(x) := \sum_{i=1}^{n} f^{(i)}(x)$$

$$\begin{aligned}
P_1 \text{ holds } f^{(1)}(1), \ldots, f^{(n)}(1) &\Rightarrow f(1) = f^{(1)}(1) + \cdots + f^{(n)}(1) \\
P_2 \text{ holds } f^{(1)}(2), \ldots, f^{(n)}(2) &\Rightarrow f(2) = f^{(1)}(2) + \cdots + f^{(n)}(2) \\
P_3 \text{ holds } f^{(1)}(3), \ldots, f^{(n)}(3) &\Rightarrow f(3) = f^{(1)}(3) + \cdots + f^{(n)}(3) \\
\vdots & \qquad\qquad \vdots \\
P_i \text{ holds } f^{(1)}(i), \ldots, f^{(n)}(i) &\Rightarrow f(i) = f^{(1)}(i) + \cdots + f^{(n)}(i) \\
\vdots & \qquad\qquad \vdots \\
P_n \text{ holds } f^{(1)}(n), \ldots, f^{(n)}(n) &\Rightarrow f(n) = f^{(1)}(n) + \cdots + f^{(n)}(n)
\end{aligned}$$

Player 1 proves correctness of shares of $f^{(1)}(x) = \sum_{i=0}^{k-1} f_i x^i$ as follows:

Player 1 proves correctness of shares of $f^{(1)}(x) = \sum_{i=0}^{k-1} f_i x^i$ as follows:

Elliptic curve points

Let $\langle P \rangle = \mathcal{P}$.

1. Player 1 commits to $Q_i = [f_i]P$ for $i = 1, \ldots, k-1$.
2. Player j can verify if

$$[f^{(1)}(j)]P = \left[\sum_{i=0}^{k-1} f_i j^i\right] P \stackrel{?}{=} \sum_{i=0}^{k-1} [j^i] Q_i$$

Player 1 proves correctness of shares of $f^{(1)}(x) = \sum_{i=0}^{k-1} f_i x^i$ as follows:

Elliptic curve points

Let $\langle P \rangle = \mathcal{P}$.

1. Player 1 commits to $Q_i = [f_i]P$ for $i = 1, \ldots, k-1$.
2. Player j can verify if

$$[f^{(1)}(j)]P = \left[\sum_{i=0}^{k-1} f_i j^i\right] P \stackrel{?}{=} \sum_{i=0}^{k-1} [j^i] Q_i$$

Very hard homogeneous spaces

- No addition! ($[a]E + [b]E$ not defined)
- Only $[a]([b]E) = [a+b]E$

Piecewise verifiable proofs

Player 1 proves correctness of shares of $f^{(1)}(x) = \sum_{i=0}^{k-1} f_i x^i$ as follows:

Elliptic curve points

Let $\langle P \rangle = \mathcal{P}$.

1. Player 1 commits to $Q_i = [f_i]P$ for $i = 1, \ldots, k-1$.
2. Player j can verify if

$$[f^{(1)}(j)]P = \left[\sum_{i=0}^{k-1} f_i j^i\right] P \stackrel{?}{=} \sum_{i=0}^{k-1} [j^i] Q_i$$

Very hard homogeneous spaces

- No addition! ($[a]E + [b]E$ not defined)
- Only $[a]([b]E) = [a+b]E$

$$\Rightarrow \text{Rather } [f^{(1)}(j)]E \text{ than } [f_i]E!$$

$$\mathrm{f}^{(1)}(0) \qquad \mathrm{f}^{(1)}(2) \qquad \mathrm{f}^{(1)}(3) \qquad \cdots \qquad \mathrm{f}^{(1)}(\mathrm{i}) \qquad \cdots \qquad \mathrm{f}^{(1)}(\mathrm{n})$$

$$[f^{(1)}(0)]E \quad [f^{(1)}(2)]E \quad [f^{(1)}(3)]E \quad \cdots \quad [f^{(1)}(i)]E \quad \cdots \quad [f^{(1)}(n)]E$$

$$[f^{(1)}(0)]E \quad [f^{(1)}(2)]E \quad [f^{(1)}(3)]E \quad \cdots \quad [f^{(1)}(i)]E \quad \cdots \quad [f^{(1)}(n)]E$$

Commit to:

Challenge:

Response:

Verification:
(for i=0,..,n)

$$[f^{(1)}(0)]E \quad [f^{(1)}(2)]E \quad [f^{(1)}(3)]E \quad \cdots \quad [f^{(1)}(i)]E \quad \cdots \quad [f^{(1)}(n)]E$$

For $j = 1,\ldots,\lambda$ generate polynomial $b_j(x)$

Commit to: $[b_j(0)]E \quad [b_j(2)]E \quad [b_j(3)]E \quad \cdots \quad [b_j(i)]E \quad \cdots \quad [b_j(n)]E$

Challenge:

Response:

Verification:
(for i=0,..,n)

$$[f^{(1)}(0)]E \quad [f^{(1)}(2)]E \quad [f^{(1)}(3)]E \quad \cdots \quad [f^{(1)}(i)]E \quad \cdots \quad [f^{(1)}(n)]E$$

For $j = 1,\ldots,\lambda$ generate polynomial $b_j(x)$

Commit to: $[b_j(0)]E \quad [b_j(2)]E \quad [b_j(3)]E \quad \cdots \quad [b_j(i)]E \quad \cdots \quad [b_j(n)]E$

Challenge: $c_j \leftarrow \{0,1\}$

Response:

Verification:
(for i=0,..,n)

$$[f^{(1)}(0)]E \quad [f^{(1)}(2)]E \quad [f^{(1)}(3)]E \quad \cdots \quad [f^{(1)}(i)]E \quad \cdots \quad [f^{(1)}(n)]E$$

For $j = 1,\ldots,\lambda$ generate polynomial $b_j(x)$

Commit to: $[b_j(0)]E \quad [b_j(2)]E \quad [b_j(3)]E \quad \cdots \quad [b_j(i)]E \quad \cdots \quad [b_j(n)]E$

Challenge: $c_j \leftarrow \{0,1\}$

Response: $r_j(x) = b_j(x) - c_j f^{(1)}(x)$

Verification:
(for i=0,..,n)

$$[f^{(1)}(0)]E \quad [f^{(1)}(2)]E \quad [f^{(1)}(3)]E \quad \cdots \quad [f^{(1)}(i)]E \quad \cdots \quad [f^{(1)}(n)]E$$

For $j = 1,\ldots,\lambda$ generate polynomial $b_j(x)$

Commit to:	$[b_j(0)]E \quad [b_j(2)]E \quad [b_j(3)]E \quad \cdots \quad [b_j(i)]E \quad \cdots \quad [b_j(n)]E$
Challenge:	$c_j \leftarrow \{0,1\}$
Response:	$r_j(x) = b_j(x) - c_j f^{(1)}(x)$
Verification: (for i=0,..,n)	$[r_j(i)][c_j f^{(1)}(i)]E) = [b_j(i)]E$

$$[f^{(1)}(0)]E \quad [f^{(1)}(2)]E \quad [f^{(1)}(3)]E \quad \cdots \quad [f^{(1)}(i)]E \quad \cdots \quad [f^{(1)}(n)]E$$

For $j = 1,\ldots,\lambda$ generate polynomial $b_j(x)$

Commit to: $[b_j(0)]E \quad [b_j(2)]E \quad [b_j(3)]E \quad \cdots \quad [b_j(i)]E \quad \cdots \quad [b_j(n)]E$

Challenge: $c_j \leftarrow \{0,1\}$

Complexity: $O(\lambda n^2)$ group actions

Response: $r_j(x) = b_j(x) - c_j f^{(1)}(x)$

Verification: $[r_j(i)][c_j f^{(1)}(i)]E) = [b_j(i)]E$
(for i=0,..,n)

$$[f^{(1)}(0)]E \quad [f^{(1)}(2)]E \quad [f^{(1)}(3)]E \quad \cdots \quad [f^{(1)}(i)]E \quad \cdots \quad [f^{(1)}(n)]E$$

For $j = 1,\ldots,\lambda$ generate polynomial $b_j(x)$

Commit to: $[b_j(0)]E \quad [b_j(2)]E \quad [b_j(3)]E \quad \cdots \quad [b_j(i)]E \quad \cdots \quad [b_j(n)]E$

Challenge: $c_j \leftarrow \{0,1\}$

Response: $r_j(x) = b_j(x) - c_j f^{(1)}(x)$

Verification: $[r_j(i)][c_j f^{(1)}(i)]E) = [b_j(i)]E$
(for i=0,..,n)

Complexity: $O(\lambda n^2)$ group actions

Commitment scheme

C: $\{0,1\}^* \times \{0,1\}^\lambda \longrightarrow \{0,1\}^{2\lambda}$

$$[f^{(1)}(0)]E \quad [f^{(1)}(2)]E \quad [f^{(1)}(3)]E \quad \cdots \quad [f^{(1)}(i)]E \quad \cdots \quad [f^{(1)}(n)]E$$

For $j = 1,\ldots,\lambda$ generate polynomial $b_j(x)$

Commit to: $[b_j(0)]E \quad C(b_j(2),y_2) \quad C(b_j(3),y_3) \quad \cdots \quad C(b_j(i),y_i) \quad \cdots \quad C(b_j(n),y_n)$

Challenge: $c_j \leftarrow \{0,1\}$

Response: $r_j(x) = b_j(x) - c_j f^{(1)}(x)$

Verification: $[r_j(i)][c_j f^{(1)}(i)]E) = [b_j(i)]E$
(for i=0,..,n)

~~Complexity: $O(\lambda n^2)$ group actions~~

Commitment scheme

C: $\{0,1\}^* \times \{0,1\}^\lambda \longrightarrow \{0,1\}^{2\lambda}$

$$[f^{(1)}(0)]E \quad [f^{(1)}(2)]E \quad [f^{(1)}(3)]E \quad \cdots \quad [f^{(1)}(i)]E \quad \cdots \quad [f^{(1)}(n)]E$$

For $j = 1,\ldots,\lambda$ generate polynomial $b_j(x)$

Commit to: $[b_j(0)]E \quad C(b_j(2),y_2) \quad C(b_j(3),y_3) \quad \cdots \quad C(b_j(i),y_i) \quad \cdots \quad C(b_j(n),y_n)$

Challenge: $c_j \leftarrow \{0,1\}$

Response: $r_j(x) = b_j(x) - c_j f^{(1)}(x)$

Verification: $[r_j(0)][c_j f^{(1)}(0)]E) = [b_j(0)]E$

$$[f^{(1)}(0)]E \quad f^{(1)}(2) \quad f^{(1)}(3) \quad \cdots \quad f^{(1)}(i) \quad \cdots \quad f^{(1)}(n)$$

For $j = 1,\ldots,\lambda$ generate polynomial $b_j(x)$

Commit to:	$[b_j(0)]E \quad C(b_j(2),y_2) \quad C(b_j(3),y_3) \quad \cdots \quad C(b_j(i),y_i) \quad \cdots \quad C(b_j(n),y_n)$
Challenge:	$c_j \leftarrow \{0,1\}$
Response:	$r_j(x) = b_j(x) - c_j f^{(1)}(x)$
Verification:	$[r_j(0)][c_j f^{(1)}(0)]E) = [b_j(0)]E$

		P_2	P_3		P_i		P_n
		$\uparrow$	$\uparrow$		$\uparrow$		$\uparrow$
$[f^{(1)}(0)]E$	$f^{(1)}(2)$		$f^{(1)}(3)$	$\cdots$	$f^{(1)}(i)$	$\cdots$	$f^{(1)}(n)$

For $j = 1,\ldots,\lambda$ generate polynomial $b_j(x)$

Commit to:	$[b_j(0)]E$	$C(b_j(2),y_2)$	$C(b_j(3),y_3)$	$\cdots$	$C(b_j(i),y_i)$	$\cdots$	$C(b_j(n),y_n)$

Challenge: $c_j \leftarrow \{0,1\}$

Response: $r_j(x) = b_j(x) - c_j f^{(1)}(x)$

Verification: $[r_j(0)][c_j f^{(1)}(0)]E) = [b_j(0)]E$

	P_2	P_3		P_i		P_n
$[f^{(1)}(0)]E$	$f^{(1)}(2),y_2$	$f^{(1)}(3),y_3$	$\cdots$	$f^{(1)}(i),y_i$	$\cdots$	$f^{(1)}(n),y_n$

For $j = 1,\ldots,\lambda$ generate polynomial $b_j(x)$

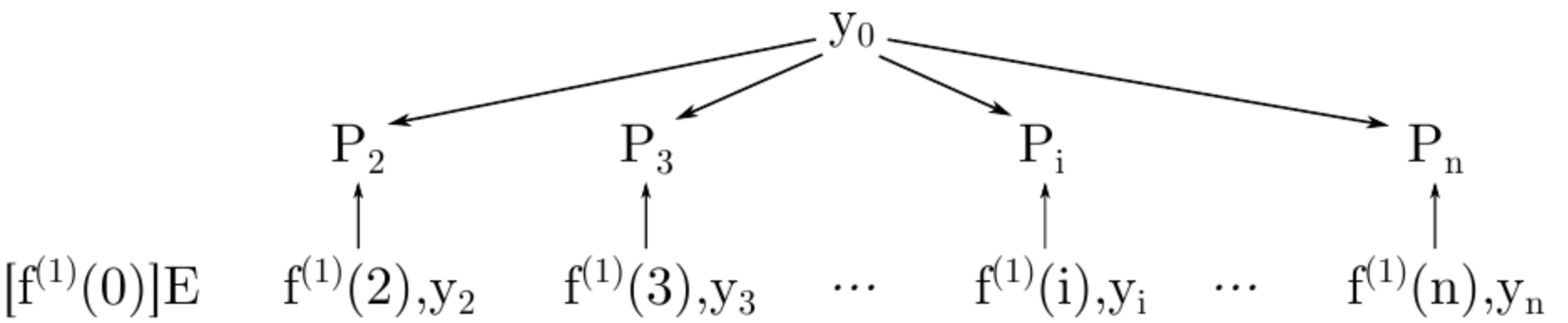
Commit to:	$[b_j(0)]E$	$C(b_j(2),y_2)$	$C(b_j(3),y_3)$	$\cdots$	$C(b_j(i),y_i)$	$\cdots$ $C(b_j(n),y_n)$

Challenge: $c_j \leftarrow \{0,1\}$

Response: $r_j(x) = b_j(x) - c_j f^{(1)}(x)$

Verification: $[r_j(0)][c_j f^{(1)}(0)]E) = [b_j(0)]E$

by P_i $C(r_j(i)+c_j f^{(1)}(i),y_i) = C(b_j(i),y_i)$



For $j = 1,...,\lambda$ generate polynomial $b_j(x)$

Commit to: $C_0 = C([b_1(0)]E,..,[b_\lambda(0)]E,y_0)$ $C_i = C(b_1(i),..,b_\lambda(i),y_i)$

$C_0' = C([f^{(1)}(0)]E,y_0)$ $C_i' = C(f(1)(i),y_i)$

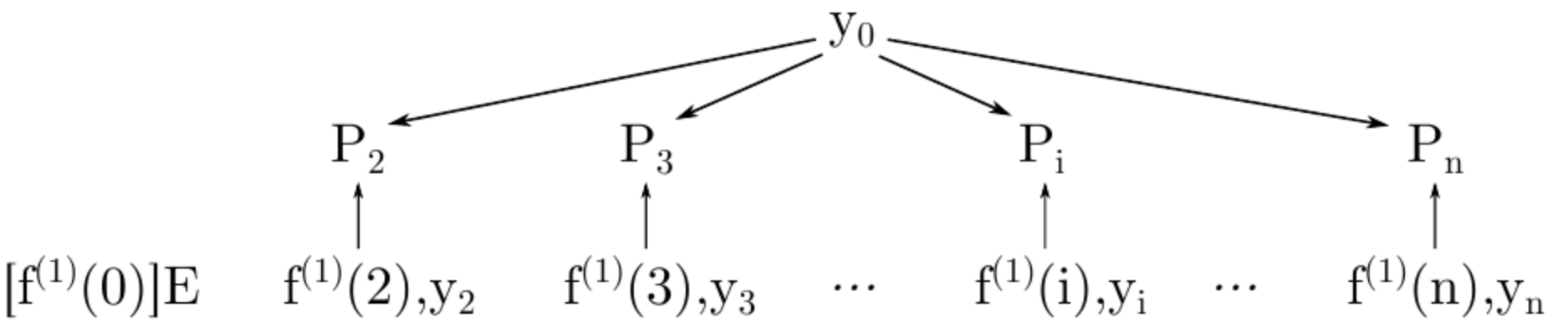
Challenge: $c = c_1..c_\lambda = H(C_0,..,C_n,C_0',..,C_n')$

Response: $r_j(x) = b_j(x) - c_j f^{(1)}(x)$

Verification: $c = H(\overline{C}_0,C_1,..,\overline{C}_i,..,C_n,C_0',..,C_n')$

by P_i $\overline{C}_0 = C([r_1(0)][c_1 f^{(1)}(0)]E,...,[r_\lambda(0)][c_\lambda f^{(1)}(0)]E,y_0)$

$\overline{C}_i = C(r_1(i)+c_1 f^{(1)}(i),...,r_\lambda(i)+c_\lambda f^{(1)}(i)E,y_i)$



For $j = 1,\ldots,\lambda$ generate polynomial $b_j(x)$

Commit to: $C_0 = C([b_1(0)]E,..,[b_\lambda(0)]E,y_0)$ $C_i = C(b_1(i),..,b_\lambda(i),y_i)$

$C_0' = C([f^{(1)}(0)]E,y_0)$ $C_i' = C(f(1)(i),y_i)$

Challenge: $c = c_1..c_\lambda = H(C_0,..,C_n,C_0',..,C_n')$

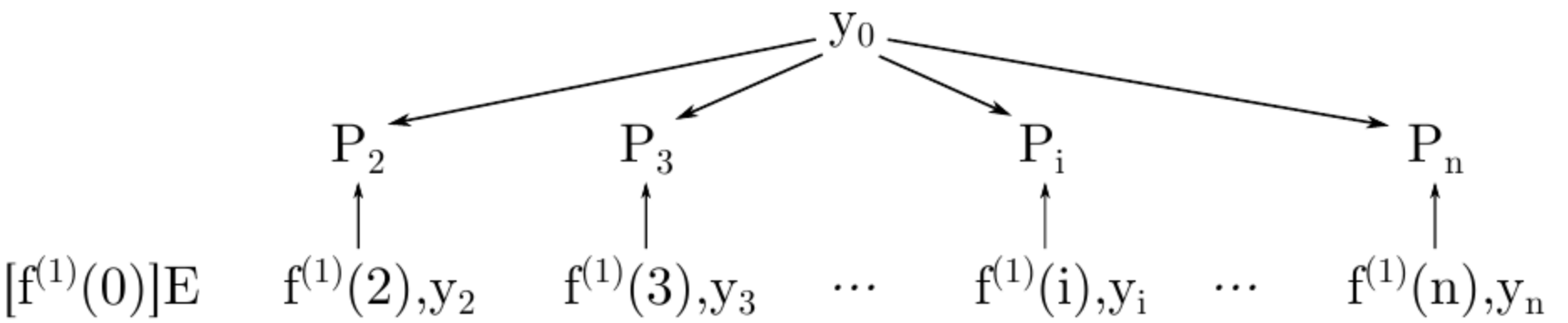
Response: $r_j(x) = b_j(x) - c_j f^{(1)}(x)$

Verification: $c = H(\overline{C}_0,C_1,..,\overline{C}_i,..,C_n,C_0',..,C_n')$

by P_i

$\overline{C}_0 = C([r_1(0)][c_1 f^{(1)}(0)]E,\ldots,[r_\lambda(0)][c_\lambda f^{(1)}(0)]E,y_0)$

$\overline{C}_i = C(r_1(i)+c_1 f^{(1)}(i),\ldots,r_\lambda(i)+c_\lambda f^{(1)}(i)E,y_i)$



For $j = 1,...,\lambda$ generate polynomial $b_j(x)$

Commit to: $C_0 = C([b_1(0)]E,..,[b_\lambda(0)]E,y_0)$ $C_i = C(b_1(i),..,b_\lambda(i),y_i)$

$C_0' = C([f^{(1)}(0)]E,y_0)$ $C_i' = C(f(1)(i),y_i)$

Challenge: $c = c_1..c_\lambda = H(C_0,..,C_n,C_0',..,C_n')$

Response: $r_j(x) = b_j(x) - c_j f^{(1)}(x)$

Verification: $c = H(\overline{C}_0,C_1,..,\overline{C}_i,..,C_n,C_0',$
by P_i

$\overline{C}_0 = C([r_1(0)][c_1 f^{(1)}(0)]E,...$

$\overline{C}_i = C(r_1(i)+c_1 f^{(1)}(i),...,r_\lambda(i)$

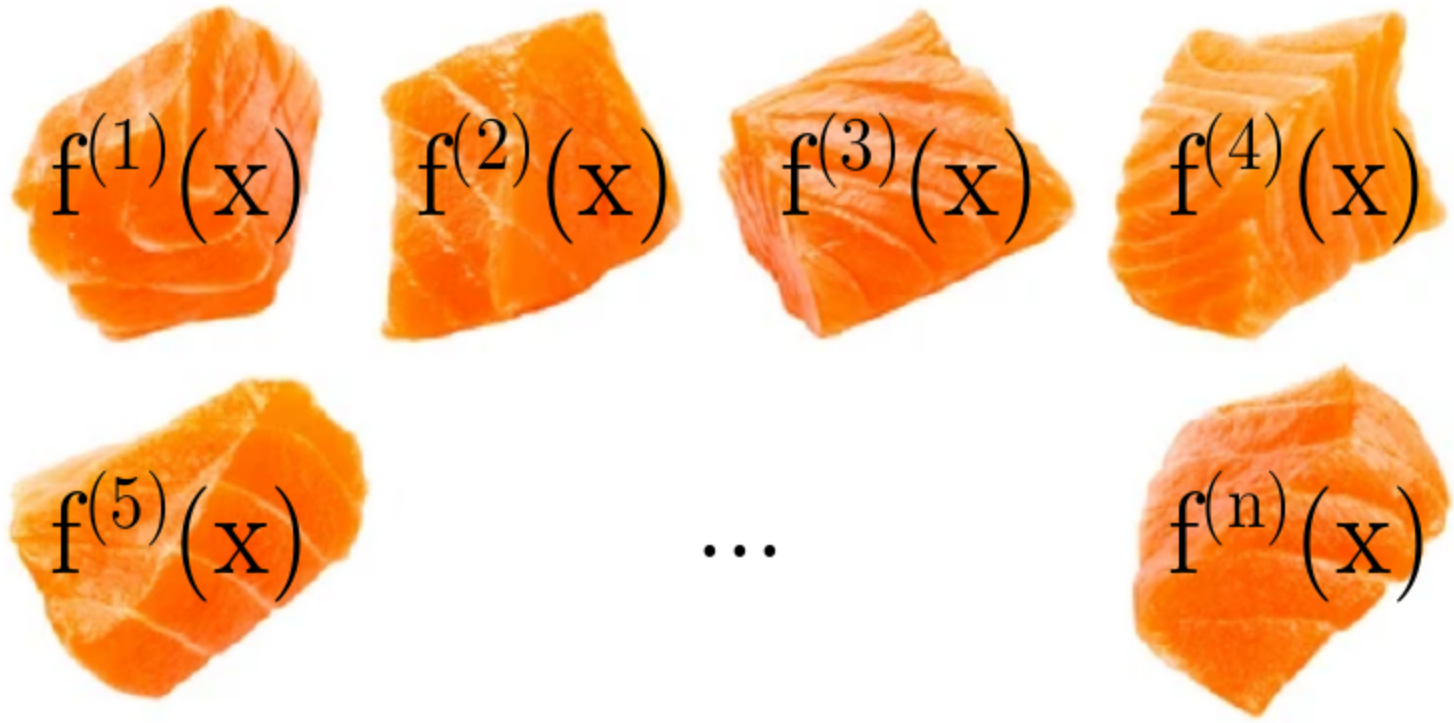
Security in the Quantum ROM

[Don,Fehr,Majenz,Schaffner'19]: Proof of knowledge

[Unruh'16], [Unruh'17]: Zero-knowledge

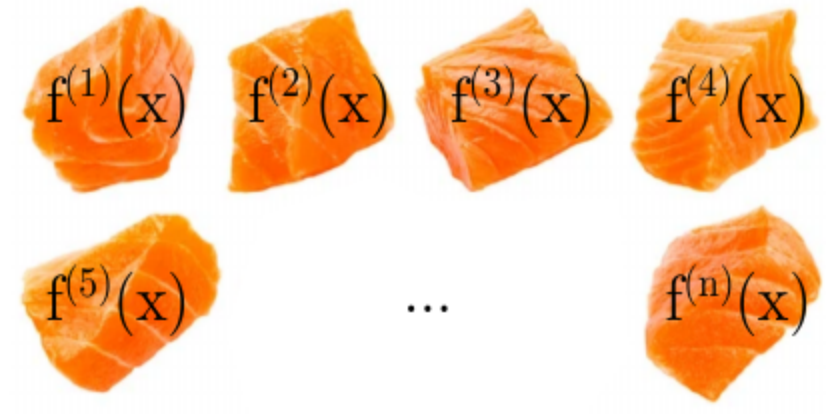
CSI-RAShi

1. Secret sharing



CSI-RAShi

1. Secret sharing

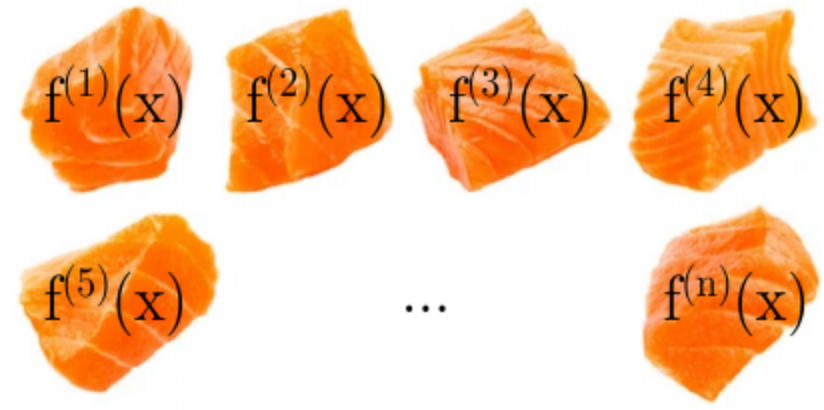


CSI-RAShi

1. Secret sharing

Player i samples $f^{(i)}(x) \leftarrow \mathbb{Z}_N[x]_{\leq k-1}$ and $R^{(i)} \leftarrow \mathcal{E}$

CSI-RAShi



1. Secret sharing

Player i samples $f^{(i)}(x) \leftarrow \mathbb{Z}_N[x]_{\leq k-1}$ and $R^{(i)} \leftarrow \mathcal{E}$

Full statement: $x^{(i)} = ((R^{(i)}, [f^{(i)}(0)]R^{(i)}), \{f^{(i)}(j)\}_{j\in\{1,\ldots,n\}})$.

CSI-RAShi

1. Secret sharing

Player i samples $f^{(i)}(x) \leftarrow \mathbb{Z}_N[x]_{\leq k-1}$ and $R^{(i)} \leftarrow \mathcal{E}$

Full statement: $x^{(i)} = ((R^{(i)}, [f^{(i)}(0)]R^{(i)}), \{f^{(i)}(j)\}_{j\in\{1,\ldots,n\}})$.

PVP for $x^{(i)}$ and $f^{(i)}(x)$: $\pi^{(i)} = (\tilde{\pi}^{(i)}, \{\pi_j^{(i)}\}_{j\in\{0,\ldots,n\}})$

CSI-RAShi

1. Secret sharing

Player i samples $f^{(i)}(x) \leftarrow \mathbb{Z}_N[x]_{\leq k-1}$ and $R^{(i)} \leftarrow \mathcal{E}$

Full statement: $x^{(i)} = ((R^{(i)}, [f^{(i)}(0)]R^{(i)}), \{f^{(i)}(j)\}_{j\in\{1,\ldots,n\}})$.

PVP for $x^{(i)}$ and $f^{(i)}(x)$: $\pi^{(i)} = (\tilde{\pi}^{(i)}, \{\pi_j^{(i)}\}_{j\in\{0,\ldots,n\}})$

$C_0,\ldots,C_n,C_0',\ldots,C_n',r$

y_j,y_j'

CSI-RAShi

1. Secret sharing

Player i samples $f^{(i)}(x) \leftarrow \mathbb{Z}_N[x]_{\leq k-1}$ and $R^{(i)} \leftarrow \mathcal{E}$

Full statement: $x^{(i)} = ((R^{(i)}, [f^{(i)}(0)]R^{(i)}), \{f^{(i)}(j)\}_{j\in\{1,\ldots,n\}})$.

PVP for $x^{(i)}$ and $f^{(i)}(x)$: $\pi^{(i)} = (\tilde{\pi}^{(i)}, \{\pi_j^{(i)}\}_{j\in\{0,\ldots,n\}})$

$C_0,\ldots,C_n,C_0',\ldots,C_n',r$ ← $\tilde{\pi}^{(i)}$

$\pi_j^{(i)}$ → y_j,y_j'

Publish $((R^{(i)}, [f^{(i)}(0)]R^{(i)}), \tilde{\pi}^{(i)}, \pi_0^{(i)})$

CSI-RAShi

1. Secret sharing

Player i samples $f^{(i)}(x) \leftarrow \mathbb{Z}_N[x]_{\leq k-1}$ and $R^{(i)} \leftarrow \mathcal{E}$

Full statement: $x^{(i)} = ((R^{(i)}, [f^{(i)}(0)]R^{(i)}), \{f^{(i)}(j)\}_{j\in\{1,\ldots,n\}})$.

PVP for $x^{(i)}$ and $f^{(i)}(x)$: $\pi^{(i)} = (\tilde{\pi}^{(i)}, \{\pi_j^{(i)}\}_{j\in\{0,\ldots,n\}})$

$C_0,\ldots,C_n,C_0',\ldots,C_n',r$ ← $\tilde{\pi}^{(i)}$

$\pi_j^{(i)}$ → y_j,y_j'

Publish $((R^{(i)}, [f^{(i)}(0)]R^{(i)}), \tilde{\pi}^{(i)}, \pi_0^{(i)})$

Send $(f^{(i)}(j), \pi_j^{(i)})$ privately to Player j

CSI-RAShi

1. Secret sharing
2. Verification and audit

CSI-RAShi

1. Secret sharing
2. Verification and audit

   For all i, each player j verifies

$$f^{(i)}(j) \text{ using } (\tilde{\pi}^{(i)}, \pi_j^{(i)})$$

$$(R^{(i)}, [f^{(i)}(0)]R^{(i)}) \text{ using } (\tilde{\pi}^{(i)}, \pi_0^{(i)})$$

CSI-RAShi

1. Secret sharing
2. Verification and audit

   For all i, each player j verifies

$$f^{(i)}(j) \text{ using } (\tilde{\pi}^{(i)}, \pi_j^{(i)})$$

$$(R^{(i)}, [f^{(i)}(0)]R^{(i)}) \text{ using } (\tilde{\pi}^{(i)}, \pi_0^{(i)})$$

If either fails, broadcast a complaint against Player i.

CSI-RAShi

1. Secret sharing
2. Verification and audit

   For all i, each player j verifies

$$f^{(i)}(j) \text{ using } (\tilde{\pi}^{(i)}, \pi_j^{(i)})$$

$$(R^{(i)}, [f^{(i)}(0)]R^{(i)}) \text{ using } (\tilde{\pi}^{(i)}, \pi_0^{(i)})$$

If either fails, broadcast a complaint against Player i.

$$\text{Player } j \xrightarrow{\text{complaint}} \text{Player } i$$

CSI-RAShi

1. Secret sharing
2. Verification and audit

   For all i, each player j verifies

$$f^{(i)}(j) \text{ using } (\tilde{\pi}^{(i)}, \pi_j^{(i)})$$

$$(R^{(i)}, [f^{(i)}(0)]R^{(i)}) \text{ using } (\tilde{\pi}^{(i)}, \pi_0^{(i)})$$

If either fails, broadcast a complaint against Player i.

$$\text{Player } j \xrightarrow{\text{complaint}} \text{Player } i \xrightarrow{(f^{(i)}(j), \pi_j^{(i)})} \text{other players}$$

Verification

CSI-RAShi

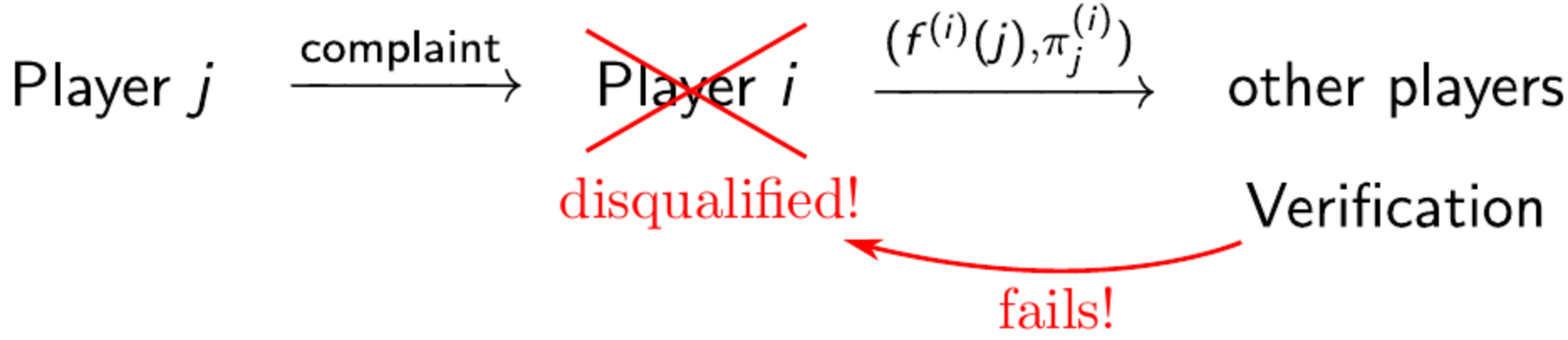
1. Secret sharing
2. Verification and audit

   For all i, each player j verifies

$$f^{(i)}(j) \text{ using } (\tilde{\pi}^{(i)}, \pi_j^{(i)})$$

$$(R^{(i)}, [f^{(i)}(0)]R^{(i)}) \text{ using } (\tilde{\pi}^{(i)}, \pi_0^{(i)})$$

If either fails, broadcast a complaint against Player i.



CSI-RAShi

1. Secret sharing
2. Verification and audit

   For all i, each player j verifies

$$f^{(i)}(j) \text{ using } (\tilde{\pi}^{(i)}, \pi_j^{(i)})$$

$$(R^{(i)}, [f^{(i)}(0)]R^{(i)}) \text{ using } (\tilde{\pi}^{(i)}, \pi_0^{(i)})$$

If either fails, broadcast a complaint against Player i.

Player j $\xrightarrow{\text{complaint}}$ ~~Player i~~

$\xrightarrow{\text{complaint}}$

$\xrightarrow{\text{complaint}}$ disqualified!

$\vdots$

$\xrightarrow{\text{complaint}}$

CSI-RAShi

1. Secret sharing
2. Verification and audit
3. Computing shares

CSI-RAShi

1. Secret sharing
2. Verification and audit
3. Computing shares

set of qualified players: $\mathcal{Q} \subseteq \{1, \ldots, n\}$

CSI-RAShi

1. Secret sharing
2. Verification and audit
3. Computing shares

set of qualified players: $\mathcal{Q} \subseteq \{1, \ldots, n\}$

Player j's share: $f(j) = \sum_{i \in \mathcal{Q}} f^{(i)}(j)$

CSI-RAShi

1. Secret sharing
2. Verification and audit
3. Computing shares

set of qualified players: $\mathcal{Q} \subseteq \{1, \ldots, n\}$

Player j's share: $$f(j) = \sum_{i \in \mathcal{Q}} f^{(i)}(j)$$

Shared secret key $$f(0) = \sum_{i \in \mathcal{Q}} f^{(i)}(0)$$

CSI-RAShi

1. Secret sharing
2. Verification and audit
3. Computing shares
4. Computing the joint public key

CSI-RAShi

1. Secret sharing
2. Verification and audit
3. Computing shares
4. Computing the joint public key

Shared secret key

$$f(0) = \sum_{i \in \mathcal{Q}} f^{(i)}(0)$$

Shared public key

$$[f(0)]E_0$$

CSI-RAShi

1. Secret sharing
2. Verification and audit
3. Computing shares
4. Computing the joint public key

Shared secret key

$$f(0) = \sum_{i \in \mathcal{Q}} f^{(i)}(0)$$

Shared public key

$$[f(0)]E_0$$

~~$\sum_{i \in \mathcal{Q}} [f^{(i)}(0)]E_0$~~ ~~$\prod_{i \in \mathcal{Q}} [f^{(i)}(0)]E_0$~~

CSI-RAShi

1. Secret sharing
2. Verification and audit
3. Computing shares
4. Computing the joint public key

Shared secret key

$$f(0) = \sum_{i \in \mathcal{Q}} f^{(i)}(0)$$

Shared public key

$$[f(0)]E_0$$

Assume (wlog) $\mathcal{Q} = \{1, \ldots, n'\}$

$$\left[\sum_{i \in \mathcal{Q}} f^{(i)}(0)\right] E_0 = [f^{(n')}(0)] \left(\ldots \left([f^{(1)}(0)]E \right) \right)$$

CSI-RAShi

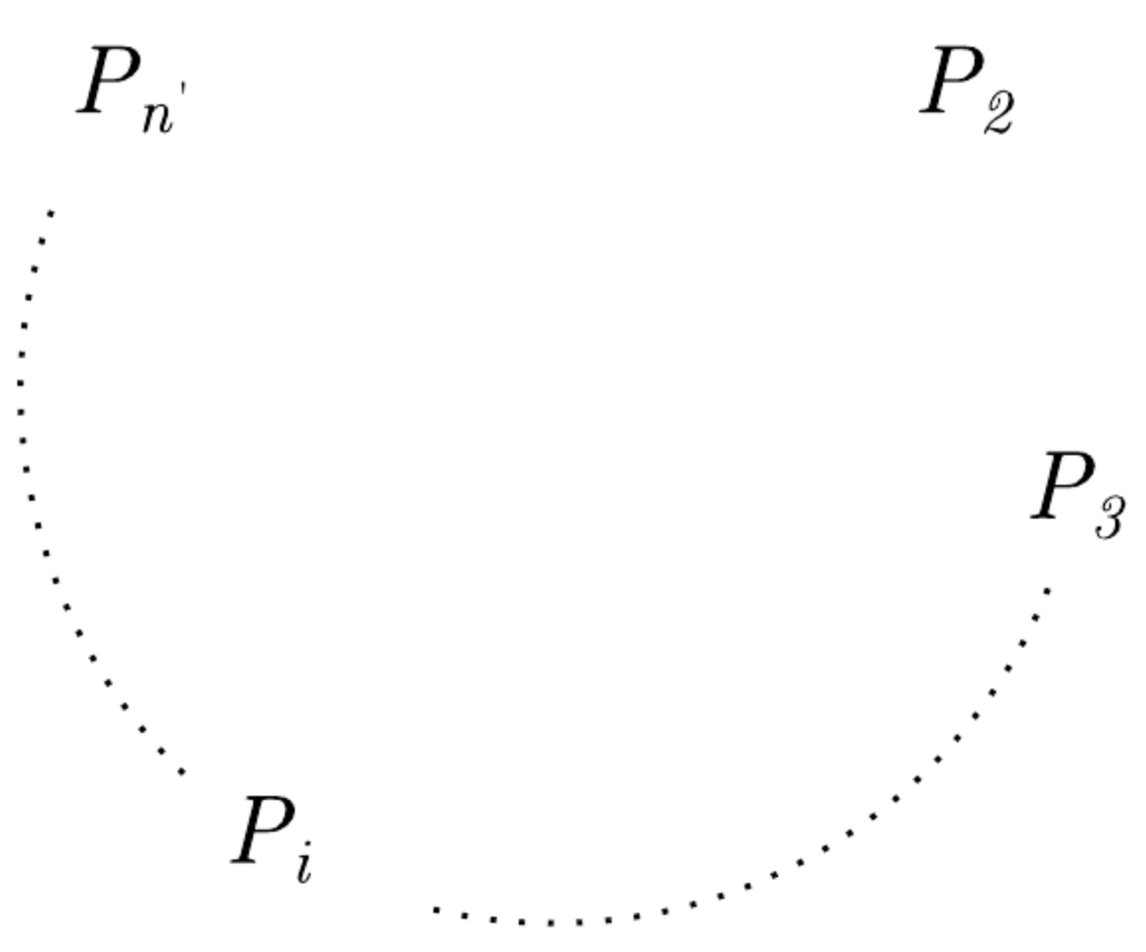
1. Secret sharing
2. Verification and audit
3. Computing shares
4. Computing the joint public key

Shared secret key

$$f(0) = \sum_{i \in \mathcal{Q}} f^{(i)}(0)$$

Shared public key

$$[f(0)]E_0$$



CSI-RAShi

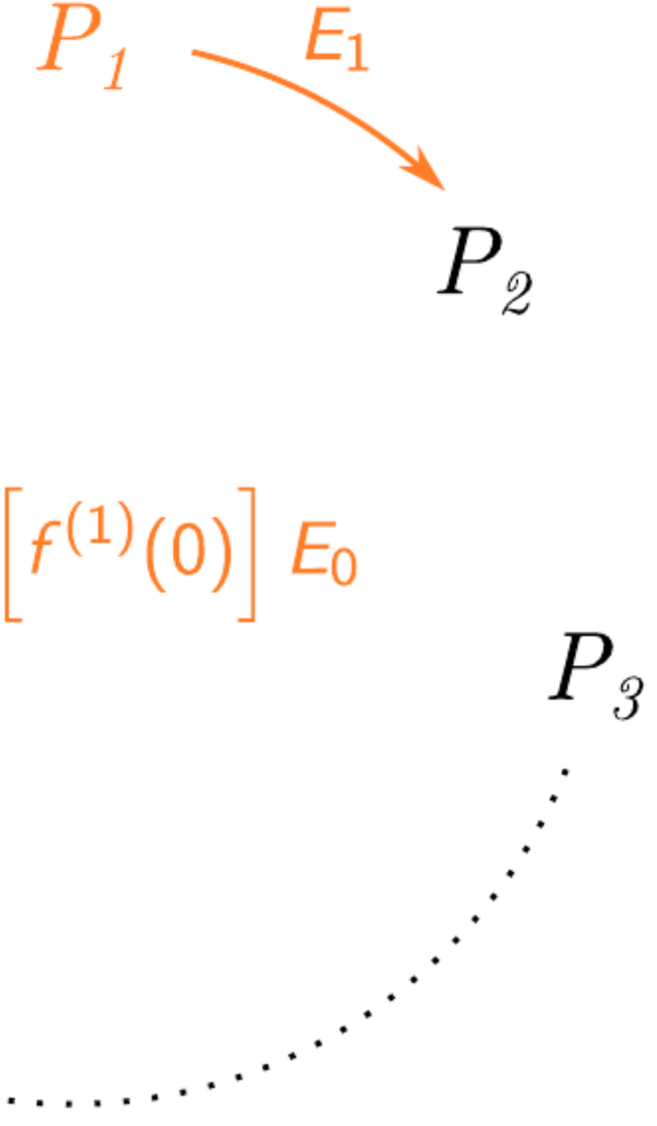
1. Secret sharing
2. Verification and audit
3. Computing shares
4. Computing the joint public key

Shared secret key

$$f(0) = \sum_{i \in \mathcal{Q}} f^{(i)}(0)$$

Shared public key

$$[f(0)]E_0$$



CSI-RAShi

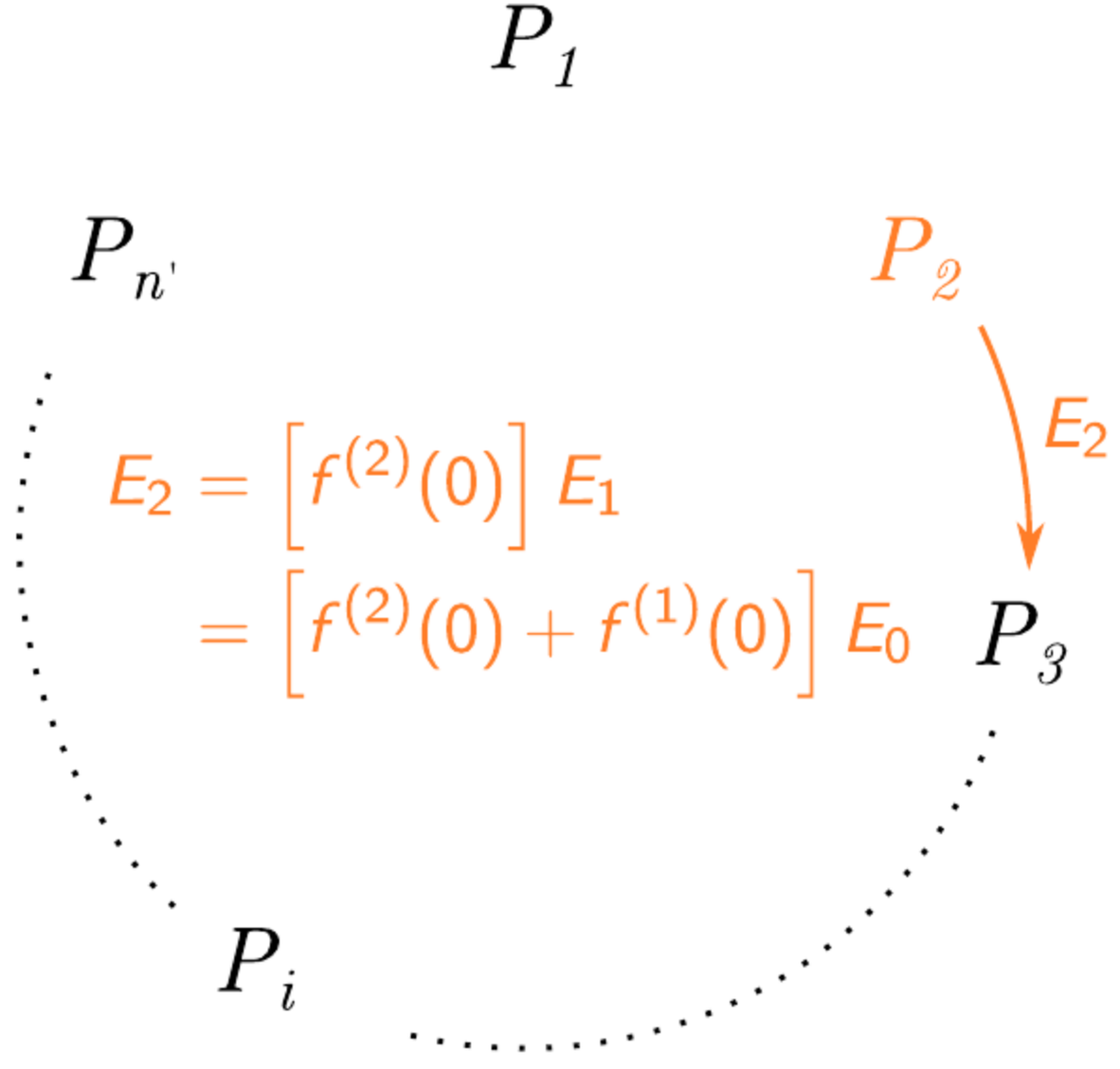
1. Secret sharing
2. Verification and audit
3. Computing shares
4. Computing the joint public key

Shared secret key

$$f(0) = \sum_{i \in \mathcal{Q}} f^{(i)}(0)$$

Shared public key

$$[f(0)]E_0$$



CSI-RAShi

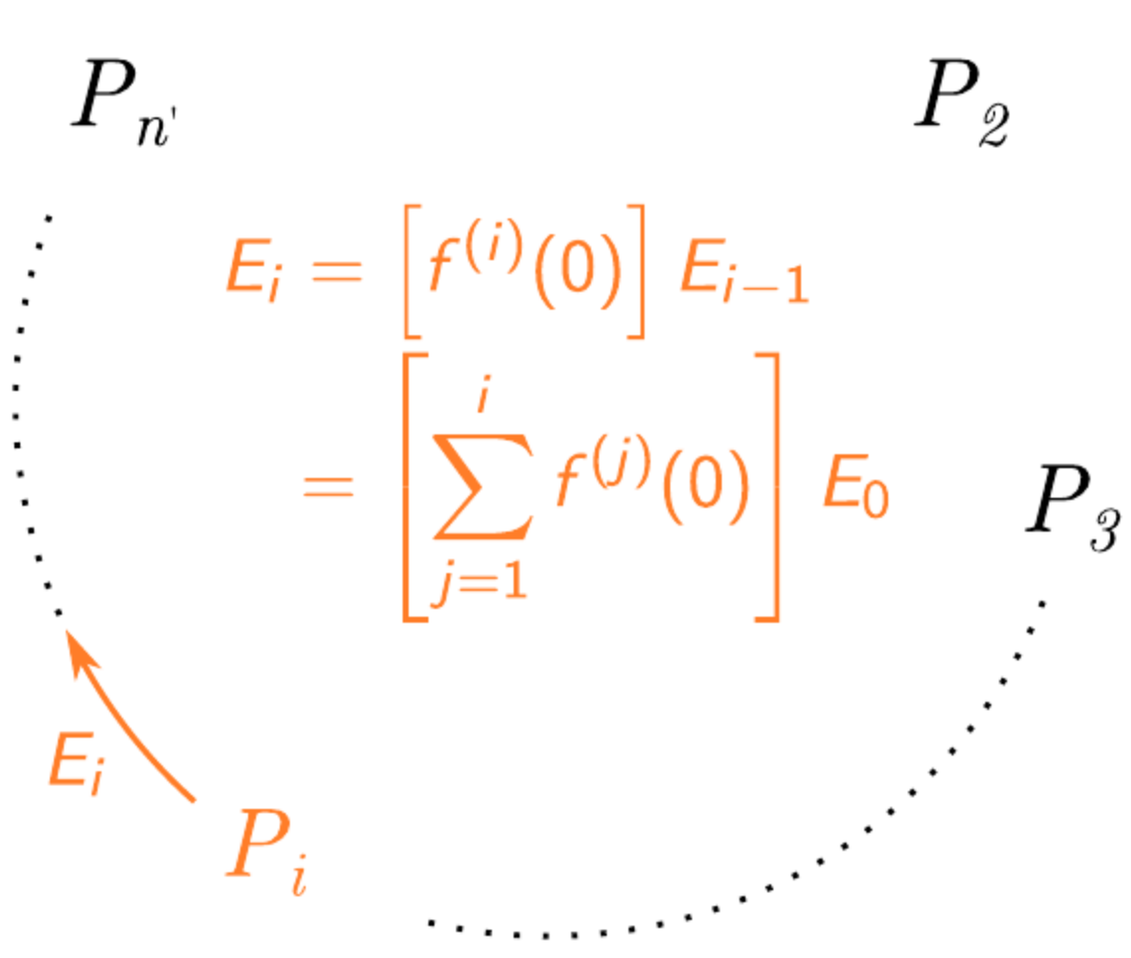
1. Secret sharing
2. Verification and audit
3. Computing shares
4. Computing the joint public key

Shared secret key

$$f(0) = \sum_{i \in \mathcal{Q}} f^{(i)}(0)$$

Shared public key

$$[f(0)]E_0$$



CSI-RAShi

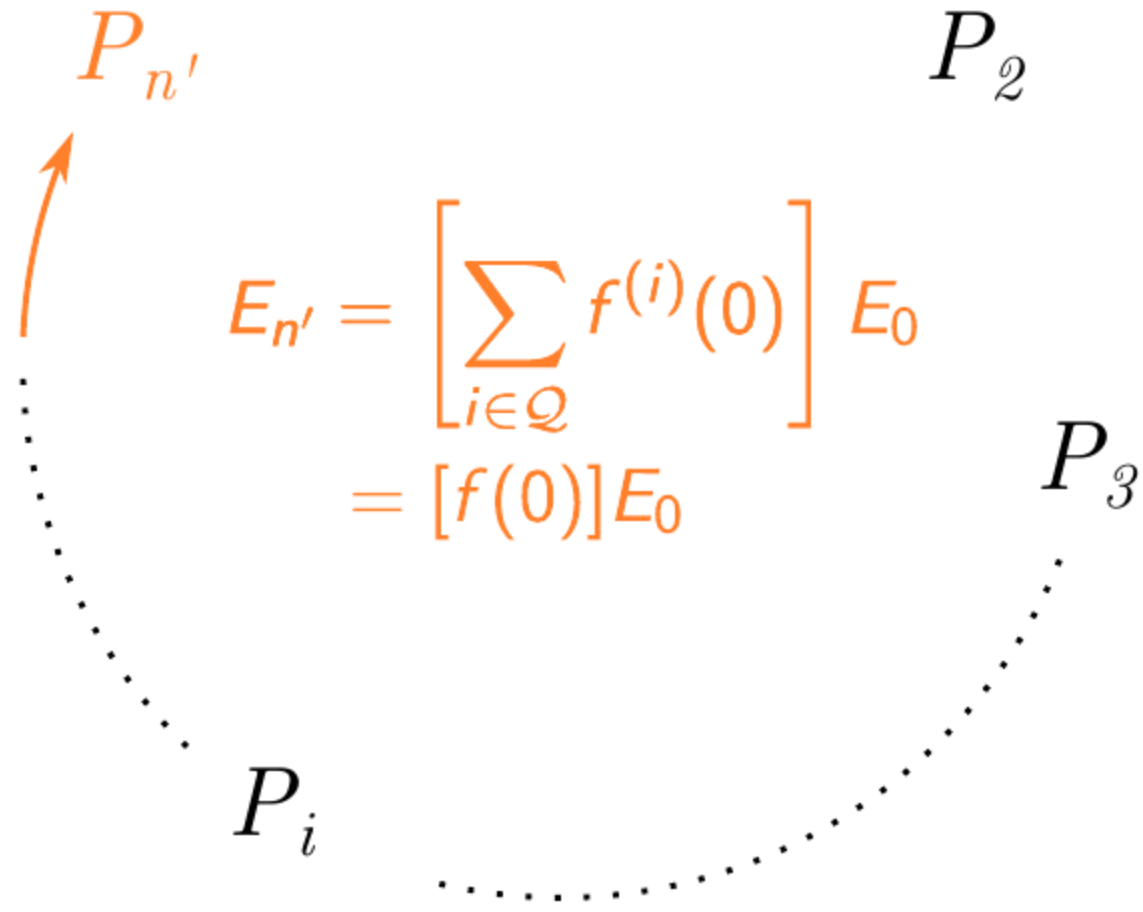
1. Secret sharing
2. Verification and audit
3. Computing shares
4. Computing the joint public key

Shared secret key

$$f(0) = \sum_{i \in \mathcal{Q}} f^{(i)}(0)$$

Shared public key

$$[f(0)]E_0$$



Proof of correct group action

Verify $E_{i-1} \mapsto E_i = [f^{(i)}(0)]E_{i-1}$

Proof of correct group action

Verify $E_{i-1} \mapsto E_i = [f^{(i)}(0)]E_{i-1}$

PVP commitment $(R^{(i)}, [f^{(i)}(0)]R^{(i)}) =: (R_{i-1}, R_i)$

Proof of correct group action

Verify $E_{i-1} \mapsto E_i = [f^{(i)}(0)]E_{i-1}$

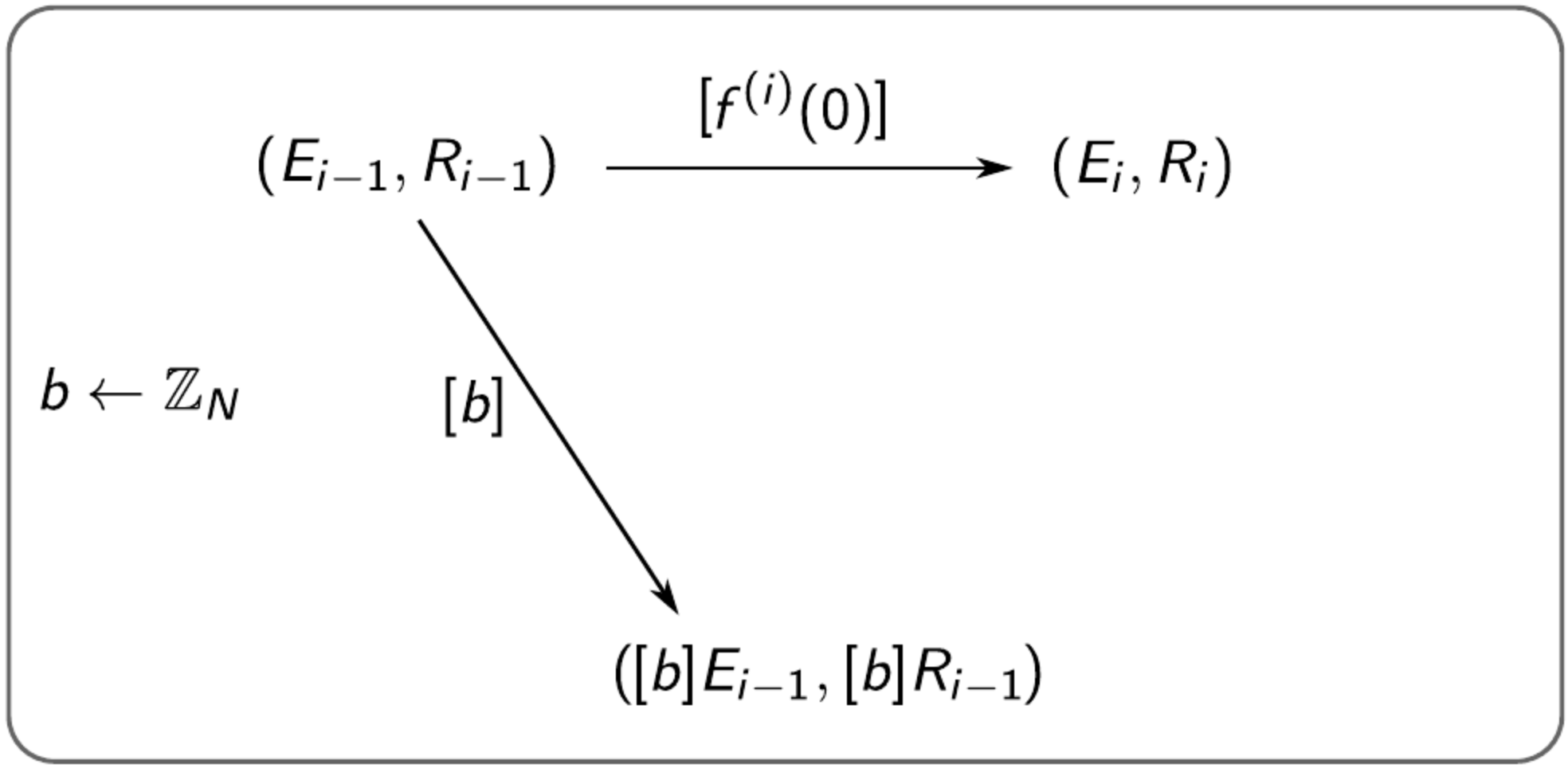
PVP commitment $(R^{(i)}, [f^{(i)}(0)]R^{(i)}) =: (R_{i-1}, R_i)$

$$(E_{i-1}, R_{i-1}) \xrightarrow{[f^{(i)}(0)]} (E_i, R_i)$$

Proof of correct group action

Verify $E_{i-1} \mapsto E_i = [f^{(i)}(0)]E_{i-1}$

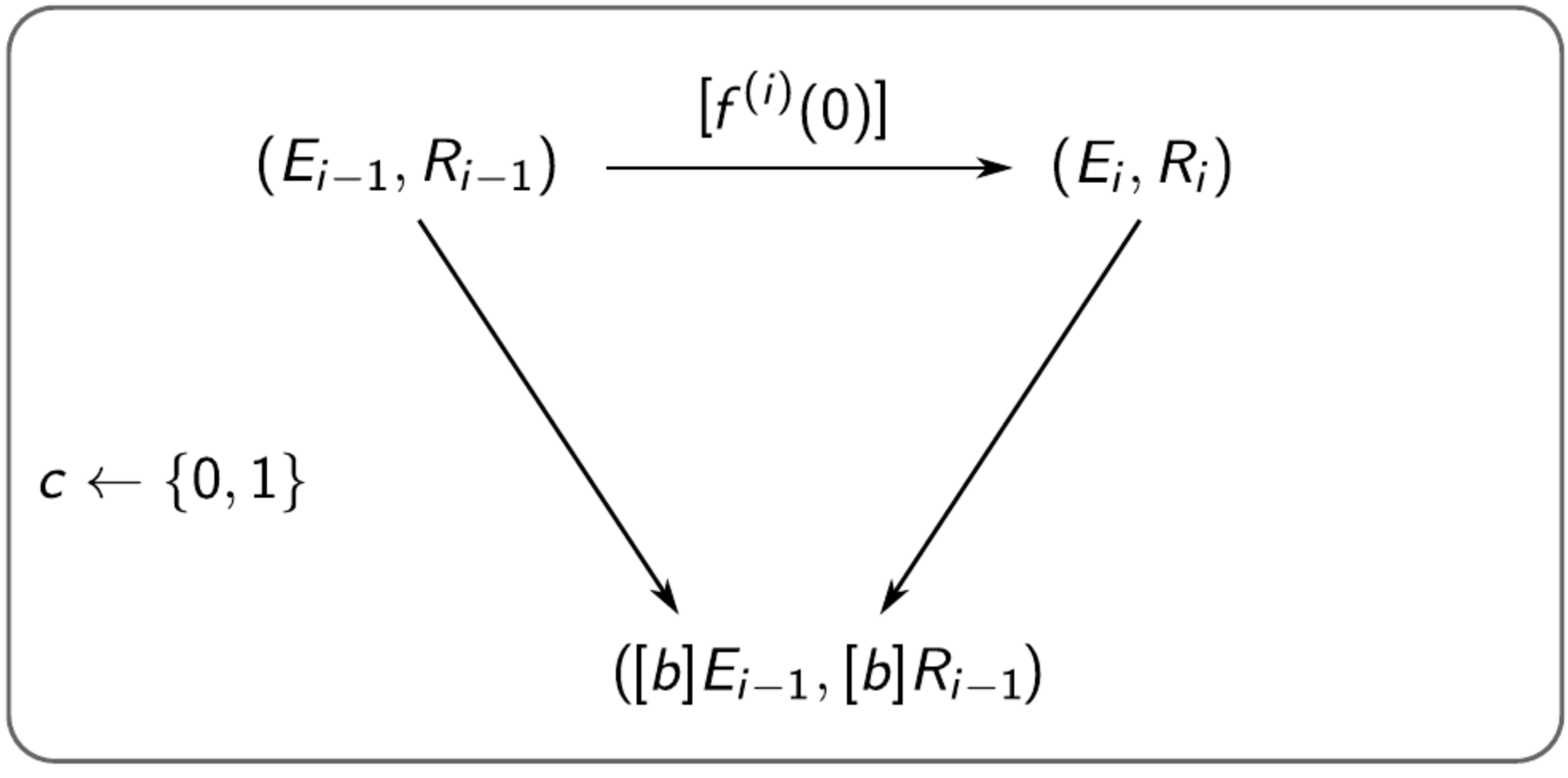
PVP commitment $(R^{(i)}, [f^{(i)}(0)]R^{(i)}) =: (R_{i-1}, R_i)$



Proof of correct group action

Verify $E_{i-1} \mapsto E_i = [f^{(i)}(0)]E_{i-1}$

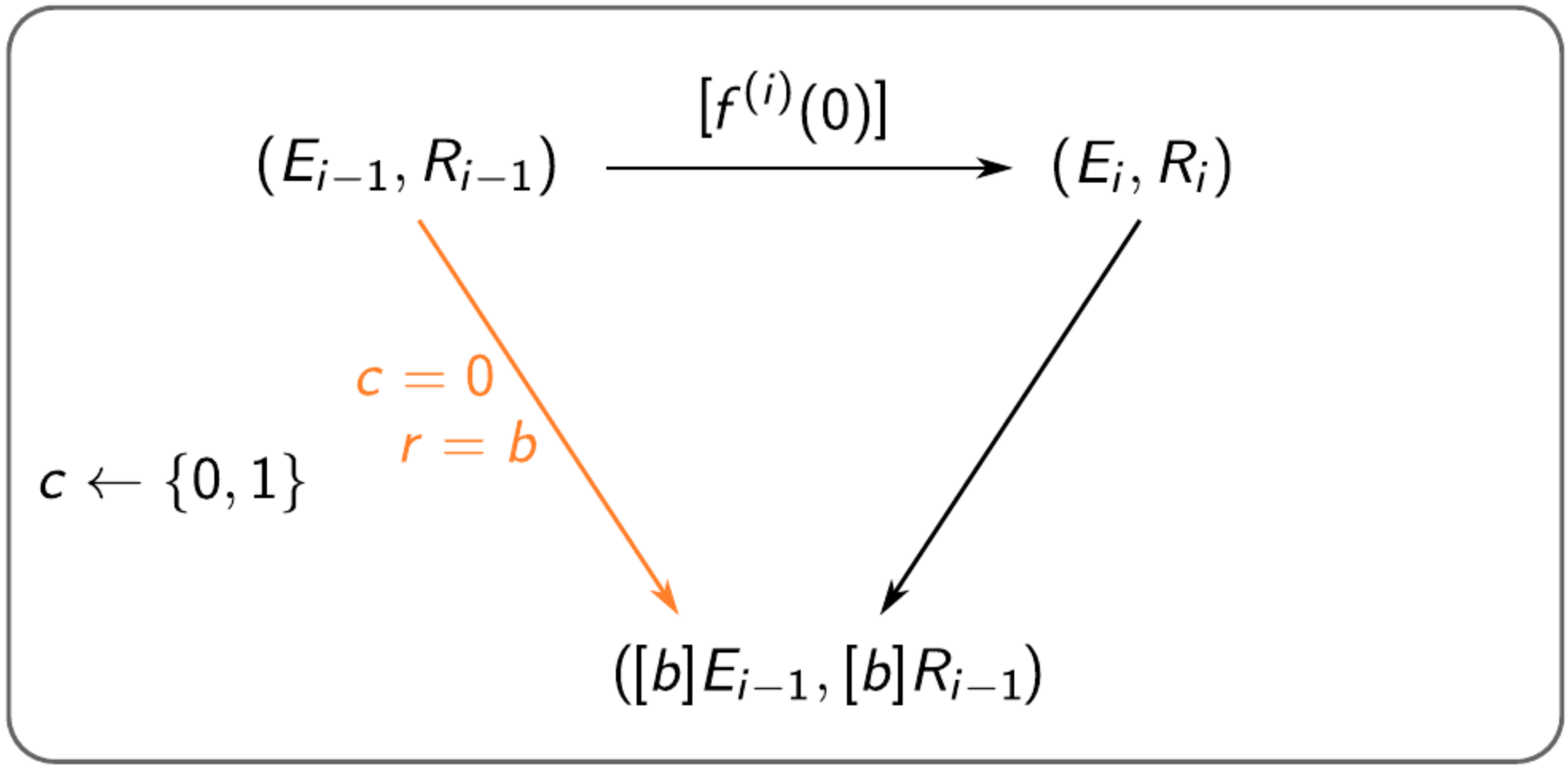
PVP commitment $(R^{(i)}, [f^{(i)}(0)]R^{(i)}) =: (R_{i-1}, R_i)$



Proof of correct group action

Verify $E_{i-1} \mapsto E_i = [f^{(i)}(0)]E_{i-1}$

PVP commitment $(R^{(i)}, [f^{(i)}(0)]R^{(i)}) =: (R_{i-1}, R_i)$

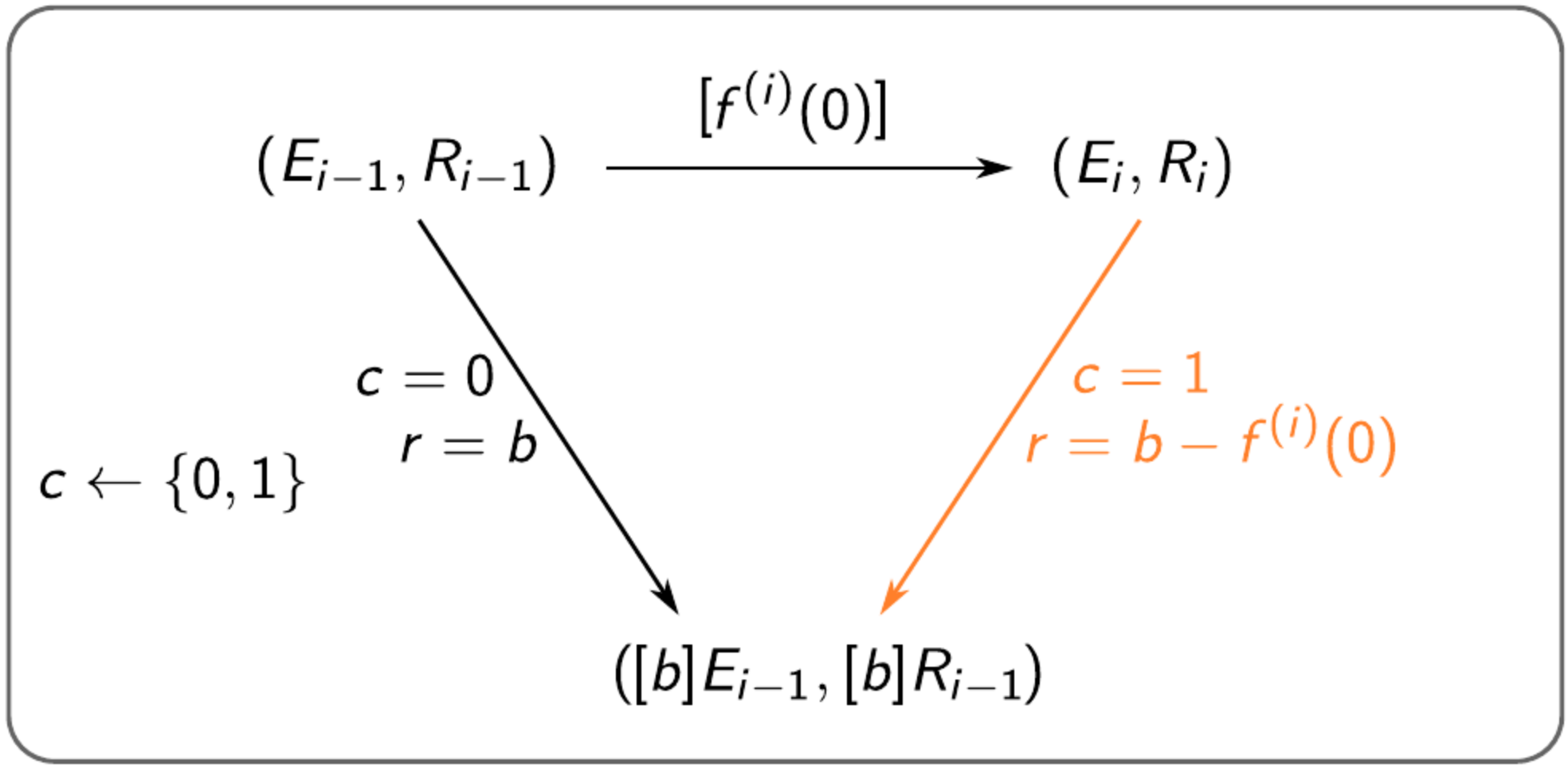


$r = b - cf^{(i)}(0)$

Proof of correct group action

Verify $E_{i-1} \mapsto E_i = [f^{(i)}(0)]E_{i-1}$

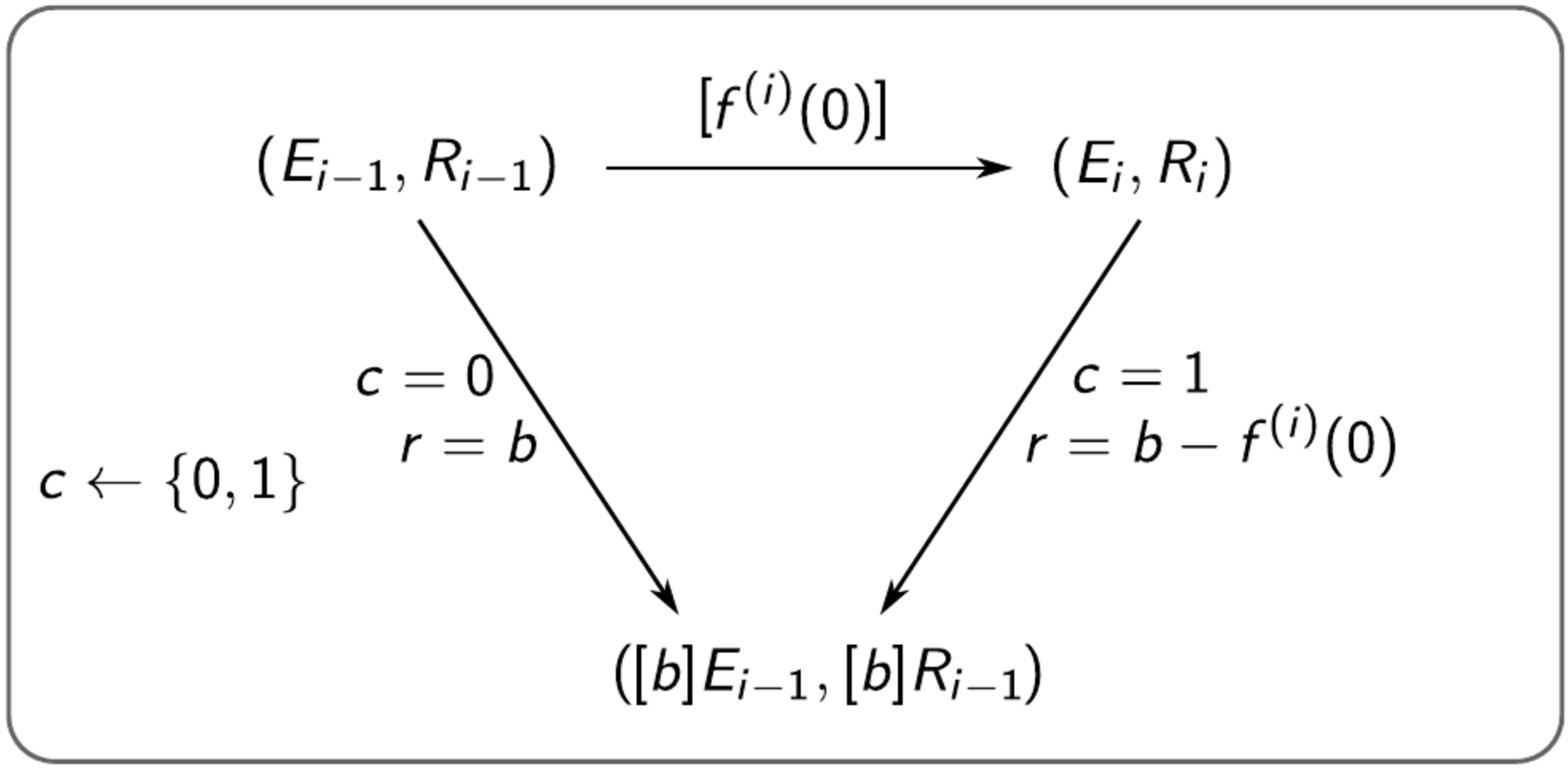
PVP commitment $(R^{(i)}, [f^{(i)}(0)]R^{(i)}) =: (R_{i-1}, R_i)$



Proof of correct group action

Verify $E_{i-1} \mapsto E_i = [f^{(i)}(0)]E_{i-1}$

PVP commitment $(R^{(i)}, [f^{(i)}(0)]R^{(i)}) =: (R_{i-1}, R_i)$



$r = b - cf^{(i)}(0)$

CSI-RAShi

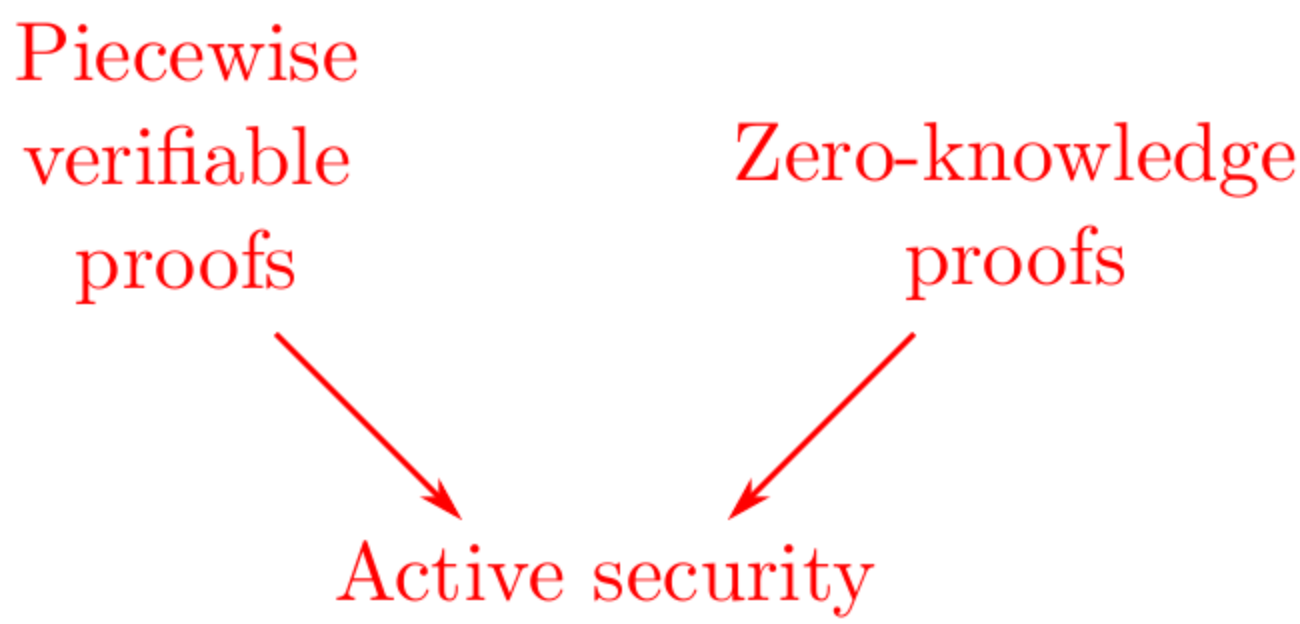
1. Secret sharing
2. Verification and audit
3. Computing shares
4. Computing the joint public key

Piecewise verifiable proofs

Zero-knowledge proofs

CSI-RAShi

1. Secret sharing
2. Verification and audit
3. Computing shares
4. Computing the joint public key



CSI-RAShi

1. Secret sharing
2. Verification and audit $\longrightarrow$ Robustness
3. Computing shares
4. Computing the joint public key

Piecewise verifiable proofs

Zero-knowledge proofs

Active security

Cost

- PVP: $2 + n\lambda$ group actions
- ZKP: $\lambda + n(1 + 3\lambda)$ group actions

Cost

- PVP: $2 + n\lambda$ group actions
- ZKP: $\lambda + n(1 + 3\lambda)$ group actions

Total cost: $T(n, \lambda) = 2 + \lambda + n(1 + 4\lambda)$

Cost per player: $T_P(n, \lambda) = 3(1 + n\lambda)$

Cost

- PVP: $2 + n\lambda$ group actions
- ZKP: $\lambda + n(1 + 3\lambda)$ group actions

Total cost: $T(n, \lambda) = 2 + \lambda + n(1 + 4\lambda)$

Cost per player: $T_P(n, \lambda) = 3(1 + n\lambda)$

Isogeny instantiation

- $\mathcal{E}$: supersingular elliptic curves defined over $\mathbb{F}_p$ with endomorphism ring $End(E) \simeq \mathcal{O}_{\mathbb{Q}(\sqrt{-p})}$.
- $\mathcal{G}$: ideal-class group $\mathrm{Cl}(\mathcal{O}_{\mathbb{Q}(\sqrt{-p})})$ with generator $\mathfrak{g}$

Cost

- PVP: $2 + n\lambda$ group actions
- ZKP: $\lambda + n(1 + 3\lambda)$ group actions

Total cost: $T(n, \lambda) = 2 + \lambda + n(1 + 4\lambda)$

Cost per player: $T_P(n, \lambda) = 3(1 + n\lambda)$

Isogeny instantiation

- $\mathcal{E}$: supersingular elliptic curves defined over $\mathbb{F}_p$ with endomorphism ring $End(E) \simeq \mathcal{O}_{\mathbb{Q}(\sqrt{-p})}$.
- $\mathcal{G}$: ideal-class group $\mathrm{Cl}(\mathcal{O}_{\mathbb{Q}(\sqrt{-p})})$ with generator $\mathfrak{g}$

$$T(n, 128) = 4.5 + 18n \text{ seconds for CSIDH-512}$$

Conclusion

- Piecewise verifiable proofs as alternative to Pedersen commitments in the very hard homogeneous spaces setting
- Robust and actively secure distributed key generation
- Post-quantum secure in the isogeny setting (QROM)

Conclusion

- Piecewise verifiable proofs as alternative to Pedersen commitments in the very hard homogeneous spaces setting
- Robust and actively secure distributed key generation
- Post-quantum secure in the isogeny setting (QROM)
- Still quite slow, mainly due to the sequential public-key computation
- Relies on knowledge of the ideal-class group

Conclusion

- Piecewise verifiable proofs as alternative to Pedersen commitments in the very hard homogeneous spaces setting
- Robust and actively secure distributed key generation
- Post-quantum secure in the isogeny setting (QROM)
- Still quite slow, mainly due to the sequential public-key computation
- Relies on knowledge of the ideal-class group

- Are PVPs adaptable to other cryptographic protocols?