Aggios: Scalable Aggregator-Based Voting

ZKProof 2024

Luxembourg National Research Fund

Pablo, grant reference (16326754).

Doron Zarchy

APSIA, University of Luxembourg

Challenges with Current Voting Systems

- We vote primarily to elect representatives who often become disconnected from the people's needs once they are in office. ”Vote for the man who promises least; he'll be the least disappointing." – Bernard Baruch
- Long terms and infrequent elections can lead to politicians prioritizing their own agendas or special interests over the public good
- Lack of accountability: Elected officials may not feel the need to respond to their constituents' concerns regularly
- "Politics is the art of looking for trouble, finding it everywhere, diagnosing it incorrectly, and applying the wrong remedies." – Groucho Marx

High-Frequency Elections

High-frequency voting: involves holding referendums or elections more frequently than in typical electoral systems.

Benefits of High-Frequency Elections

- **Increased Public Engagement:** Regular voting opportunities keep the public actively involved in the democratic process.
- **Enhanced Accountability:** Frequent elections ensure politicians remain responsive to their constituents' needs and preferences.
- **Empowerment:** Voters feel more empowered as their voices are heard more regularly, strengthening the democratic process.
- **Increase Happiness**: Empirical scientists, e.g. Bruno S. Frey among many, show that direct democracy, contribute to stability and happiness.

Examples: Switzerland conducts frequent referendums on a wide range of issues, with almost 600 national votes since 1848, fostering continuous public involvement

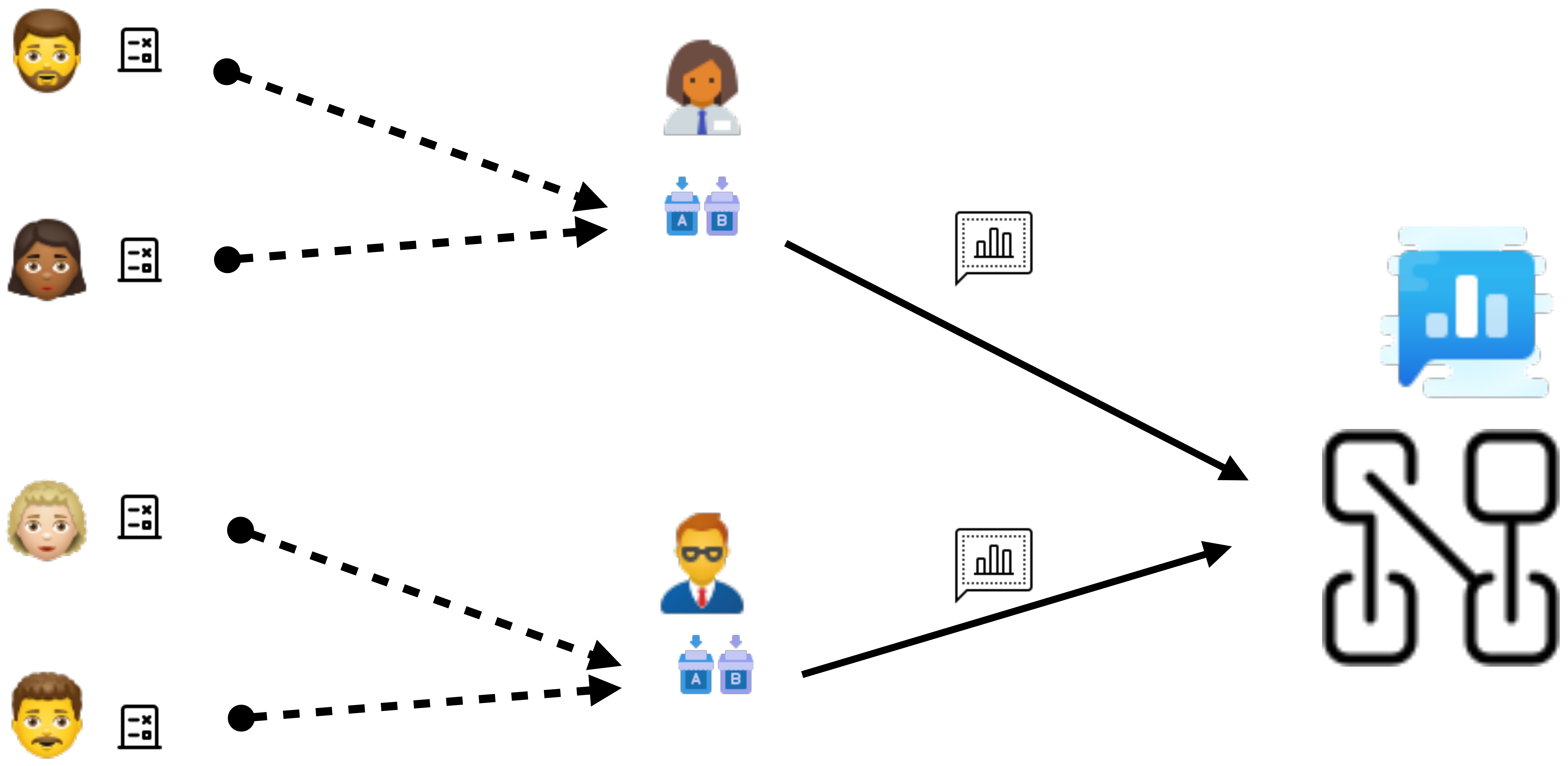
Challenges to High-Frequency Voting

- **High Manpower Requirements:** Traditional offline voting requires significant manpower for setup, monitoring, and counting votes, which can be resource-intensive.
- **Electronic Voting Challenges:** Electronic systems demand high bandwidth and substantial time for secure and accurate vote verification.
- Cost Implications: Both offline and electronic voting systems incur substantial costs due to their regular occurrence.
- Logistical Challenges: Managing frequent elections can be complex, requiring efficient systems to handle logistics without delays.
- **Security Concerns:** Ensuring the integrity and security of frequent elections is critical to maintaining public trust.

The Need for Advanced Solutions:

Modern electoral systems must scale effectively to accommodate multiple events without losing performance on security and accuracy.

Aggregation Based Voting



Enhancing Scalability with Aggios

Aggios:

- Aggios is a proxy voting scheme that is based on aggregator
- Utilizes new accumulator scheme to manage scalability issues in high-frequency voting.

Benefits of Aggios:

- Reduced Costs and Complexity: save communication.
- Quick Processing: Accelerates vote counting and results dissemination.
- Intgrity and Confidentiality: Each vote is secure and private.

Aggios: A new paradigm in voting

- Aggios Voting System components:
  - **Voters**: Individuals who participate in the election process by submitting their votes through a secure interface.
  - **Aggregators**: Collect and tally votes. They use accumulator to ensure that the aggregation process is secure and verifiable.
    - A ZK proof for the subset membership argument, ensuring that votes are valid
  - **Validators:** Independent parties responsible for integrity of the vote tally. Verify the MSA provided by the aggregators.

Proving Subset Membership

- Accumulator for ZK-subset membership
  - Merkle Tree. zk-SNARK (Groth16, Plonk) - uses Hashing (expensive operation). Large constants involved in arithmetizing hash functions
  - RSA accumulator. Verification is linear
- Lookup Table based accumulators (Caulk, Caulk+, etc.)
  - Short proving time
  - Constant size proof and verification time
  - ZK-for multiset (and not subset)

Accumulator based on Lookup Tables

- Lookup argument argument: given a collection of values $c_i \in \overrightarrow{C}$ (called "table") and a collection of values $a_i \in A$, a lookup argument shows that all elements of A occur in C

- Define $Z_{I_j}(X) = \prod_{i \in I_j} (x - i)$

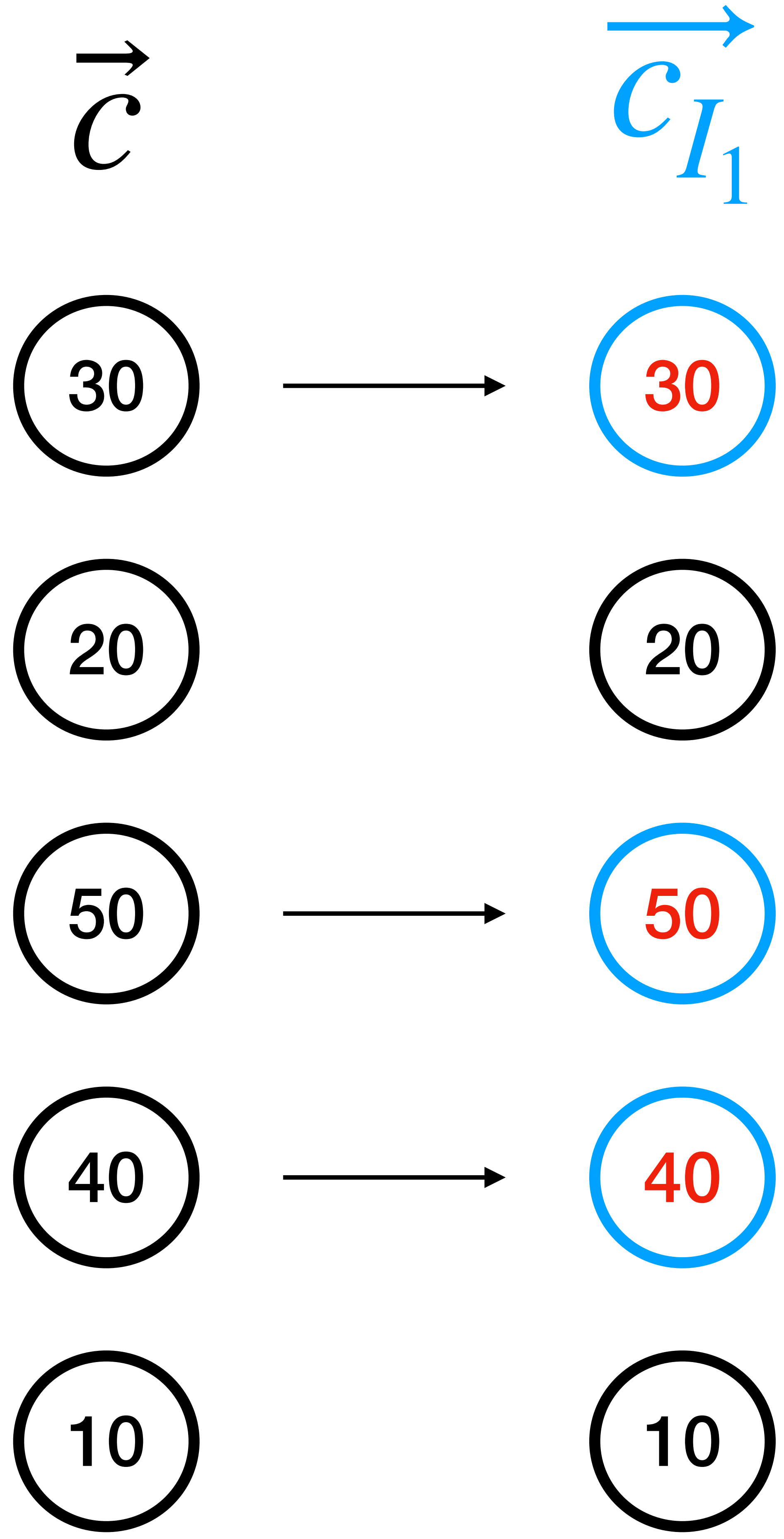
- $C(X) - C_I(X) = 0 \bmod Z_I(X)$

$\vec{c} = (30,20,50,40,10),$

$I = 1,2,3,4,5,6$

$I_1 = \{1,3,4\}$

$\overrightarrow{c_{I_1}} = (30,50,40)$



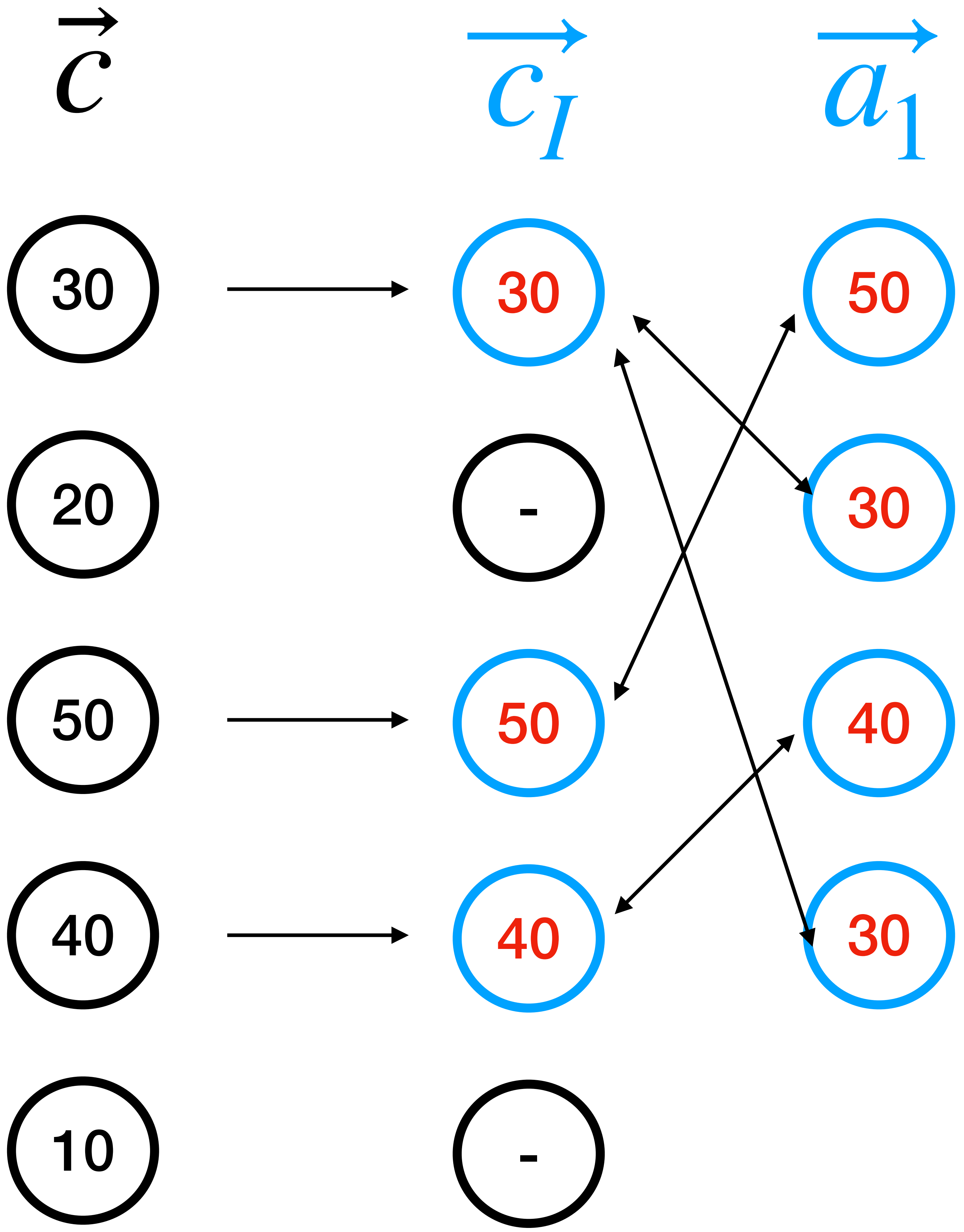
Accumulator based on Lookup Tables

- Lookup argument argument: given a collection of values $c_i \in \overrightarrow{C}$ (called "table") and a collection of values $a_i \in \overrightarrow{A}$, a lookup argument shows that all elements of A occur in C
- Define $Z_{I_j}(X) = \prod_{i \in I_j} (x - i)$
- $C(X) - C_I(X) = 0 \bmod Z_I(X)$
- $A(X) := C_I(U(X))$

$$\vec{c} = (30,20,50,40,10)$$

$$I = \{1,3,4\}$$

$$\overrightarrow{c_I} = (30,50,40), \overrightarrow{A_I} = (30,50,40,30)$$



Voting

Security goals

- Voter's Integrity.
  - does not delegates more than one vote to
    1. the same aggregator
    2. different aggregators
- Aggregator Integrity:
  - only registered voters are allowed to vote
  - correctly aggregates the votes
  - security against non-cooperative voter
- Privacy of the delegated votes:
  - Third parties should not be able to link the elements in $\overrightarrow{C}$ to the elements in $\overrightarrow{A}$

Proposed Scheme: Multi-Subset Membership Argument (MSMA)

- **Setup**: SRS, invertible functions $f_1, \ldots, f_k$ for $f_i : \mathbb{Z}_p \to \mathbb{Z}_p$
- **Witness:** Lookup table $C = (c_1, \ldots, c_n)$
- **Public input:** A public commitment [C] to C
- **Prover outputs**:

  1. k commitments $[A_1]$,…, $[A_k]$ to polynomials $A_1(X)$,…, $A_k(X)$ and their "sizes" $T_1, \ldots, T_k$ (tally) such that $\sum_{j \in [k]} T_j = n$

  2. A Proof Π that for any $c \in C$, there is $a = \Phi(c)$ s.t. $a \in \cup_{j \in [k]}$ and Φ is one-to-one

MSMA - One Subset

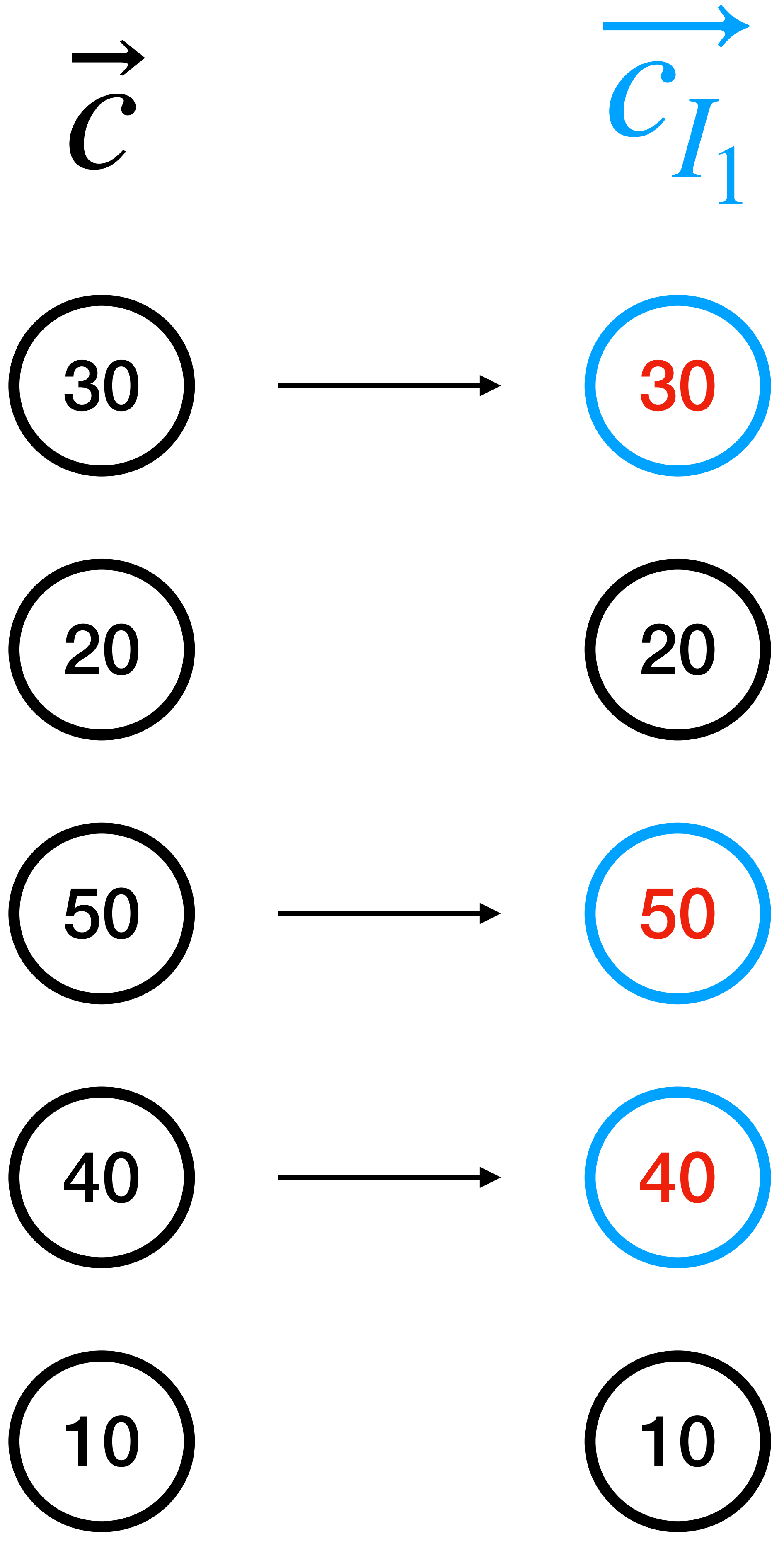
- Define $Z_{I_j}(X) = \prod_{i \in I_j} (x - i)$
- $C(X) - C_{I_j}(X) = 0$ mod $Z_{I_j}(X)$

$\vec{c} = (30,20,50,40,10),$

$I = 1,2,3,4,5,6$

$I_1 = \{1,3,4\}$

$\overrightarrow{c_{I_1}} = (30,0,50,40,0)$



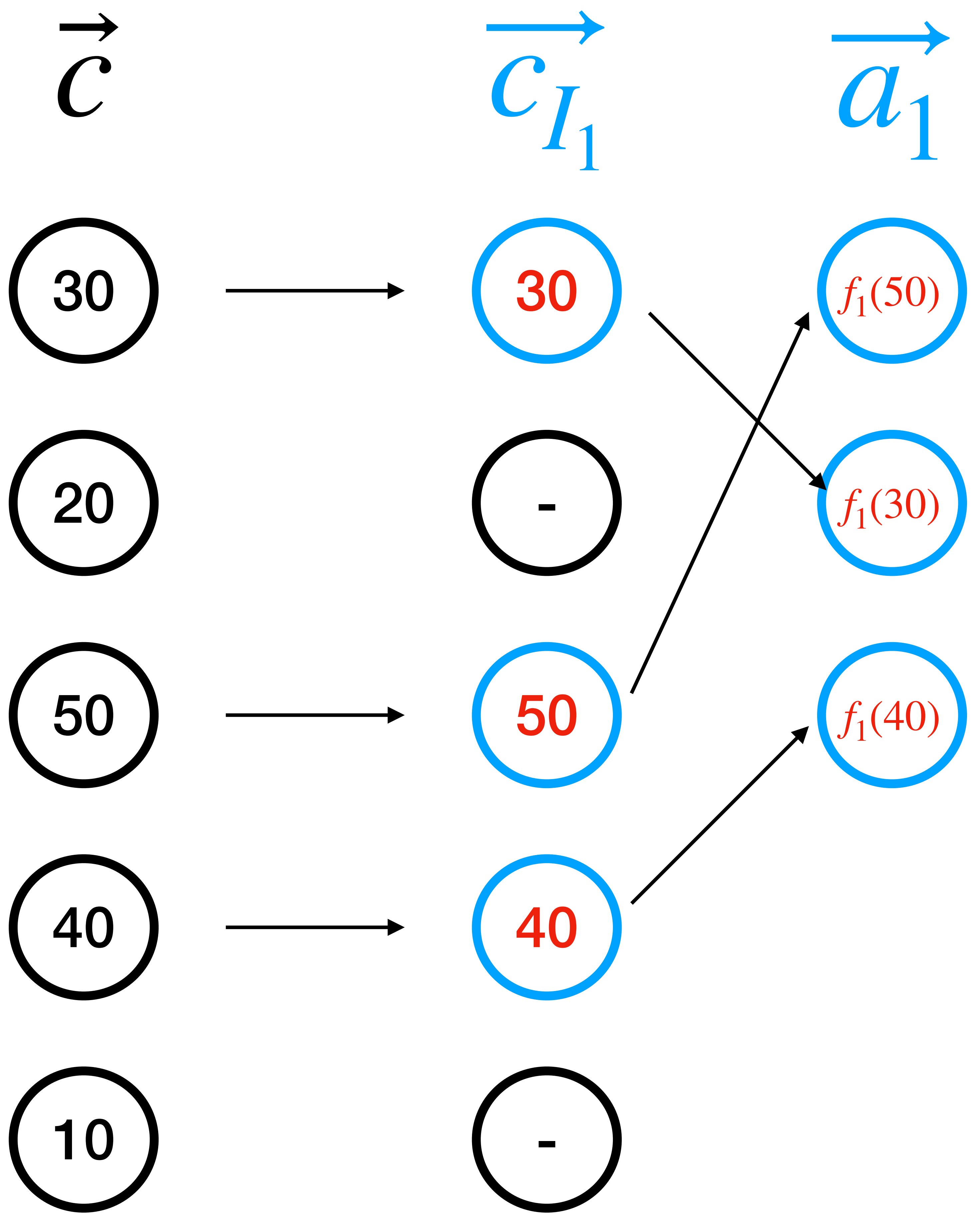
MSMA - adding permutations

- $C(X) - C_{I_j}(X) = 0 \bmod Z_{I_j}(X)$

$\Phi(c_1) = a_2, \Phi(c_3) = a_1, \Phi(c_4) = a_3$

- Choose random mapping $\Pi : [n] \to [\,|A_j|\,]$

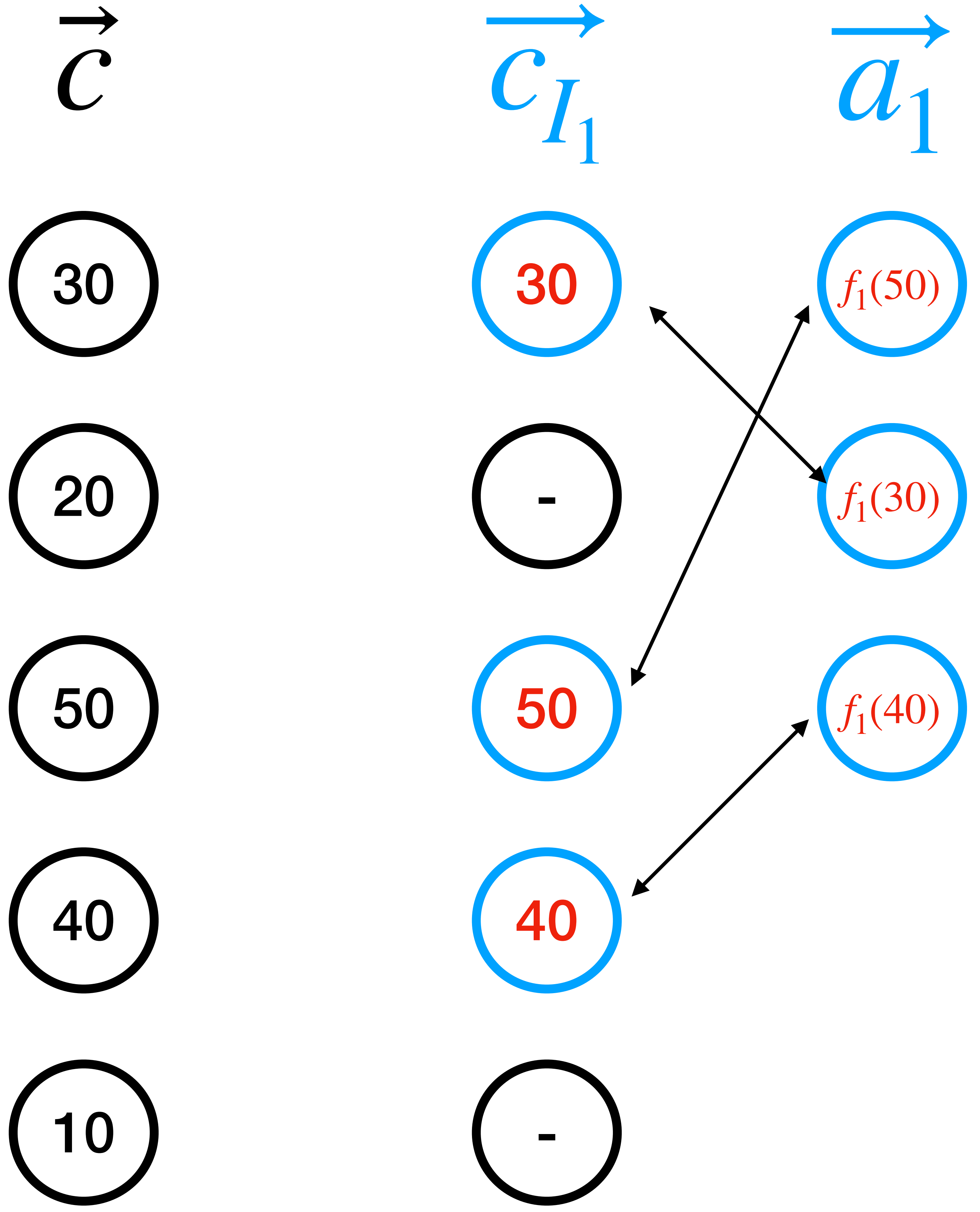
- Define $A_j(X_i) := F_j(C_{I_j}(\Pi(X_i)))$



MSMA

Variant of Grand-Product Argument (Sonic, Plonk)

- Replace $A_j(X) := f_j(C_{I_j}(U(X)))$ with grand product argument:
  - $h_j(X) = A_j(X) + S_{ID}(X)\beta_j + \gamma_j$
  - $g_j(X) = f_j(C_{I_j}(X)) + S_{ID}(\Pi(X))\beta_j + \gamma_j$
  - $h_j(X)z_j(X) - g_j(X)z_j(X+1) = 0 \mod Z_{V_j}(X)$



MSMA - Multiple Subsets

$$I = \{I_1 = \{1,3,4\}, I_2 = \{5\}\}$$

$$C(X) - C_{I_1}(X) = 0 \text{ mod } Z_{I_1}(X)$$

$$C(X) - C_{I_2}(X) = 0 \text{ mod } Z_{I_2}(X)$$

$$\overrightarrow{c_{I_1}} = (30,0,50,40,0)$$

$$\overrightarrow{c_{I_2}} = (0,20,50,0,10)$$

$$I_1 \cap I_2 \neq \emptyset$$

$\vec{c}$	$\overrightarrow{c_{I_1}}, \overrightarrow{c_{I_2}}$
30	30
20	20
50	50
40	40
10	10

MSMA - Multiple Subsets

$$I = \{I_1 = \{1,3,4\}, I_2 = \{2,5\}\}$$

1. $C(X) - \sum_{j \in k} C_{I_j}(X) = 0 \bmod Z_I(X)$
2. $\forall\, j \in [k], C(X) - C_{I_j}(X) = 0 \bmod Z_{I_j}(X)$

$$\overrightarrow{c_{I_1}} = (30,0,50,40,0)$$

$$\overrightarrow{c_{I_2}} = (0,20,0,0,10)$$

$$I_{j_1} \cap I_{j_2} = \varnothing$$

$\vec{c}$	$\overrightarrow{c_{I_1}}, \overrightarrow{c_{I_2}}$
30	30
20	20
50	50
40	40
10	10

MSMA - Multiple Subsets

1. $C(X) - \sum_{j \in k} C_{I_j}(X) = 0 \bmod Z_I(X)$
2. $\forall\, j \in [k], C(X) - C_{I_j}(X) = 0 \bmod Z_{I_j}(X)$

$$\vec{c} = (30,20,50,40,10)$$

$$I = \{I_1 = \{1,3\}, I_2 = \{2,4,5\}, I_3 = \{1,2,3,4,5\}\}$$

$\overrightarrow{c_{I_1}} = (30,0,50,40,0)$ $\quad$ $\overrightarrow{c_{I_2}} = (0,20,0,0,10)$ $\quad$ $\overrightarrow{c_{I_3}} = (30,20,50,40,10)$

$$I_1 \cap I_3 \neq \emptyset$$

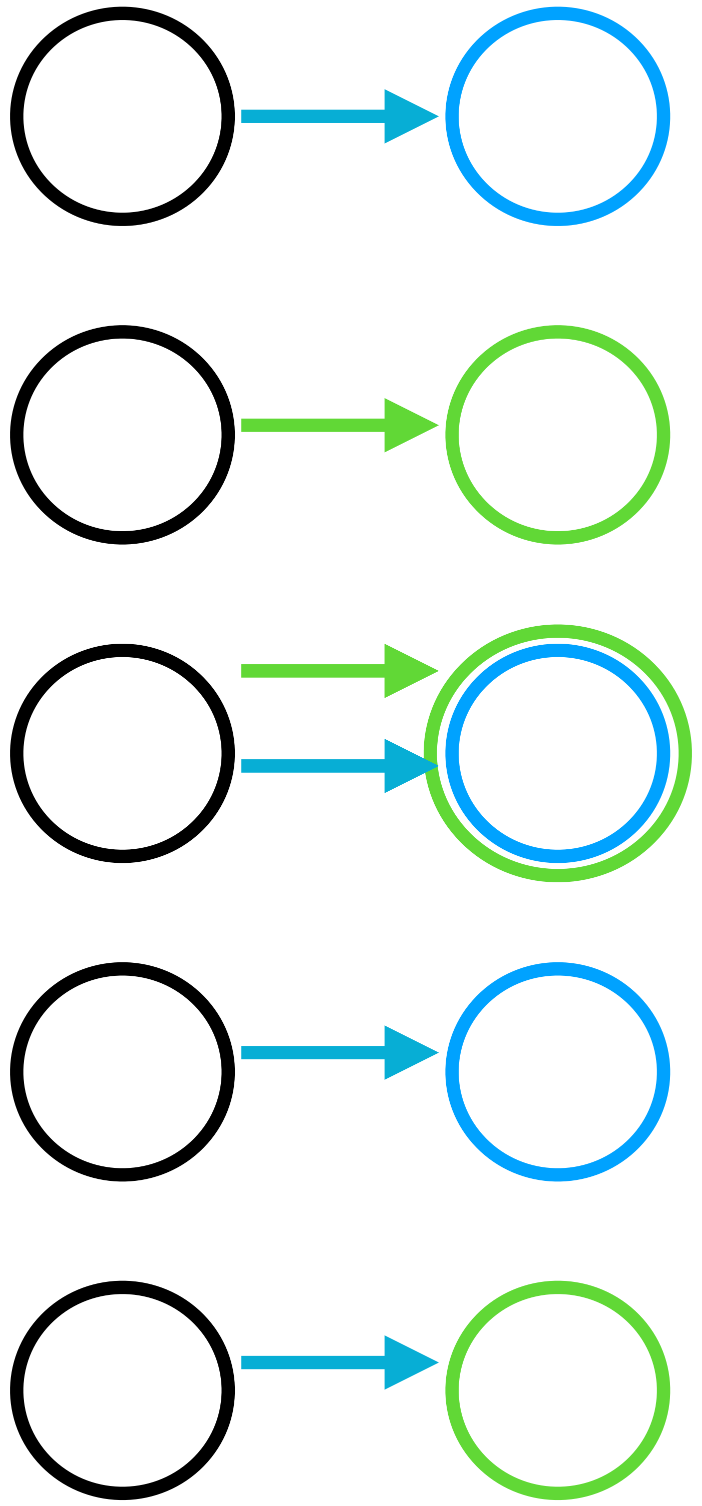
The set $\{\overrightarrow{c_{I_j}}\}_{j \in [k]}$ covers $\vec{c}$

MSMA - Multiple Subsets

1. $C(X) - \sum_{j \in k} C_{I_j}(X) = 0 \bmod Z_I(X)$
2. $\forall\, j \in [k], C(X) - C_{I_j}(X) = 0 \bmod Z_{I_j}(X)$

$$I = \{I_1, I_2\}$$

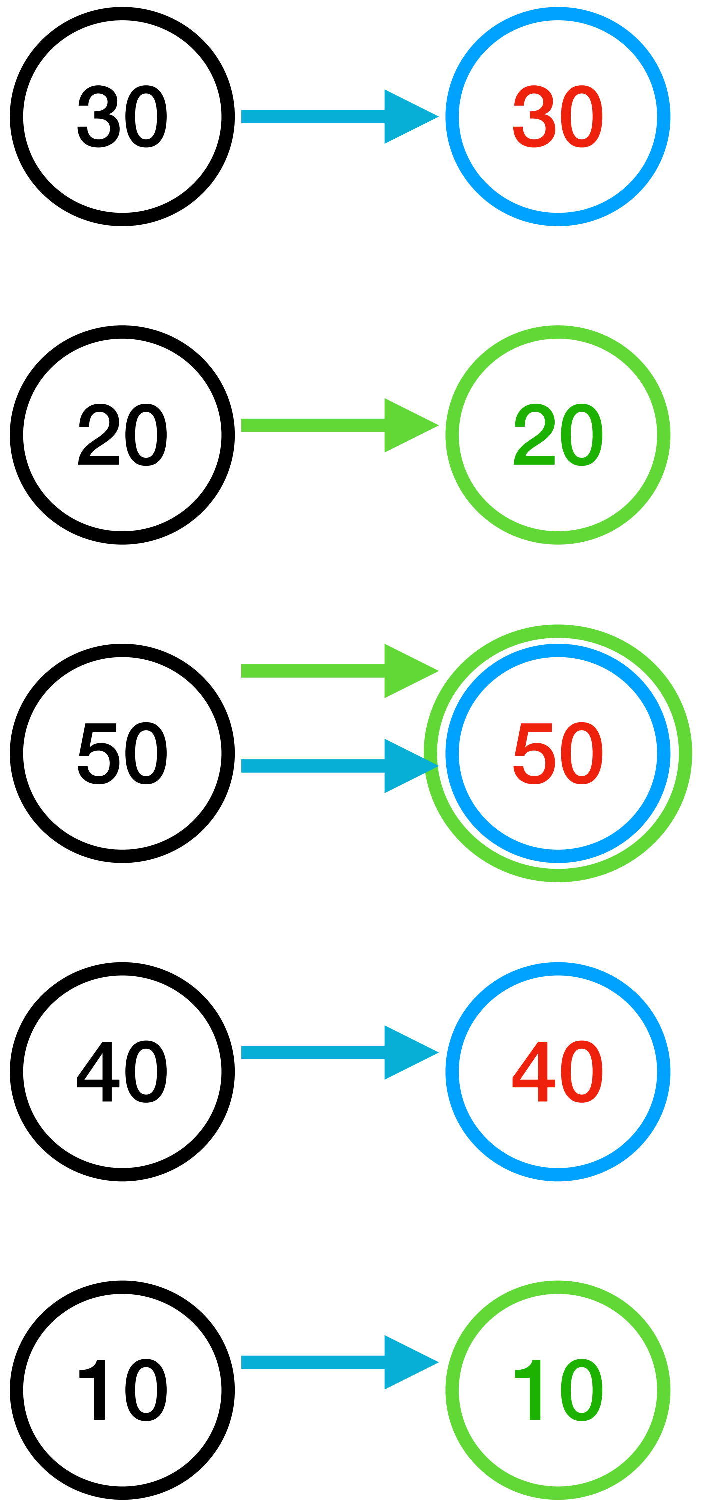
$$I_1 \cap I_2 \neq \varnothing$$



Multi-Subset Membership Argument

- Choose a secret partition of [n], $I_1, \ldots I_k$
- publish KZG commitments $[C_{I_1}], \ldots, [C_{I_k}]$ and $[A_1], \ldots, [A_k]$ s.t.,
  - Prove that C and $\cup_{j\in[k]} C_{I_j}$ agree on all element in [n]:

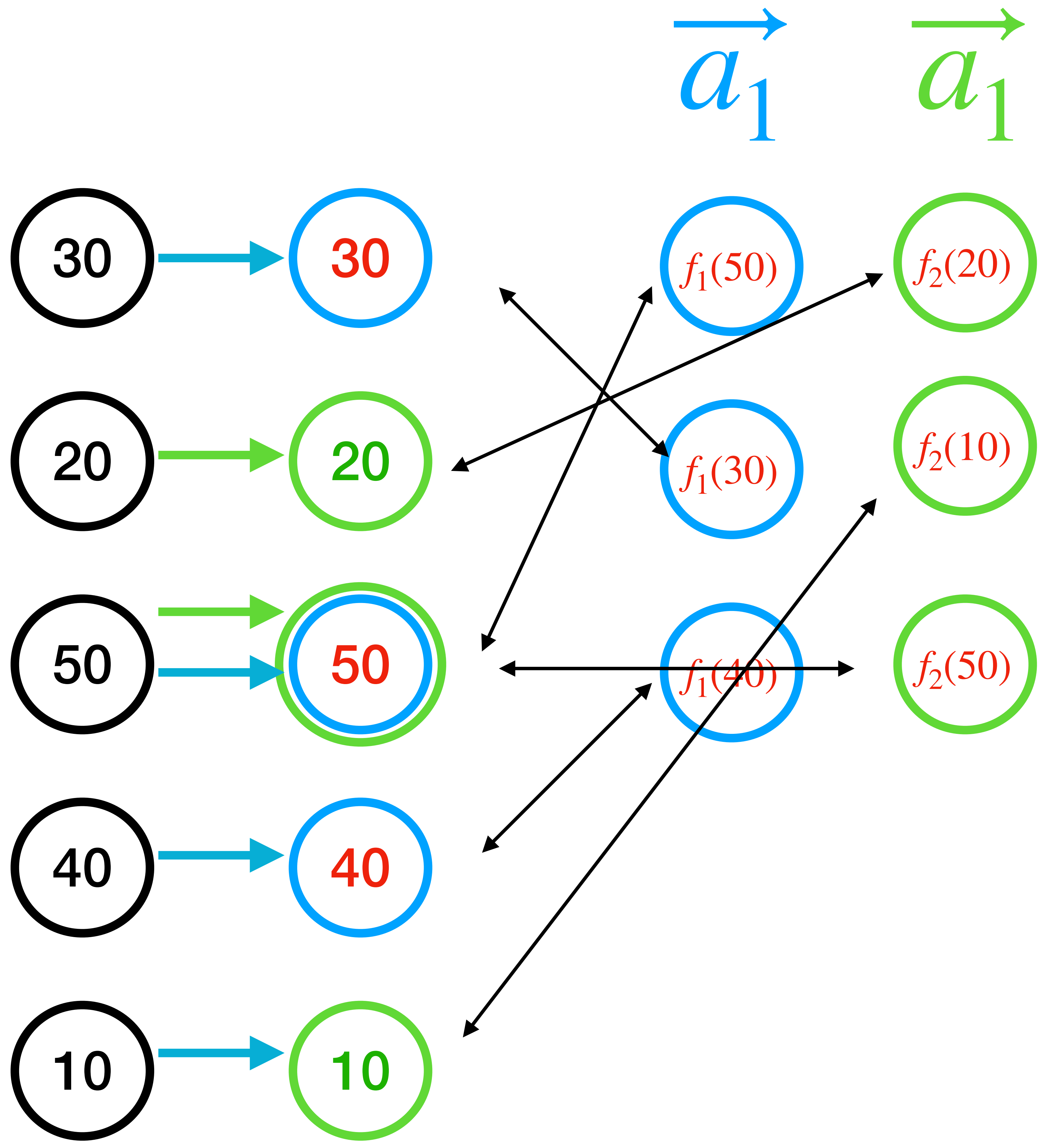
1. $$C(X) - \sum_{j\in k} C_{I_j}(X) = 0 \bmod Z_I(X)$$
2. $$\forall\, j \in [k], C(X) - C_{I_j}(X) = 0 \bmod Z_{I_j}(X)$$
3. Permutation argument
   - $h_j(X) = A_j(X) + S_{ID}(X)\beta_j + \gamma_j$
   - $g_j(X) = f_i(C_{I_j}(X)) + S_{ID}(U(X))\beta_j + \gamma_j$
   - $h_j(X)z_j(X) - g_j(X)z_j(X+1) = 0 \mod Z_{V_j}(X)$



Multi-Subset Membership Argument

- Choose a secret partition of [n], $I_1, ... I_k$
- publish KZG commitments $[C_{I_1}], ..., [C_{I_k}]$ and $[A_1], ..., [A_k]$ s.t.,
  - Prove that C and $\cup_{j\in[k]} C_{I_j}$ agree on all element in [n]:

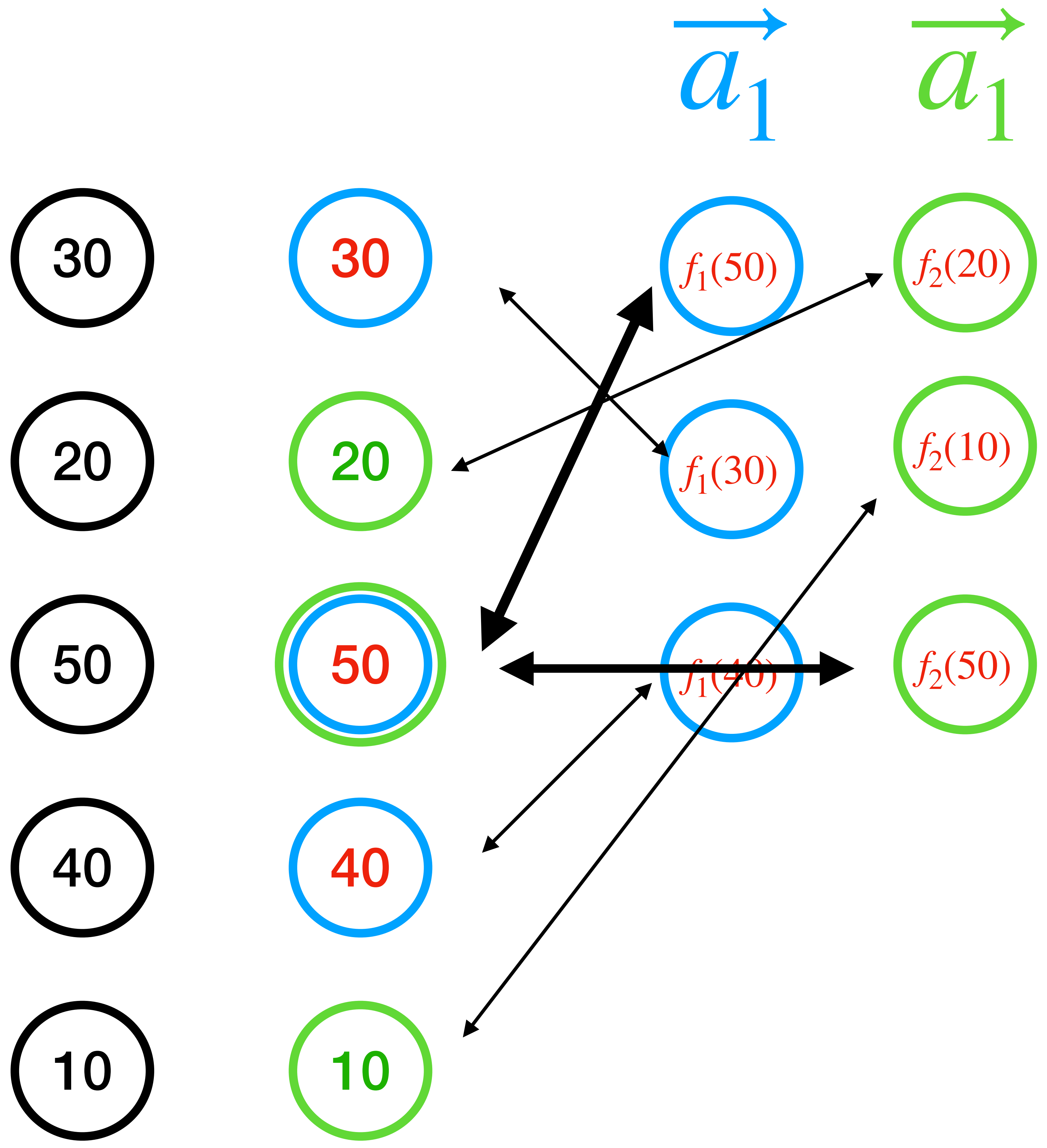
1. $C(X) - \sum_{j\in k} C_{I_j}(X) = 0 \bmod Z_I(X)$
2. $\forall\, j \in [k], C(X) - C_{I_j}(X) = 0 \bmod Z_{I_j}(X)$
3. Permutation argument
   - $h_j(X) = A_j(X) + S_{ID}(X)\beta_j + \gamma_j$
   - $g_j(X) = f_i(C_{I_j}(X)) + S_{ID}(U(X))\beta_j + \gamma_j$
   - $h_j(X)z_j(X) - g_j(X)z_j(X+1) = 0 \quad \bmod Z_{V_j}(X)$



Multi-Subset Membership Argument

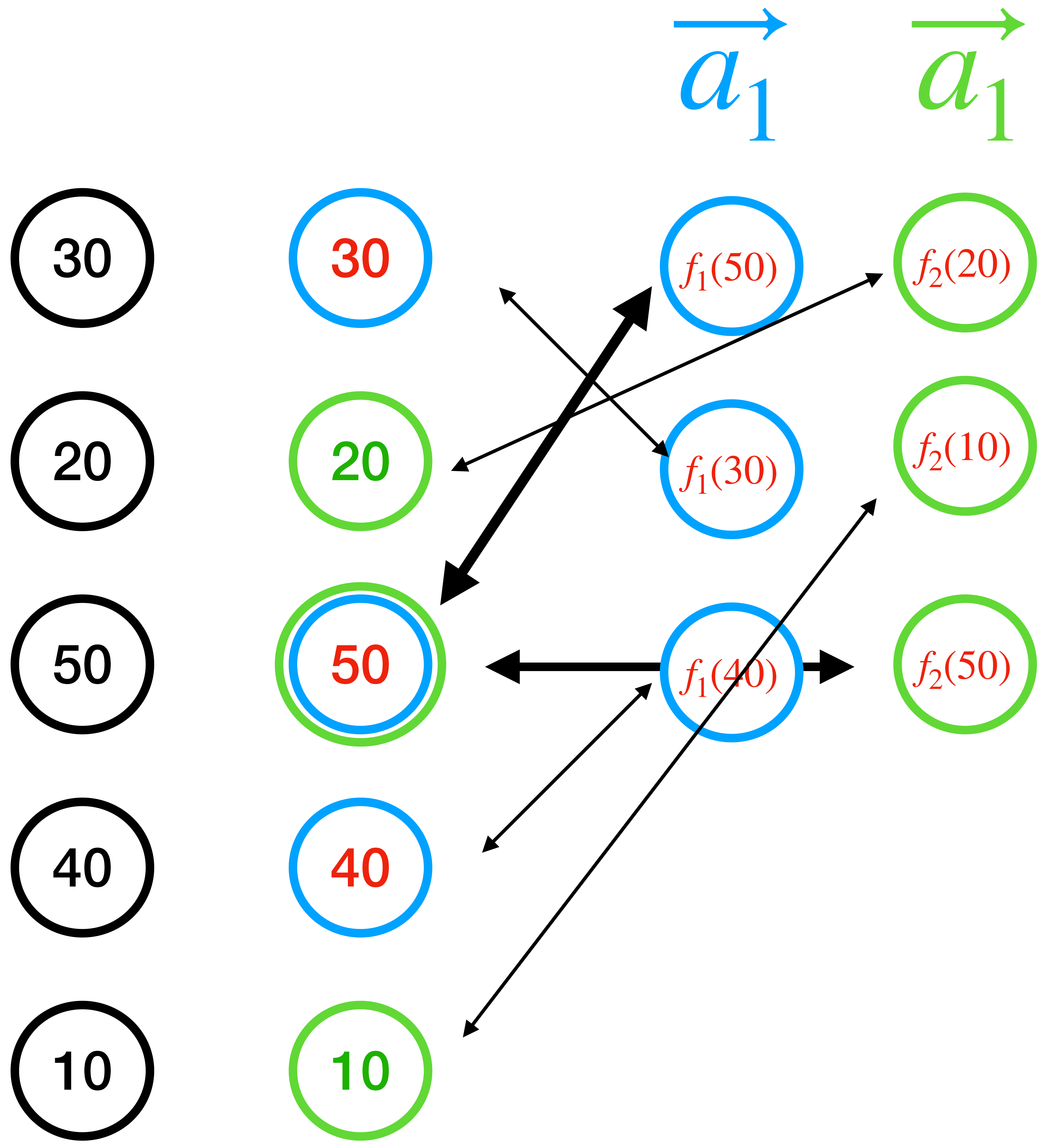
- Choose a secret partition of [n], $I_1, \ldots I_k$
- publish KZG commitments $[C_{I_1}], \ldots, [C_{I_k}]$ and $[A_1], \ldots, [A_k]$ s.t.,
    - Prove that C and $\cup_{j \in [k]} C_{I_j}$ agree on all element in [n]:

1. $$C(X) - \sum_{j \in k} C_{I_j}(X) = 0 \bmod Z_I(X)$$
2. $$\forall\ j \in [k], C(X) - C_{I_j}(X) = 0 \bmod Z_{I_j}(X)$$
3. Permutation argument
    - $h_j(X) = A_j(X) + S_{ID}(X)\beta_j + \gamma_j$
    - $g_j(X) = f_i(C_{I_j}(X)) + S_{ID}(U(X))\beta_j + \gamma_j$
    - $h_j(X)z_j(X) - g_j(X)z_j(X+1) = 0 \quad \bmod Z_{V_j}(X)$



Multi-Subset Membership Argument

- Verifier checks: $\sum_{j\in[k]} T_j = n$
- If the intersection is not empty then $\sum_{j\in[k]} T_j > n$

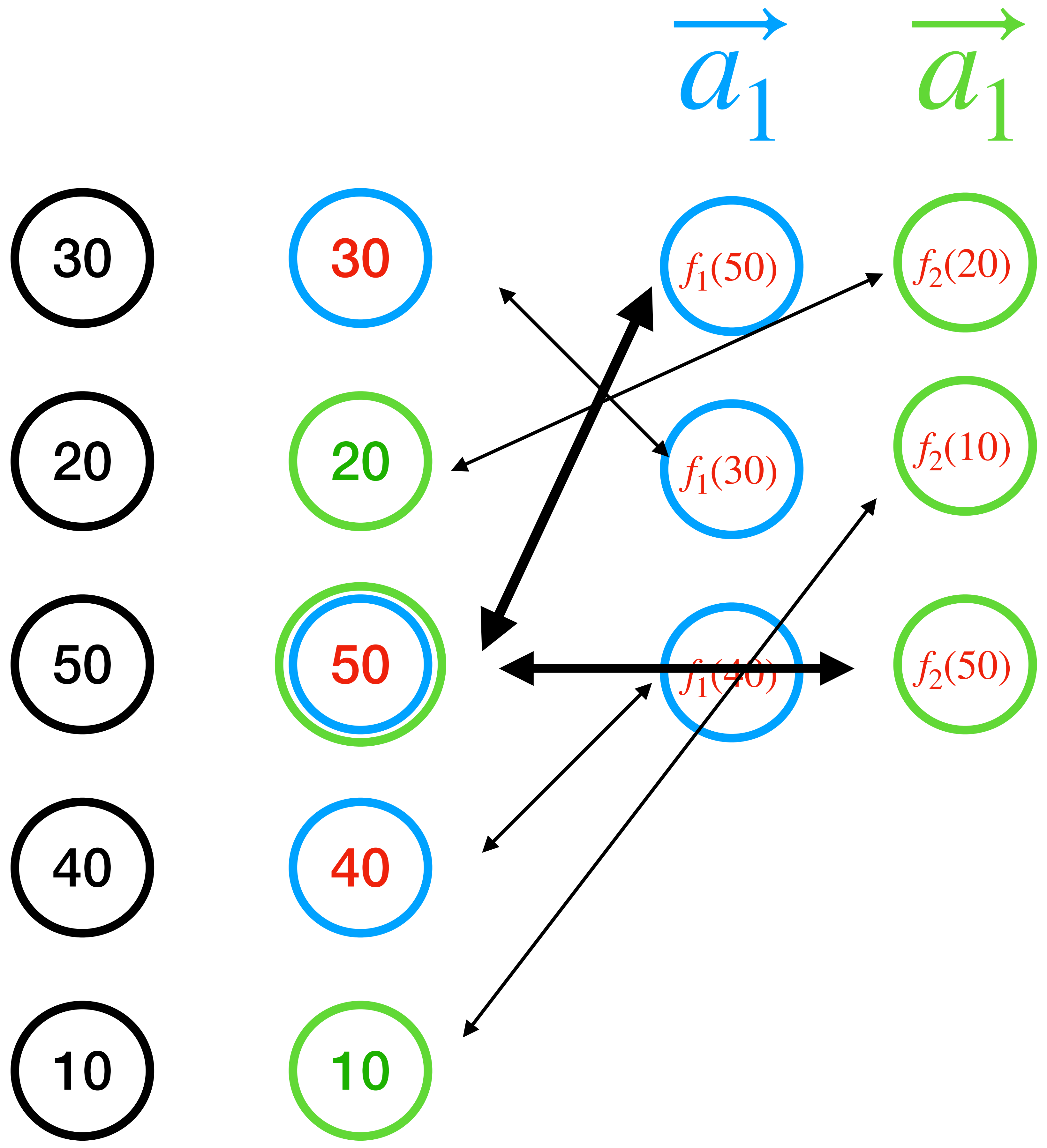


One to One Correspondence

Show that $\Phi(c_i) \neq \Phi(c_j)$

- If there are $j \neq j'$ such that $c \in C_{I_j}, c \in C_{I_{j'}}$ then $\sum_{j \in [k]} T_j > n$

Check: 3+3>5

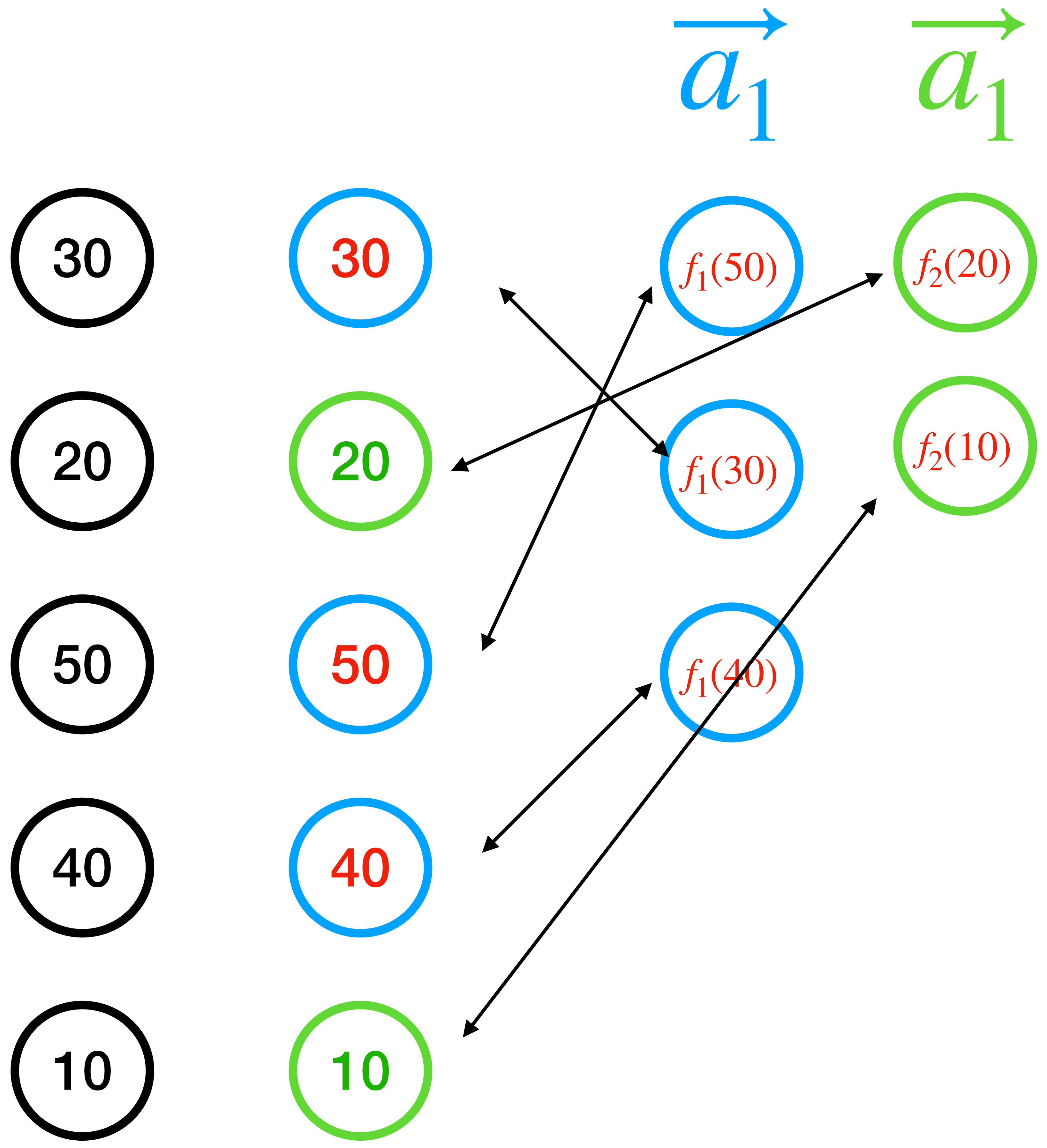


Multi-Subset Membership Argument

- Choose a secret partition of [n], $I_1, ... I_k$
- publish KZG commitments $[C_{I_1}], \ldots, [C_{I_k}]$ and $[A_1], \ldots, [A_k]$ s.t.,
    - Prove that C and $\cup_{j\in[k]} C_{I_j}$ agree on all element in [n]:

1. $C(X) - \sum_{j\in k} C_{I_j}(X) = 0 \text{ mod } Z_I(X)$
2. $\forall\, j \in [k], C(X) - C_{I_j}(X) = 0 \text{ mod } Z_{I_j}(X)$
3. Permutation argument
    - $h_j(X) = A_j(X) + S_{ID}(X)\beta_j + \gamma_j$
    - $g_j(X) = f_i(C_{I_j}(X)) + S_{ID}(U(X))\beta_j + \gamma_j$
    - $h_j(X)z_j(X) - g_j(X)z_j(X+1) = 0 \quad \text{mod } Z_{V_j}(X)$

- Verifier checks: $\sum_{j\in[k]} T_j = n$



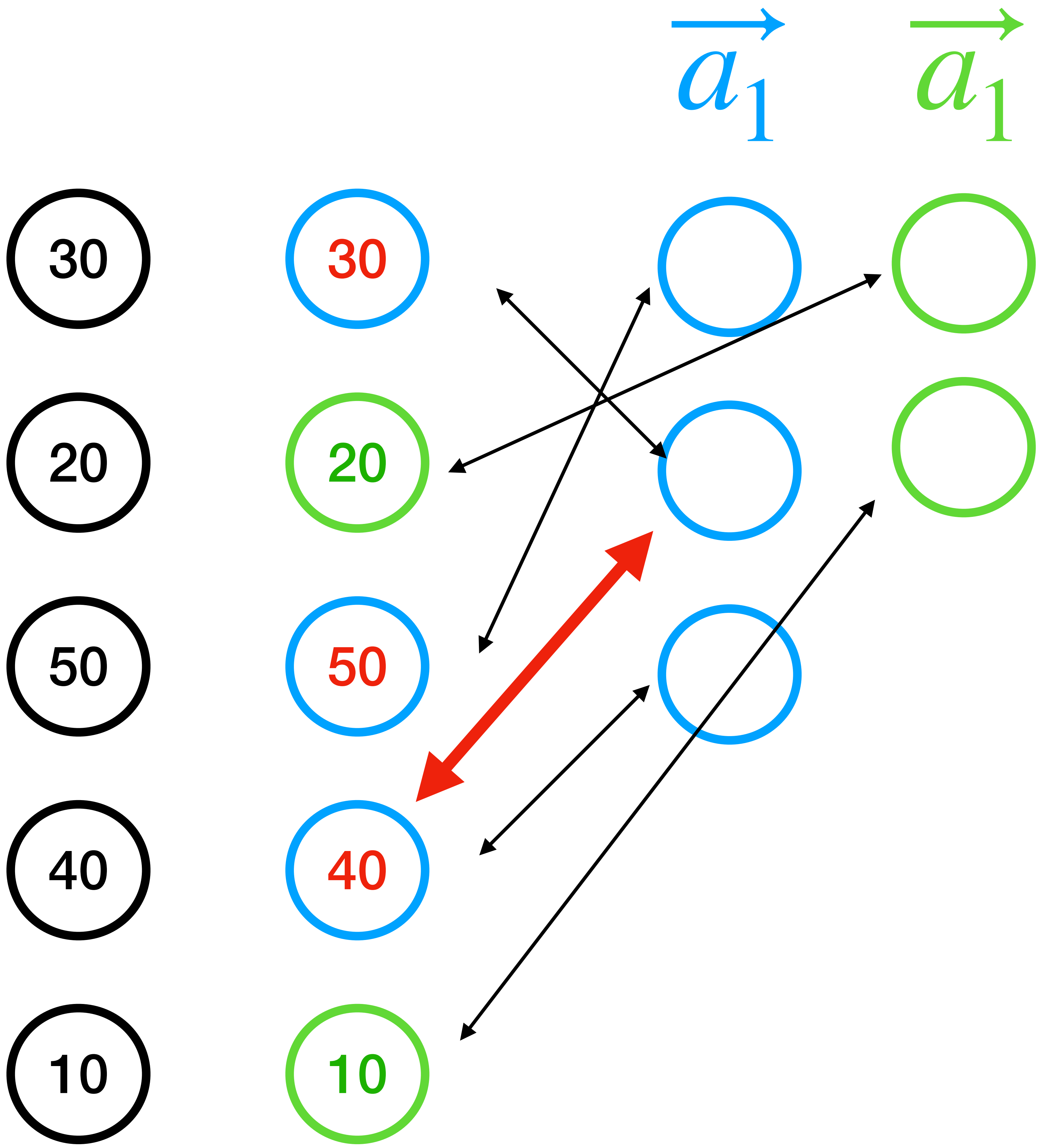
Checks: 3+2=5

Multi-Subset Membership Argument

- Prover:
  - Choose a secret partition of [n], $I_1, \ldots I_k$
  - publish KZG commitments $\{[C_{I_j}]\}_{j\in[k]}, \{[A_{I_j}]\}_{j\in[k]}$, and proofs $\{[W_{I_j}]\}_{j\in[k]}, [W_I]$
- Verifier:
  - $\forall j \in [k], e(([C] - [C_{I_j}]) + \alpha^j[x^n - 1], [1]) = e([Z_j], [W_j])$
  - $e(([C] - \sum_{j\in k} [C_{I_j}]) + \alpha^{j+1}[x^n - 1], [1]) = e([Z_I], [W_I])$

MSMA

- If there are two element c_1, c_2 such that $\Phi(c_1) = \Phi(c_2) = a$
  - c_1, c_2 cannot be from the same (permutation)
  - => $c_1 \in C_{I_1}, c_2 \in C_{I_2}$

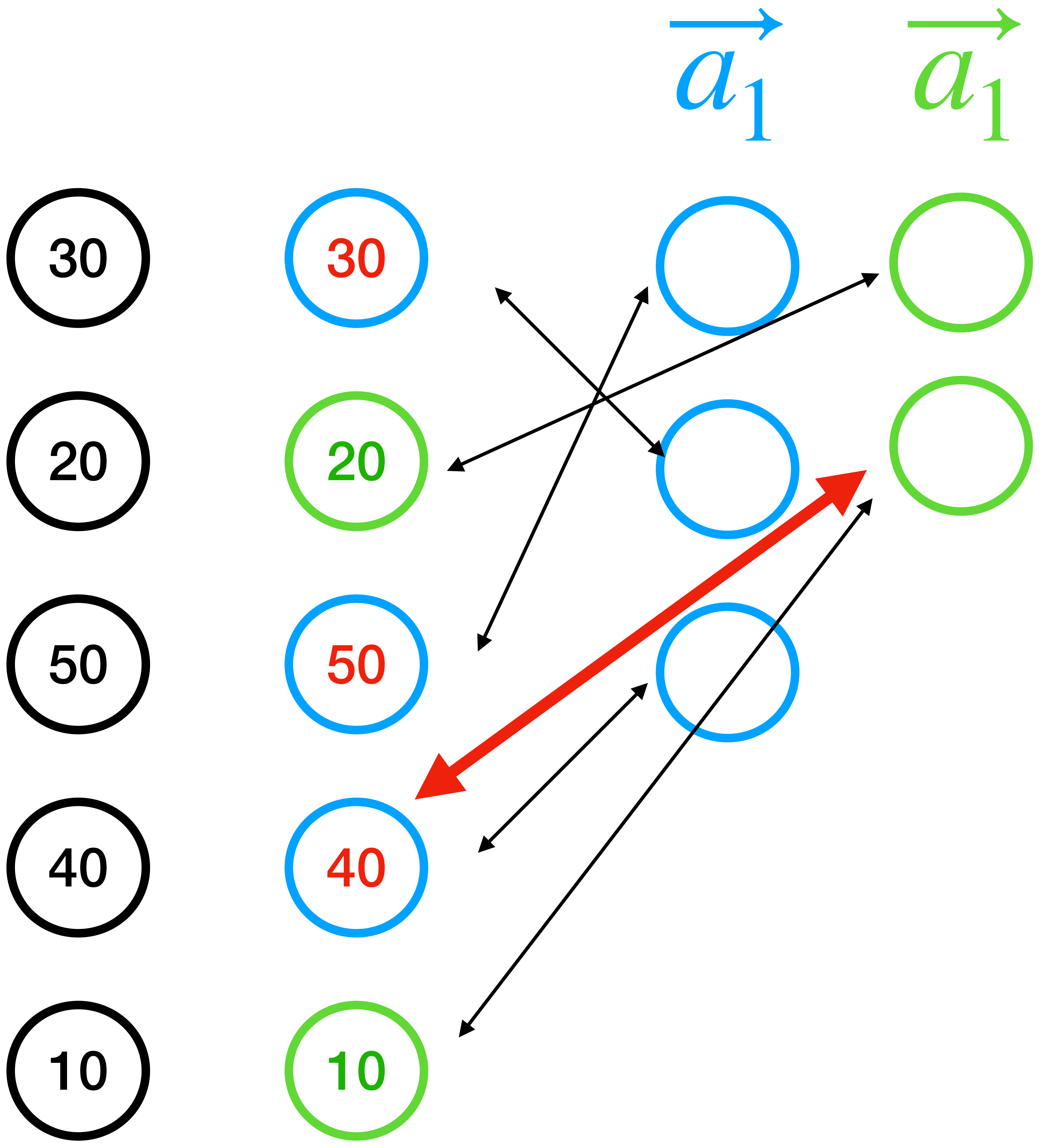


Checks: 3+2=5

MSMA

- If there are two element c_1, c_2 such that $\Phi(c_1) = \Phi(c_2) = a$
  - c_1, c_2 cannot be from the same (permutation)
  - => $c_1 \in C_{I_1}, c_2 \in C_{I_2}$

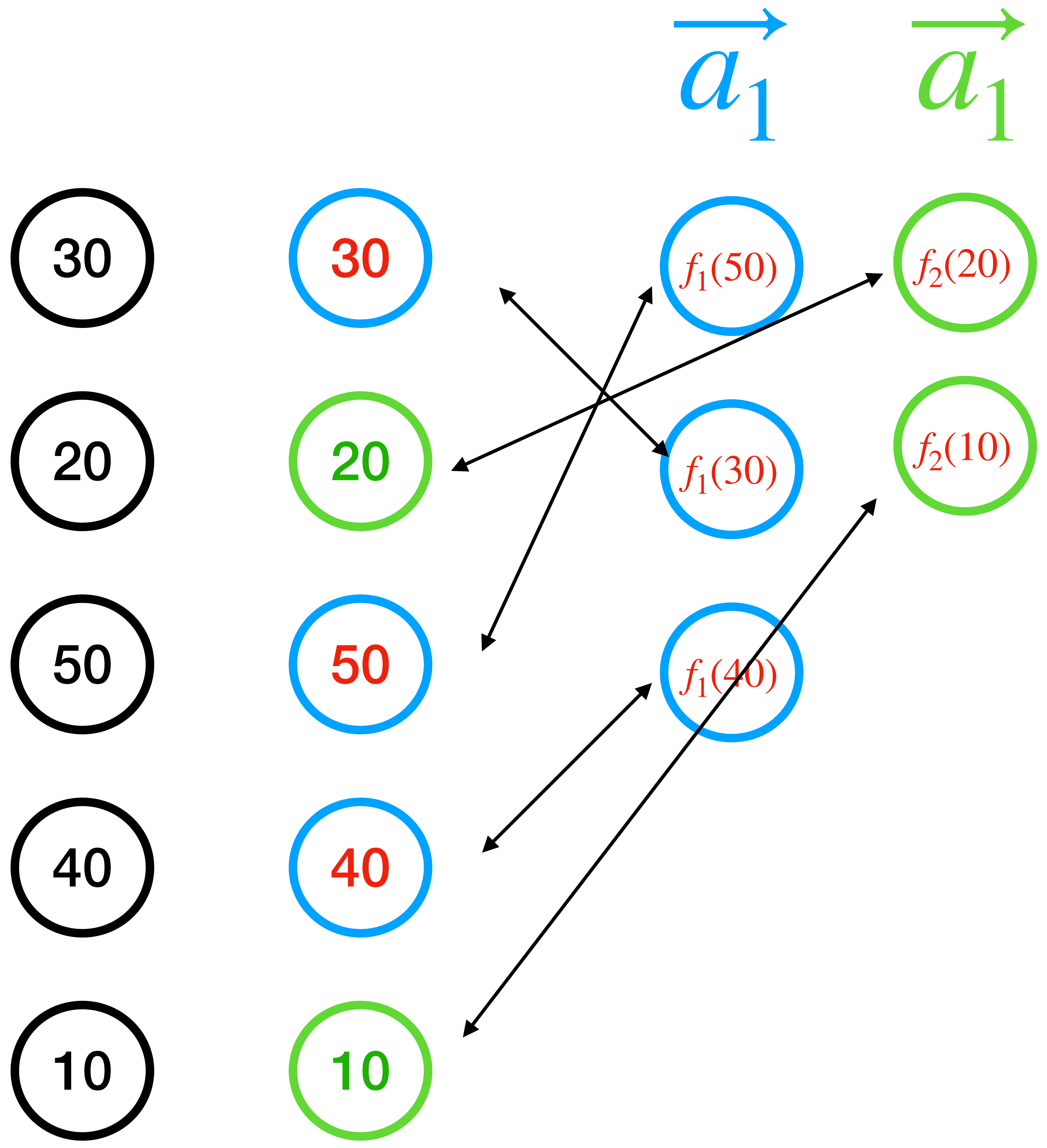
  For every $c_i \in C, c_i \in \cup_{j\in[k]} C_{I_j}$
  - => $\sum_{j\in[k]} |a_j| = \sum_{j\in[k]} |T_j| > k$



Multi-Subset Membership Argument

- Choose a secret partition of [n], $I_1, \ldots I_k$
- publish KZG commitments $[C_{I_1}], \ldots, [C_{I_k}]$ and $[A_1], \ldots, [A_k], T_1, \ldots, T_k$ s.t.,
  - $C(X) - \sum_{j \in k} C_{I_j}(X) = 0$ mod $Z_I(X)$ (Lagrangian of I_j's are over large basis)
  - Permutation argument
    - $f_j(X) = A_j(X) + S_{ID}(X)\beta_j + \gamma_j$
    - $g_j(X) = C_{I_j}(X)^{t_j} + S_{ID}(U(X))\beta_j + \gamma_j$
    - $f_j(X)z_j(X) - g_j(X)z_j(X+1) = 0 \mod Z_{T_j}(X)$
- Verifier checks: $\sum_{j \in [k]} T_j = n$
  - => $\forall j_1 \neq j_2, I_{j_1} \cap I_{j_2} = \varnothing$
  - $\forall c_1 \neq c_2 \in C, \Phi(c_1) \neq \Phi(c_2)$

One to one correspondence!



Overview of the Voting Scheme

Protocol Phases

- **Setup Phase.**
  - Establish SRS and PKI
- **Registration.**
  - Voters register their voting keys with their chosen aggregator. This phase includes:
    - Sharing of secret voting keys between voters and aggregators.
    - Use of aggregatable signatures to prevent a voter from registering with multiple aggregators and to simplify the validation process.
- **Voting.**
  - Voters delegate their vote over a private channel
  - Aggregators submit the local tally with multi-shuffle argument of knowledge.

Operational Highlights:

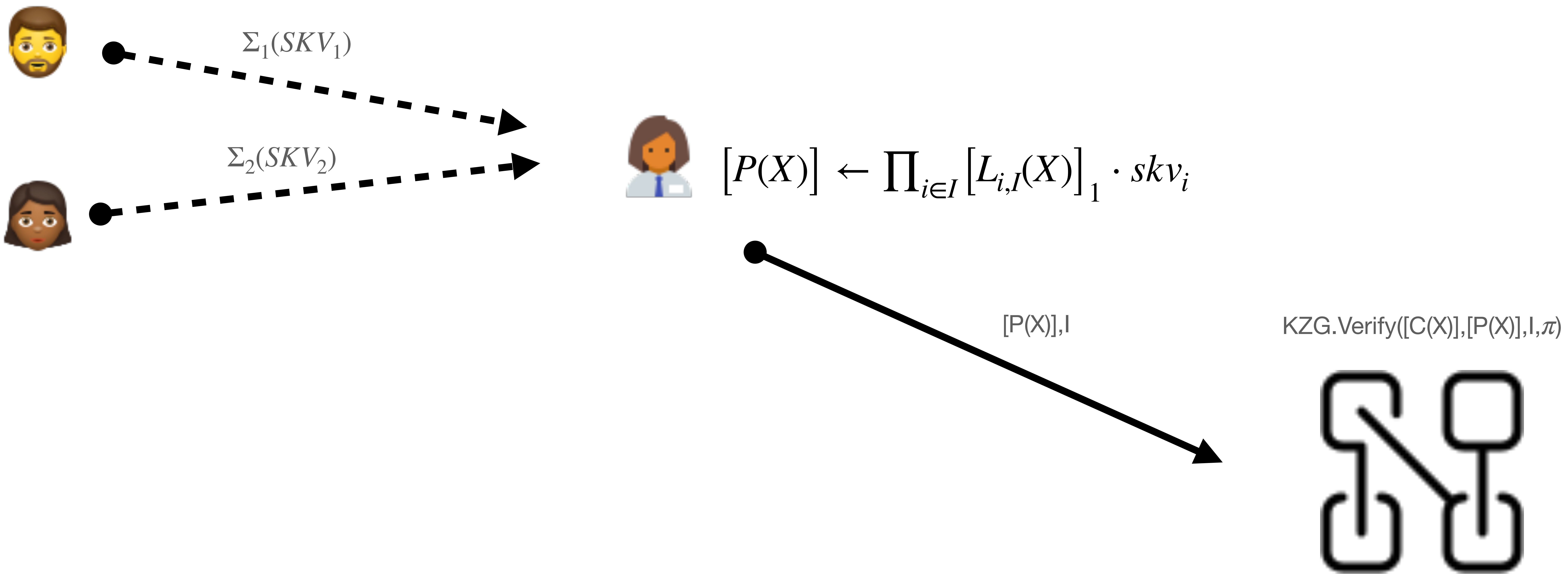
- If voters change their aggregator, they generate new voting keys, enhancing security and flexibility.
- All interactions and transactions are cryptographically signed to ensure non-repudiation and integrity.

Voting Protocol

Setup and Registration

Setup: $\Sigma_1(PKV_1), \ldots, \Sigma_n(PKV_n)$, KZG.SRS

Publicly compute polynomial commitment $C(X)$ from $PKV_1, \ldots, PKV_n$

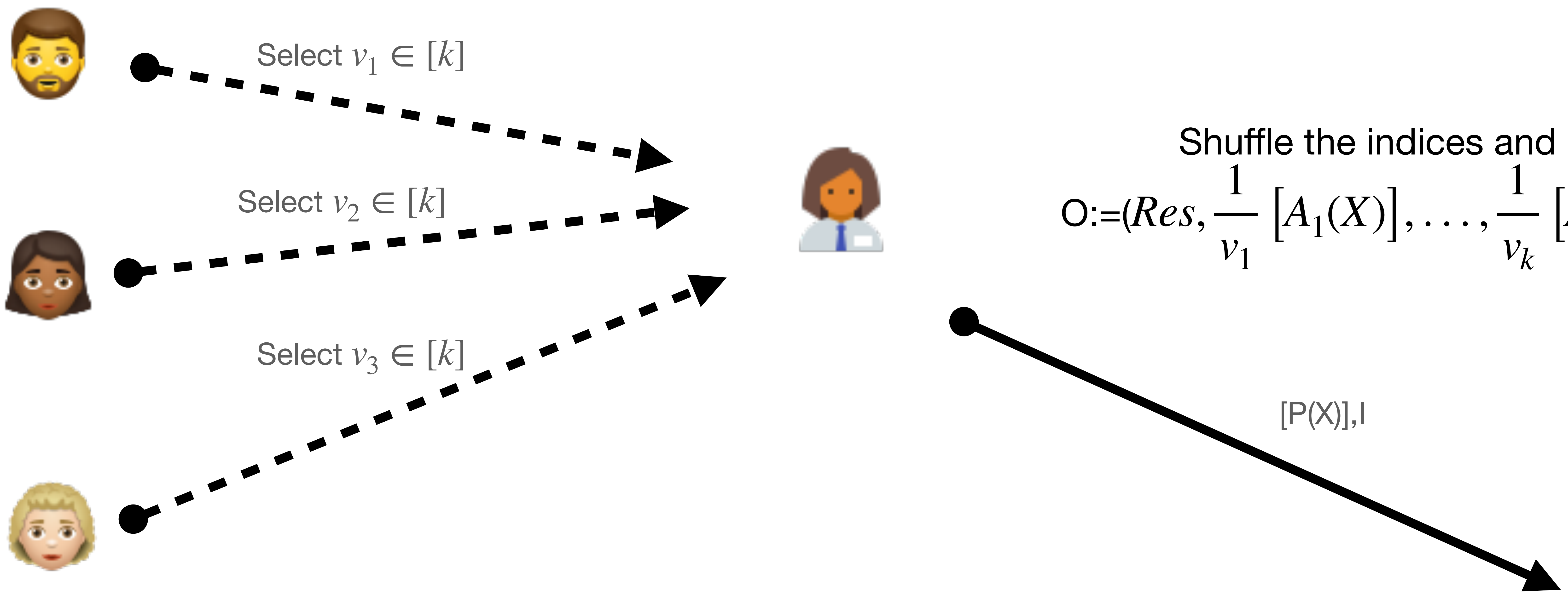


Voting Protocol

Voting and Aggregating

Setup: $\Sigma_1(PKV_1), \ldots, \Sigma_n(PKV_n)$, KZG.SRS

Publicly compute polynomial commitment $C(X)$ from $PKV_1, \ldots, PKV_n$



Shuffle the indices and compute

O:=$(Res, \frac{1}{v_1}\left[A_1(X)\right], \ldots, \frac{1}{v_k}\left[A_k(X)\right], \Pi_{MSMA})$

Performance

- Proof time O(k) (independent in number of gates)
  - (8k+2) G_1
  - 1 G_2 for [W]
  - Verifier time: 4k pairings + 2

Protocol	Proof size	Proof time	Verification time	Trusted setup	Succinct	Post-quantum
Groth16	$2\ \mathbb{G}_1$, $1\ \mathbb{G}_2$	$3n+m-\ell\mathbb{G}_1\ exp$, $n\ \mathbb{G}_2\ exp$	$3\ P, \ell\ G_1$ exp	T, per-circuit	✓	✗
Plonk	$7\ \mathbb{G}_1$,$7\ \mathbb{F}$	$11(n+a)\ G_1\ exp, 54(n+a)log(n+a)\ F$	$2P$, $18\ G_1$ exp	T, U, Up	✓	✗
Plonk (fast prover)	$9\ \mathbb{G}_1$,$7\ \mathbb{F}$	$9(n+a)\ G_1\ exp, 54(n+a)log(n+a)\ F$	$2P$, $16\ G_1$ exp	T,U,Up	✓	✗

The number of wires is denoted by m. KoE stands for Knowledge of Exponent. P denotes pairing computation and ℓ is the number of public input. T stands for Trusted, U for Universal and Up for Updatable.