An Incremental PoSW for General Weight Distributions

Hamza Abusalah and Valerio Cini

institute IMDEA software

AUSTRIAN INSTITUTE OF TECHNOLOGY

Eurocrypt 2023 in Lyon, France

FWF Der Wissenschaftsfonds.

Outline

Proofs of Sequential Work, Standalone (PoSW) and Incremental (iPoSW)

The Skiplist PoSW

Make it Incremental (iPoSW)

Generalize it to *General Weight Distributions* (Motivated by Blockchain Applications)

All Constructions Are in the ROM

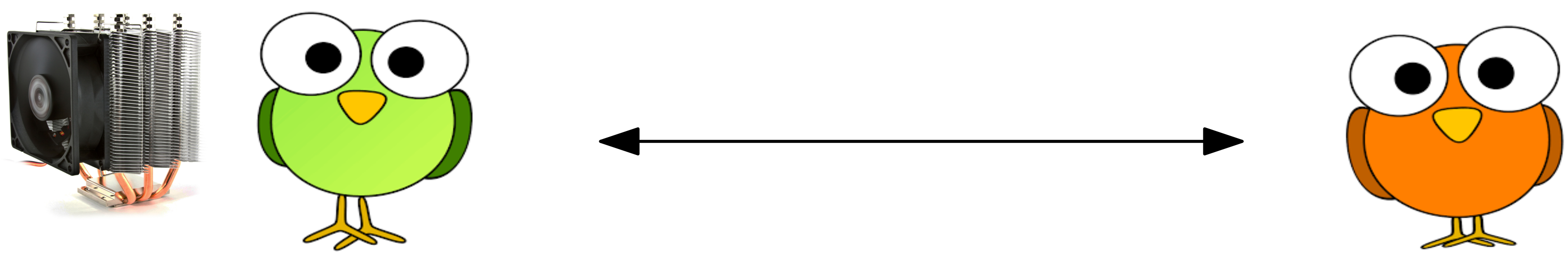
(We don't cover (continuous) verifiable delay functions)

PoSW

[Mahmoody-Moran-Vadhan'13]

Parameter: n

$\mathsf{P}^{\tau(\cdot)}$ $\mathsf{V}^{\tau(\cdot)}$



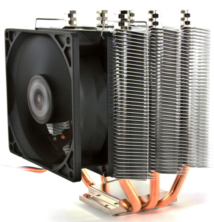
Completeness: Honest $\mathsf{P}^{\tau(\cdot)}$ making n **sequential** $\tau(\cdot)$ queries makes V accept w.p. 1

PoSW

[Mahmoody-Moran-Vadhan'13]

Parameter: n

$\mathsf{P}^{\tau(\cdot)}$



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

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

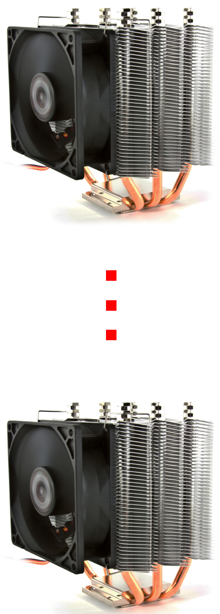
Completeness: Honest $\mathsf{P}^{\tau(\cdot)}$ making n **sequential** $\tau(\cdot)$ queries makes V accept w.p. 1

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

PoSW

[Mahmoody-Moran-Vadhan'13]

Parameter: n



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

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

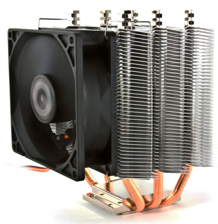
poly$(\lambda, \log n)$ time

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

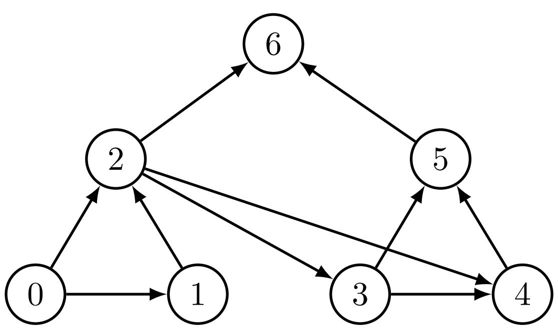
All PoSW Constructions Look Like

Parameters: $G_n, t, \cdots$

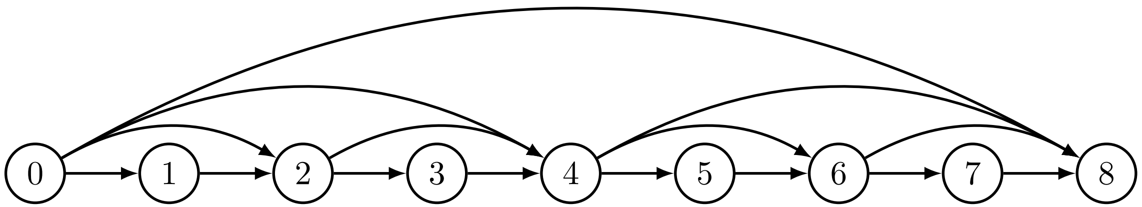
$\mathsf{P}^{\tau(\cdot)}$



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



[Cohen-Pietrzak'18]



[Abusalah-Kamath-Klein-Walter-Pietrzak'19][Abusalah-Fuchsbauer-Gaži-Klein'22]

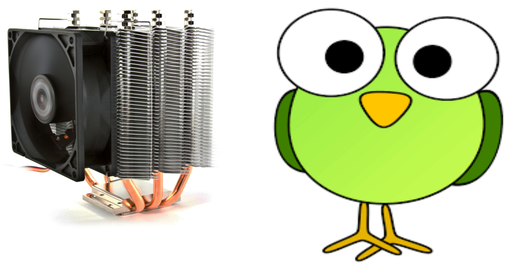
All PoSW Constructions Look Like

Parameters: $G_n, t, \cdots$

$\mathsf{P}^{\tau(\cdot)}$ $\qquad$ $\mathsf{V}^{\tau(\cdot)}$

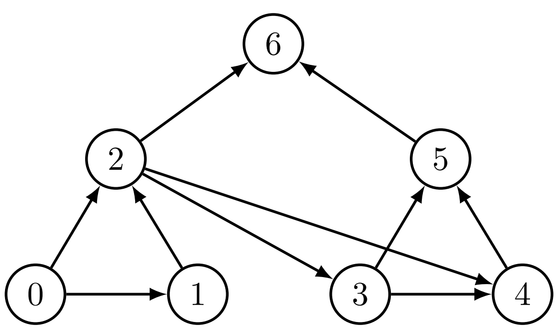
$$\mathsf{P} \xleftarrow{\ \chi\ } \mathsf{V}$$

$$\mathsf{P} \xrightarrow{\ L(n)\ } \mathsf{V}$$

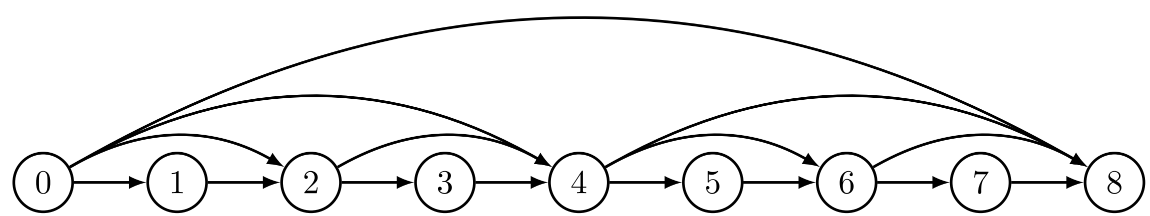


Random oracle $\tau : \{0,1\}^* \to \{0,1\}^\lambda$ with $\tau := \tau(\chi, \cdot)$

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



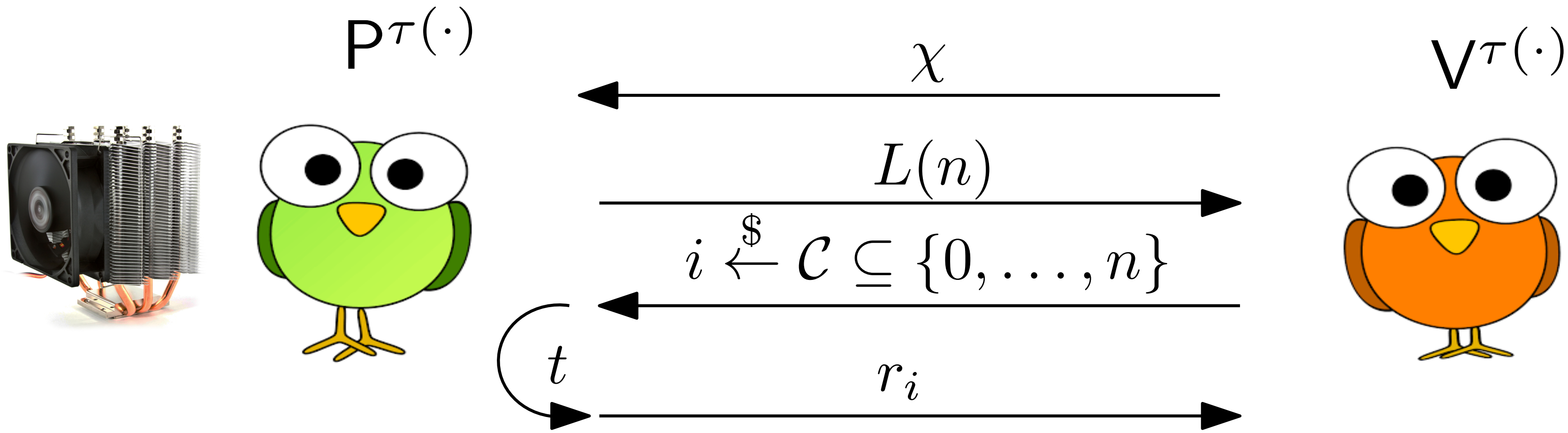
[Cohen-Pietrzak'18]



[Abusalah-Kamath-Klein-Walter-Pietrzak'19][Abusalah-Fuchsbauer-Gaži-Klein'22]

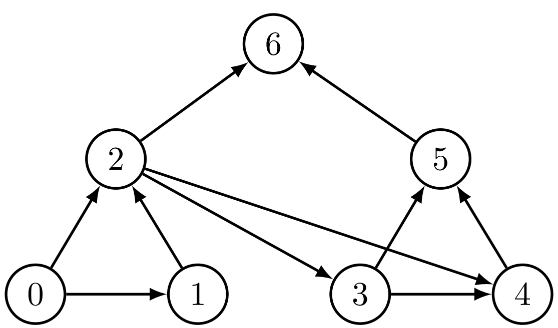
All PoSW Constructions Look Like

Parameters: $G_n, t, \cdots$

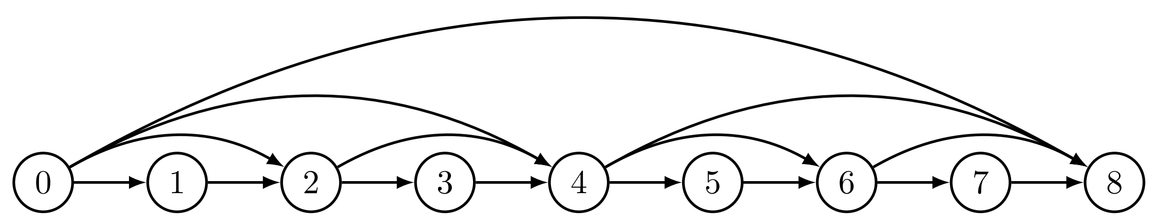


Random oracle $\tau : \{0,1\}^* \to \{0,1\}^\lambda$ with $\tau := \tau(\chi, \cdot)$

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



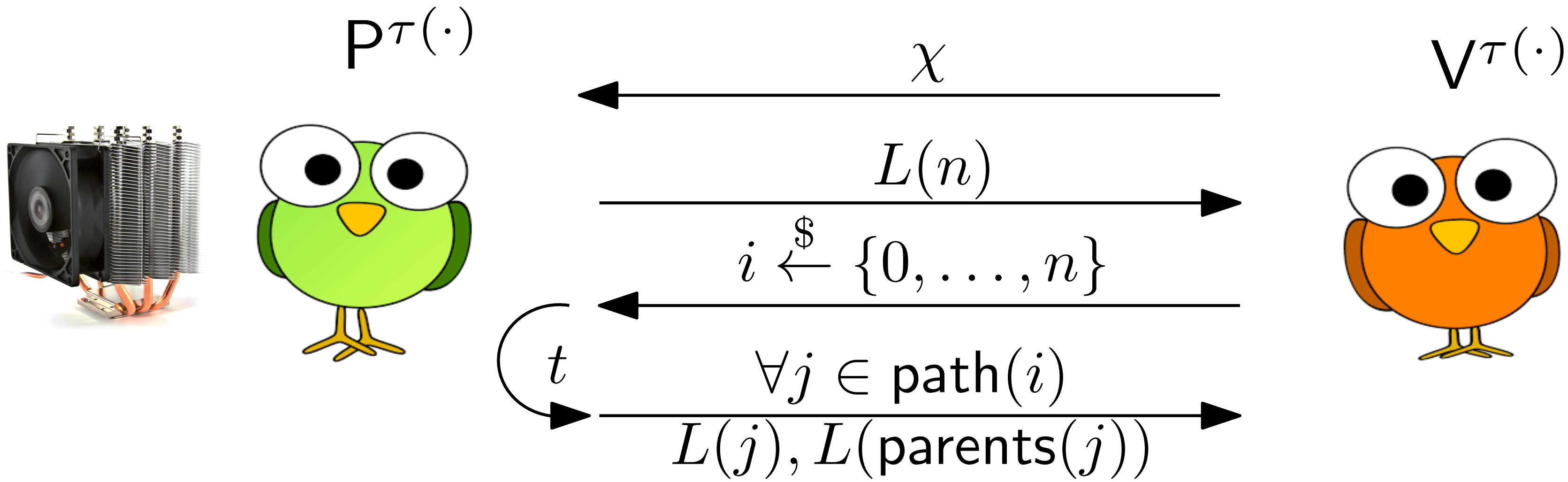
[Cohen-Pietrzak'18]

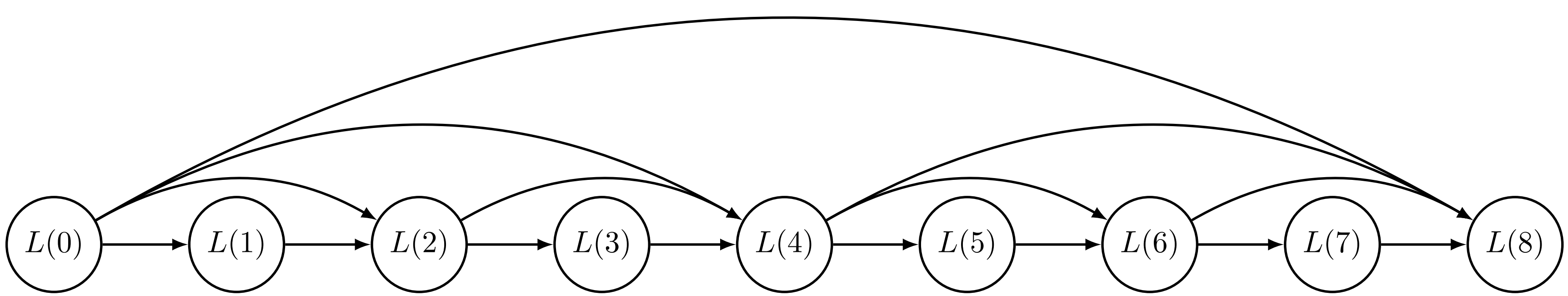


[Abusalah-Kamath-Klein-Walter-Pietrzak'19][Abusalah-Fuchsbauer-Gaži-Klein'22]

The Skiplist PoSW

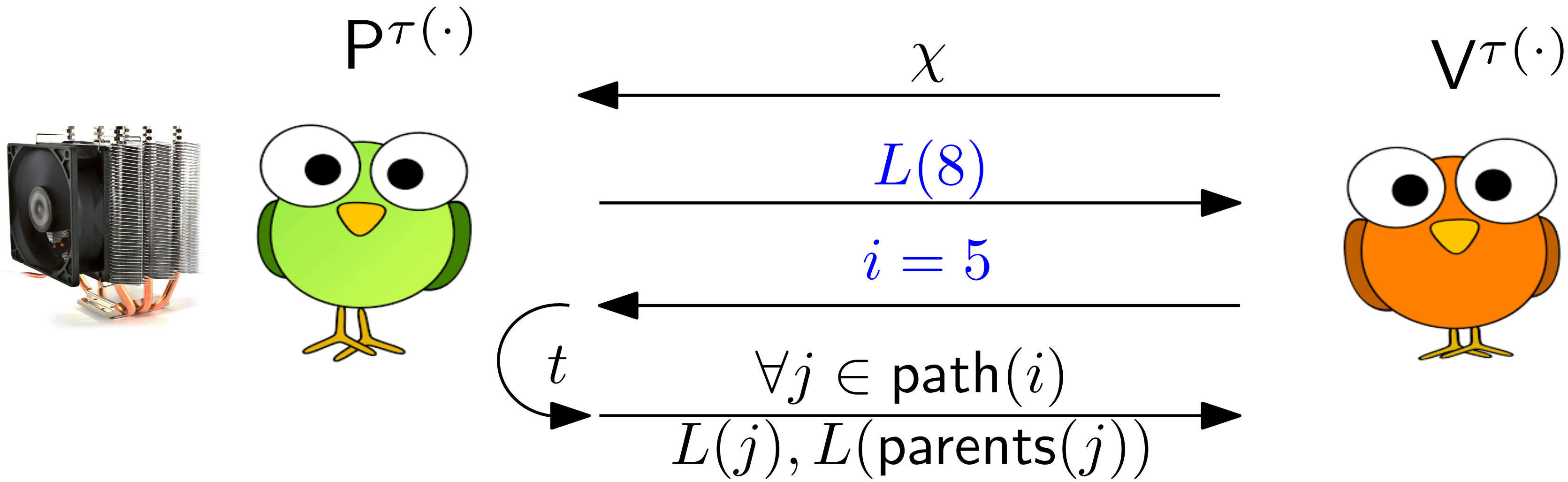
[Abusalah-Fuchsbauer-Gaži-Klein'22]

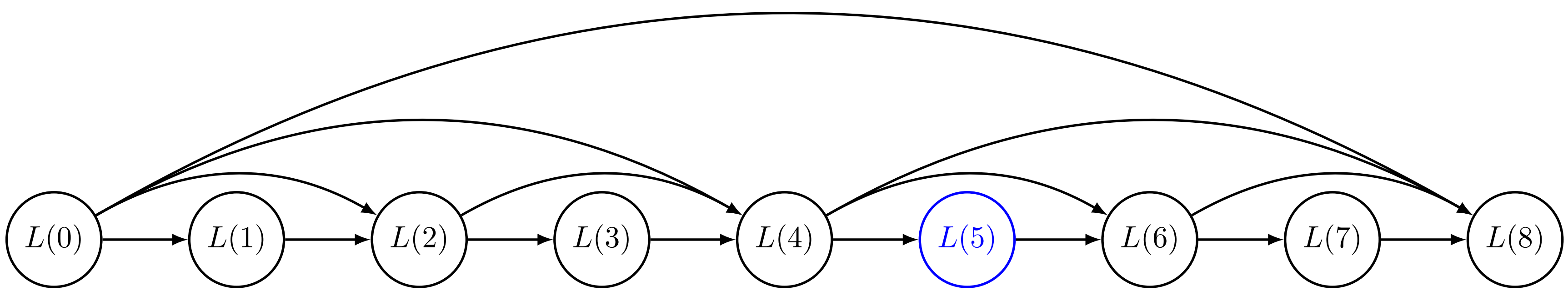




The Skiplist PoSW

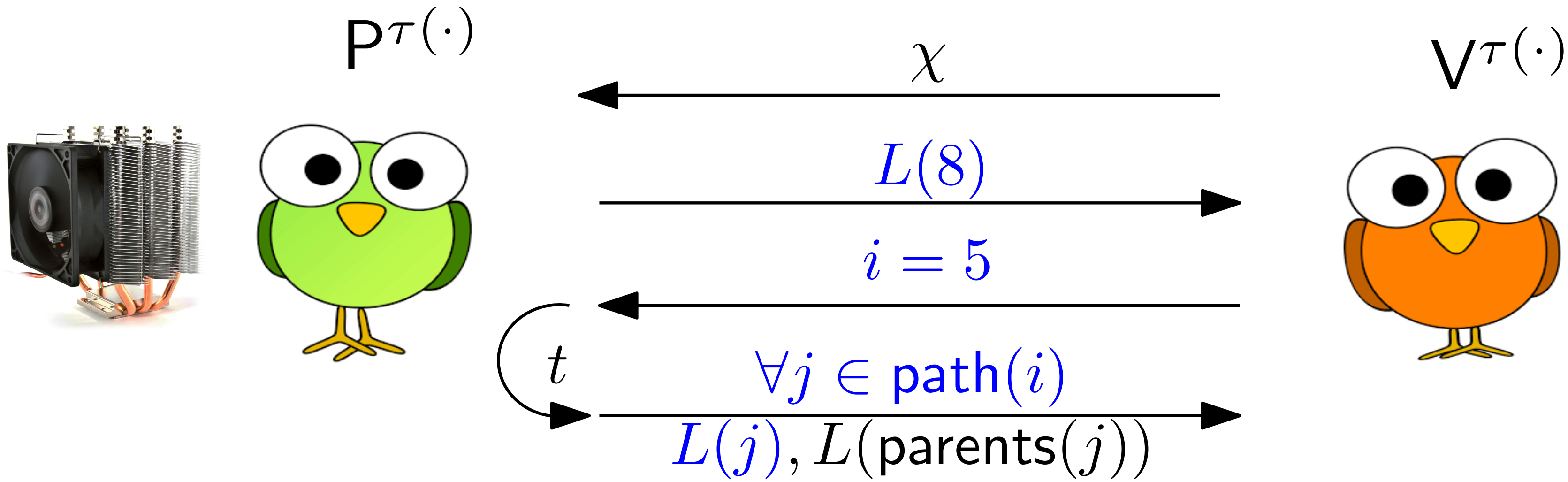
[Abusalah-Fuchsbauer-Gaži-Klein'22]

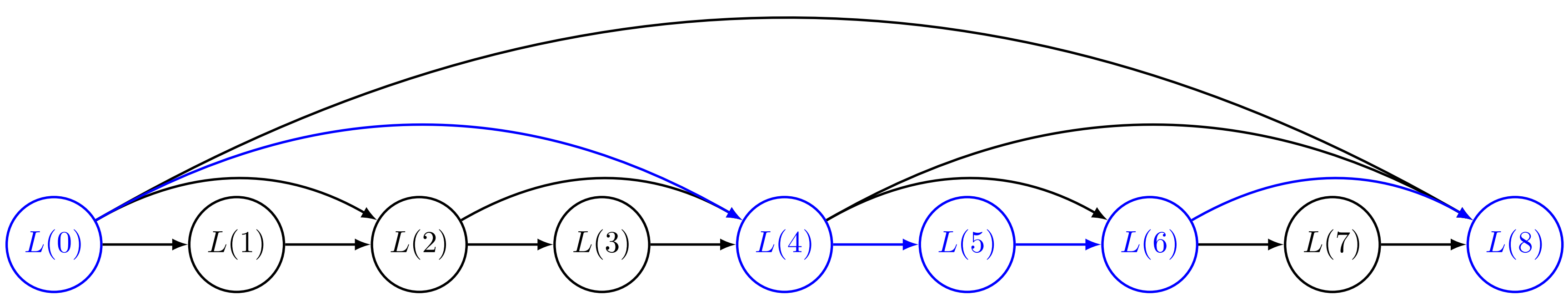




The Skiplist PoSW

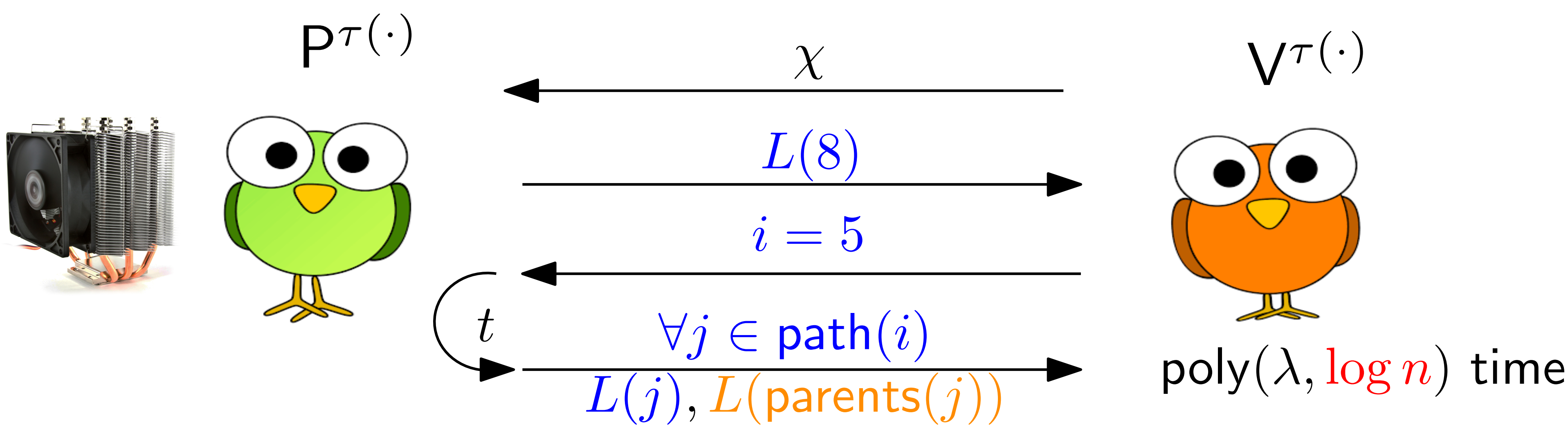
[Abusalah-Fuchsbauer-Gaži-Klein'22]

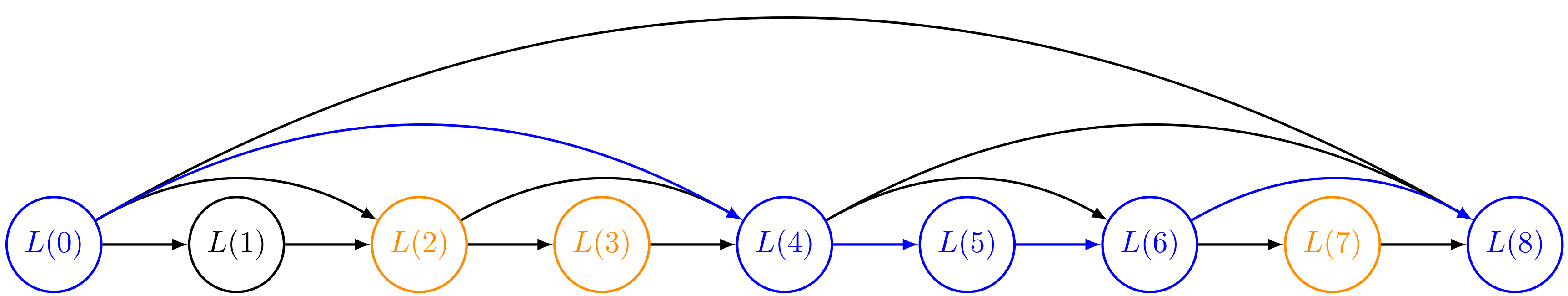




The Skiplist PoSW

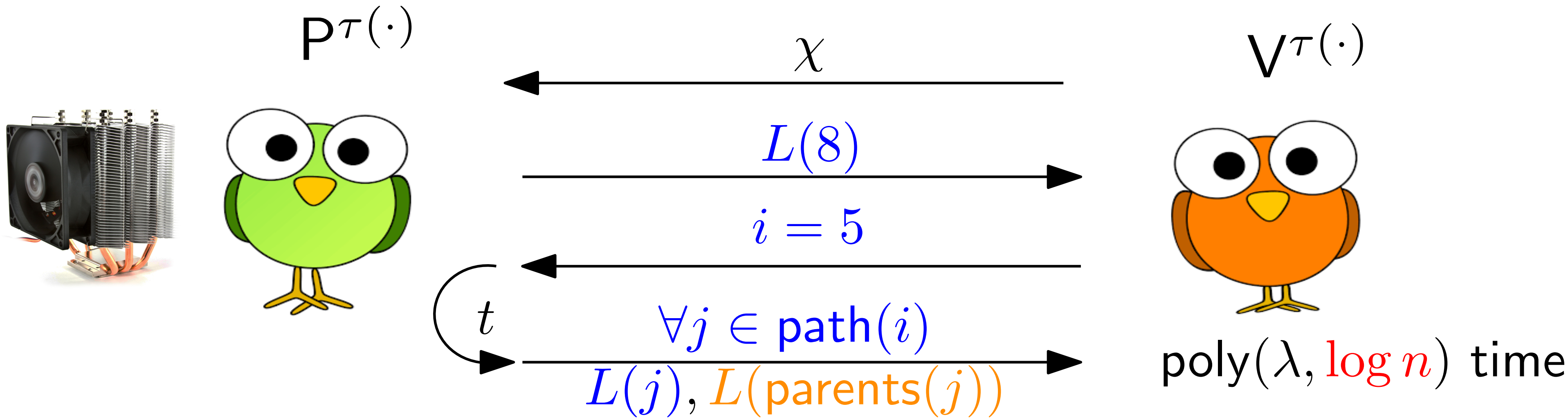
[Abusalah-Fuchsbauer-Gaži-Klein'22]





The Skiplist PoSW

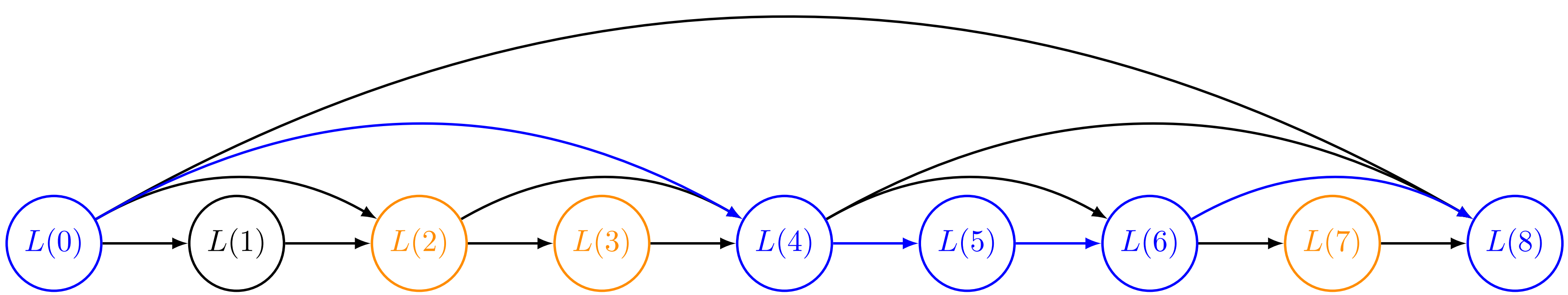
[Abusalah-Fuchsbauer-Gaži-Klein'22]



Thm: If 1. $\tilde{\mathsf{P}}_1$ made $\leq \alpha \cdot n$ sequential queries to $\tau(\cdot)$ before sending $L(n)$

2. $\tilde{\mathsf{P}} := (\tilde{\mathsf{P}}_1, \tilde{\tilde{\mathsf{P}}}_2)$ made a total of $\leq q$ queries to $\tau(\cdot)$

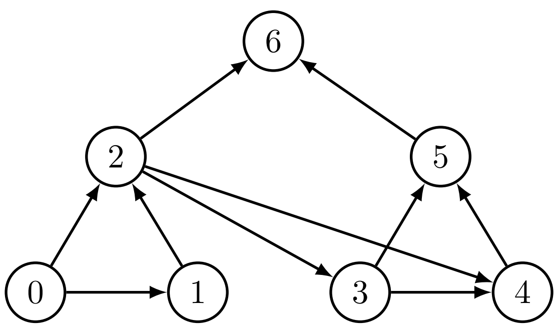
Then $\tilde{\mathsf{P}}$ makes V accept w.p. $\leq \alpha^t + 3 \cdot q^2/2^\lambda$

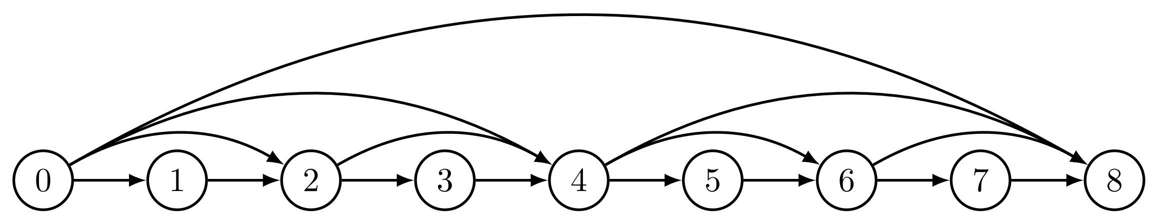


On Our Way to iPoSW

To answer challenges, P has two extremes

- **store all labels** $L(0), \ldots, L(n)$: answering a challenge is just a look-up
- store nothing and spend an extra n **sequential steps** to relabel and answer



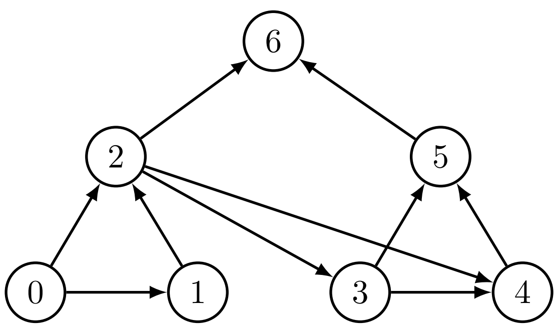


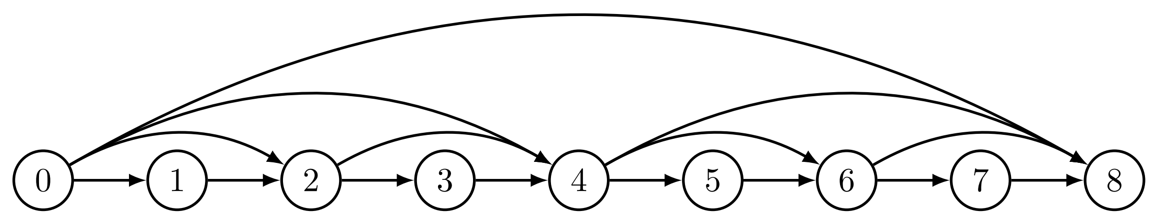
On Our Way to iPoSW

To answer challenges, P has two extremes

- **store all labels** $L(0), \ldots, L(n)$: answering a challenge is just a look-up

- store nothing and spend an extra n **sequential steps** to relabel and answer





Space-time tradeoffs:

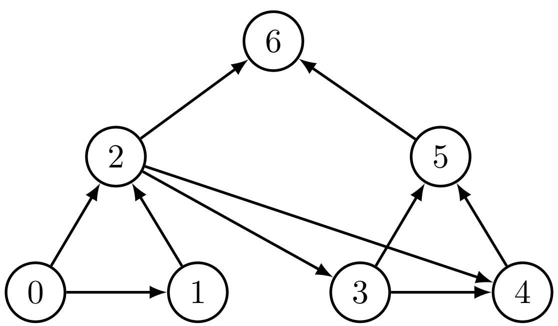
store $\sqrt{n}$ labels and spend an extra $\sqrt{n}$ sequential steps

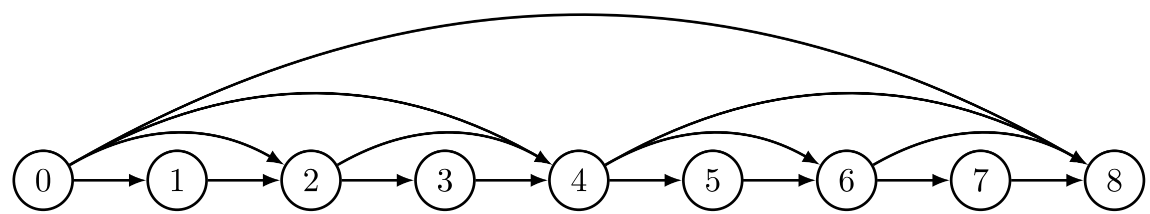
On Our Way to iPoSW

To answer challenges, P has two extremes

- **store all labels** $L(0), \ldots, L(n)$: answering a challenge is just a look-up

- store nothing and spend an extra n **sequential steps** to relabel and answer





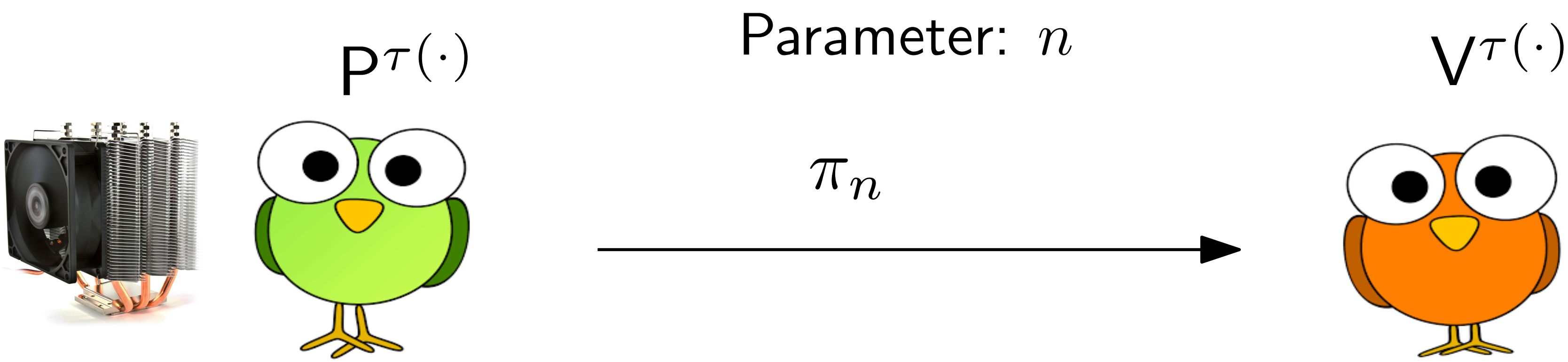
Space-time tradeoffs:

store $\sqrt{n}$ labels and spend an extra $\sqrt{n}$ sequential steps

Question: Best of both worlds: can we store a succinct state and spend no extra time?

iPoSW

[Döttling-Lai-Malavolta'19]

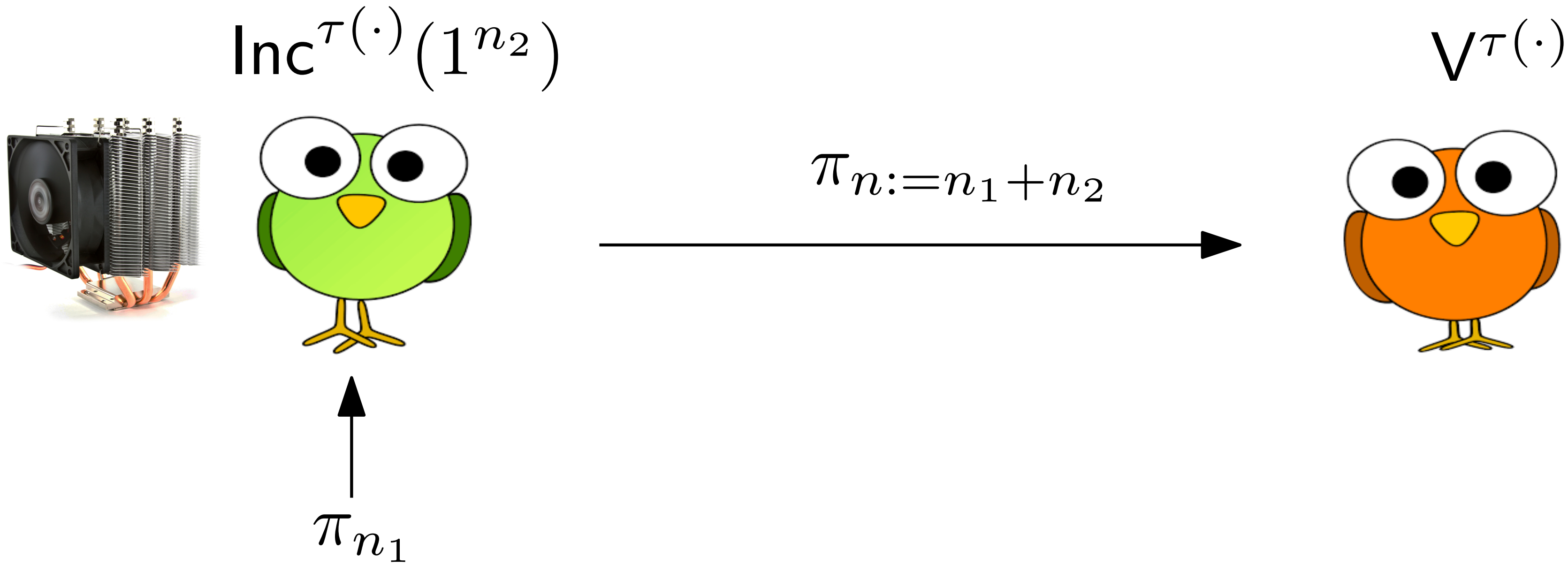


An iPoSW is a *non-interactive* proof system $(\mathsf{P}, \mathsf{V}, \mathsf{Inc})$ where

- (P, V) is a PoSW: complete, sound, and succinct

iPoSW

[Döttling-Lai-Malavolta'19]

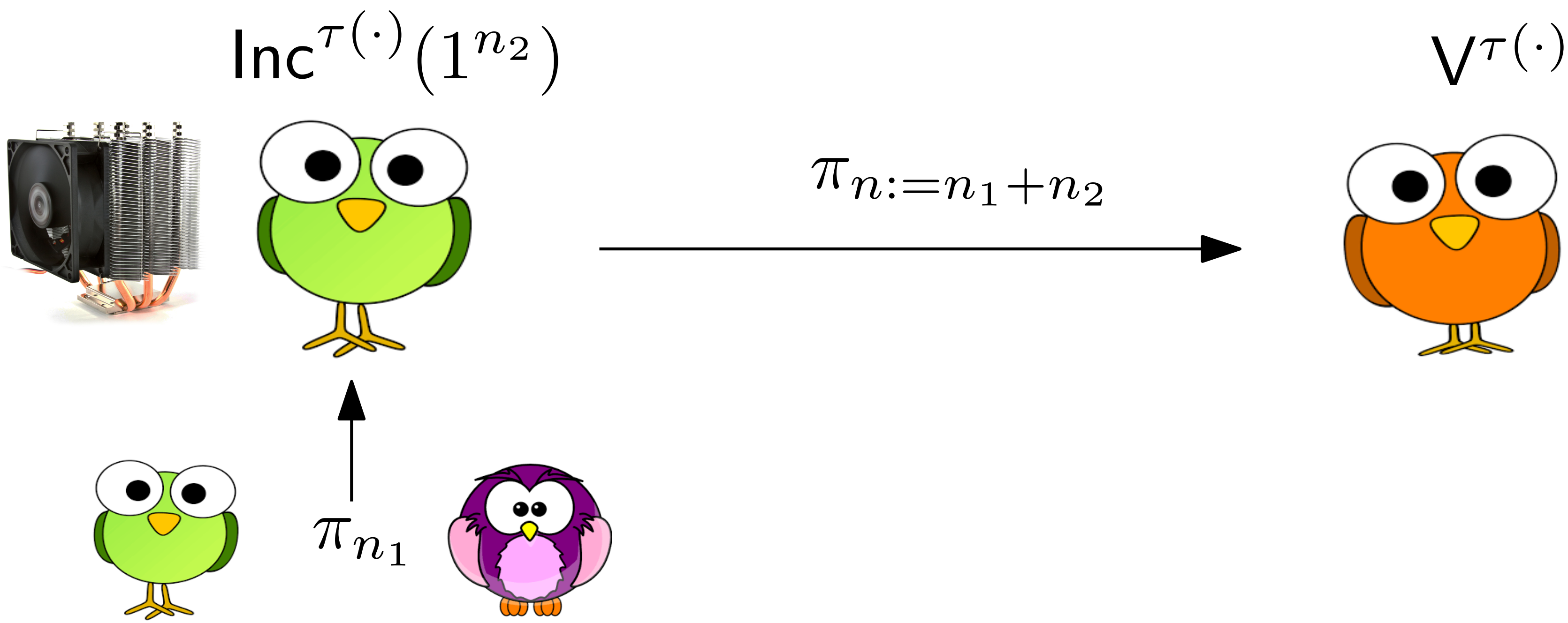


An iPoSW is a *non-interactive* proof system $(\mathsf{P}, \mathsf{V}, \mathsf{Inc})$ where

- (P, V) is a PoSW: complete, sound, and succinct
- Inc: given an **accepting** π_{n_1}, Inc making n_2 **sequential** $\tau(\cdot)$ queries makes V accept

iPoSW

[Döttling-Lai-Malavolta'19]

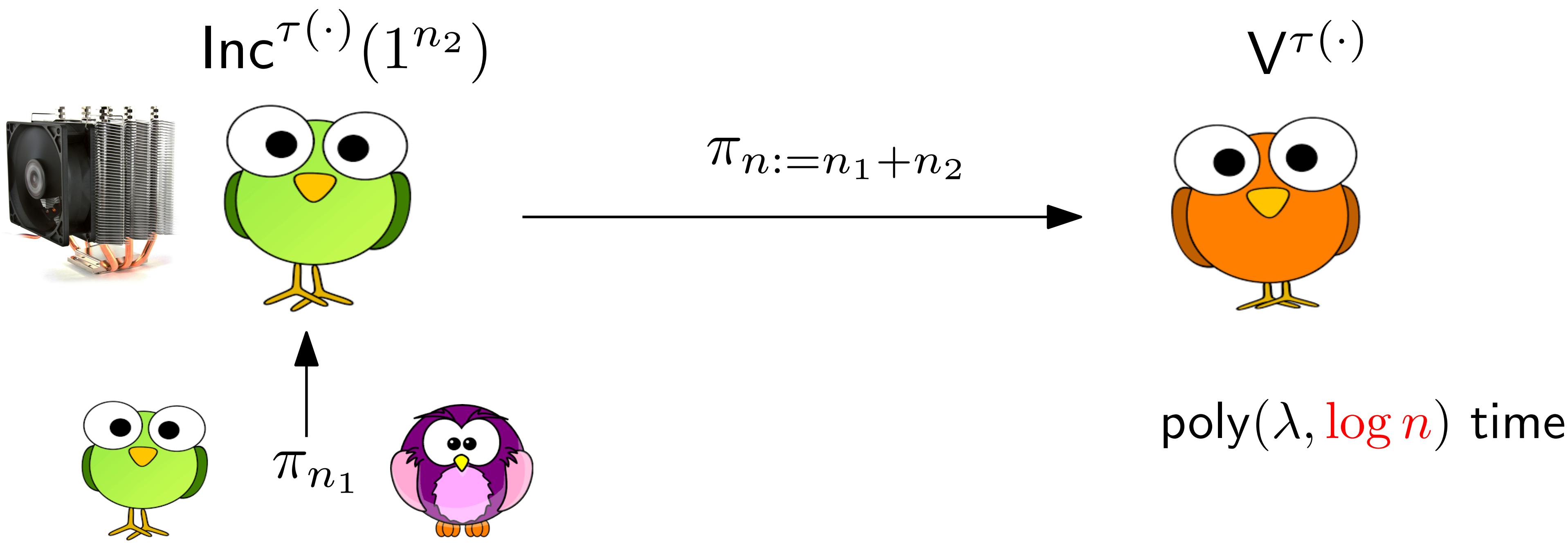


An iPoSW is a *non-interactive* proof system $(\mathsf{P}, \mathsf{V}, \mathsf{Inc})$ where

- (P, V) is a PoSW: complete, sound, and succinct
- Inc: given an **accepting** π_{n_1}, Inc making n_2 **sequential** $\tau(\cdot)$ queries makes V accept

iPoSW

[Döttling-Lai-Malavolta'19]



An iPoSW is a *non-interactive* proof system $(\mathsf{P}, \mathsf{V}, \mathsf{Inc})$ where

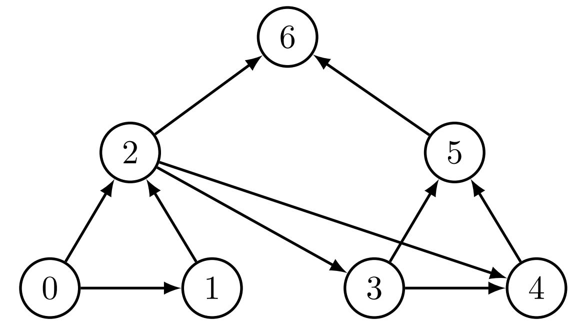
- (P, V) is a PoSW: complete, sound, and succinct
- Inc: given an **accepting** π_{n_1}, Inc making n_2 **sequential** $\tau(\cdot)$ queries makes V accept

$$\pi_{n_1} \to \mathsf{Inc}(n_2) \to \pi_{n_1+n_2} \to \mathsf{Inc}(n_3) \to \pi_{n_1+n_2+n_3} \to \cdots \to \pi_{n=n_1+\cdots+n_k}$$

An iPoSW Construction

[Döttling-Lai-Malavolta'19]

- Döttling-Lai-Malavolta made Cohen-Pietrzak'18 incremental by *sampling challenges on the fly*
- Efficient, yet incurs an extra small security loss

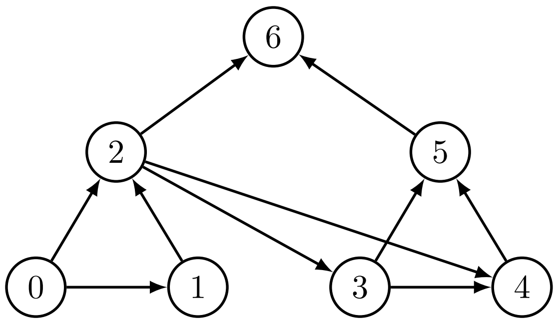


[Cohen-Pietrzak'18]

An iPoSW Construction

[Döttling-Lai-Malavolta'19]

- Döttling-Lai-Malavolta made Cohen-Pietrzak'18 incremental by *sampling challenges on the fly*
- Efficient, yet incurs an extra small security loss



[Cohen-Pietrzak'18]

In this work

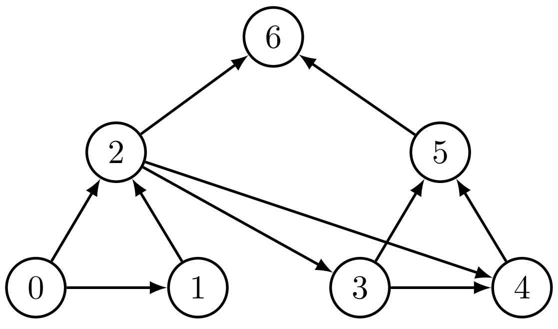
1. We similarily make the skiplist PoSW incremental

   We apply the same on-the-fly sampling

An iPoSW Construction

[Döttling-Lai-Malavolta'19]

- Döttling-Lai-Malavolta made Cohen-Pietrzak'18 incremental by *sampling challenges on the fly*
- Efficient, yet incurs an extra small security loss



[Cohen-Pietrzak'18]

In this work

1. We similarily make the skiplist PoSW incremental

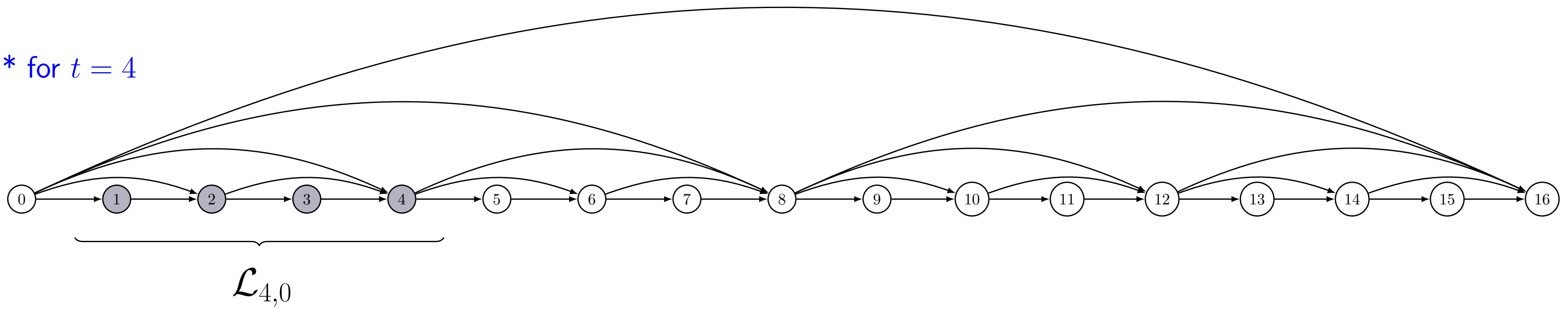
   We apply the same on-the-fly sampling

2. We generalize the skiplist iPoSW to general weight distributions

   We devise a new variant of the on-the-fly sampling technique

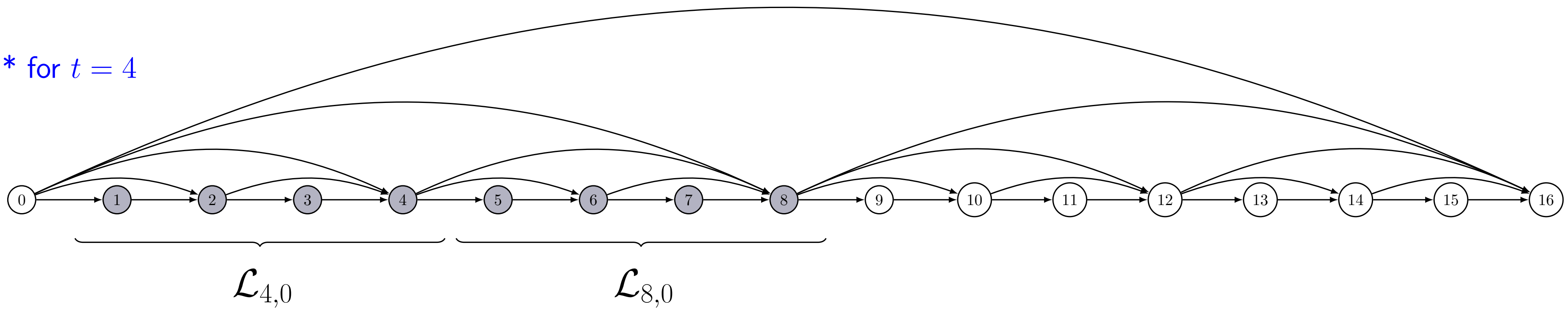
Our Standalone iPoSW: The Prover P

* for $t = 4$

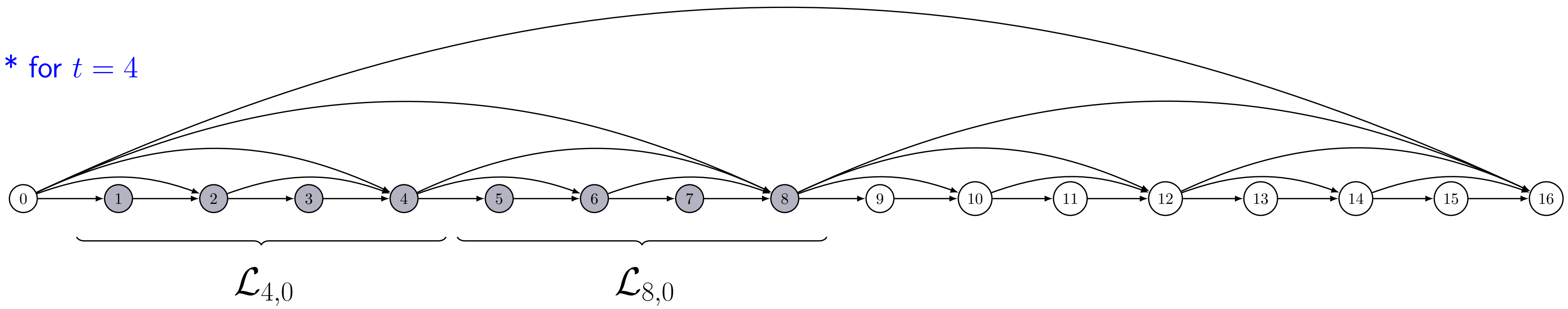


Our Standalone iPoSW: The Prover P

* for $t = 4$

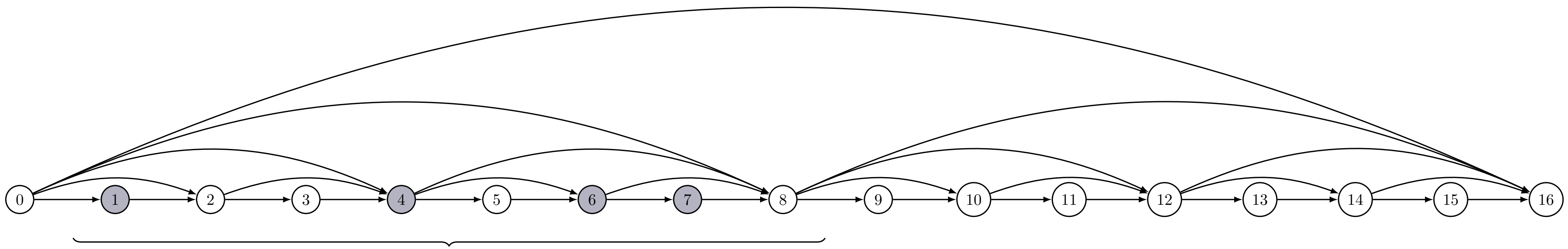


Our Standalone iPoSW: The Prover P



Use randomness from $L(8)$ to randomly sample a set of size 4 from $\{1, \ldots, 8\}$

Our Standalone iPoSW: The Prover P

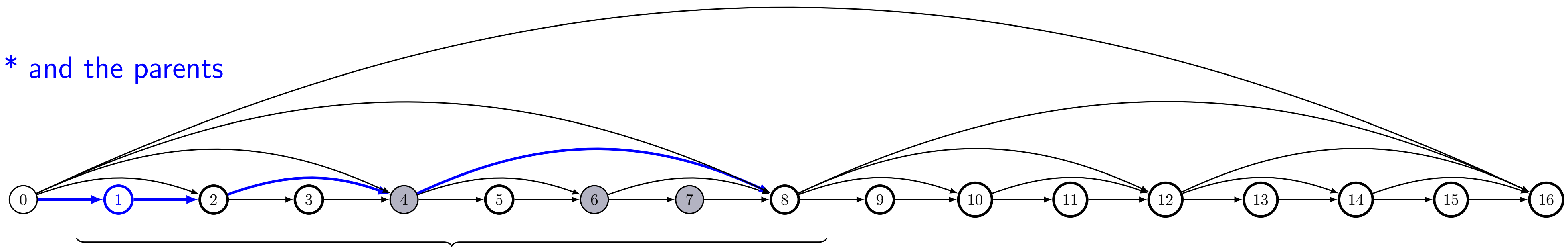


$$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\}$$

$$\pi_i := L(j), L(\text{parents}(j)) \quad \forall j \in \text{path}(i) \text{ in } G_{[0:8]}$$

We have all labels to complile π_i

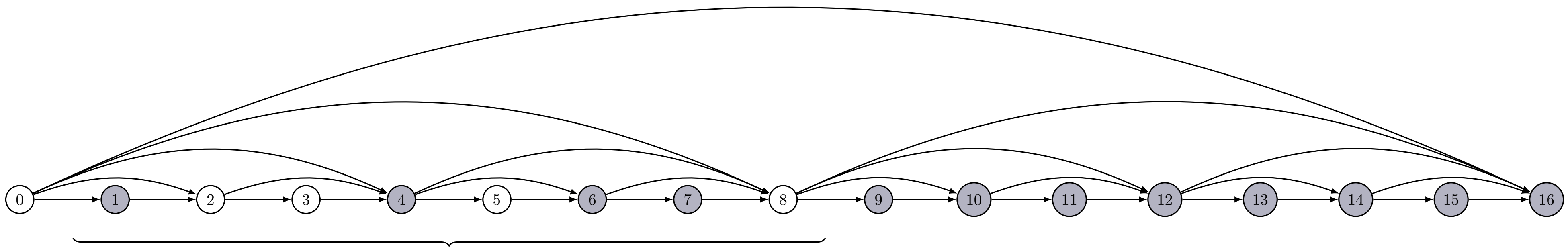
Our Standalone iPoSW: The Prover P



$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\}$

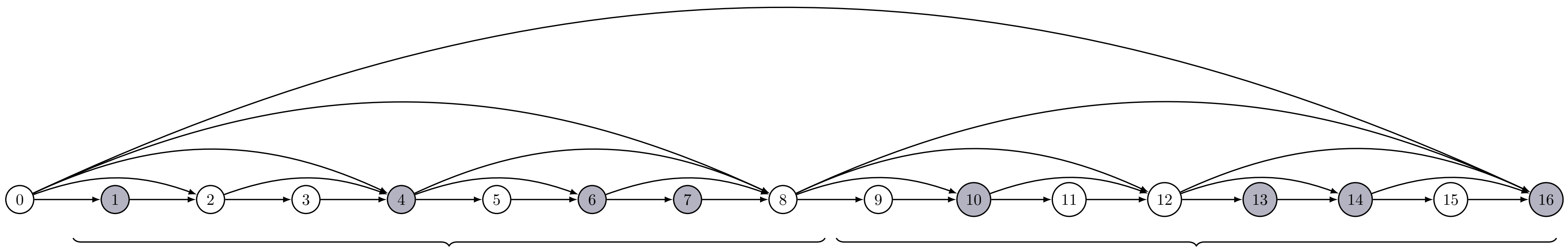
$$\pi_i := L(j), L(\mathsf{parents}(j)) \quad \forall j \in \mathsf{path}(i) \text{ in } G_{[0:8]}$$

Our Standalone iPoSW: The Prover P



$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\}$

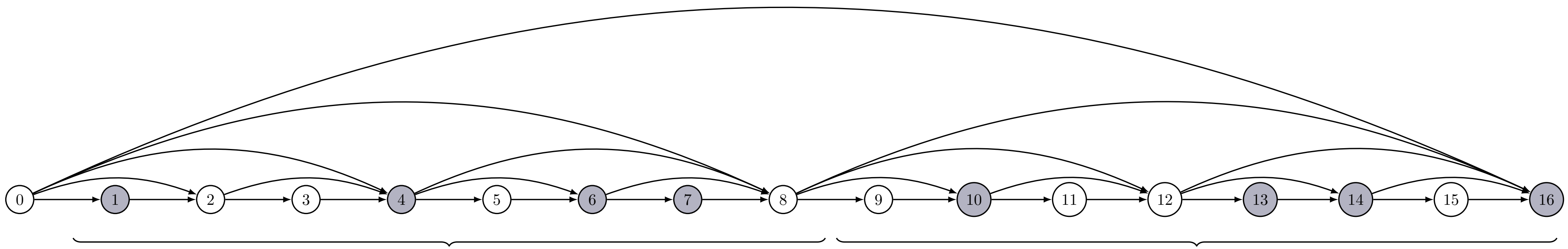
Our Standalone iPoSW: The Prover P



$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\}$ $\mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$

$$\pi_i := L(j), L(\mathsf{parents}(j)) \quad \forall j \in \mathsf{path}(i) \text{ in } G_{[8:16]}$$

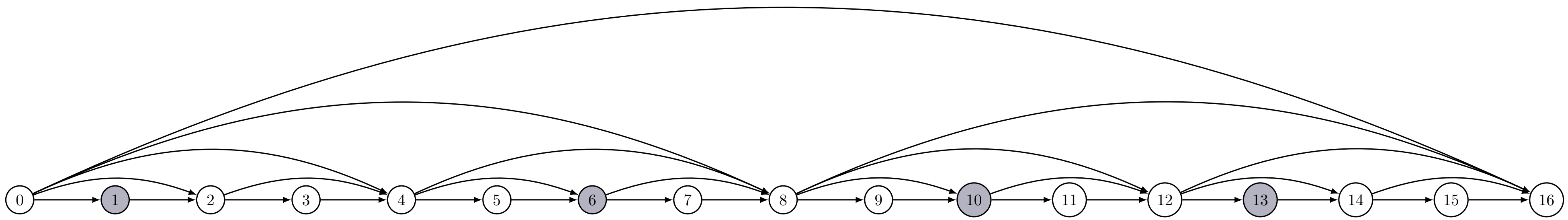
Our Standalone iPoSW: The Prover P



$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\}$ $\mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$

Use randomness from $L(16)$ to randomly sample a set of size 4 from $\{9, \ldots, 16\}$

Our Standalone iPoSW: The Prover P



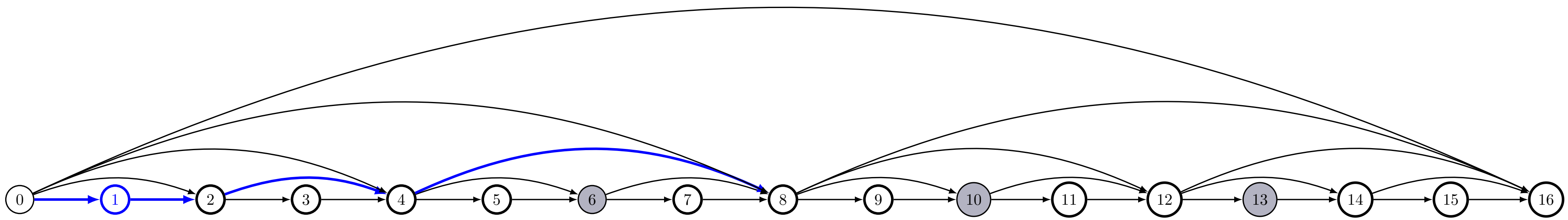
$$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\} \qquad \mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$$

$$\mathcal{L}_{16,2} = \{\pi_1, \pi_6, \pi_{10}, \pi_{13}\}$$

Computing $\mathcal{L}_{16,2}$ from $\mathcal{L}_{8,1}$ and $\mathcal{L}_{16,1}$:

$$\pi_i := L(j), L(\text{parents}(j)) \quad \forall j \in \text{path}(i) \text{ in } G_{[0:8]}, G_{[8:18]} \text{respt.}$$

Our Standalone iPoSW: The Prover P



$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\}$ $\mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$

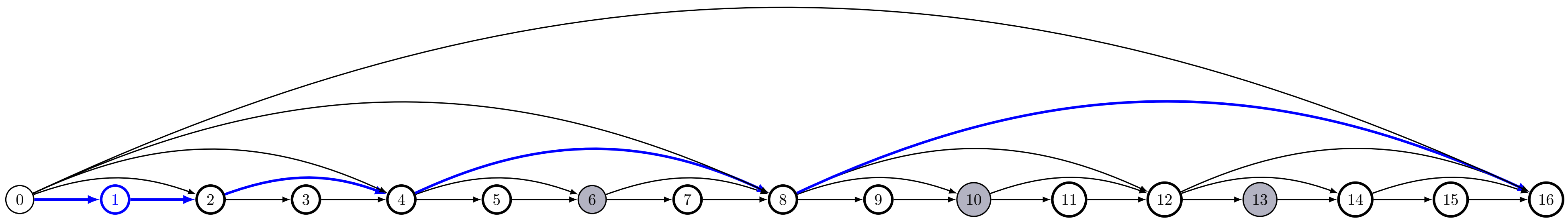
$\mathcal{L}_{16,2} = \{\pi_1, \pi_6, \pi_{10}, \pi_{13}\}$

Computing $\mathcal{L}_{16,2}$ from $\mathcal{L}_{8,1}$ and $\mathcal{L}_{16,1}$:

$$\pi_i := L(j), L(\text{parents}(j)) \quad \forall j \in \text{path}(i) \text{ in } G_{[0:8]}, G_{[8:18]} \text{respt.}$$

$$\underbrace{\pi_i \text{ in } G_{[0:8]}}_{\text{from } \mathcal{L}_{8,1}}$$

Our Standalone iPoSW: The Prover P



$$\underbrace{}_{\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\}} \quad \underbrace{}_{\mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}}$$

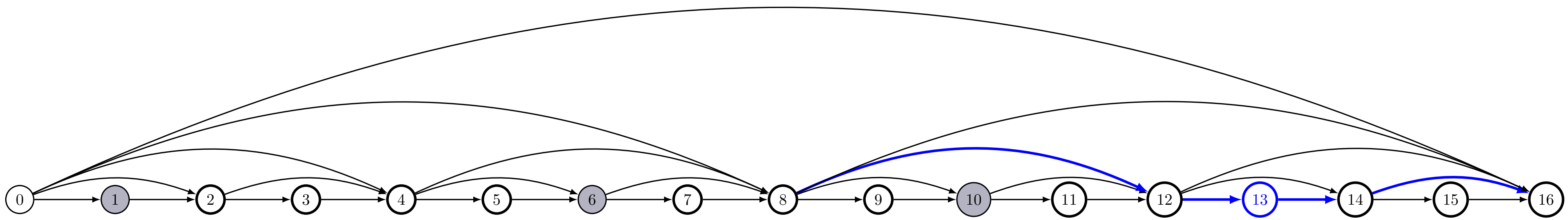
$$\mathcal{L}_{16,2} = \{\pi_1, \pi_6, \pi_{10}, \pi_{13}\}$$

Computing $\mathcal{L}_{16,2}$ from $\mathcal{L}_{8,1}$ and $\mathcal{L}_{16,1}$:

$$\pi_i := L(j), L(\text{parents}(j)) \quad \forall j \in \text{path}(i) \text{ in } G_{[0:8]}, G_{[8:18]} \text{respt.}$$

$$\underbrace{\pi_i \text{ in } G_{[0:8]}}_{\text{from } \mathcal{L}_{8,1}} + \underbrace{L(16), \text{parents}(16) \text{ in } G_{[8:16]}}_{\text{from } \mathcal{L}_{16,1}} \implies \pi_i \text{ in } G_{16}$$

Our Standalone iPoSW: The Prover P



$$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\} \qquad \mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$$

$$\mathcal{L}_{16,2} = \{\pi_1, \pi_6, \pi_{10}, \pi_{13}\}$$

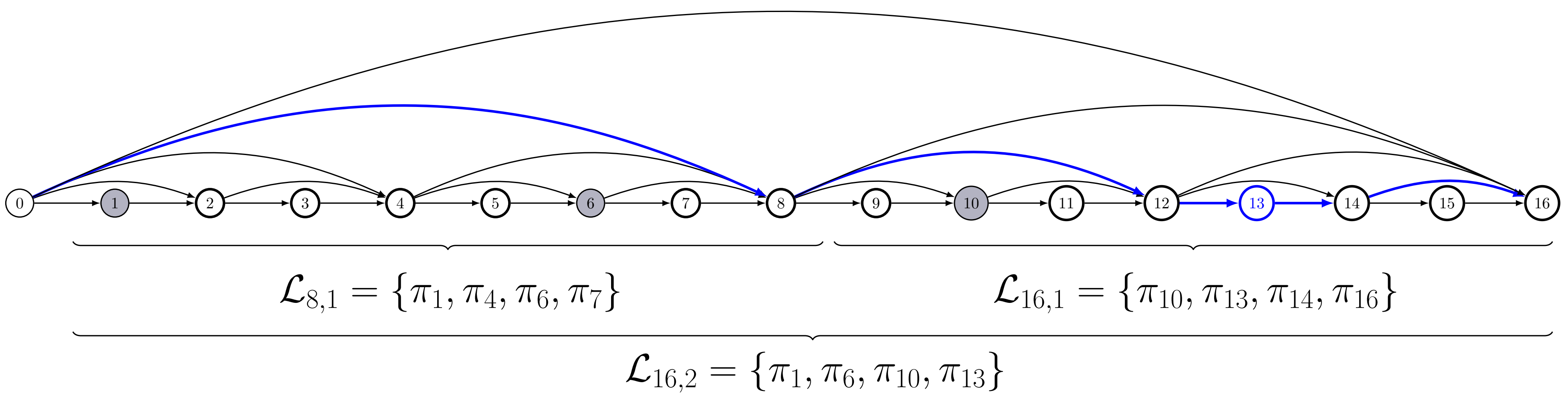
Computing $\mathcal{L}_{16,2}$ from $\mathcal{L}_{8,1}$ and $\mathcal{L}_{16,1}$:

$$\pi_i := L(j), L(\text{parents}(j)) \quad \forall j \in \text{path}(i) \text{ in } G_{[0:8]}, G_{[8:18]} \text{respt.}$$

$$\underbrace{\pi_i \text{ in } G_{[0:8]}}_{\text{from } \mathcal{L}_{8,1}} + \underbrace{L(16), \text{parents}(16) \text{ in } G_{[8:16]}}_{\text{from } \mathcal{L}_{16,1}} \implies \pi_i \text{ in } G_{16}$$

$$\underbrace{\pi_i \text{ in } G_{[8:16]}}_{\text{from } \mathcal{L}_{16,1}}$$

Our Standalone iPoSW: The Prover P



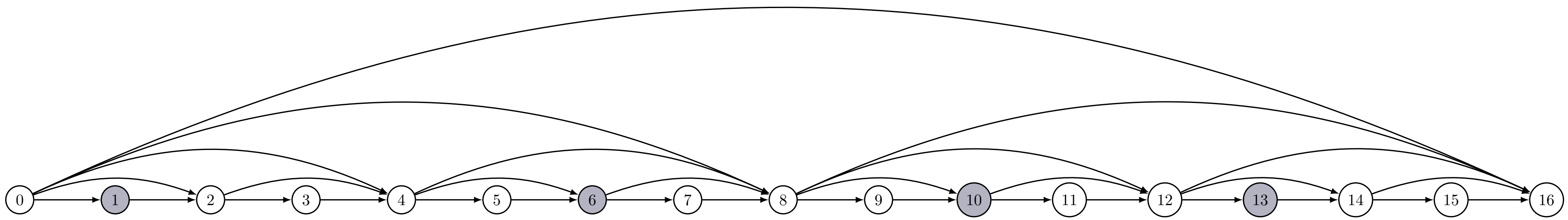
Computing $\mathcal{L}_{16,2}$ from $\mathcal{L}_{8,1}$ and $\mathcal{L}_{16,1}$:

$$\pi_i := L(j), L(\text{parents}(j)) \quad \forall j \in \text{path}(i) \text{ in } G_{[0:8]}, G_{[8:18]} \text{respt.}$$

$$\underbrace{\pi_i \text{ in } G_{[0:8]}}_{\text{from } \mathcal{L}_{8,1}} + \underbrace{L(16), \text{parents}(16) \text{ in } G_{[8:16]}}_{\text{from } \mathcal{L}_{16,1}} \implies \pi_i \text{ in } G_{16}$$

$$\underbrace{\pi_i \text{ in } G_{[8:16]}}_{\text{from } \mathcal{L}_{16,1}} + \underbrace{L(8), \text{parents}(8) \text{ in } G_{[0:8]}}_{\text{from } \mathcal{L}_{8,1}} \implies \pi_i \text{ in } G_{16}$$

Our Standalone iPoSW: The Inc Algorithm

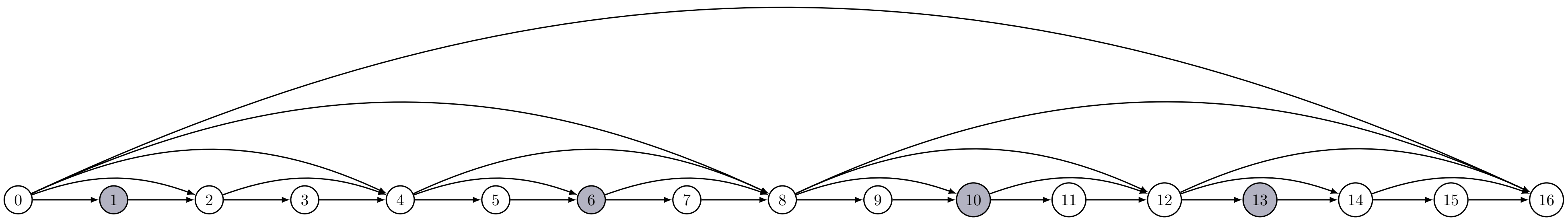


$$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\} \qquad \mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$$

$$\mathcal{L}_{16,2} = \{\pi_1, \pi_6, \pi_{10}, \pi_{13}\}$$

Inc **works exactly as** P: it picks up the computation where P leaves it and it constinues exactly as P would have continued

Standalone iPoSW Verifier V

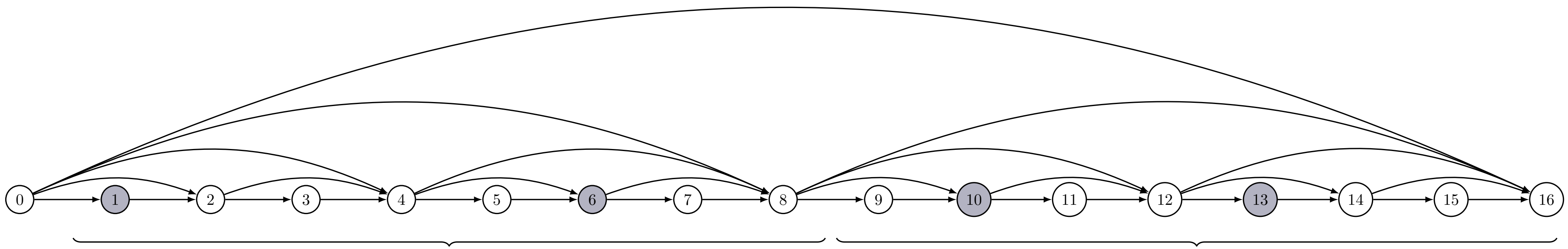


$$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\} \qquad \mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$$

$$\mathcal{L}_{16,2} = \{\pi_1, \pi_6, \pi_{10}, \pi_{13}\}$$

V recursively checks that challenges in are consistent with the sampling

Standalone iPoSW Verifier V



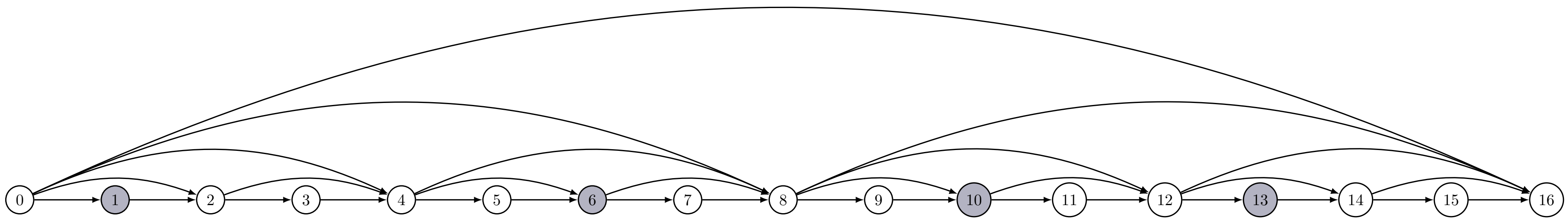
$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\}$ $\mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$

$\mathcal{L}_{16,2} = \{\pi_1, \pi_6, \pi_{10}, \pi_{13}\}$

V recursively checks that challenges in are consistent with the sampling

- P provides V for every π_i with an index set $\mathcal{I}_i$
- $\mathcal{I}_i = (i_1, \ldots, i_\ell) \in [t]^\ell$ where $\ell = \#$ of samplings in π_i
- i_j is associated with the jth sampling from sets $S_{0,j}$ and $S_{1,j}$
- $i \in S_{b,j} \quad \Rightarrow \quad i_j$ is the index within $S_{b,j}$

Standalone iPoSW Verifier V



$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\}$ $\mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$

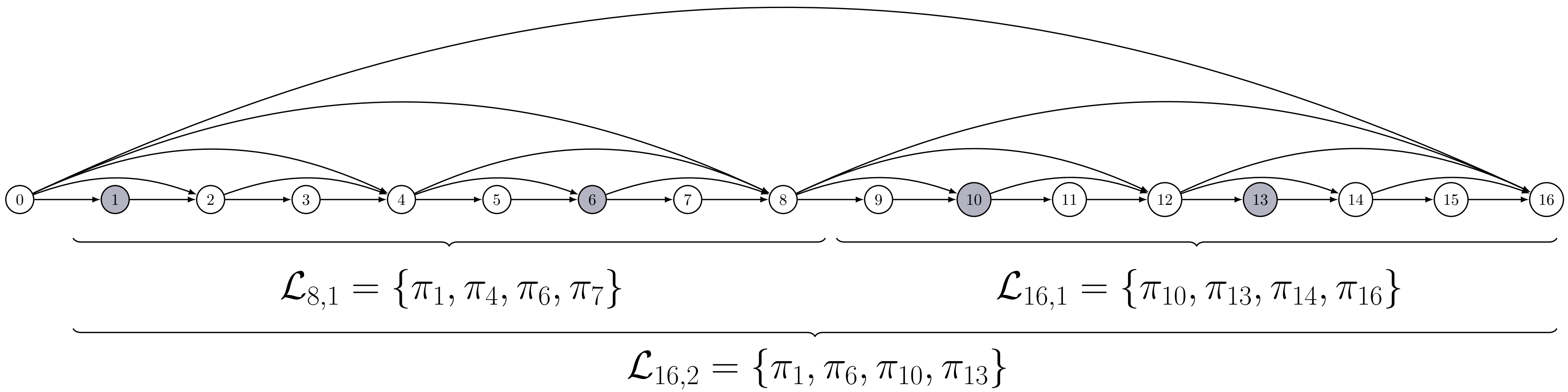
$\mathcal{L}_{16,2} = \{\pi_1, \pi_6, \pi_{10}, \pi_{13}\}$

V recursively checks that challenges in are consistent with the sampling

- P provides V for every π_i with an index set $\mathcal{I}_i$
- $\mathcal{I}_i = (i_1, \ldots, i_\ell) \in [t]^\ell$ where $\ell = \#$ of samplings in π_i
- i_j is associated with the jth sampling from sets $S_{0,j}$ and $S_{1,j}$
- $i \in S_{b,j} \quad \Rightarrow \quad i_j$ is the index within $S_{b,j}$

An important observation: the sampling sets are implictly given to V and are of size t

Standalone iPoSW Verifier V



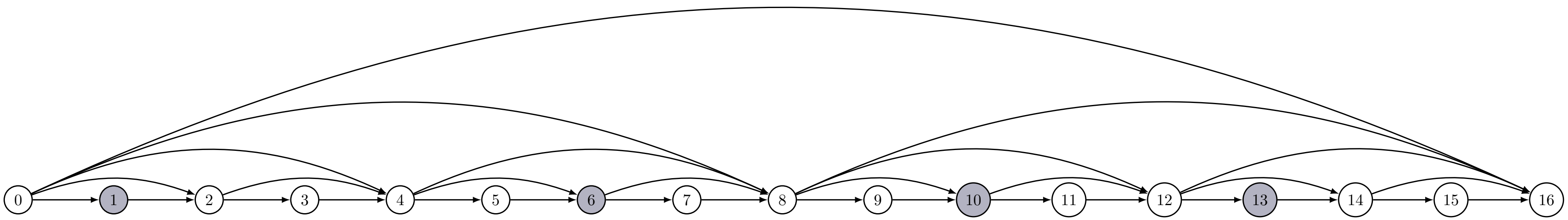
V recursively checks that challenges in are consistent with the sampling

- P provides V for every π_i with an index set $\mathcal{I}_i$
- $\mathcal{I}_i = (i_1, \ldots, i_\ell) \in [t]^\ell$ where $\ell = \#$ of samplings in π_i
- i_j is associated with the jth sampling from sets $S_{0,j}$ and $S_{1,j}$
- $i \in S_{b,j} \quad \Rightarrow \quad i_j$ is the index within $S_{b,j}$

An important observation: the sampling sets are implictly given to V and are of size t

For general weight distributions: the sampling sets are of size t **on expectation**

Soundness



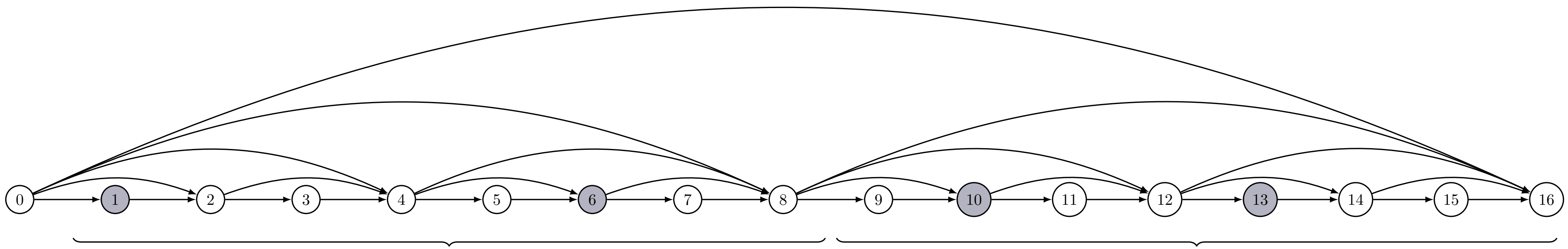
$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\}$ $\quad$ $\mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$

$\mathcal{L}_{16,2} = \{\pi_1, \pi_6, \pi_{10}, \pi_{13}\}$

Proof strategy:

1. Bound the advantage the on-the-fly sampling gives a malicous $\tilde{\mathsf{P}}$
2. Reduce security to the standalone PoSW

Soundness



$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\}$ $\qquad$ $\mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$

$\mathcal{L}_{16,2} = \{\pi_1, \pi_6, \pi_{10}, \pi_{13}\}$

Proof strategy:

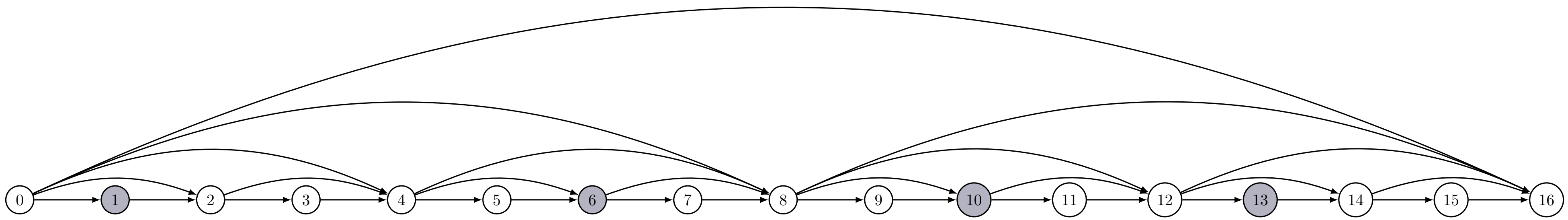
1. Bound the advantage the on-the-fly sampling gives a malicous $\tilde{\mathsf{P}}$
2. Reduce security to the standalone PoSW

The On-The-Fly Sampling Lemma:

1. $S := \mathcal{L}_{16,2}$ sampled from $S_0 := \mathcal{L}_{8,1} \cup S_1 := \mathcal{L}_{16,1}$ as above, or
2. S sampled directly from $\{1, \ldots, 16\}$

Show that the $\%$ of incosnsistent nodes in S in 1. and 2. are close

Soundness



$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\}$ $\quad$ $\mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$

$\mathcal{L}_{16,2} = \{\pi_1, \pi_6, \pi_{10}, \pi_{13}\}$

Proof strategy:

1. Bound the advantage the on-the-fly sampling gives a malicous $\tilde{\mathsf{P}}$
2. Reduce security to the standalone PoSW

The On-The-Fly Sampling Lemma:

1. $S := \mathcal{L}_{16,2}$ sampled from $S_0 := \mathcal{L}_{8,1} \cup S_1 := \mathcal{L}_{16,1}$ as above, or
2. S sampled directly from $\{1, \ldots, 16\}$

Show that the % of incosnsistent nodes in S in 1. and 2. are close

This follows easily from a Hoeffding bound

Security Statement

Thm: $(\mathsf{P}, \mathsf{V}, \mathsf{Inc})$ an (α, ϵ)-sound **iPoSW** for $\alpha \in (0, 1]$ and

$$\epsilon = \frac{1 + q^2}{2^\lambda} + \frac{q(q-1)}{2^{\lambda+1}} + q \cdot e^{-2t \cdot \left(\frac{1-\alpha}{\log n}\right)^2} \quad .$$

Security Statement

Thm: $(\mathsf{P}, \mathsf{V}, \mathsf{Inc})$ an (α, ϵ)-sound **iPoSW** for $\alpha \in (0, 1]$ and

$$\epsilon = \frac{1+q^2}{2^\lambda} + \frac{q(q-1)}{2^{\lambda+1}} + q \cdot e^{-2t \cdot \left(\frac{1-\alpha}{\log n}\right)^2} .$$

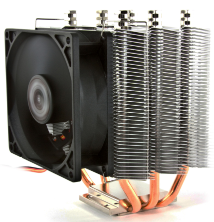
Thm (Standalone PoSW): (P, V) is an (α, ϵ)-sound **PoSW** for $\alpha \in (0, 1]$ and

$$\epsilon = \frac{3 \cdot q^2}{2^\lambda} + \alpha^t .$$

General Weight/Challenge Distributions

Parameters: $G_n, t, \cdots$

$\mathsf{P}^{\tau(\cdot)}$ $\qquad$ $\mathsf{V}^{\tau(\cdot)}$



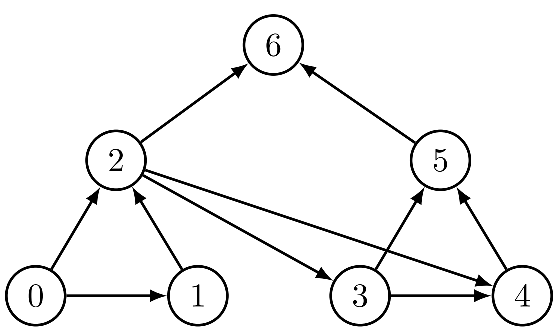
$\xleftarrow{\quad \chi \quad}$

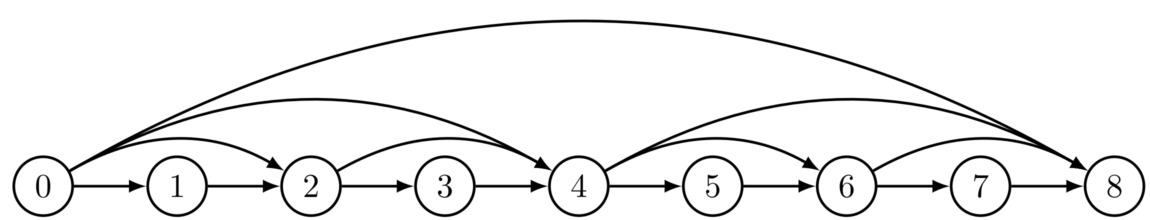
$\xrightarrow{\quad L(n) \quad}$

$\xleftarrow{\quad S \xleftarrow{\$} \mathcal{C} \subseteq \{0, \ldots, n\} \quad}$

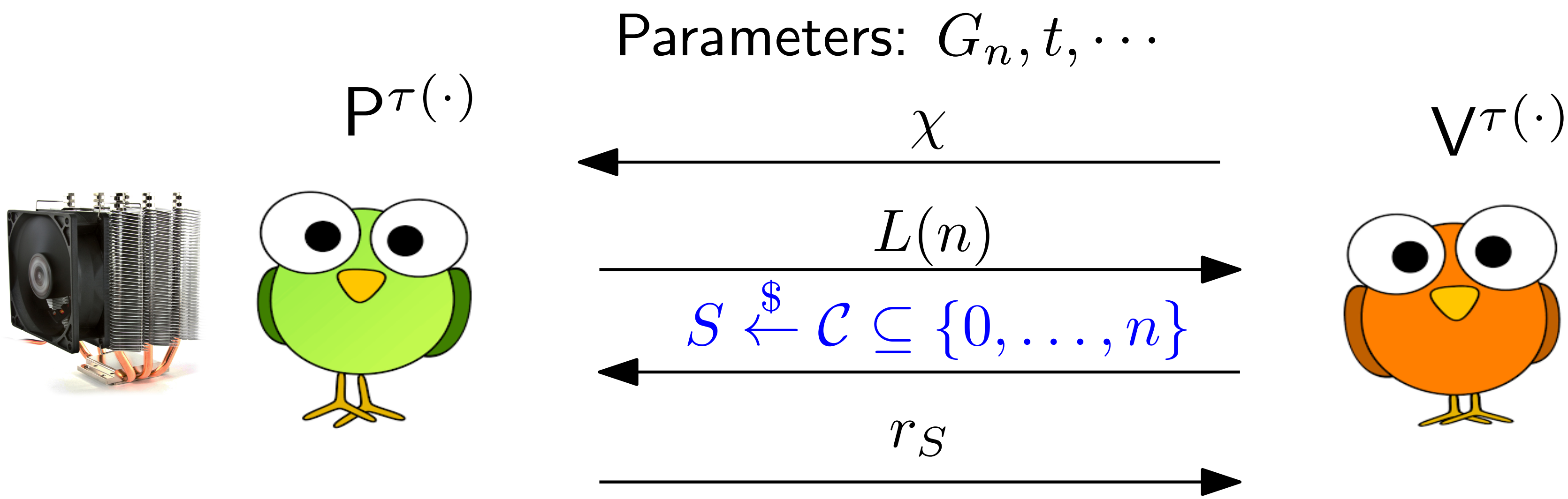
$\xrightarrow{\quad r_S \quad}$

Standalone (i)PoSW: Sample a random S of size t from $\mathcal{C}$



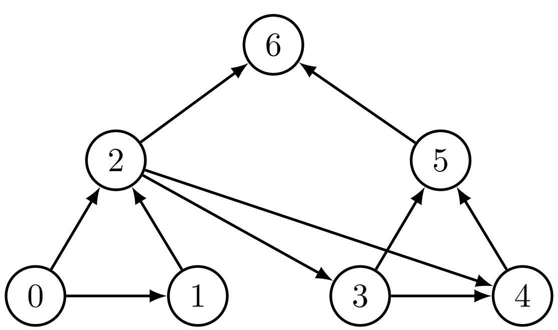


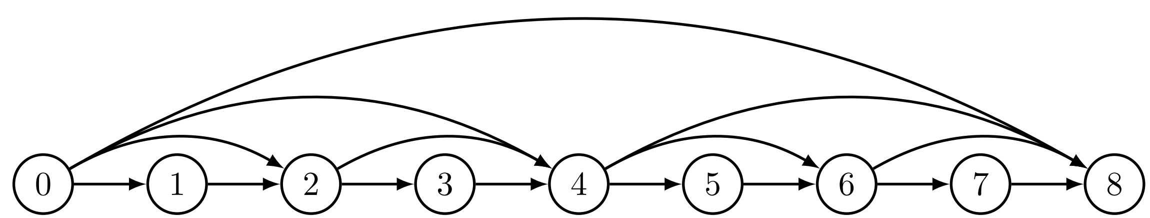
General Weight/Challenge Distributions



Standalone (i)PoSW: Sample a random S of size t from $\mathcal{C}$

For some applications, not all challenges in $\mathcal{C}$ are treated equally



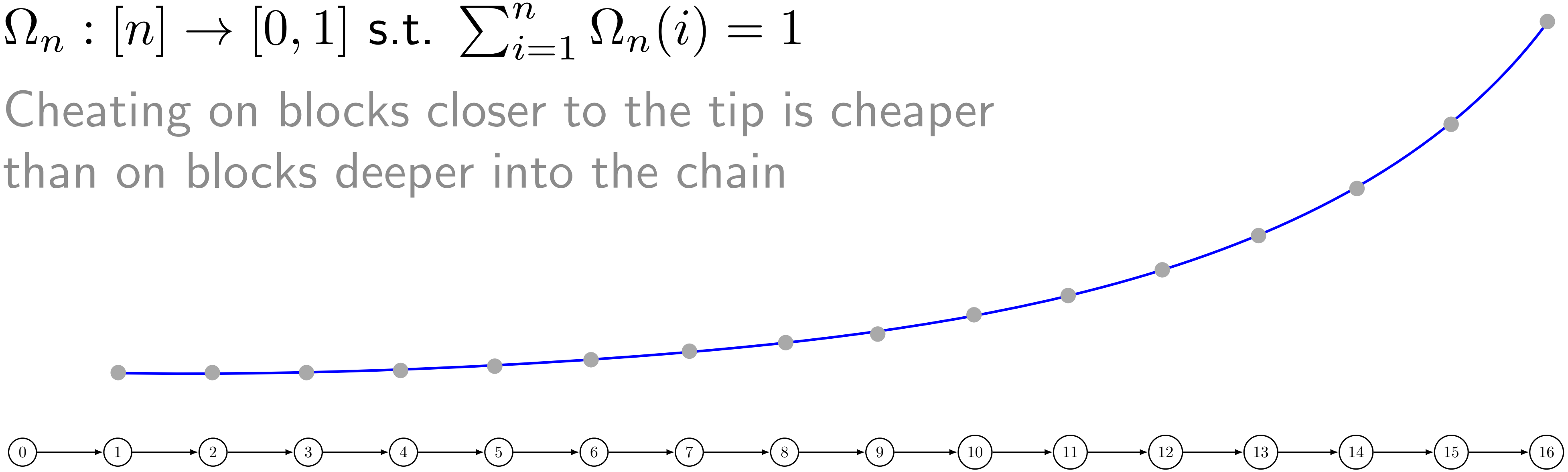


The SNACK Weight Distribution

The SNACK weight distribution $\Omega_n(i) \sim \frac{1}{n-i+c}$

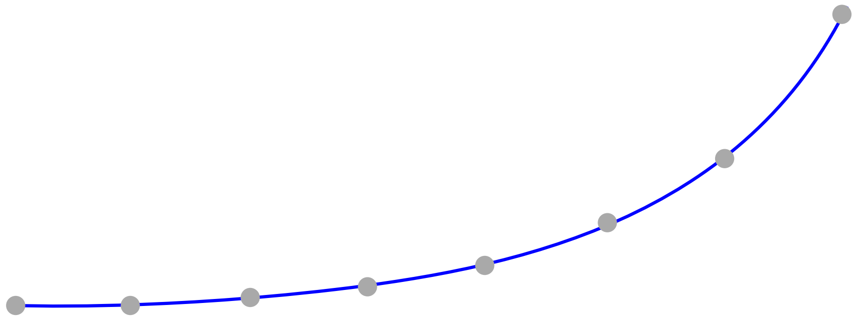
$\Omega_n : [n] \to [0,1]$ s.t. $\sum_{i=1}^{n} \Omega_n(i) = 1$

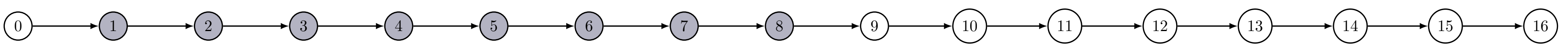
Cheating on blocks closer to the tip is cheaper than on blocks deeper into the chain



The SNACK Weight Distribution Is Incrementable

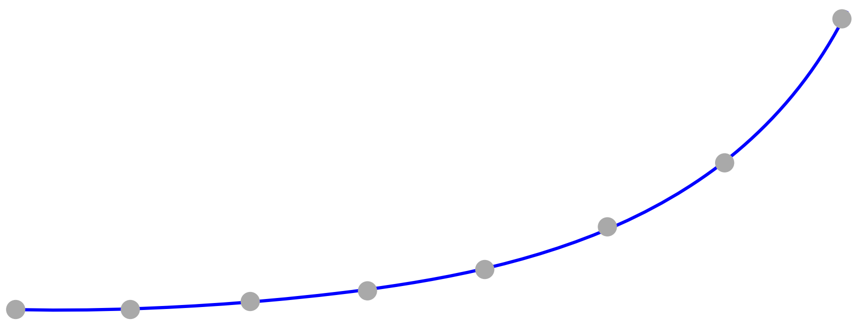
The SNACK weight distribution $\Omega_n(i) \sim \frac{1}{n-i+c}$

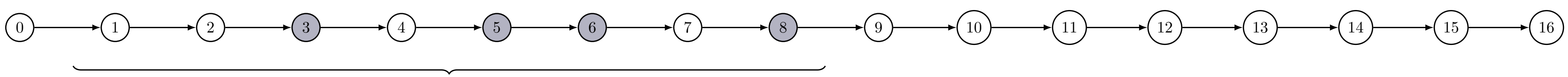




The SNACK Weight Distribution Is Incrementable

The SNACK weight distribution $\Omega_n(i) \sim \frac{1}{n-i+c}$



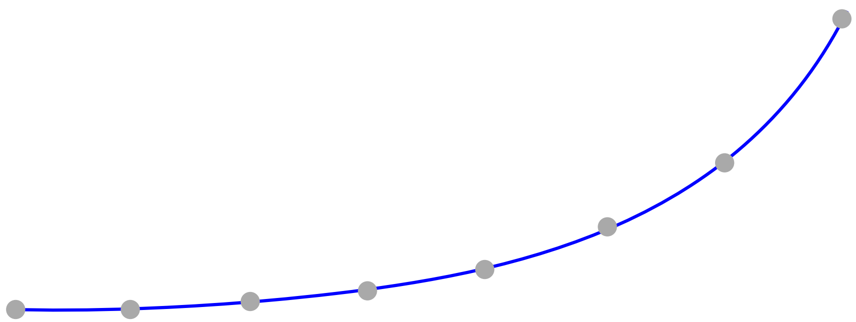


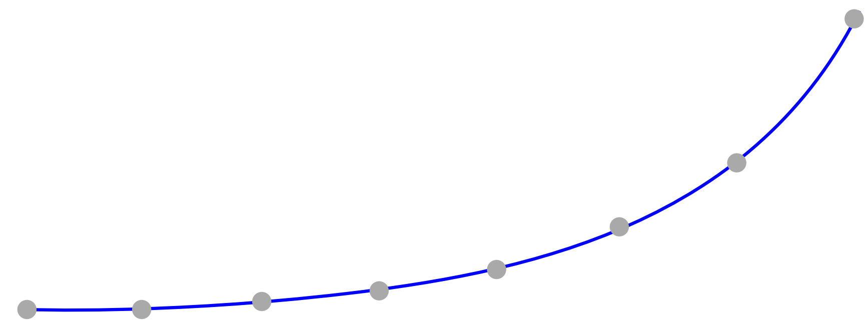
$\mathcal{L}_{8,1} = \{\pi_3, \pi_5, \pi_6, \pi_8\}$

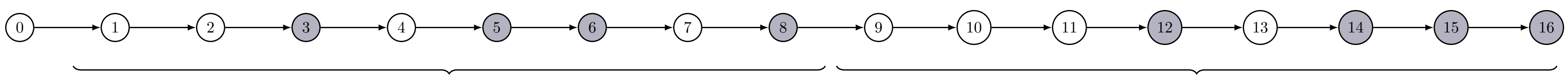
Add π_i to $\mathcal{L}_{8,1}$ w.p. $t \cdot \Omega_n(i)$ $\Rightarrow$ $|\mathcal{L}_{8,1}| = t$ on expectation

The SNACK Weight Distribution Is Incrementable

The SNACK weight distribution $\Omega_n(i) \sim \frac{1}{n-i+c}$





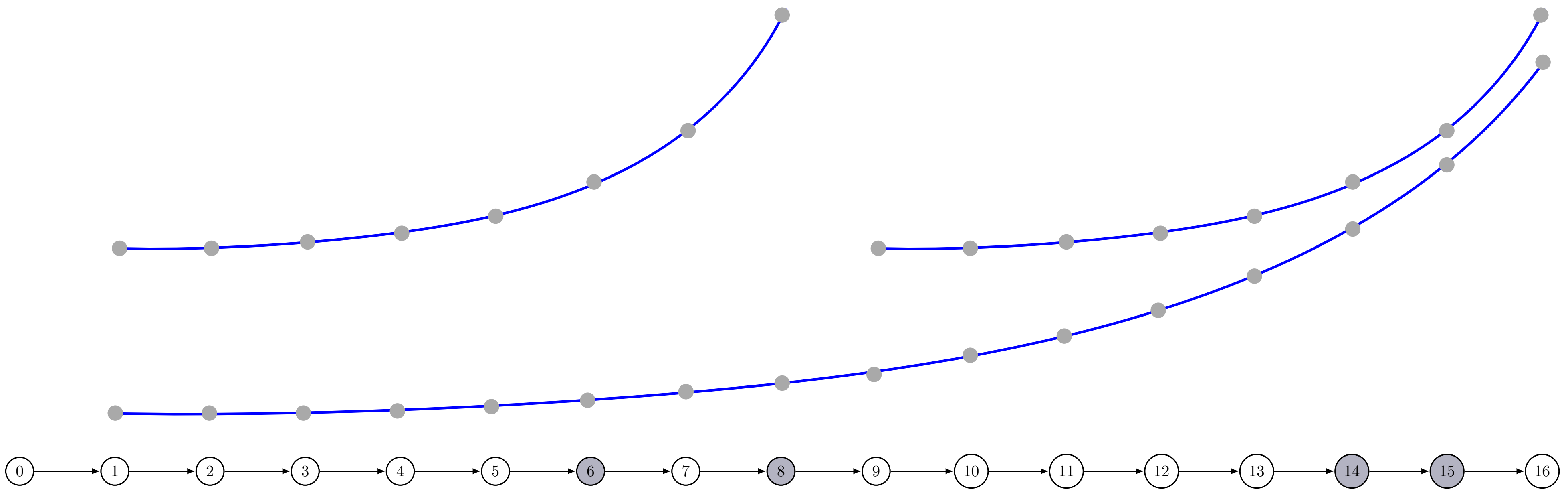


$\mathcal{L}_{8,1} = \{\pi_3, \pi_5, \pi_6, \pi_8\}$ $\qquad$ $\mathcal{L}_{16,1} = \{\pi_{12}, \pi_{14}, \pi_{15}, \pi_{16}\}$

Add π_i to $\mathcal{L}_{16,1}$ w.p. $t \cdot \Omega_n(i-8)$ $\quad \Rightarrow \quad$ $|\mathcal{L}_{16,1}| = t$ on expectation

The SNACK Weight Distribution Is Incrementable

The SNACK weight distribution $\Omega_n(i) \sim \frac{1}{n-i+c}$



$\mathcal{L}_{8,1} = \{\pi_3, \pi_5, \pi_6, \pi_8\}$ $\mathcal{L}_{16,1} = \{\pi_{12}, \pi_{14}, \pi_{15}, \pi_{16}\}$

$\mathcal{L}_{16,2} = \{\pi_6, \pi_8, \pi_{14}, \pi_{15}\}$

Add π_i to $\mathcal{L}_{16,2}$ w.p. $t \cdot \Omega_n(i)$ $\Rightarrow$ $|\mathcal{L}_{16,2}| = t$ on expectation

Our iPoSW for General Weight Distributions

We characterize distributions that can be sampled incrementally:
t-**Incrementally Sampleable Distributions**

Our iPoSW for General Weight Distributions

We characterize distributions that can be sampled incrementally:
t-**Incrementally Sampleable Distributions**

Give a constrution for any t-incrementally sampleable distribution over the skiplist graph

Our iPoSW for General Weight Distributions

We characterize distributions that can be sampled incrementally:
t-**Incrementally Sampleable Distributions**

Give a constrution for any t-incrementally sampleable distribution over the skiplist graph

New challenges to resolve:

- In the standalone case: the sampling sets are implicitly defined and are of size t exactly

Our iPoSW for General Weight Distributions

We characterize distributions that can be sampled incrementally:
t-**Incrementally Sampleable Distributions**

Give a constrution for any t-incrementally sampleable distribution over the skiplist graph

New challenges to resolve:

- In the standalone case: the sampling sets are implicitly defined and are of size t exactly

- In the general case: the sampling sets are no longer implicitly defined and are of size t **on expectation**

Our iPoSW for General Weight Distributions

We characterize distributions that can be sampled incrementally:
t-**Incrementally Sampleable Distributions**

Give a constrution for any t-incrementally sampleable distribution over the skiplist graph

New challenges to resolve:

- In the standalone case: the sampling sets are implicitly defined and are of size t exactly
- In the general case: the sampling sets are no longer implicitly defined and are of size t **on expectation**

Solution:

- P commits to the sampling sets in a tree fashion

Our iPoSW for General Weight Distributions

We characterize distributions that can be sampled incrementally:
t-**Incrementally Sampleable Distributions**

Give a constrution for any t-incrementally sampleable distribution over the skiplist graph

New challenges to resolve:

- In the standalone case: the sampling sets are implicitly defined and are of size t exactly
- In the general case: the sampling sets are no longer implicitly defined and are of size t **on expectation**

Solution:

- P commits to the sampling sets in a tree fashion
- P now provides the sampling sets explicitly as part of the proof

Our iPoSW for General Weight Distributions

We characterize distributions that can be sampled incrementally:
t-**Incrementally Sampleable Distributions**

Give a constrution for any t-incrementally sampleable distribution over the skiplist graph

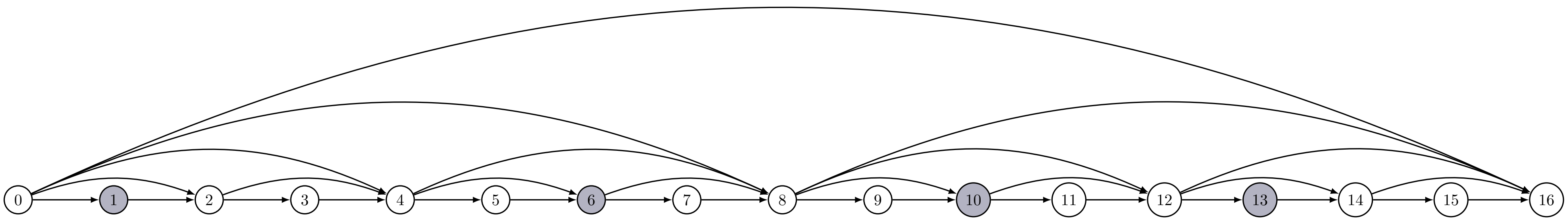
New challenges to resolve:

- In the standalone case: the sampling sets are implicitly defined and are of size t exactly
- In the general case: the sampling sets are no longer implicitly defined and are of size t **on expectation**

Solution:

- P commits to the sampling sets in a tree fashion
- P now provides the sampling sets explicitly as part of the proof
- V recursively checks the consistency of the samplings as before and that these sets are within their expected size

Soundness



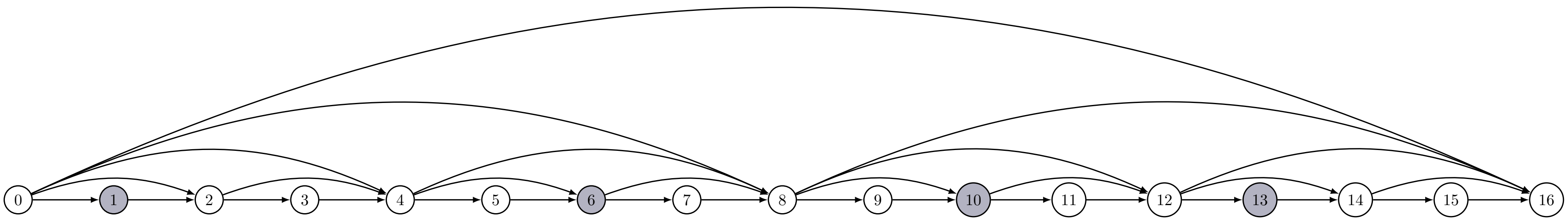
$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\}$ $\mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$

$\mathcal{L}_{16,2} = \{\pi_1, \pi_6, \pi_{10}, \pi_{13}\}$

Proof strategy:

1. Bound the advantage $\tilde{\mathsf{P}}$ get from choosing malicious sampling sets
2. Generalize the on-the-fly sampling bound to t-incrementally sampleable distributions
3. Reduce security to the standalone PoSW

Soundness



$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\}$ $\qquad$ $\mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$

$\mathcal{L}_{16,2} = \{\pi_1, \pi_6, \pi_{10}, \pi_{13}\}$

Proof strategy:

1. Bound the advantage $\tilde{\mathsf{P}}$ get from choosing malicious sampling sets
2. Generalize the on-the-fly sampling bound to t-incrementally sampleable distributions
3. Reduce security to the standalone PoSW

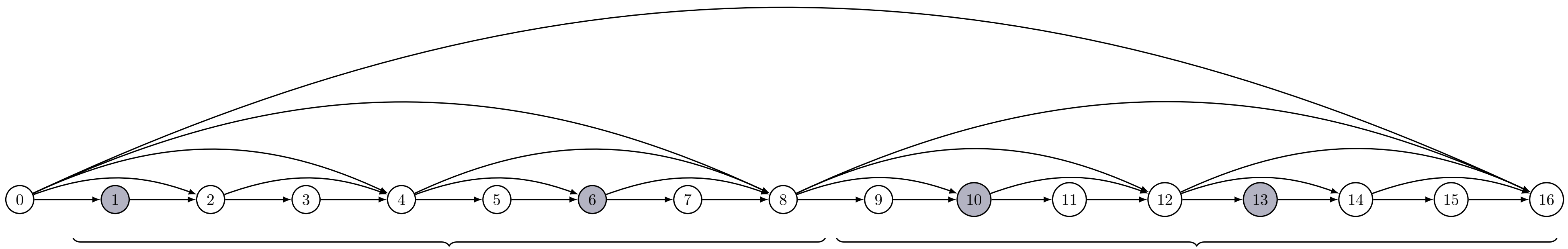
We give bounds for any t-incrementally sampleable weight distributions

We give concrete bounds for the uniform and SNACK distributions

Thank you

Additional Material

Efficiency Measures



$$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\} \qquad \mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$$

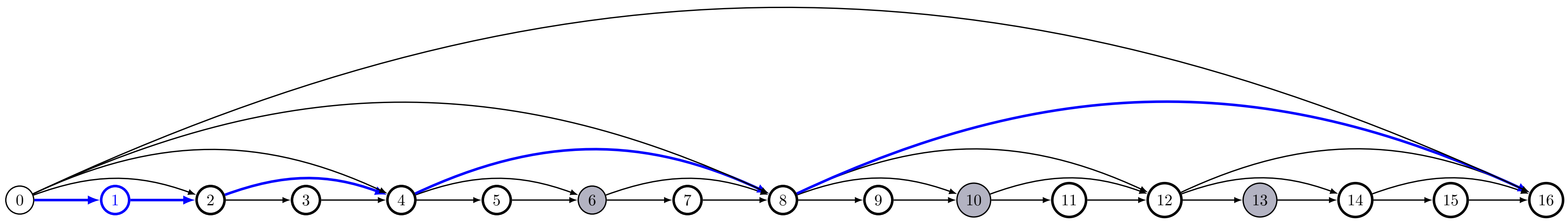
$$\mathcal{L}_{16,2} = \{\pi_1, \pi_6, \pi_{10}, \pi_{13}\}$$

Prover space complexity:

1. The skiplist G_n can be topologically labeled with space $(\log n + 1)\lambda$ bits
2. At no time P or Inc keeps more than $\log n + 1$ lists $\mathcal{L}_{v,i}$, each of succinct size

$\Rightarrow$ proof size: $O(\lambda \cdot t \cdot \log^3 n)$

Our Standalone iPoSW: The Verifier V



$$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\} \qquad \mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$$

$$\mathcal{L}_{16,2} = \{\pi_1, \pi_6, \pi_{10}, \pi_{13}\}$$

$\mathcal{I}_1 = \{1, 1\}$

$$S_{0,1} = \{1,2,3,4\} \cup S_{1,1} = \{5,6,7,8\}$$

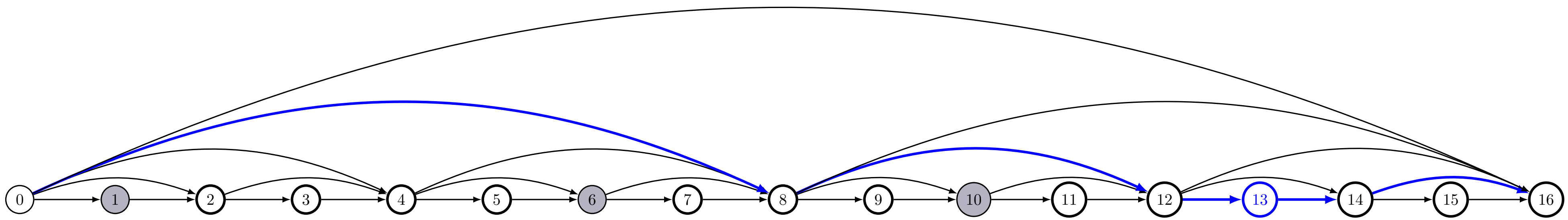
$$\rightarrow_{L(8)} S_1 = \{1,4,6,7\}$$

$$S_{0,2} = \{1,4,6,7\} \cup S_{1,2} = \{10,13,14,16\}$$

$$\rightarrow_{L(16)} S_2 = \{1,6,10,13\}$$

Note: $L(8), L(16) \in \pi_1$

Our Standalone iPoSW: The Verifier V



$$\mathcal{L}_{8,1} = \{\pi_1, \pi_4, \pi_6, \pi_7\} \qquad \mathcal{L}_{16,1} = \{\pi_{10}, \pi_{13}, \pi_{14}, \pi_{16}\}$$

$$\mathcal{L}_{16,2} = \{\pi_1, \pi_6, \pi_{10}, \pi_{13}\}$$

$\mathcal{I}_{13} = \{1, 2\}$

$$S_{0,1} = \{9, 10, 11, 12\} \cup S_{1,1} = \{13, 14, 15, 16\}$$

$$\to_{L(16)} S_1 = \{10, 13, 14, 16\}$$

$$S_{0,2} = \{1, 4, 6, 7\} \cup S_{1,2} = \{10, 13, 14, 16\}$$

$$\to_{L(16)} S_2 = \{1, 6, 10, 13\}$$

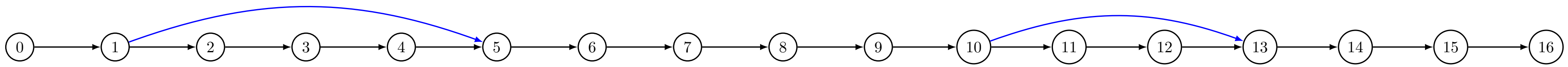
Note: $L(16) \in \pi_{13}$

Succinct Non-interactive Arguments of Chain Knowledge

[Abusalah-Fuchsbauer-Gaži-Klein'22]

Weighted Blockchain: $\Gamma_n = (H_n, \Omega_n)$ with weight function

$$\Omega_n : [n] \to [0,1] \text{ s.t. } \sum_{i=1}^{n} \Omega_n(i) = 1$$

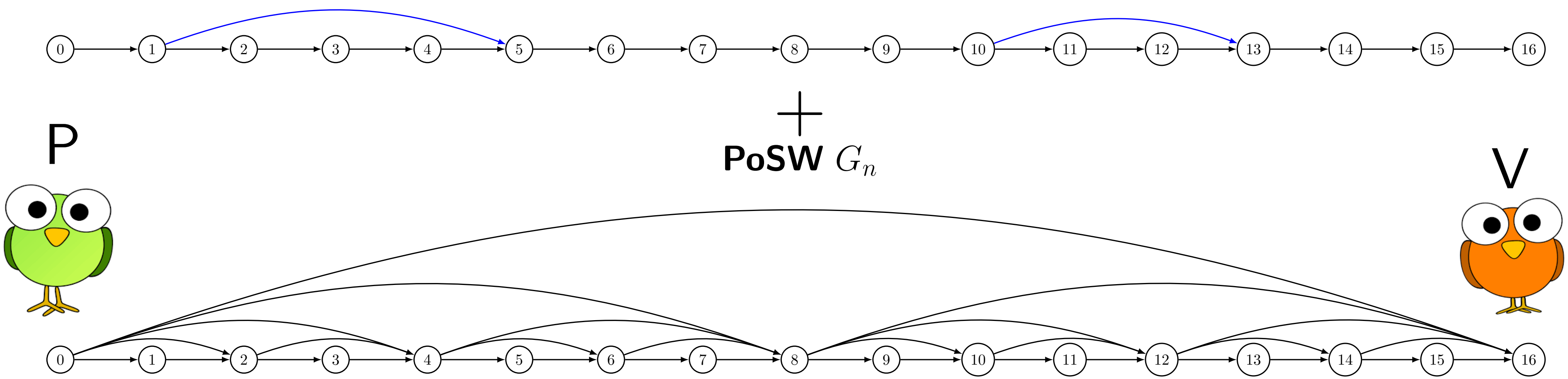


Succinct Non-interactive Arguments of Chain Knowledge

[Abusalah-Fuchsbauer-Gaži-Klein'22]

Weighted Blockchain: $\Gamma_n = (H_n, \Omega_n)$ with weight function

$$\Omega_n : [n] \to [0,1] \text{ s.t. } \sum_{i=1}^{n} \Omega_n(i) = 1$$

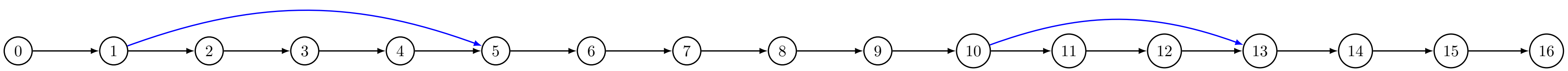


Succinct Non-interactive Arguments of Chain Knowledge

[Abusalah-Fuchsbauer-Gaži-Klein'22]

Weighted Blockchain: $\Gamma_n = (H_n, \Omega_n)$ with weight function

$$\Omega_n : [n] \to [0,1] \text{ s.t. } \sum_{i=1}^{n} \Omega_n(i) = 1$$

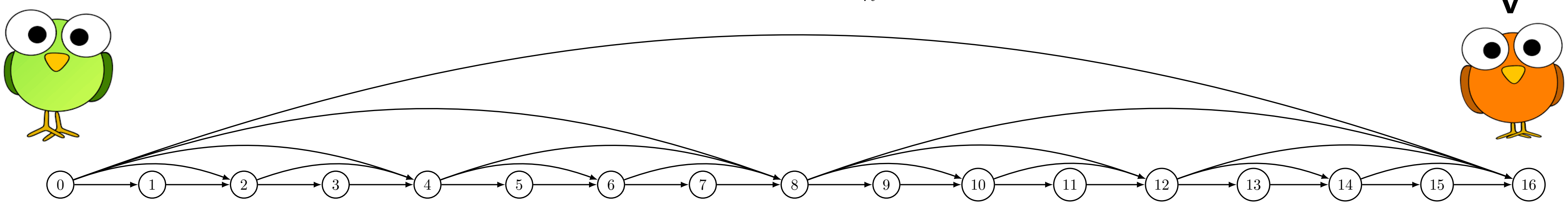


$+$

P

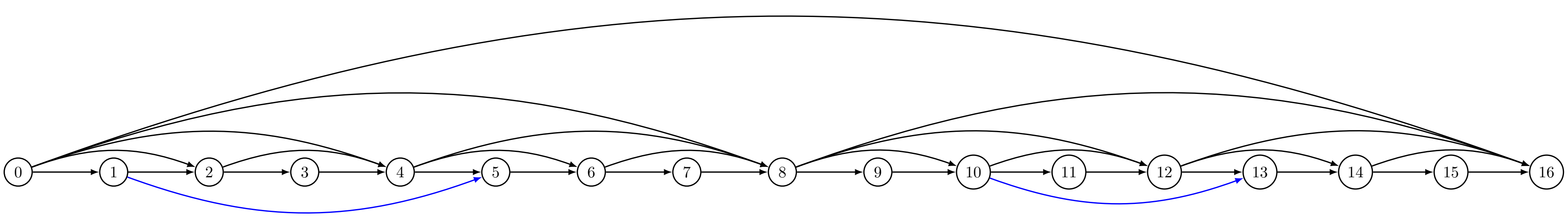
PoSW G_n

V



$=$

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

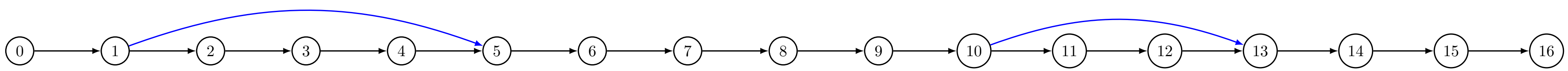


Succinct Non-interactive Arguments of Chain Knowledge

[Abusalah-Fuchsbauer-Gaži-Klein'22]

Weighted Blockchain: $\Gamma_n = (H_n, \Omega_n)$ with weight function

$$\Omega_n : [n] \to [0,1] \text{ s.t. } \sum_{i=1}^{n} \Omega_n(i) = 1$$

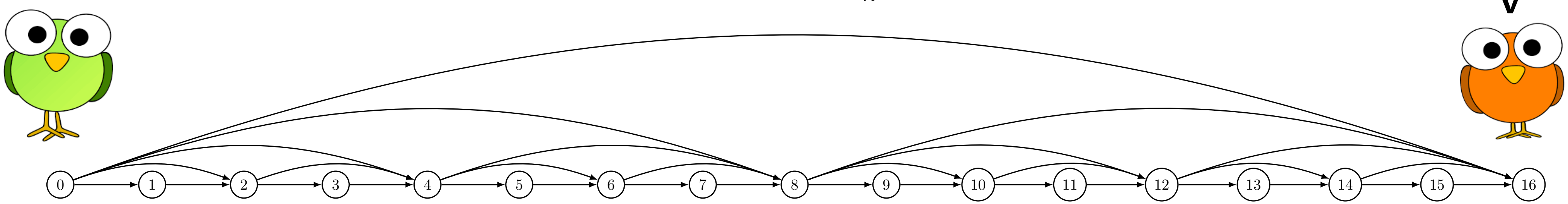


$+$

PoSW G_n

P

V



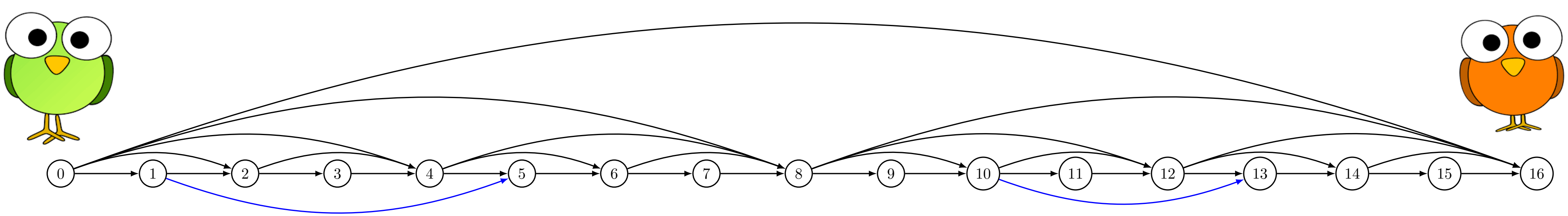
$=$

$\wr$ $\wr$

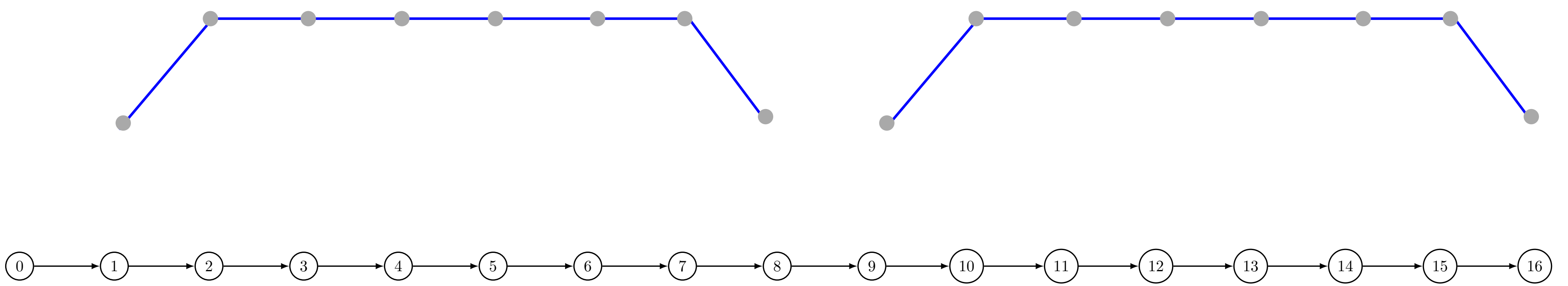
Augmented DAG $(K := ([n]_0, E_K := E_H \cup E_G), \Omega_n)$

P′

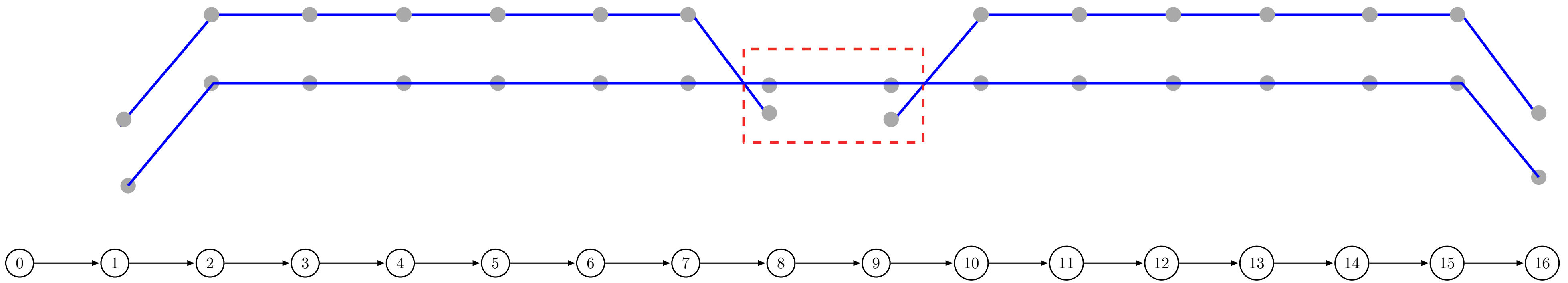
V′



Not All Distributions Are Incrementable



Not All Distributions Are Incrementable



We characterize distributions that can be sampled incrementally:
t-incrementally Sampleable Distributions

Give a constrution for any t-sampleable distribution over the skiplist graph

Malicious Sampling Sets

$\tilde{\mathsf{P}}$ can maliciously choose the sampling sets

Malicious Sampling Sets

$\tilde{\mathsf{P}}$ can maliciously choose the sampling sets

Let S_0 and S_1 be the sampling sets from which S is sampled at level k

Malicious Sampling Sets

$\tilde{\mathsf{P}}$ can maliciously choose the sampling sets

Let S_0 and S_1 be the sampling sets from which S is sampled at level k

Assume $i \in S_0$ is inconsistent and that $i \in S$

Malicious Sampling Sets

$\tilde{\mathsf{P}}$ can maliciously choose the sampling sets

Let S_0 and S_1 be the sampling sets from which S is sampled at level k

Assume $i \in S_0$ is inconsistent and that $i \in S$

$\tilde{\mathsf{P}}$ malicoulsy manipulates S_0 into $\tilde{S}_0$ s.t. $i \notin S$ where S is sampled from $\tilde{S}_0 \cup S_1$

Malicious Sampling Sets

$\tilde{\mathsf{P}}$ can maliciously choose the sampling sets

Let S_0 and S_1 be the sampling sets from which S is sampled at level k

Assume $i \in S_0$ is inconsistent and that $i \in S$

$\tilde{\mathsf{P}}$ malicoulsy manipulates S_0 into $\tilde{S}_0$ s.t. $i \notin S$ where S is sampled from $\tilde{S}_0 \cup S_1$

S contains some $j \in \tilde{S}_0$

Malicious Sampling Sets

$\tilde{\mathsf{P}}$ can maliciously choose the sampling sets

Let S_0 and S_1 be the sampling sets from which S is sampled at level k

Assume $i \in S_0$ is inconsistent and that $i \in S$

$\tilde{\mathsf{P}}$ malicoulsy manipulates S_0 into $\tilde{S}_0$ s.t. $i \notin S$ where S is sampled from $\tilde{S}_0 \cup S_1$

S contains some $j \in \tilde{S}_0$

V catches $\tilde{\mathsf{P}}$ in level $k-1$ unless
$\tilde{P}$ breaks the commitment

Malicious Sampling Sets

$\tilde{\mathsf{P}}$ can maliciously choose the sampling sets

Let S_0 and S_1 be the sampling sets from which S is sampled at level k

Assume $i \in S_0$ is inconsistent and that $i \in S$

$\tilde{\mathsf{P}}$ malicoulsy manipulates S_0 into $\tilde{S}_0$ s.t. $i \notin S$ where S is sampled from $\tilde{S}_0 \cup S_1$

S contains some $j \in \tilde{S}_0$

S doesn't contain any $j \in \tilde{S}_0$

V catches $\tilde{\mathsf{P}}$ in level $k-1$ unless
$\tilde{P}$ breaks the commitment

Malicious Sampling Sets

$\tilde{\mathsf{P}}$ can maliciously choose the sampling sets

Let S_0 and S_1 be the sampling sets from which S is sampled at level k

Assume $i \in S_0$ is inconsistent and that $i \in S$

$\tilde{\mathsf{P}}$ malicoulsy manipulates S_0 into $\tilde{S}_0$ s.t. $i \notin S$ where S is sampled from $\tilde{S}_0 \cup S_1$

S contains some $j \in \tilde{S}_0$

V catches $\tilde{\mathsf{P}}$ in level $k-1$ unless
$\tilde{P}$ breaks the commitment

S doesn't contain any $j \in \tilde{S}_0$

$\tilde{\mathsf{P}}$ gains advantage

Malicious Sampling Sets

$\tilde{\mathsf{P}}$ can maliciously choose the sampling sets

Let S_0 and S_1 be the sampling sets from which S is sampled at level k

Assume $i \in S_0$ is inconsistent and that $i \in S$

$\tilde{\mathsf{P}}$ malicoulsy manipulates S_0 into $\tilde{S}_0$ s.t. $i \notin S$ where S is sampled from $\tilde{S}_0 \cup S_1$

S contains some $j \in \tilde{S}_0$

V catches $\tilde{\mathsf{P}}$ in level $k-1$ unless
$\tilde{P}$ breaks the commitment

S doesn't contain any $j \in \tilde{S}_0$

$\tilde{\mathsf{P}}$ gains advantage

We bound the advantage of $\tilde{\mathsf{P}}$ in these cases and give concrete bounds for the uniform and SNACK distributions

Security Statement

Thm (iPoSW for the uniform weight distribution): $(\mathsf{P}, \mathsf{V}, \mathsf{Inc})$ an (α, ϵ)-sound **iPoSW** for $\alpha \in (0, 1]$ and

$$\epsilon = \frac{1+q^2}{2^\lambda} + \frac{q(q-1)}{2^{\lambda+1}} + q \cdot e^{-2t \cdot \left(\frac{1-\alpha}{\log n}\right)^2} + q \cdot 2^{-\zeta \cdot t} \quad .$$

Thm (Standalone iPoSW): $(\mathsf{P}, \mathsf{V}, \mathsf{Inc})$ an (α, ϵ)-sound **iPoSW** for $\alpha \in (0, 1]$ and

$$\epsilon = \frac{1+q^2}{2^\lambda} + \frac{q(q-1)}{2^{\lambda+1}} + q \cdot e^{-2t \cdot \left(\frac{1-\alpha}{\log n}\right)^2} \quad .$$

Thm (Standalone PoSW): (P, V) is an (α, ϵ)-sound **PoSW** for $\alpha \in (0, 1]$ and

$$\epsilon = \frac{3 \cdot q^2}{2^\lambda} + \alpha^t \quad .$$