SublonK

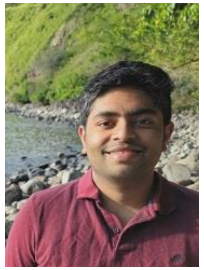
Sublinear Prover PlonK

Arka Rai Choudhuri
NTT Research


Sanjam Garg
UC Berkeley


Aarushi Goel
NTT Research


Sruthi Sekar
UC Berkeley


Rohit Sinha
Swirlds Labs

IACR
@IACR_News

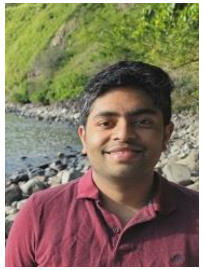
#ePrint $\mathcal{S}\mathfrak{ublon}\mathcal{K}$: Sublinear Prover $\mathcal{P} \mathfrak{lon}\mathcal{K}$

Arka Rai Choudhuri
NTT Research


Sanjam Garg
UC Berkeley


Aarushi Goel
NTT Research


Sruthi Sekar
UC Berkeley


Rohit Sinha
Swirlds Labs

IACR
@IACR_News

#ePrint $\mathcal{S}\mathfrak{ublon}\mathcal{K}$: Sublinear Prover $\mathcal{P} \mathfrak{lon}\mathcal{K}$

Muhammed Esgin
@mfesgin

What a catchy paper title!

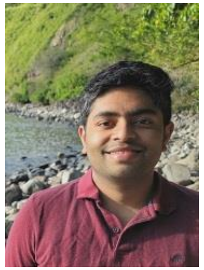
Kostas Kryptos
@kostascrypto

Only cryptographers can read this title

Jack
@j_mcph

lol



Arka Rai Choudhuri
NTT Research

Sanjam Garg
UC Berkeley

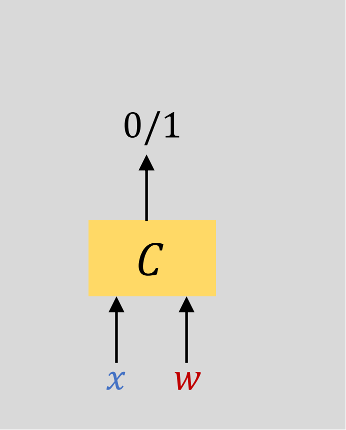
Aarushi Goel
NTT Research

Sruthi Sekar
UC Berkeley

Rohit Sinha
Swirlds Labs

Succinct Non-Interactive Arguments of Knowledge (SNARKs)

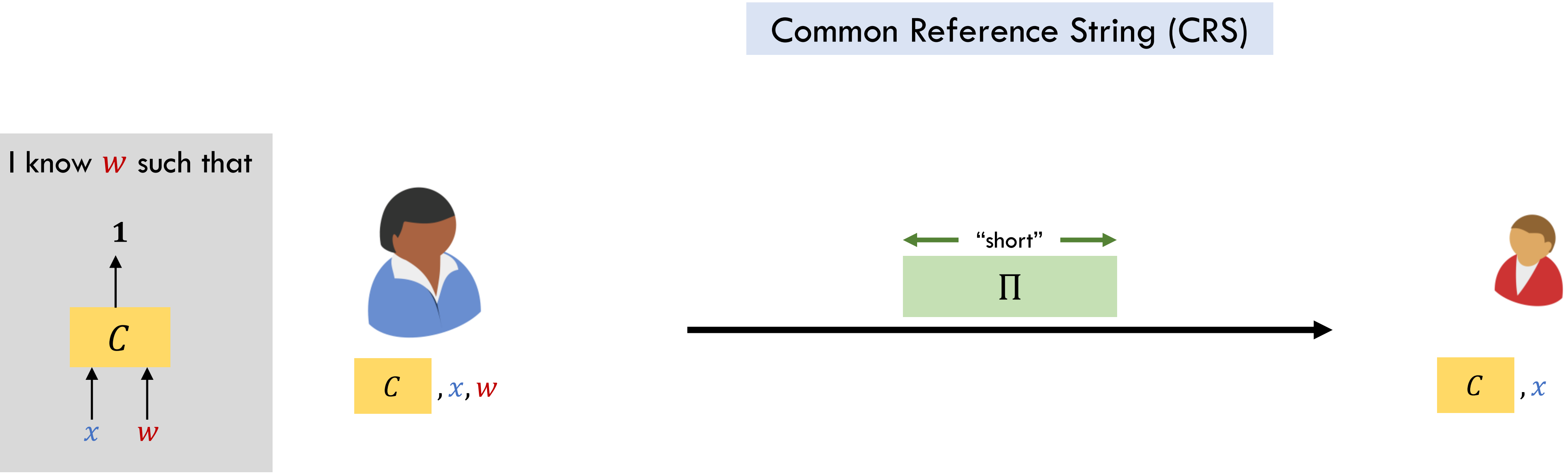
Common Reference String (CRS)



C, x, w

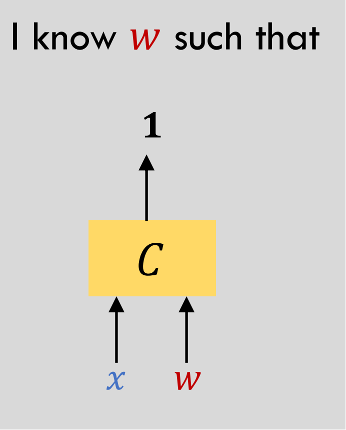
C, x

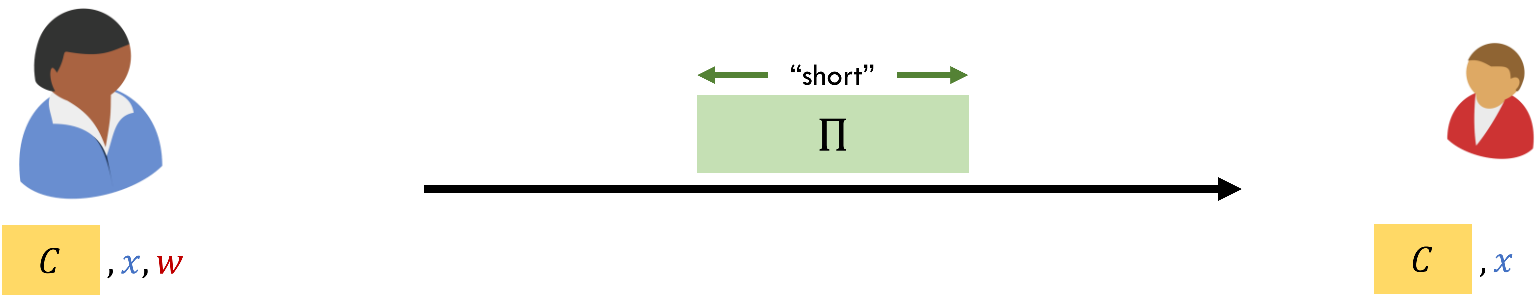
Succinct Non-Interactive Arguments of Knowledge (SNARKs)



Succinct Non-Interactive Arguments of Knowledge (SNARKs)

Common Reference String (CRS)

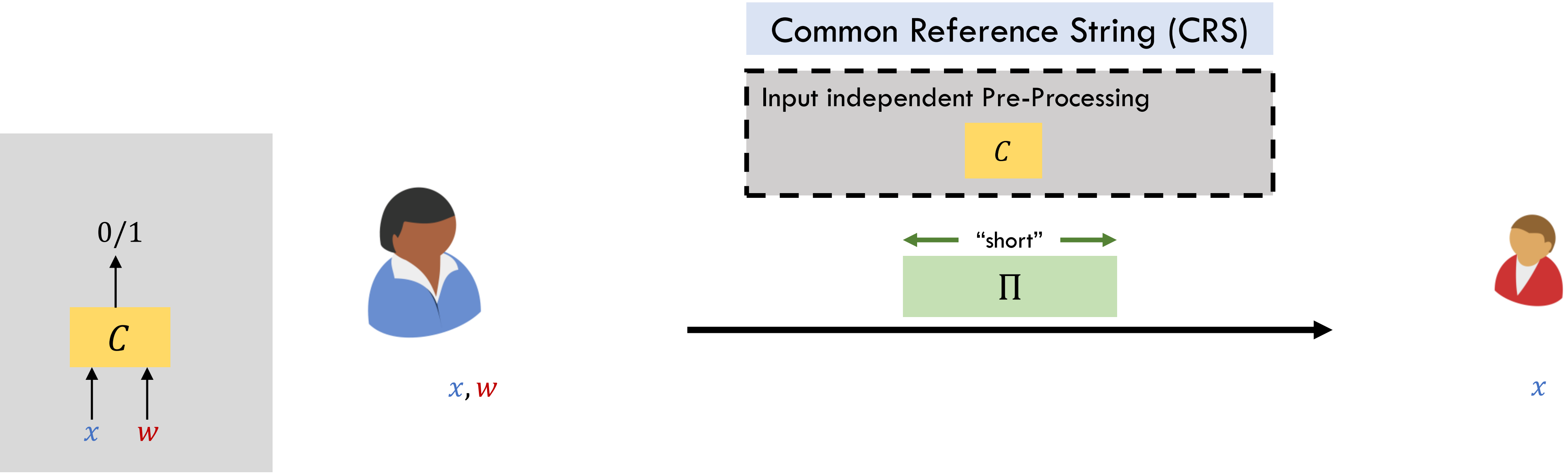




Short proof: $|\Pi| < |w|$

Prover Time: Grows with $|C|$

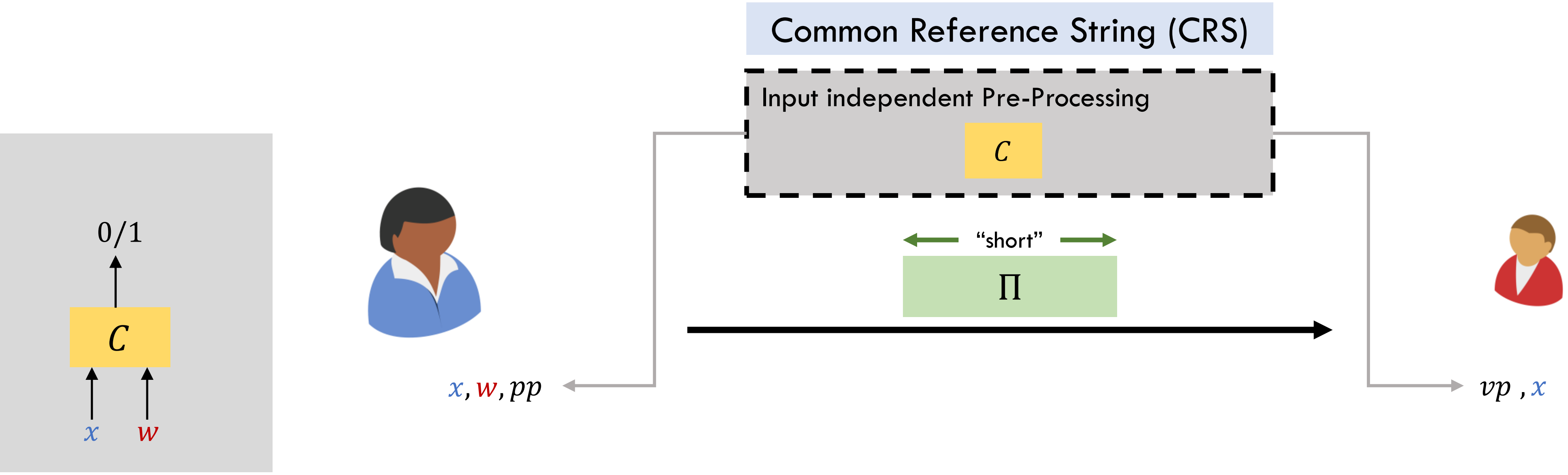
Succinct Non-Interactive Arguments of Knowledge (SNARKs)



Short proof: $|\Pi| < |w|$

Prover Time: Grows with $|C|$

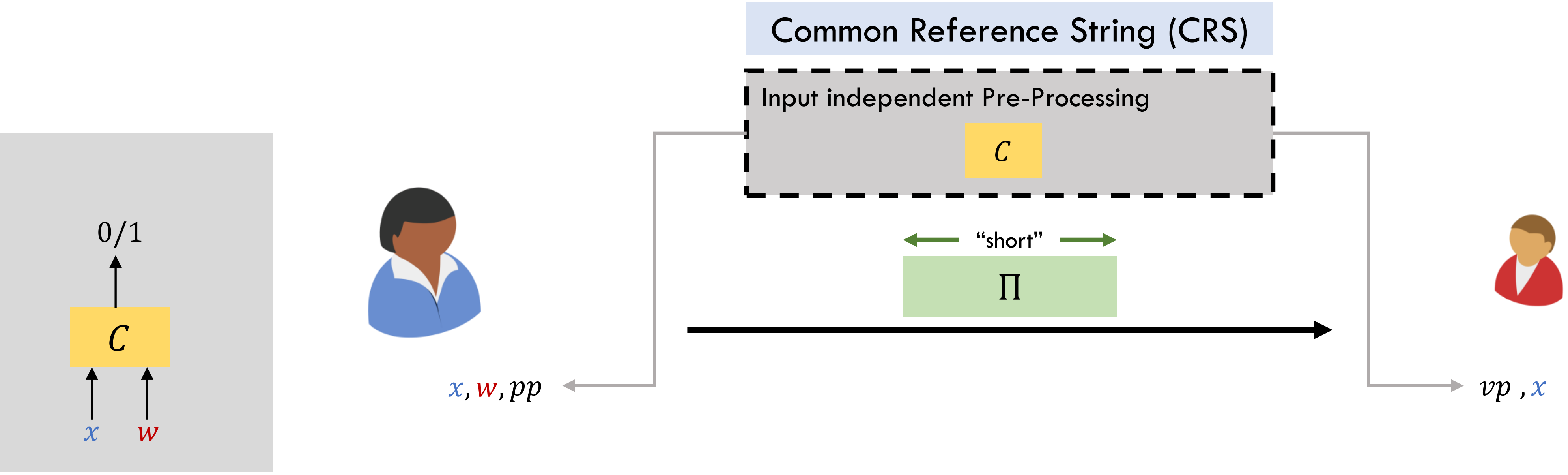
Succinct Non-Interactive Arguments of Knowledge (SNARKs)



Short proof: $|\Pi| < |w|$

Prover Time: Grows with $|C|$

Succinct Non-Interactive Arguments of Knowledge (SNARKs)

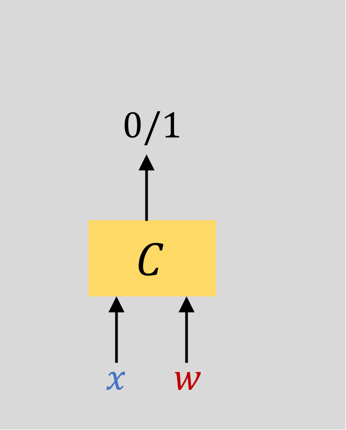


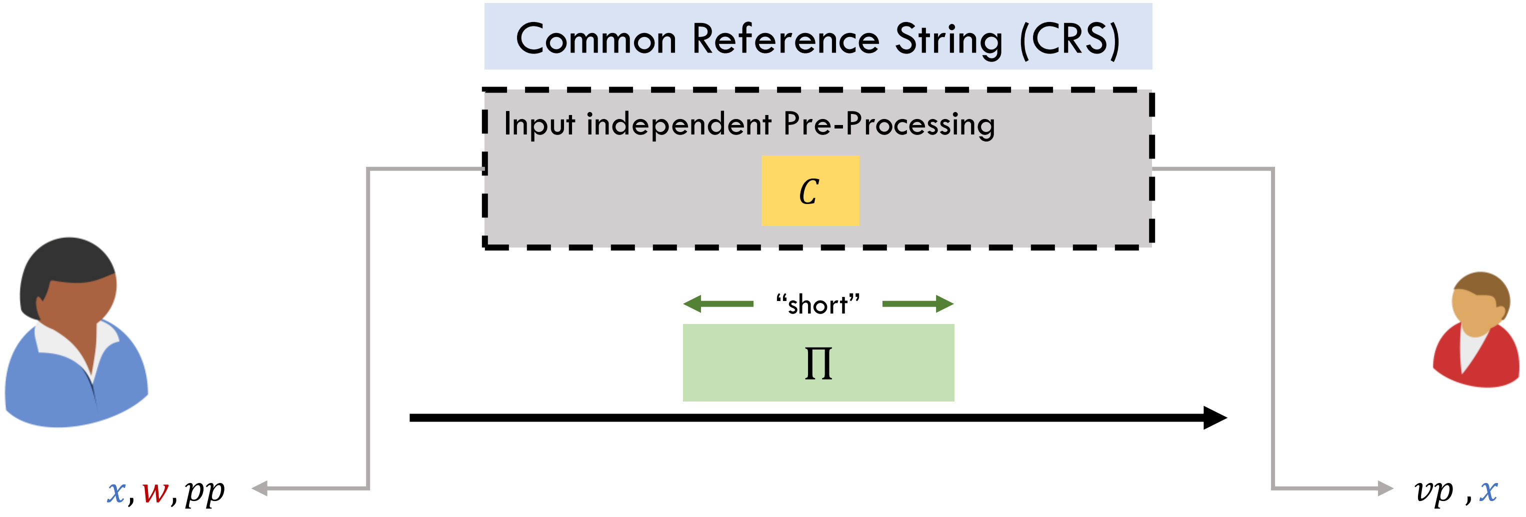
Short proof: $|\Pi| < |w|$

Prover Time: Grows with $|C|$

Verification Time $< |x| + |C|$

Succinct Non-Interactive Arguments of Knowledge (SNARKs)





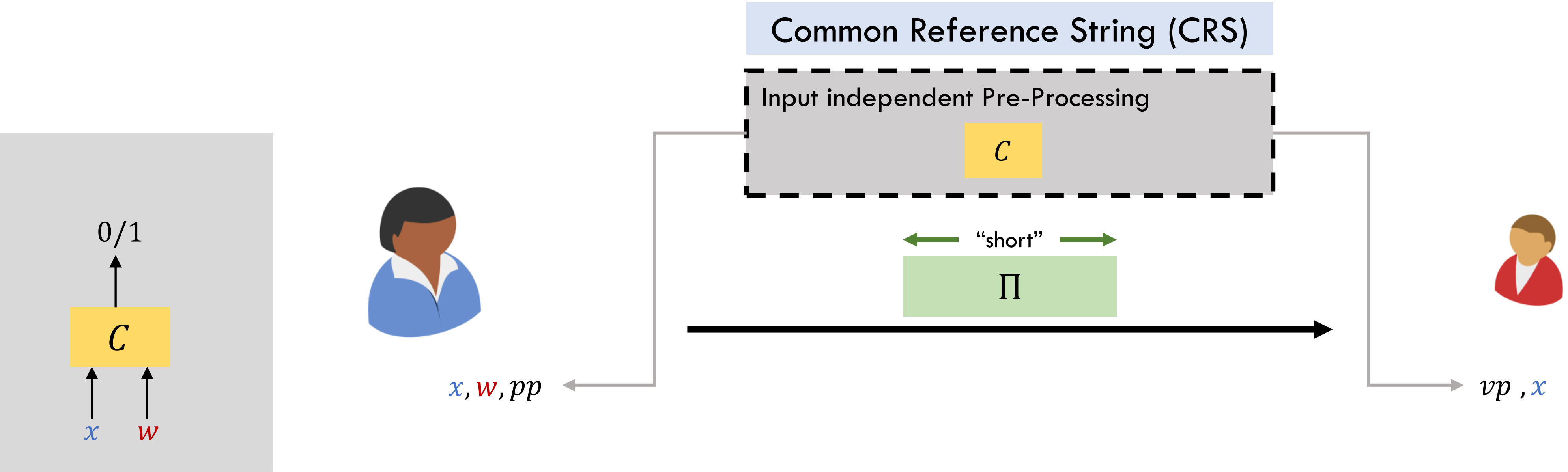
$\mathcal{P}\mathfrak{lon}\mathcal{K}$ [Gabizon-Williamson-Ciobotaru'19]

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

Prover Time: Grows with $|C|$

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

Succinct Non-Interactive Arguments of Knowledge (SNARKs)



$\mathcal{P}\mathfrak{lon}\mathcal{K}$ [Gabizon-Williamson-Ciobotaru'19]

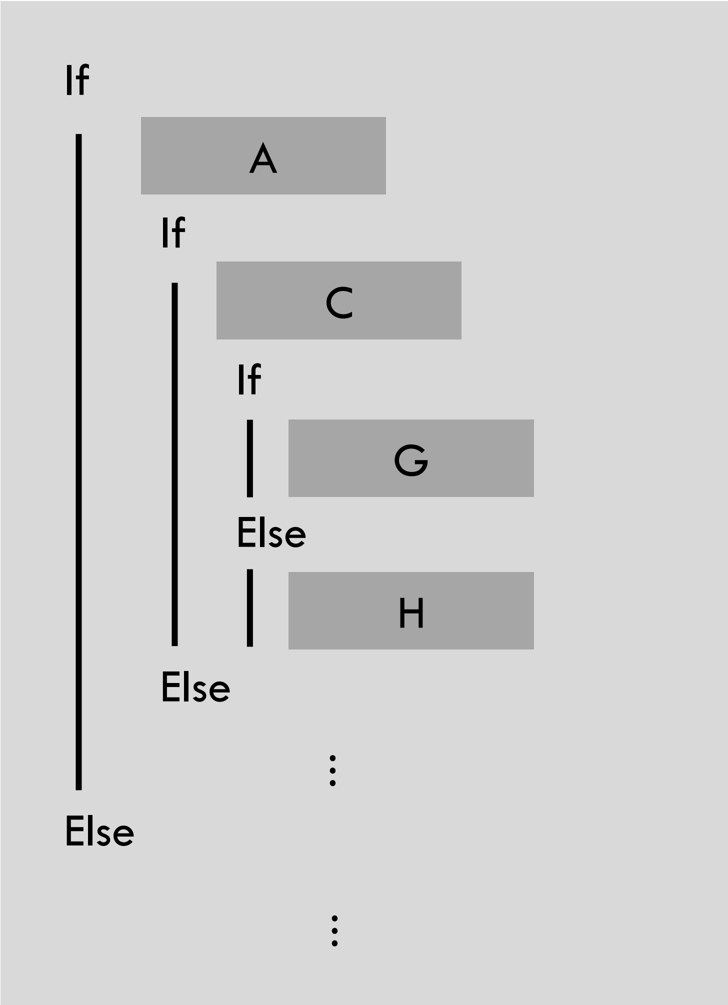
Short proof: $|\Pi| = O(1)$

Prover Time: Grows with $|C|$

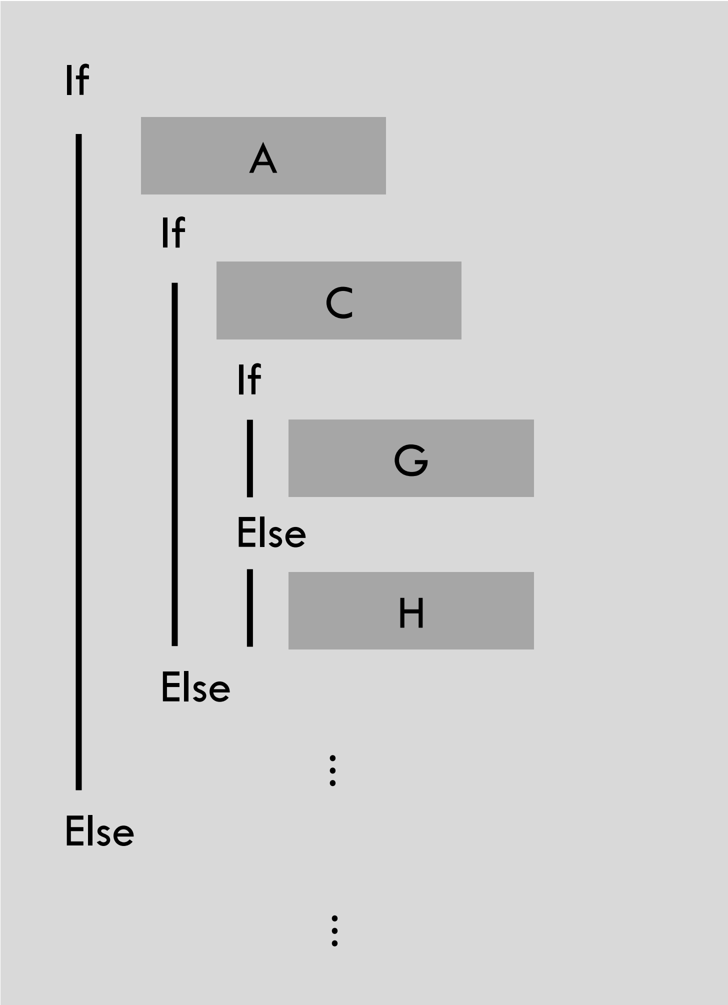
Verification Time $= |x| + O(1)$

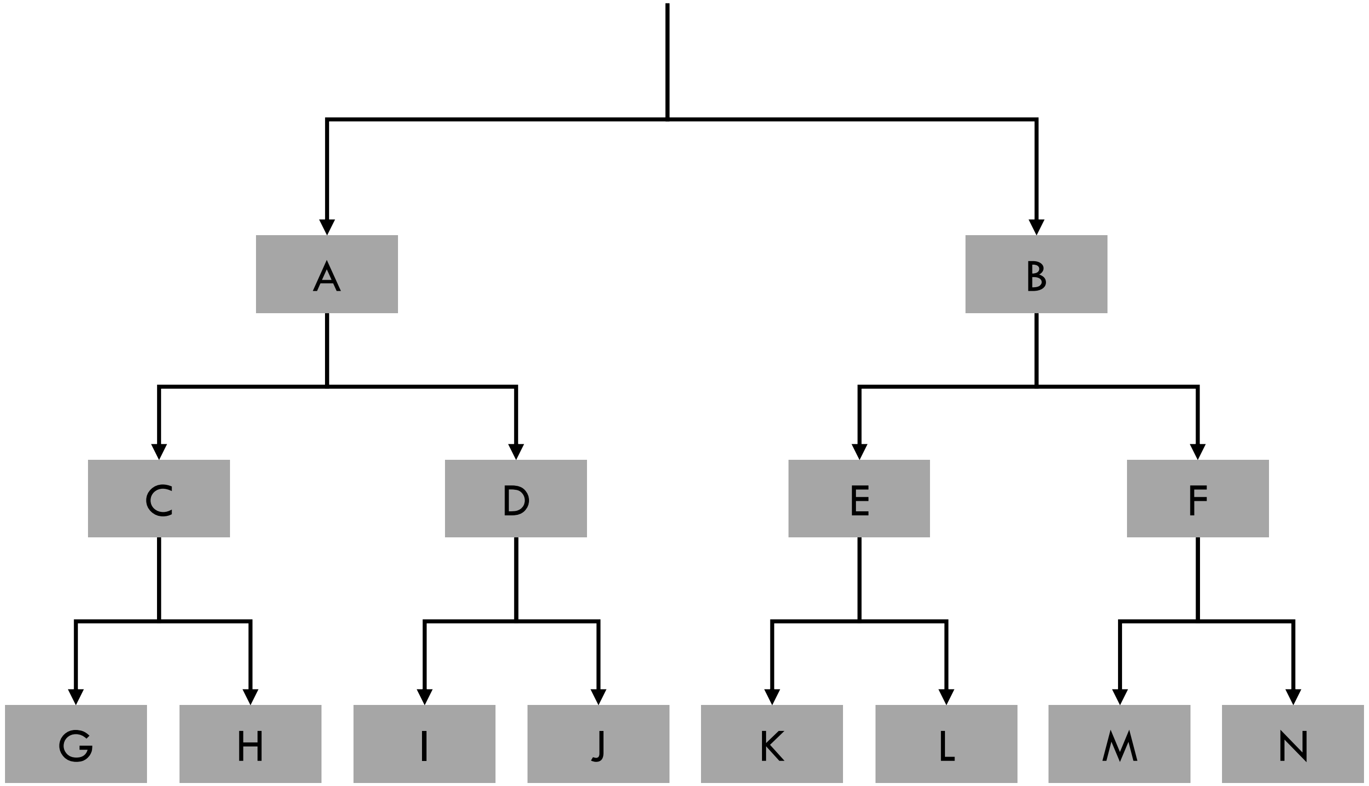
- widely used in practice
- proof size of 600 bytes
- support for custom and lookup gates.

SNARKs for Nested-Ifs



SNARKs for Nested-Ifs

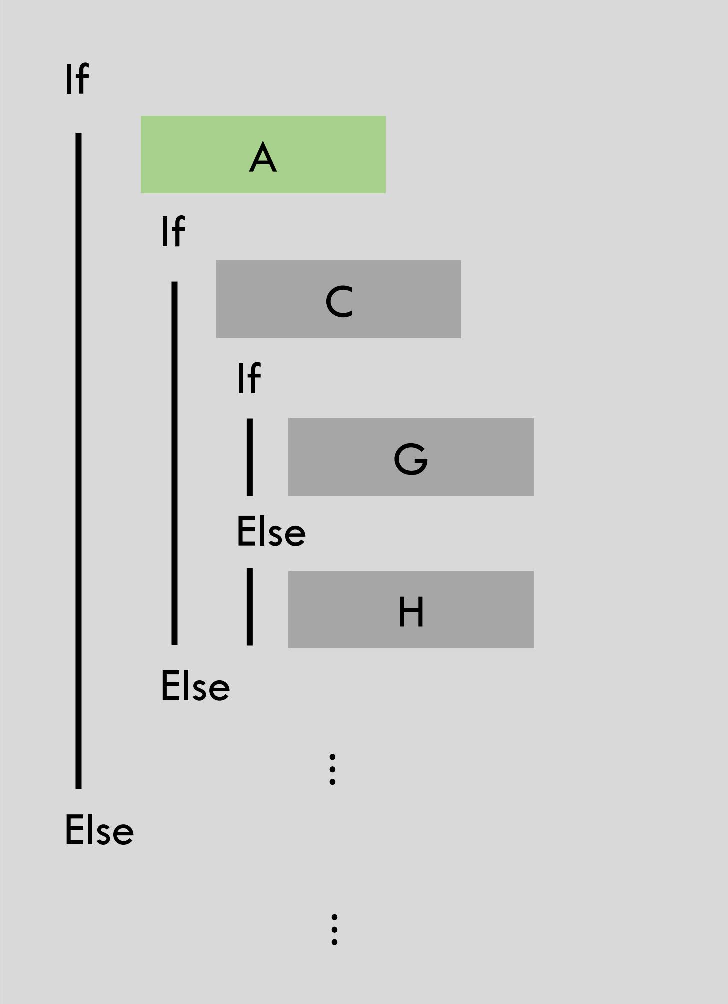


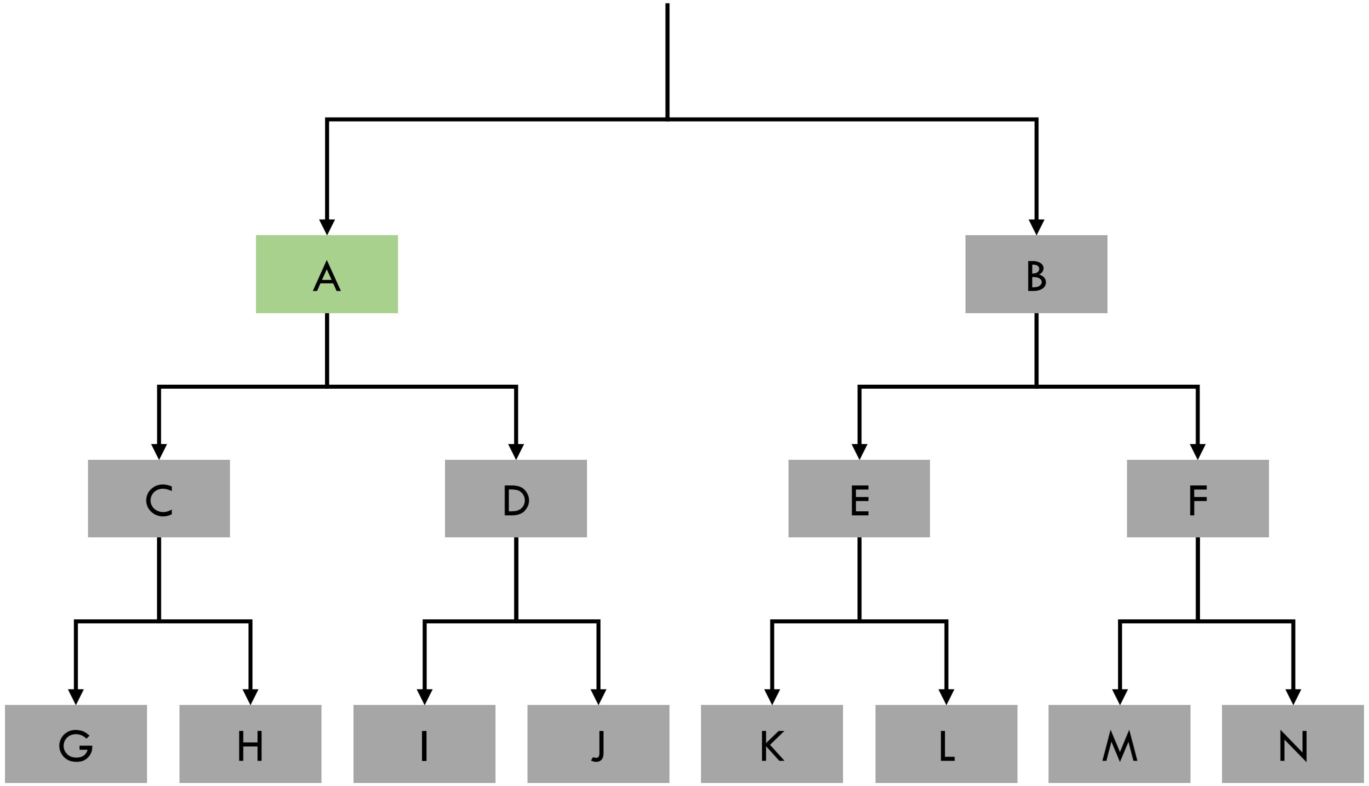


Circuit representation of the program

SNARKs for Nested-Ifs

Example execution on some input

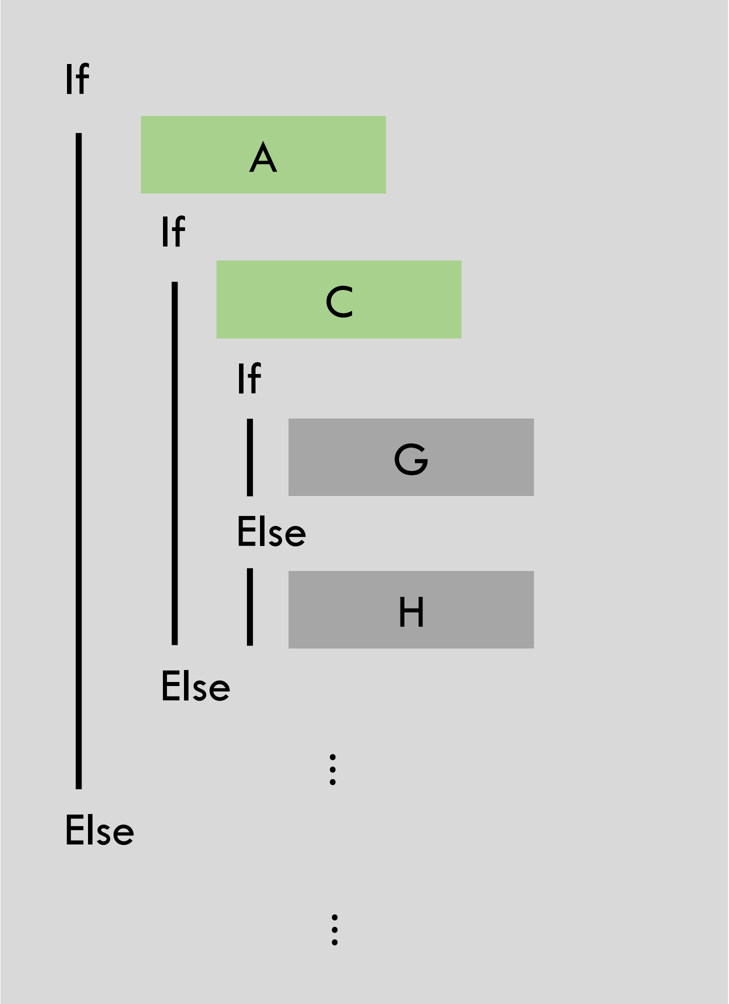


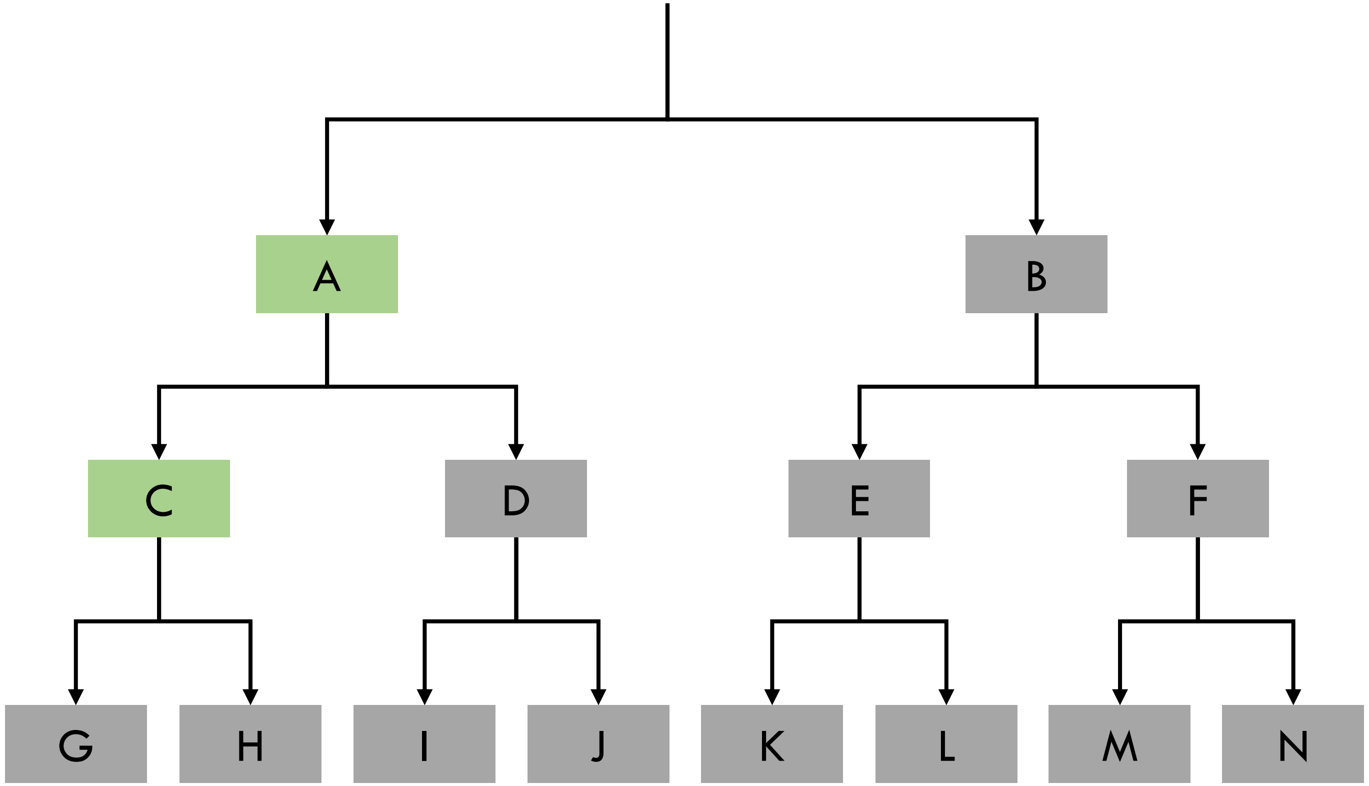


Circuit representation of the program

SNARKs for Nested-Ifs

Example execution on some input

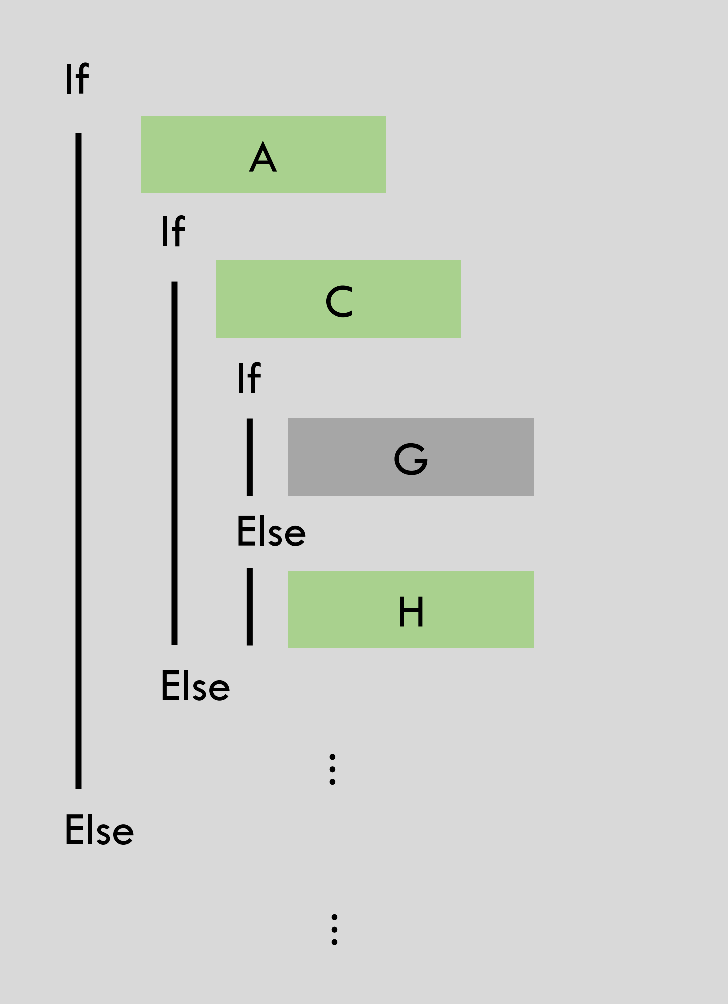


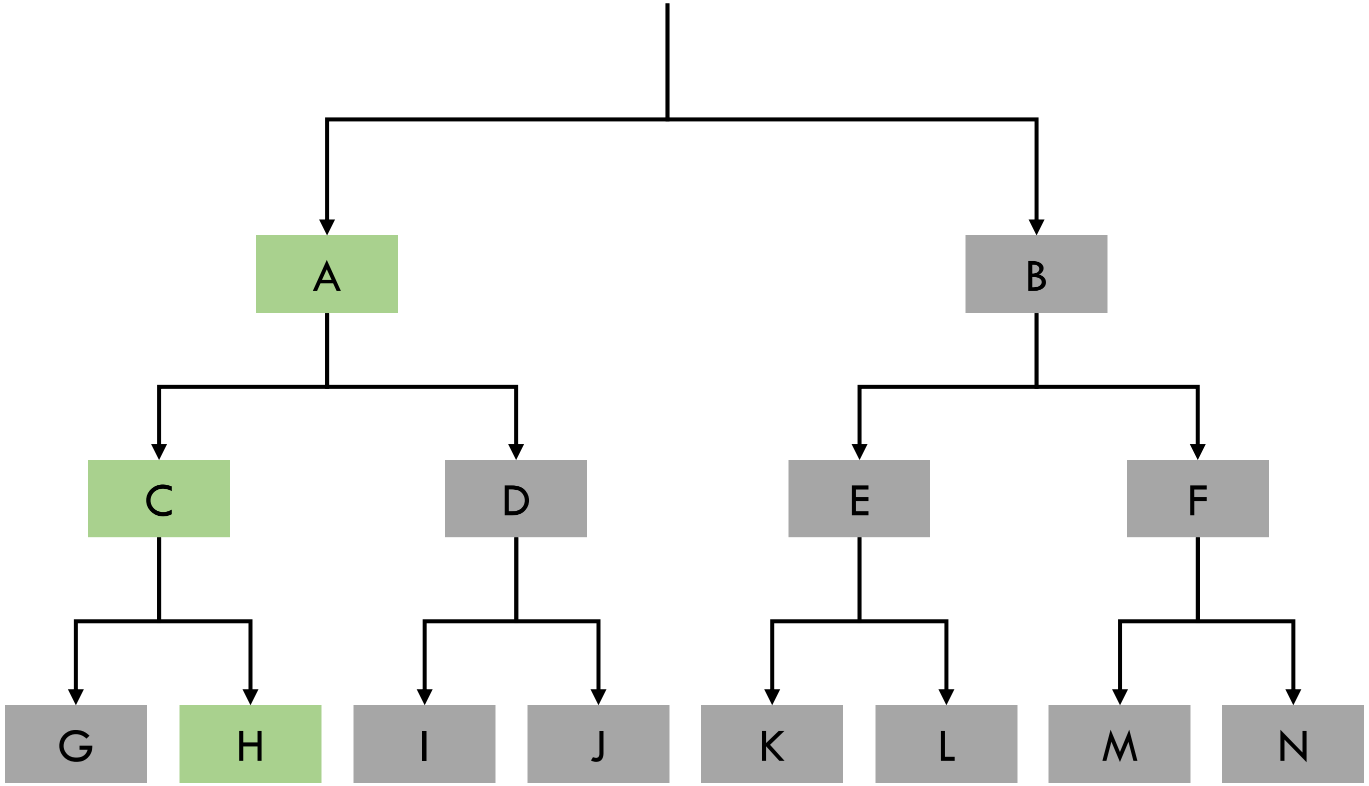


Circuit representation of the program

SNARKs for Nested-Ifs

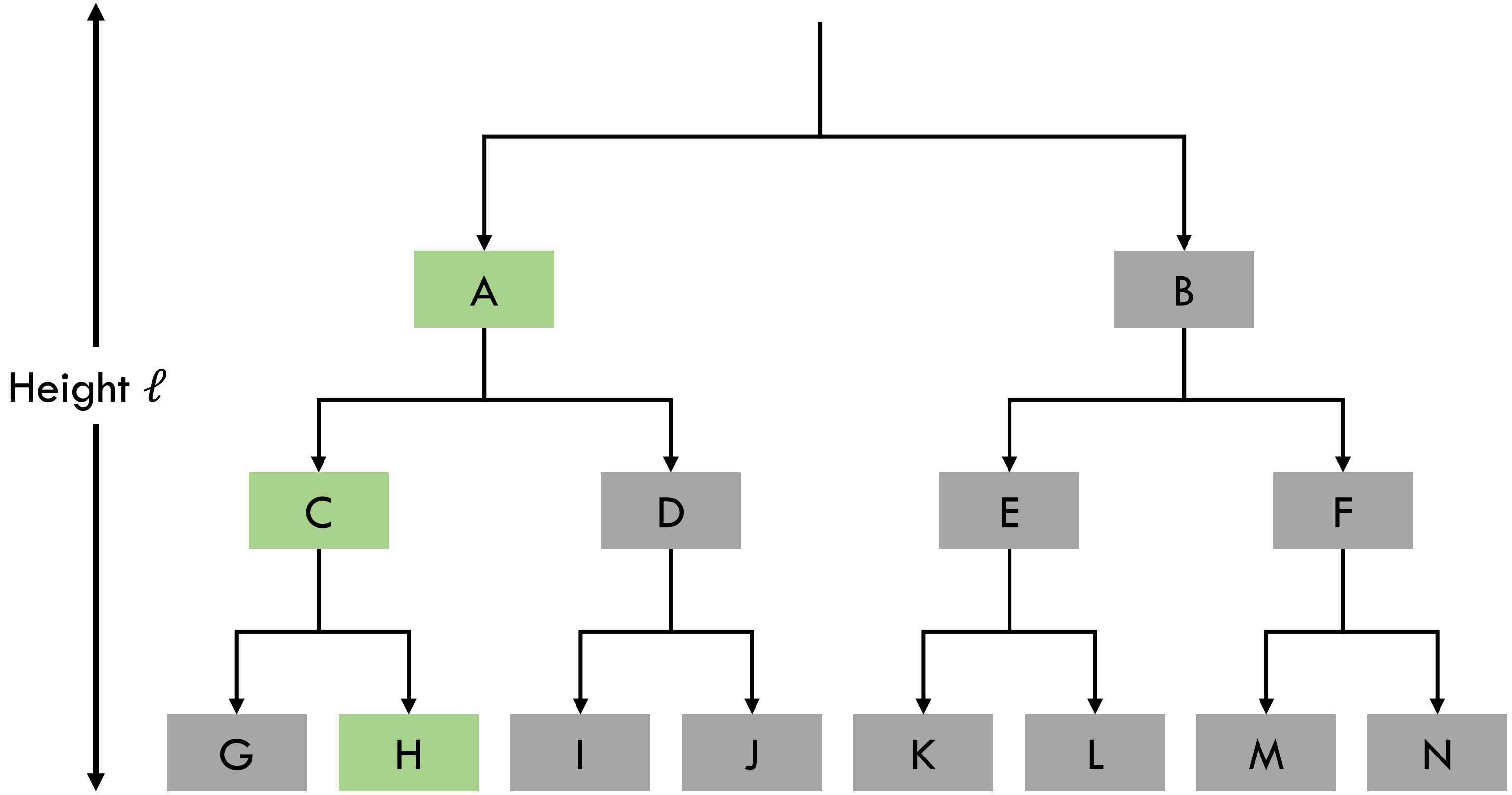
Example execution on some input





Circuit representation of the program

SNARKs for Nested-Ifs

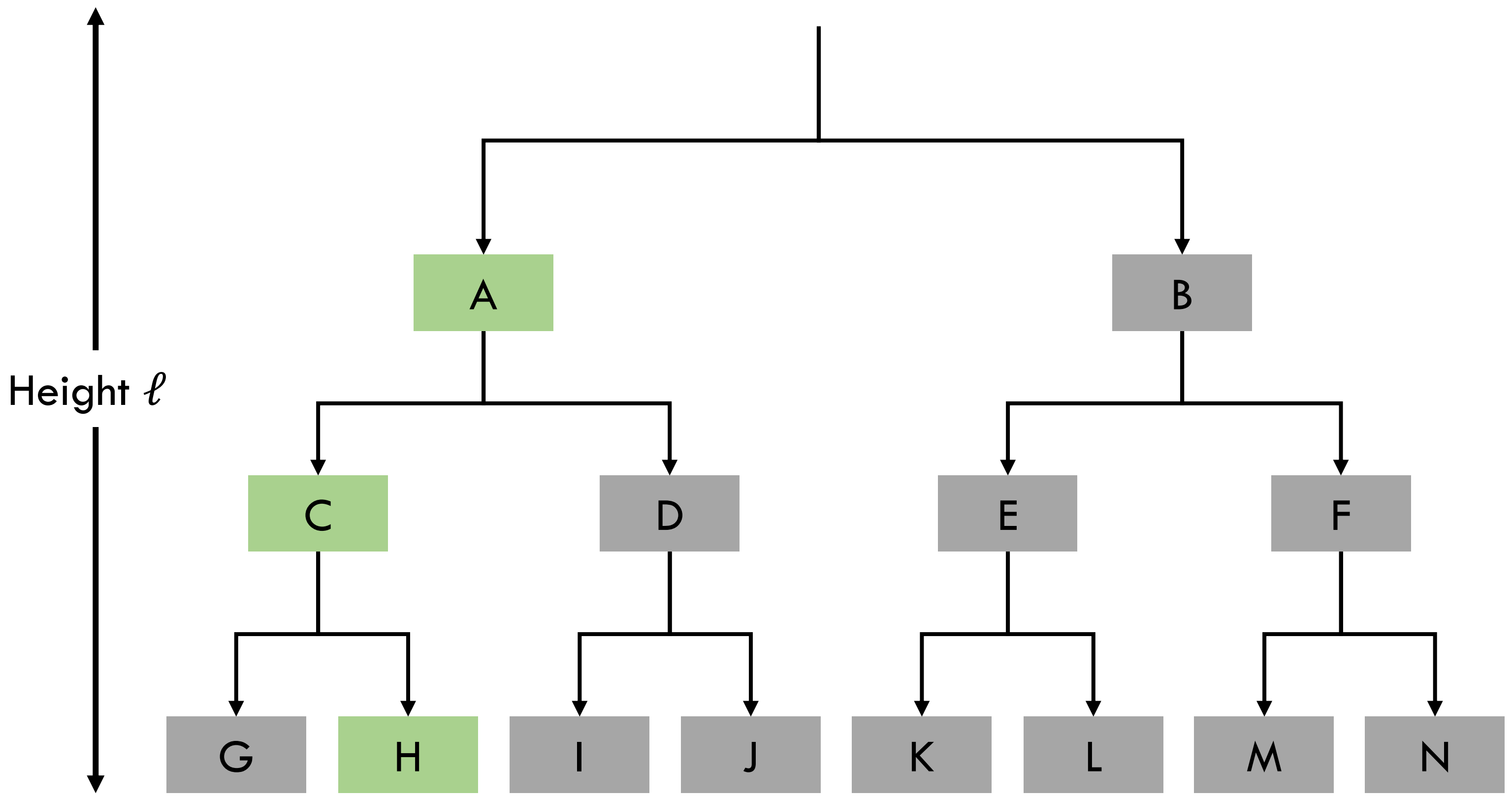


Circuit representation of the program

SNARKs for Nested-Ifs

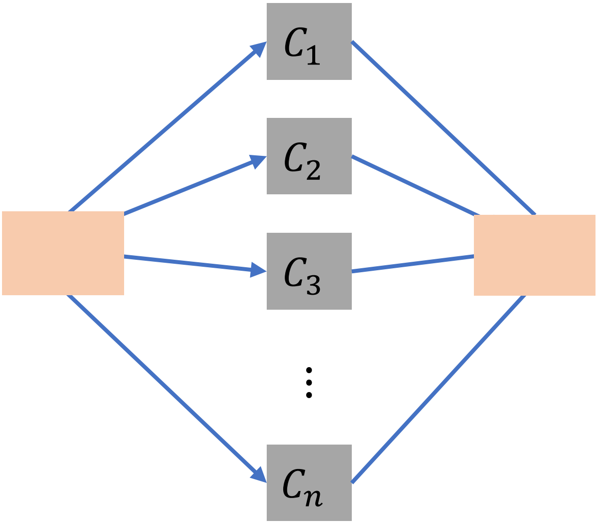
Time to compute SNARK: grows with $O(2^\ell)$

Fraction of active circuit: $\ell/2^\ell$



Circuit representation of the program

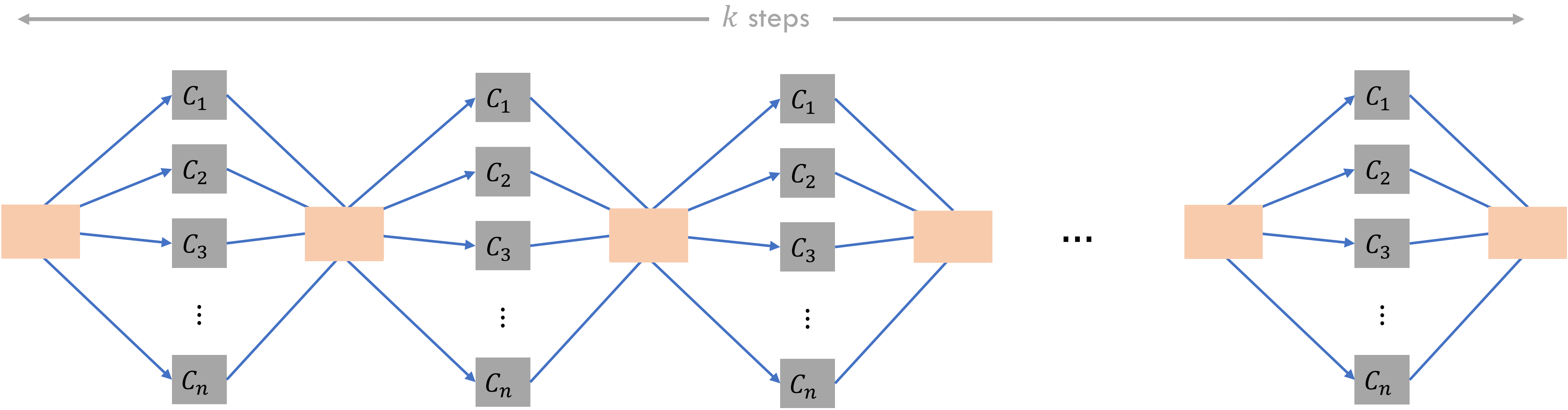
Layered Branching Circuit



Choice on n code segments, each of size s.

Only one code segment active for any input.

Layered Branching Circuit

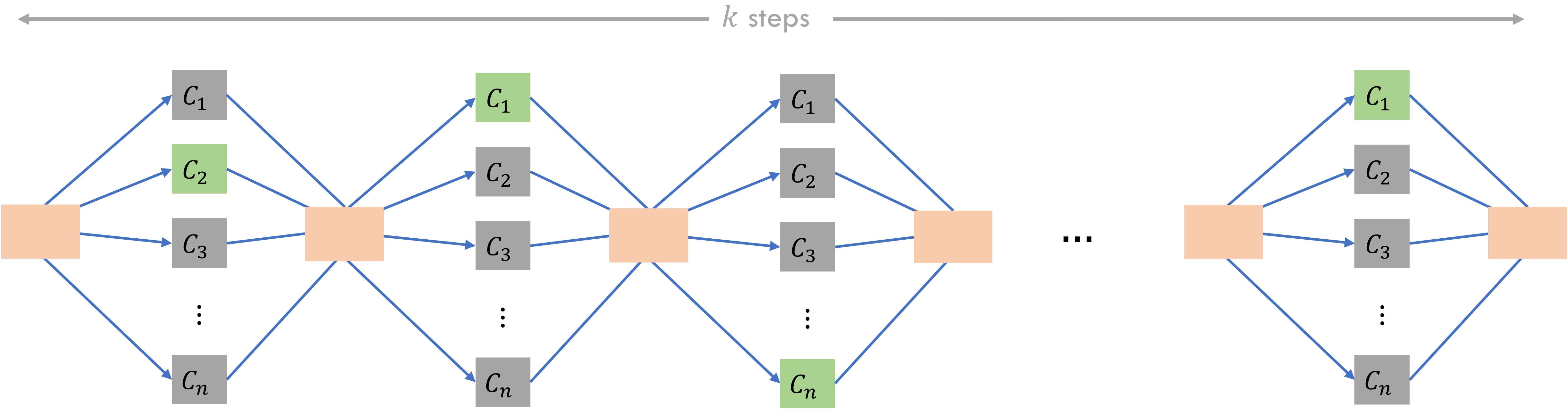


Choice on n code segments, each of size s.

Only one code segment active for any input.

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

Layered Branching Circuit

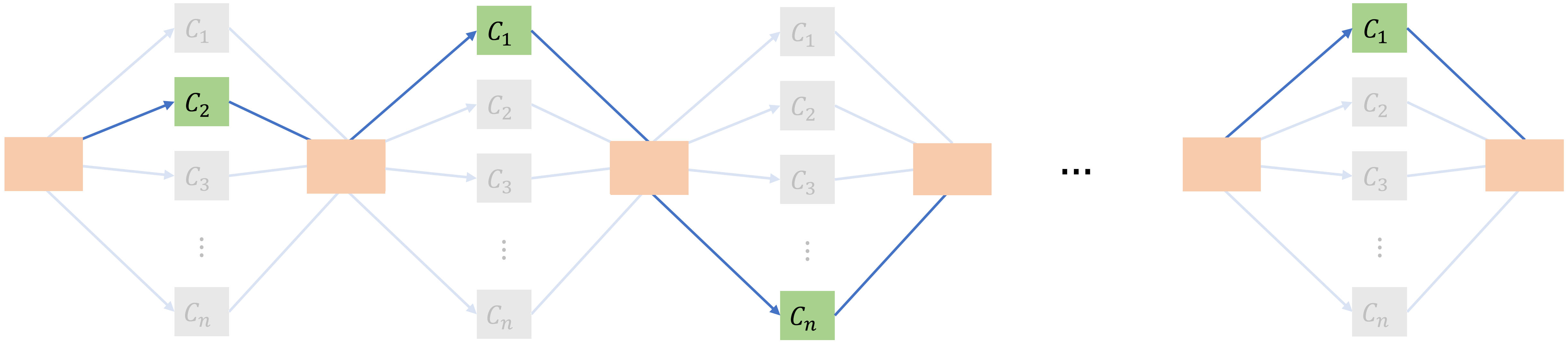


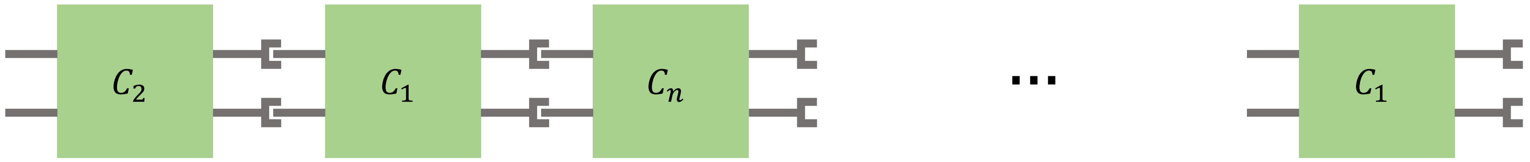
Choice on n code segments, each of size s.

Only one code segment active for any input.

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

Layered Branching Circuit



$\boldsymbol{C_{x,w}} =$ 

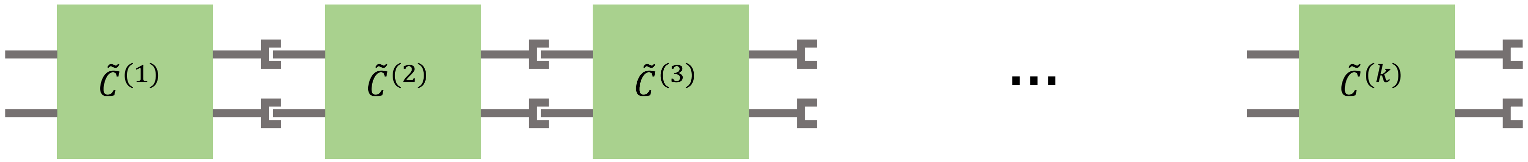
Active Circuit

Size of active circuit

$k \cdot s$

Layered Branching Circuit

Active Circuit for an input x, w

$\boldsymbol{C_{x,w}} =$ 

$j \in [k]$

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

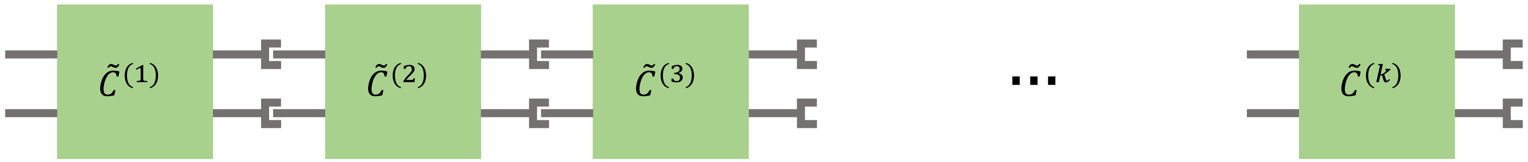
$i \in [n]$

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

Layered Branching Circuit

Also captures the notion of Rollups in Blockchains.

Active Circuit for an input x, w

$\boldsymbol{C_{x,w}} =$ 

$j \in [k]$

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

$i \in [n]$

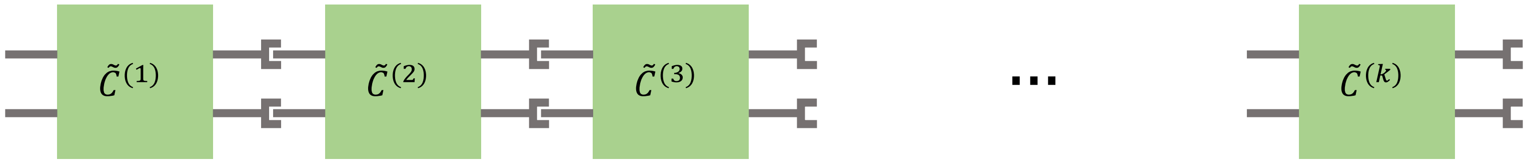
$$|\boldsymbol{C_{x,w}}| = k \cdot s$$

Layered Branching Circuit

Also captures the notion of Rollups in Blockchains.

Active Circuit for an input x, w

$\boldsymbol{C_{x,w}} =$



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

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

Layered Branching Circuit

Active Circuit for an input x, w

$C_{x,w} =$

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

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

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

Prior Works

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]

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]

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]

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]

Prior Works

Not constant proof size

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]

Prior Works

Not constant proof size

Not black-box in Cryptography.

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]

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]

Not constant proof size

Not black-box in Cryptography.

Input dependent prover pre-processing.

Our Result

Theorem: Sublinear Prover $\mathcal{PlonK}$

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.

Our Result

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

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

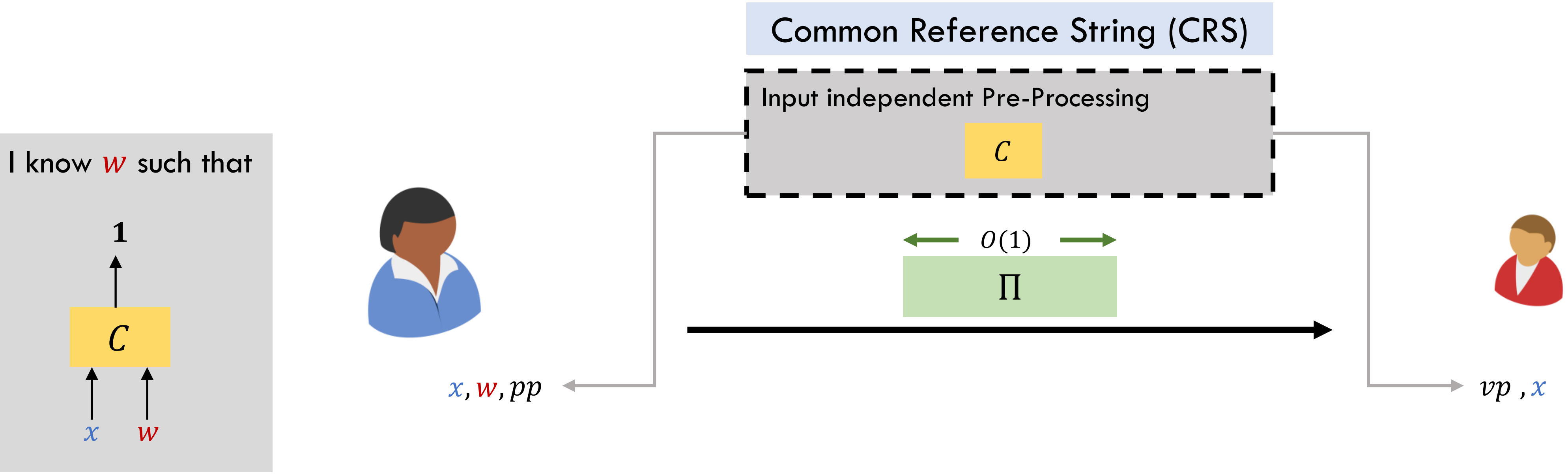
Proof Size - $O(1)$

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 to achieve an almost identical result where the onine prover time is $\tilde{O}((k+n)s)$.

PlonK [Gabizon-Williamson-Ciobotaru'19]

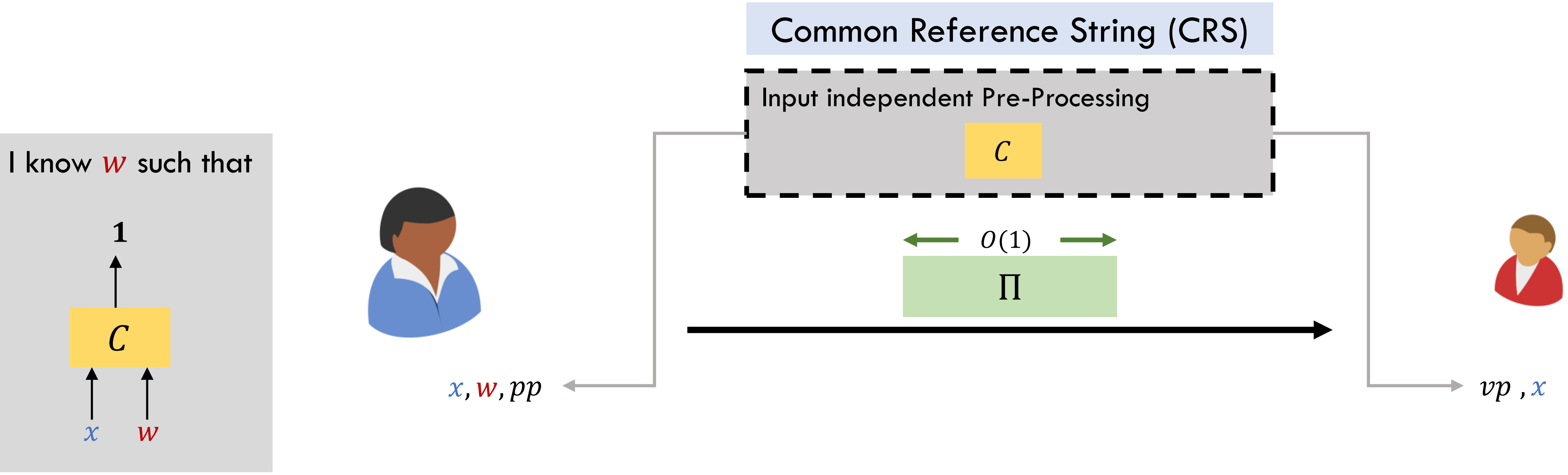


Short proof: $|\Pi| = O(1)$

Prover Time: Grows with $|C|$

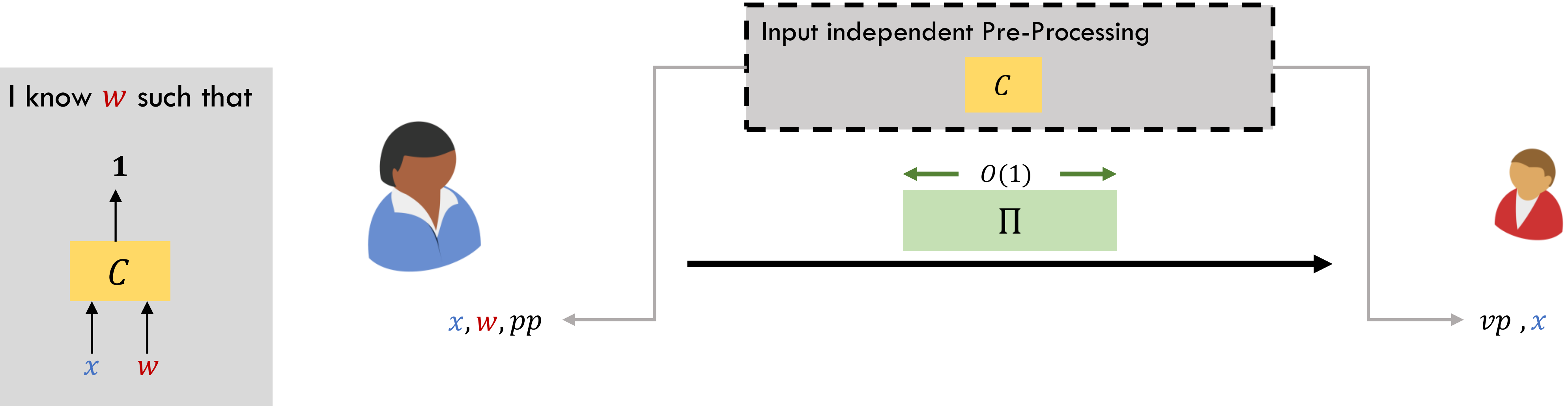
Verification Time $\approx |x| + O(1)$

PlonK [Gabizon-Williamson-Ciobotaru’19]

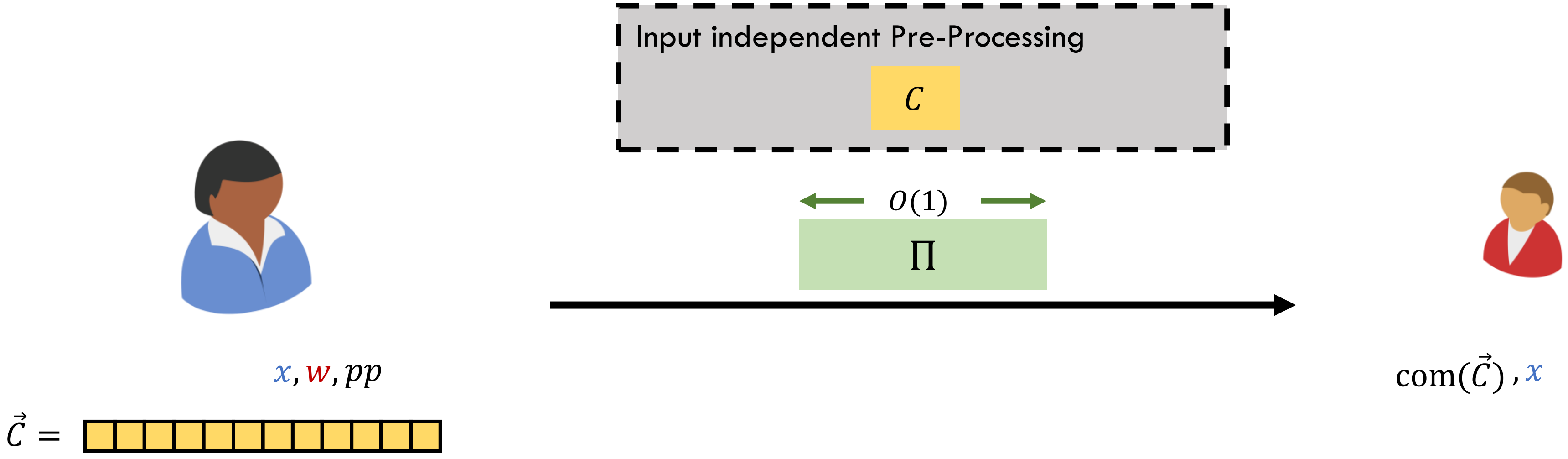


Ignore CRS in this talk for simplicity.

PlonK [Gabizon-Williamson-Ciobotaru'19]

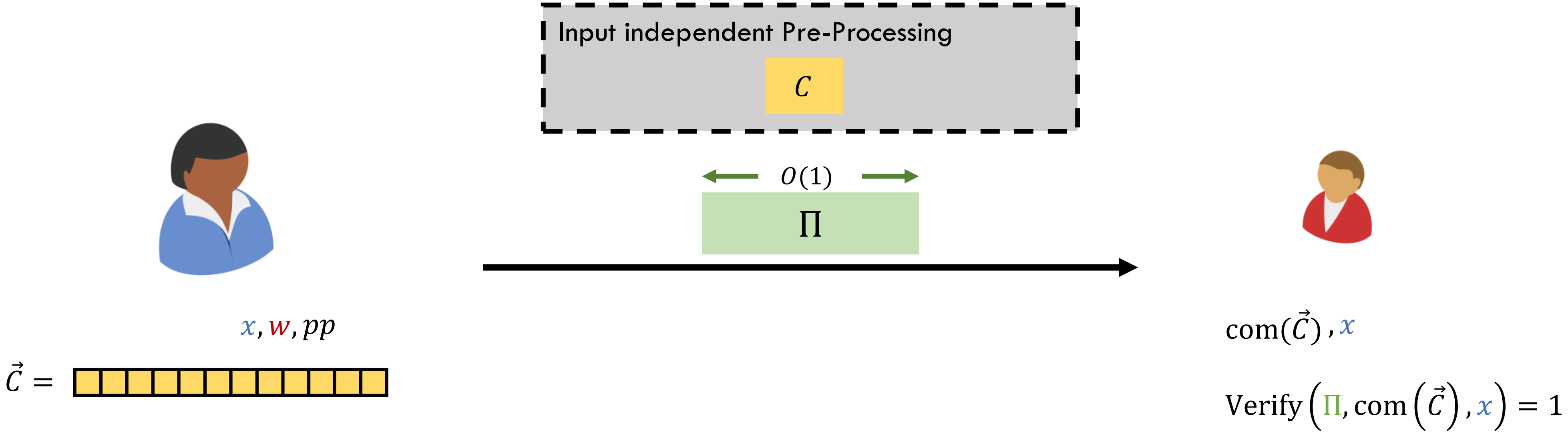


$\mathcal{P}\mathfrak{lon}\mathcal{K}$ [Gabizon-Williamson-Ciobotaru'19]



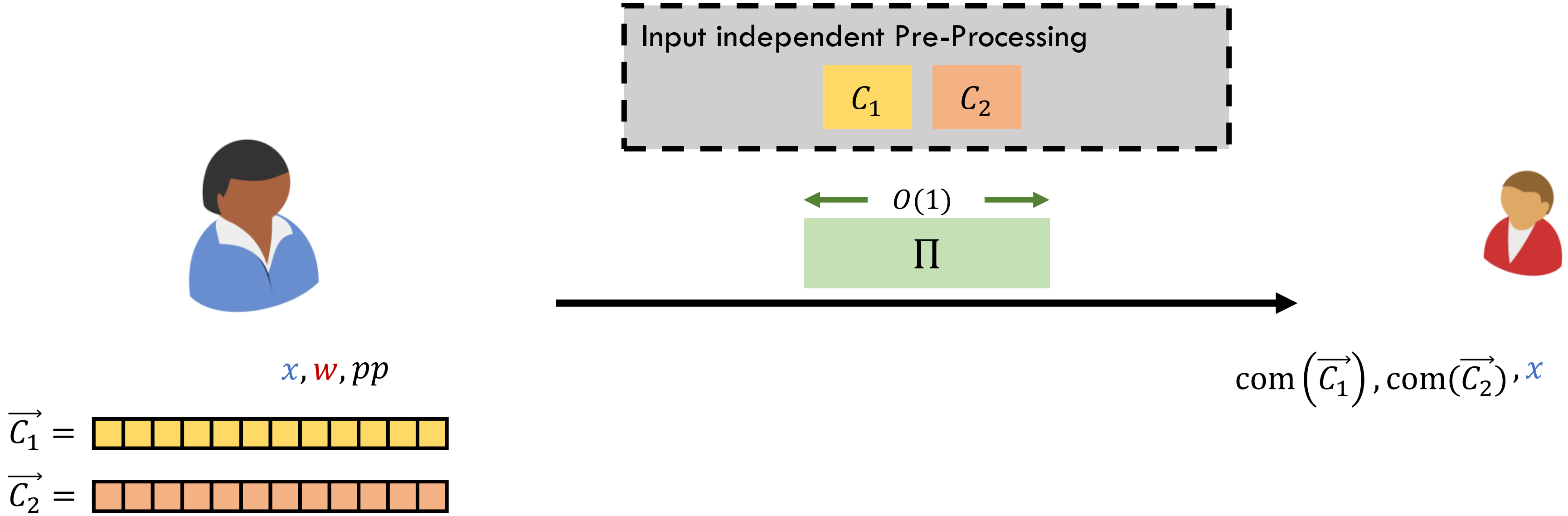
Vector representing circuit constraints.

PlonK [Gabizon-Williamson-Ciobotaru'19]

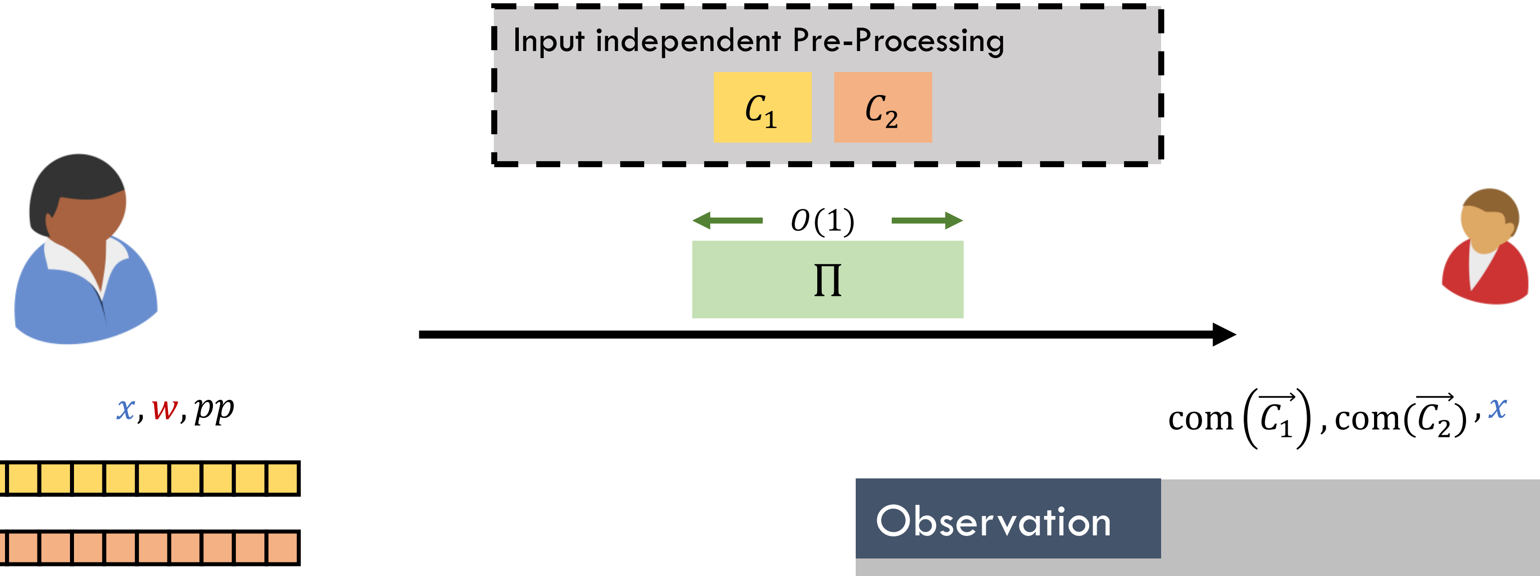


Vector representing circuit constraints.

PlonK [Gabizon-Williamson-Ciobotaru'19]

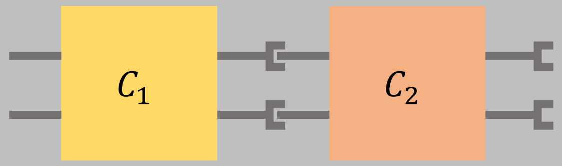


$\mathcal{P}$lon$\mathcal{K}$ [Gabizon-Williamson-Ciobotaru'19]



Observation

$\mathrm{com}\left(\overrightarrow{C_1}\right) \circ \mathrm{com}\left(\overrightarrow{C_2}\right) = \mathrm{com}\left(\overrightarrow{C_1} \middle\| \overrightarrow{C_2}\right)$ is pre-processed commitment* for



Sublinear Prover $\mathcal{P}\mathfrak{lon}\mathcal{K}$

Input independent Pre-Processing

C_1 C_2 $\cdots$ C_n

$\mathrm{com}(C_1), \mathrm{com}(C_2), \ldots, \mathrm{com}(C_n)$

x, w, pp

x

Sublinear Prover $\mathcal{P}\mathfrak{lon}\mathcal{K}$

Input independent Pre-Processing

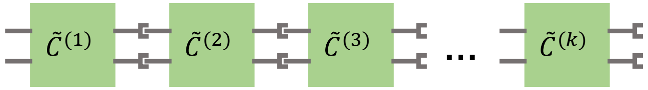
C_1 C_2 $\cdots$ C_n

$\mathrm{com}(C_1), \mathrm{com}(C_2), \ldots, \mathrm{com}(C_n)$

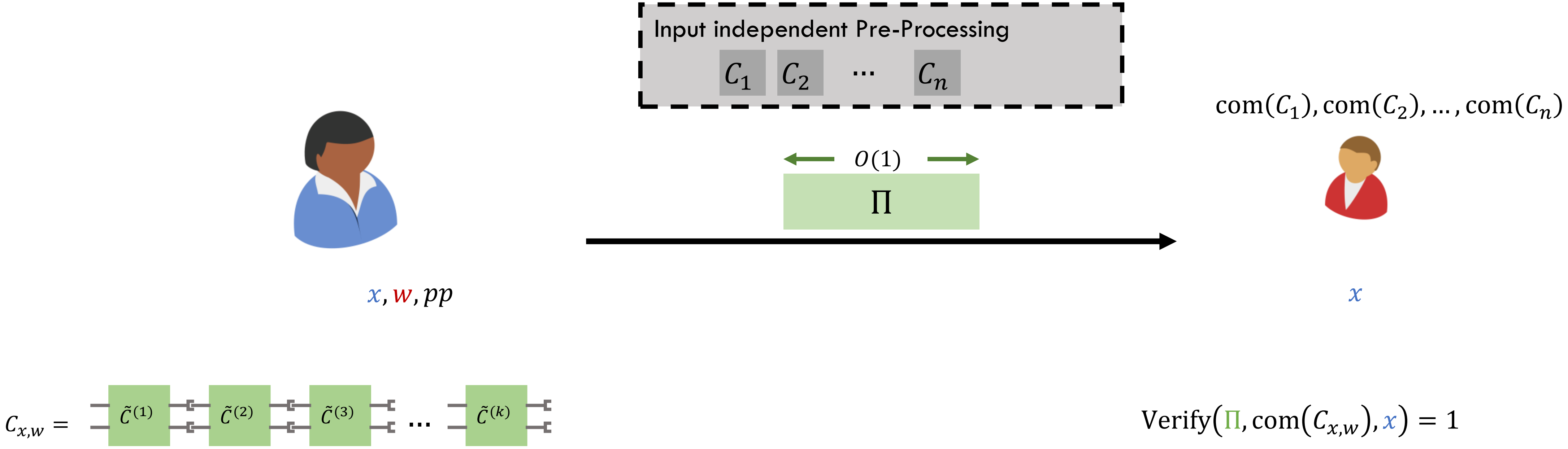
x, w, pp

x

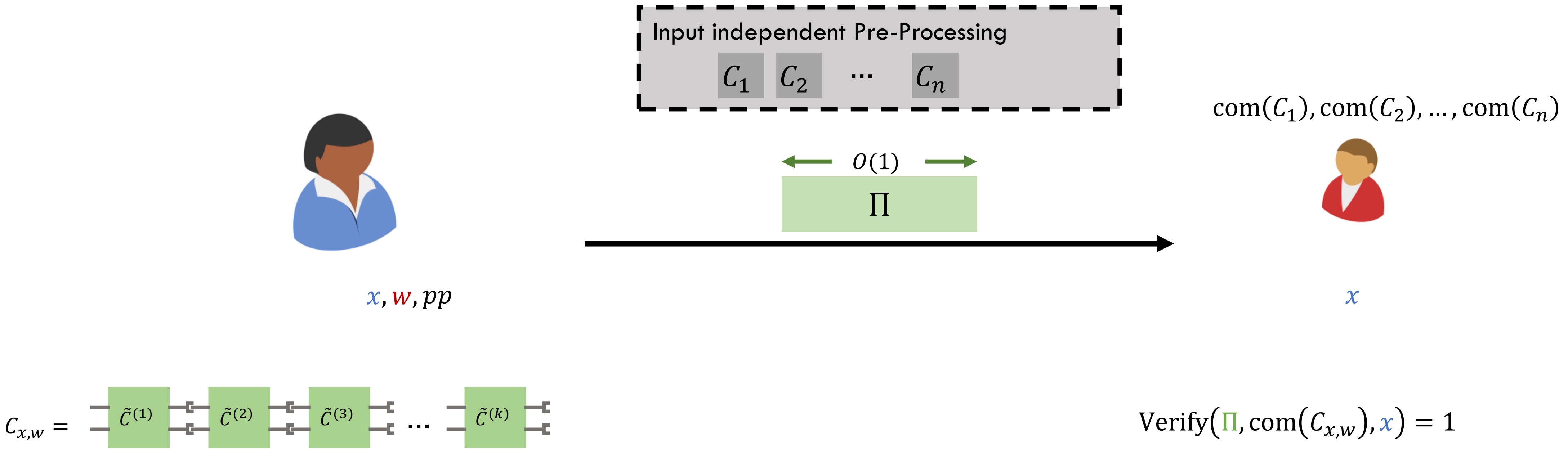
$C_{x,w} =$



Sublinear Prover $\mathcal{P}$lon$\mathcal{K}$

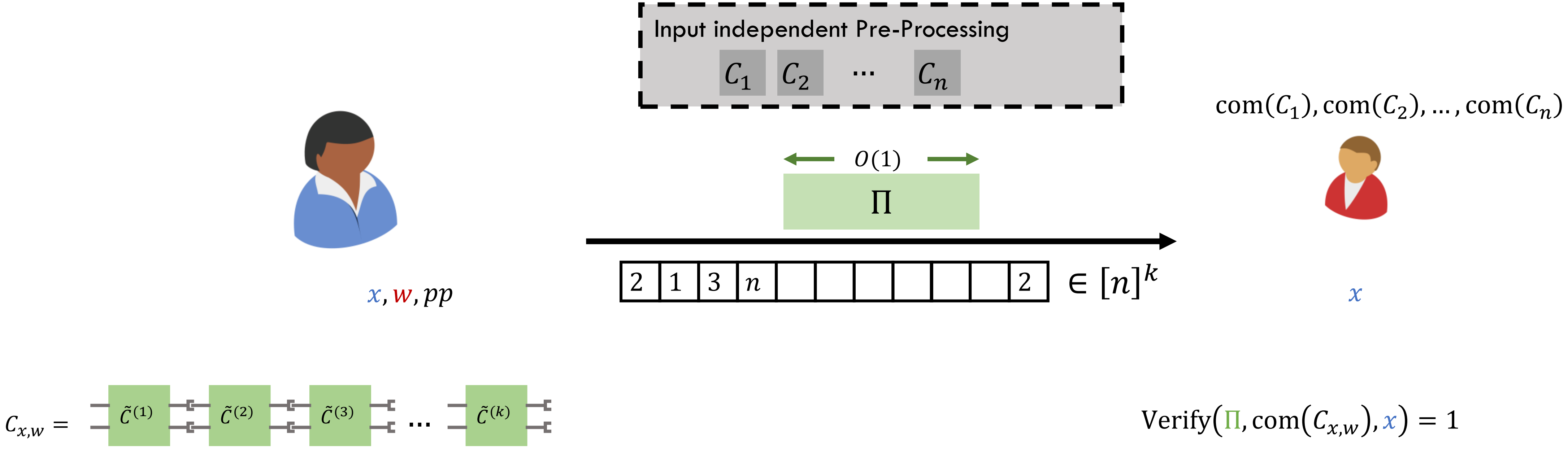


Sublinear Prover $\mathcal{P}\mathfrak{lon}\mathcal{K}$



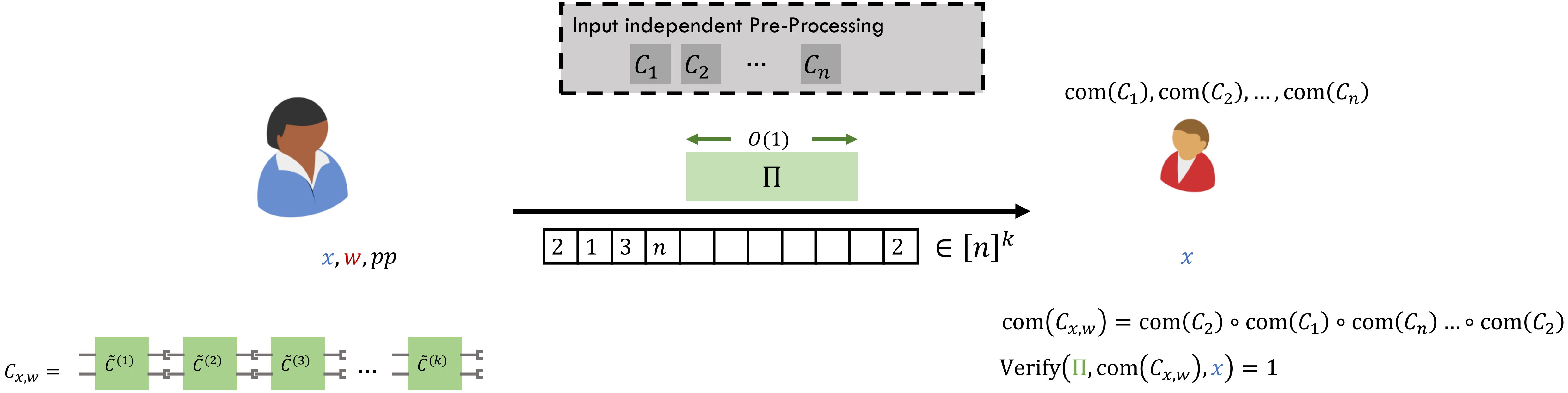
Key Insight: Generate $\mathrm{com}\left(C_{x,w}\right)$ on the fly

Sublinear Prover $\mathcal{P}$lon$\mathcal{K}$



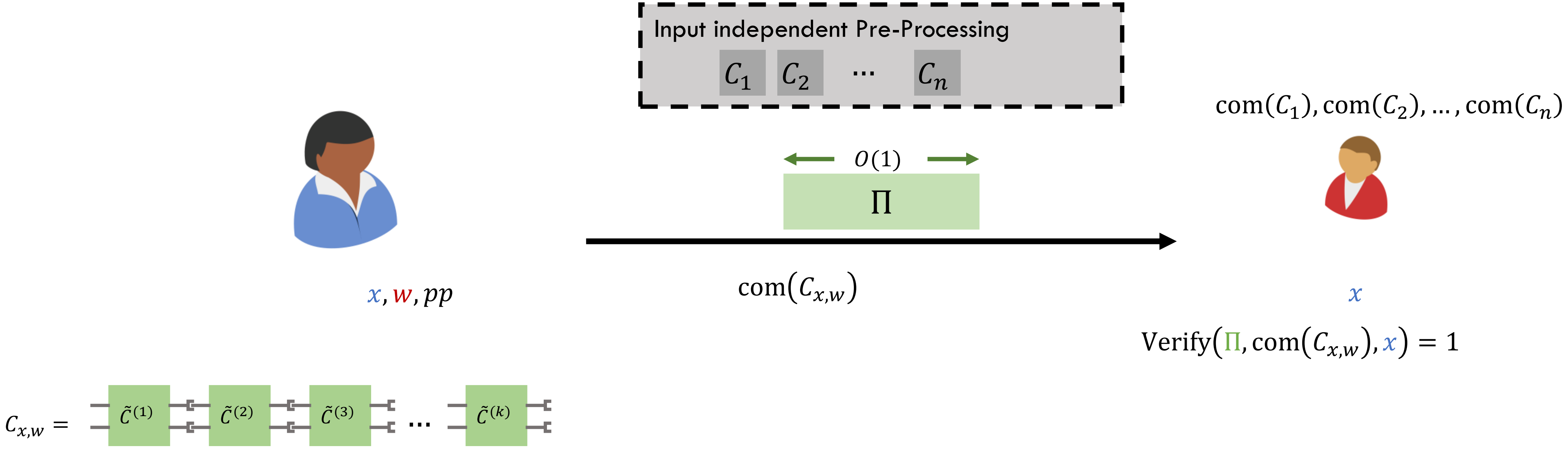
Key Insight: Generate $\mathrm{com}\left(C_{x,w}\right)$ on the fly

Sublinear Prover $\mathcal{PlonK}$

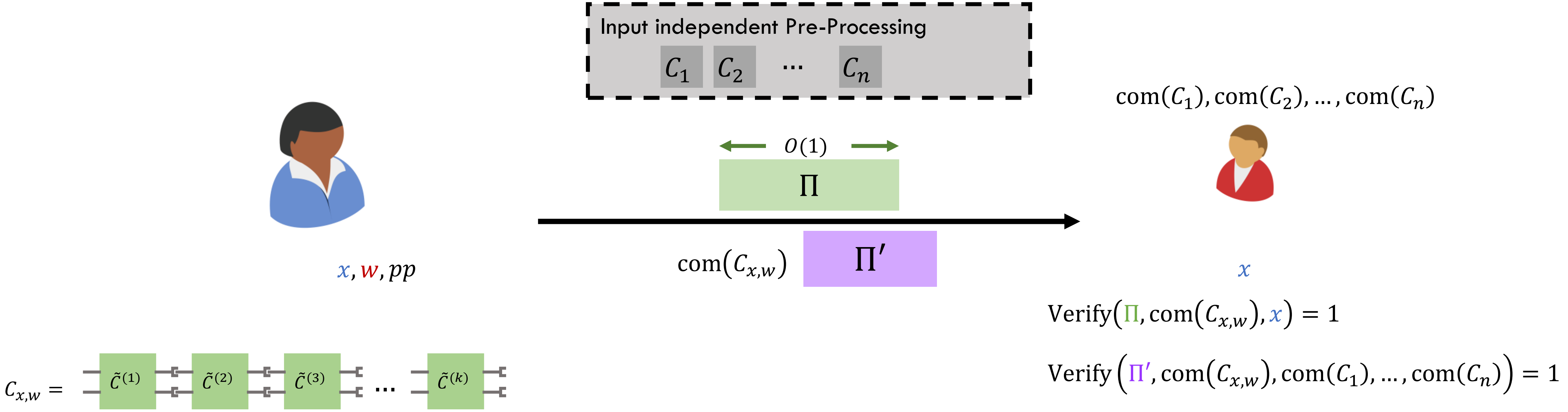


Key Insight: Generate $\mathrm{com}(C_{x,w})$ on the fly

Sublinear Prover $\mathcal{P}\mathfrak{lon}\mathcal{K}$



Sublinear Prover $\mathcal{P}$lon$\mathcal{K}$



Sublinear Prover $\mathcal{P}\mathfrak{lon}\mathcal{K}$

Input independent Pre-Processing

C_1 C_2 $\cdots$ C_n

x, w, pp

$\longleftarrow O(1) \longrightarrow$

Π

$\mathrm{com}(C_{x,w})$ Π'

$\mathrm{com}(C_1), \mathrm{com}(C_2), \ldots, \mathrm{com}(C_n)$

x

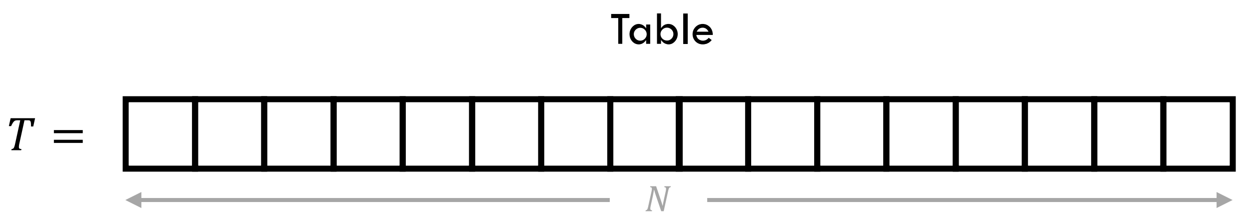
$\mathrm{Verify}\big(\Pi, \mathrm{com}(C_{x,w}), x\big) = 1$

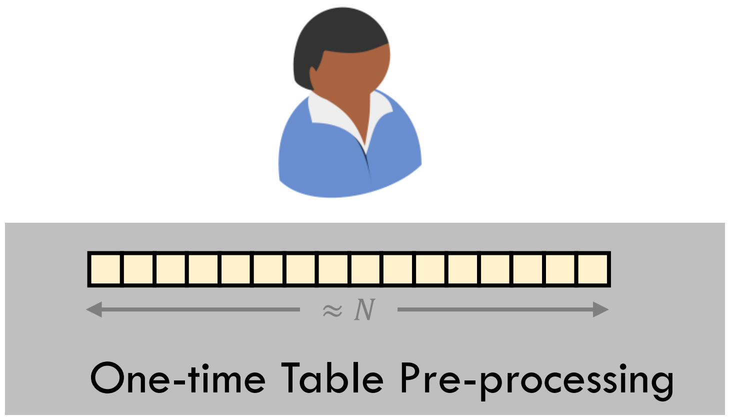
$\mathrm{Verify}\Big(\Pi', \mathrm{com}(C_{x,w}), \mathrm{com}(C_1), \ldots, \mathrm{com}(C_n)\Big) = 1$

$C_{x,w} = \tilde{C}^{(1)} \; \tilde{C}^{(2)} \; \tilde{C}^{(3)} \cdots \tilde{C}^{(k)}$

Tool to generate Π': Table Lookups

Table Lookup





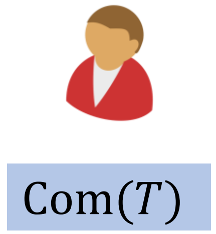
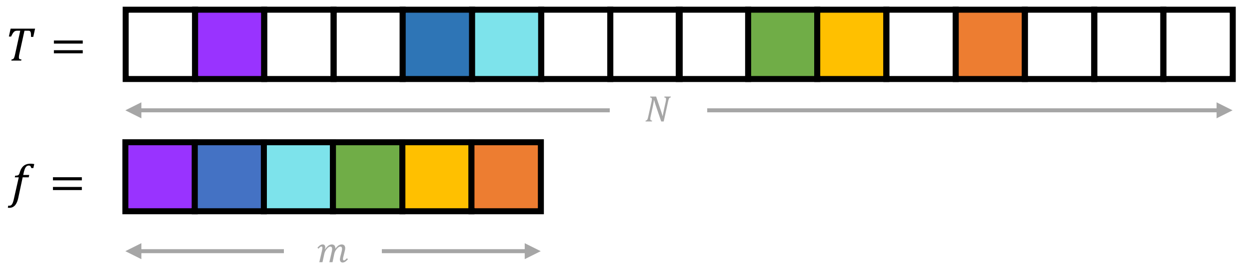
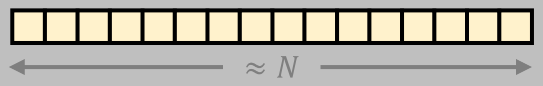


Table Lookup

Table

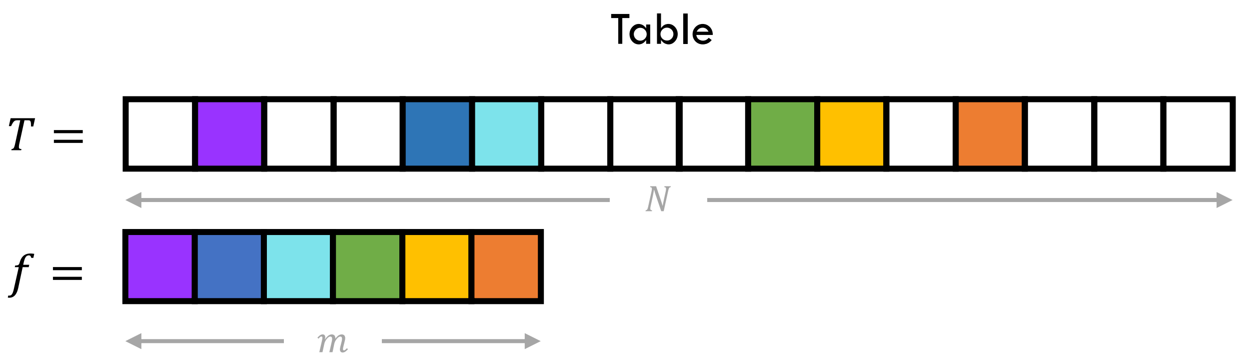




One-time Table Pre-processing

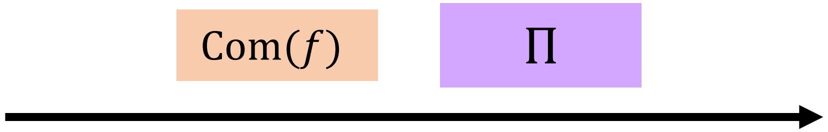
$\mathrm{Com}(T)$

Table Lookup



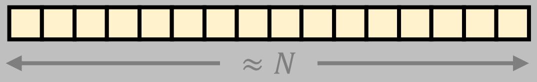
$\forall i \in [m], \exists j \in [N]$ such that

$$f[i] = T[j]$$



$\mathrm{Com}(T)$

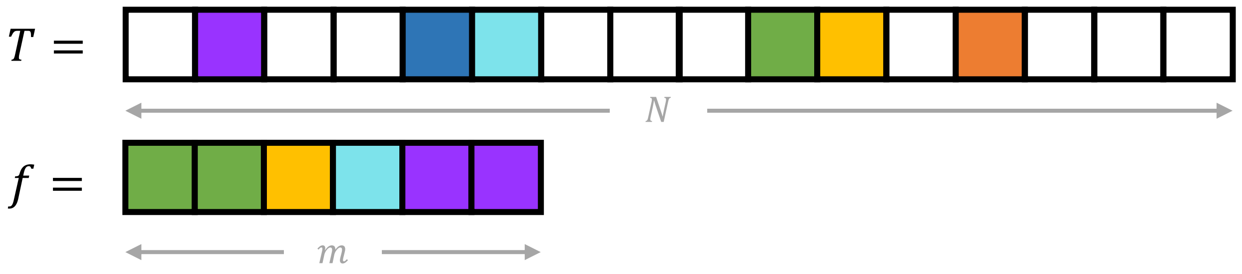
$\mathrm{Verify}\big(\Pi, \mathrm{com}(f), \mathrm{com}(T)\big) = 1$



One-time Table Pre-processing

Table Lookup

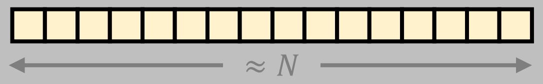
Table



$\forall i \in [m], \exists j \in [N]$ such that

$$f[i] = T[j]$$

Allows out of order and repetitions.



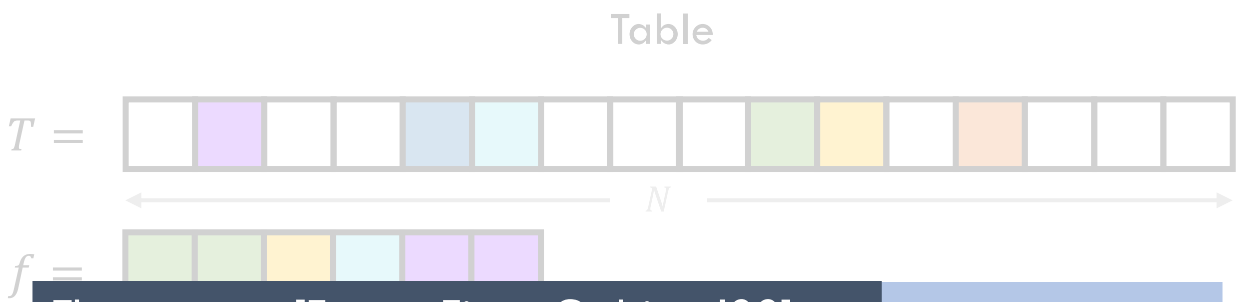
One-time Table Pre-processing

$\mathrm{Com}(f)$ Π

$\mathrm{Com}(T)$

$\mathrm{Verify}(\Pi, \mathrm{com}(f), \mathrm{com}(T)) = 1$

Table Lookup

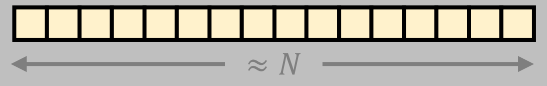


Theorem: cq [Eagen-Fiore-Gabizon'22]

Online Prover Time - $O(m \log m)$

Proof Size* - $O(1)$

Black-box in Cryptography



One-time Table Pre-processing

$\forall i \in [m], \exists j \in [N]$ such that

$$f[i] = T[j]$$

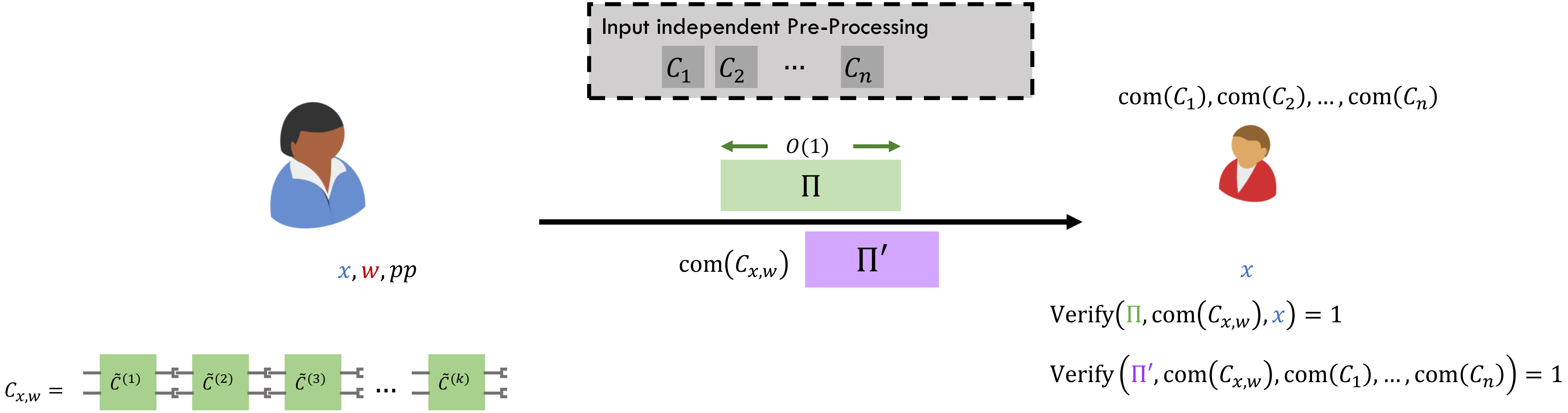
Allows out of order and repetitions.

$\mathrm{Com}(T)$

$\mathrm{Verify}\big(\Pi, \mathrm{com}(f), \mathrm{com}(T)\big) = 1$

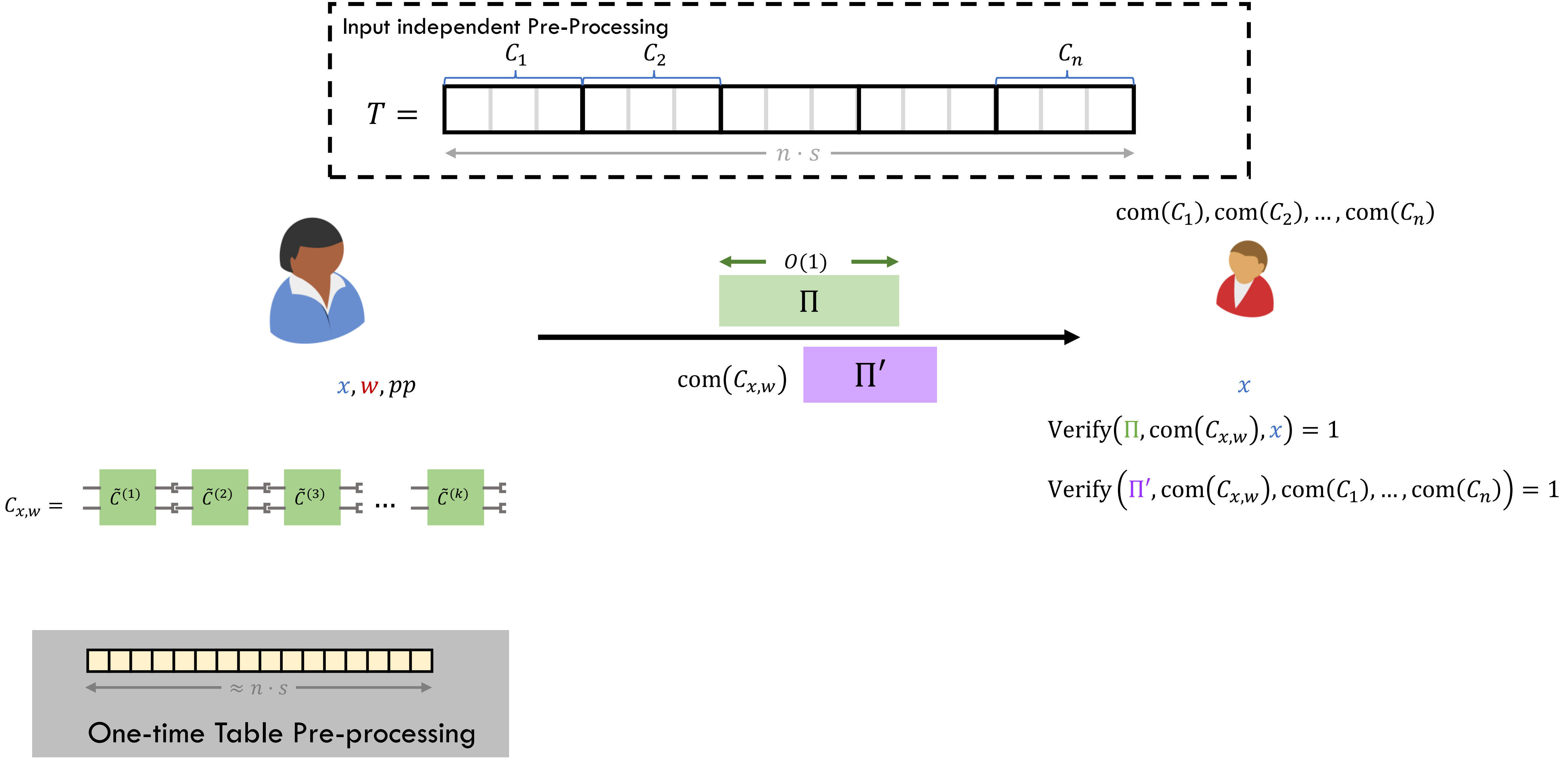
*Proof Size includes size of $\mathrm{com}(f)$

Sublinear Prover $\mathcal{P}\mathfrak{lon}\mathcal{K}$

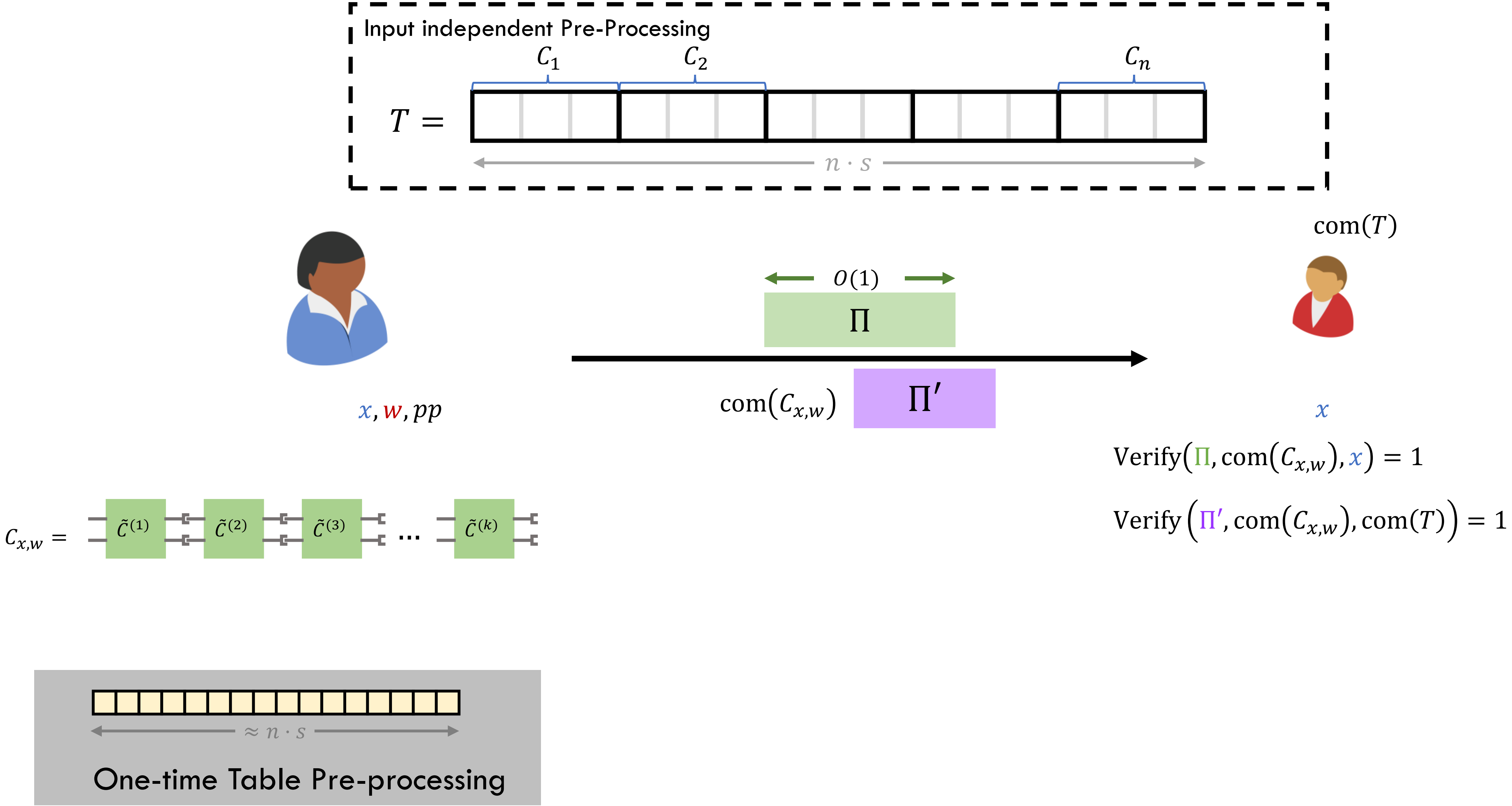


Tool to generate Π': Table Lookups

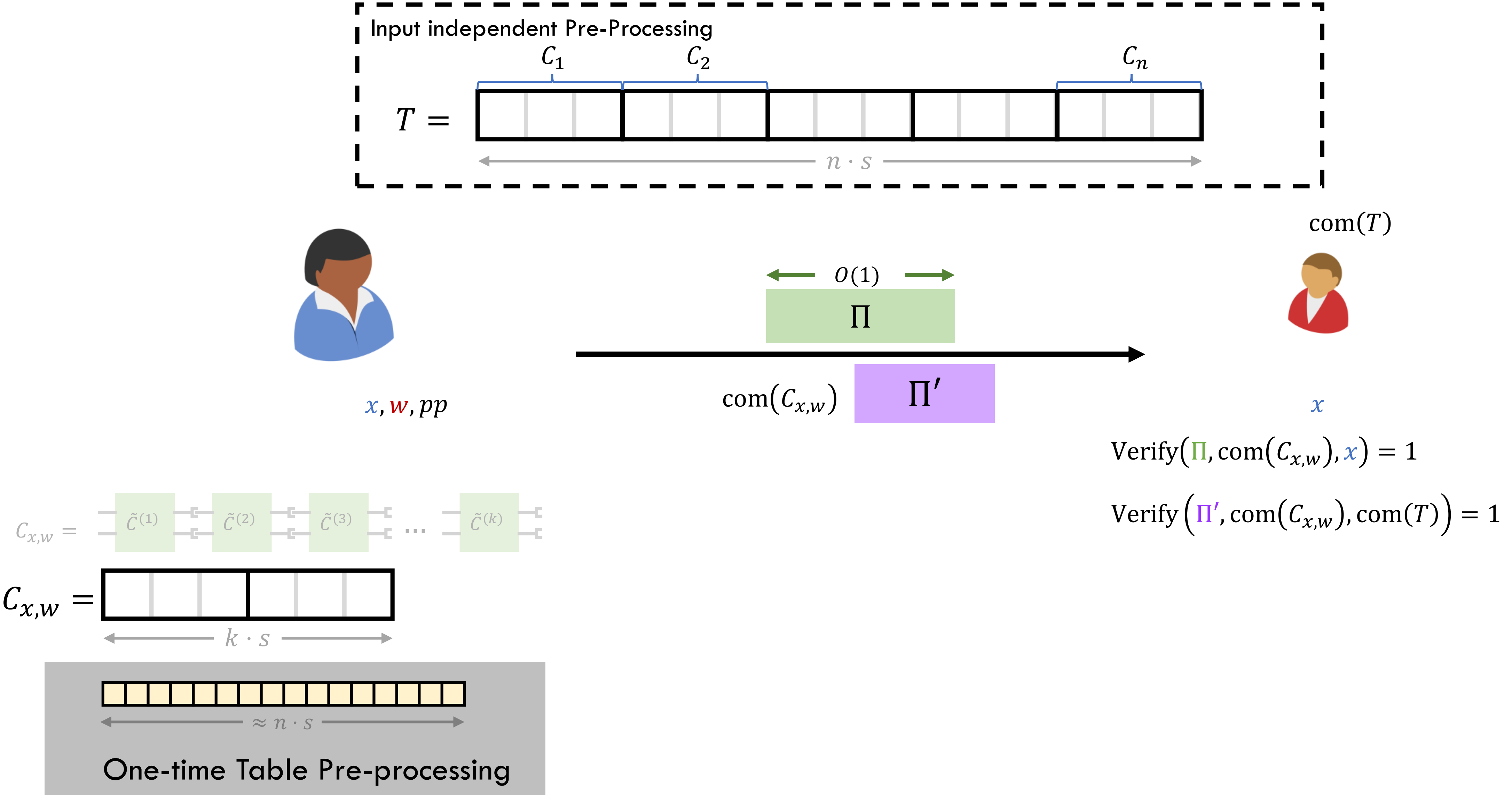
Sublinear Prover $\mathcal{PlonK}$



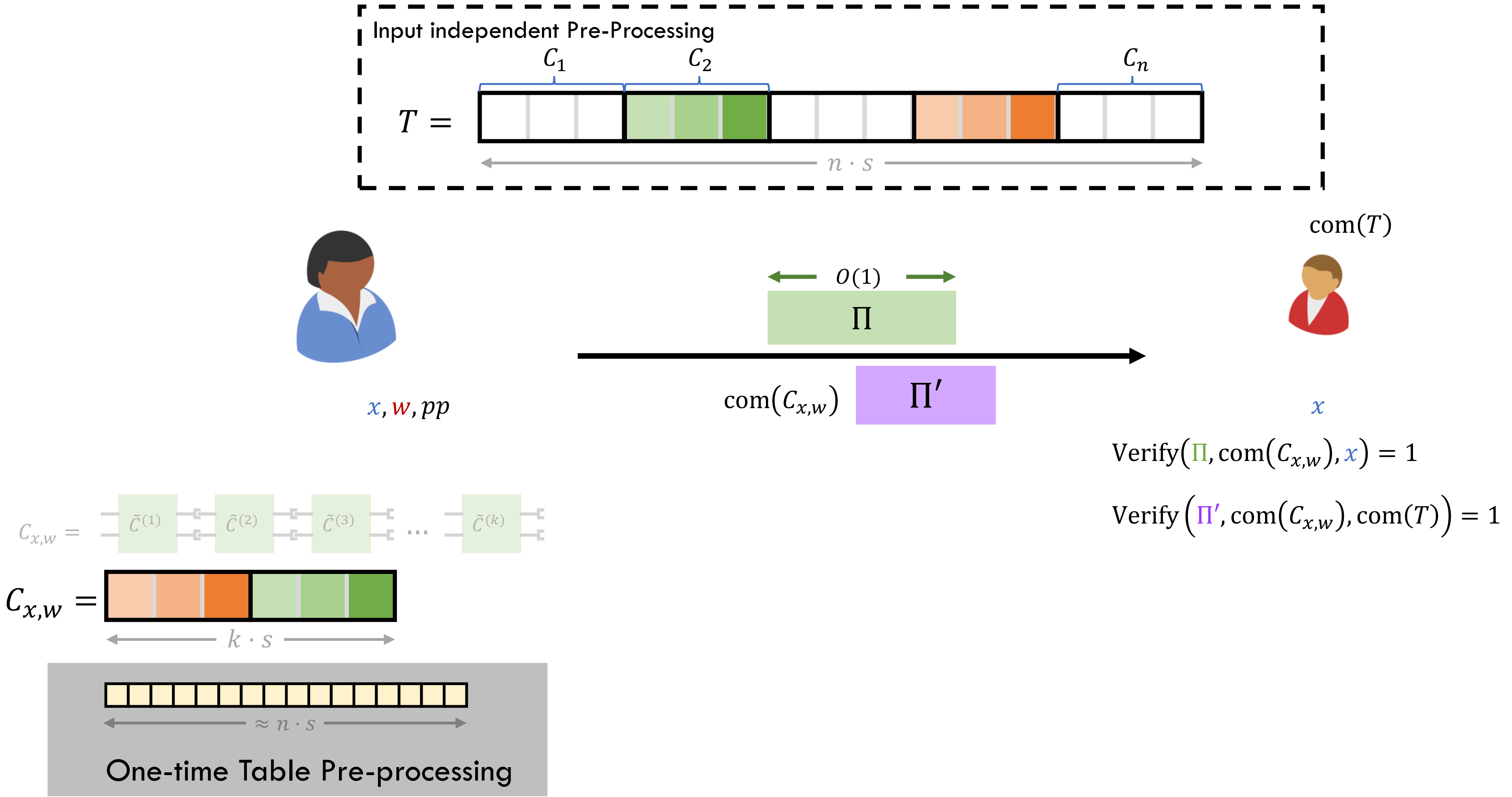
Sublinear Prover PlonK



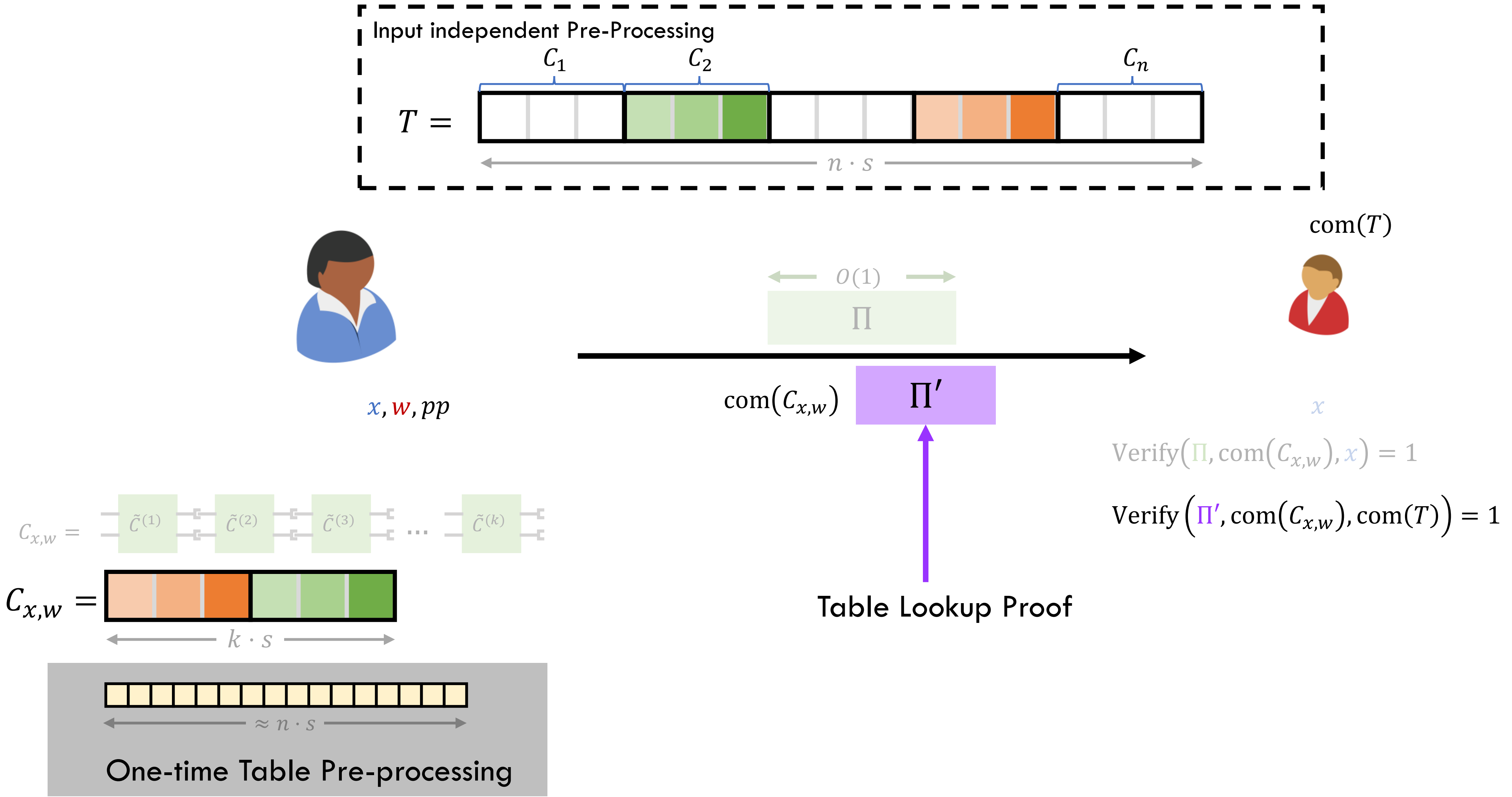
Sublinear Prover $\mathcal{P}lon\mathcal{K}$



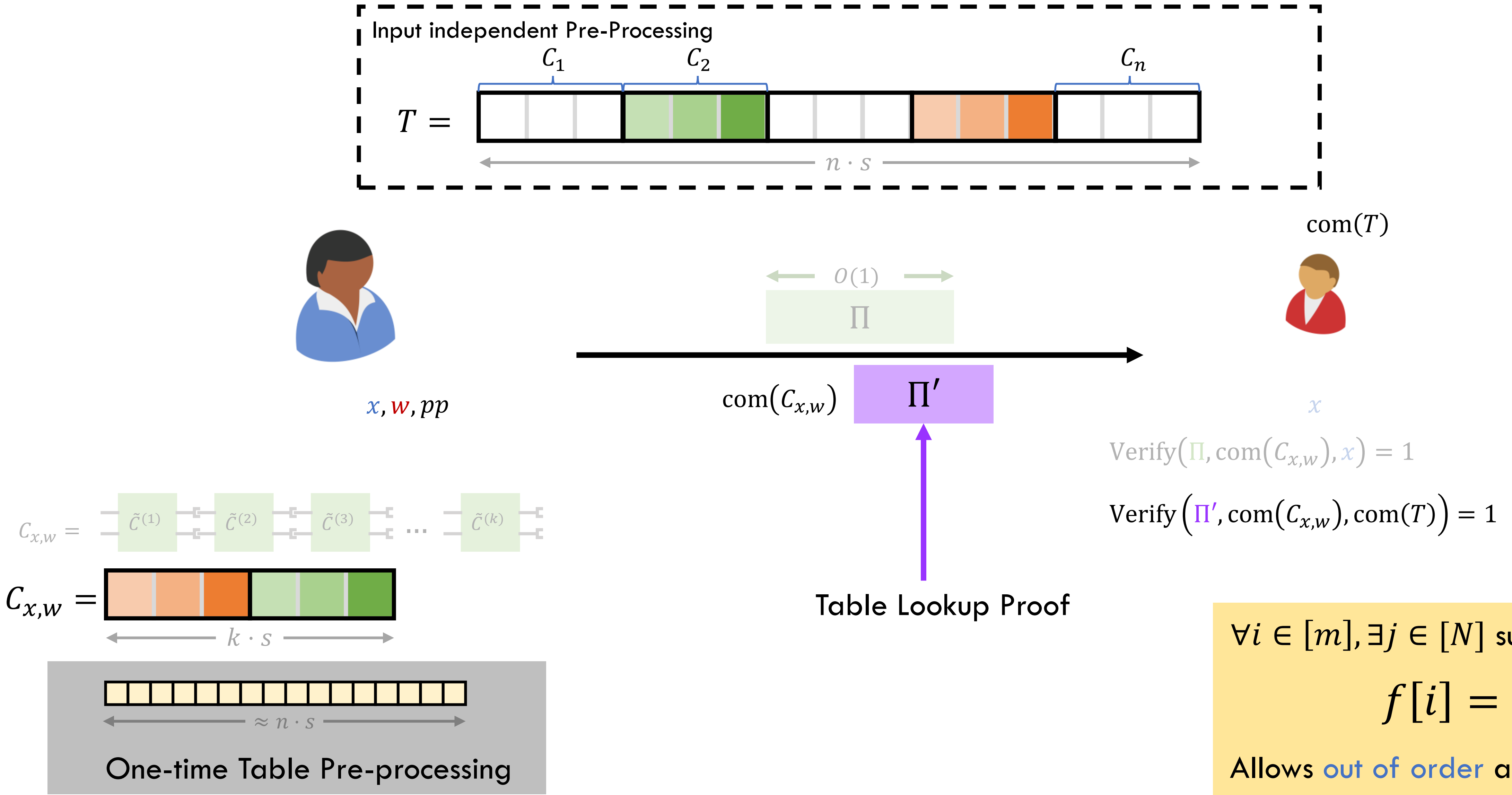
Sublinear Prover $\mathcal{P}$lon$\mathcal{K}$



Sublinear Prover $\mathcal{P}\mathfrak{lon}\mathcal{K}$



Sublinear Prover $\mathcal{P}\mathfrak{lon}\mathcal{K}$

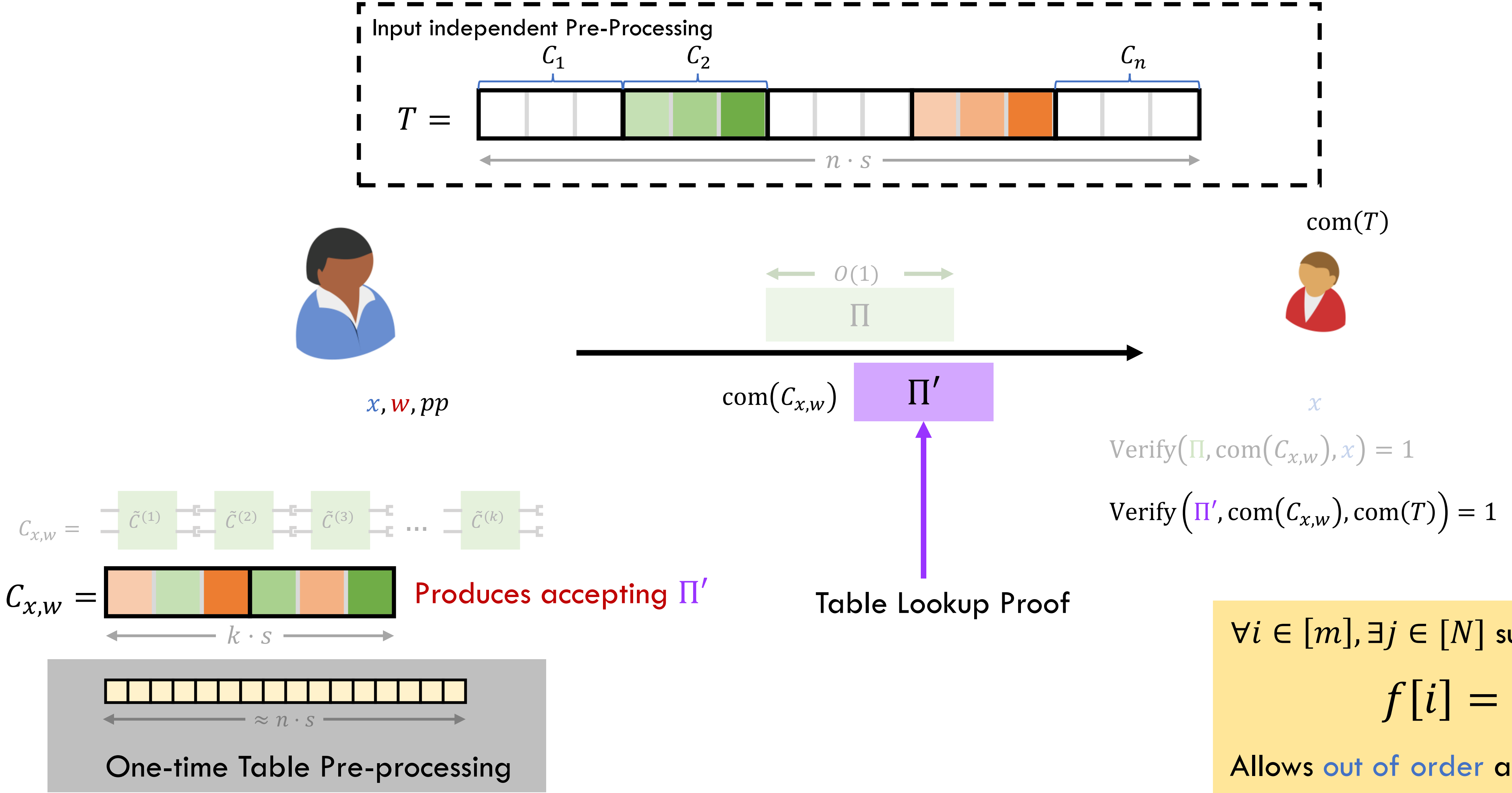


$\forall i \in [m], \exists j \in [N]$ such that

$$f[i] = T[j]$$

Allows out of order and repetitions.

Sublinear Prover $\mathcal{PlonK}$



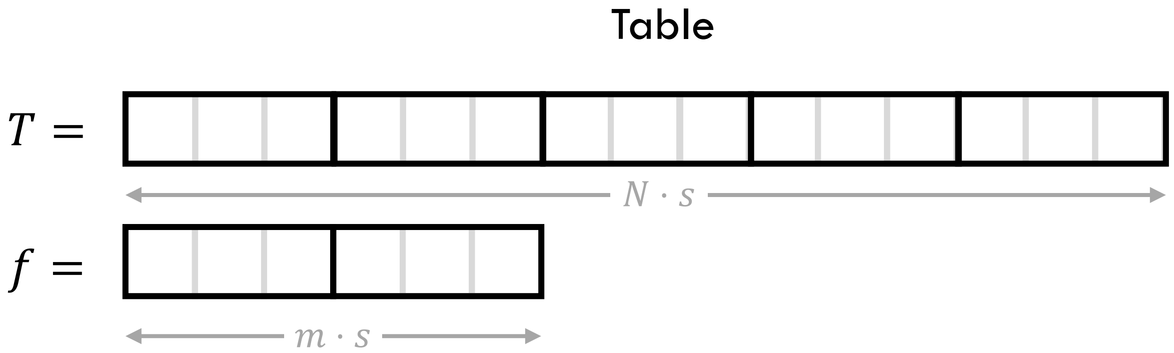
$\forall i \in [m], \exists j \in [N]$ such that

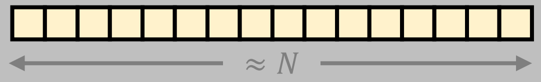
$$f[i] = T[j]$$

Allows out of order and repetitions.

Our Contribution: Segment Lookup

N segments each of size s





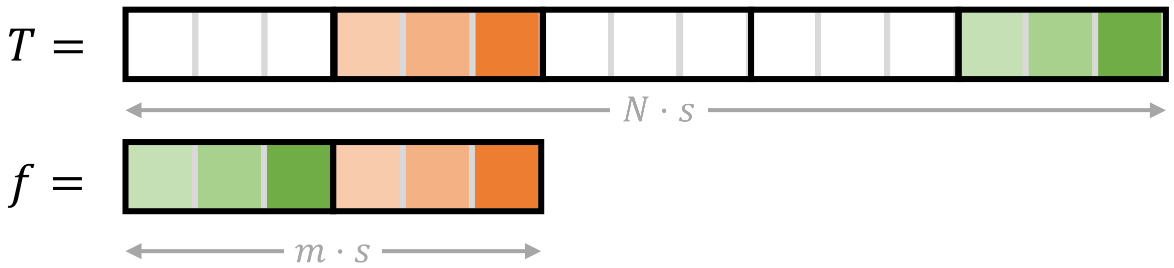
One-time Table Pre-processing

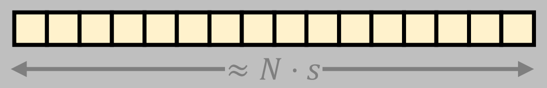
$\mathrm{Com}(T)$

Our Contribution: Segment Lookup

N segments each of size s

Table





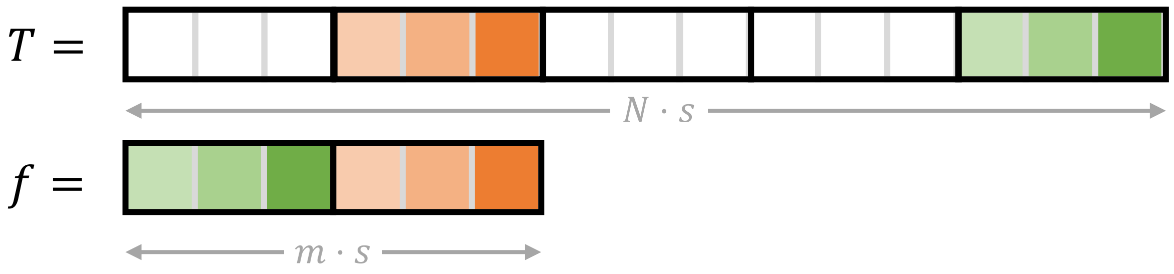
One-time Table Pre-processing

$\mathrm{Com}(T)$

Our Contribution: Segment Lookup

N segments each of size s

Table

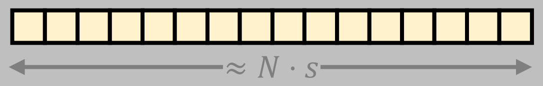


$\forall i \in [m], \exists j \in [N]$ such that

$$f[is+1] = T[js+1],$$
$$f[is+2] = T[js+2]$$
$$\vdots$$
$$f[(i+1)s] = T[(j+1)s]$$

$\mathrm{Com}(f)$ Π

$\mathrm{Com}(T)$

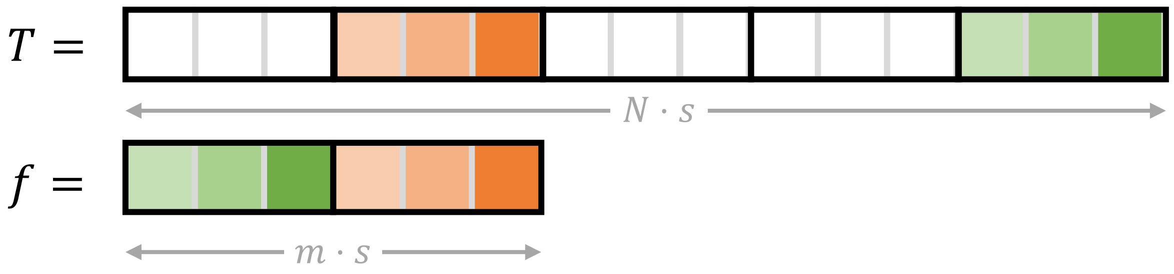


One-time Table Pre-processing

Our Contribution: Segment Lookup

N segments each of size s

Table



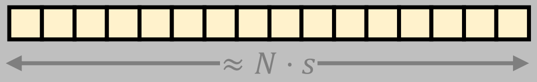
$\forall i \in [m], \exists j \in [N]$ such that

$$f[is+1] = T[js+1],$$
$$f[is+2] = T[js+2]$$
$$\vdots$$
$$f[(i+1)s] = T[(j+1)s]$$

Allows out of order and repeated segments.

$\mathrm{Com}(f)$ Π

$\mathrm{Com}(T)$



One-time Table Pre-processing

Our Contribution: Segment Lookup

N segments each of size s

Table

$T =$

$N \cdot s$

$f =$

Theorem: Segment Lookup

Online Prover Time - $O(ms\,(\log ms + \log N))$

Proof Size* - $O(1)$

Black-box in Cryptography

$\forall i \in [m], \exists j \in [N]$ such that

$$f[is+1] = T[js+1],$$
$$f[is+2] = T[js+2]$$
$$\vdots$$
$$f[(i+1)s] = T[(j+1)s]$$

Allows out of order and repeated segments.

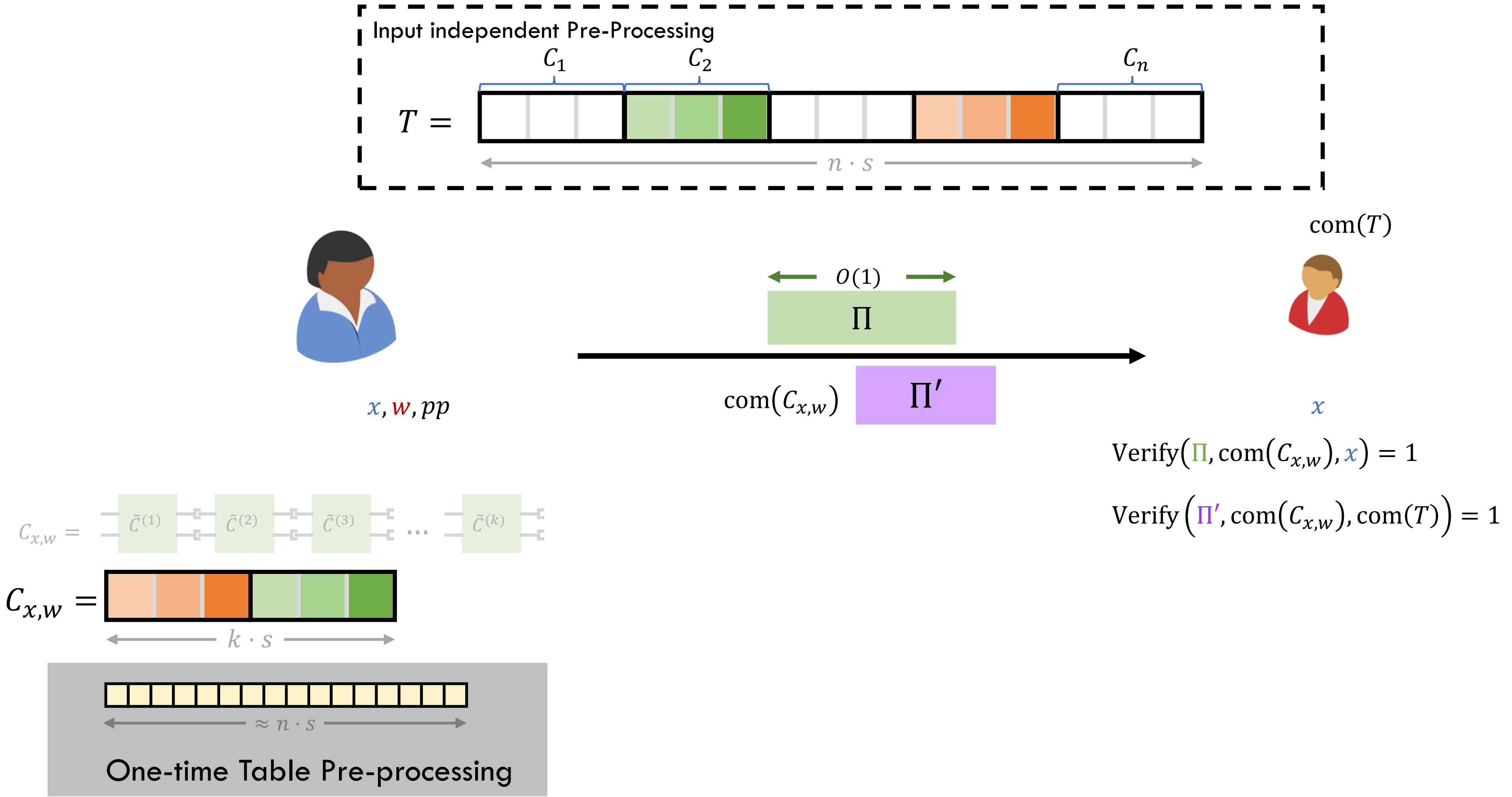
$\text{Com}(T)$

$\approx N \cdot s$

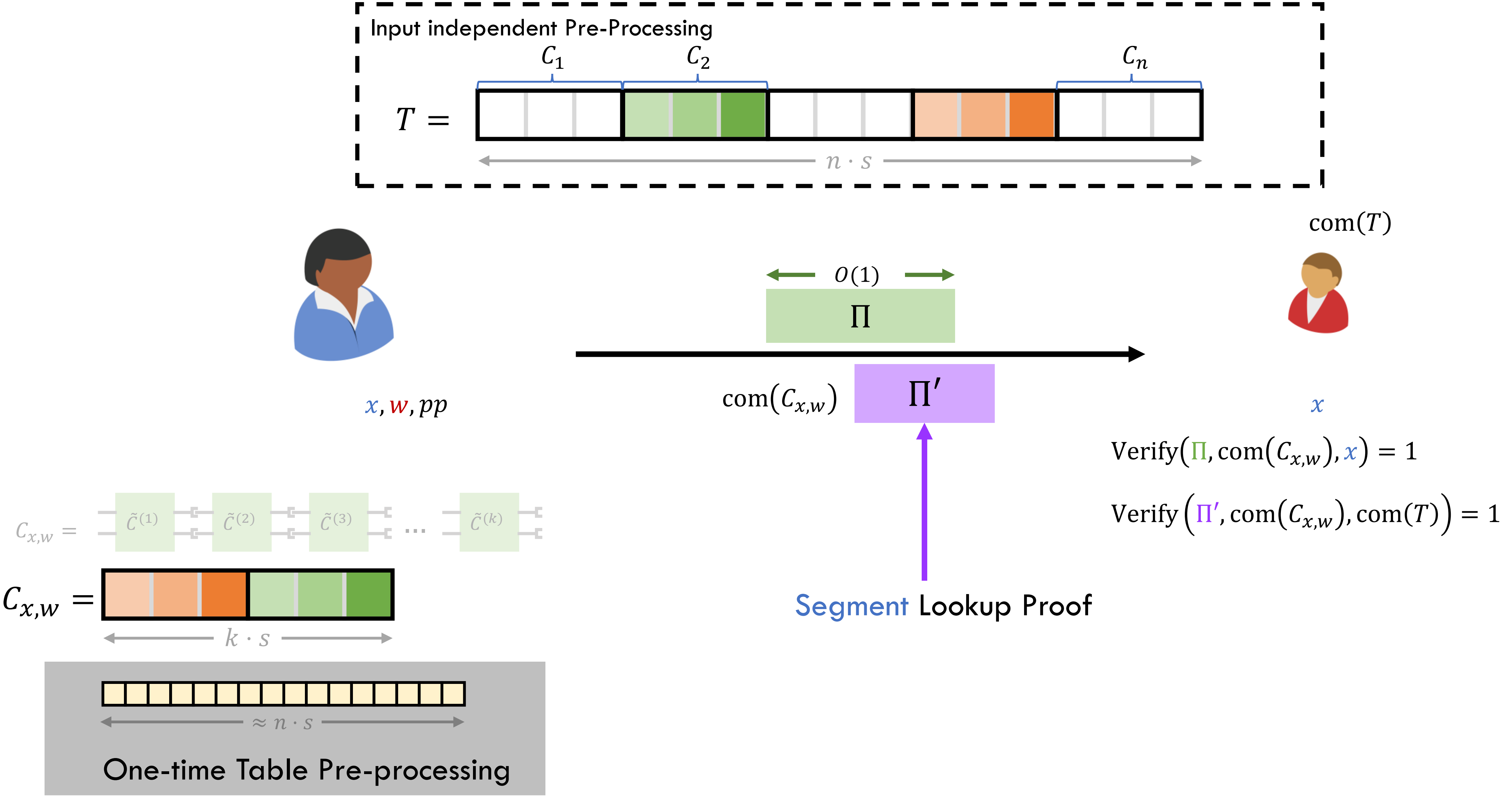
One-time Table Pre-processing

*Proof Size includes size of $\text{com}(f)$

Sublinear Prover $\mathcal{PlonK}$



Sublinear Prover $\mathcal{P}\mathfrak{lon}\mathcal{K}$



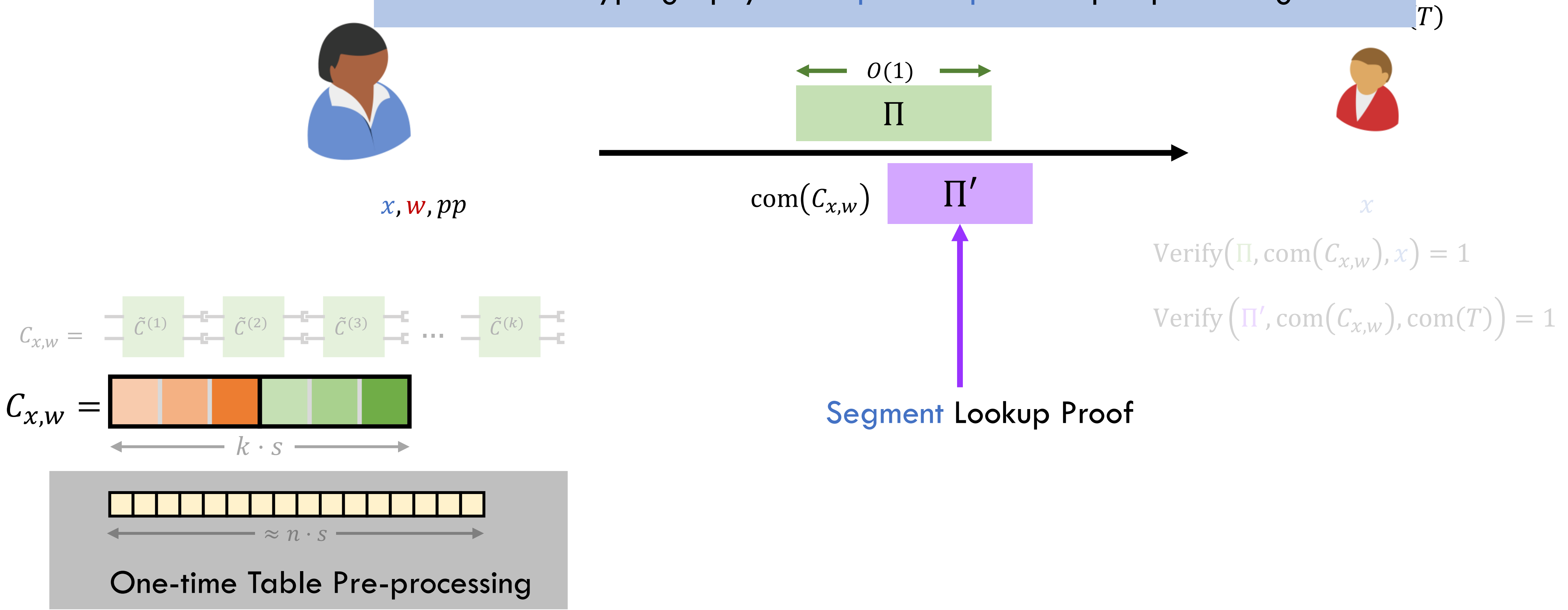
Sublinear Prover $\mathcal{PlonK}$

Theorem: Sublinear Prover $\mathcal{PlonK}$

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

Proof Size* - $O(1)$

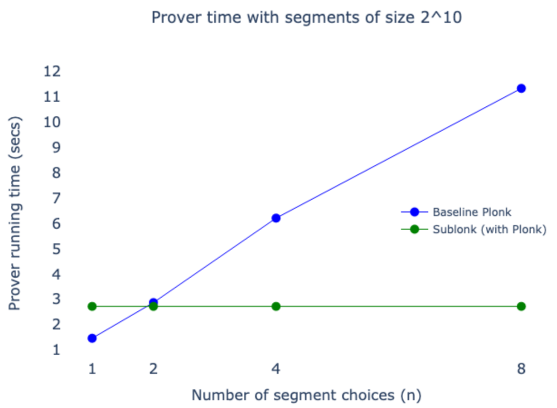
Black-box in Cryptography with input independent pre-processing



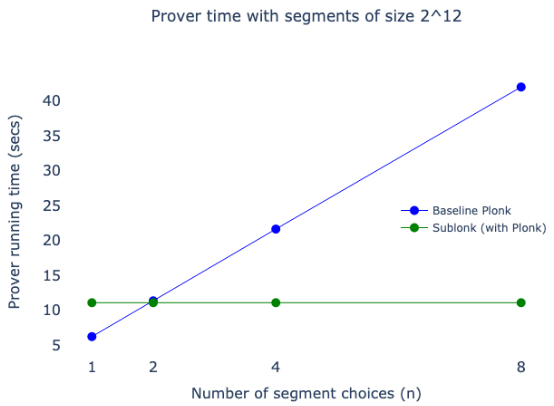
Experimental Results

Prover Time (Comparison to Baseline $\mathcal{P}\mathfrak{lon}\mathcal{K}$)

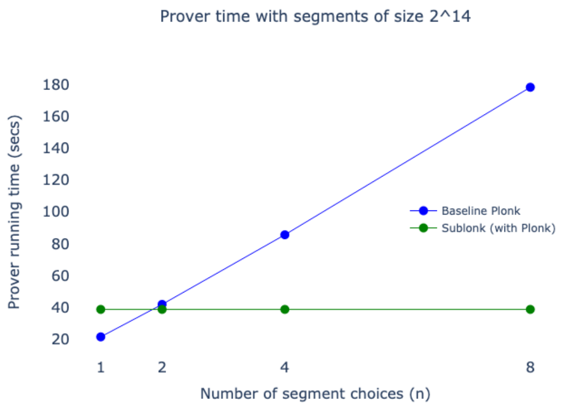
$s = 2^{10}$



$s = 2^{12}$



$s = 2^{14}$



$k = 128$

Experimental Results

Proof Size and Verification Cost (Comparison to Baseline $\mathcal{P}\mathfrak{lon}\mathcal{K}$)

	$\mathcal{P}\mathfrak{lon}\mathcal{K}$	Sub$\mathfrak{lon}\mathcal{K}$
Proof size	$9\ \mathbb{G}$ and $6\ \mathbb{F}$	$42\ \mathbb{G}$ and $12\ \mathbb{F}$
Verification Cost	$18\ \mathbb{G}$ and 2 Pairings	$27\ \mathbb{G}$ and 23 Pairings

Thank you. Questions?

Arka Rai Choudhuri

arkarai.choudhuri@ntt-research.com

ia.cr/2023/902