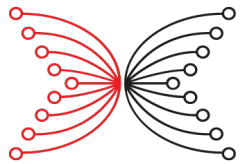
SNACKs: Leveraging Proofs of Sequential Work for Blockchain Light Clients

Hamza Abusalah[1], Georg Fuchsbauer[1], Peter Gaži[2], Karen Klein[3]



INPUT | OUTPUT

ETH zürich

Asiacrypt 2022 in Taipei, Taiwan

Outline

Light Client Blockchain Bootstrapping

Define SNACKs
(Succinct Non-interactive Argument of Chain Knowledge)

Construct SNACKs
(generically from Proofs of Sequential Work Schemes)

SNACKs to Bootstrap, generically

(Light-Client) Blockchain Protocols

Full nodes ($\mathcal{P}$)

storing the entire C

Light clients ($\mathcal{V}$)

minimal chain info C

(Light-Client) Blockchain Protocols

Full nodes ($\mathcal{P}$)

storing the entire C

Light clients ($\mathcal{V}$)

minimal chain info C

- sublinear effort ($\mathcal{V}$)
- sublinear communication
- non-interactive

(Light-Client) Blockchain Protocols

Full nodes ($\mathcal{P}$)
storing the entire C

Light clients ($\mathcal{V}$)
minimal chain info C

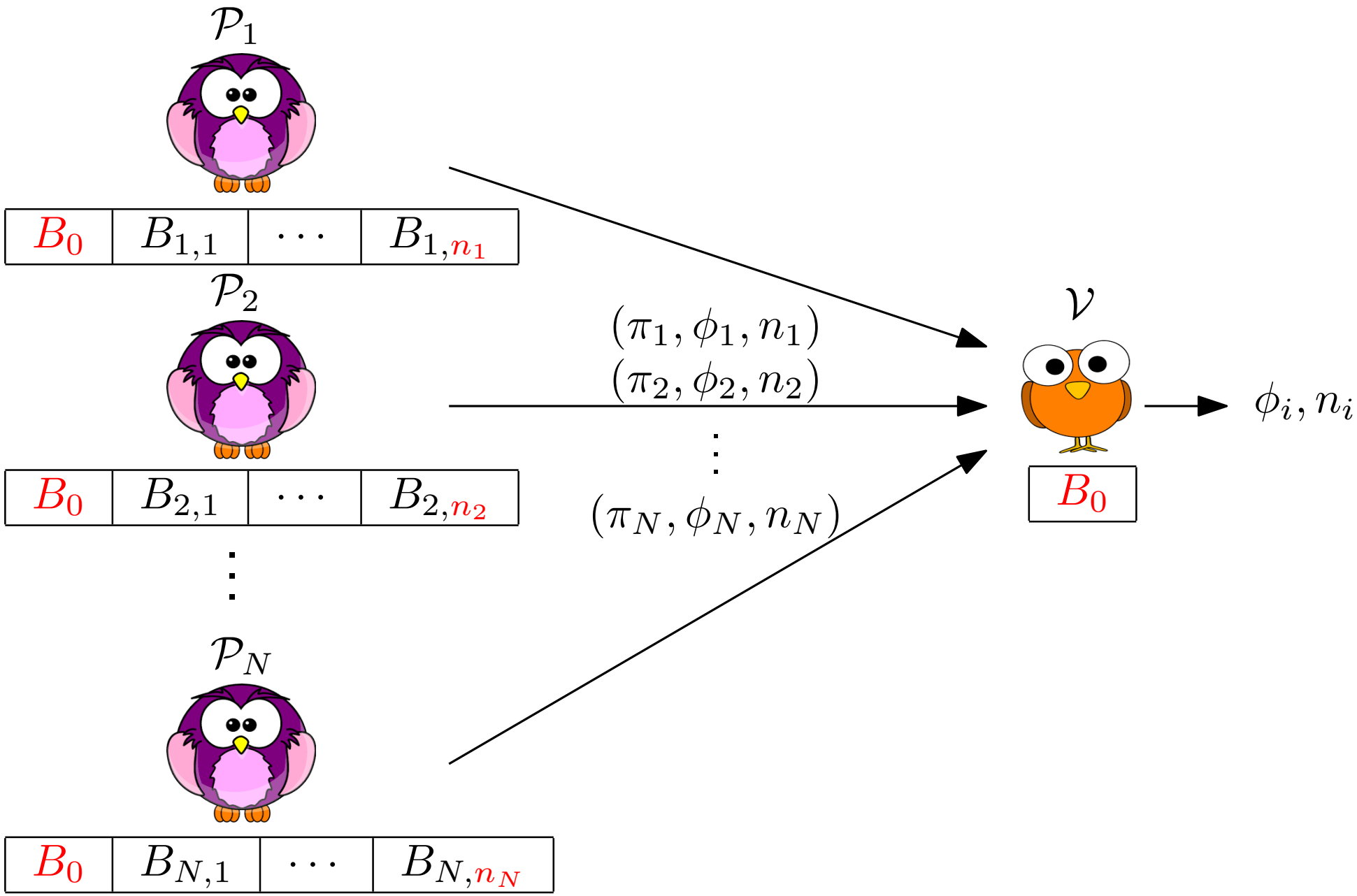
- sublinear effort ($\mathcal{V}$)
- sublinear communication
- non-interactive

$+$ sublinear effort ($\mathcal{P}$)

$+$ no setup assumptions

Bootstrapping

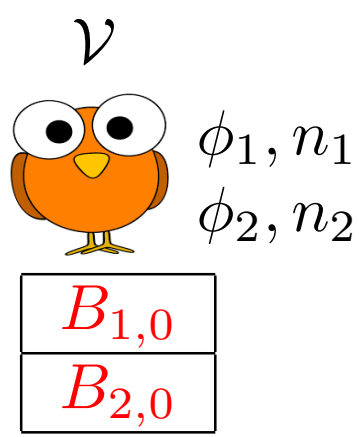
Ultimate goal of $\mathcal{V}$: a commitment to the stable prefix of the blockchain



Bootstrapped Verifiers

Bitcoin

$B_{1,0}$	$B_{1,1}$	$\cdots$	B_{1,n_1}



Chia

$B_{2,0}$	$B_{2,1}$	$\cdots$	B_{2,n_2}

Bootstrapped Verifiers

Bitcoin | $B_{1,0}$ | $B_{1,1}$ | $\cdots$ | B_{1,n_1} |

$\mathcal{P}$

Statement over transactions in
$B_{1,i}, B_{2,j}, \text{open}_i, \text{open}_j$

$\mathcal{V}$

ϕ_1, n_1
ϕ_2, n_2

$B_{1,0}$
$B_{2,0}$

Chia | $B_{2,0}$ | $B_{2,1}$ | $\cdots$ | B_{2,n_2} |

Solutions to Bootstrapping

- SNARKs

1), 3)

- Non-interactive proofs of proofs of work [Kiayias-Miller-Zindros 2020]

2), 3), 4)

- Flyclient [Bünz-Kiffer-Luu-Zamani 2020]

2), 4)

Some limitations

1) Setup assumption

2) Limited applicability: PoW chains

3) Limited usefulness: no commitments

4) Non-standard models: **multiple** provers with ≥ 1 is **honest**

This Work: SNACKs

Define a **classical proof system** underlying all these applications:
Succinct Non-interactive Arguments of Chain Knowledge

Construct SNACKs from any Graph-Labeling Proof of Sequential Work

Unify and generalize GL-PoSW schemes, and provide a new construction

Application: realize bootstrapping from SNACKs

This Work: SNACKs

Define a **classical proof system** underlying all these applications:
Succinct Non-interactive Arguments of Chain Knowledge

Construct SNACKs from any Graph-Labeling Proof of Sequential Work

Unify and generalize GL-PoSW schemes, and provide a new construction

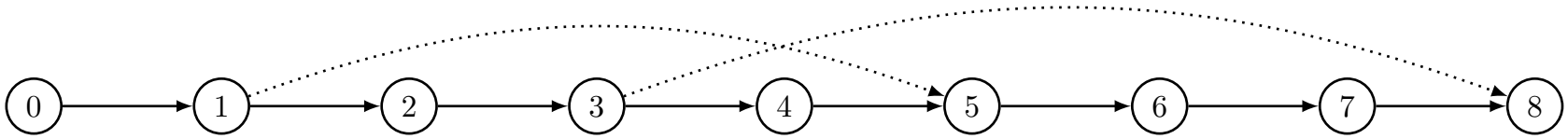
Application: realize bootstrapping from SNACKs

Advantages:

1. Classical soundness guarantees
2. Simplicity and modularity
3. Exchange of ideas between PoSW and Blockchains: *incremental* PoSW imply *incremental* SNACKs

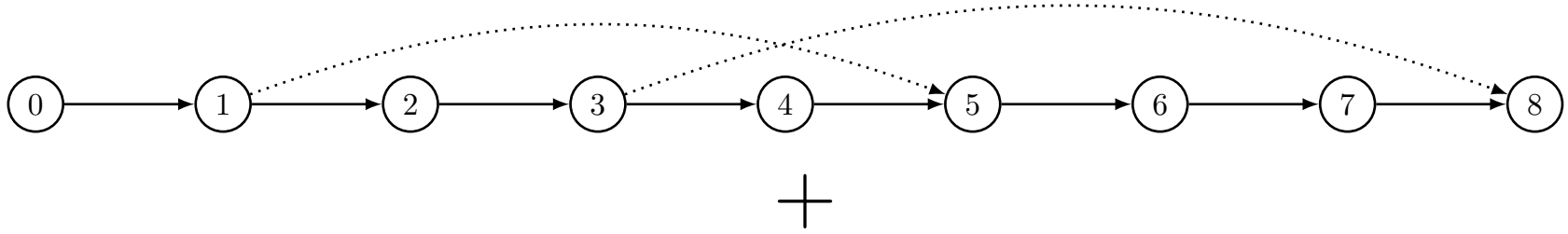
Constructing SNACKs

Blockchain DAG H with validity relation R_ψ s.t. $R_\psi(i, L(i), L(\text{paranets}(i))) = 1$ iff ith block is valid

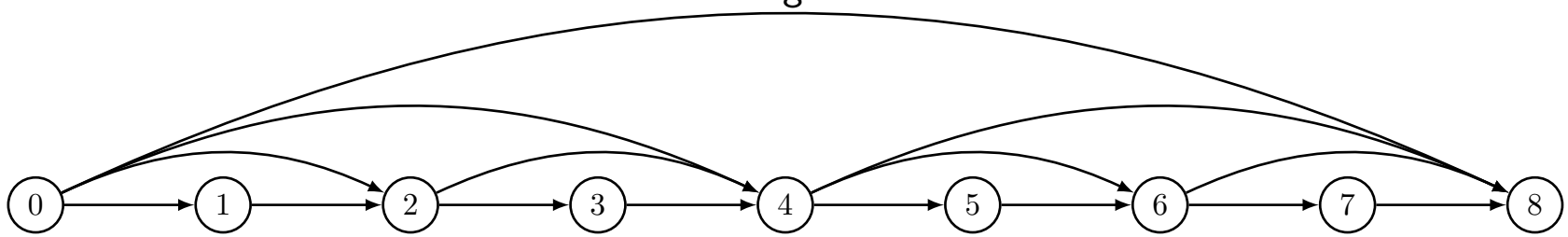


Constructing SNACKs

Blockchain DAG H with validity relation R_ψ s.t. $R_\psi(i, L(i), L(\text{paranets}(i))) = 1$ iff ith block is valid



PoSW weighted DAG G

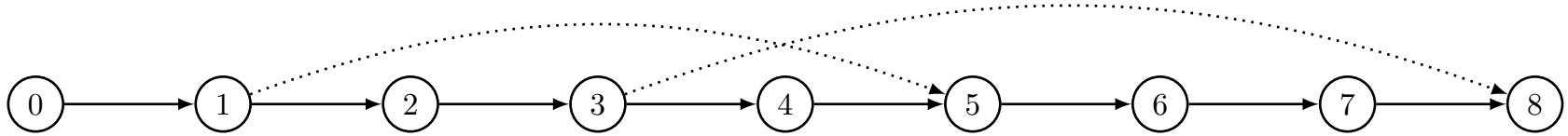


$\mathcal{P}$

$\mathcal{V}$

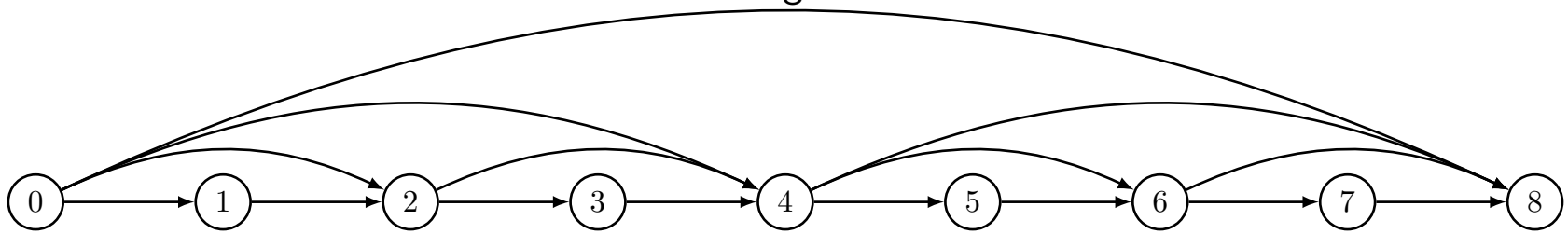
Constructing SNACKs

Blockchain DAG H with validity relation R_ψ s.t. $R_\psi(i, L(i), L(\text{paranets}(i))) = 1$ iff ith block is valid



+

PoSW weighted DAG G

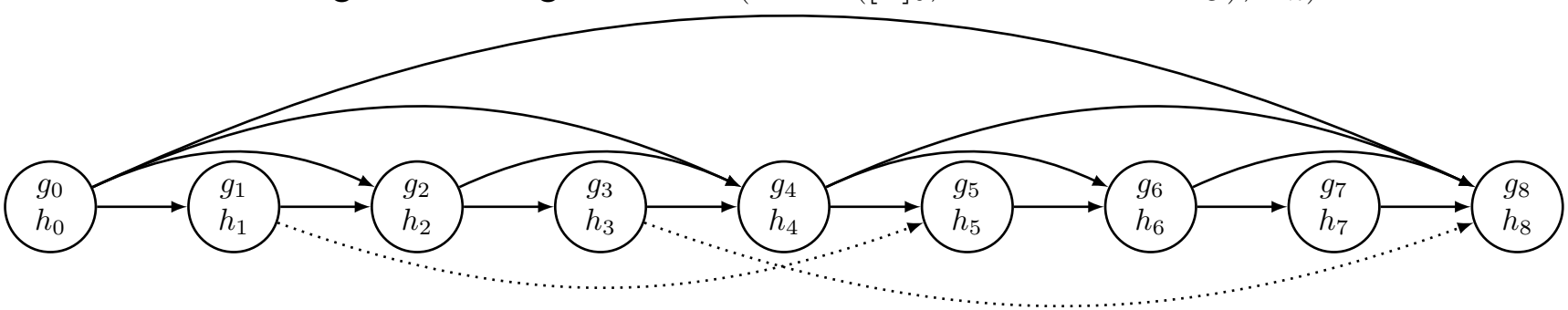


$\mathcal{P}$

$\mathcal{V}$

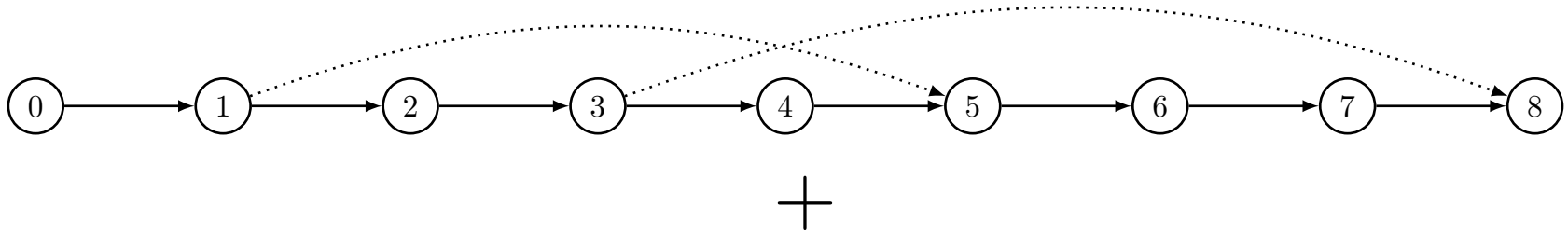
=

Augmented weighted DAG $(K := ([n]_0, E_K := E_H \cup E_G), \Omega_n)$

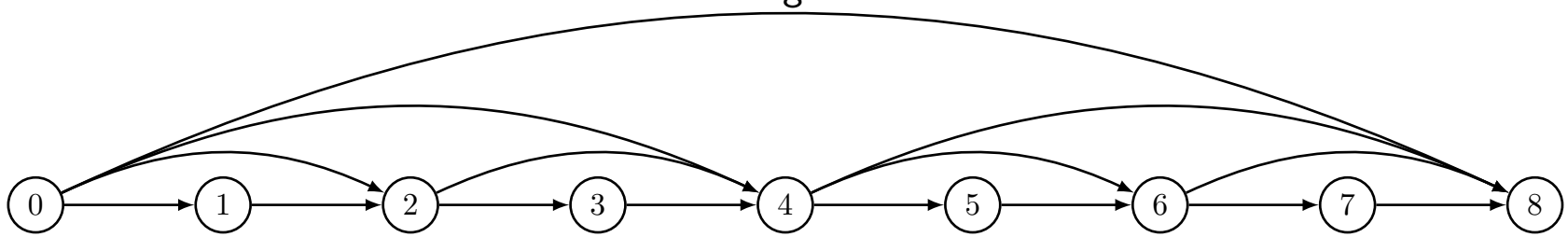


Constructing SNACKs

Blockchain DAG H with validity relation R_ψ s.t. $R_\psi(i, L(i), L(\text{paranets}(i))) = 1$ iff ith block is valid



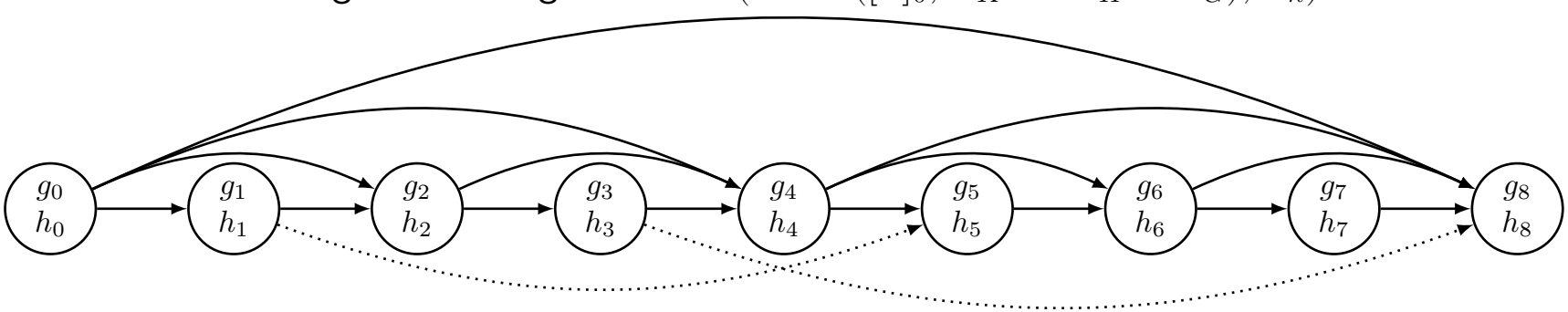
PoSW weighted DAG G



$\mathcal{P}$

$\mathcal{V}$

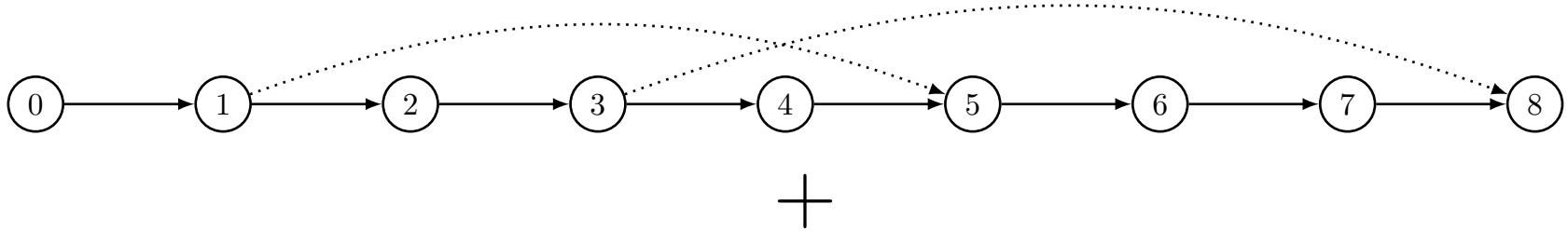
Augmented weighted DAG $(K := ([n]_0, E_K := E_H \cup E_G), \Omega_n)$



$$\mathcal{R}^{(\alpha)} = \left\{ \left((\phi_n, n), (P, L_P, (w_i)_{i\in P}, \rho)\right) \ : \ \begin{array}{c} (P, L_P, (w_i)_{i\in P}) \text{ is } (\alpha, R)\text{-valid} \\ \wedge\ \text{Comm.ver}(\phi_n, L_P, P, \rho) = 1 \end{array} \right\}$$

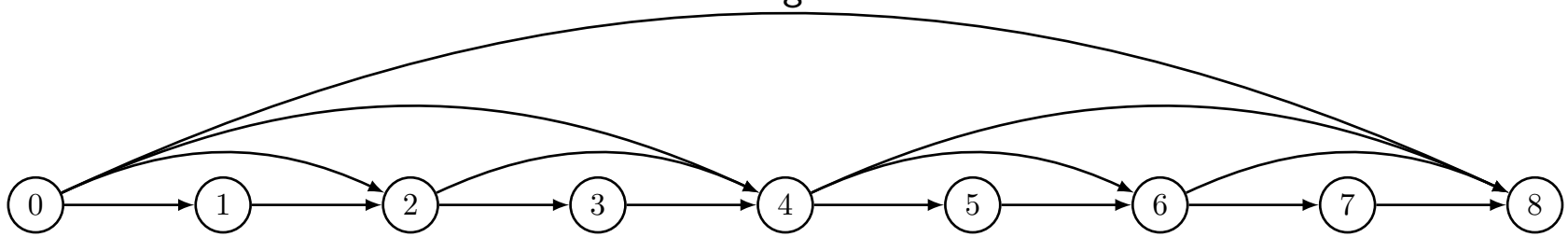
Constructing SNACKs

Blockchain DAG H with validity relation R_ψ s.t. $R_\psi(i, L(i), L(\text{paranets}(i))) = 1$ iff ith block is valid



+

PoSW weighted DAG G



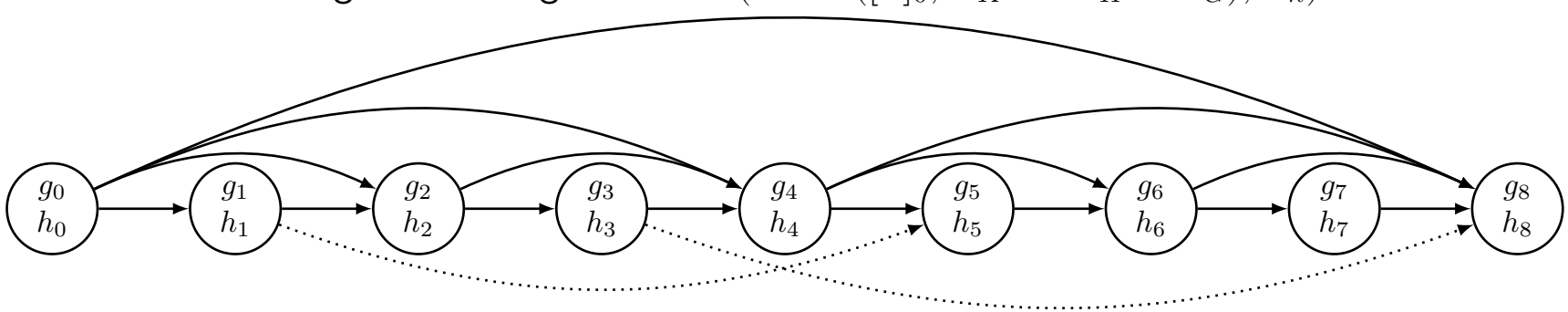
$\mathcal{P}$

$\mathcal{V}$

=

Augmented weighted DAG $(K := ([n]_0, E_K := E_H \cup E_G), \Omega_n)$

≀



≀

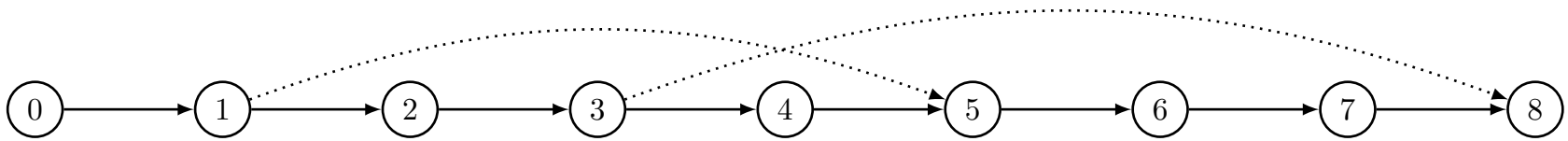
$\mathcal{P}'$

$\mathcal{V}'$

$$\mathcal{R}^{(\alpha)} = \left\{ \left((\phi_n, n), (P, L_P, (w_i)_{i\in P}, \rho)\right) \ : \ \begin{array}{l}(P, L_P, (w_i)_{i\in P}) \text{ is } (\alpha, R)\text{-valid} \\ \wedge\ \text{Comm.ver}(\phi_n, L_P, P, \rho) = 1\end{array} \right\}$$

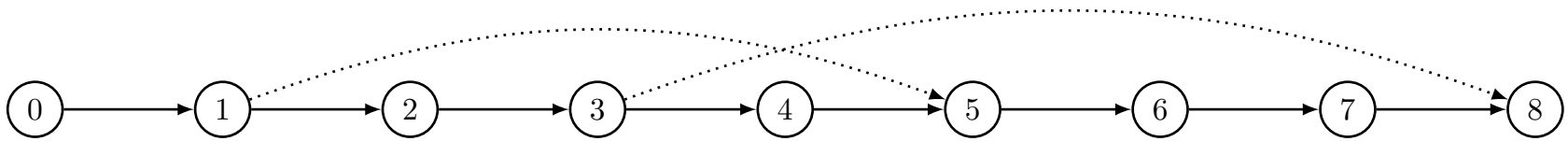
DAG Labeling

Chain: $C_n = ([n]_0, E_n)$ a DAG with $E_n \supseteq \{(i-1, i) : i \in [n]\}$



DAG Labeling

Chain: $C_n = ([n]_0, E_n)$ a DAG with $E_n \supseteq \{(i-1, i) : i \in [n]\}$



τ-based graph labeling: $\tau : \{0,1\}^* \to \{0,1\}^\lambda$ (modeled as a RO)

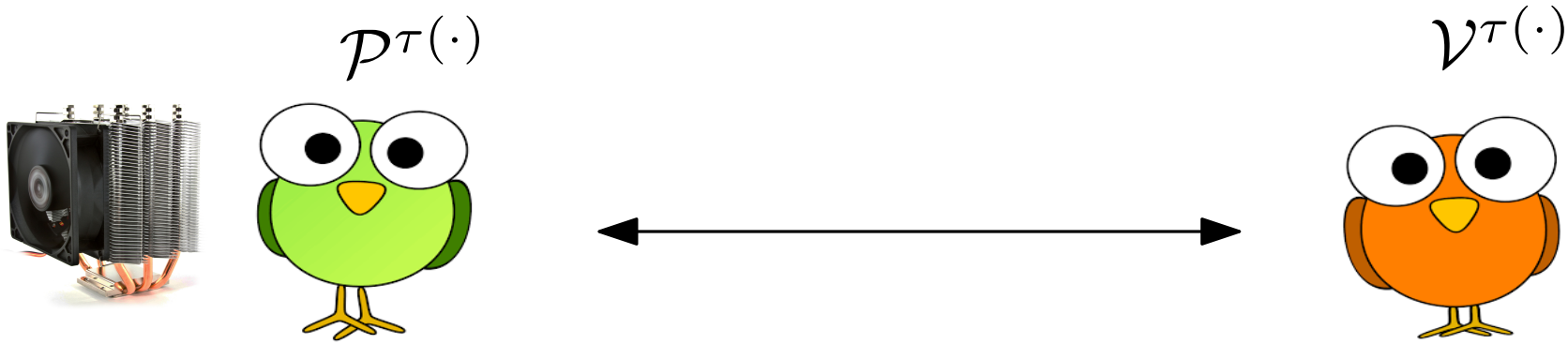
$$L(i) := \begin{cases} \tau(i) & \text{if parents}(i) = \emptyset, \\ \tau\big(i, L(\text{parents}(i))\big) & \text{otherwise.} \end{cases}$$

$E.g., L(5) = \tau(5, L(1), L(4))$

Proof of Sequential Work

[Mahmoody-Moran-Vadhan 2013]

Parameters: DAG $G_n, \ldots$



Completeness: Honest $\mathcal{P}^{\tau(\cdot)}$ making n **sequential** $\tau(\cdot)$ queries makes $\mathcal{V}$ accept w.p. 1

Proof of Sequential Work

[Mahmoody-Moran-Vadhan 2013]

Parameters: DAG $G_n, \ldots$

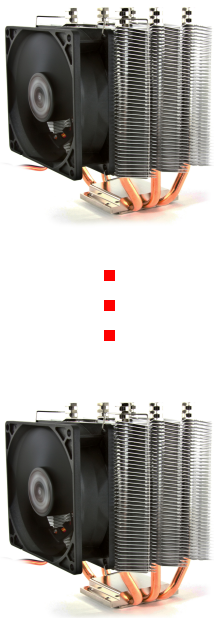
Completeness: Honest $\mathcal{P}^{\tau(\cdot)}$ making n **sequential** $\tau(\cdot)$ queries makes $\mathcal{V}$ accept w.p. 1

Succinctness: For every honest proof π :
$|\pi| \leq \text{poly}(\lambda, \log n)$,Time$(\mathcal{V}) \leq \text{poly}(\lambda, \log n)$, and Time$(\mathcal{P}) \leq \text{poly}(\lambda, n)$

Proof of Sequential Work

[Mahmoody-Moran-Vadhan 2013]

Parameters: DAG $G_n, \ldots$



$\tilde{\mathcal{P}}^{\tau(\cdot)}$

$\mathcal{V}^{\tau(\cdot)}$

poly$(\lambda, \log n)$ time

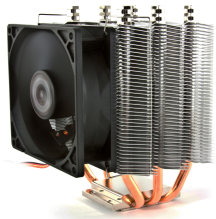
(α, ϵ)-**Soundness**: A parallel $\tilde{\mathcal{P}}^{\tau(\cdot)}$ making $\leq \alpha \cdot n$ **sequential** queries to $\tau(\cdot)$ fails with prob. $\geq 1 - \epsilon(\lambda)$

A Simple PoSW

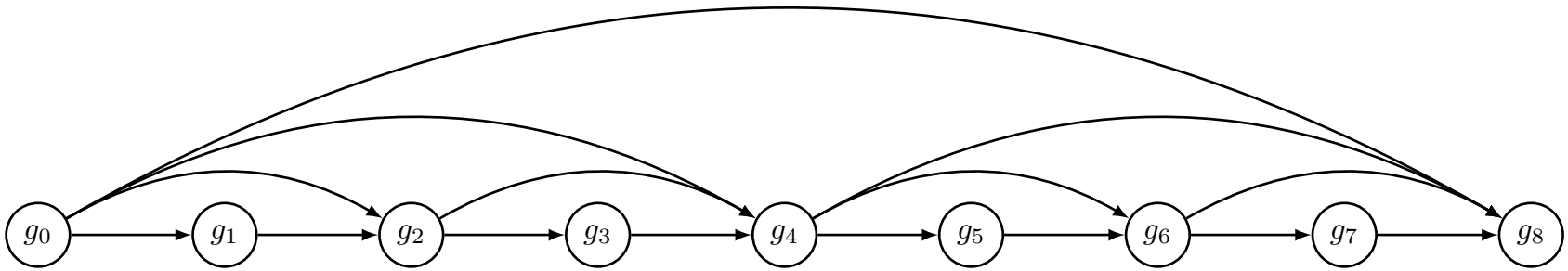
This work

Parameters: $G_n, \ldots$

$\mathcal{P}^{\mathcal{O}(\cdot)}$



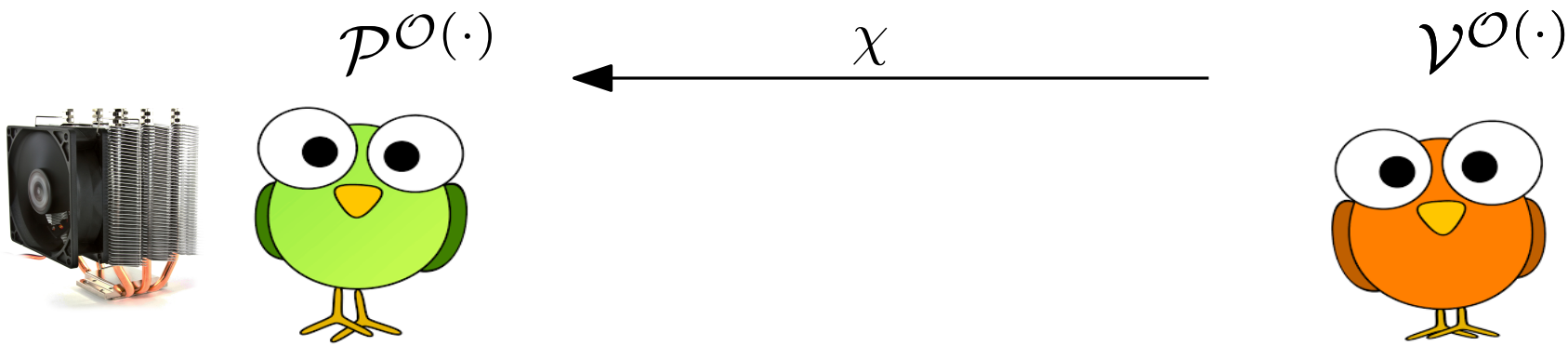
$\mathcal{V}^{\mathcal{O}(\cdot)}$



A Simple PoSW

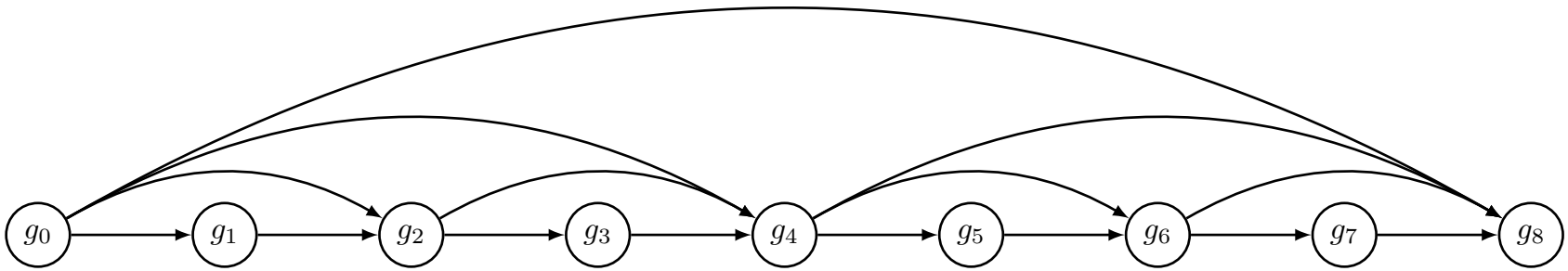
This work

Parameters: $G_n, \ldots$



τ-based graph labeling: $\tau : \{0,1\}^* \to \{0,1\}^\lambda$ with $\tau := \mathcal{O}(\chi, \cdot)$

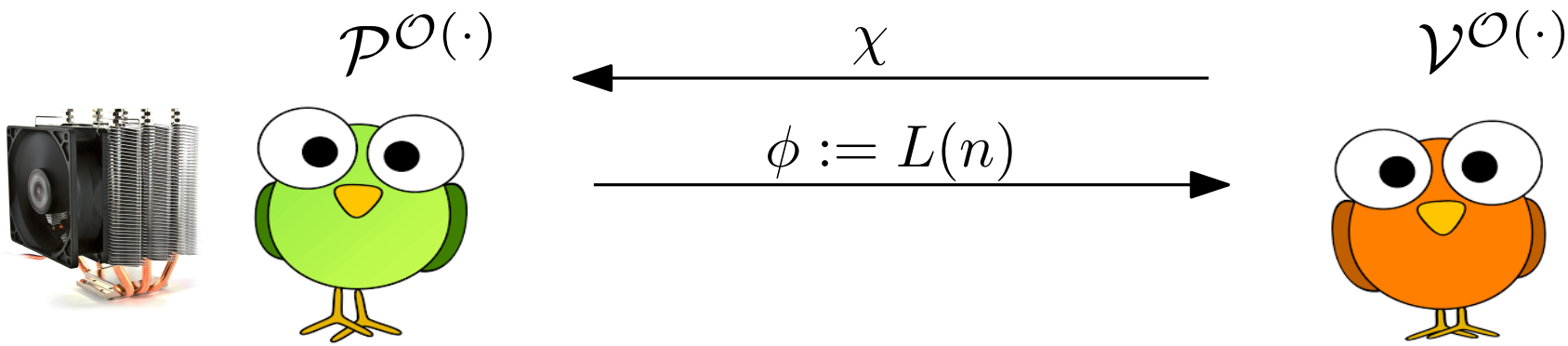
$$L(i) := \begin{cases} \tau(i) & \text{if parents}(i) = \emptyset, \\ \tau\big(i, L(\text{parents}(i))\big) & \text{otherwise.} \end{cases}$$



A Simple PoSW

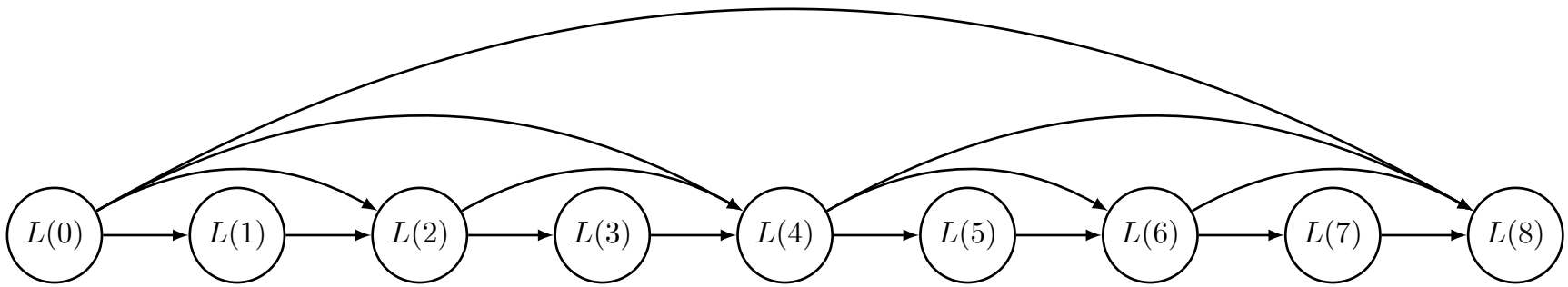
This work

Parameters: $G_n, \ldots$



τ-based graph labeling: $\tau : \{0,1\}^* \to \{0,1\}^\lambda$ with $\tau := \mathcal{O}(\chi, \cdot)$

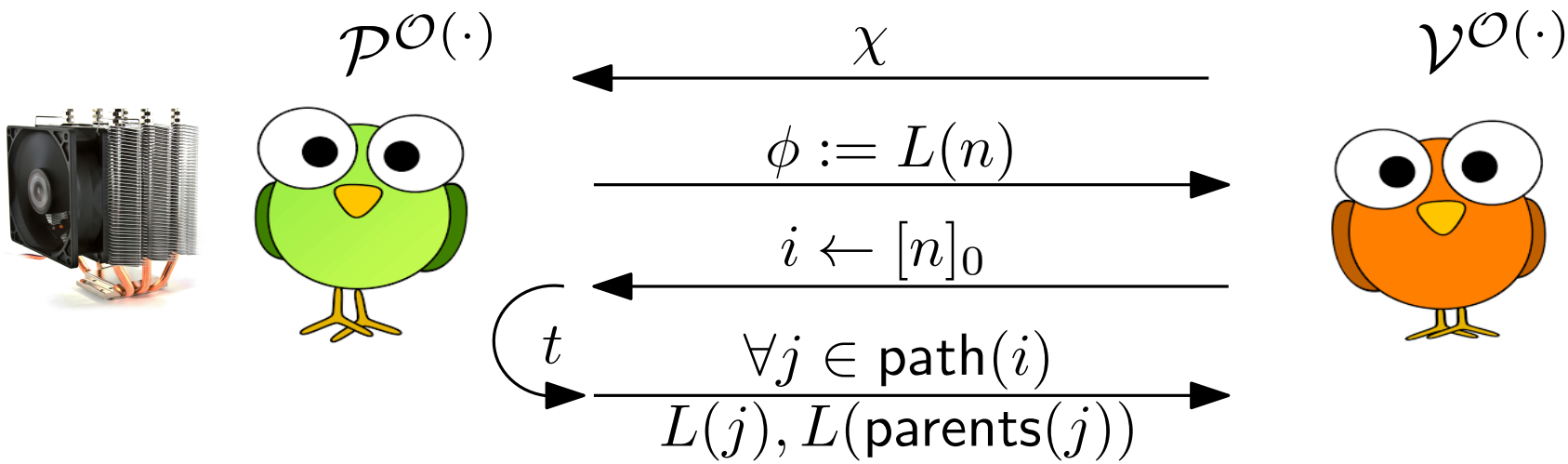
$$L(i) := \begin{cases} \tau(i) & \text{if parents}(i) = \emptyset, \\ \tau\big(i, L(\text{parents}(i))\big) & \text{otherwise.} \end{cases}$$



A Simple PoSW

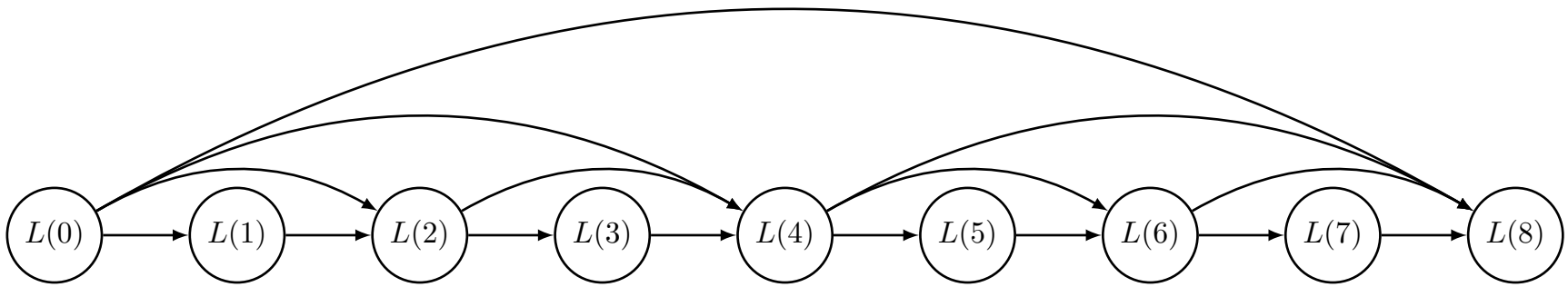
This work

Parameters: $G_n, \ldots$



τ-based graph labeling: $\tau : \{0,1\}^* \to \{0,1\}^\lambda$ with $\tau := \mathcal{O}(\chi, \cdot)$

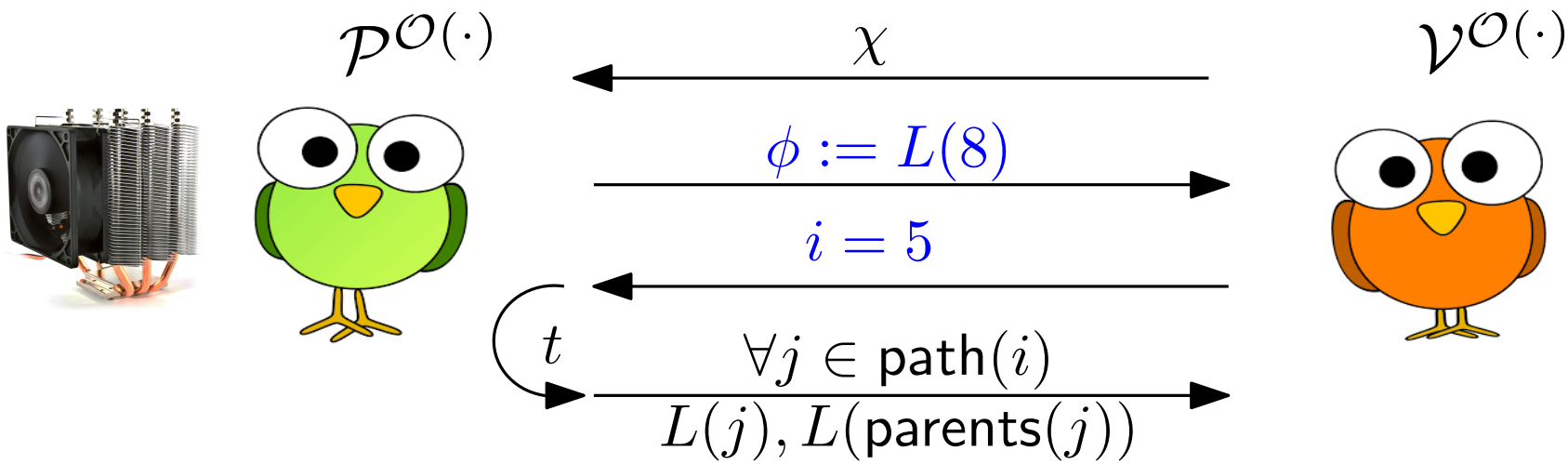
$$L(i) := \begin{cases} \tau(i) & \text{if } \mathsf{parents}(i) = \emptyset, \\ \tau\big(i, L(\mathsf{parents}(i))\big) & \text{otherwise.} \end{cases}$$



A Simple PoSW

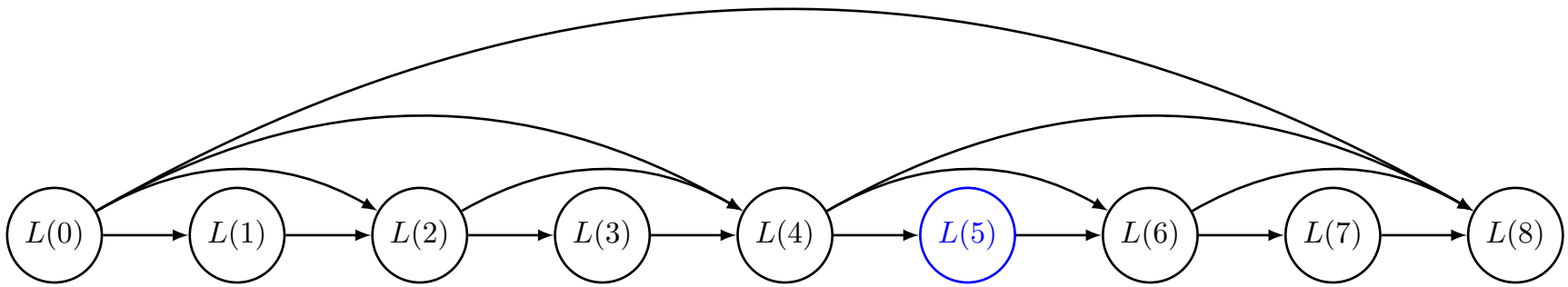
This work

Parameters: $G_n, \ldots$



τ-based graph labeling: $\tau : \{0,1\}^* \to \{0,1\}^\lambda$ with $\tau := \mathcal{O}(\chi, \cdot)$

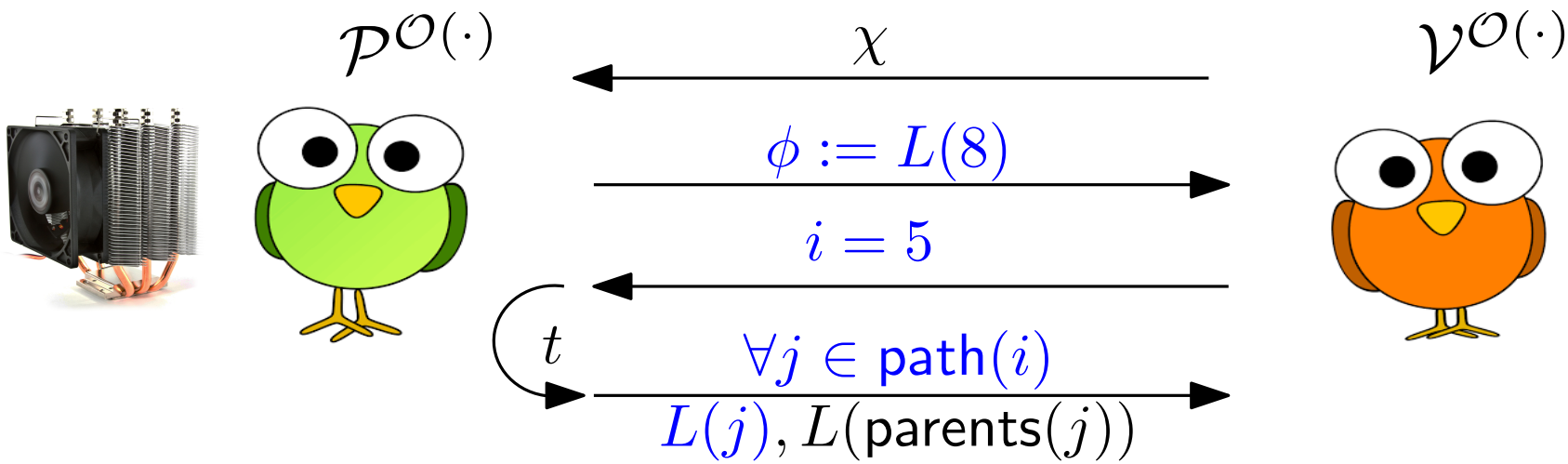
$$L(i) := \begin{cases} \tau(i) & \text{if } \mathsf{parents}(i) = \emptyset, \\ \tau\big(i, L(\mathsf{parents}(i))\big) & \text{otherwise.} \end{cases}$$



A Simple PoSW

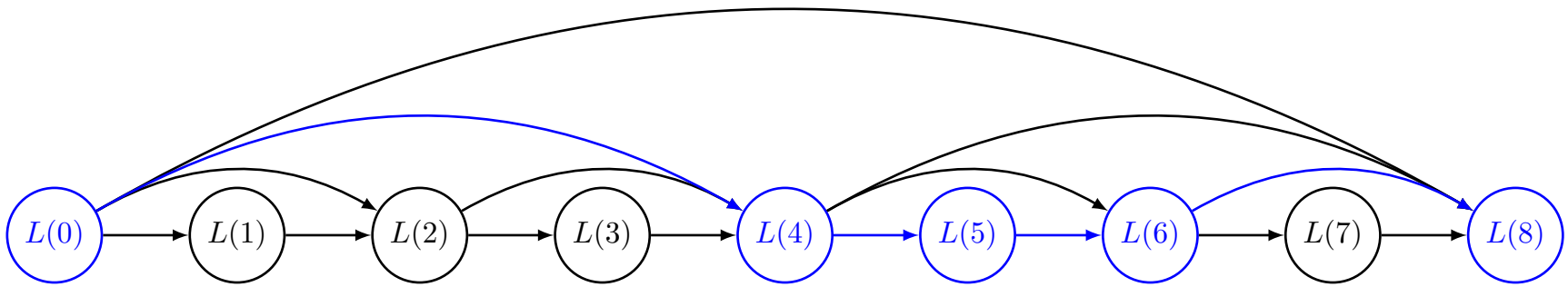
This work

Parameters: $G_n, \ldots$



τ-based graph labeling: $\tau : \{0,1\}^* \to \{0,1\}^\lambda$ with $\tau := \mathcal{O}(\chi, \cdot)$

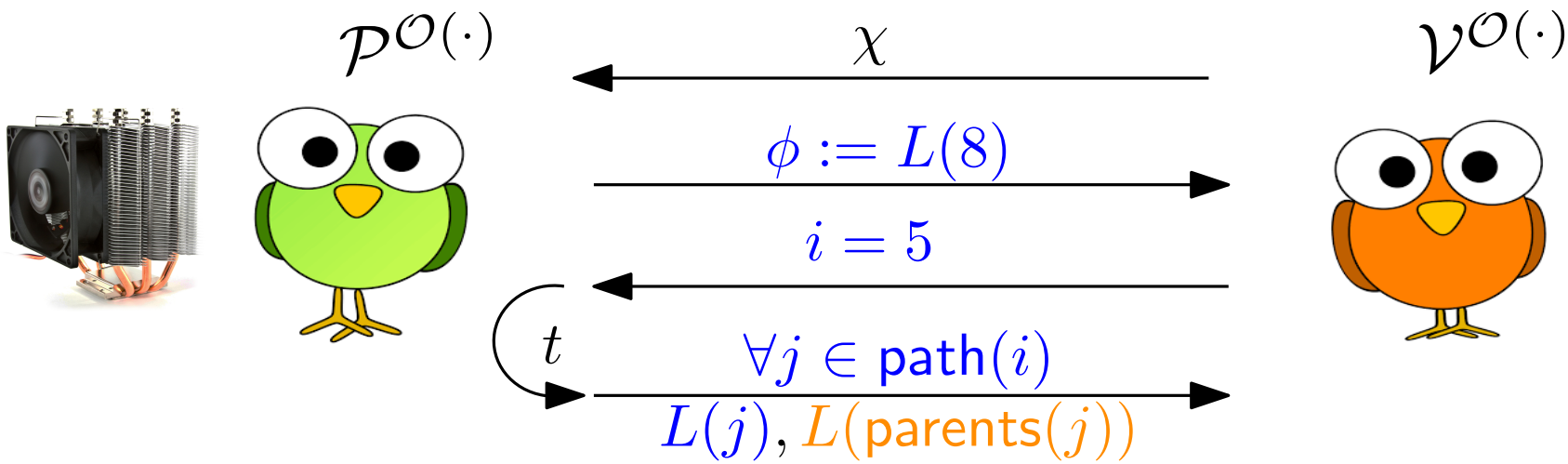
$$L(i) := \begin{cases} \tau(i) & \text{if } \mathsf{parents}(i) = \emptyset, \\ \tau\big(i, L(\mathsf{parents}(i))\big) & \text{otherwise.} \end{cases}$$



A Simple PoSW

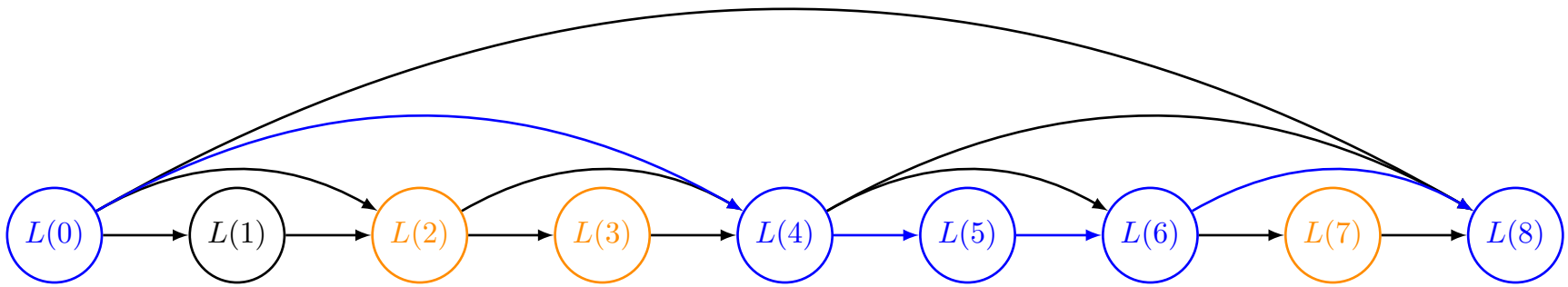
This work

Parameters: $G_n, \ldots$



τ-based graph labeling: $\tau : \{0,1\}^* \to \{0,1\}^\lambda$ with $\tau := \mathcal{O}(\chi, \cdot)$

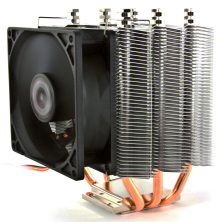
$$L(i) := \begin{cases} \tau(i) & \text{if } \mathsf{parents}(i) = \emptyset, \\ \tau\big(i, L(\mathsf{parents}(i))\big) & \text{otherwise.} \end{cases}$$



A Simple PoSW

This work

Parameters: $G_n, \ldots$



$\mathcal{P}^{\mathcal{O}(\cdot)}$

χ

$\phi := L(8)$

$i = 5$

t

$\forall j \in \mathsf{path}(i)$

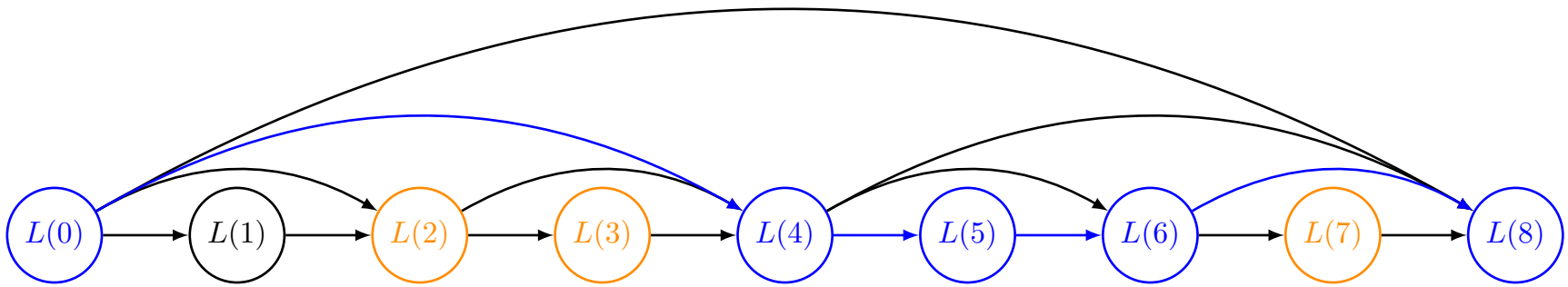
$L(j), L(\mathsf{parents}(j))$

$\mathcal{V}^{\mathcal{O}(\cdot)}$

$\mathrm{poly}(\lambda, \log n)$ time

τ-based graph labeling: $\tau : \{0,1\}^* \to \{0,1\}^\lambda$ with $\tau := \mathcal{O}(\chi, \cdot)$

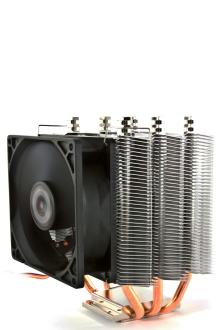
$$L(i) := \begin{cases} \tau(i) & \text{if } \mathsf{parents}(i) = \emptyset, \\ \tau\big(i, L(\mathsf{parents}(i))\big) & \text{otherwise.} \end{cases}$$



A Simple PoSW

This work

Parameters: $G_n, \ldots$



$\mathcal{P}^{\mathcal{O}(\cdot)}$

χ

$\phi := L(8)$

$i = 5$

t

$\forall j \in \text{path}(i)$

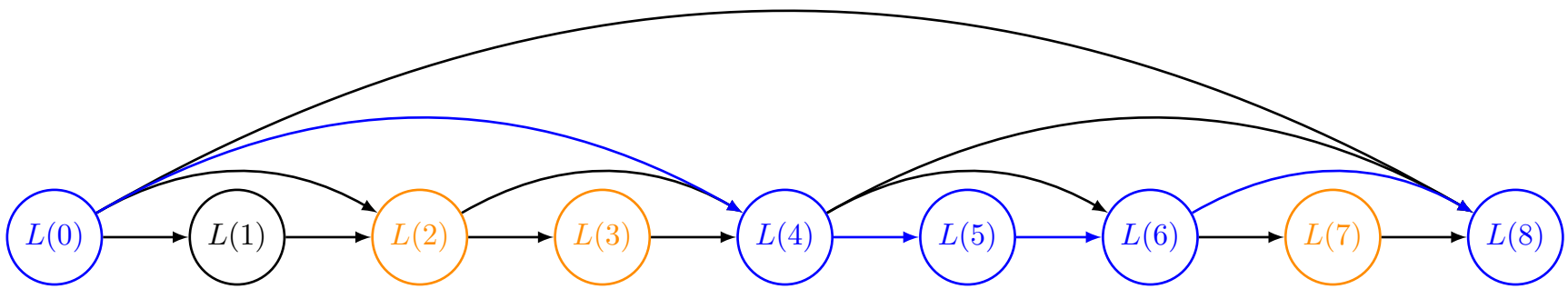
$L(j), L(\text{parents}(j))$

$\mathcal{V}^{\mathcal{O}(\cdot)}$

$\text{poly}(\lambda, \log n)$ time

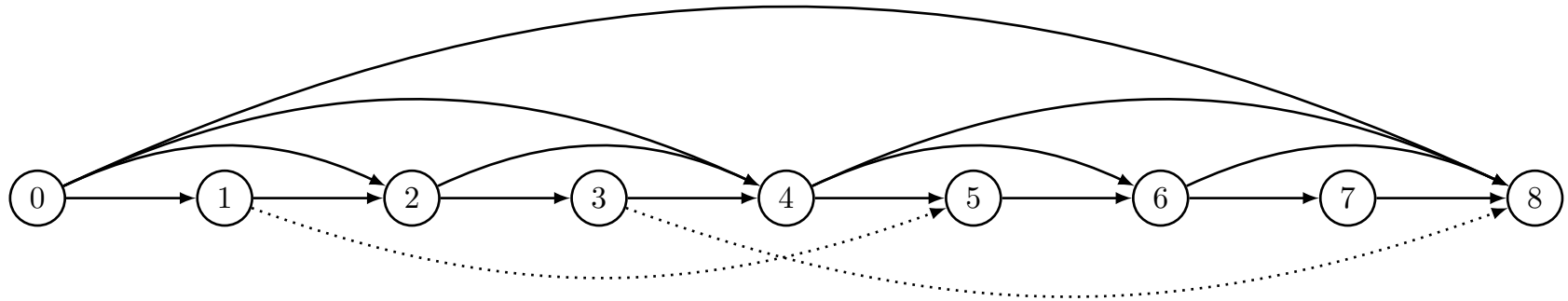
Thm: If 1. $\tilde{\mathcal{P}}_1$ made $\leq \alpha \cdot n$ sequential queries to $\tau(\cdot)$ before sending ϕ
2. $\tilde{\mathcal{P}} := (\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2)$ made a total of $\leq q$ queries to $\tau(\cdot)$
Then $\tilde{\mathcal{P}}$ makes $\mathcal{V}$ accept with probability at most

$$\epsilon := \alpha^t + 3 \cdot q^2 / 2^\lambda$$



Towards Defining SNACKs

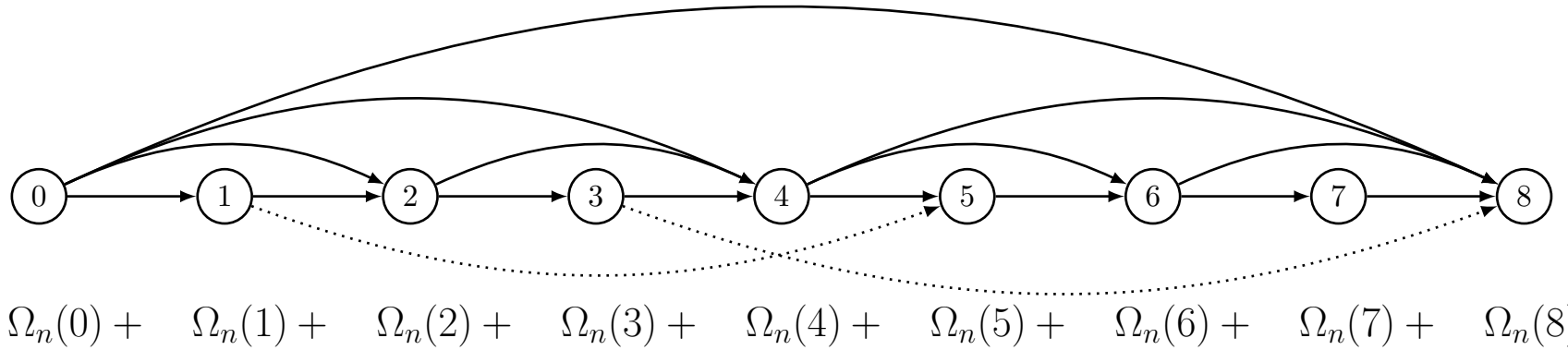
Weighted: $\Gamma_n = (C_n, \Omega_n)$ with $\Omega_n : [n]_0 \to [0,1]$ s.t. $\sum_{i=0}^{n} \Omega_n(i) = 1$



$\Omega_n(0) + \quad \Omega_n(1) + \quad \Omega_n(2) + \quad \Omega_n(3) + \quad \Omega_n(4) + \quad \Omega_n(5) + \quad \Omega_n(6) + \quad \Omega_n(7) + \quad \Omega_n(8) = 1$

Towards Defining SNACKs

Weighted: $\Gamma_n = (C_n, \Omega_n)$ with $\Omega_n : [n]_0 \to [0,1]$ s.t. $\sum_{i=0}^{n} \Omega_n(i) = 1$

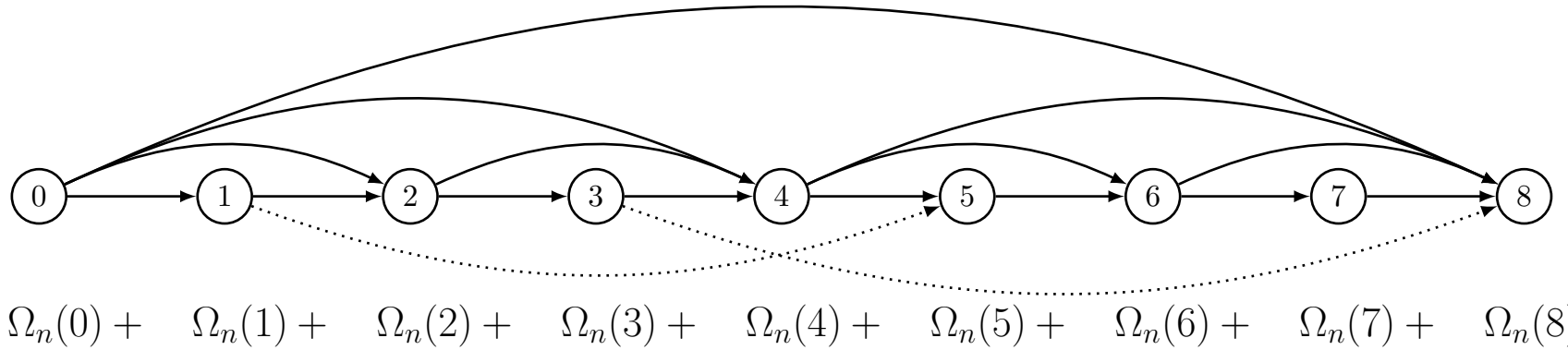


$(x_0, \ldots, x_n)$**-augmented** τ**-based graph labeling:** $\tau : \{0,1\}^* \to \{0,1\}^\lambda$

$$L(i) := \begin{cases} \tau(i) \| x_i & \text{if parents}(i) = \emptyset, \\ \tau\big(i, L(\text{parents}(i))\big) \| x_i & \text{otherwise.} \end{cases}$$

Towards Defining SNACKs

Weighted: $\Gamma_n = (C_n, \Omega_n)$ with $\Omega_n : [n]_0 \to [0,1]$ s.t. $\sum_{i=0}^{n} \Omega_n(i) = 1$



$(x_0, \ldots, x_n)$**-augmented** τ**-based graph labeling:** $\tau : \{0,1\}^* \to \{0,1\}^\lambda$

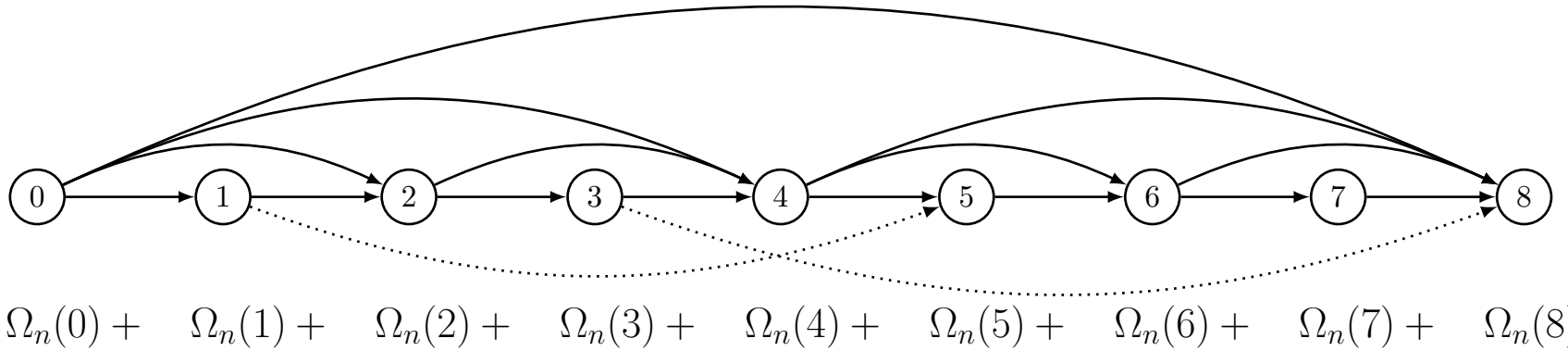
$$L(i) := \begin{cases} \tau(i) \| x_i & \text{if parents}(i) = \emptyset, \\ \tau\big(i, L(\text{parents}(i))\big) \| x_i & \text{otherwise.} \end{cases}$$

Validty of augmented labels via $R \subseteq \mathbb{N}_0 \times (\{0,1\})^2$:

$$R\big(i, L(i), w_i\big) = 1 \text{ iff } \; L(i) = \tau(i, w_i) \| x'_i \quad \wedge \quad R_\psi(i, L(i), w_i) = 1$$

Towards Defining SNACKs

Weighted: $\Gamma_n = (C_n, \Omega_n)$ with $\Omega_n : [n]_0 \to [0,1]$ s.t. $\sum_{i=0}^{n} \Omega_n(i) = 1$



$(x_0, \ldots, x_n)$-**augmented** τ-**based graph labeling:** $\tau : \{0,1\}^* \to \{0,1\}^\lambda$

$$L(i) := \begin{cases} \tau(i) \| x_i & \text{if parents}(i) = \emptyset, \\ \tau\big(i, L(\text{parents}(i))\big) \| x_i & \text{otherwise.} \end{cases}$$

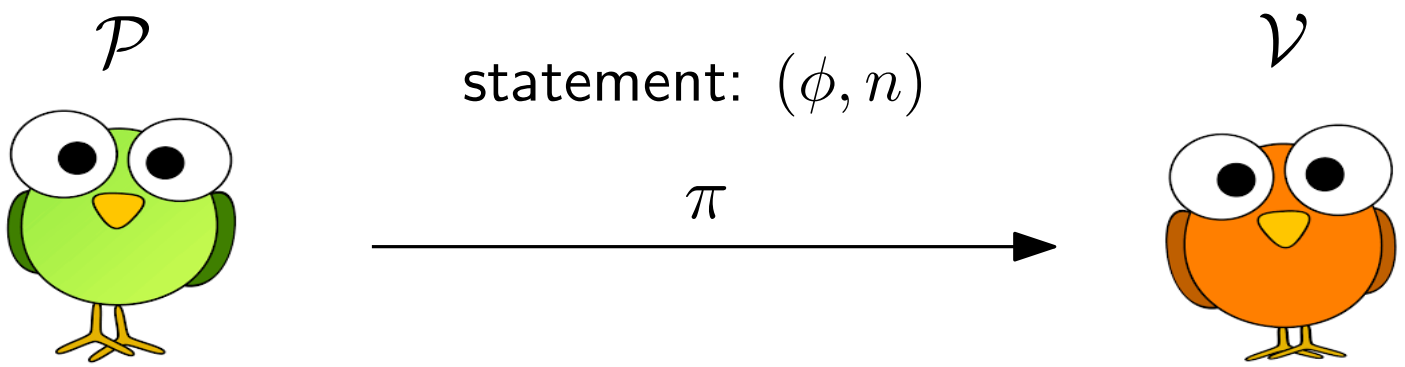
Validty of augmented labels via $R \subseteq \mathbb{N}_0 \times (\{0,1\})^2$:

$$R\big(i, L(i), w_i\big) = 1 \text{ iff } \; L(i) = \tau(i, w_i) \| x'_i \quad \wedge \quad R_\psi(i, L(i), w_i) = 1$$

Valid path: $(P, L_P, (w_i)_{i \in P})$ in Γ_n for R is (α, R)-**valid** path if

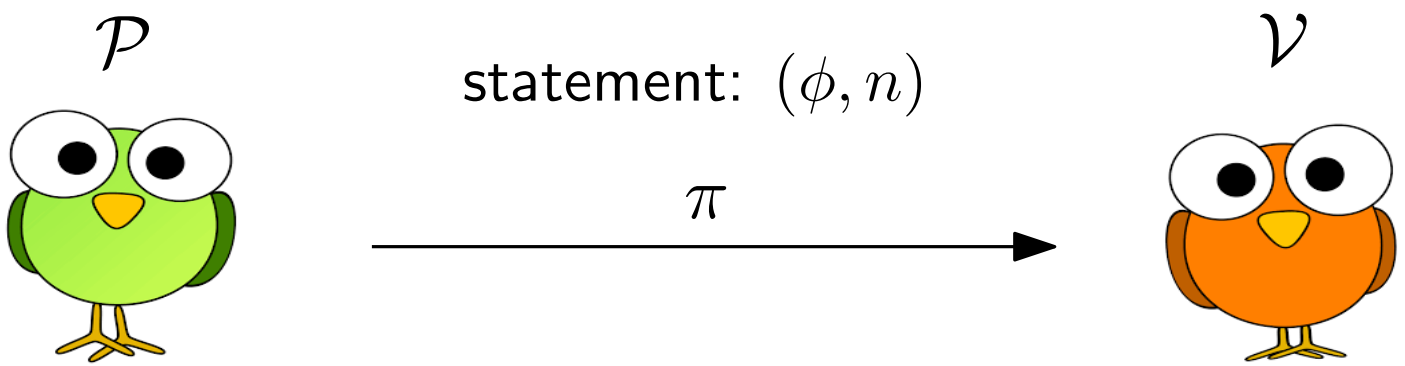
$$\forall i \in P : R\big(i, L_P(i), w_i\big) = 1 \quad \text{and} \quad \Omega_n(P) \geq \alpha$$

Defining SNACKs



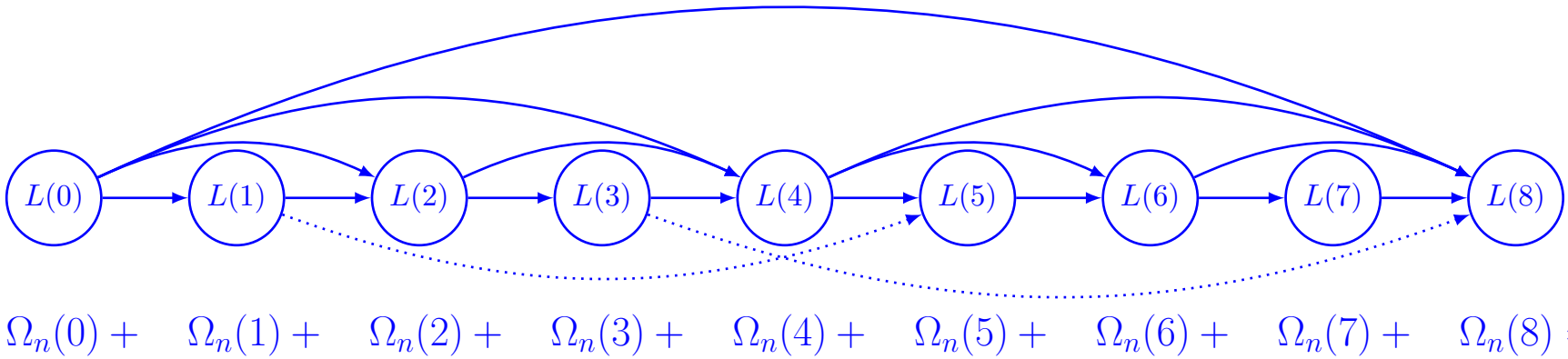
$$\mathcal{R}^{(\alpha)} = \left\{ \begin{array}{l} \big((\phi, n), \\ (P, L_P, (w_i)_{i \in P}, \rho)\big) \end{array} : \begin{array}{c} (P, L_P, (w_i)_{i \in P}) \text{ is } (\alpha, R)\text{-valid} \\ \wedge\ \mathsf{Comm.ver}(\phi, L_P, P, \rho) = 1 \end{array} \right\}$$

Defining SNACKs

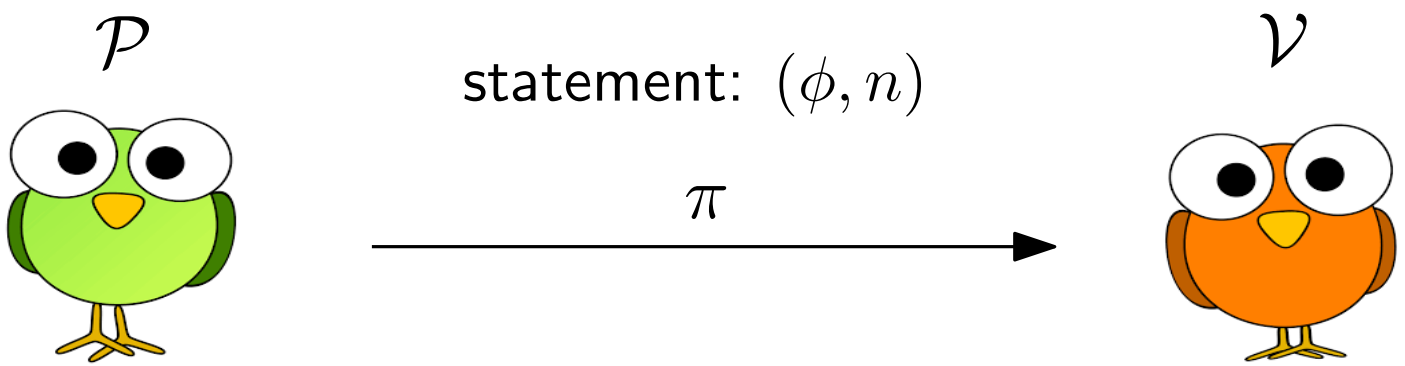


$$\mathcal{R}^{(\alpha)} = \left\{ \begin{array}{l} ((\phi, n), \\ (P, L_P, (w_i)_{i \in P}, \rho)) \end{array} : \begin{array}{c} (P, L_P, (w_i)_{i \in P}) \text{ is } (\alpha, R)\text{-valid} \\ \wedge \ \text{Comm.ver}(\phi, L_P, P, \rho) = 1 \end{array} \right\}$$

Completeness: For every $((\phi, n), s) \in \mathcal{R}^{(\alpha=1)}$, $\mathcal{V}$ accepts w.p. 1

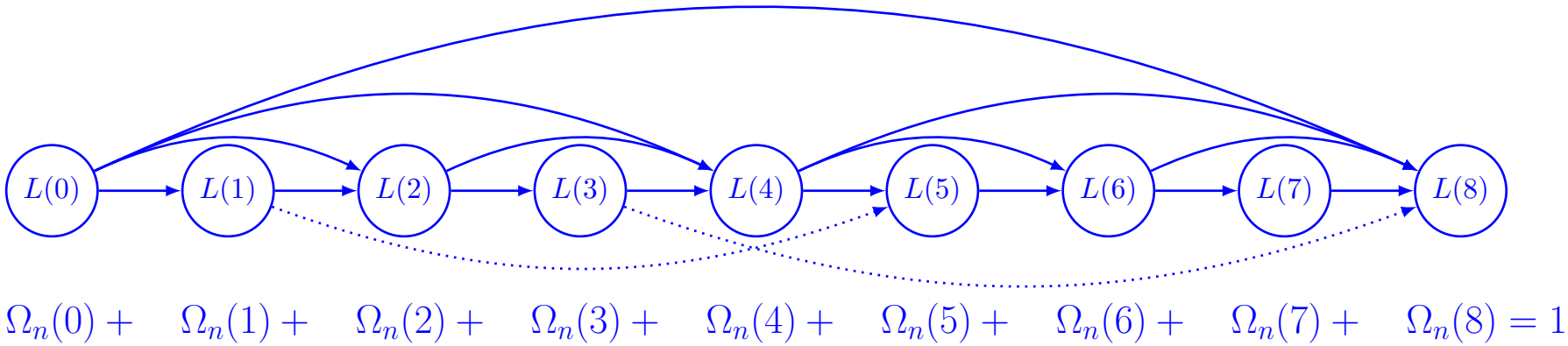


Defining SNACKs



$$\mathcal{R}^{(\alpha)} = \left\{ \begin{array}{l} ((\phi, n), \\ (P, L_P, (w_i)_{i\in P}, \rho)) \end{array} : \begin{array}{c} (P, L_P, (w_i)_{i\in P}) \text{ is } (\alpha, R)\text{-valid} \\ \wedge\ \mathsf{Comm.ver}(\phi, L_P, P, \rho) = 1 \end{array} \right\}$$

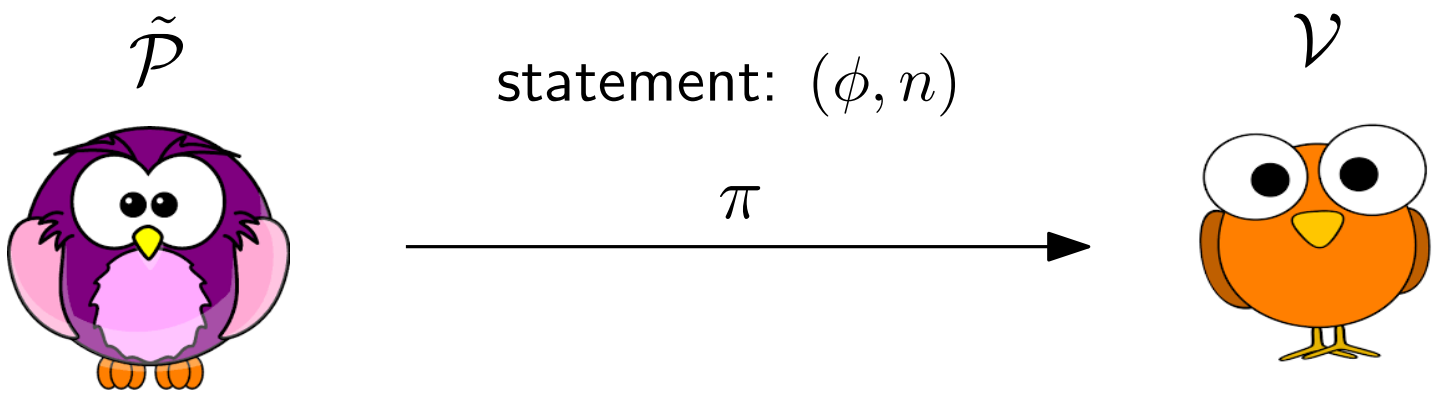
Completeness: For every $((\phi, n), s) \in \mathcal{R}^{(\alpha=1)}$, $\mathcal{V}$ accepts w.p. 1



Succinctness: For every honest π

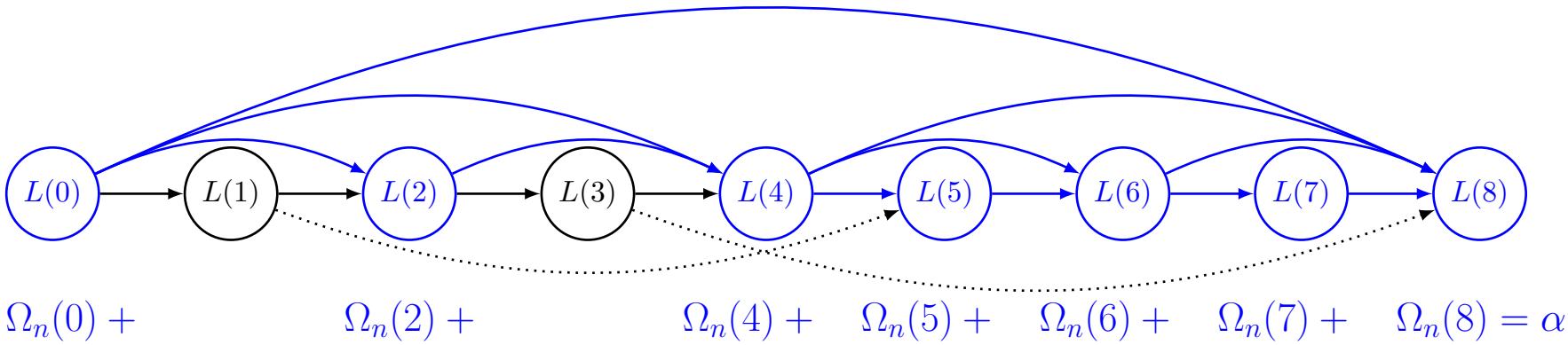
$|\pi| \leq \mathsf{poly}(\lambda, \log n)$, $\mathsf{Time}(\mathcal{V}) \leq \mathsf{poly}(\lambda, \log n)$, and $\mathsf{Time}(\mathcal{P}) \leq \mathsf{poly}(\lambda, n)$

Defining SNACKs



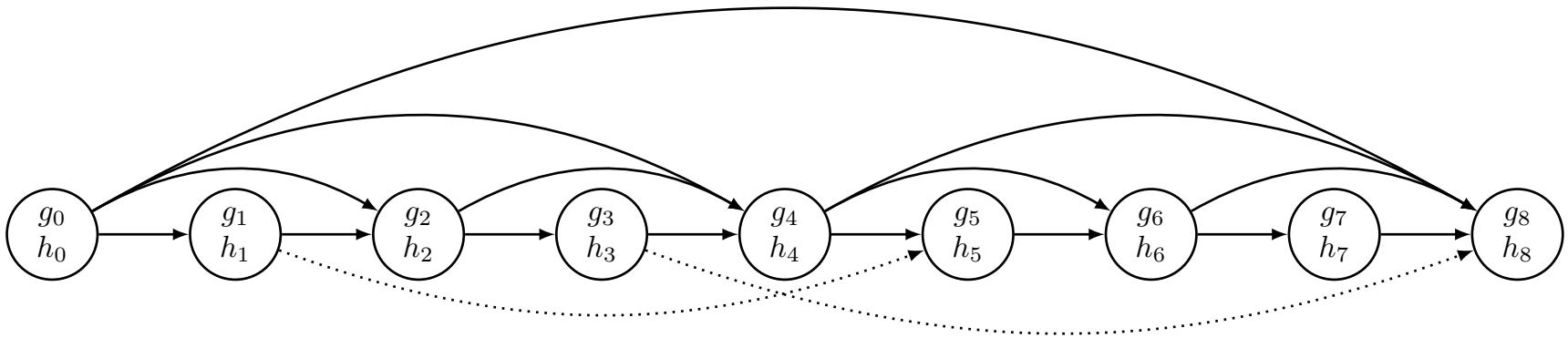
$$\mathcal{R}^{(\alpha)} = \left\{ \begin{array}{l} ((\phi, n), \\ (P, L_P, (w_i)_{i \in P}, \rho)) \end{array} : \begin{array}{c} (P, L_P, (w_i)_{i \in P}) \text{ is } (\alpha, R)\text{-valid} \\ \wedge \ \text{Comm.ver}(\phi, L_P, P, \rho) = 1 \end{array} \right\}$$

(α, ϵ)-**Knowledge Soundness**: $\forall \tilde{\mathcal{P}}$, there exists an efficient $\mathcal{E}$ that extracts witness s s.t. $((\phi, n), s) \in \mathcal{R}^{(\alpha)}$ w.p. $\geq 1 - \epsilon(\lambda)$



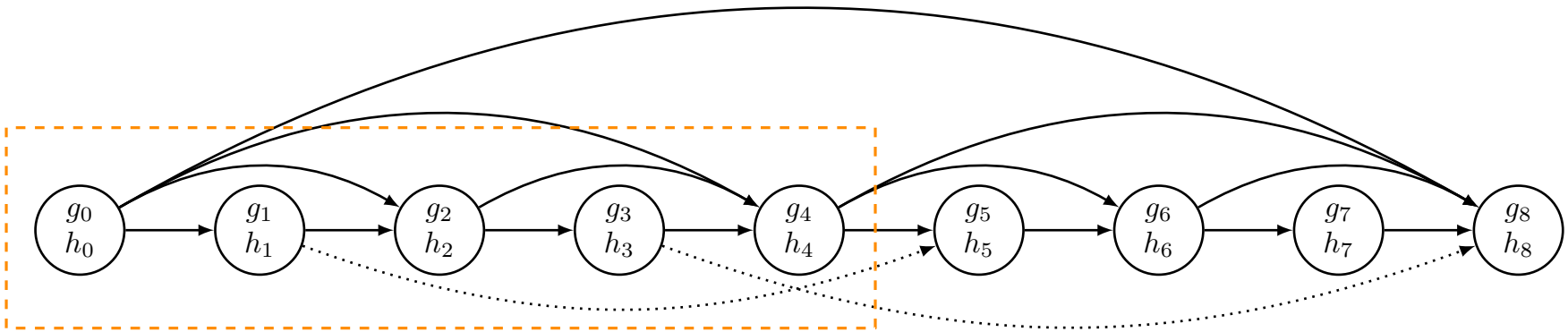
Labeling the PoSW-Augmented Blockchain

Augmented weighted DAG $\Gamma_n := (K := ([n]_0, E_K := E_H \cup E_G), \Omega_n)$



Labeling the PoSW-Augmented Blockchain

Augmented weighted DAG $\Gamma_n := (K := ([n]_0, E_K := E_H \cup E_G), \Omega_n)$



Algorithm Init:

On input 1^λ and ψ:

1. $\chi \leftarrow \{0,1\}^\lambda$
2. $\ell_0 := \tau_0(\varepsilon)$
3. $(\phi_0, \mathsf{aux}_0) \leftarrow \mathsf{Com.commit}(\ell_0)$
4. $g_0 := \ell_0 \| \phi_0$
5. $h_0 := (0, d_0 := \psi \| \chi \| \pi_0 := \varepsilon)$
6. **return** $(\psi := L_K(0) := k_0 := (g_0, h_0), \mathsf{aux}_0)$

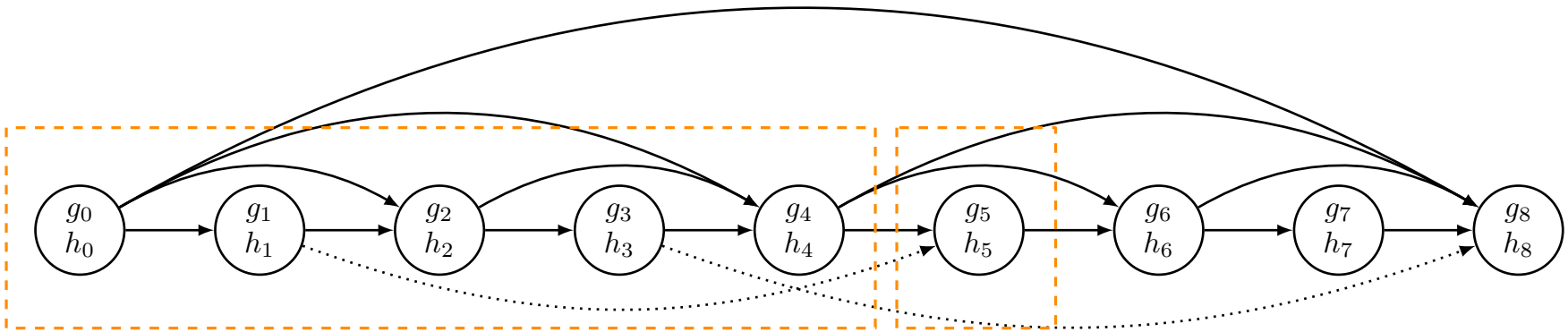
Algorithm Mine:

On input $\big((k_j := (g_j, h_j))_{j \in [i-1]_0}, \mathsf{data}_i\big)$:

1. $\ell_i := \tau(i, L_K(\mathsf{parents}_K(i)))$
2. $(\phi_i, \mathsf{aux}_i) \leftarrow \mathsf{Com.commit}((k_j)_{j \in [i-1]_0} \| \ell_i)$
3. $g_i := \ell_i \| \phi_i$
4. Compute π_i such that $R_\psi\big(i, (g_i, h_i := (i, \mathsf{data}_i, \pi_i)), L_K(\mathsf{parents}_H(i))\big) = 1$
5. **return** $(L_K(i) := k_i := (g_i, h_i), \mathsf{aux}_i)$

Labeling the PoSW-Augmented Blockchain

Augmented weighted DAG $\Gamma_n := (K := ([n]_0, E_K := E_H \cup E_G), \Omega_n)$



Algorithm Init:

On input 1^λ and ψ:

1. $\chi \leftarrow \{0,1\}^\lambda$
2. $\ell_0 := \tau_0(\varepsilon)$
3. $(\phi_0, \mathsf{aux}_0) \leftarrow \mathsf{Com.commit}(\ell_0)$
4. $g_0 := \ell_0 \| \phi_0$
5. $h_0 := (0, d_0 := \psi \| \chi \| \pi_0 := \varepsilon)$
6. **return** $(\psi := L_K(0) := k_0 := (g_0, h_0), \mathsf{aux}_0)$

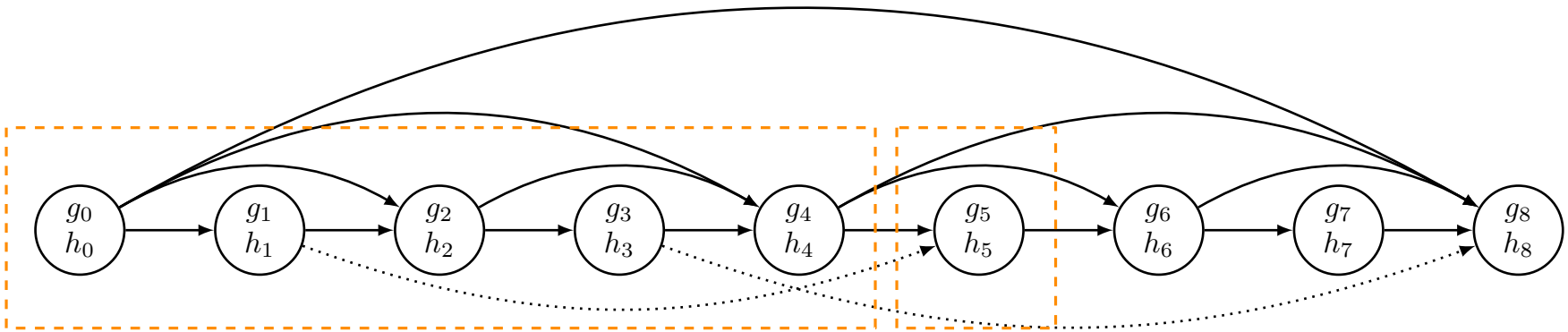
Algorithm Mine:

On input $\big((k_j := (g_j, h_j))_{j \in [i-1]_0}, \mathsf{data}_i\big)$:

1. $\ell_i := \tau(i, L_K(\mathsf{parents}_K(i)))$
2. $(\phi_i, \mathsf{aux}_i) \leftarrow \mathsf{Com.commit}((k_j)_{j \in [i-1]_0} \| \ell_i)$
3. $g_i := \ell_i \| \phi_i$
4. Compute π_i such that $R_\psi\big(i, (g_i, h_i := (i, \mathsf{data}_i, \pi_i)), L_K(\mathsf{parents}_H(i))\big) = 1$
5. **return** $(L_K(i) := k_i := (g_i, h_i), \mathsf{aux}_i)$

Labeling the PoSW-Augmented Blockchain

Augmented weighted DAG $\Gamma_n := (K := ([n]_0, E_K := E_H \cup E_G), \Omega_n)$



Algorithm Init:

On input 1^λ and ψ:

1. $\chi \leftarrow \{0,1\}^\lambda$
2. $\ell_0 := \tau_0(\varepsilon)$
3. ~~$(\phi_0, \mathsf{aux}_0) \leftarrow \mathsf{Com.commit}(\ell_0)$~~
4. $g_0 := \ell_0 \| \sout{\phi_0}$
5. $h_0 := (0, d_0 := \psi \| \chi \| \pi_0 := \varepsilon)$
6. **return** $(\psi := L_K(0) := k_0 := (g_0, h_0), \mathsf{aux}_0)$

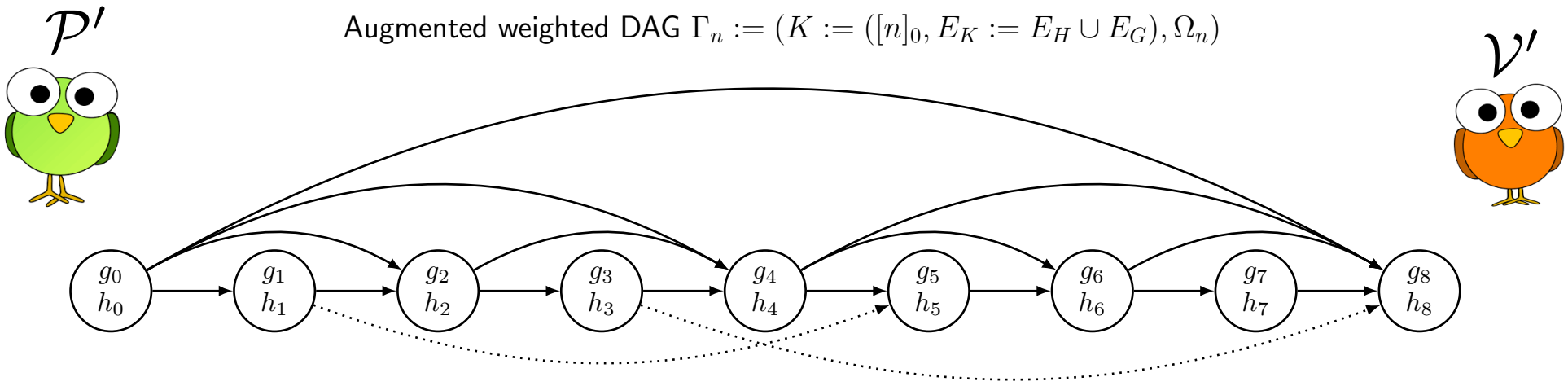
Algorithm Mine:

On input $\big((k_j := (g_j, h_j))_{j \in [i-1]_0}, \mathsf{data}_i\big)$:

1. $\ell_i := \tau(i, L_K(\mathsf{parents}_K(i)))$
2. ~~$(\phi_i, \mathsf{aux}_i) \leftarrow \mathsf{Com.commit}((k_j)_{j \in [i-1]_0} \| \ell_i)$~~
3. $g_i := \ell_i \| \sout{\phi_i}$

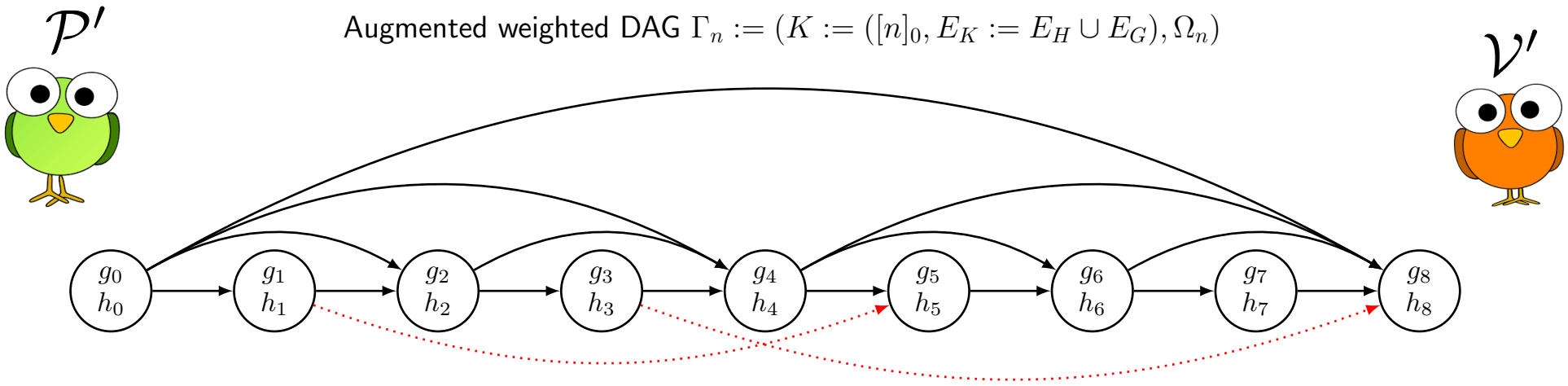
   for this PoSW constr.: $\phi_i := \ell_i$
4. Compute π_i such that $R_\psi\big(i, (g_i, h_i := (i, \mathsf{data}_i, \pi_i)), L_K(\mathsf{parents}_H(i))\big) = 1$
5. **return** $(L_K(i) := k_i := (g_i, h_i), \mathsf{aux}_i)$

SNACK



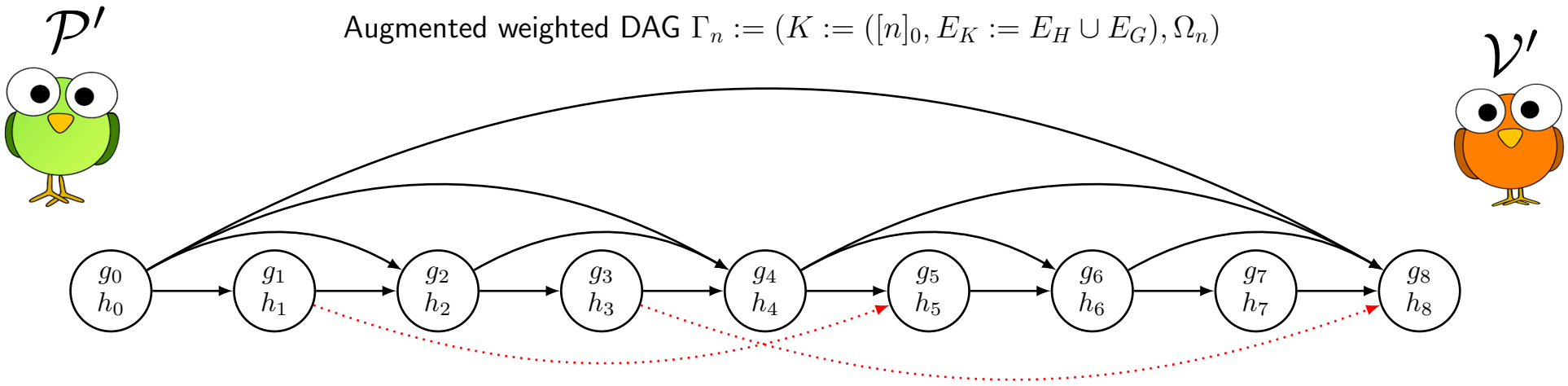
$$\mathcal{R}^{(\alpha)} = \left\{ \begin{array}{l} ((\phi_n, n), \\ (P, L_P, (w_i)_{i\in P}, \rho)) \end{array} : \begin{array}{l} (P, L_P, (w_i)_{i\in P}) \text{ is } (\alpha, R)\text{-valid} \\ \wedge\ \mathsf{Comm.ver}(\phi_n, L_P, P, \rho) = 1 \end{array} \right\}$$

SNACK



$$\mathcal{R}^{(\alpha)} = \left\{ \begin{array}{l} ((\phi_n, n), \\ (P, L_P, (w_i)_{i\in P}, \rho)) \end{array} : \begin{array}{l} (P, L_P, (w_i)_{i\in P}) \text{ is } (\alpha, R)\text{-valid} \\ \wedge\ \mathsf{Comm.ver}(\phi_n, L_P, P, \rho) = 1 \end{array} \right\}$$

SNACK



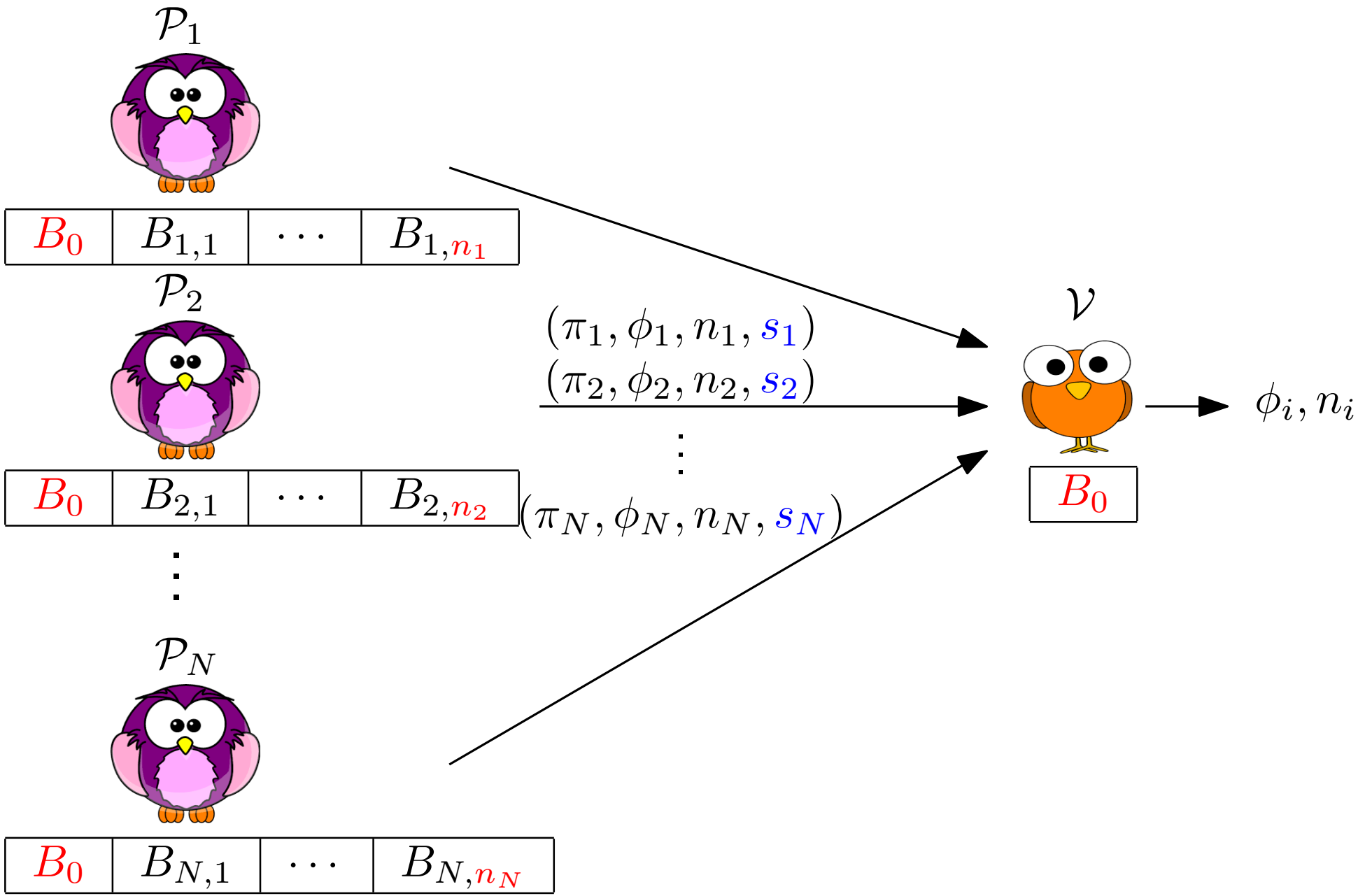
$$\mathcal{R}^{(\alpha)} = \left\{ \begin{array}{l} ((\phi_n, n), \\ (P, L_P, (w_i)_{i \in P}, \rho)) \end{array} : \begin{array}{l} (P, L_P, (w_i)_{i \in P}) \text{ is } (\alpha, R)\text{-valid} \\ \wedge\ \text{Comm.ver}(\phi_n, L_P, P, \rho) = 1 \end{array} \right\}$$

Thm: (α, ϵ)-knowledge soundness of SNACK follows directly from any (α, ϵ)-knowledge soundness of GL-PoSW.

(after applying Fiat-Shamir)

SNACK Bootstrapping

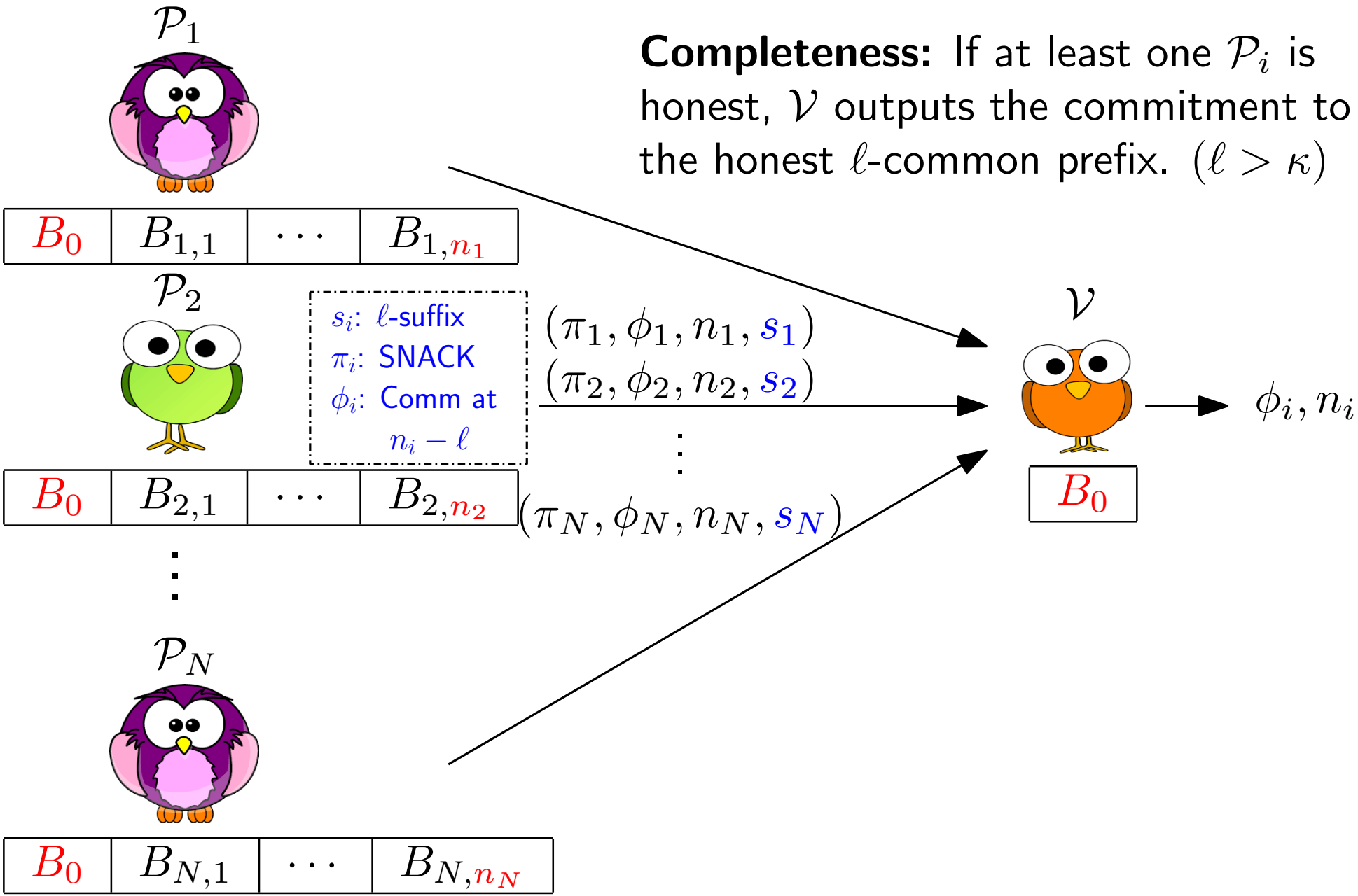
Goal: $\mathcal{V}$ holds a commitment to a sufficiently-long stable prefix



SNACK Bootstrapping

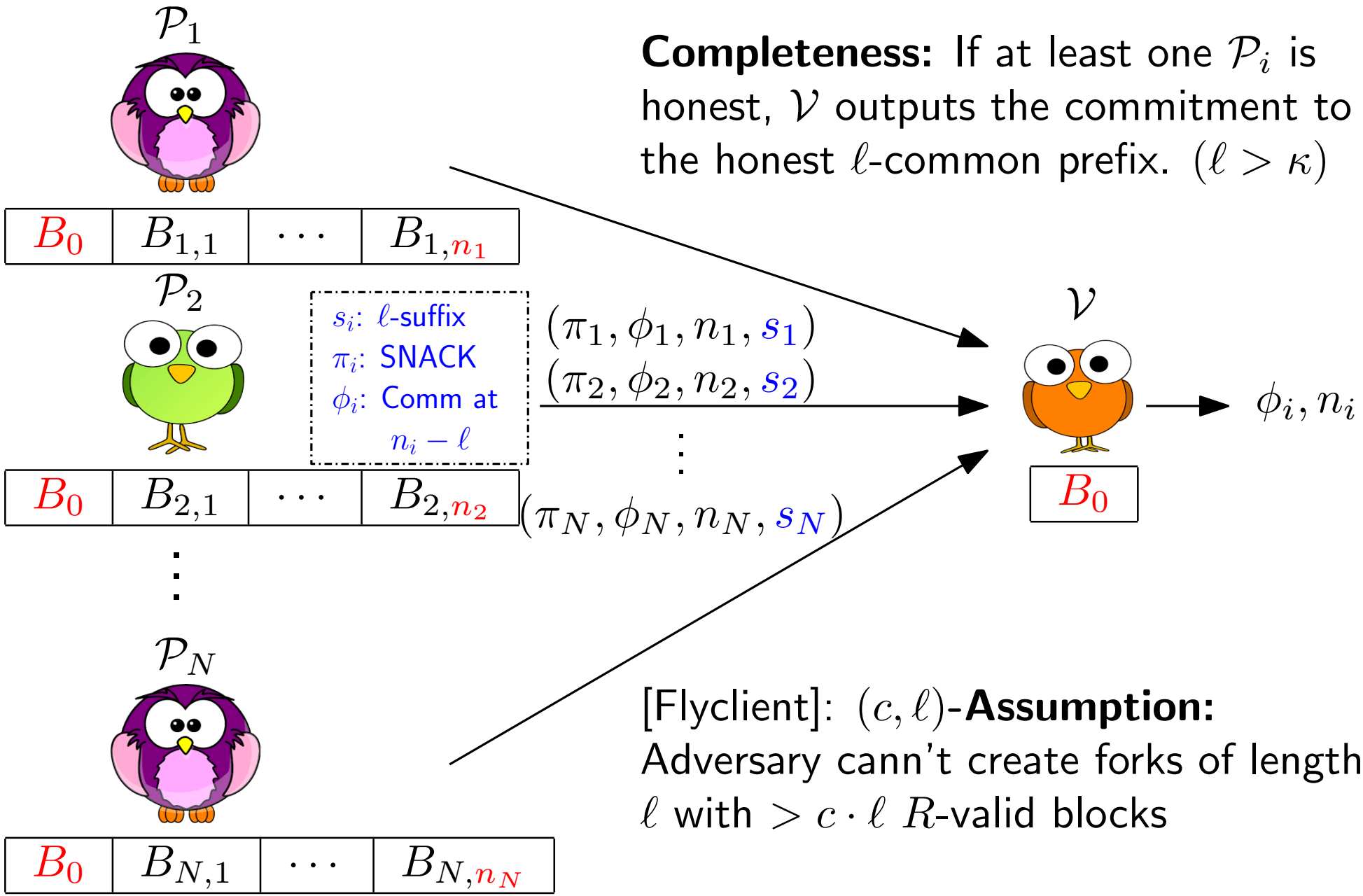
Goal: $\mathcal{V}$ holds a commitment to a sufficiently-long stable prefix

Completeness: If at least one $\mathcal{P}_i$ is honest, $\mathcal{V}$ outputs the commitment to the honest ℓ-common prefix. $(\ell > \kappa)$



SNACK Bootstrapping

Goal: $\mathcal{V}$ holds a commitment to a sufficiently-long stable prefix



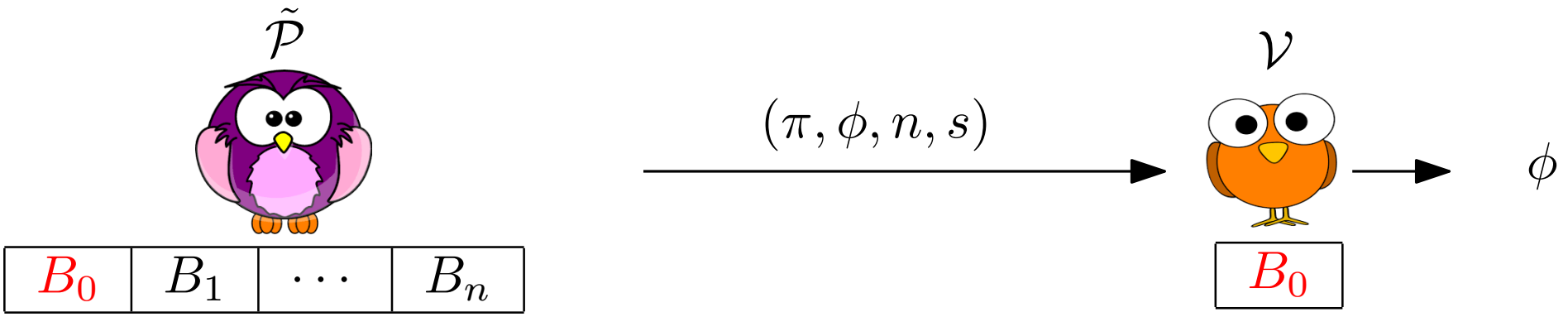
Completeness: If at least one $\mathcal{P}_i$ is honest, $\mathcal{V}$ outputs the commitment to the honest ℓ-common prefix. $(\ell > \kappa)$

[Flyclient]: (c, ℓ)-**Assumption:** Adversary cann't create forks of length ℓ with $> c \cdot \ell$ R-valid blocks

SNACK Bootstrapping

(α_n, ϵ)-**Knowledge-Soundness:** From any malicious $\tilde{\mathcal{P}}$ we can extract an (α_n, R)-valid path except with ϵ prob.

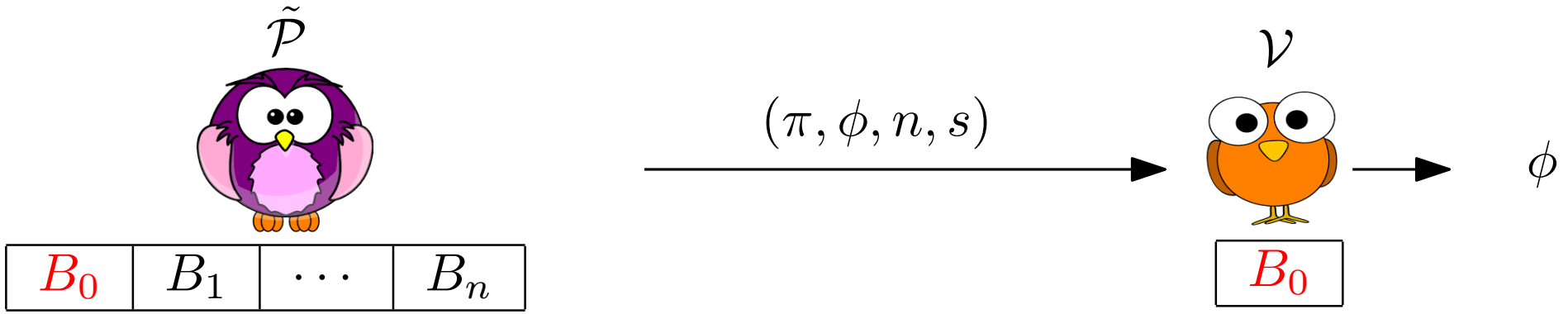
$$\alpha_n = 1 - \left(\log_c \left(\frac{\ell - 1}{n + \ell} \right) \right)^{-1}$$



SNACK Bootstrapping

(α_n, ϵ)-**Knowledge-Soundness:** From any malicious $\tilde{\mathcal{P}}$ we can extract an (α_n, R)-valid path except with ϵ prob.

$$\alpha_n = 1 - \left(\log_c \left(\frac{\ell - 1}{n + \ell} \right) \right)^{-1}$$



Almost best possible guarantee: Meaningful for blockchains where it is moderately hard to generate a satisfying input for R

Thank you