SNARGs and BARGs from LWE

Arka Rai Choudhuri
Johns Hopkins University

Abhishek Jain
Johns Hopkins University

Zhengzhong Jin
Johns Hopkins University

Succinct Non-Interactive Arguments (SNARGs)

Common Reference String (CRS)

$\mathcal{M}$, x

$\mathcal{M}$, x

$x \longrightarrow \boxed{\mathcal{M}} \longrightarrow$ accept

within T steps

Succinct Non-Interactive Arguments (SNARGs)

Common Reference String (CRS)

$\boxed{\mathcal{M}}, x$

$\boxed{\mathcal{M}}, x$

$x \longrightarrow \boxed{\mathcal{M}} \longrightarrow$ accept

within T steps

 wants to delegate computation to 

Succinct Non-Interactive Arguments (SNARGs)

Common Reference String (CRS)

Π

$\mathcal{M}$, x

$\mathcal{M}$, x

$x \longrightarrow \boxed{\mathcal{M}} \longrightarrow$ accept

within T steps

Succinct Non-Interactive Arguments (SNARGs)

Common Reference String (CRS)

$\mathcal{M}$, x

Π

$\mathcal{M}$, x

Π is publicly verifiable

$x \longrightarrow \boxed{\mathcal{M}} \longrightarrow$ accept

within T steps

Succinct Non-Interactive Arguments (SNARGs)

Common Reference String (CRS)

$\mathcal{M}$, x

$\longleftarrow$ $\mathrm{polylog}(T)$ $\longrightarrow$

Π

$\mathcal{M}$, x

Verifier running time:
$\mathrm{polylog}(T)$

Π is publicly verifiable

$x \longrightarrow \boxed{\mathcal{M}} \longrightarrow$ accept

within T steps

Succinct Non-Interactive Arguments (SNARGs)

Common Reference String (CRS)

$\leftarrow$ polylog(T) $\rightarrow$

Π

$\mathcal{M}$, x $\longrightarrow$ $\mathcal{M}$, x

Verifier running time: polylog(T)

Π is publicly verifiable

$x \longrightarrow \boxed{\mathcal{M}} \longrightarrow$ accept

within T steps

No PPT can produce accepting Π if

$x \longrightarrow \boxed{\mathcal{M}} \not\longrightarrow$ accept

within T steps

Succinct Non-Interactive Arguments (SNARGs)

Common Reference String (CRS)

$\leftarrow$ polylog(T) $\rightarrow$

Π

$\mathcal{M}$, x

$\mathcal{M}$, x

Verifier running time: polylog(T)

Π is publicly verifiable

$x \longrightarrow \boxed{\mathcal{M}} \longrightarrow$ accept

within T steps

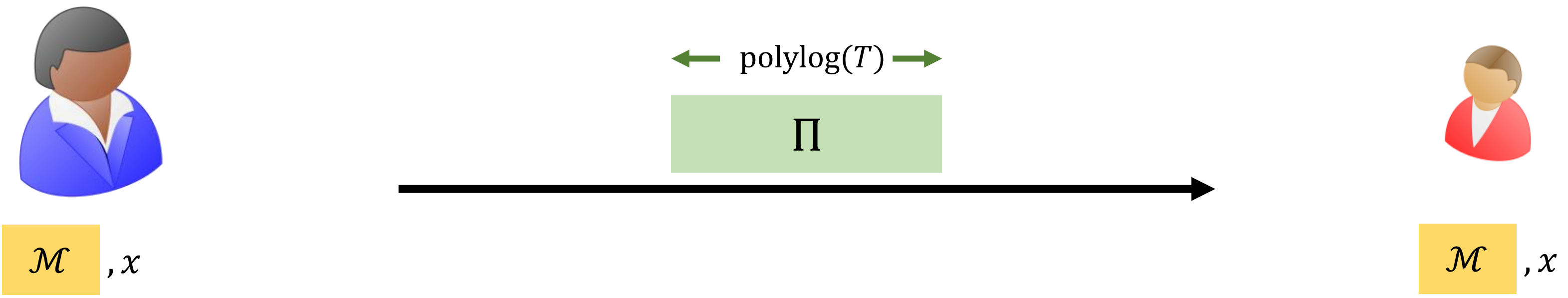
No PPT can produce accepting x, Π if

$x \longrightarrow \boxed{\mathcal{M}} \not\longrightarrow$ accept

within T steps

Succinct Non-Interactive Arguments (SNARGs)

Common Reference String (CRS)



Verifier running time: polylog(T)

Π is publicly verifiable

What kind of computation can we hope to delegate based on standard assumptions?

Succinct Non-Interactive Arguments (SNARGs)

Common Reference String (CRS)

$\mathcal{M}$, x

$\leftarrow$ polylog(T) $\rightarrow$

Π

$\mathcal{M}$, x

Verifier running time: polylog(T)

Π is publicly verifiable

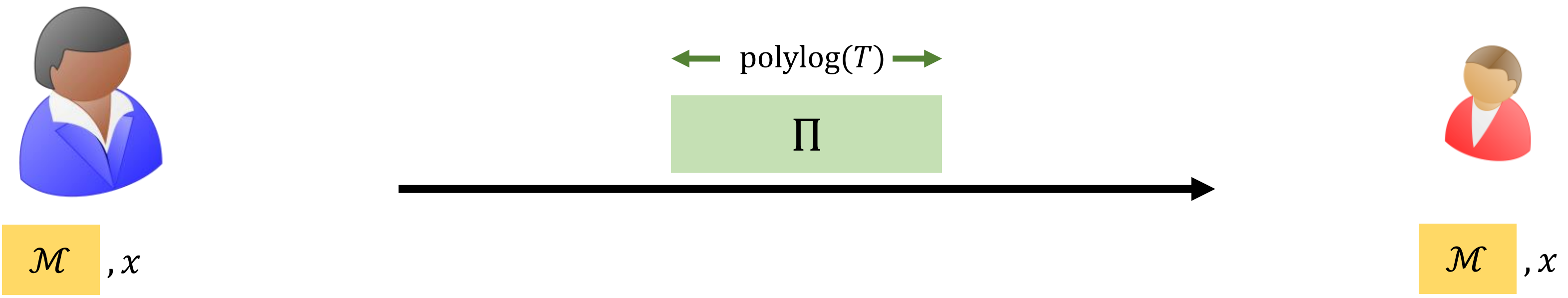
What kind of computation can we hope to delegate based on standard assumptions?

- Nondeterministic polynomial-time computation (NP)? Unlikely! [Gentry-Wichs'11]

x

Succinct Non-Interactive Arguments (SNARGs)

Common Reference String (CRS)



Verifier running time: polylog(T)

Π is publicly verifiable

What kind of computation can we hope to delegate based on standard assumptions?

- Nondeterministic polynomial-time computation (NP)? Unlikely! [Gentry-Wichs'11]
- Deterministic polynomial-time computation (P)?

x

Succinct Non-Interactive Arguments (SNARGs)

Common Reference String (CRS)

$\leftarrow$ polylog(T) $\rightarrow$

Π

$\mathcal{M}$, x

$\mathcal{M}$, x

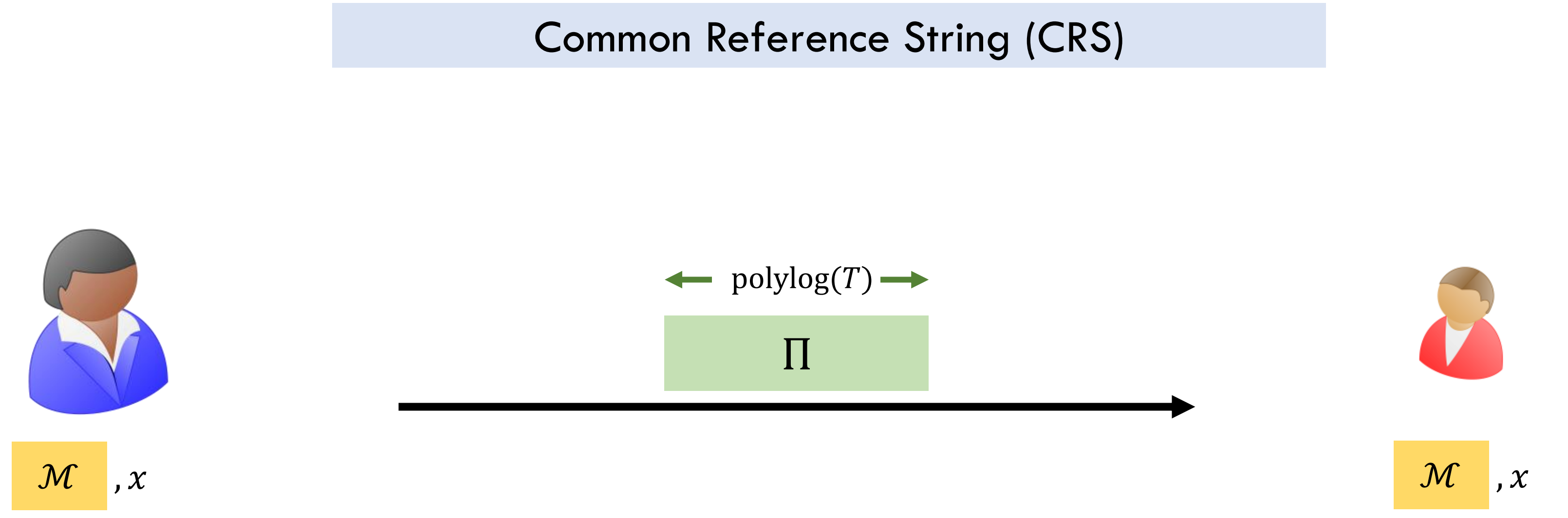
Verifier running time: polylog(T)

Π is publicly verifiable

What kind of computation can we hope to delegate based on standard assumptions?

- Nondeterministic polynomial-time computation (NP)? Unlikely! [Gentry-Wichs'11]
- Deterministic polynomial-time computation (P)?
- Sub-classes of NP?

Succinct Non-Interactive Arguments (SNARGs)



Verifier running time: polylog(T)

Π is publicly verifiable

What kind of computation can we hope to delegate based on standard assumptions?

- Nondeterministic polynomial-time computation (NP)? Unlikely! [Gentry-Wichs'11]
- Deterministic polynomial-time computation (P)?
- Sub-classes of NP?

Non-Interactive Batch Arguments (BARGs)

CRS

$C, x_1, \cdots, x_k$

Π

$C, x_1, \cdots, x_k$

Π is publicly verifiable

$\mathrm{SAT} = \{(C, x) \mid \exists w \; s.t. \; C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \mathrm{SAT}$

Non-Interactive Batch Arguments (BARGs)

CRS

$C, x_1, \cdots, x_k$

$\ll |w| \cdot k$

Π

$C, x_1, \cdots, x_k$

Verifier running time:
$k \cdot |x| + |\Pi|$

Π is publicly verifiable

$\mathrm{SAT} = \{(C, x) \mid \exists w \; s.t. \; C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \mathrm{SAT}$

Prior Works

Prior Works

Non-falsifiable assumptions/ Random oracle model

[Micali'94, Groth'10, Lipmaa'12, Damgård-Faust-Hazay'12, Gennaro-Gentry-Parno-Raykova'13, Bitansky-Chiesa-Ishai-Ostrovsky-Paneth'13, Bitansky-Canetti-Chiesa-Tromer'13, Bitanksy-Canetti-Chiesa-Goldwasser-Lin-Rubinstein-Tromer'17]

Some works can delegate NP

Prior Works

Non-falsifiable assumptions/ Random oracle model

[Micali'94, Groth'10, Lipmaa'12, Damgård-Faust-Hazay'12, Gennaro-Gentry-Parno-Raykova'13, Bitansky-Chiesa-Ishai-Ostrovsky-Paneth'13, Bitansky-Canetti-Chiesa-Tromer'13, Bitanksy-Canetti-Chiesa-Goldwasser-Lin-Rubinstein-Tromer'17]

Some works can delegate NP

"Less standard" assumptions

[Canetti-Holmgren-Jain-Vaikuntanathan'15, Koppula-Lewko-Waters'15, Bitansky-Garg-Lin-Pass-Telang'15, Canetti-Holmgren'16, Ananth-Chen-Chung-Lin-Lin'16, Chen-Chow-Chung-Lai-Lin-Zhou'16, Paneth-Rothblum'17, Canetti-Chen-Holmgren-Lombardi-Rothblum-Rothblum-Wichs'19, Kalai-Paneth-Yang'19]

Delegation for P and some for batch NP

Prior Works

Non-falsifiable assumptions/ Random oracle model

[Micali'94, Groth'10, Lipmaa'12, Damgård-Faust-Hazay'12, Gennaro-Gentry-Parno-Raykova'13, Bitansky-Chiesa-Ishai-Ostrovsky-Paneth'13, Bitansky-Canetti-Chiesa-Tromer'13, Bitanksy-Canetti-Chiesa-Goldwasser-Lin-Rubinstein-Tromer'17]

Some works can delegate NP

"Less standard" assumptions

[Canetti-Holmgren-Jain-Vaikuntanathan'15, Koppula-Lewko-Waters'15, Bitansky-Garg-Lin-Pass-Telang'15, Canetti-Holmgren'16, Ananth-Chen-Chung-Lin-Lin'16, Chen-Chow-Chung-Lai-Lin-Zhou'16, Paneth-Rothblum'17, Canetti-Chen-Holmgren-Lombardi-Rothblum-Rothblum-Wichs'19, Kalai-Paneth-Yang'19]

Delegation for P and some for batch NP

Designated Verifier (standard assumptions)

[Kalai-Raz-Rothblum'13, Kalai-Raz-Rothblum'14, Kalai-Paneth'16, Brakerski-Holgren-Kalai'17, Badrinarayanan-Kalai-Khurana-Sahai-Wichs'18, Holmgren-Rothblum'18, Brakerski-Kalai'20]

Delegation for P and some for batch NP

Our Results

BARGs

	Proof size	Assumptions
[C-Jain-Jin'21a]	$\tilde{O}(\|C\| + \sqrt{k\|C\|})$	QR + (LWE/sub-exp DDH)

QR – Quadratic residuosity, LWE – Learning with Error, DDH – Decisional Diffie-Hellman

$\mathrm{SAT} = \{(C, x) \mid \exists w \; s.t. \; C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \mathrm{SAT}$

Our Results

BARGs

	Proof size	Assumptions
[C-Jain-Jin'21a]	$\tilde{O}(\|C\| + \sqrt{k\|C\|})$	QR + (LWE/sub-exp DDH)
[C-Jain-Jin'21b]	$\mathrm{poly}(\log k, \log\|C^*\|, \|w\|)$	LWE

QR – Quadratic residuosity, LWE – Learning with Error, DDH – Decisional Diffie-Hellman

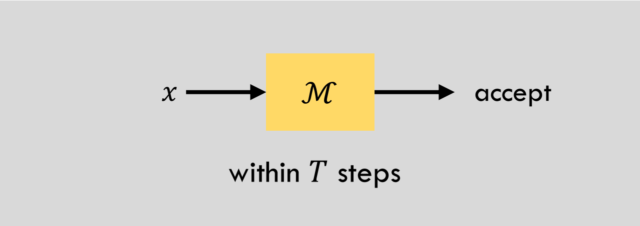
$\mathrm{SAT} = \{(C, x) \mid \exists w \; s.t. \; C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \mathrm{SAT}$

Our Results

SNARGs

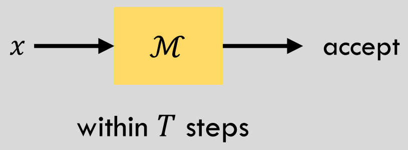
	Model	Assumptions
[C-Jain-Jin'21b]	RAM	LWE



Our Results

SNARGs

	Model	Assumptions
[C-Jain-Jin'21b]	RAM	LWE

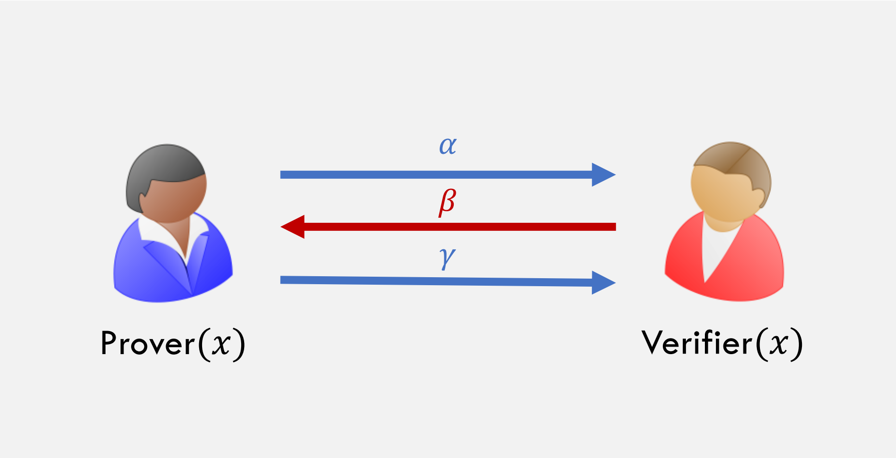


Previously best known: [Jawale-Kalai-Khurana-Zhang'21] for depth bounded computation based on sub-exponential hardness of LWE.

Key Insights

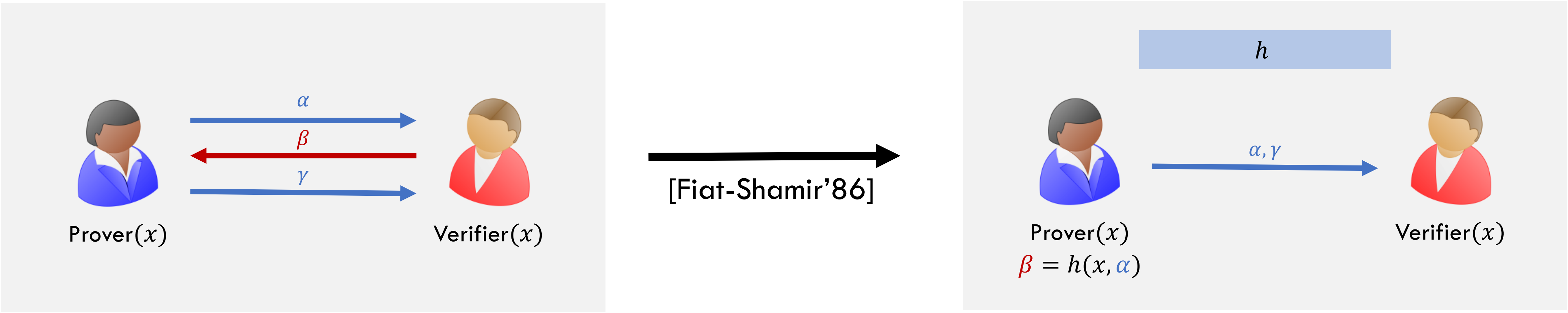
Fiat-Shamir (FS) Methodology

Fiat-Shamir (FS) Methodology



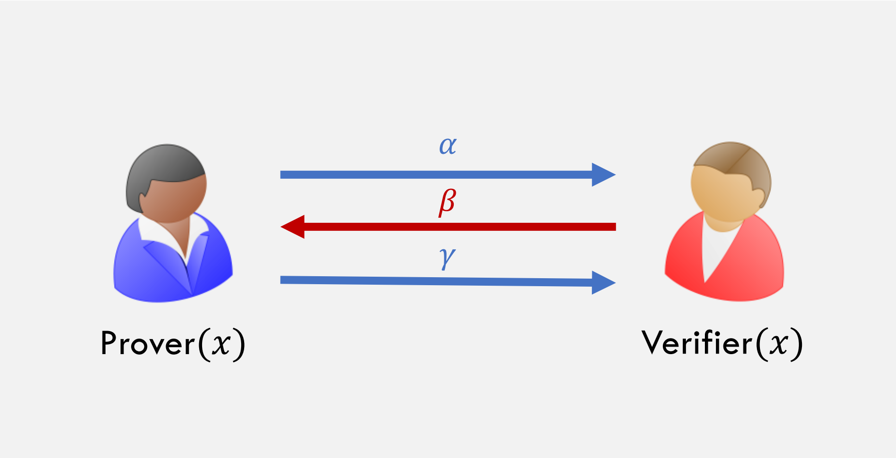
β is a random string

Fiat-Shamir (FS) Methodology

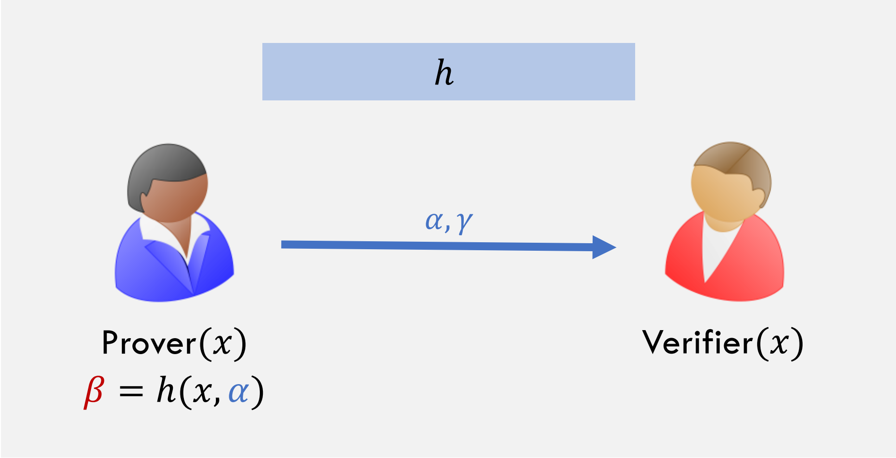


β is a random string

Fiat-Shamir (FS) Methodology



[Fiat-Shamir'86]

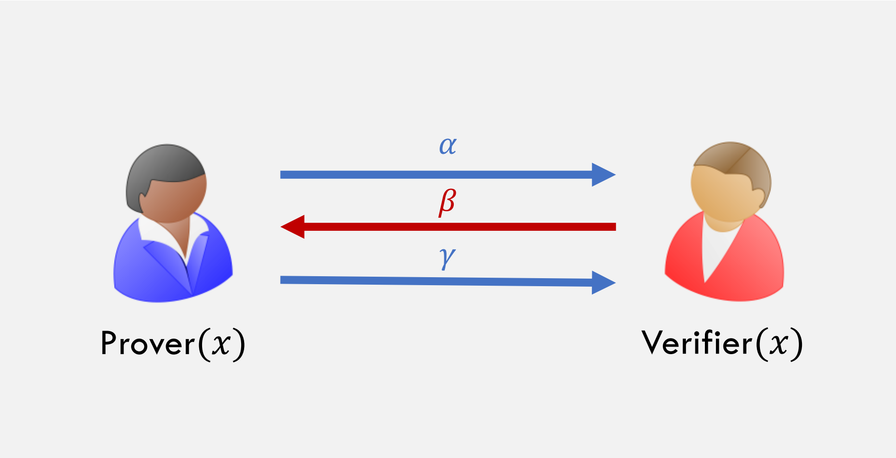


β is a random string

FS methodology is secure for certain protocols under a variety of assumptions (via correlation intractable hash functions)

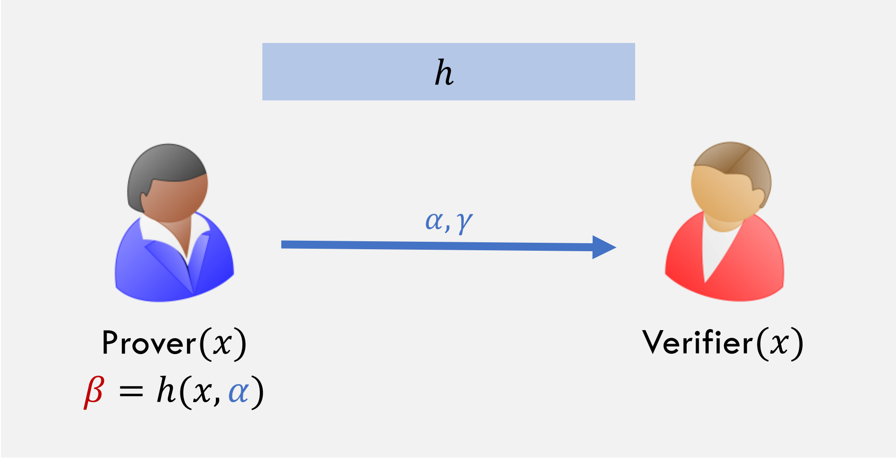
[Kalai-Rothblum-Rothblum'17, Canetti-Chen-Reyzin-Rothblum'18, Holmgren-Lombardi'18, Canetti-Chen-Holmgren-Lombardi-Rothblum-Rothblum-Wichs'19, Peikert-Sheihian'19, Brakerski-Koppula-Mour'20, Couteau-Katsumata-Ursu'20, Jain-Jin'21, Jawale-Kalai-Khurana-Zhang'21, Holmgren-Lombardi-Rothblum'21]

Fiat-Shamir (FS) Methodology



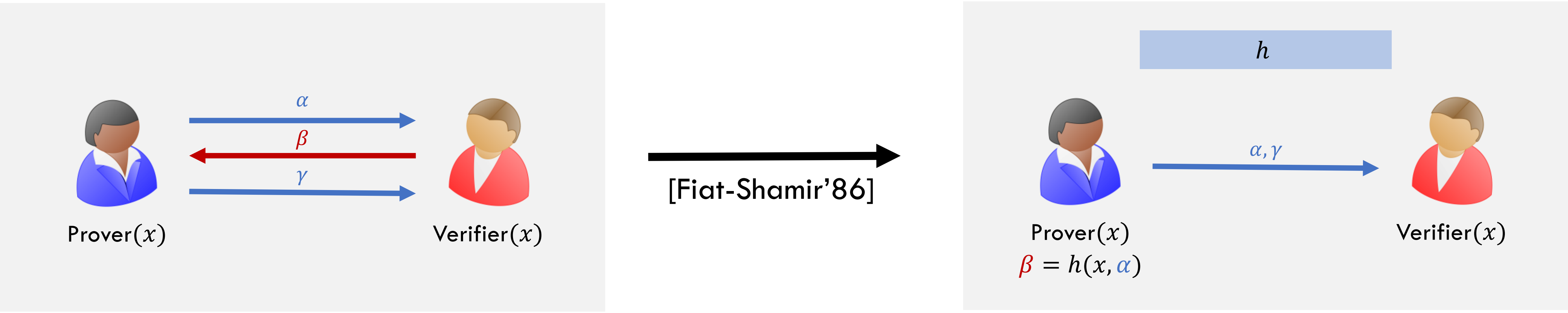
β is a random string

[Fiat-Shamir'86]



FS methodology is secure for certain protocols under a variety of assumptions (via correlation intractable hash functions)

Fiat-Shamir (FS) Methodology

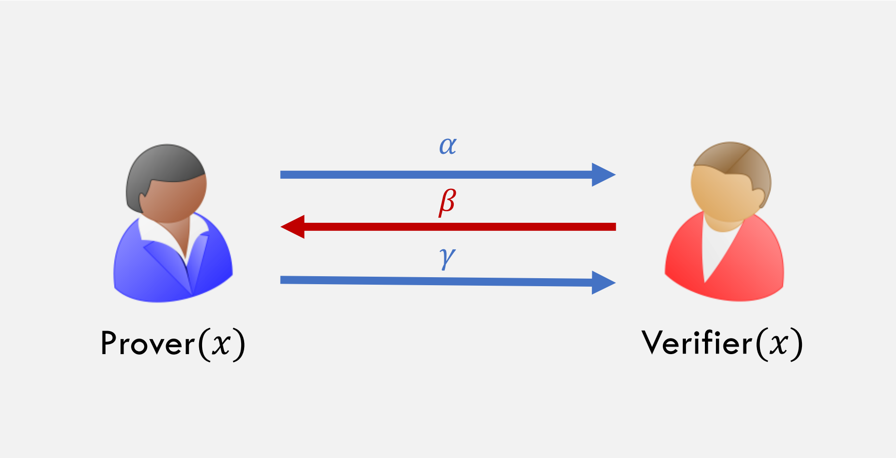


β is a random string

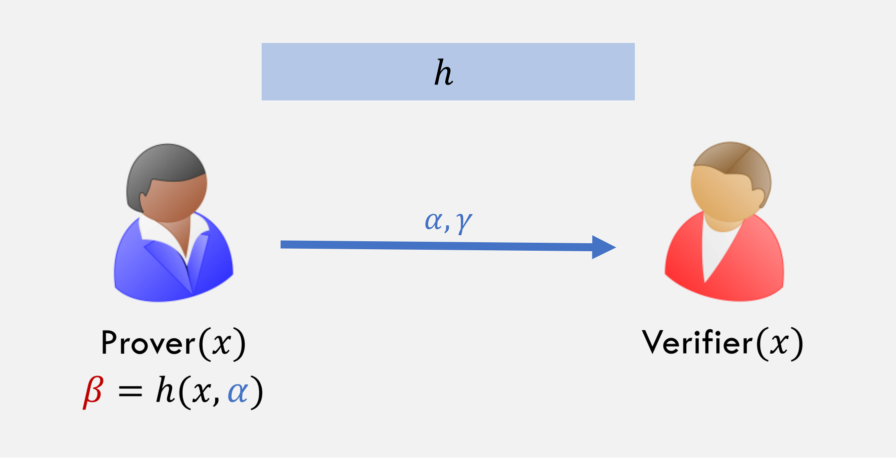
FS methodology is secure for certain protocols under a variety of assumptions (via correlation intractable hash functions)

Proven secure if starting with statistically secure interactive protocols (interactive proofs).

Fiat-Shamir (FS) Methodology



[Fiat-Shamir'86]



β is a random string

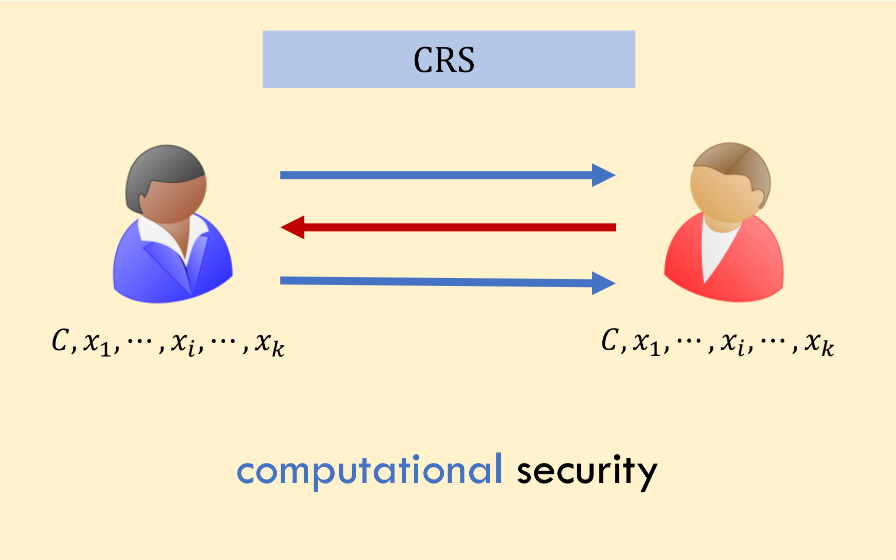
FS methodology is secure for certain protocols under a variety of assumptions (via correlation intractable hash functions)

Proven secure if starting with statistically secure interactive protocols (interactive proofs).

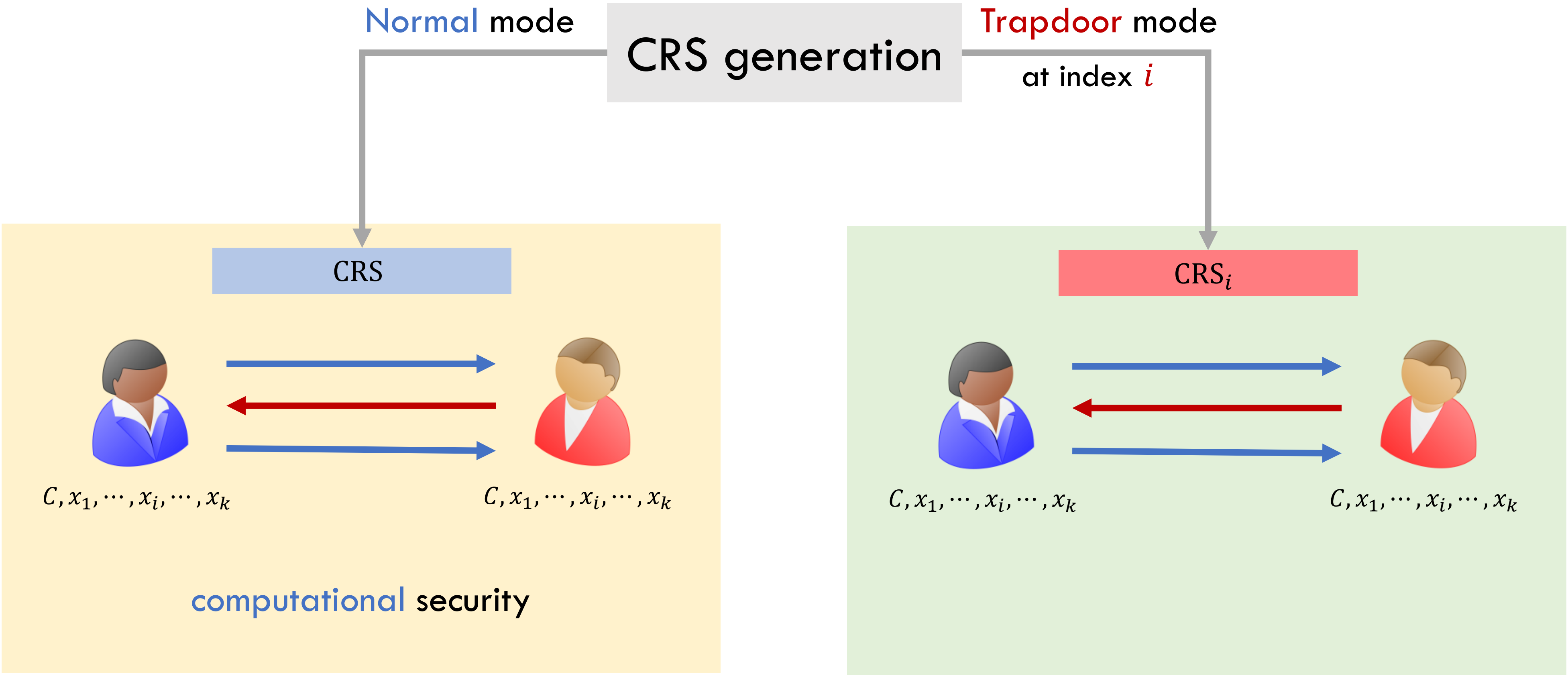
No known interactive proofs for batch NP or delegating deterministic polynomial-time computation.

Dual-Mode Interactive Batch Arguments

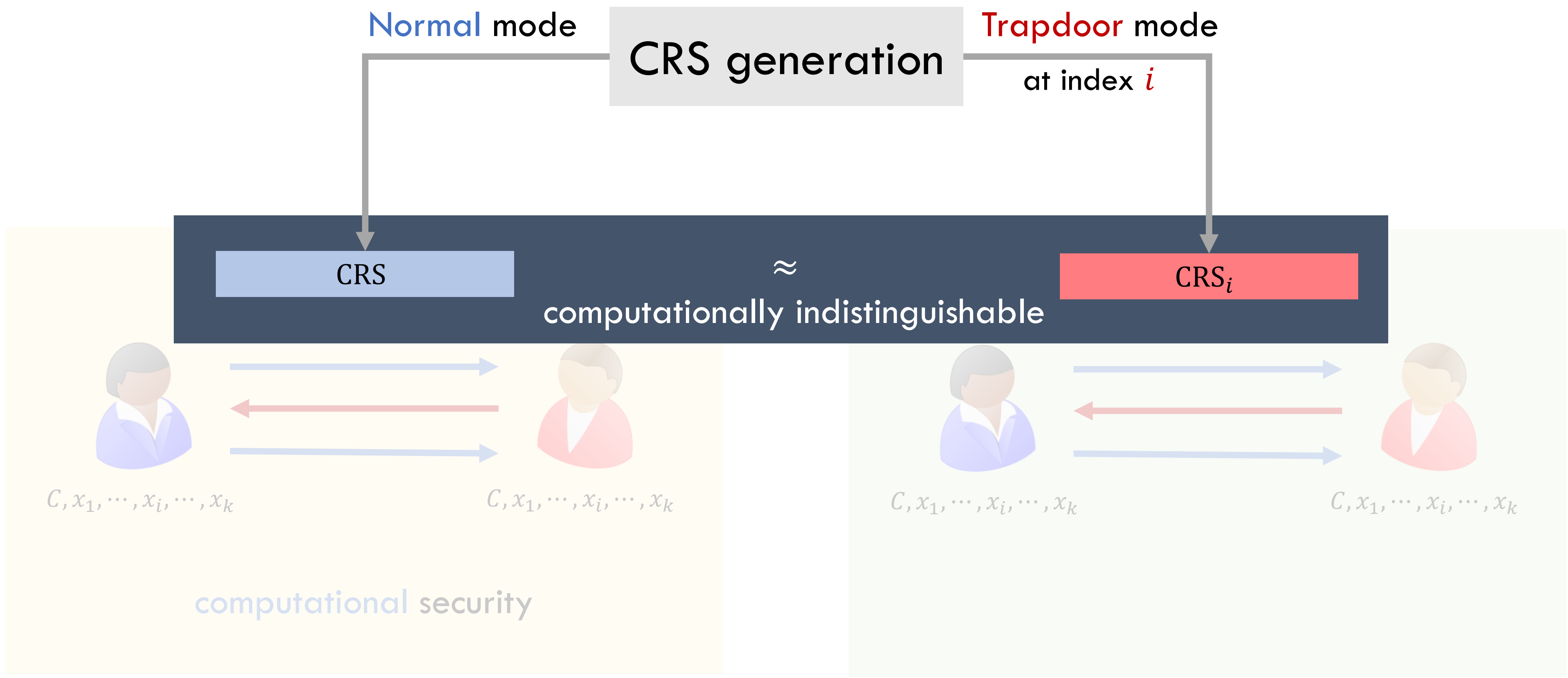
Dual-Mode Interactive Batch Arguments



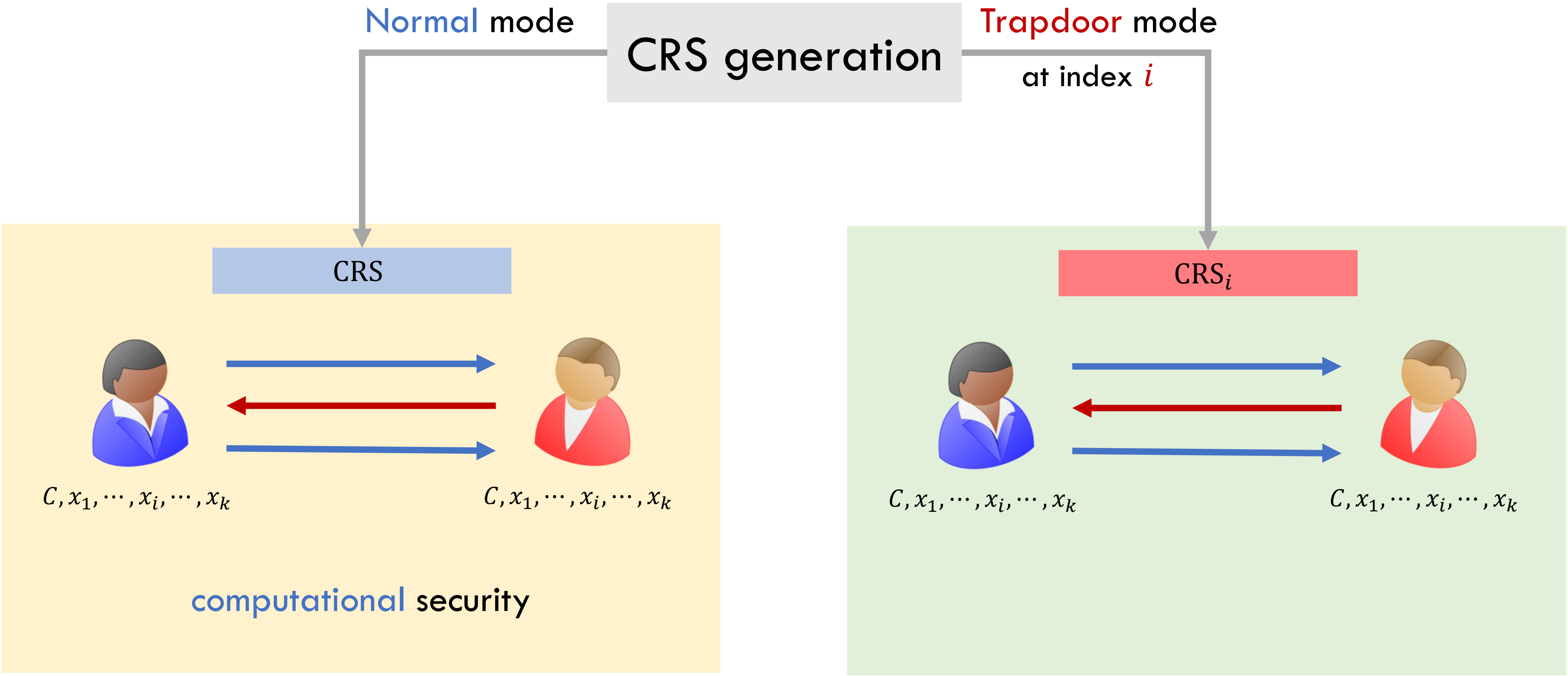
Dual-Mode Interactive Batch Arguments



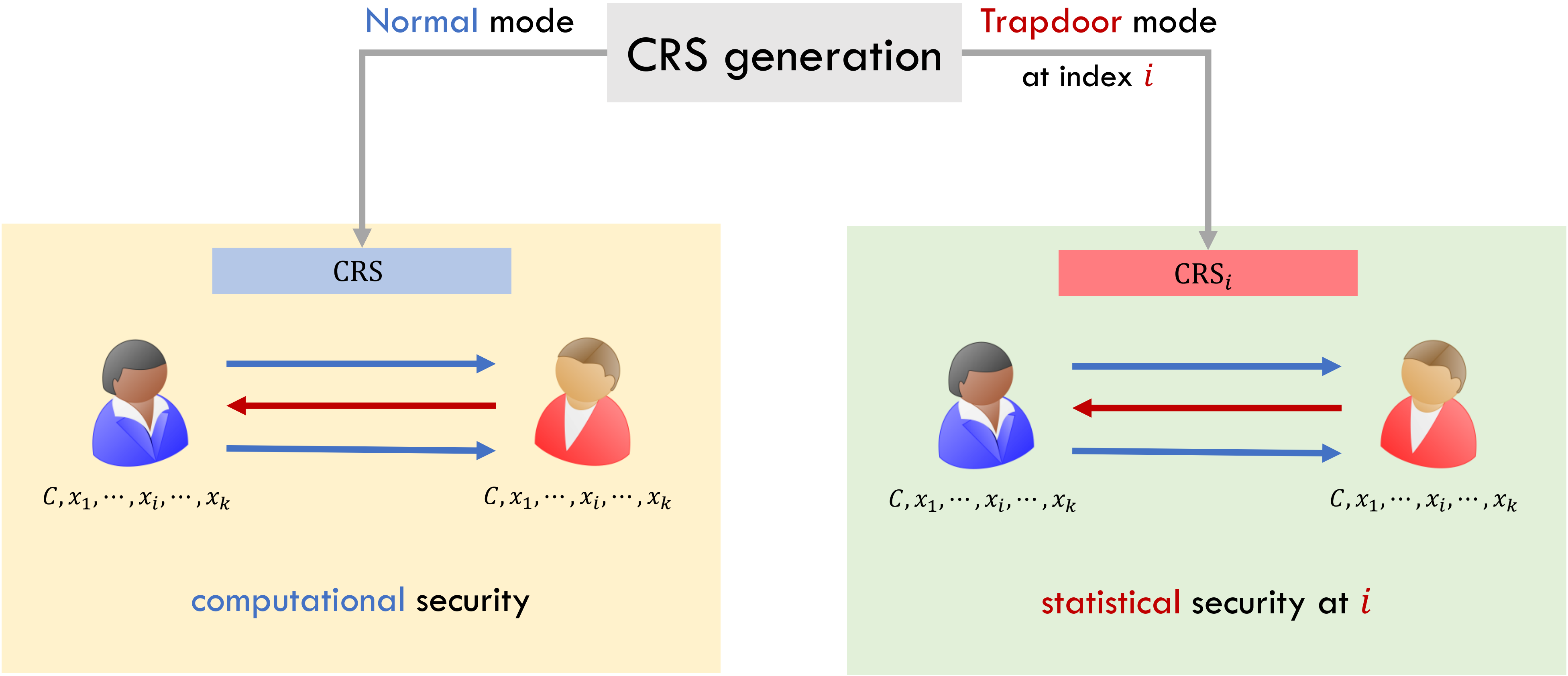
Dual-Mode Interactive Batch Arguments



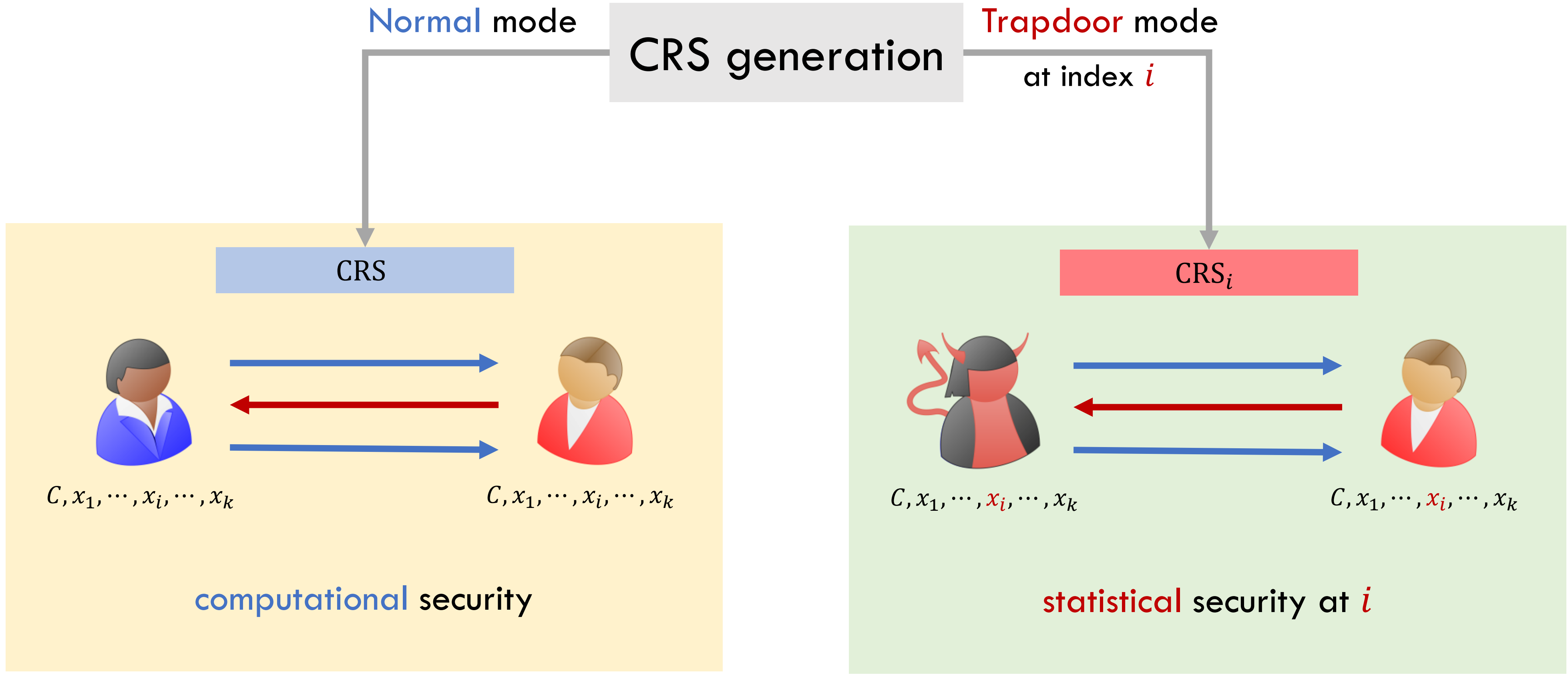
Dual-Mode Interactive Batch Arguments



Dual-Mode Interactive Batch Arguments

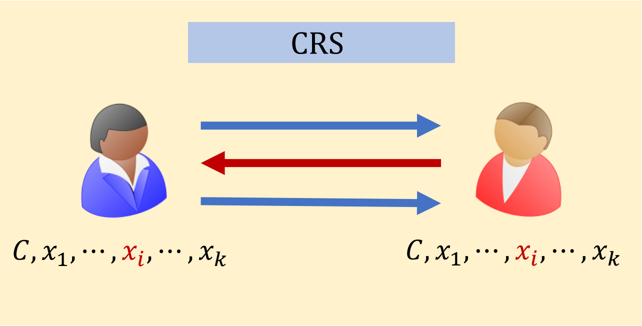


Dual-Mode Interactive Batch Arguments

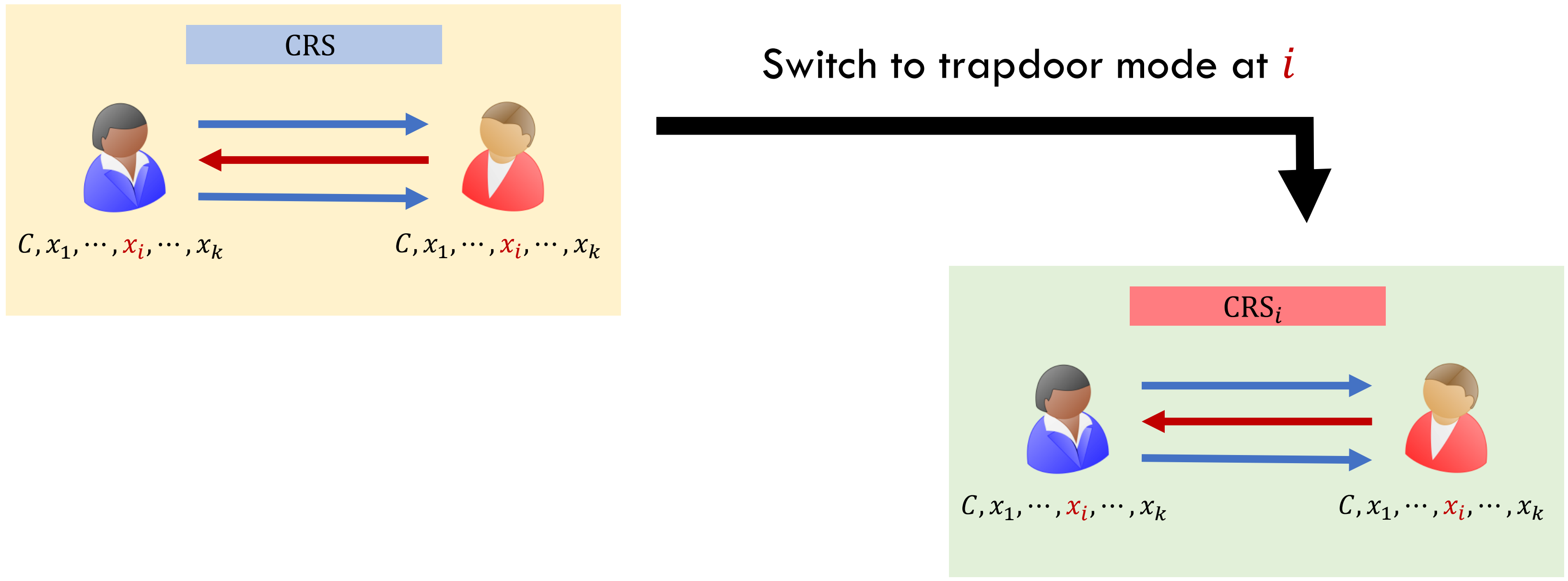


Even unbounded cannot make accept if $(C, x_i) \notin \mathrm{SAT}$

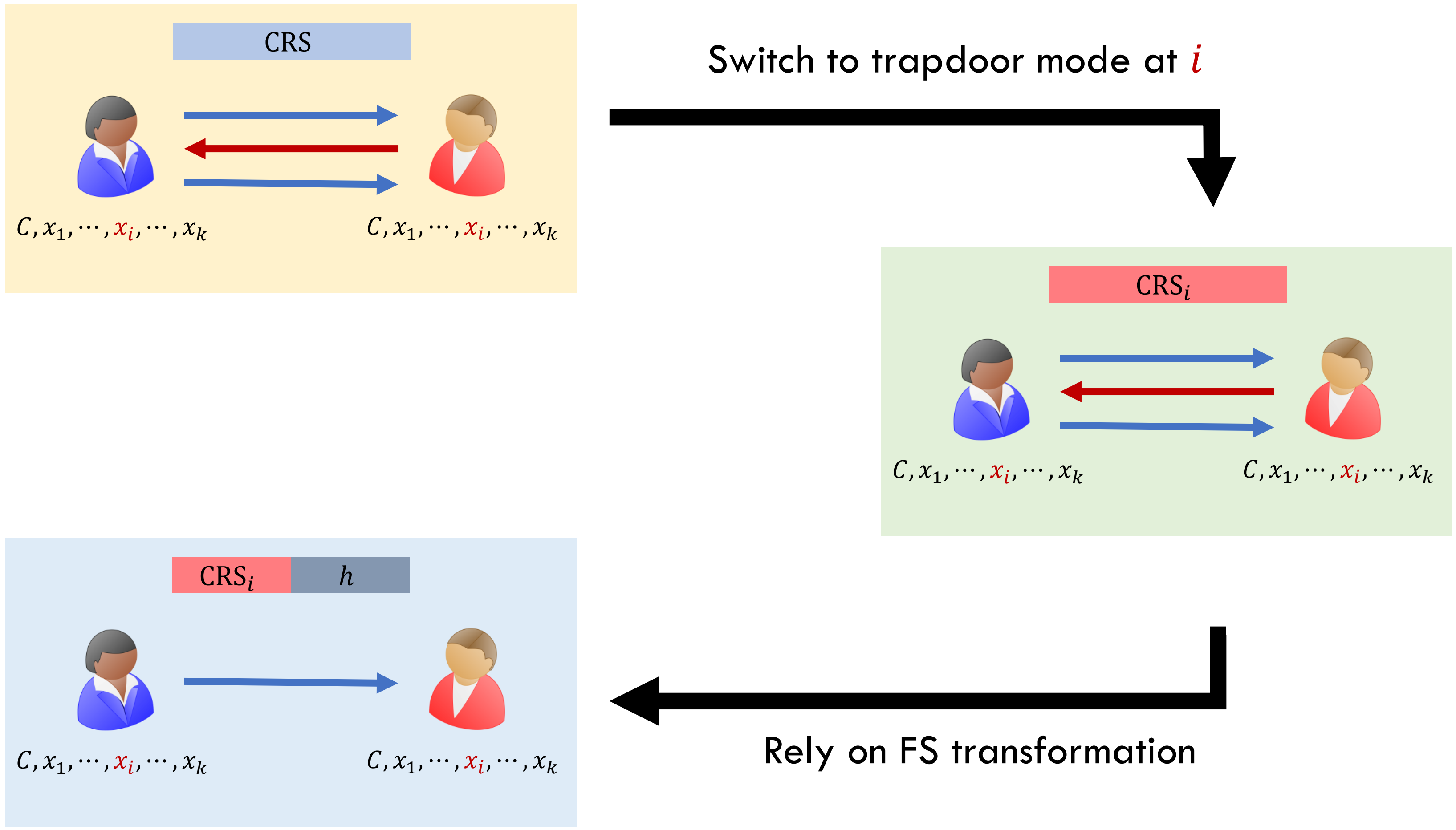
Security Intuition



Security Intuition

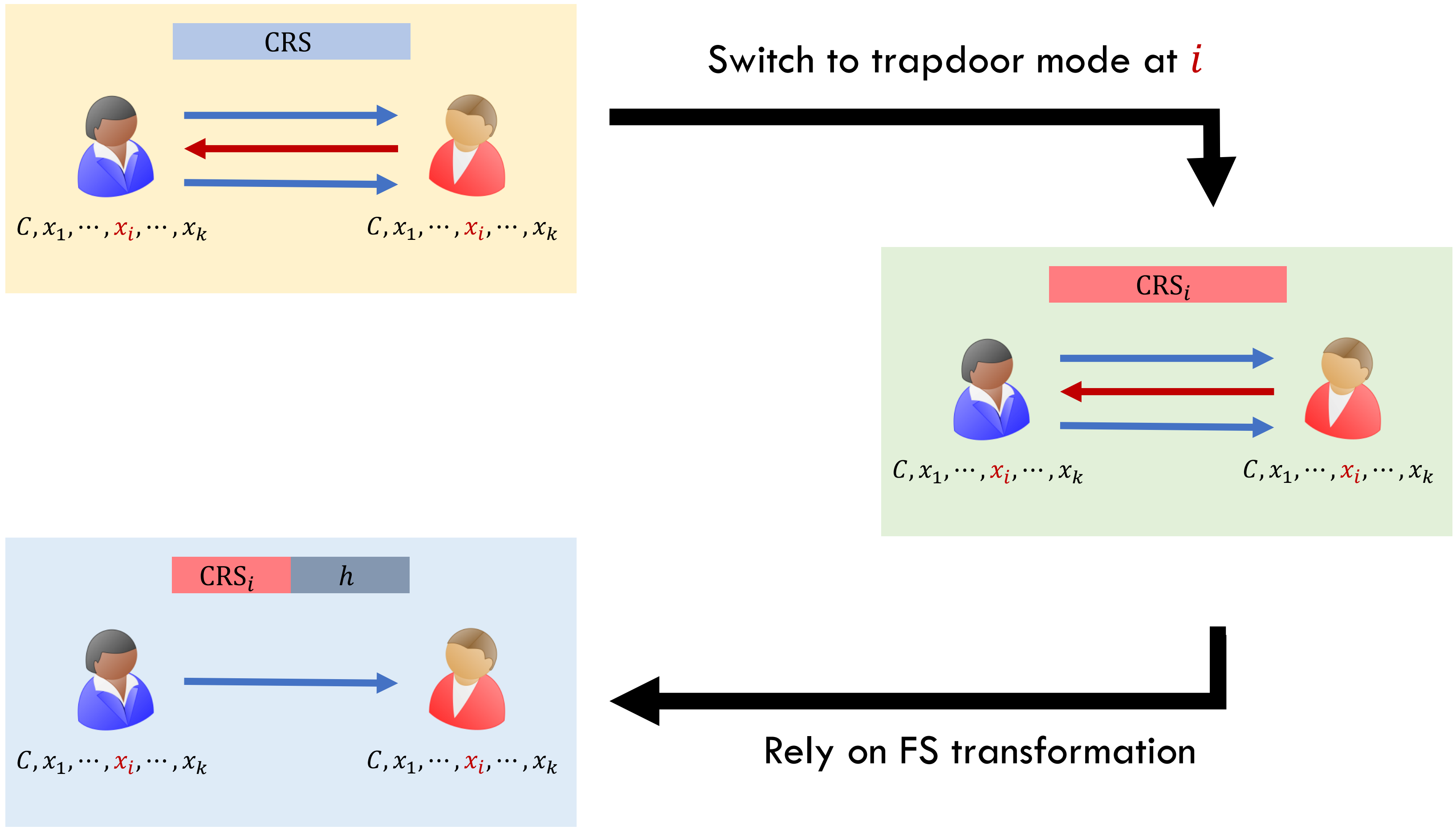


Security Intuition



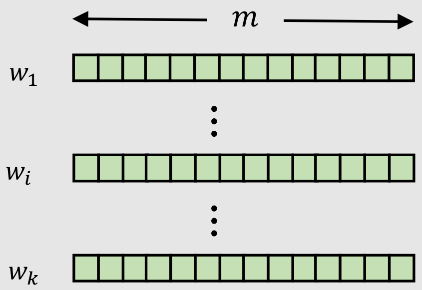
Security Intuition

Non-adaptive security



Dual Mode Batch Argument

Protocol Template

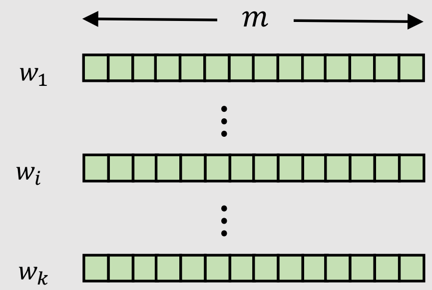


$\mathrm{SAT} = \{(C, x) \mid \exists w \ s.t. \ C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \mathrm{SAT}$

Dual Mode Batch Argument

Protocol Template



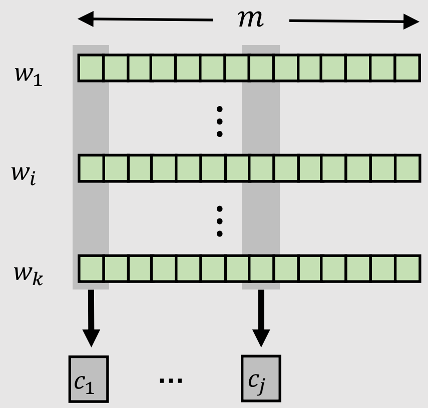
commitment key K

$\mathrm{SAT} = \{(C, x) \mid \exists w \ s.t. \ C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \mathrm{SAT}$

Dual Mode Batch Argument

Protocol Template



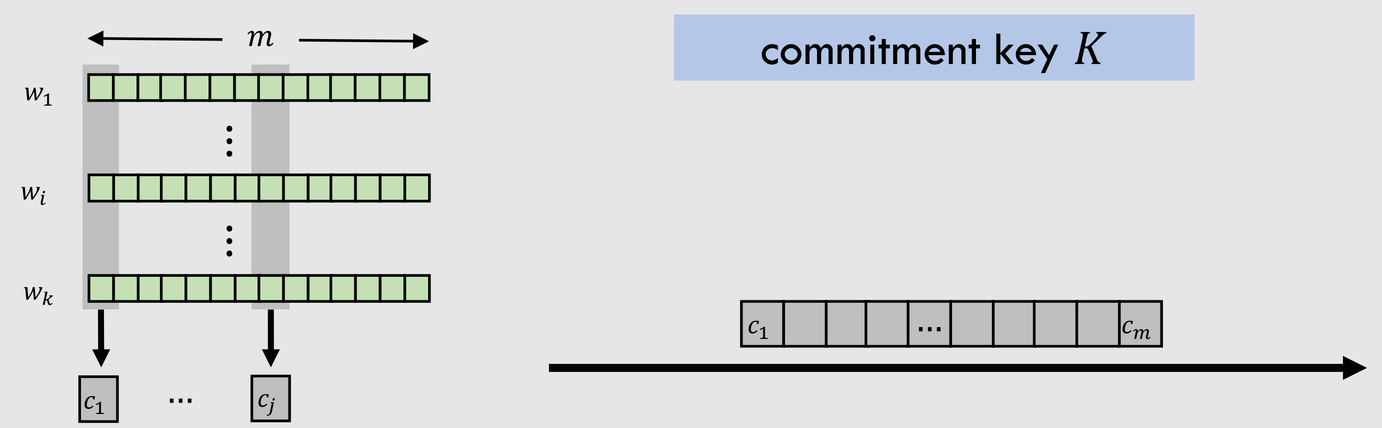
commitment key K

$\mathrm{SAT} = \{(C, x) \mid \exists w \ s.t. \ C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \mathrm{SAT}$

Dual Mode Batch Argument

Protocol Template



SAT = $\{(C, x) \mid \exists w \; s.t. \; C(x, w) = 1\}$

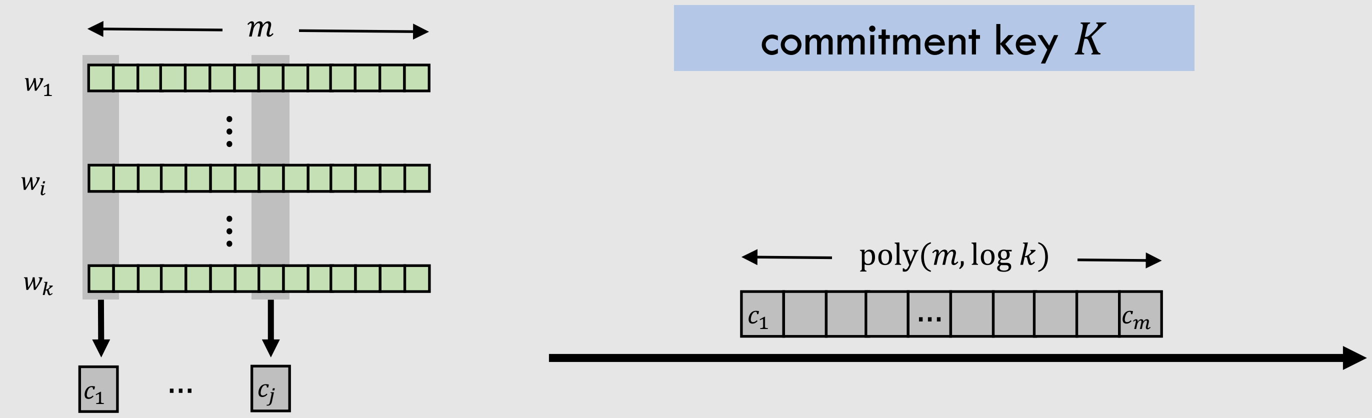
$\forall i \in [k], (C, x_i) \in \text{SAT}$

Dual Mode Batch Argument

Protocol Template

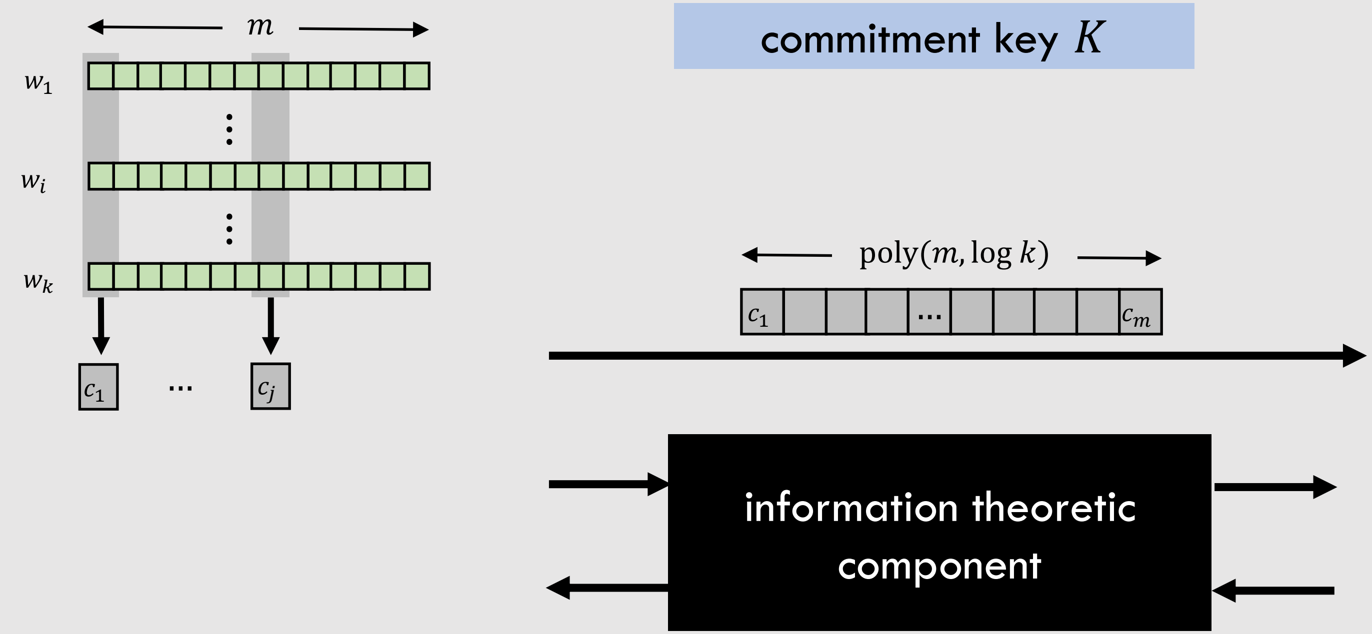
$\mathrm{SAT} = \{(C, x) \mid \exists w \ s.t.\ C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \mathrm{SAT}$



Dual Mode Batch Argument

Protocol Template

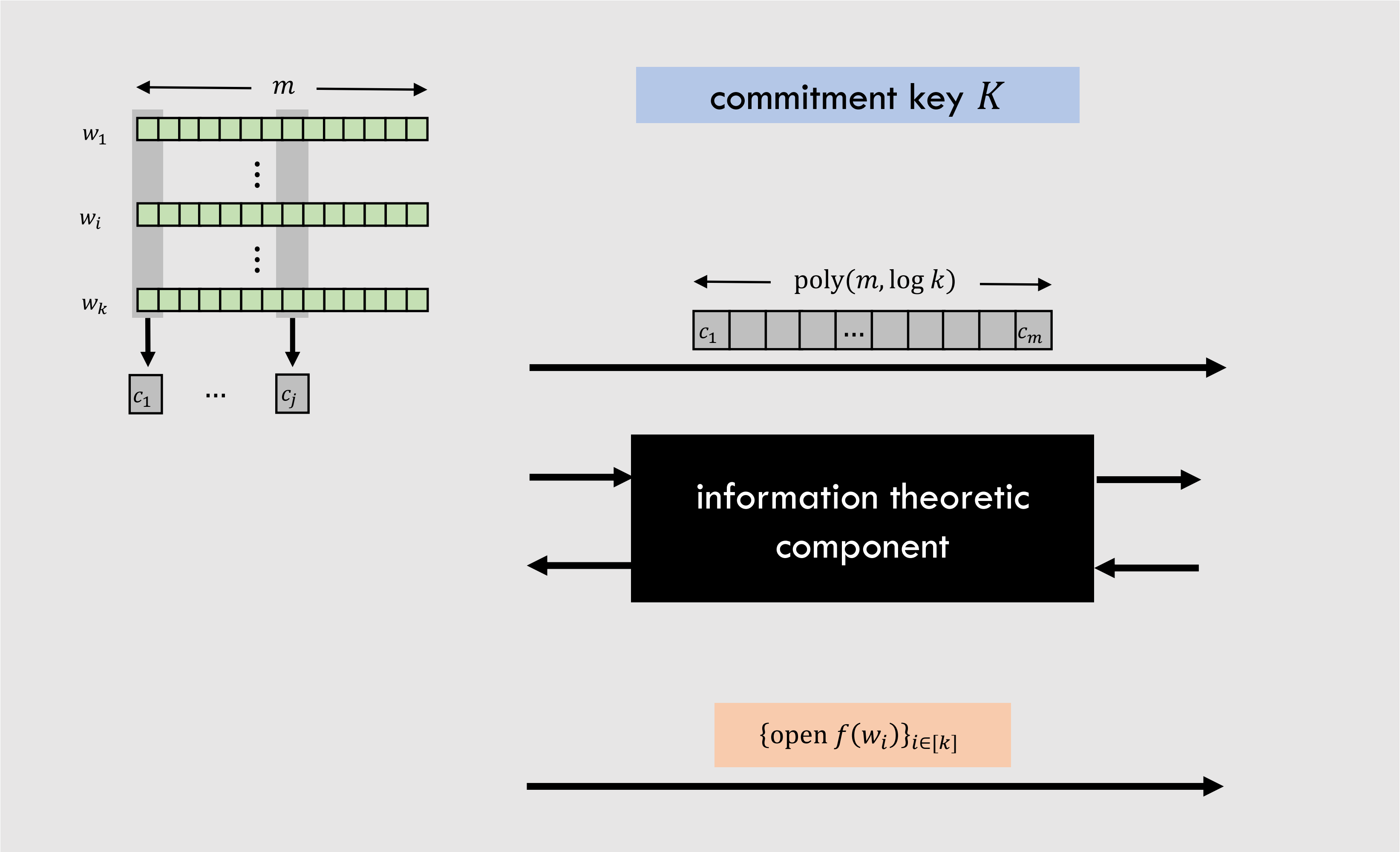


$\text{SAT} = \{(C, x) \mid \exists w \; s.t. \; C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \text{SAT}$

Dual Mode Batch Argument

Protocol Template



$\mathrm{SAT} = \{(C, x) \mid \exists w \; s.t. \; C(x, w) = 1\}$

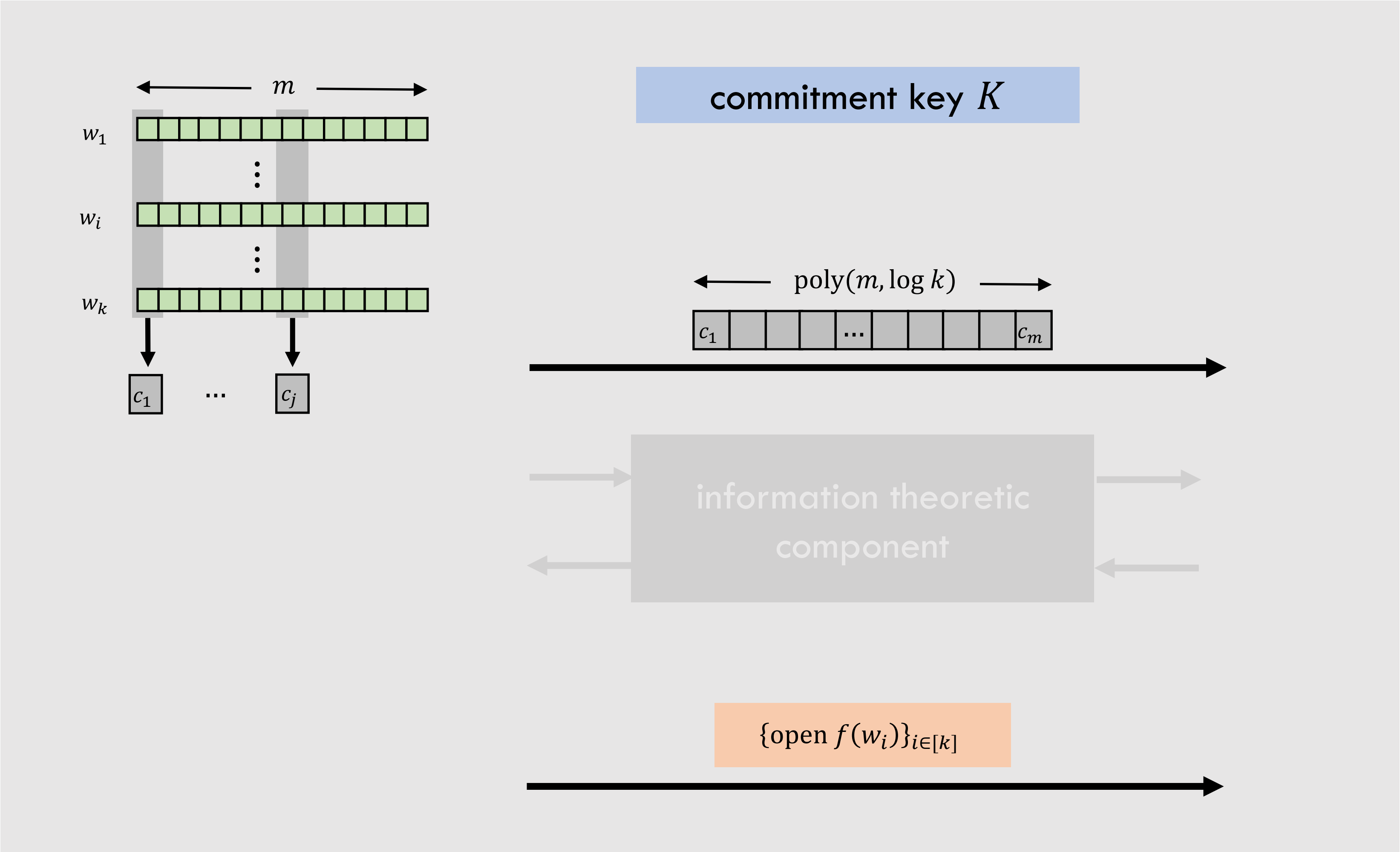
$\forall i \in [k], (C, x_i) \in \mathrm{SAT}$

Dual Mode Batch Argument

Protocol Template

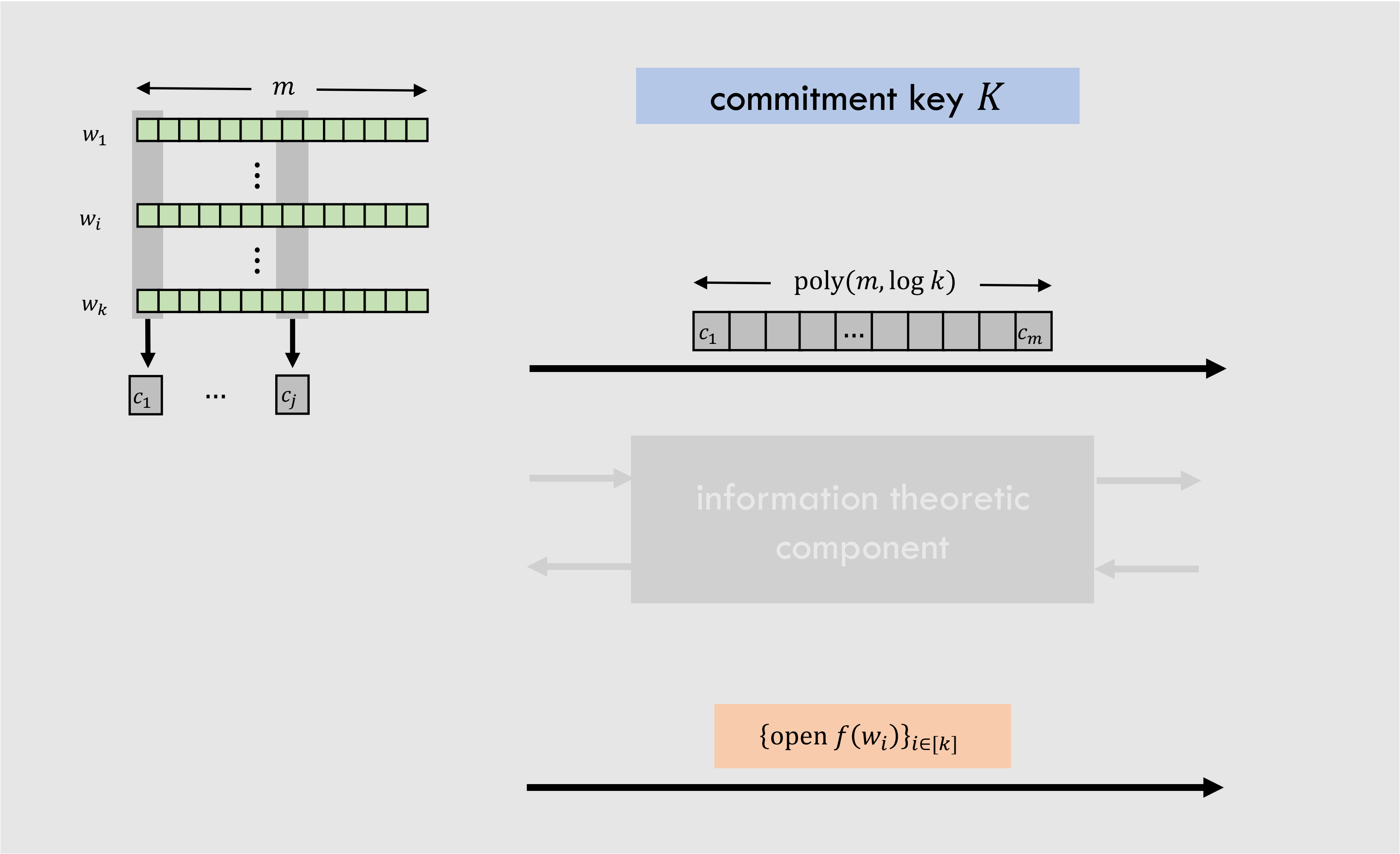
$\mathrm{SAT} = \{(C, x) \mid \exists w \ s.t. \ C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \mathrm{SAT}$



Dual Mode Batch Argument

Protocol Template



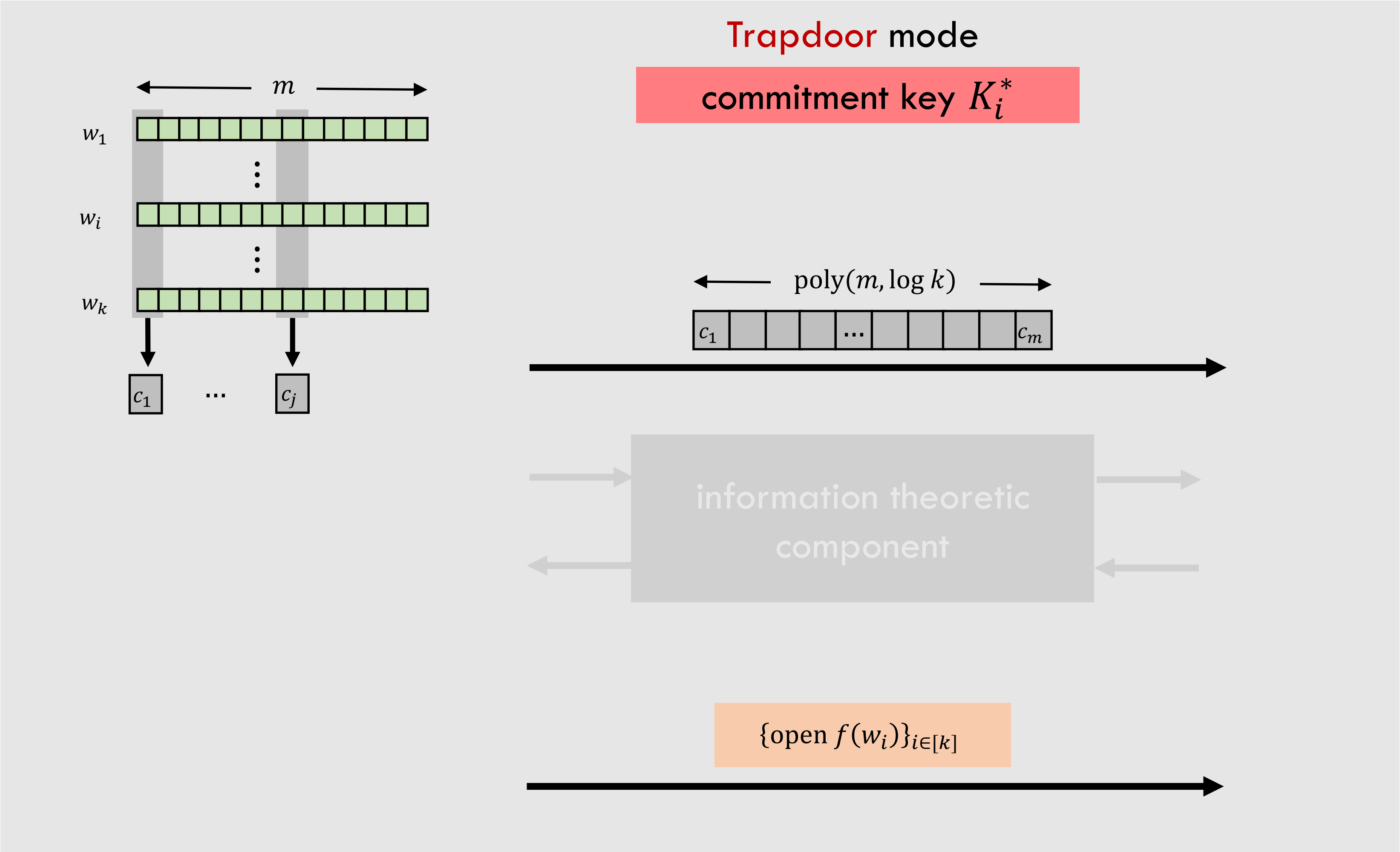
$\text{SAT} = \{(C, x) \mid \exists w \ s.t.\ C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \text{SAT}$

Somewhere Statistically Binding (SSB) Commitment Scheme

Dual Mode Batch Argument

Protocol Template



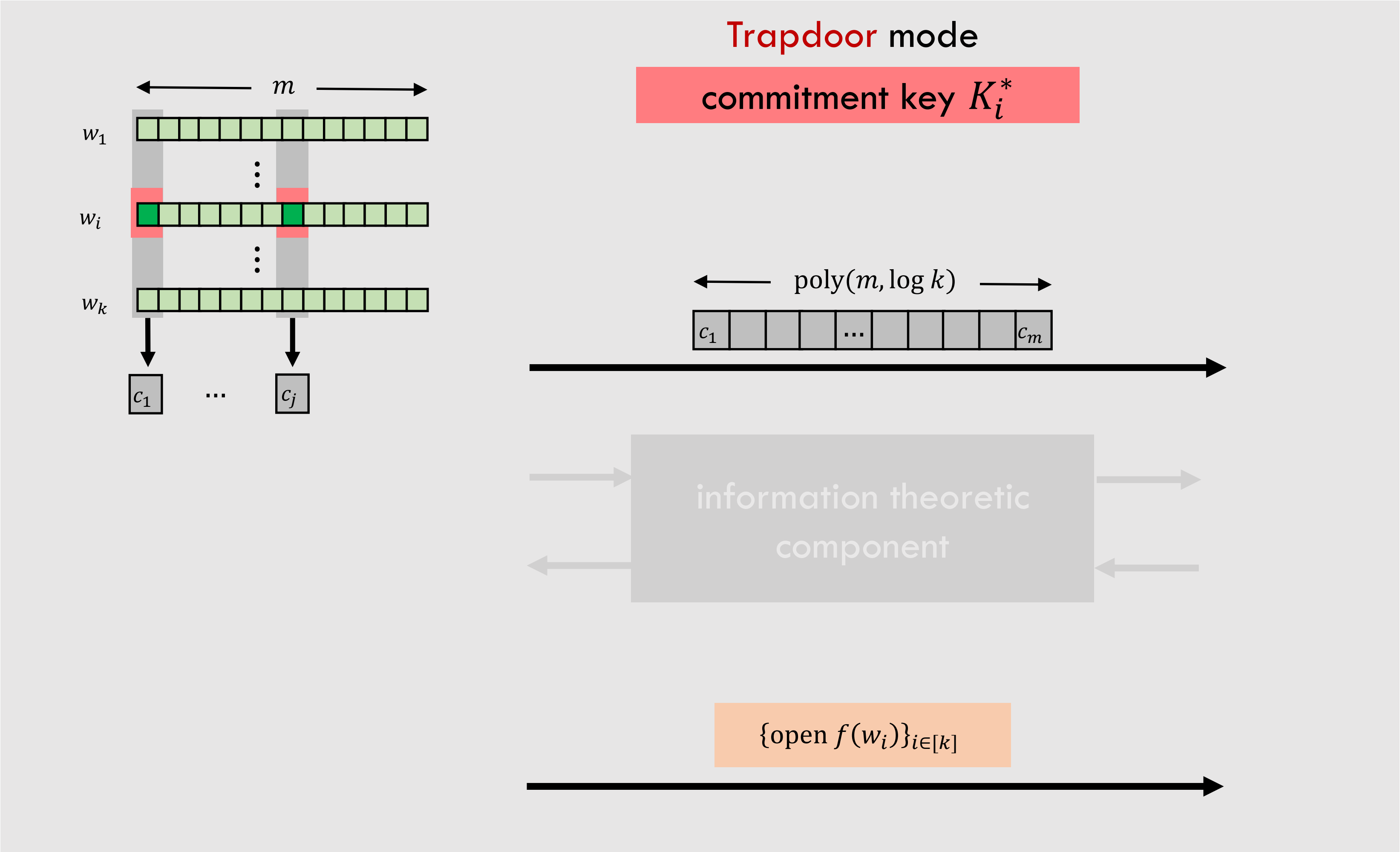
$\text{SAT} = \{(C, x) \mid \exists w \text{ s.t. } C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \text{SAT}$

Somewhere Statistically Binding (SSB) Commitment Scheme

Dual Mode Batch Argument

Protocol Template



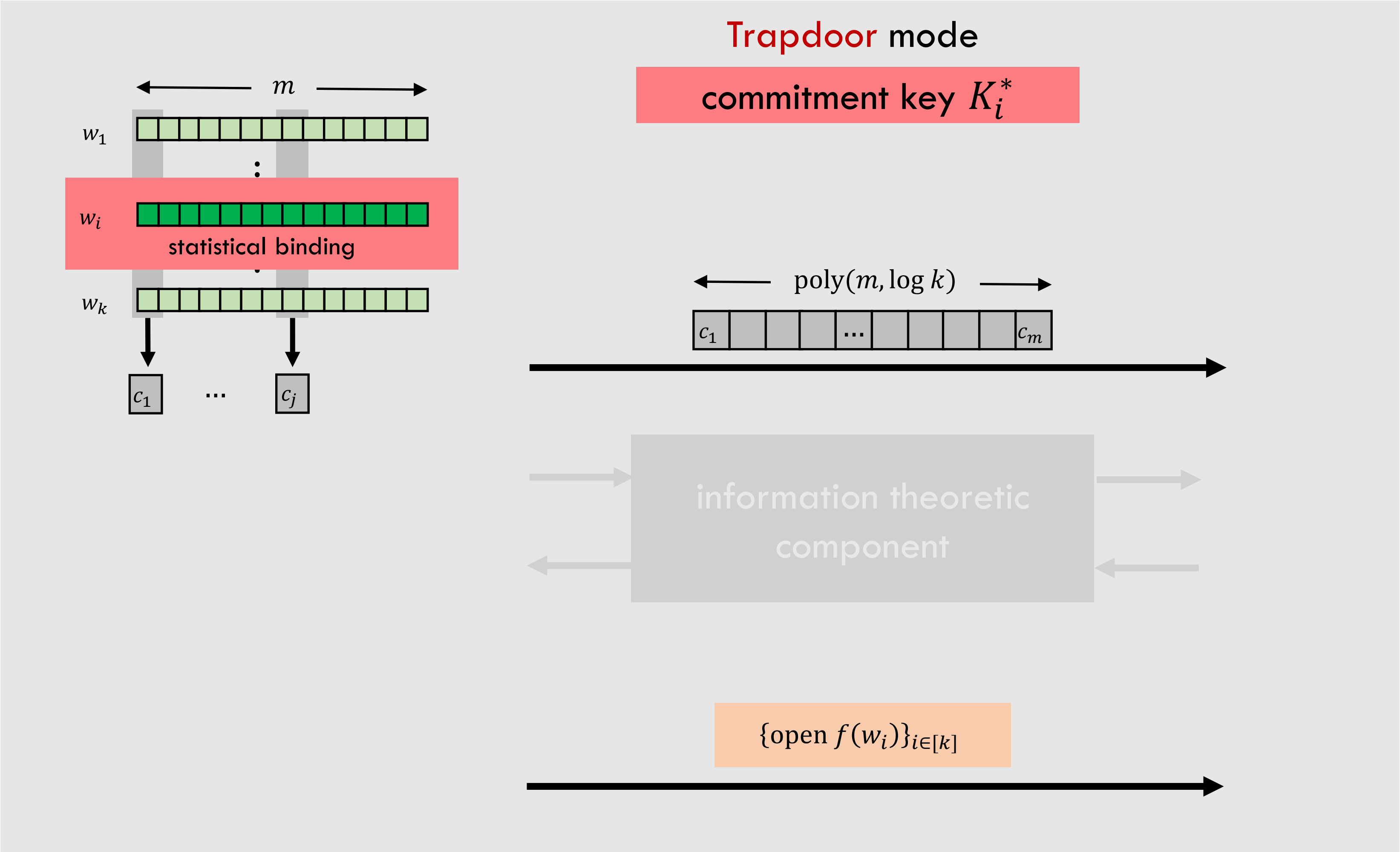
$\text{SAT} = \{(C, x) \mid \exists w \ s.t. \ C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \text{SAT}$

Somewhere Statistically Binding (SSB) Commitment Scheme

Dual Mode Batch Argument

Protocol Template



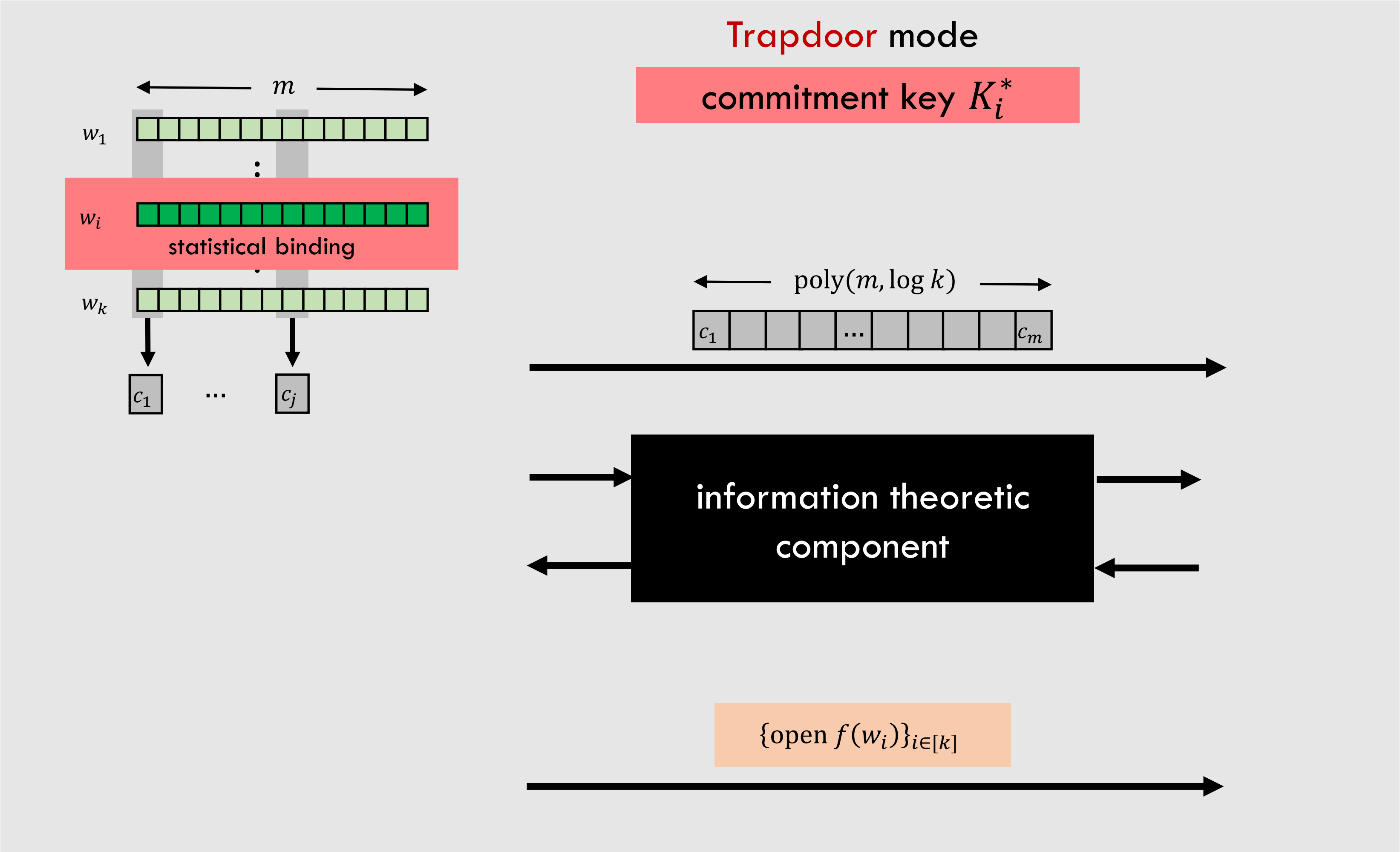
$\mathrm{SAT} = \{(C, x) \mid \exists w \ s.t.\ C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \mathrm{SAT}$

Somewhere Statistically Binding (SSB) Commitment Scheme

Dual Mode Batch Argument

Protocol Template



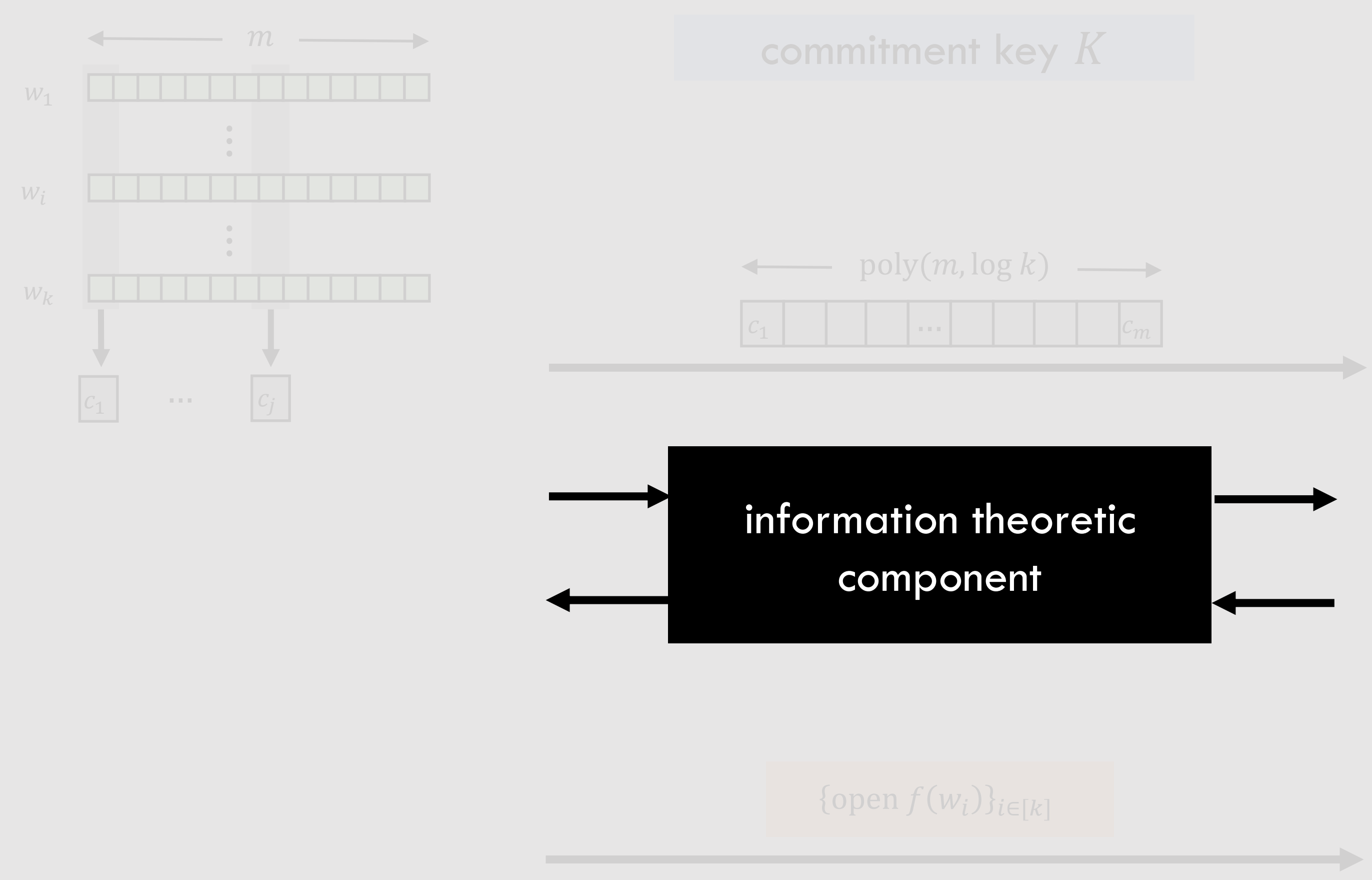
$\mathrm{SAT} = \{(C, x) \mid \exists w \ s.t. \ C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \mathrm{SAT}$

Somewhere Statistically Binding (SSB) Commitment Scheme

Dual Mode Batch Argument

Protocol Template



$\mathrm{SAT} = \{(C, x) \mid \exists w\ s.t.\ C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \mathrm{SAT}$

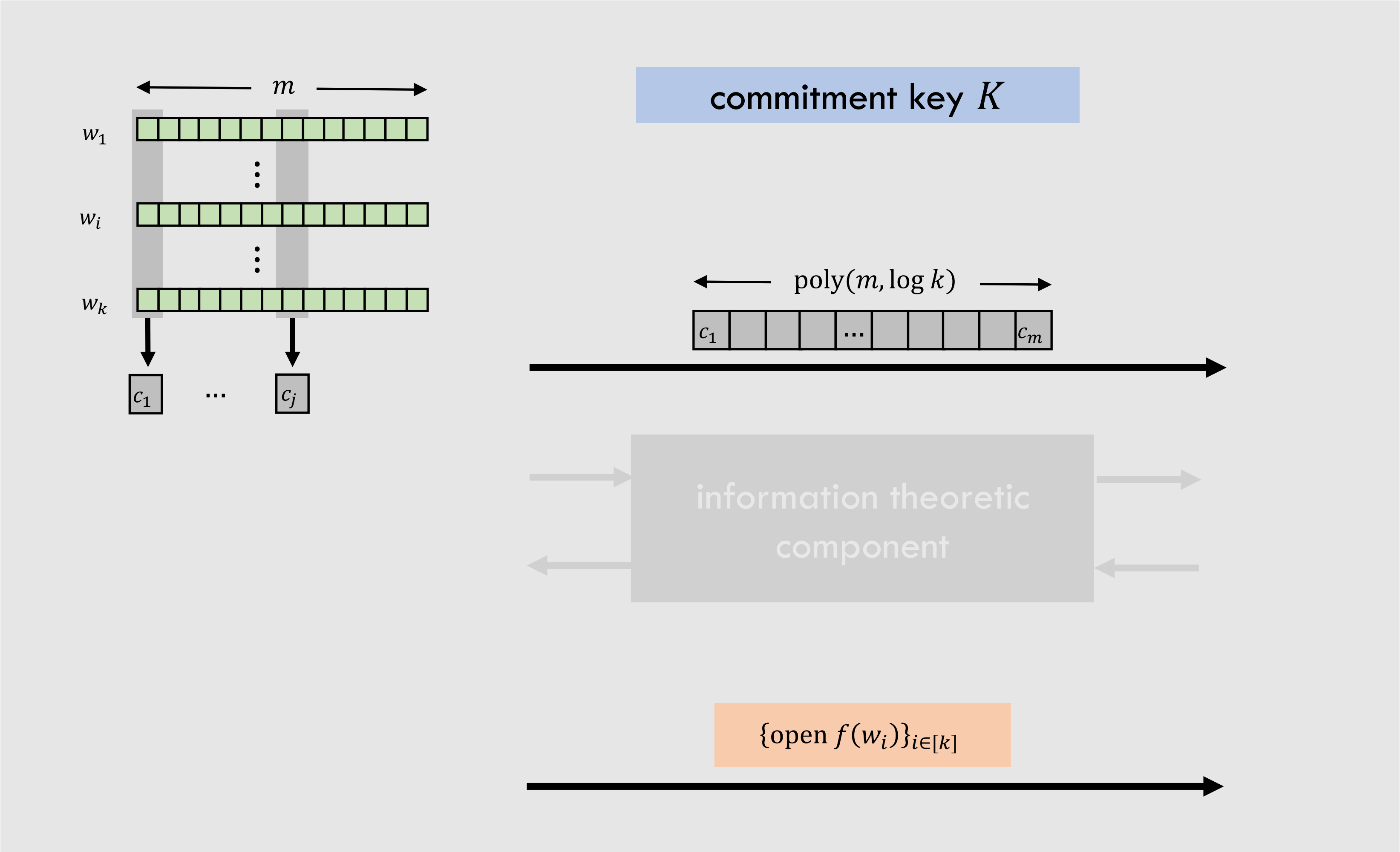
Somewhere Statistically Binding (SSB) Commitment Scheme

Needs to be Fiat-Shamir friendly.

Based on LWE/sub-exp DDH

Dual Mode Batch Argument

Protocol Template



$\text{SAT} = \{(C, x) \mid \exists w \; s.t. \; C(x, w) = 1\}$

$\forall i \in [k], (C, x_i) \in \text{SAT}$

Somewhere Statistically Binding (SSB) Commitment Scheme

Needs to be Fiat-Shamir friendly.

Based on LWE/sub-exp DDH

SSB with appropriate opening to f

[CJJ'21a]: (with additional properties) based on QR

[CJJ'21b]: based on LWE

BARGs

	Proof size	Assumptions
[C-Jain-Jin'21a]	$\tilde{O}(\lvert C\rvert + \sqrt{k\lvert C\rvert})$	QR + (LWE/sub-exp DDH)
[C-Jain-Jin'21b]	$\mathrm{poly}(\log k, \log\lvert C^*\rvert, \lvert w\rvert)$	LWE

SNARGs

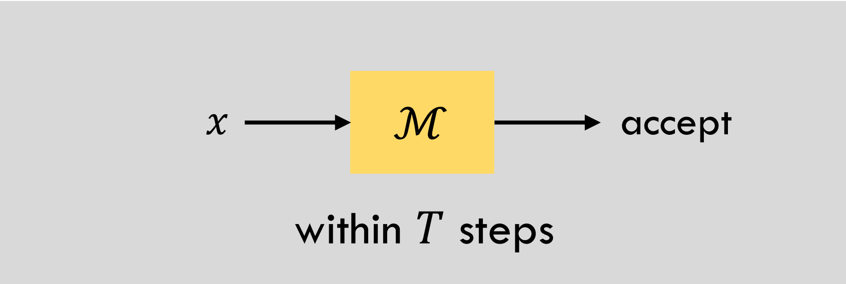
	Model	Assumptions
[C-Jain-Jin'21b]	RAM	LWE

Thank you. Questions?

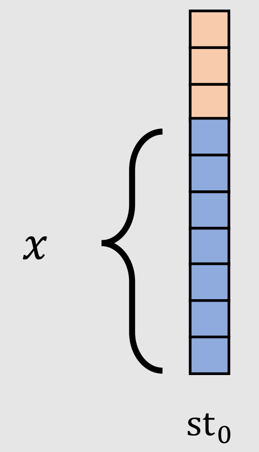
Arka Rai Choudhuri

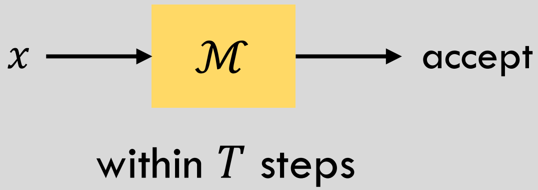
achoud@cs.jhu.edu

SNARGs for Polynomial-time Computation

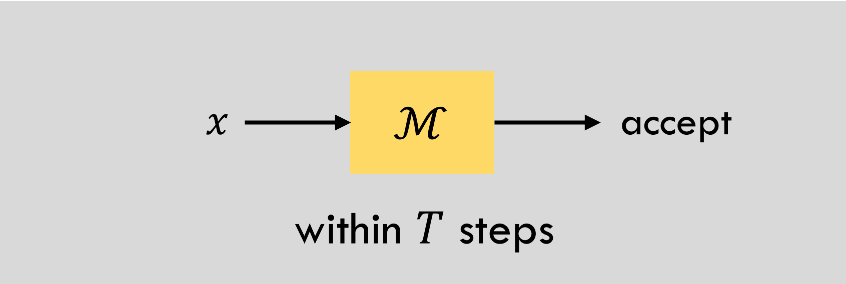


SNARGs for Polynomial-time Computation

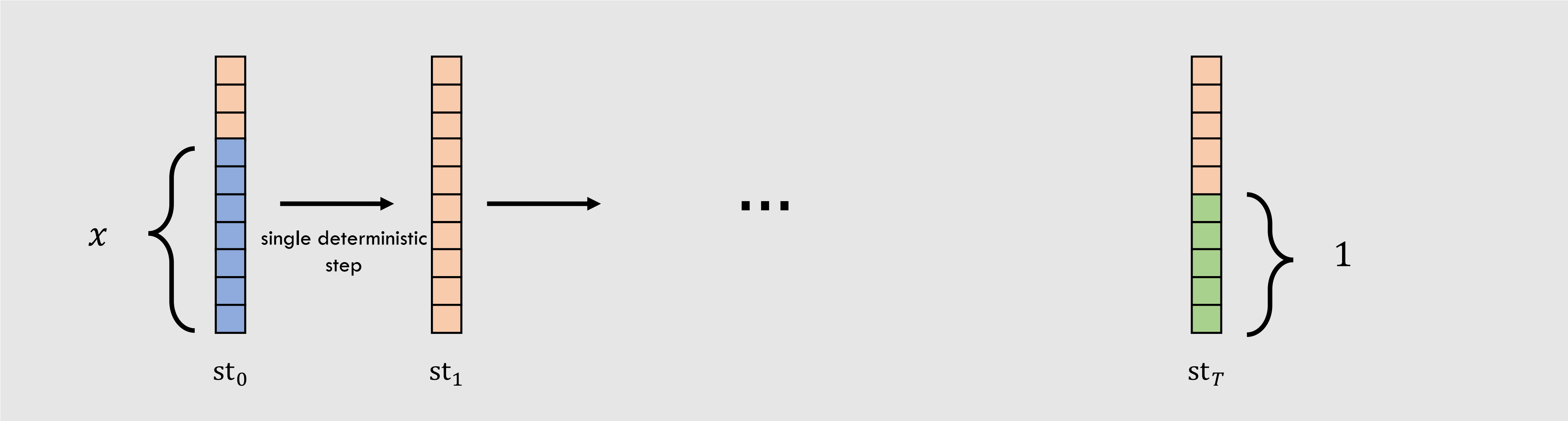


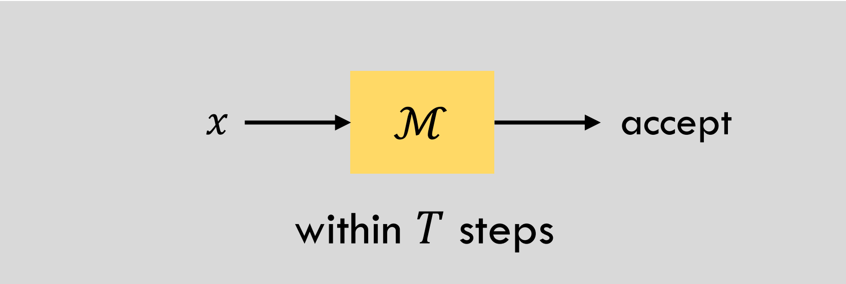


SNARGs for Polynomial-time Computation

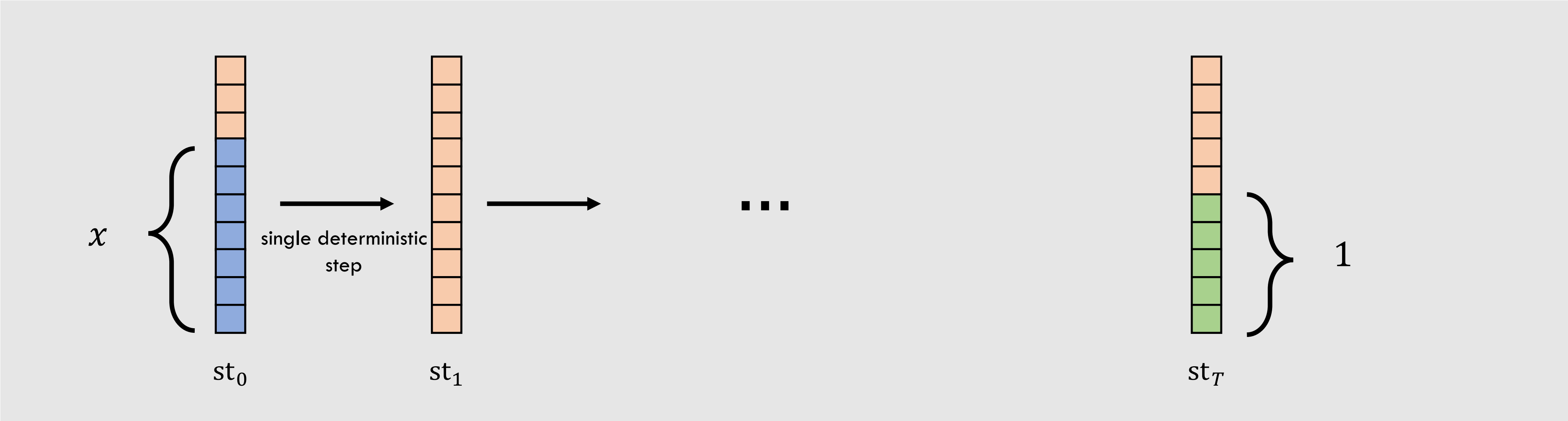


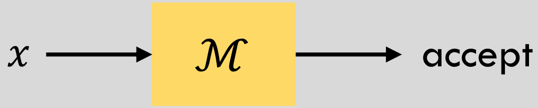
SNARGs for Polynomial-time Computation





SNARGs for Polynomial-time Computation





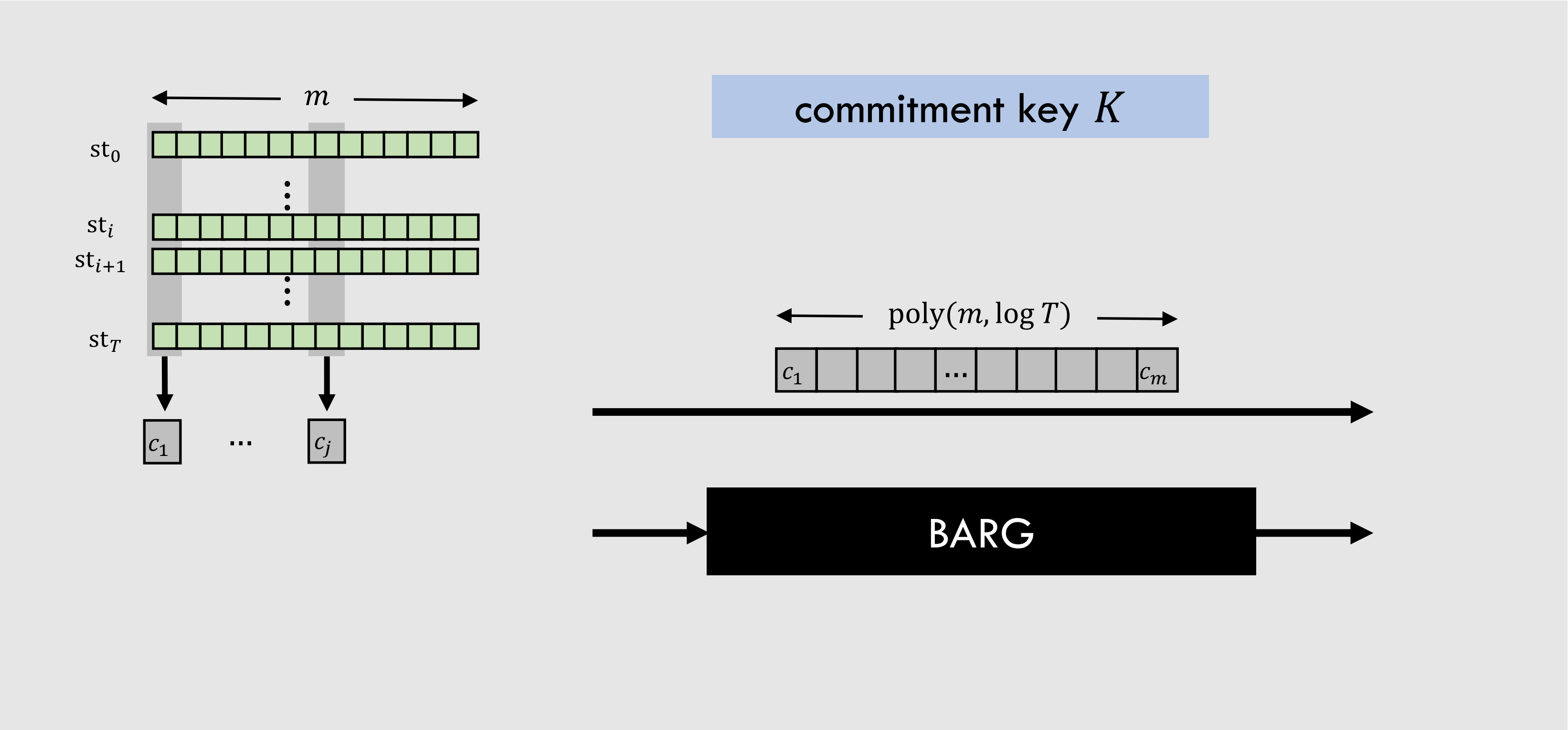
within T steps

Prove for every $i \in [0, \ldots, T-1]$

$$\mathrm{st}_i \to \mathrm{st}_{i+1}$$

is the correct transition.

SNARGs for Polynomial-time Computation

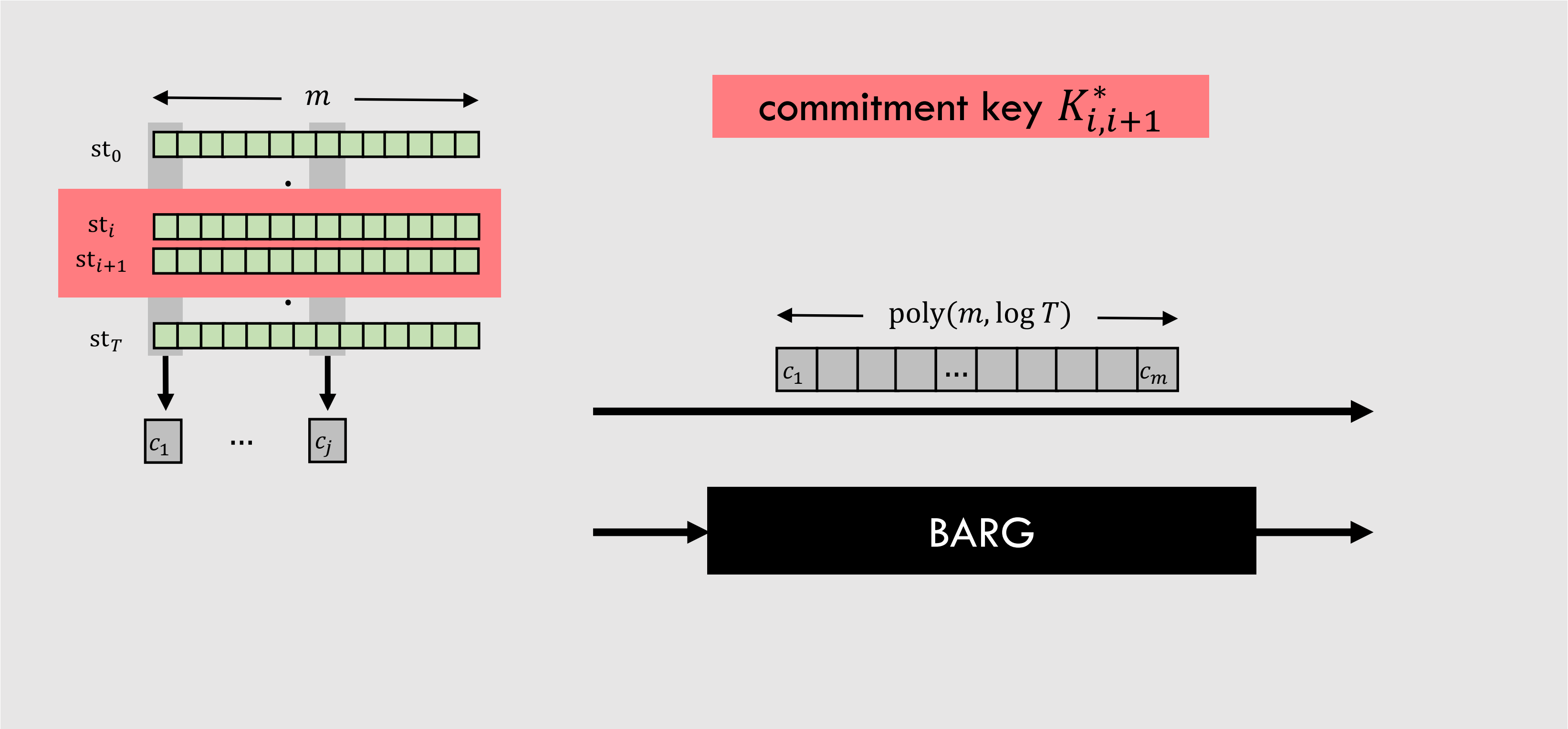


BARG

For every $i \in [0, \dots, T-1]$

1. Commitment contains st_i and st_{i+1}
2. Valid transition $\mathrm{st}_i \to \mathrm{st}_{i+1}$

SNARGs for Polynomial-time Computation

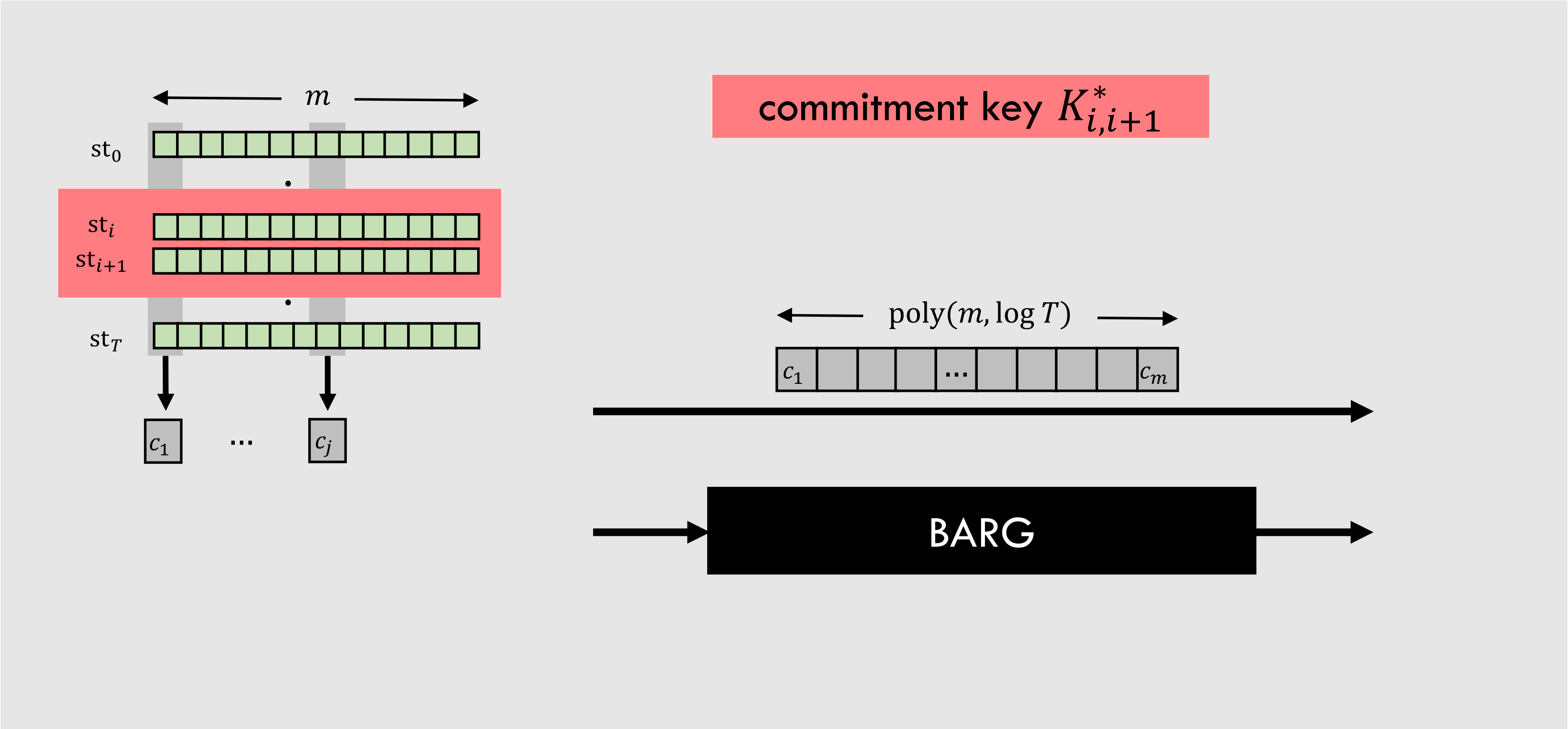


BARG

For every $i \in [0, \ldots, T-1]$

1. Commitment contains st_i and st_{i+1}
2. Valid transition $\mathsf{st}_i \rightarrow \mathsf{st}_{i+1}$

SNARGs for Polynomial-time Computation



BARG

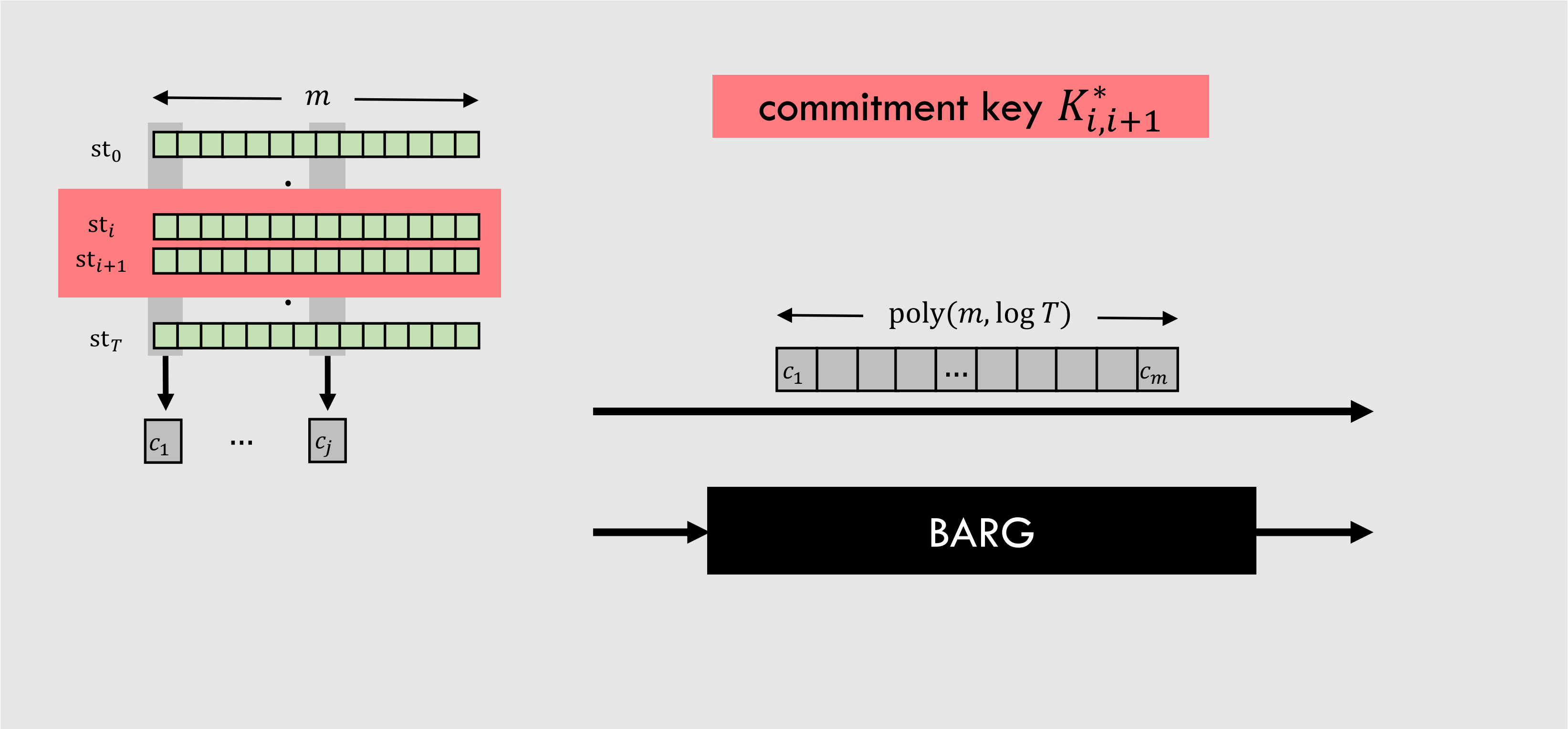
For every $i \in [0, \ldots, T-1]$

1. Commitment contains st_i and st_{i+1}
2. Valid transition $\mathsf{st}_i \to \mathsf{st}_{i+1}$

Local Soundness

i-th state transition correct

SNARGs for Polynomial-time Computation



BARG

For every $i \in [0, \dots, T-1]$

1. Commitment contains st_i and st_{i+1}
2. Valid transition $\mathrm{st}_i \to \mathrm{st}_{i+1}$

Local Soundness

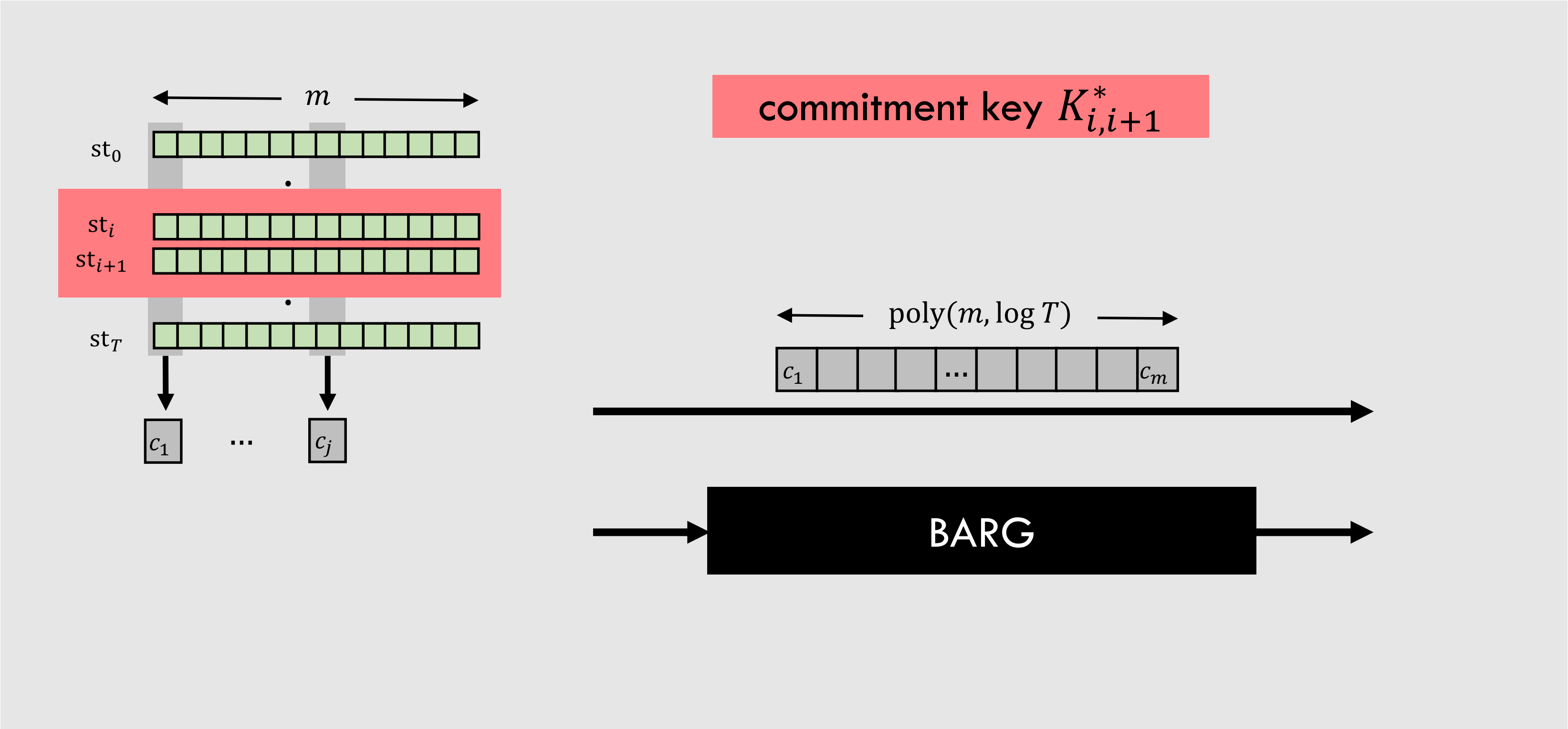
i-th state transition correct

→

Global Soundness

Local soundness at all i

SNARGs for Polynomial-time Computation



No-Signaling Somewhere Statistically Binding (SSB) Commitment Scheme [González-Zacharakis'21]

BARG

For every $i \in [0, \dots, T-1]$

1. Commitment contains st_i and st_{i+1}
2. Valid transition $\mathsf{st}_i \to \mathsf{st}_{i+1}$

Local Soundness

i-th state transition correct

→

Global Soundness

Local soundness at all i