

SublonK: Sublinear Prover PlonK

PETS 2024, Bristol



Arka Rai Choudhuri
NTT Research



Sanjam Garg
UC Berkeley



Aarushi Goel
NTT Research



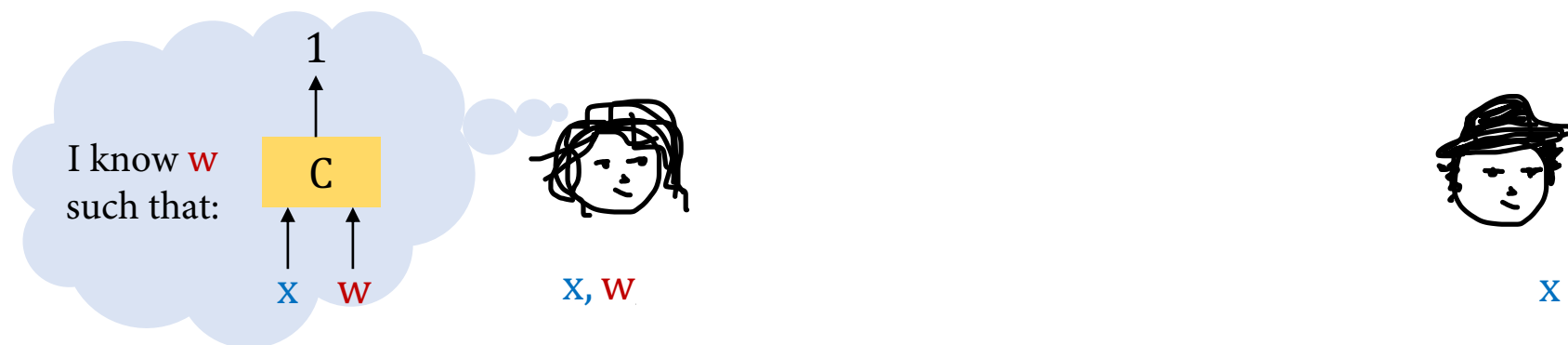
Sruthi Sekar
IIT Bombay



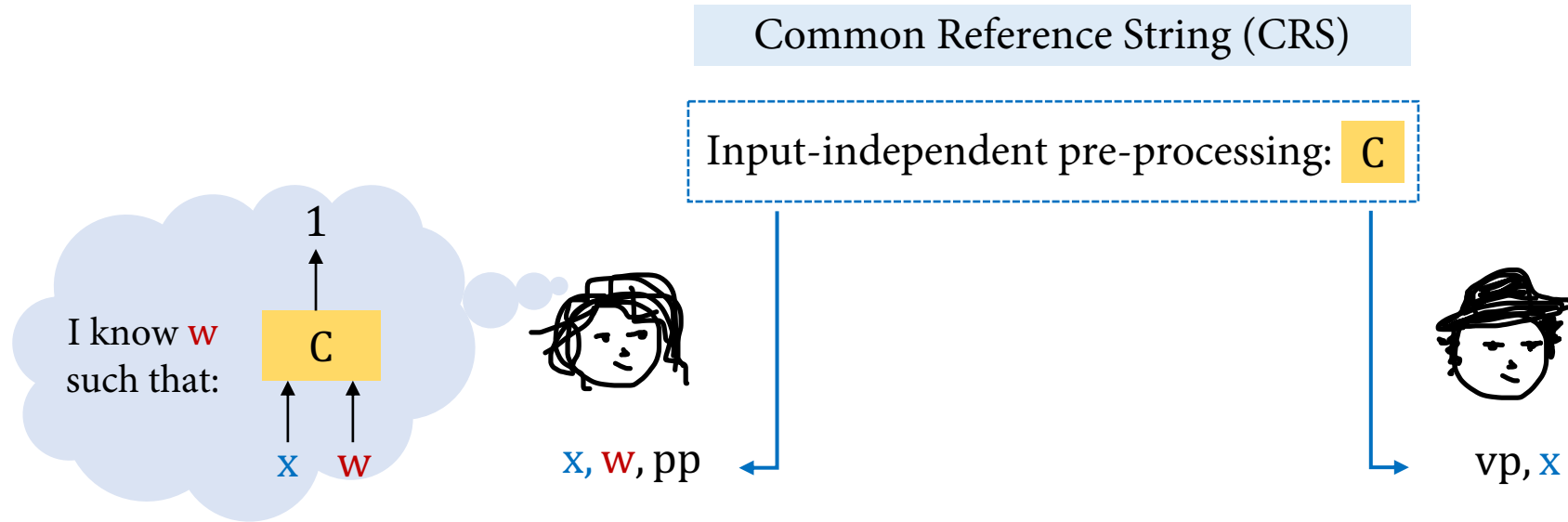
Rohit Sinha
Swirlds Labs

SNARKs: Succinct Non-interactive Arguments of Knowledge

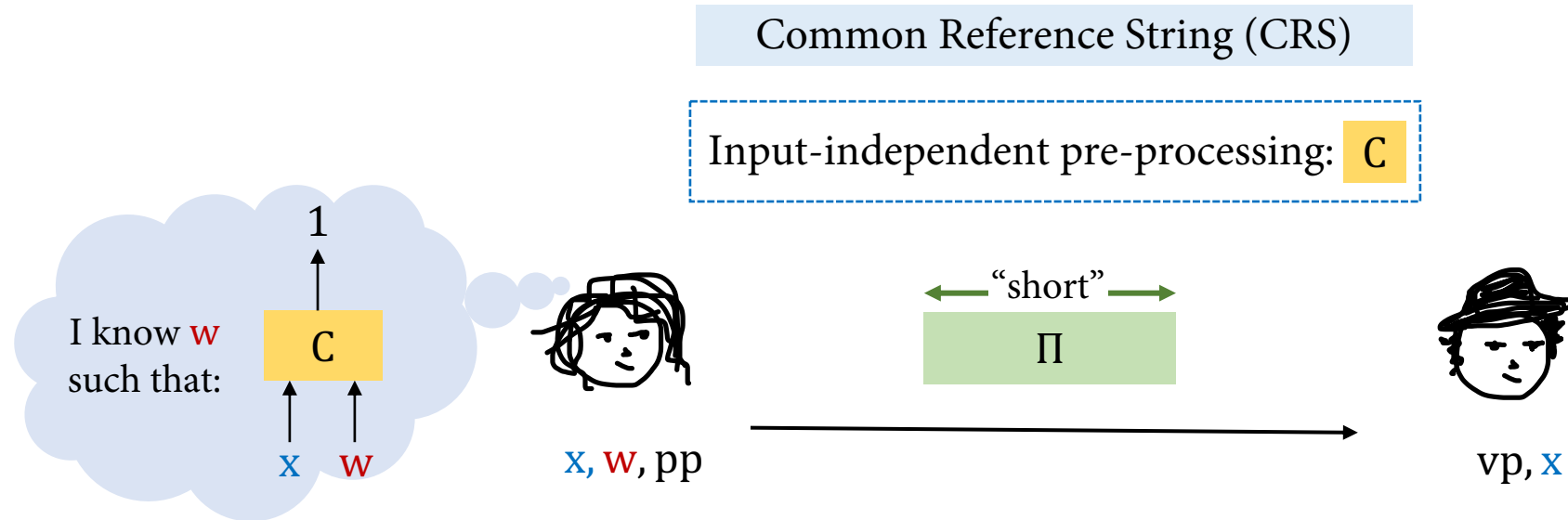
Common Reference String (CRS)



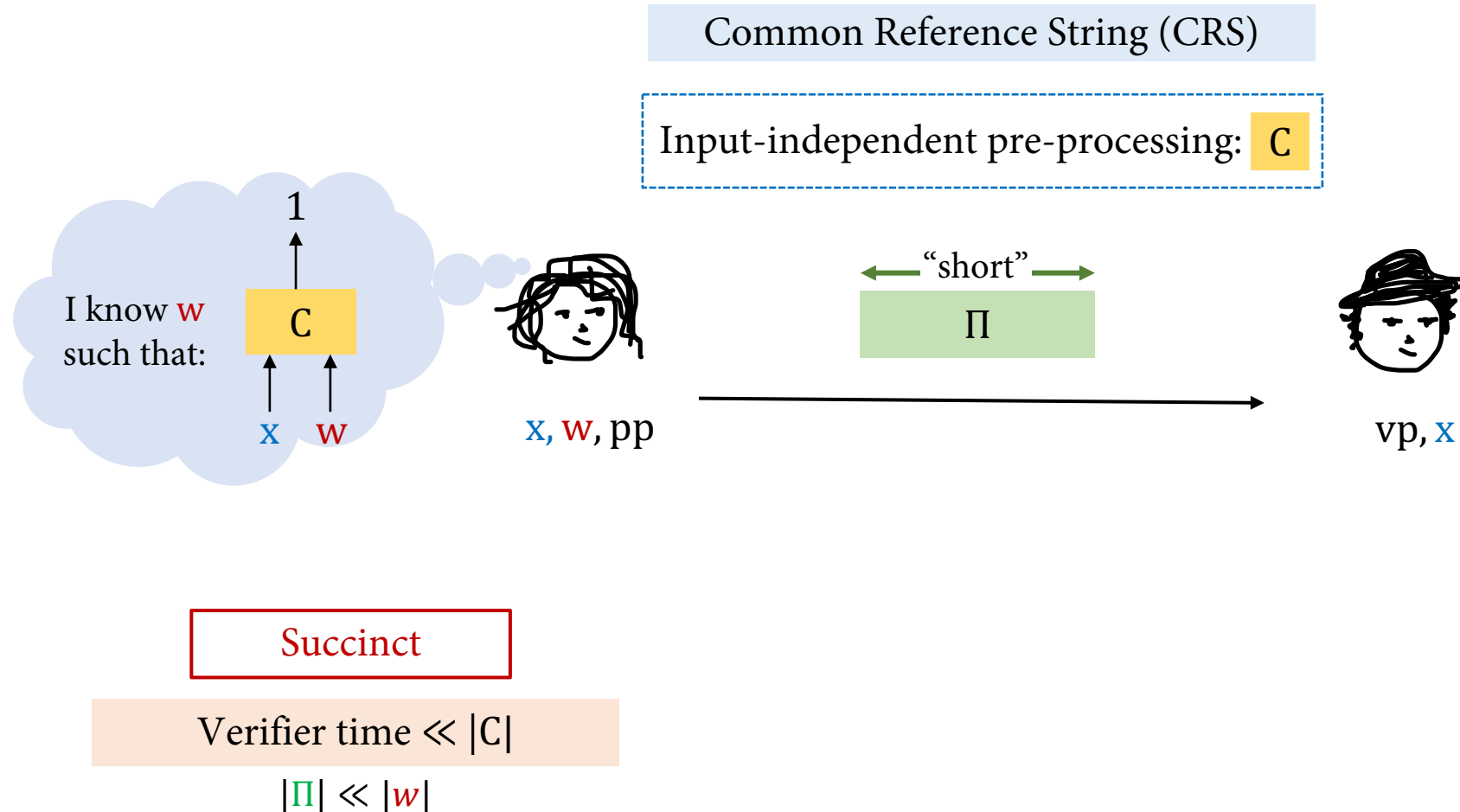
SNARKs: Succinct Non-interactive Arguments of Knowledge



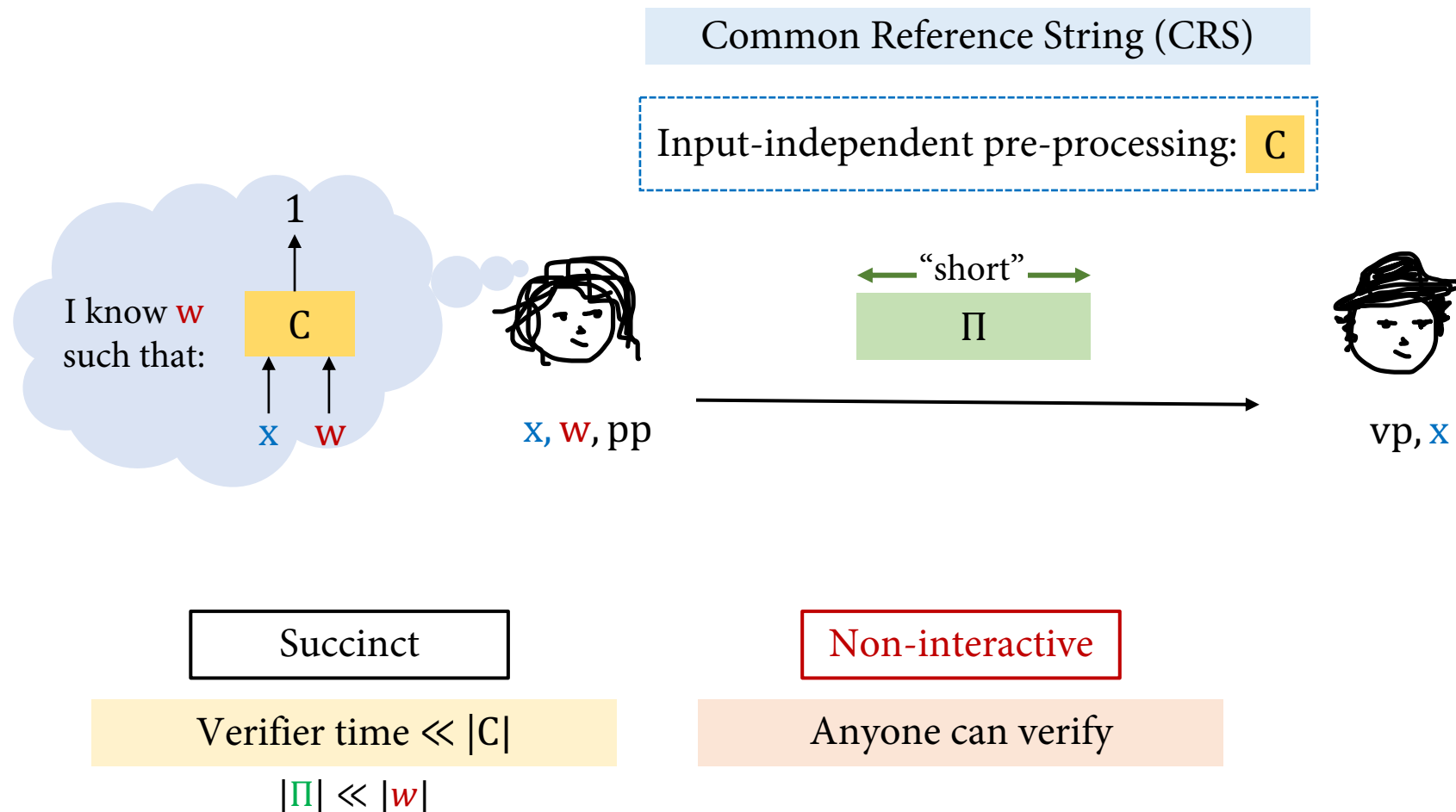
SNARKs: Succinct Non-interactive Arguments of Knowledge



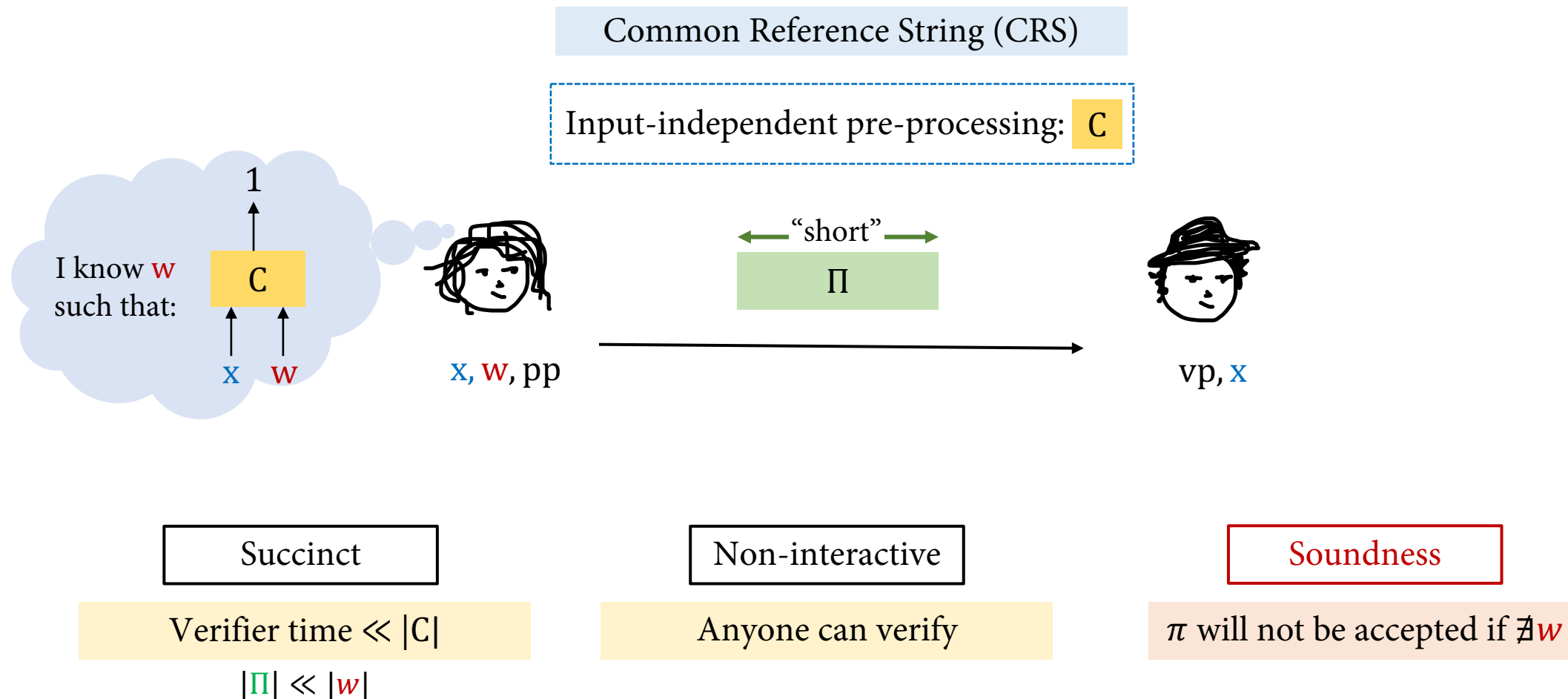
SNARKs: Succinct Non-interactive Arguments of Knowledge



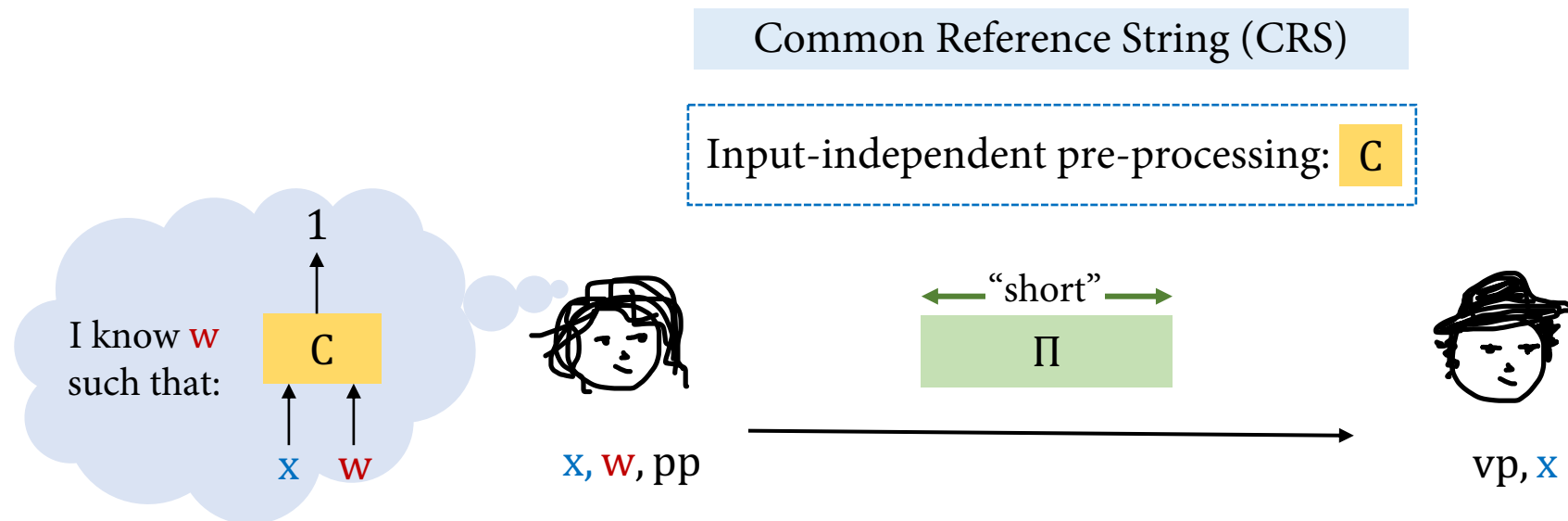
SNARKs: Succinct **Non-interactive** Arguments of Knowledge



SNARKs: Succinct Non-interactive Arguments of Knowledge



SNARKs: Succinct Non-interactive Arguments of Knowledge



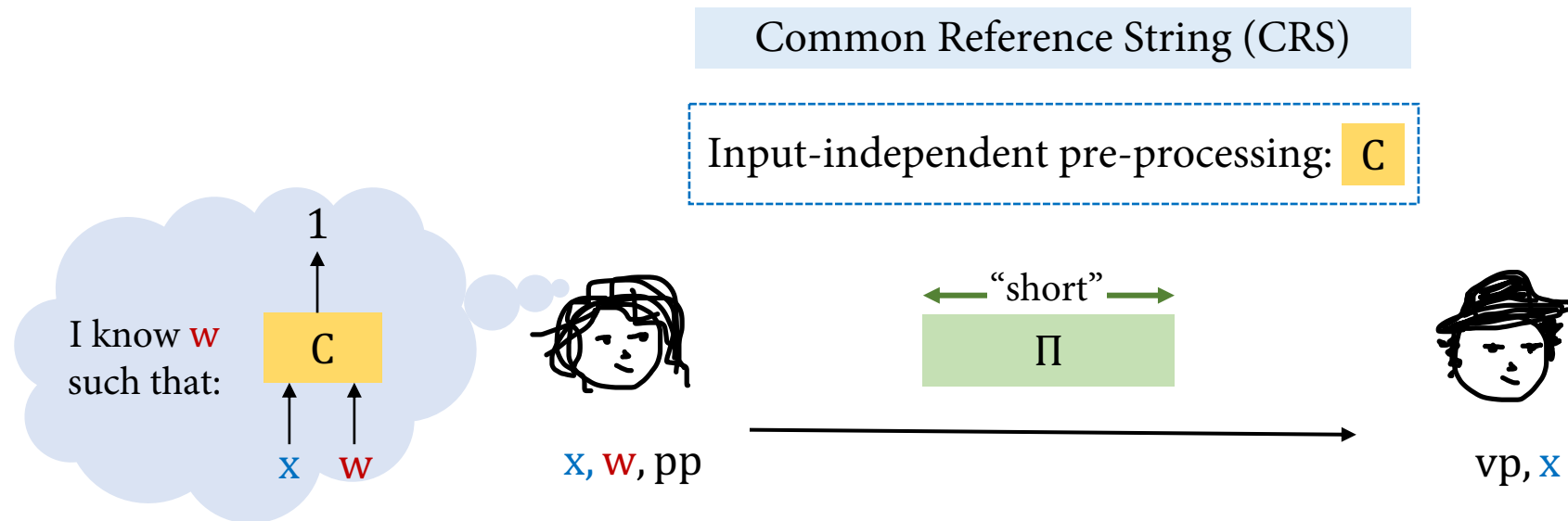
PlonK [Gabizon-Williamson-Ciobotaru'19]

Short proof: $|\Pi| = O(1)$

Prover Time: Grows with $|C|$

Verification Time = $|x| + O(1)$

SNARKs: Succinct Non-interactive Arguments of Knowledge



WHY PLONK?

- widely used in practice
- proof size of 600 bytes
- support for custom and lookup gates.

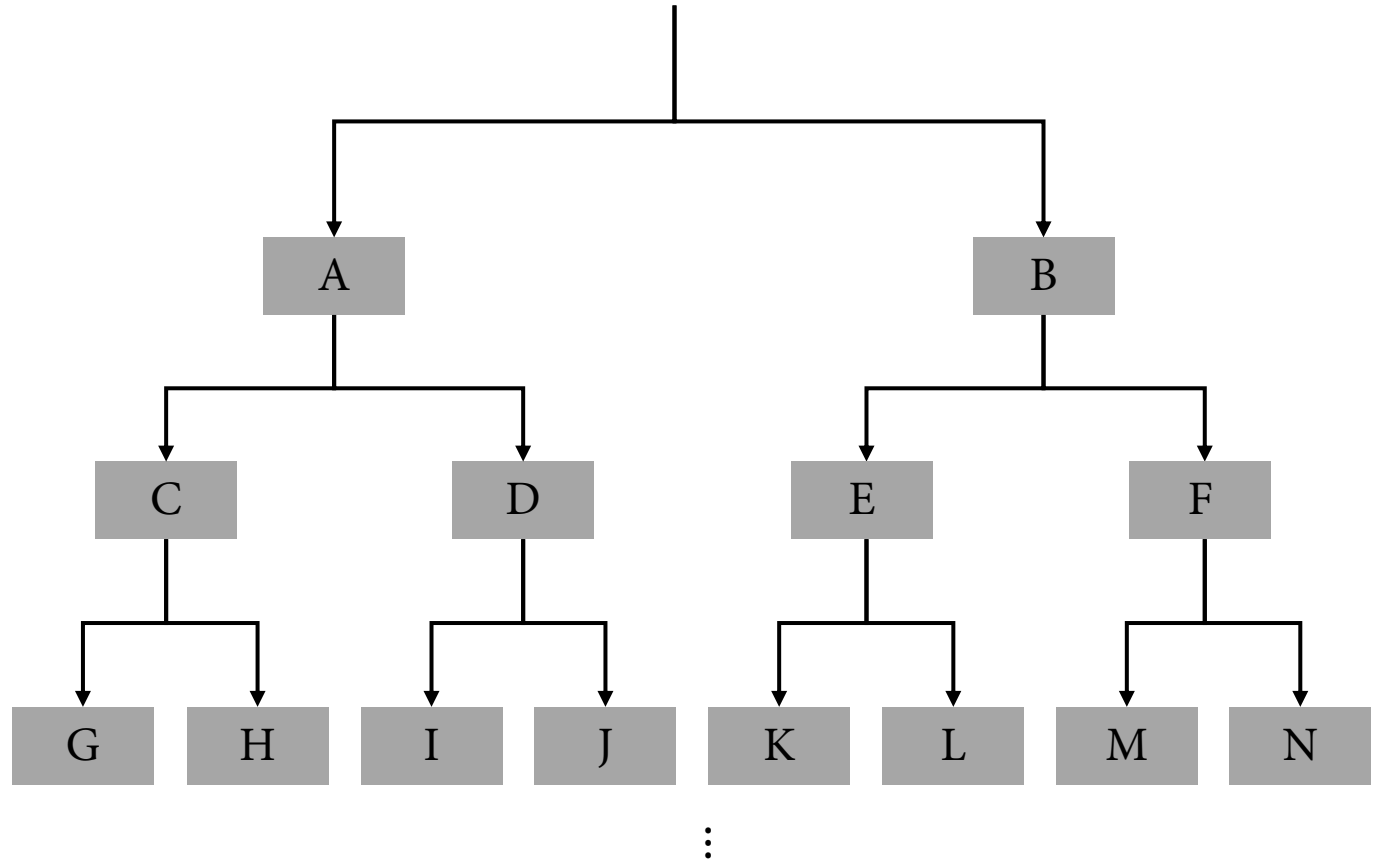
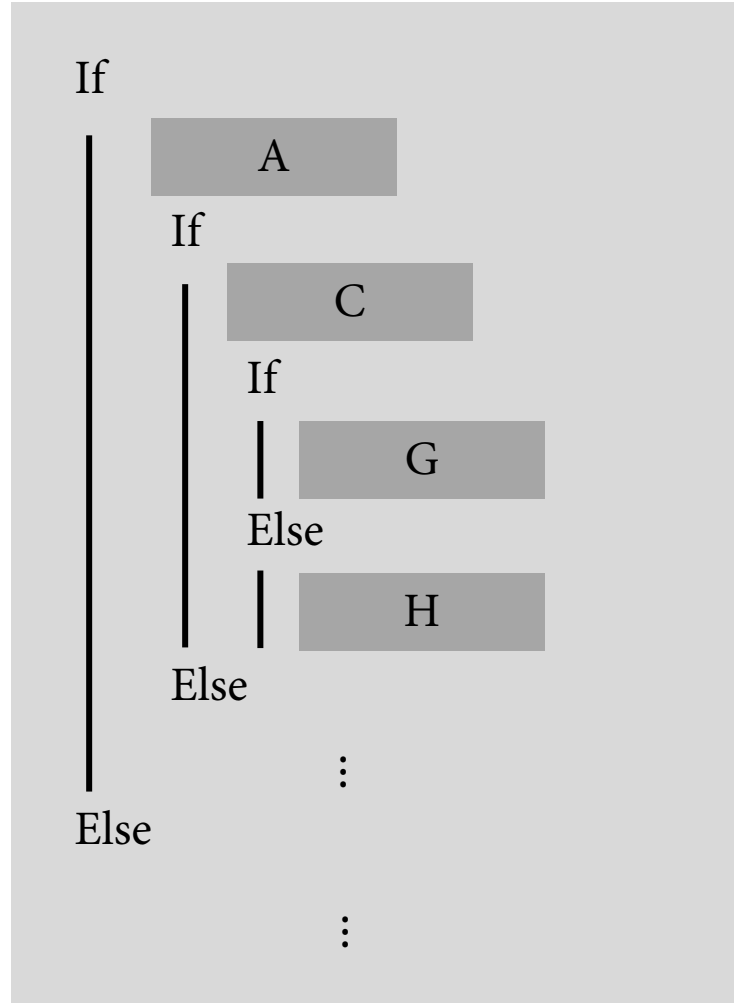
PlonK [Gabizon-Williamson-Ciobotaru'19]

Short proof: $|\Pi| = O(1)$

Prover Time: Grows with $|C|$

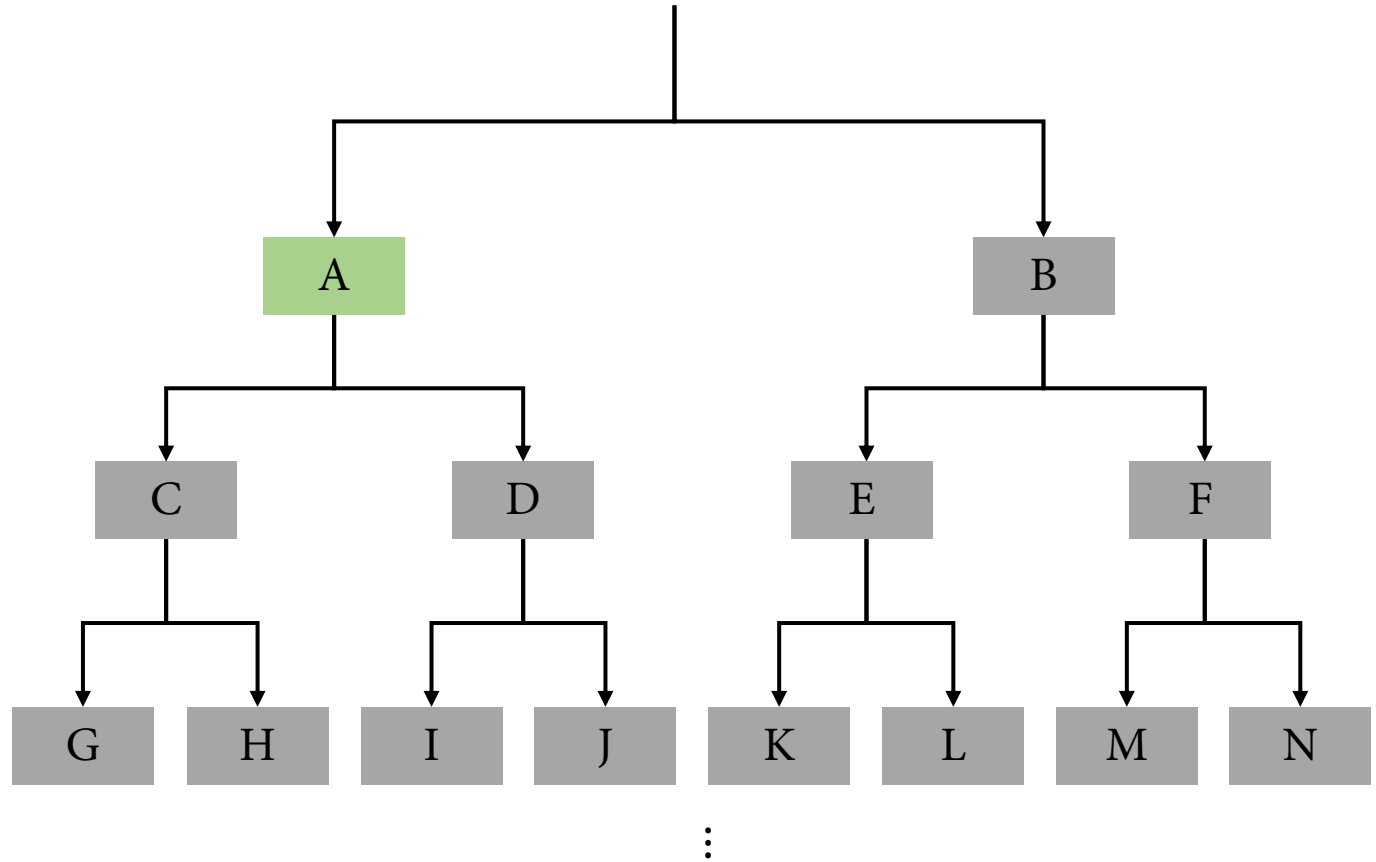
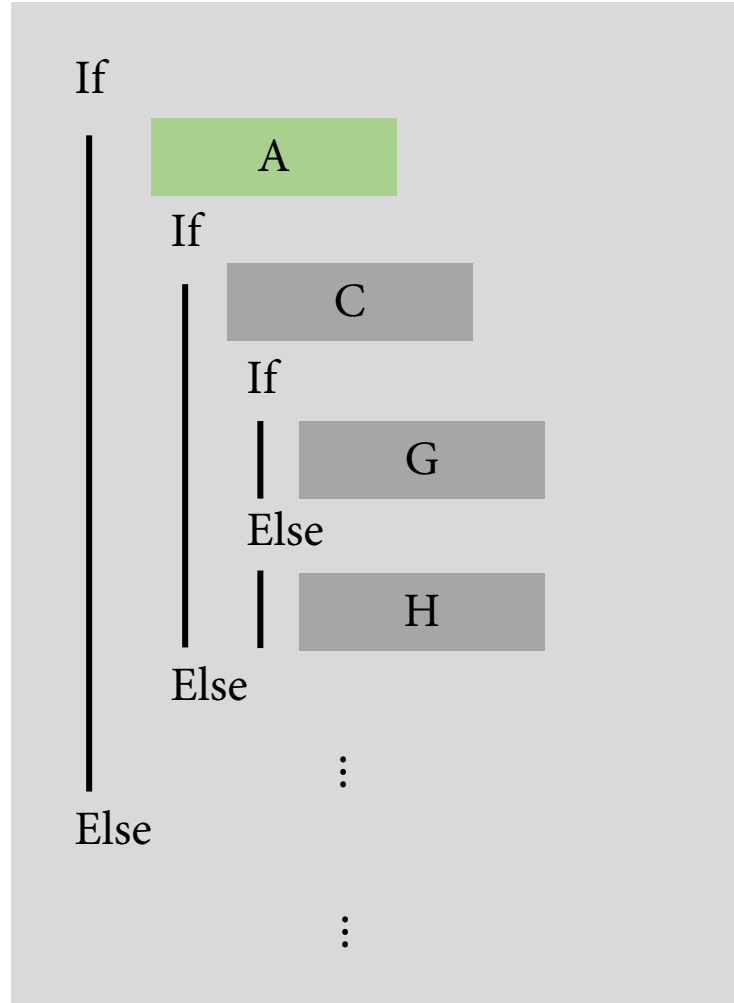
Verification Time = $|x| + O(1)$

SNARKs for Nested-Ifs



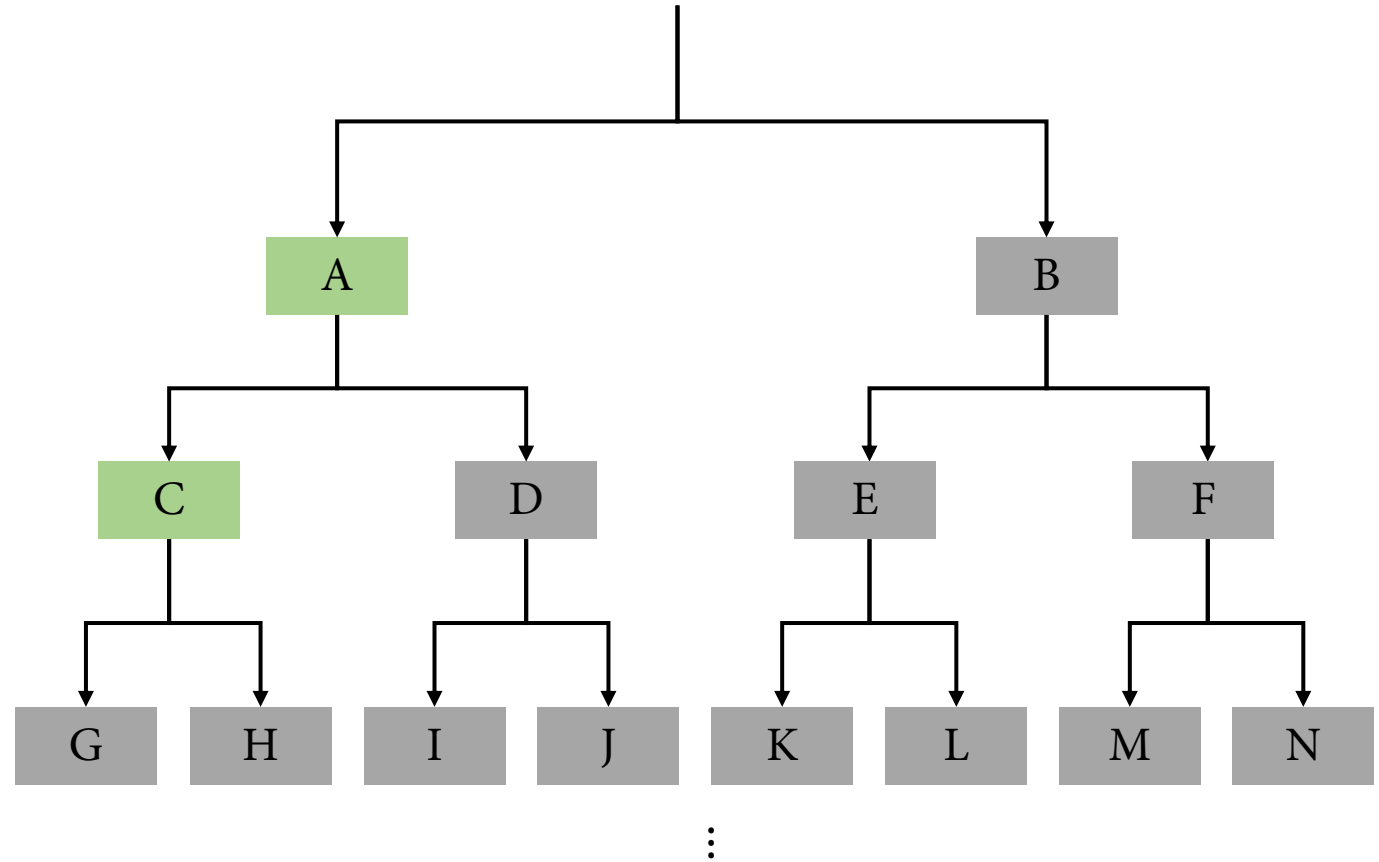
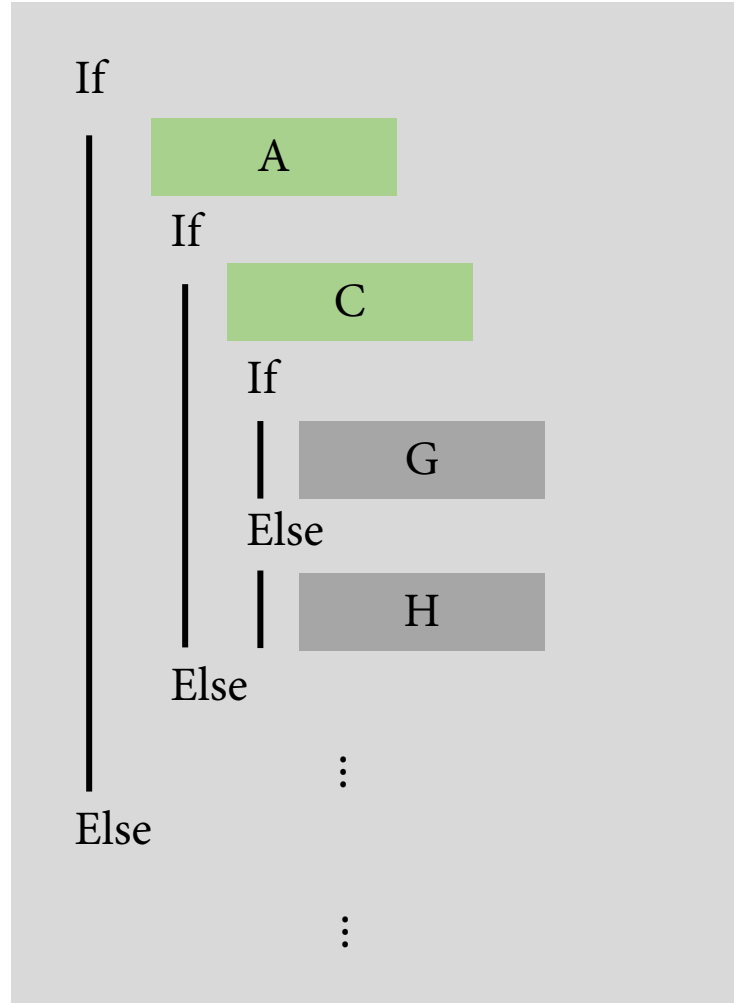
Circuit representation of the program

SNARKs for Nested-Ifs



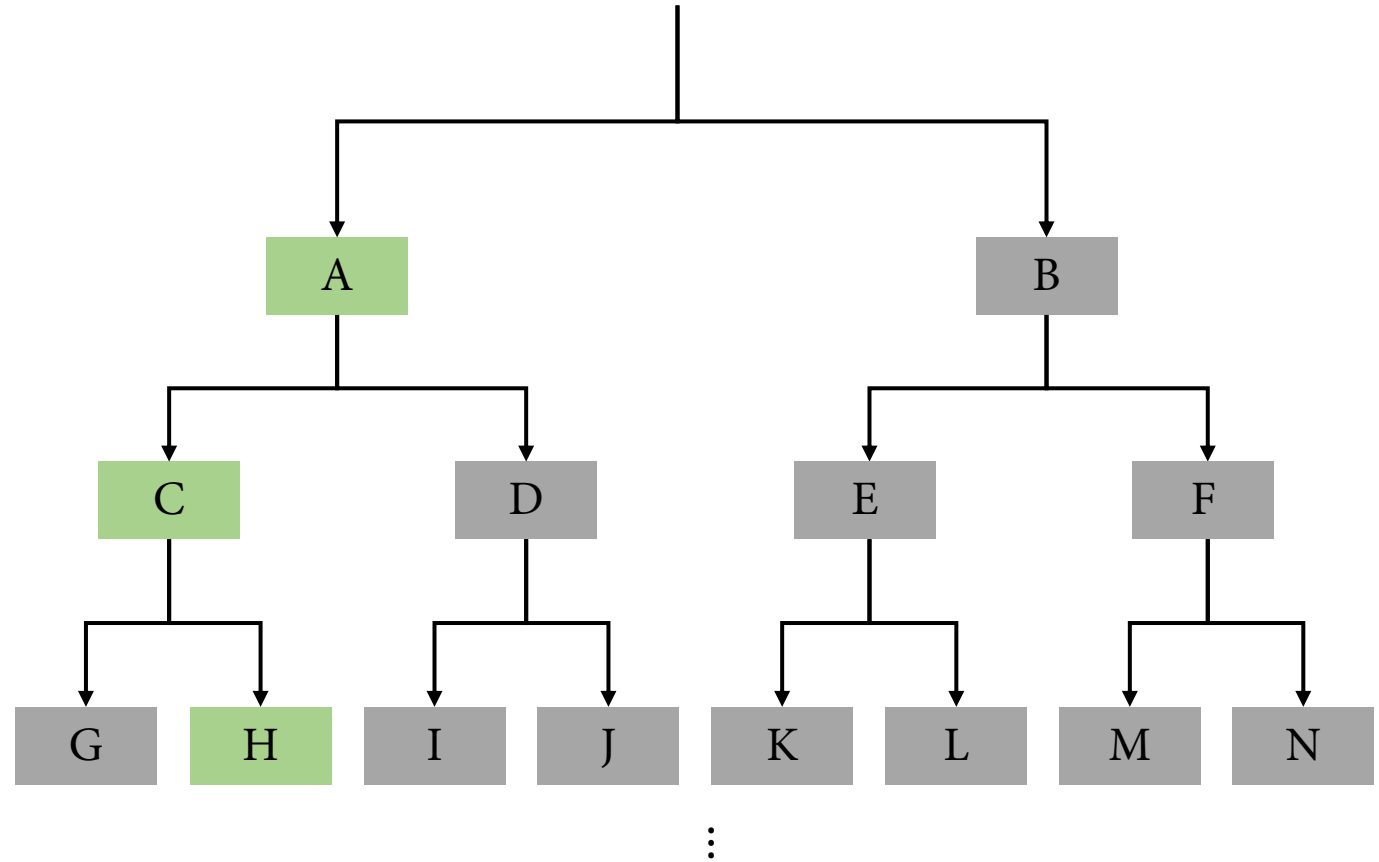
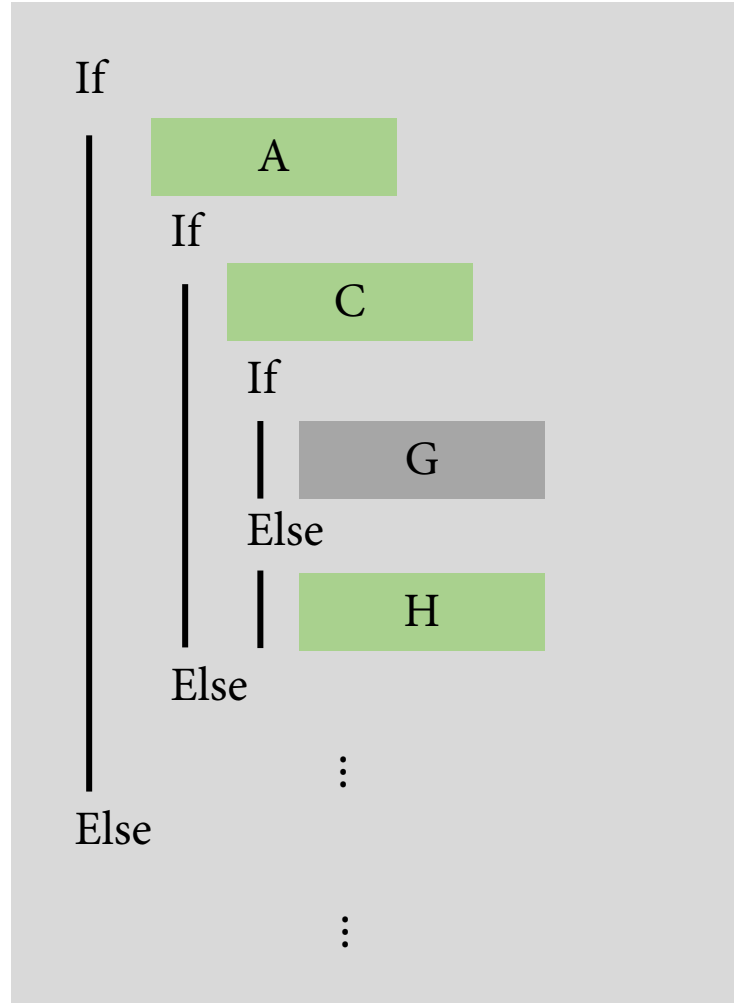
Circuit representation of the program

SNARKs for Nested-Ifs



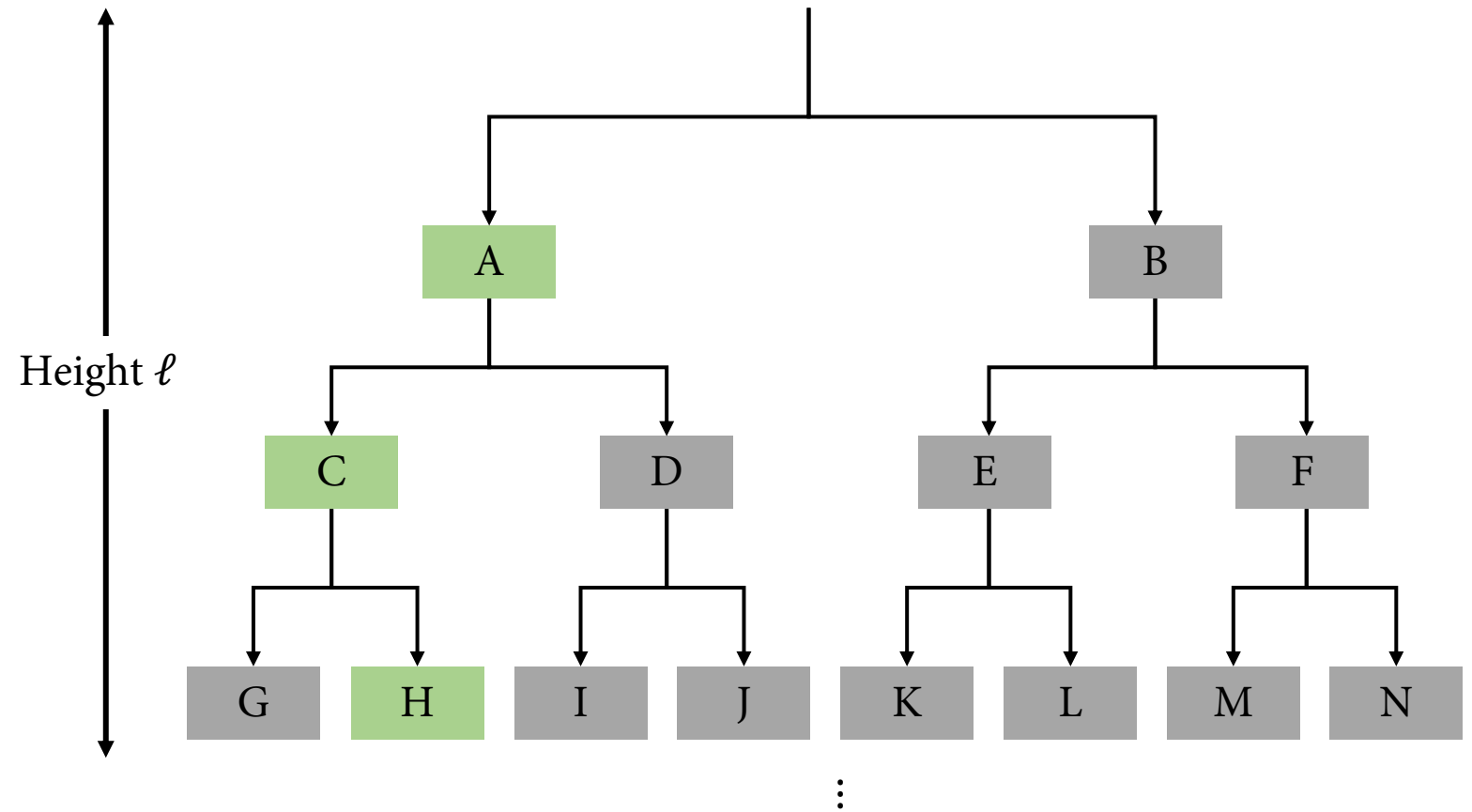
Circuit representation of the program

SNARKs for Nested-Ifs



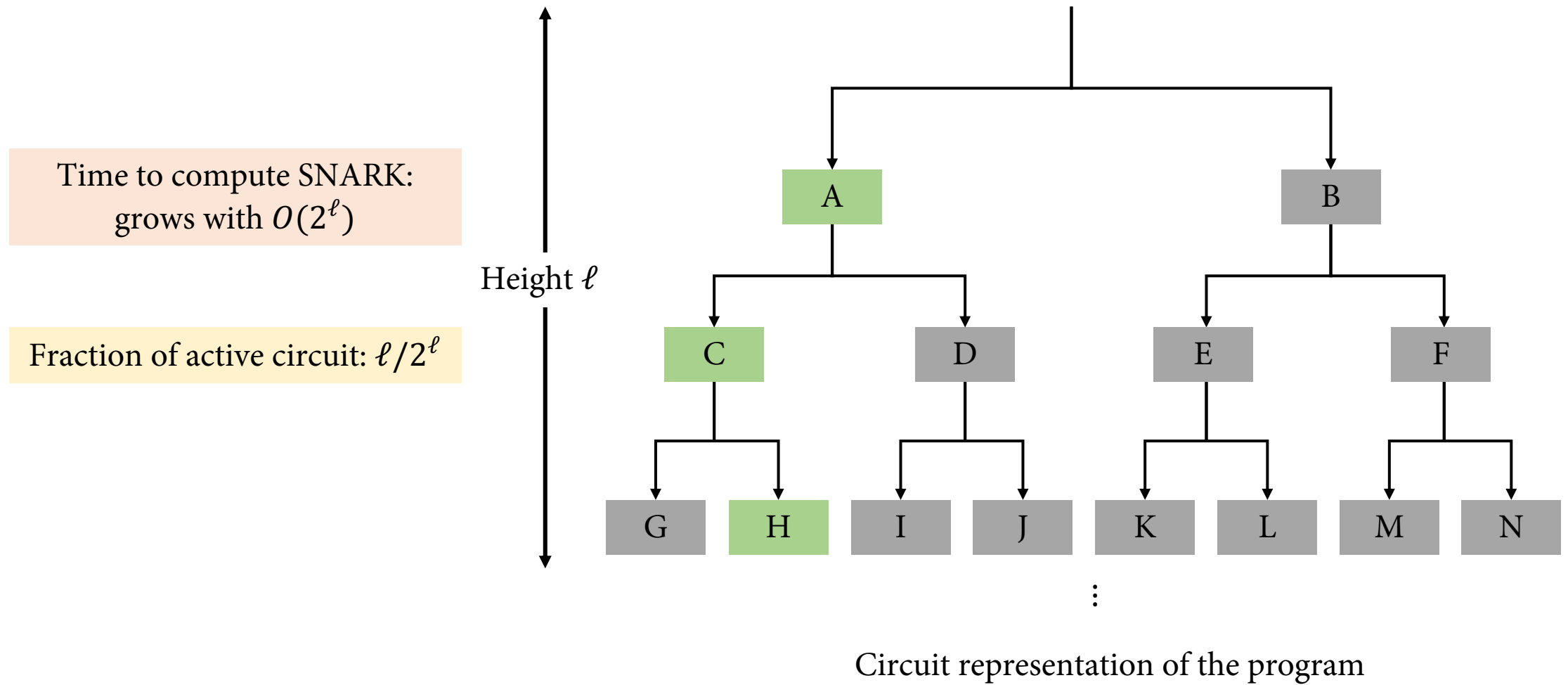
Circuit representation of the program

Drawback: SNARKs for Nested-Ifs

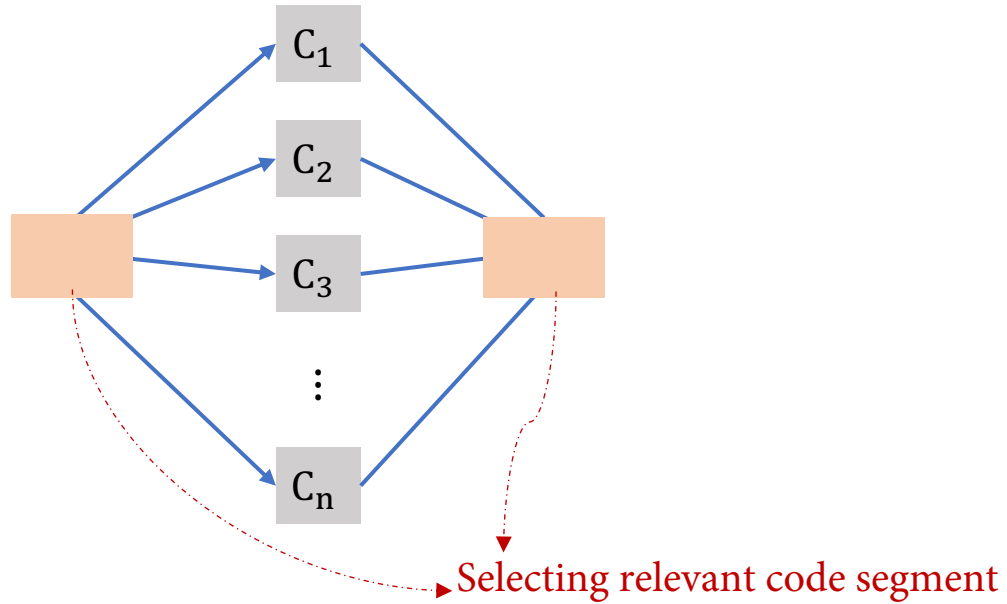


Circuit representation of the program

Drawback: SNARKs for Nested-Ifs

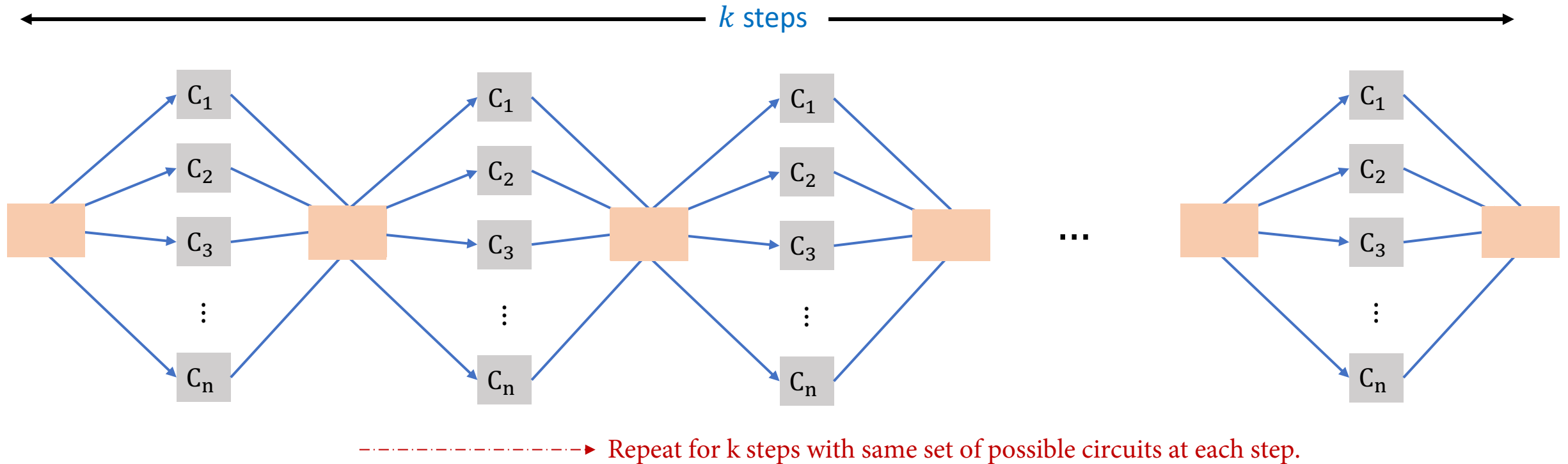


Our Model: Layered Branching Circuit



Choice of n code segments, each of size s .
Only one code segment active for any input.

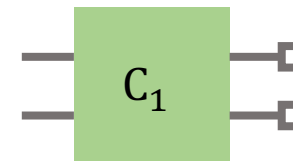
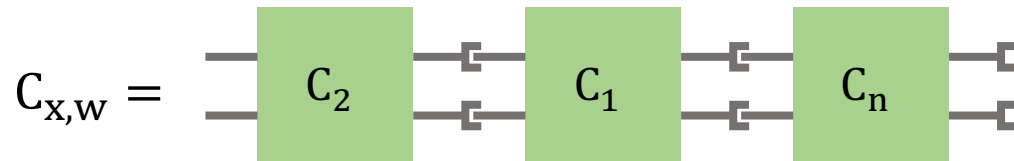
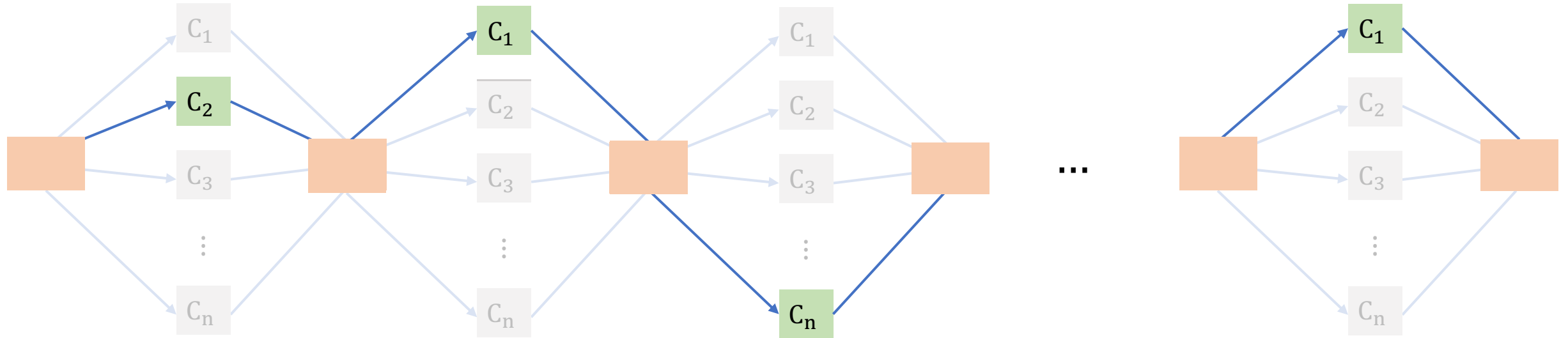
Our Model: Layered Branching Circuit



Choice of n code segments, each of size s .
Only one code segment active for any input.

$$|C| = n \cdot k \cdot s$$

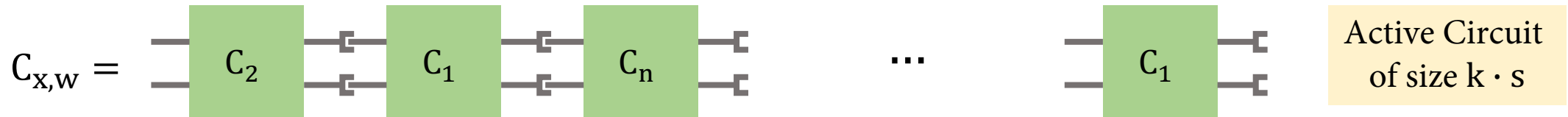
Our Model: Layered Branching Circuit



Active Circuit
of size $k \cdot s$

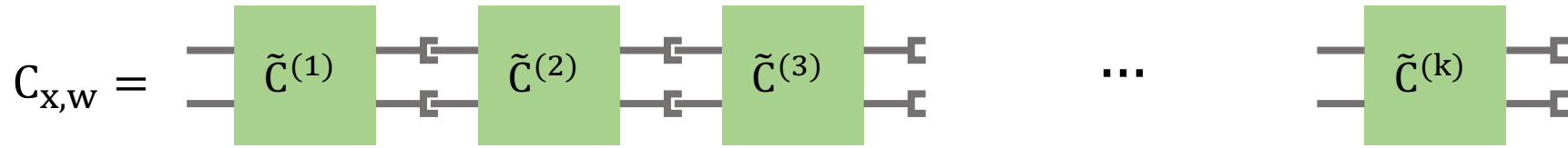
Our Model: Layered Branching Circuit

Apart from Nested-Ifs, this model also captures **Rollups** in Blockchain!
At each step, $n = 100$ (say) **possible transactions** to be executed,
and only one gets executed at each step, depending on the input!



Our Model: Layered Branching Circuit

Active Circuit for an input x, w



$j \in [k]$

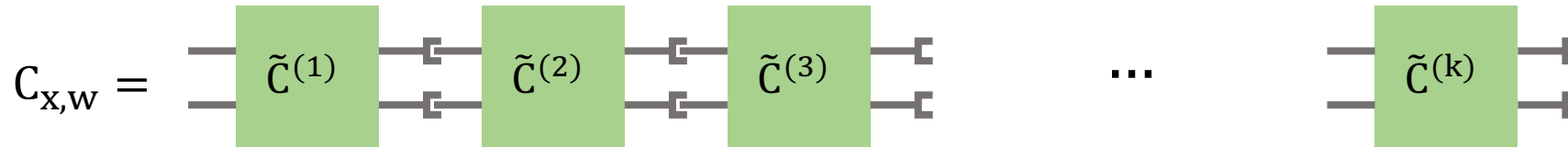
Each $\tilde{C}^{(j)}$ is a circuit C_i

$i \in [n]$

$$|C_{x,w}| = k \cdot s$$

Our Model: Layered Branching Circuit

Active Circuit for an input x, w



$j \in [k]$

Each $\tilde{C}^{(j)}$ is a circuit C_i

$$|C_{x,w}| = k \cdot s$$

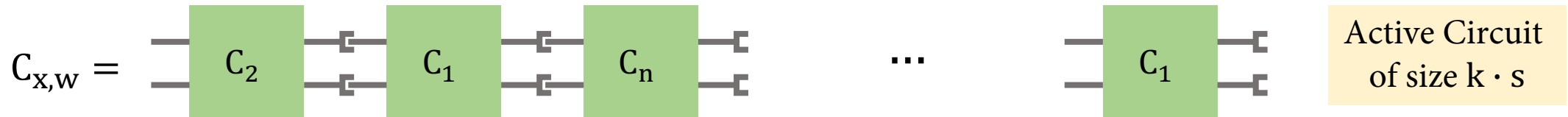
$i \in [n]$

Apart from Nested-Ifs, this model also captures **Rollups** in Blockchain!
At each step, $n = 100$ (say) **possible transactions** to be executed,
and only one gets executed at each step, depending on the input!

Our Model: Layered Branching Circuit

Can we construct SNARKs for Layered Branching Circuits where the
online prover time grows only with the size of the active circuit?

*Allow the prover to perform a one-time input-independent pre-processing of the entire circuit C



Prior Works

- A la carte proof cost
 - Buffet [Wahby-Setty-Ren-Blumberg-Walfish'15], vRAM [Zhang-Genkin-Katz-Papadopoulos-Papamanthou'18]
 - Mirage [Kosba-Papadopoulos-Papamanthou-Song'20]
 - (Non-Uniform) Incrementally Verifiable Proof
 - Sangria, SuperNova [Kothapalli-Setty'22], ProtoStar [Bünz-Chen'23]
 - STARKs
 - eSTARK [Masip-Ardevol-Guzmán-Albiol-Baylina-Melé-Muñoz-Tapia'23]
 - Commit and Prove SNARKs
 - LegoSNARK [Campanelli-Fiore-Querol'19], Gepetto [Costello-Fournet-Howell-Kohlweiss-Kreuter-Naehrig-Parno-Zahur'15]
-
- ```
graph LR; Buffet[Buffet [Wahby-Setty-Ren-Blumberg-Walfish'15], vRAM [Zhang-Genkin-Katz-Papadopoulos-Papamanthou'18]] -- red --> A[Not constant proof size]; Mirage[Mirage [Kosba-Papadopoulos-Papamanthou-Song'20]] -- green --> B[Input-dependent prover pre-processing.]; Sangria[Sangria, SuperNova [Kothapalli-Setty'22], ProtoStar [Bünz-Chen'23]] -- green --> B; eSTARK[eSTARK [Masip-Ardevol-Guzmán-Albiol-Baylina-Melé-Muñoz-Tapia'23]] -- blue --> C[Not black-box in Cryptography.]
```

# Our Result

Theorem: Sublinear Prover  $\mathcal{P}_{\text{lon}\mathcal{K}}$

Online Prover Time -  $O(k_s (\log k_s + \log n))$

Proof Size -  $O(1)$

**Black-box** in Cryptography with **input-independent** pre-processing for prover and verifier.

\*Universal setup, support for custom and lookup gates.

# Our Result

Theorem: Sublinear Prover  $\mathcal{P}_{\text{lon}\mathcal{K}}$

Online Prover Time -  $O(k_s (\log k_s + \log n))$

Proof Size -  $O(1)$

**Black-box** in Cryptography with **input-independent** pre-processing for prover and verifier.

\*Universal setup, support for custom and lookup gates.

Main idea: generalize the lookup argument to a “*segment lookup*” argument.

# Our Result

Theorem: Sublinear Prover  $\mathcal{P}_{\text{lon}\mathcal{K}}$

Online Prover Time -  $O(ks (\log ks + \log n))$

Proof Size -  $O(1)$

Black-box in Cryptography with input-independent pre-processing for prover and verifier.

\*Universal setup, support for custom and lookup gates.

A concurrent work [Di-Xia-Nguyen-Tyagi'23] uses similar ideas (“tuple lookup”) to achieve an almost identical result where the online prover time is  $\tilde{O}((k + n)s)$  as opposed to  $\tilde{O}(ks)$ .

# Experimental Results

Proof Size and Verification Cost (Comparison to Baseline PlonK)

|                          | <u>PlonK</u>                      | <u>SublonK</u>                      |
|--------------------------|-----------------------------------|-------------------------------------|
| <b>Proof size</b>        | 9 $\mathbb{G}$ and 6 $\mathbb{F}$ | 42 $\mathbb{G}$ and 12 $\mathbb{F}$ |
| <b>Verification Cost</b> | 18 $\mathbb{G}$ and 2 Pairings    | 27 $\mathbb{G}$ and 23 Pairings     |

# Experimental Results

## Proof Size and Verification Cost (Comparison to Baseline PlonK)

|                          | <u>PlonK</u>                      | <u>SublonK</u>                      |
|--------------------------|-----------------------------------|-------------------------------------|
| <b>Proof size</b>        | 9 $\mathbb{G}$ and 6 $\mathbb{F}$ | 42 $\mathbb{G}$ and 12 $\mathbb{F}$ |
| <b>Verification Cost</b> | 18 $\mathbb{G}$ and 2 Pairings    | 27 $\mathbb{G}$ and 23 Pairings     |

\*A follow-up work [Campanelli-Faonio-Fiore-Li-Lipmaa'23] improves efficiency by generalizing our segment lookup to a “*matrix lookup*”.

# Thank You

<https://petsymposium.org/popets/2024/popets-2024-0080.pdf>