Toward Malicious Constant-Rate 2PC via Arithmetic Garbling

Carmit Hazay, Bar-Ilan University
Yibin Yang, Georgia Tech

Toward Malicious Constant-Rate 2PC via Arithmetic **Garbling**

Garbling Scheme [BHR12]

Garbling Scheme [BHR12]

A garbling scheme consists of four procedures:

- **A garbling procedure:** $\mathtt{Gb}(1^\kappa, \mathrm{C}) \to \hat{\mathrm{C}}, e, d$
- **An encoding procedure:** $\mathtt{En}(e, \overrightarrow{x}) \to \overrightarrow{X}$
- **An evaluation procedure:** $\mathtt{Ev}(\hat{\mathrm{C}}, \overrightarrow{X}) \to \overrightarrow{Y}$
- **A decoding procedure:** $\mathtt{De}(d, \overrightarrow{Y}) \to \overrightarrow{y}$

Garbling Scheme [BHR12]

A garbling scheme consists of four procedures:

- **A garbling procedure:** $\mathtt{Gb}(1^\kappa, \mathrm{C}) \rightarrow \hat{\mathrm{C}}, e, d$
- **An encoding procedure:** $\mathtt{En}(e, \overrightarrow{x}) \rightarrow \overrightarrow{X}$
- **An evaluation procedure:** $\mathtt{Ev}(\hat{\mathrm{C}}, \overrightarrow{X}) \rightarrow \overrightarrow{Y}$
- **A decoding procedure:** $\mathtt{De}(d, \overrightarrow{Y}) \rightarrow \overrightarrow{y}$

Correctness Obliviousness **Privacy** Authenticity

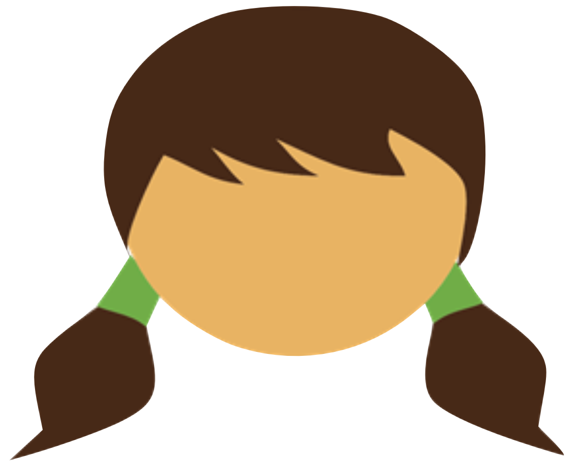
Toward Malicious Constant-Rate
2PC via Arithmetic Garbling

2PC

2PC

$$C(\overrightarrow{x_0}, \overrightarrow{x_1})$$

$\overrightarrow{x_0}$



Alice

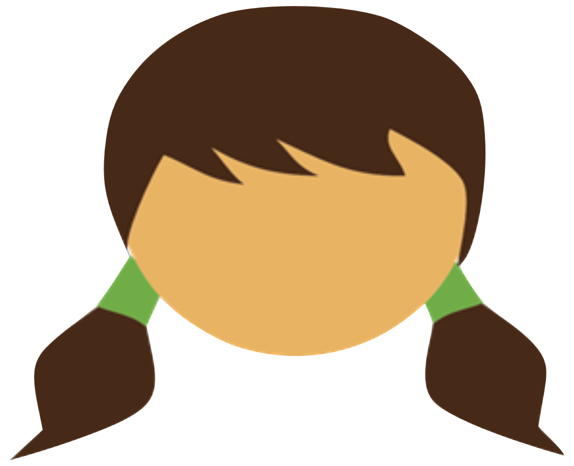
$\overrightarrow{x_1}$

Bob

Semi-Honest 2PC

$$\mathrm{C}(\overrightarrow{x_0}, \overrightarrow{x_1})$$

$\overrightarrow{x_0}$



Garbler

$\overrightarrow{x_1}$

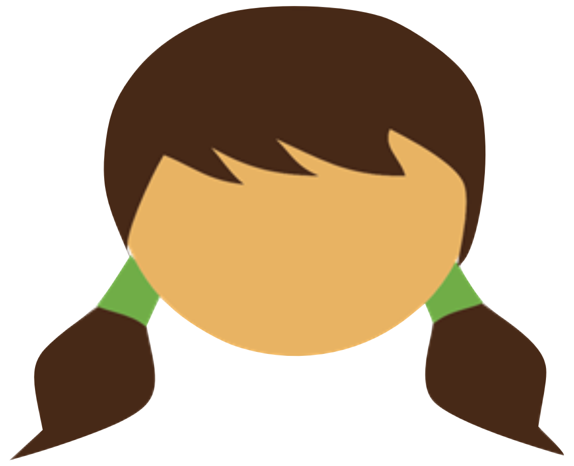
Evaluator

Semi-Honest 2PC

$$\mathrm{C}(\overrightarrow{x_0}, \overrightarrow{x_1})$$

$\boxed{\hat{\mathrm{C}}}$

$\overrightarrow{x_0}$



Garbler

$\overrightarrow{x_1}$

Evaluator

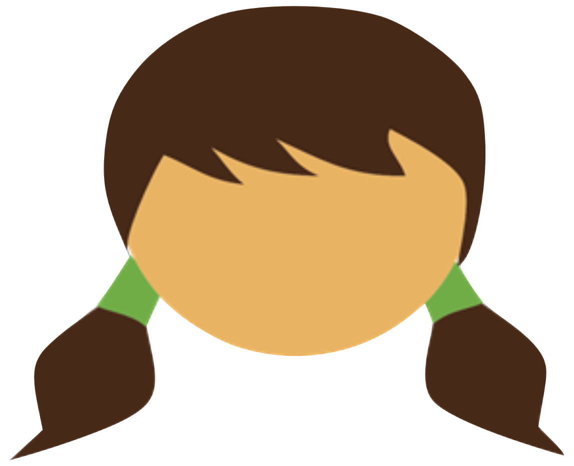
Semi-Honest 2PC

$C(\overrightarrow{x_0}, \overrightarrow{x_1})$

$\boxed{\hat{C}}$

$\boxed{\hat{C}}$ $\overrightarrow{X_0}$

$\overrightarrow{x_0}$

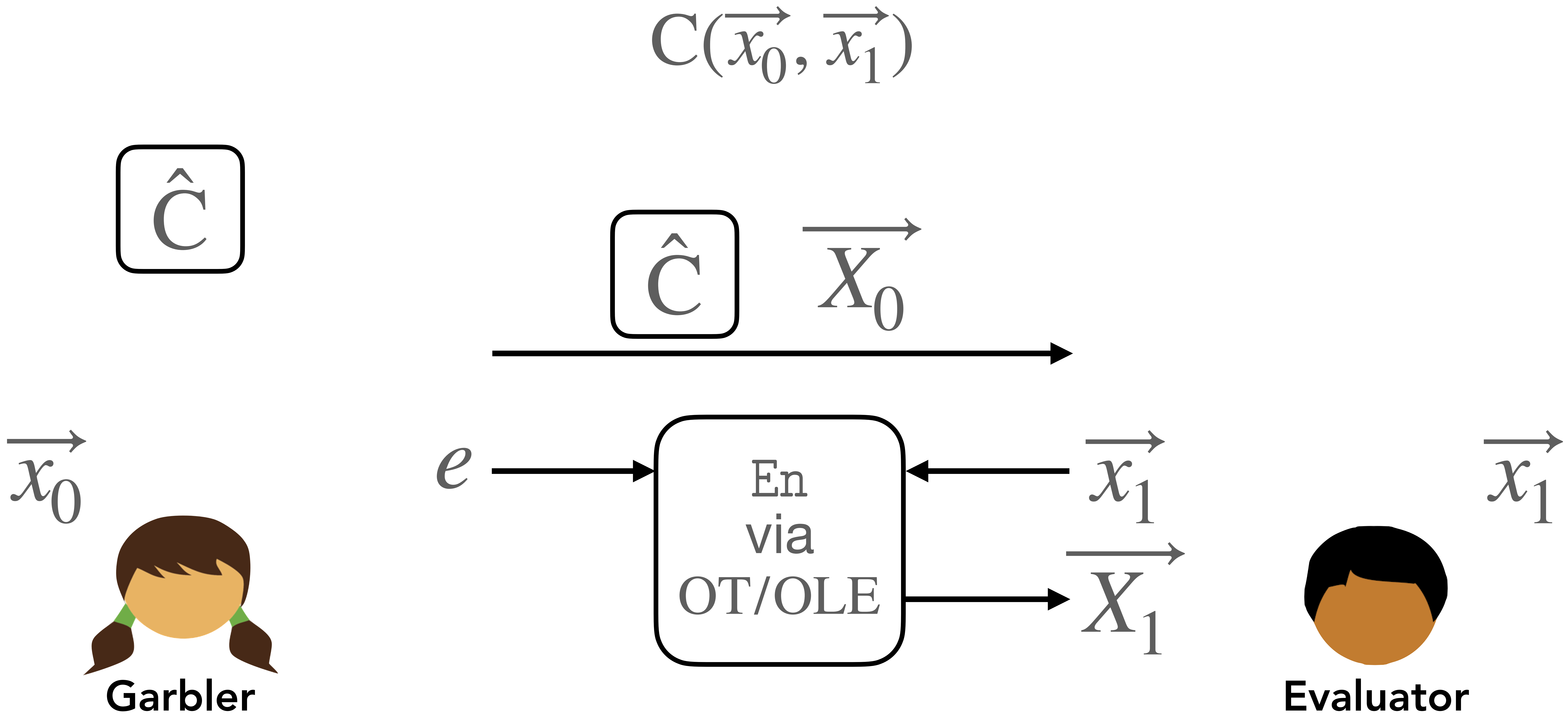


Garbler

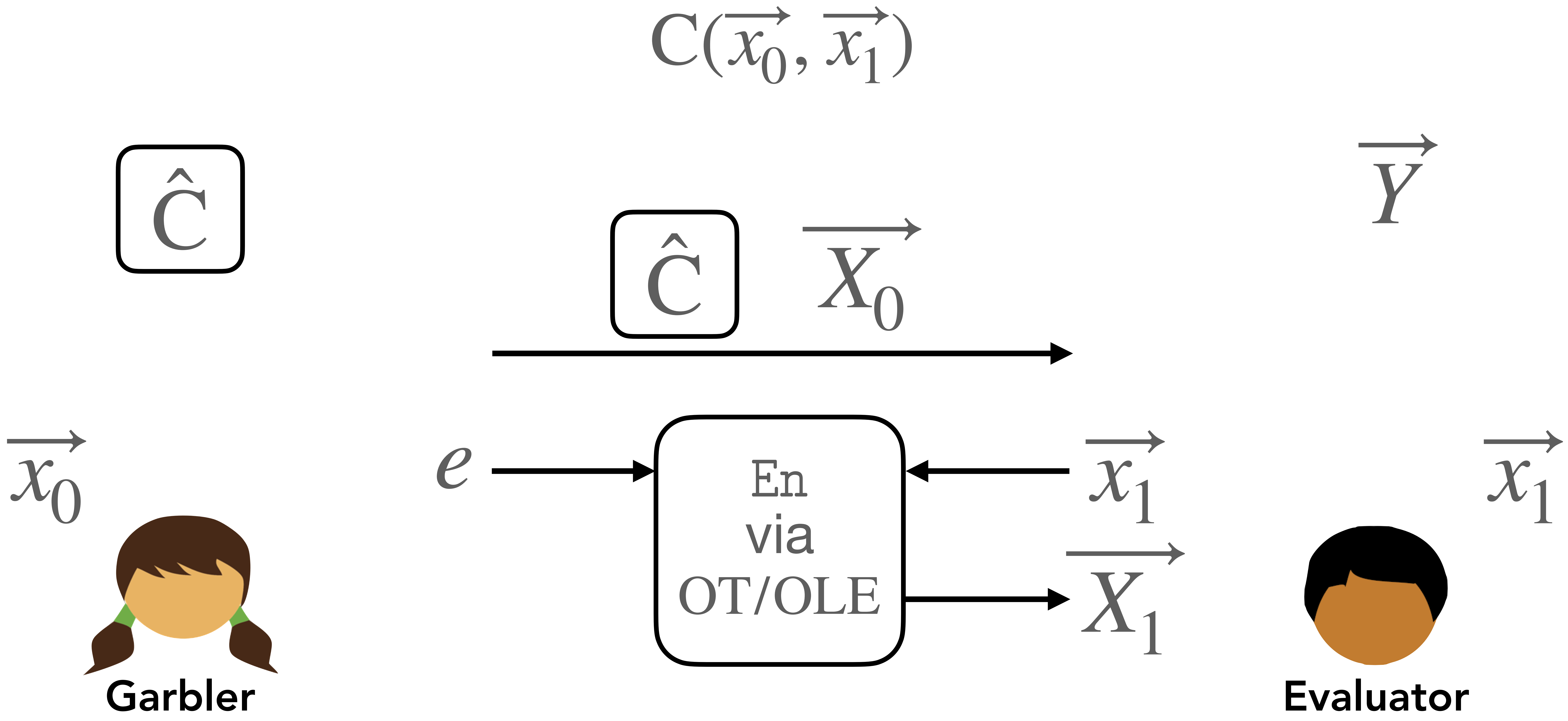
$\overrightarrow{x_1}$

Evaluator

Semi-Honest 2PC

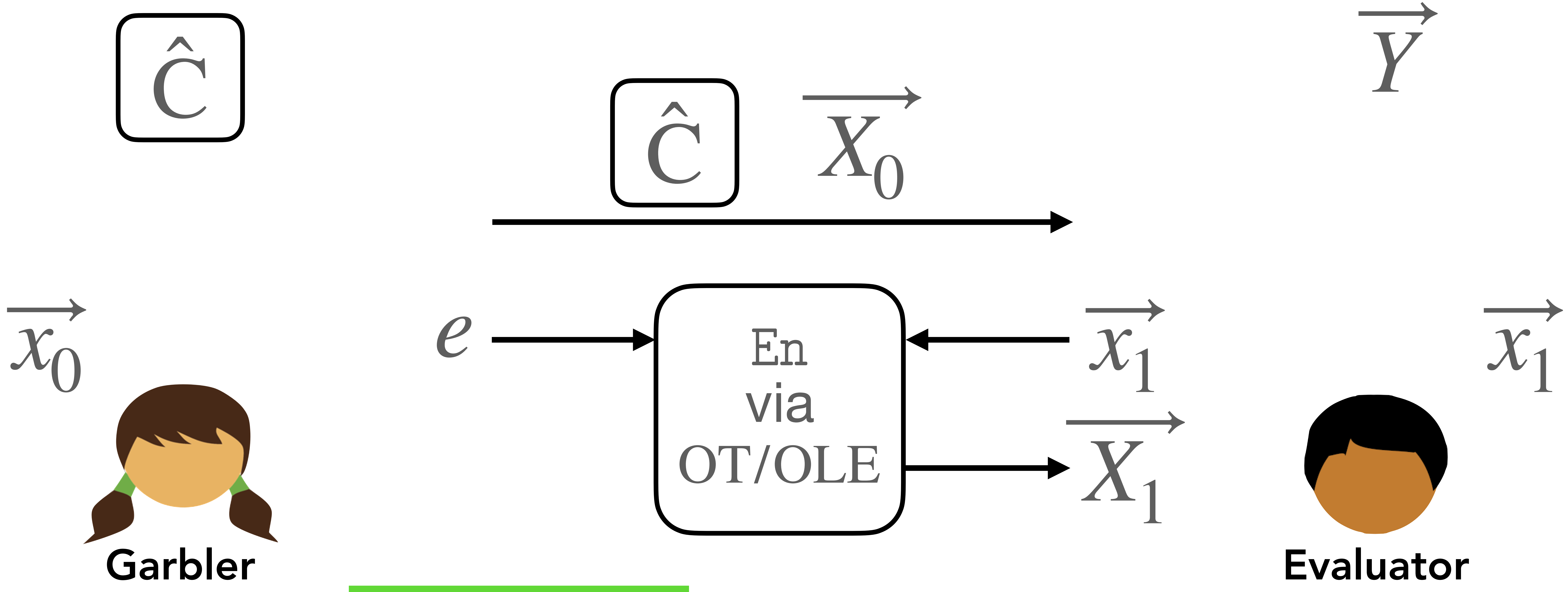


Semi-Honest 2PC



Semi-Honest 2PC

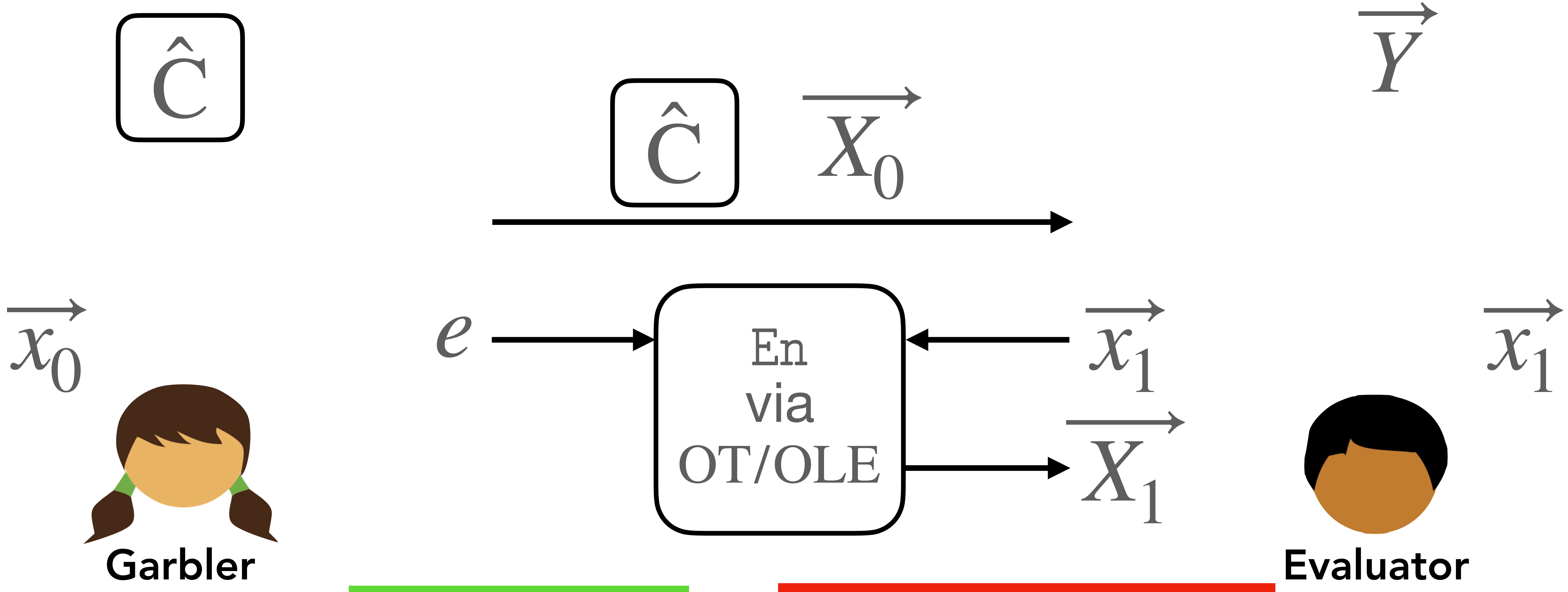
$$\mathrm{C}(\overrightarrow{x_0}, \overrightarrow{x_1})$$



constant-round

Semi-Honest 2PC

$$\mathrm{C}(\overrightarrow{x_0}, \overrightarrow{x_1})$$



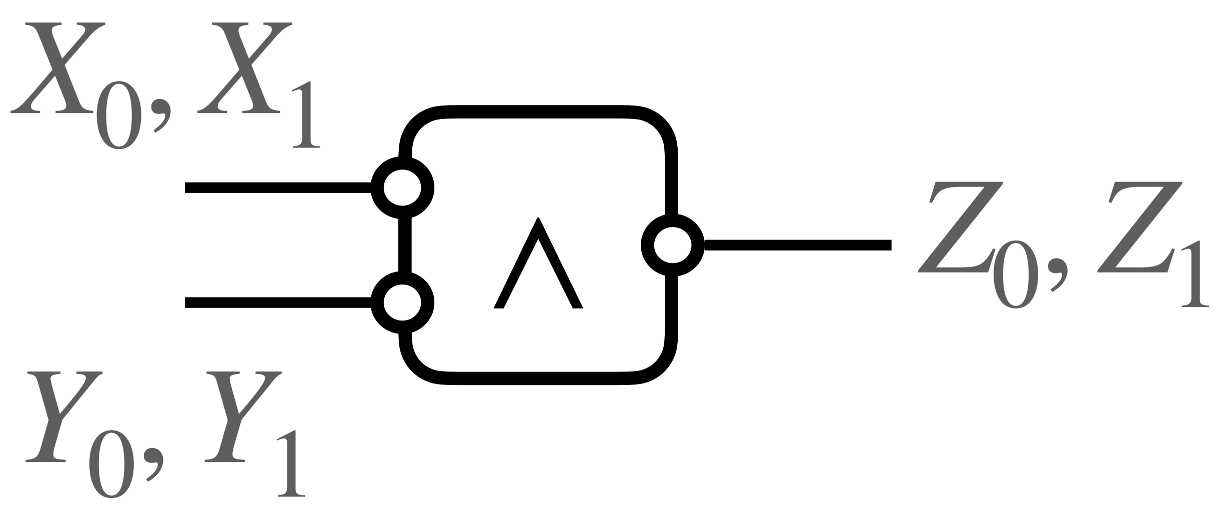
constant-round

$\tilde{O}(|\mathrm{C}|)$-communication

Garbled <u>Boolean</u> Circuits [Yao86, LP09]

Garbled Boolean Circuits [Yao86, LP09]

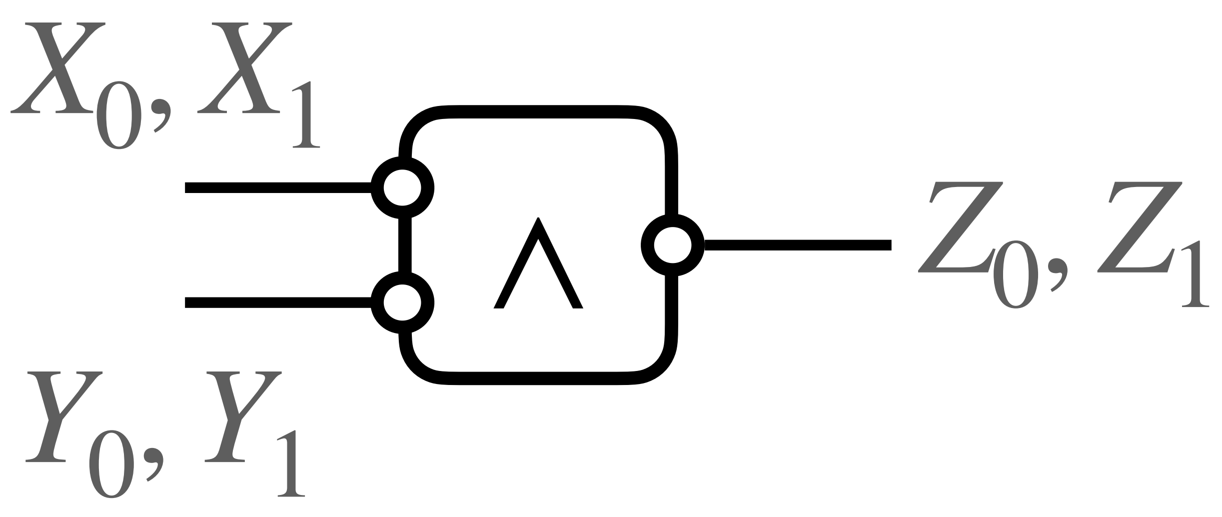
Garbled truth tables gate by gate



$$\mathtt{Enc}_{X_0,Y_0}(Z_0)$$
$$\mathtt{Enc}_{X_0,Y_1}(Z_0)$$
$$\mathtt{Enc}_{X_1,Y_0}(Z_0)$$
$$\mathtt{Enc}_{X_1,Y_1}(Z_1)$$

Garbled Boolean Circuits [Yao86, LP09]

Garbled truth tables gate by gate

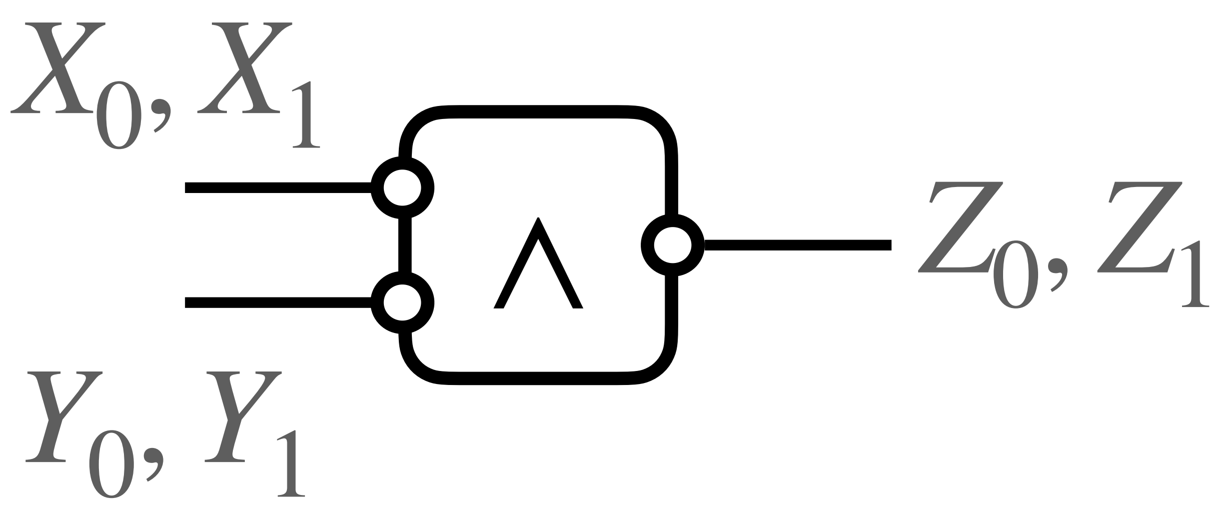


$$\mathtt{Enc}_{X_0,Y_0}(Z_0)$$
$$\mathtt{Enc}_{X_0,Y_1}(Z_0)$$
$$\mathtt{Enc}_{X_1,Y_0}(Z_0)$$
$$\mathtt{Enc}_{X_1,Y_1}(Z_1)$$

	$\wedge$	$\oplus$
[Yao86, LP09]	4κ	4κ

Garbled Boolean Circuits [Yao86, LP09]

Garbled truth tables gate by gate

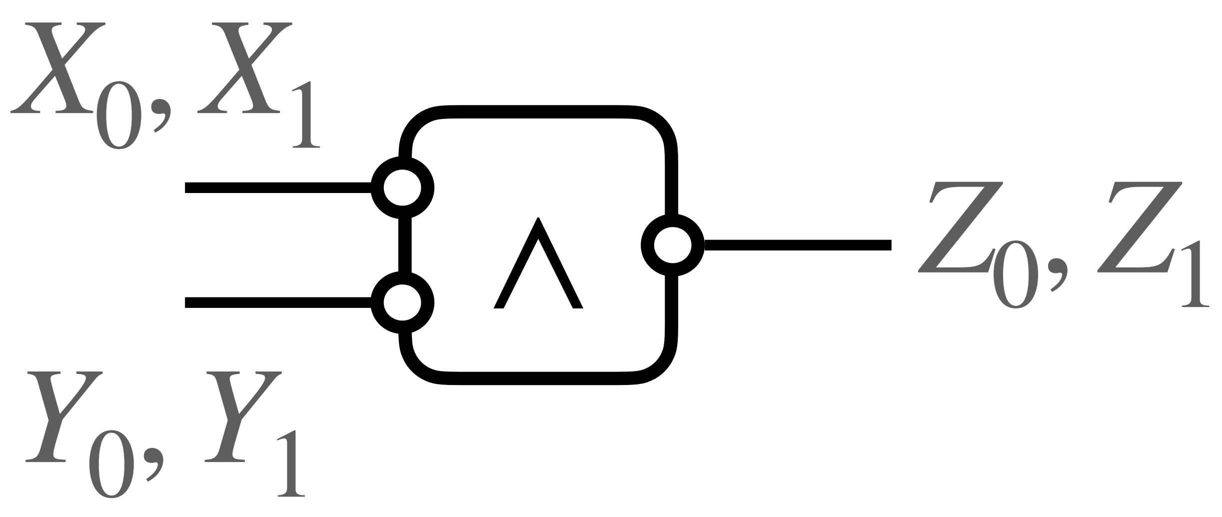


$$\mathtt{Enc}_{X_0,Y_0}(Z_0)$$
$$\mathtt{Enc}_{X_0,Y_1}(Z_0)$$
$$\mathtt{Enc}_{X_1,Y_0}(Z_0)$$
$$\mathtt{Enc}_{X_1,Y_1}(Z_1)$$

	$\boxed{\wedge}$	$\boxed{\oplus}$
[Yao86, LP09]	4κ	4κ
[NPS99]	3κ	3κ
[KS08]	3κ	0
[ZRE15]	2κ	0
[RR21]	1.5κ	0

Garbled <u>Boolean</u> Circuits [Yao86, LP09]

Garbled truth tables gate by gate



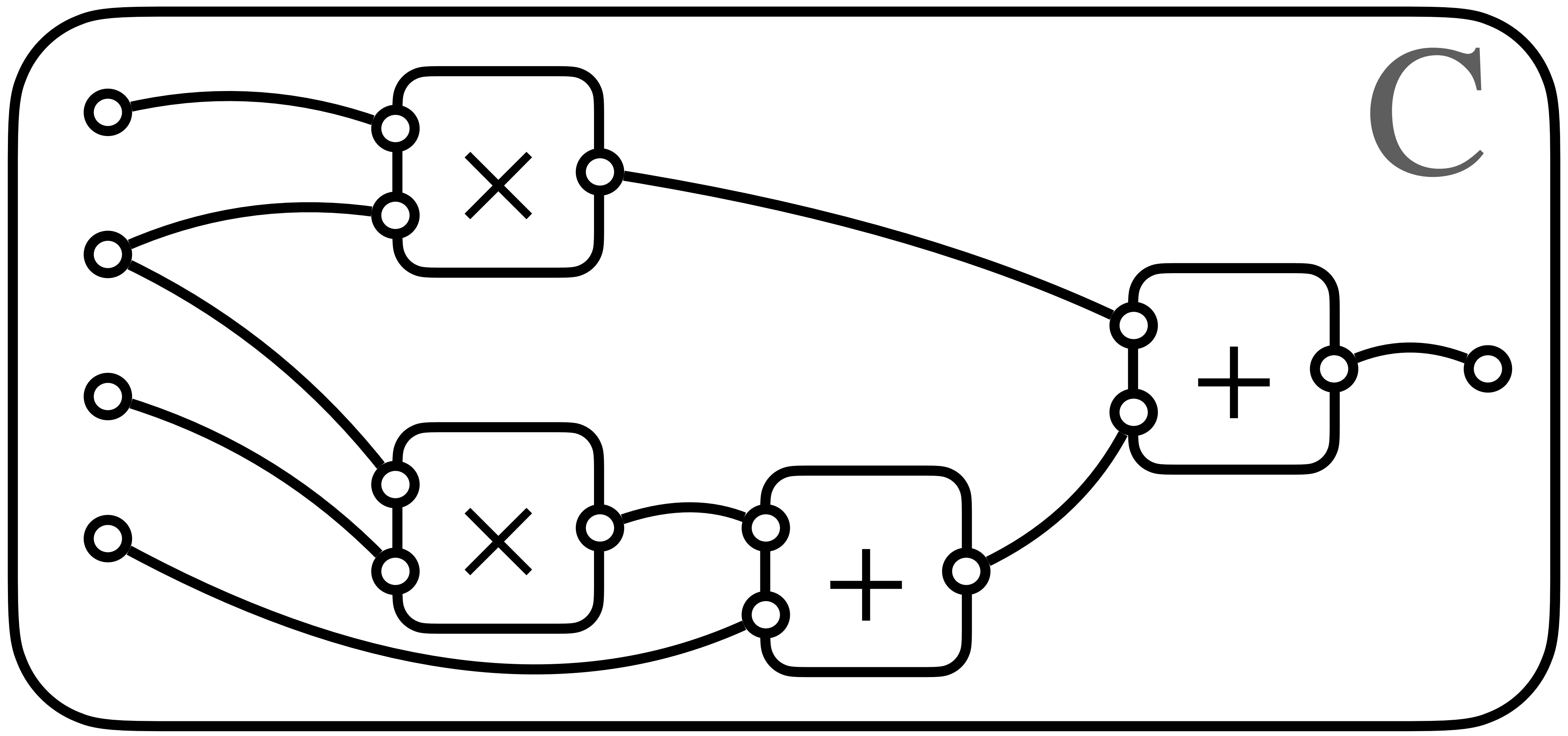
$$\mathtt{Enc}_{X_0,Y_0}(Z_0)$$
$$\mathtt{Enc}_{X_0,Y_1}(Z_0)$$
$$\mathtt{Enc}_{X_1,Y_0}(Z_0)$$
$$\mathtt{Enc}_{X_1,Y_1}(Z_1)$$

	$\wedge$	$\oplus$
[Yao86, LP09]	4κ	4κ
[NPS99]	3κ	3κ
[KS08]	3κ	0
[ZRE15]	2κ	0
[RR21]	1.5κ	0

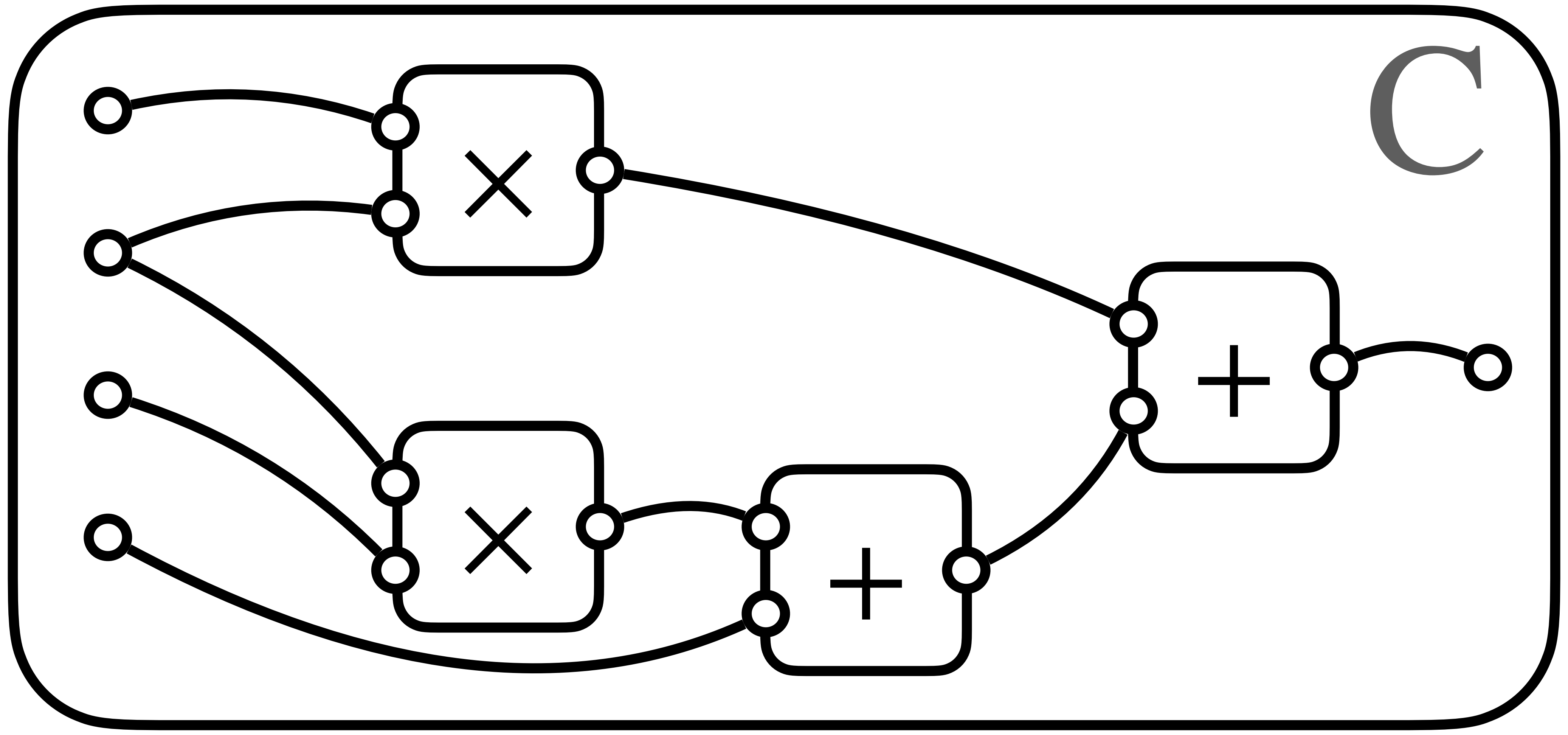
Communication Rate $O(\kappa)$

Toward Malicious Constant-Rate 2PC via **Arithmetic Garbling**

Garbled Arithmetic Circuits

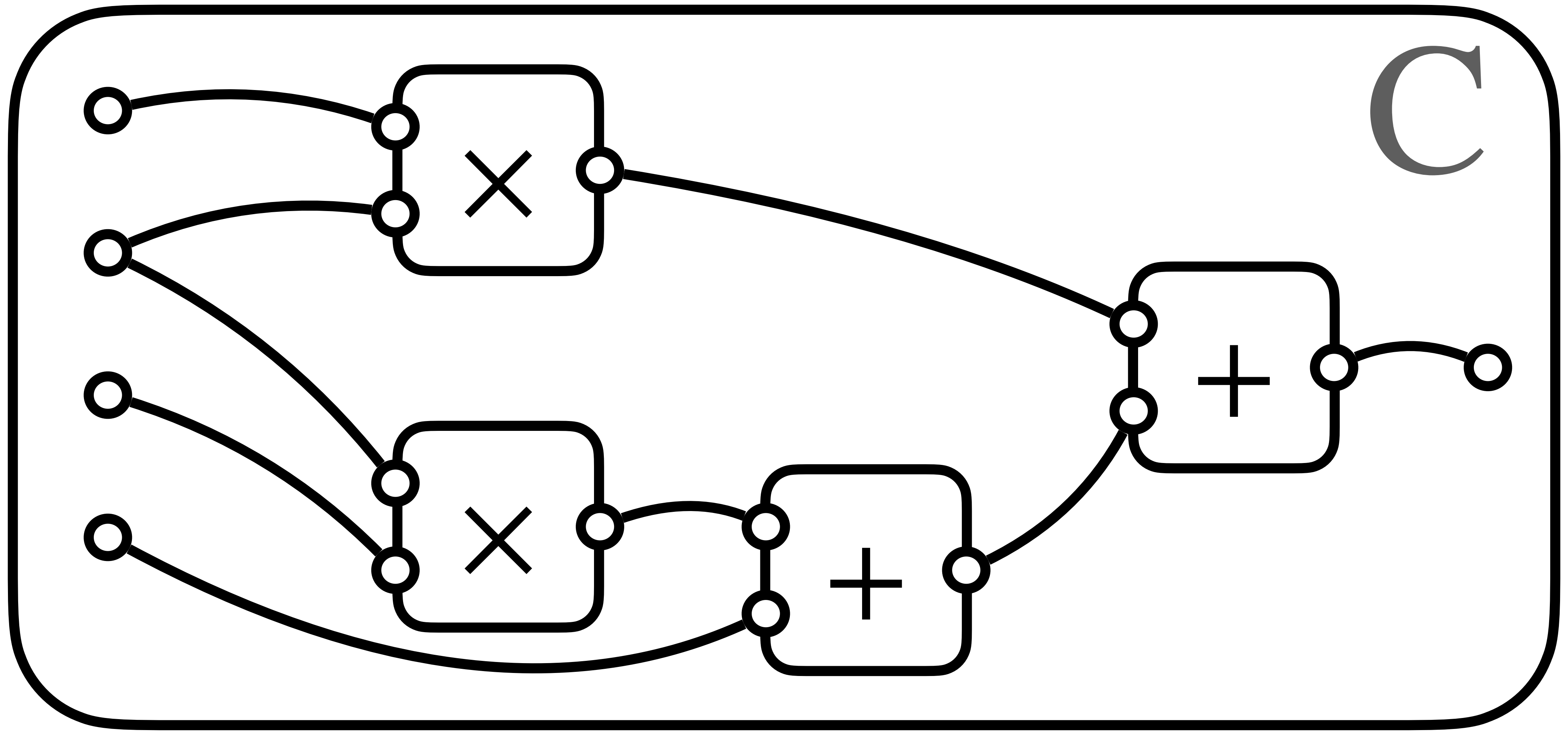


Garbled Arithmetic Circuits



[AIK11], [BMR16], [BLLL23], [LL24], [Heath24]

Garbled Arithmetic Circuits

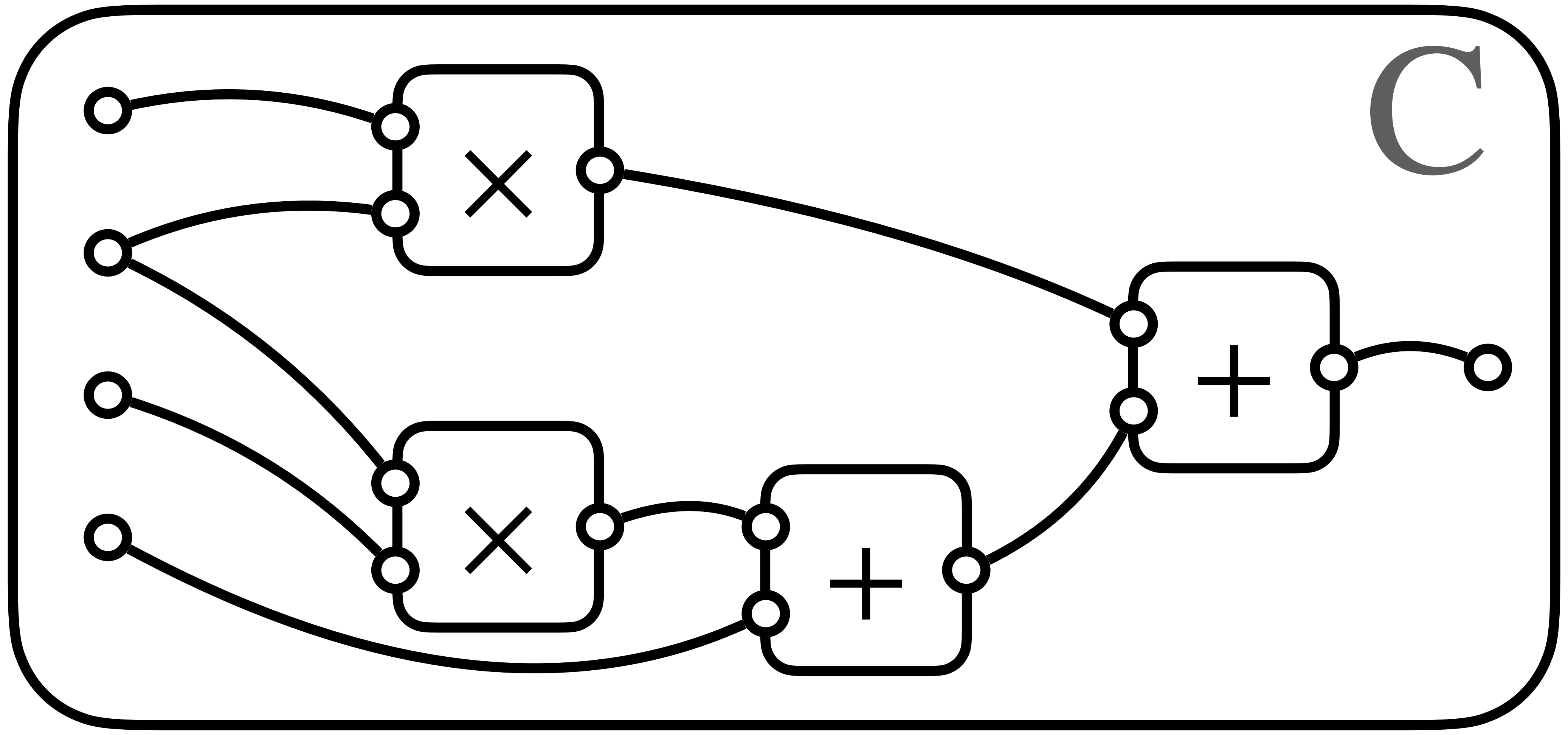


$O(1)$

[AIK11], [BMR16], **[BLLL23]**, [LL24], [Heath24]

Garbled Arithmetic Circuits

Bounded Integer Computation (BIC)

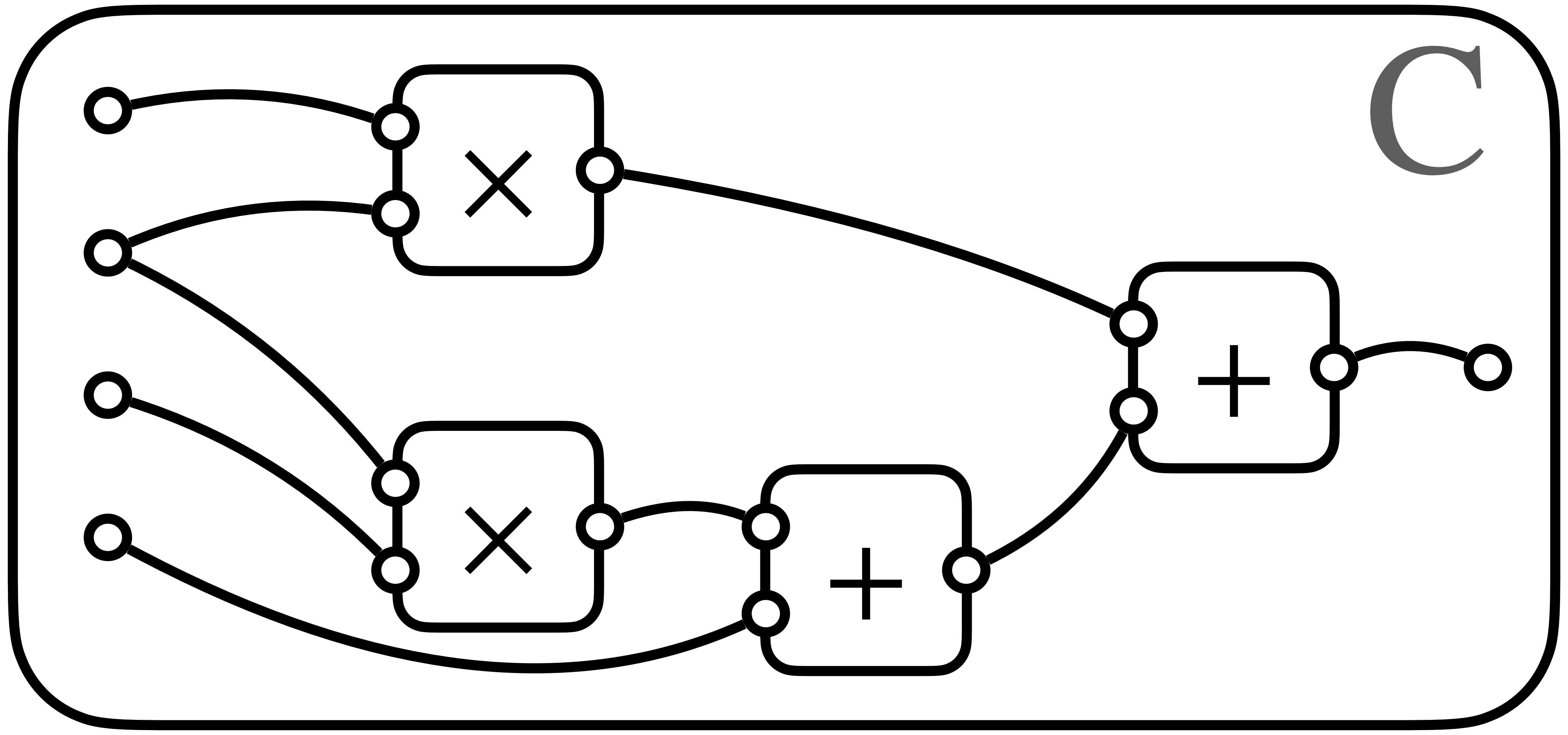


$O(1)$

[AIK11], [BMR16], **[BLLL23]**, [LL24], [Heath24]

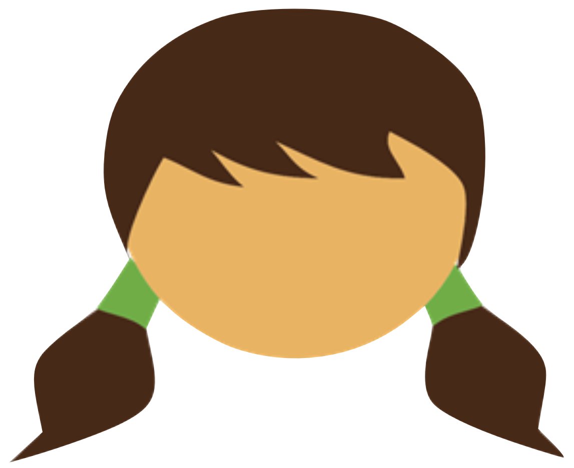
Garbled Arithmetic Circuits

Bounded Integer Computation (BIC)



$O(1)$

[AIK11], [BMR16], **[BLLL23]**, [LL24], [Heath24]



Garbler

$B \in \mathbb{Z}^+$

$\overrightarrow{x_0} \in \mathbb{Z}^*$

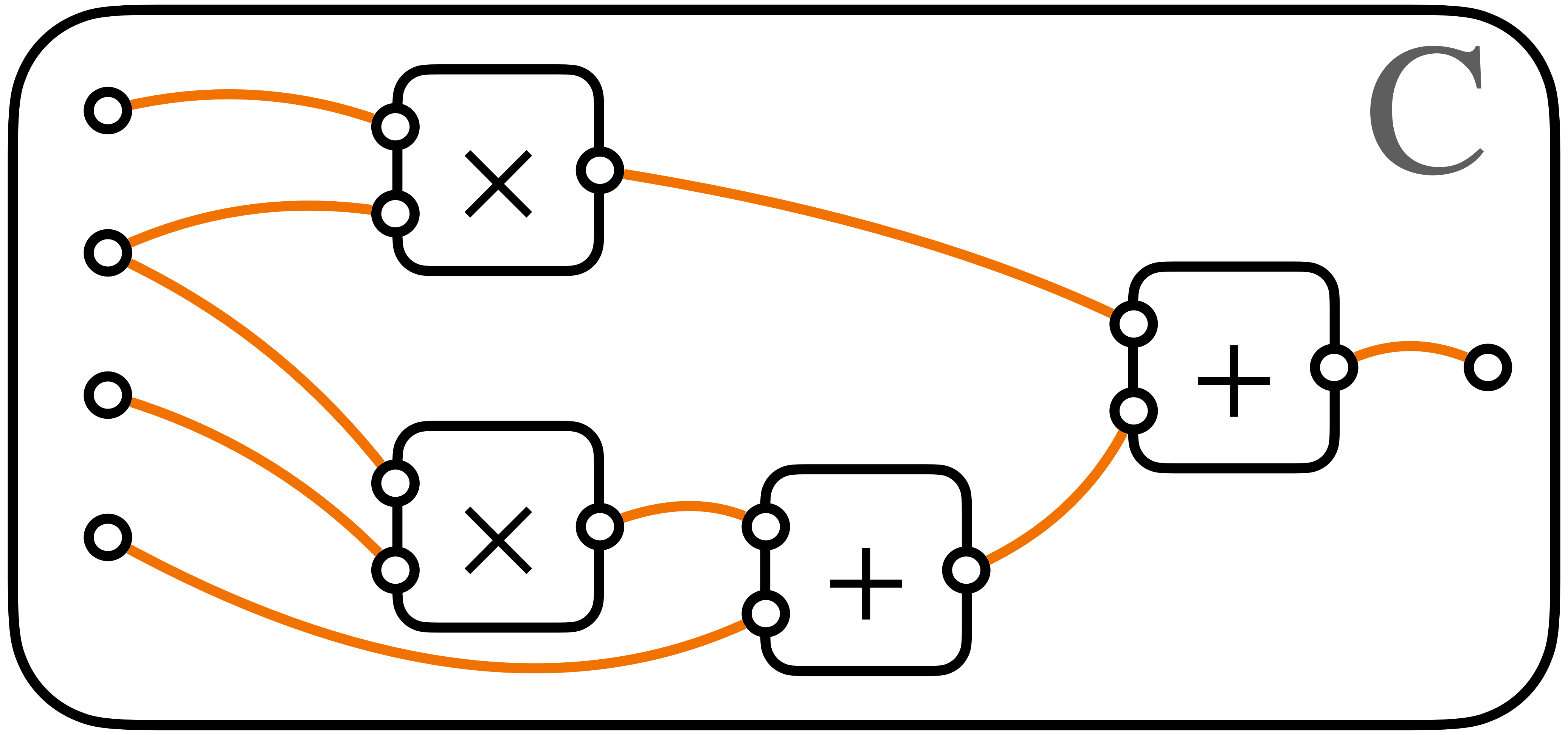
Evaluator

$B \in \mathbb{Z}^+$

$\overrightarrow{x_1} \in \mathbb{Z}^*$

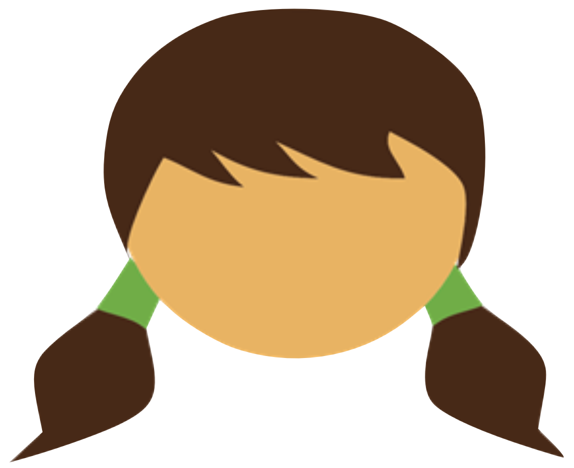
Garbled Arithmetic Circuits

Bounded Integer Computation (BIC)



$O(1)$

[AIK11], [BMR16], **[BLLL23]**, [LL24], [Heath24]



Garbler

$B \in \mathbb{Z}^+$

$\overrightarrow{x_0} \in \mathbb{Z}^*$

Evaluator

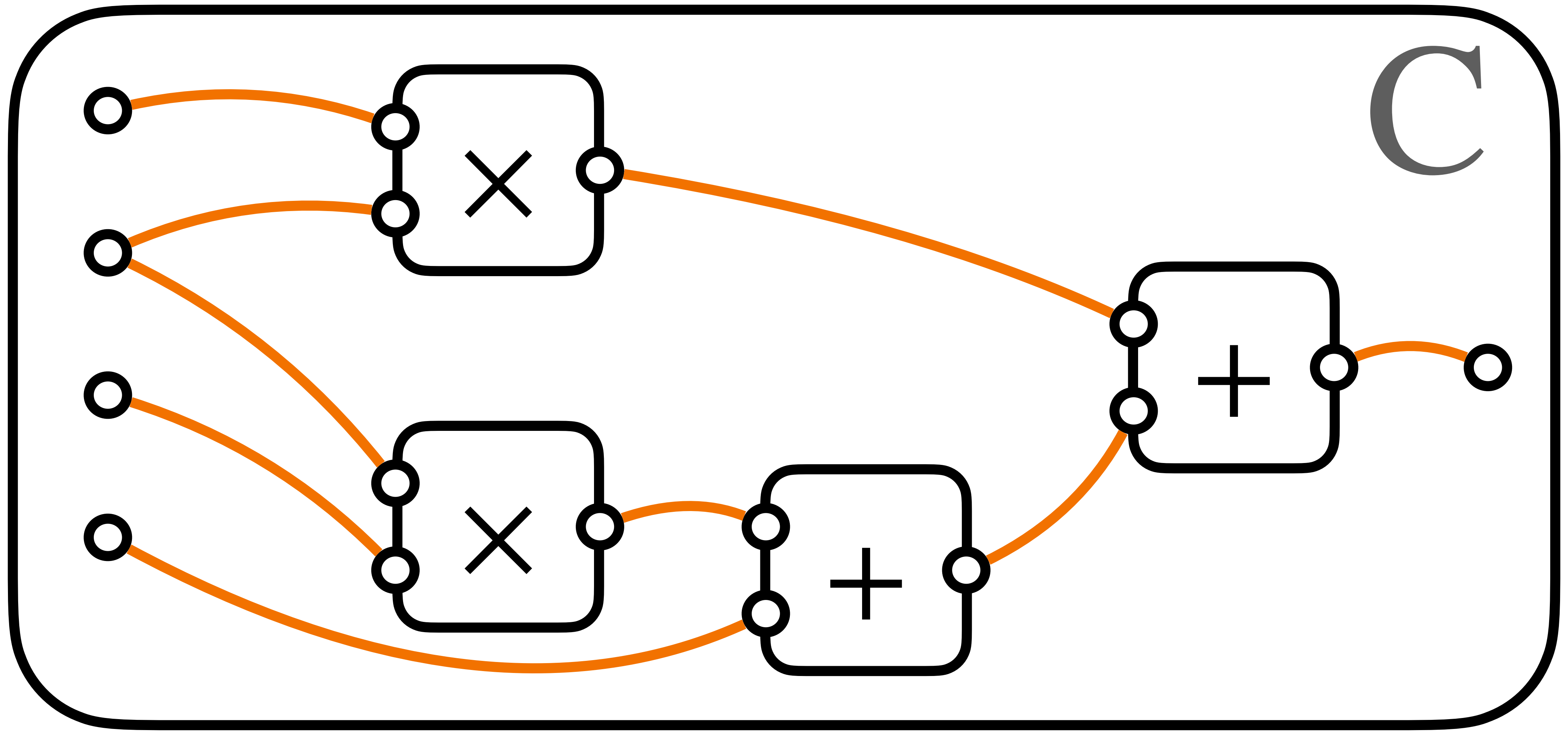
$B \in \mathbb{Z}^+$

$\overrightarrow{x_1} \in \mathbb{Z}^*$

Every wire is B-bounded.

Garbled Arithmetic Circuits

Bounded Integer Computation (BIC)



$O(1)$

[AIK11], [BMR16], **[BLLL23]**, [LL24], [Heath24]

Assume DCR, there exists a constant-rate arithmetic garbling in the bounded integer model

Toward **Malicious Constant-Rate** 2PC via Arithmetic Garbling

Toward **Malicious Constant-Rate** 2PC via Arithmetic Garbling

★ Our focus

semi-honest $\xrightarrow{\text{Rate-preserving}}$ malicious

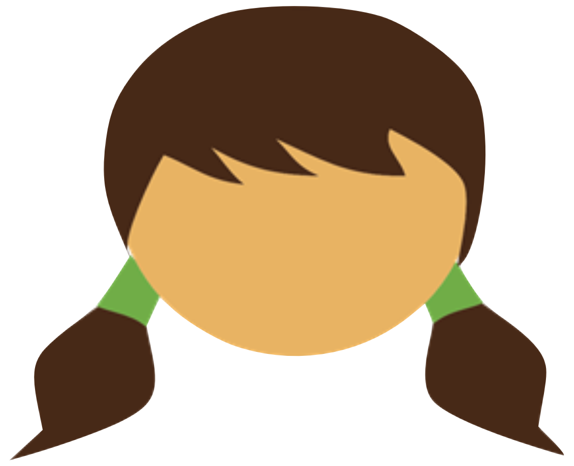
Semi-honest ⇒ Malicious

$C(\overrightarrow{x_0}, \overrightarrow{x_1})$

$\boxed{\hat{C}}$

$\boxed{\hat{C}}$ $\overrightarrow{X_0}$

$\overrightarrow{x_0}$



Garbler

$\overrightarrow{x_1}$

Evaluator

Semi-honest ⇒ Malicious

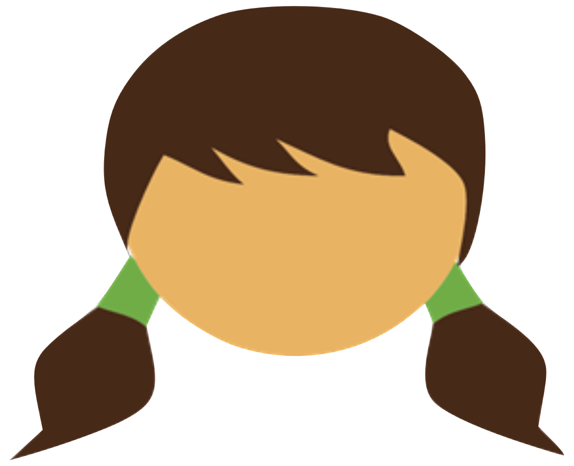
$$C(\overrightarrow{x_0}, \overrightarrow{x_1})$$

$\boxed{\hat{C}}$

$\boxed{\hat{C}}$ $\overrightarrow{X_0}$

Ensure this is correctly garbled

$\overrightarrow{x_0}$



Garbler

(& malicious secure OT/OLE)

$\overrightarrow{x_1}$

Evaluator

Semi-honest $\Rightarrow$ Malicious

$C(\overrightarrow{x_0}, \overrightarrow{x_1})$

$\boxed{\hat{C}}$ $\overrightarrow{X_0}$

Ensure this is correctly garbled

(& malicious secure OT/OLE)

For the Boolean garblings, this is possible:

Semi-honest ⇒ Malicious

$C(\overrightarrow{x_0}, \overrightarrow{x_1})$

$\boxed{\hat{C}}$ $\overrightarrow{X_0}$

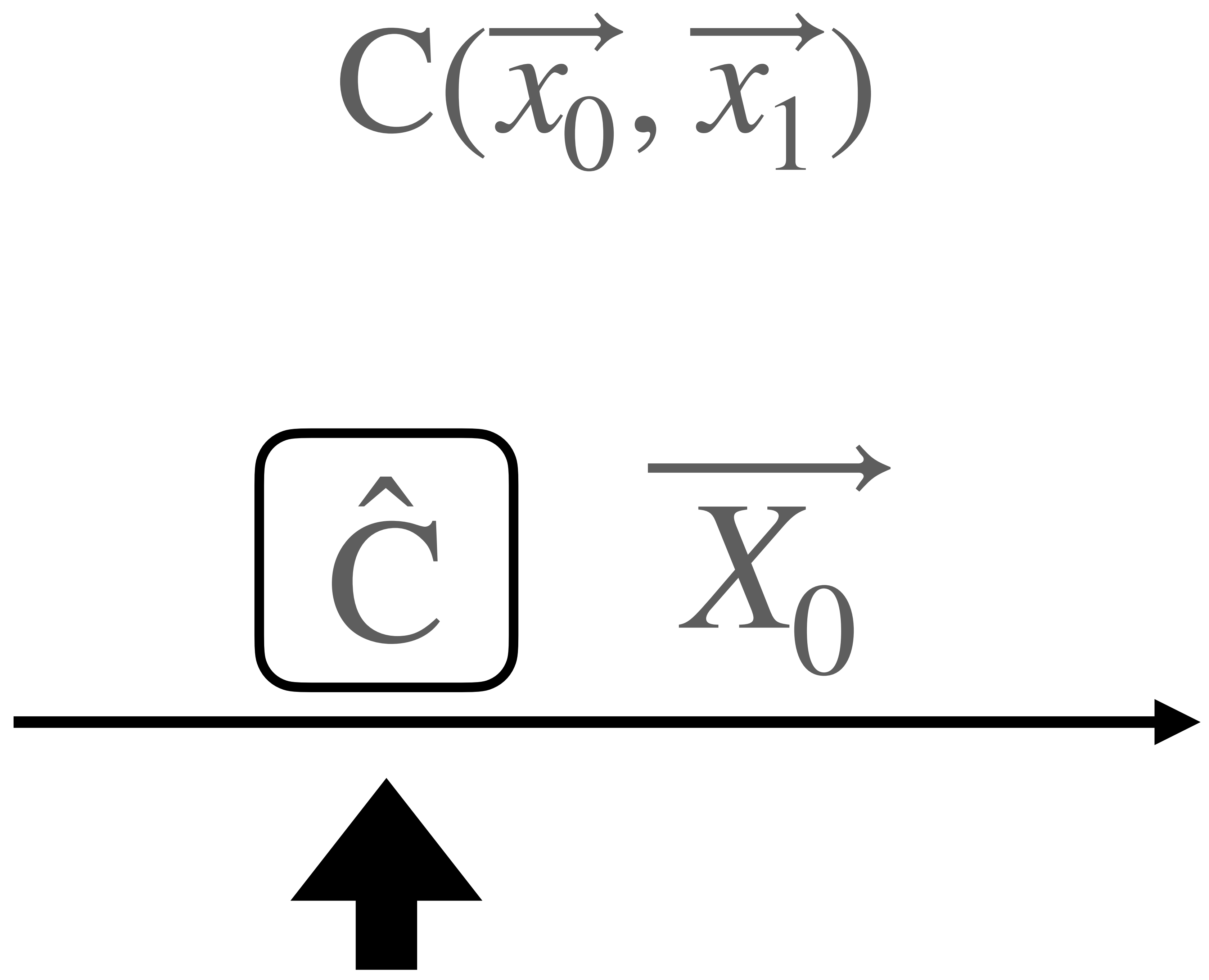
Ensure this is correctly garbled

(& malicious secure OT/OLE)

For the Boolean garblings, this is possible:

- ❌ **Cut-and-choose** $O(\lambda)$**-rate** [LP07, …]

Semi-honest ⇒ Malicious



Ensure this is correctly garbled

(& malicious secure OT/OLE)

For the Boolean garblings, this is possible:

- ❌ **Cut-and-choose** $O(\lambda)$**-rate** [LP07, …]
- **Authenticated garblings** $O(1)$**-rate** [IKO+11, WRK17, …]

Semi-honest ⇒ Malicious

$C(\vec{x}_0, \vec{x}_1)$

$\boxed{\hat{C}}$ $\vec{X}_0$

Ensure this is correctly garbled

(& malicious secure OT/OLE)

For the Boolean garblings, this is possible:

- ❌ **Cut-and-choose** $O(\lambda)$**-rate**
  [LP07, …]
- **Authenticated garblings** $O(1)$**-rate**
  [IKO+11, WRK17, …]
- **Dual execution** $O(1)$**-rate**
  [MF06, …] ⚠️ **1-bit leakage**

Semi-honest ⇒ Malicious

$C(\overrightarrow{x_0}, \overrightarrow{x_1})$

$\boxed{\hat{C}}$ $\overrightarrow{X_0}$

Ensure this is correctly garbled

(& malicious secure OT/OLE)

For the Boolean garblings, this is possible:

❌ **Cut-and-choose** $O(\lambda)$**-rate**
[LP07, …]

❌ **Authenticated garblings** $O(1)$**-rate**
[IKO+11, WRK17, …]

❌ **Dual execution** $O(1)$**-rate**
[MF06, …] ⚠️ **1-bit leakage**

Our Results

Our Results

- Constant-rate BLLL GC is *not* well-defined w.r.t. the malicious security.

Our Results

- Constant-rate BLLL GC is *not* well-defined w.r.t. the malicious security.

- There exists an “overflow attack” that even works for a fully correct GC.

Our Results

- Constant-rate BLLL GC is *not* well-defined w.r.t. the malicious security.

- There exists an “overflow attack” that even works for a fully correct GC.

- The best achievable security is against:
    - malicious G with 1-bit leakage
    - semi-honest E

Our Results

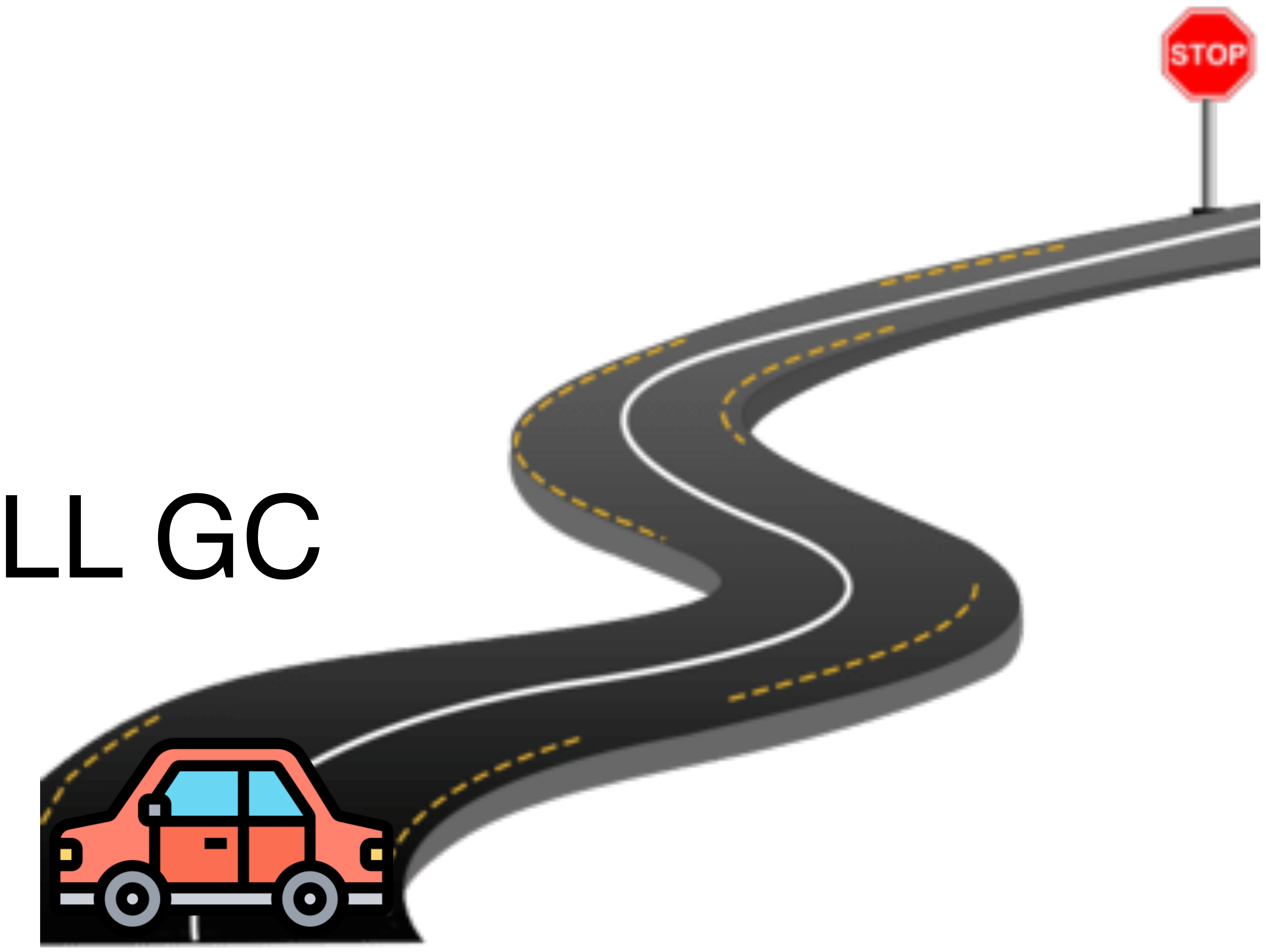
- Constant-rate BLLL GC is *not* well-defined w.r.t. the malicious security.
- There exists an “overflow attack” that even works for a fully correct GC.
- The best achievable security is against:
    - malicious G with 1-bit leakage
    - semi-honest E
- We can achieve the best achievable security *efficiently*.

Roadmap

- Constant-rate BLLL GC
- The overflow attack
- Our protocol based-on BLLL GC

Roadmap

- **Constant-rate BLLL GC**
- The overflow attack
- Our protocol based-on BLLL GC



Review of Constant-Rate BLLL GC

Review of Constant-Rate BLLL GC

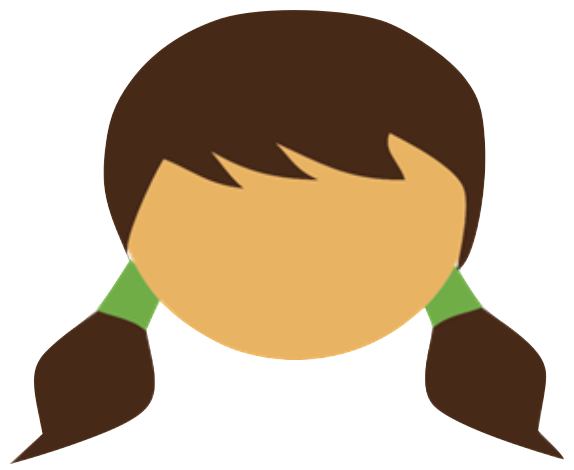
AIK GC paradigm over a ring

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

GC labels

w


Garbler

$\overrightarrow{k_0^w}, \overrightarrow{k_1^w} \in \mathscr{R}^n$


Evaluator

$w\,\overrightarrow{k_0^w} + \overrightarrow{k_1^w} \in \mathscr{R}^n$

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

GC labels

Communication-free information-theoretic ADD/MUL

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

GC labels

Communication-free information-theoretic ADD/MUL

Boolean

G

AIK

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

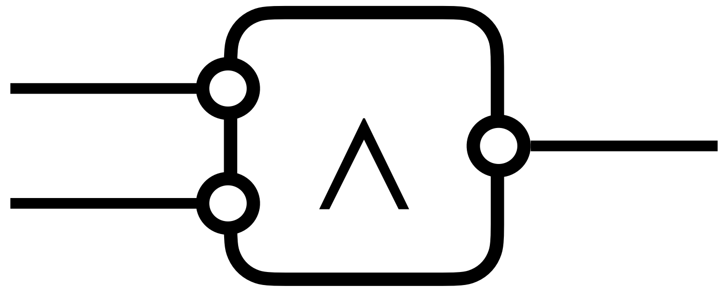
GC labels

Communication-free information-theoretic ADD/MUL

Boolean

G

AIK

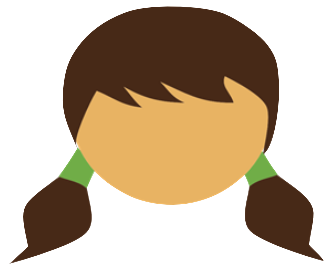


Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

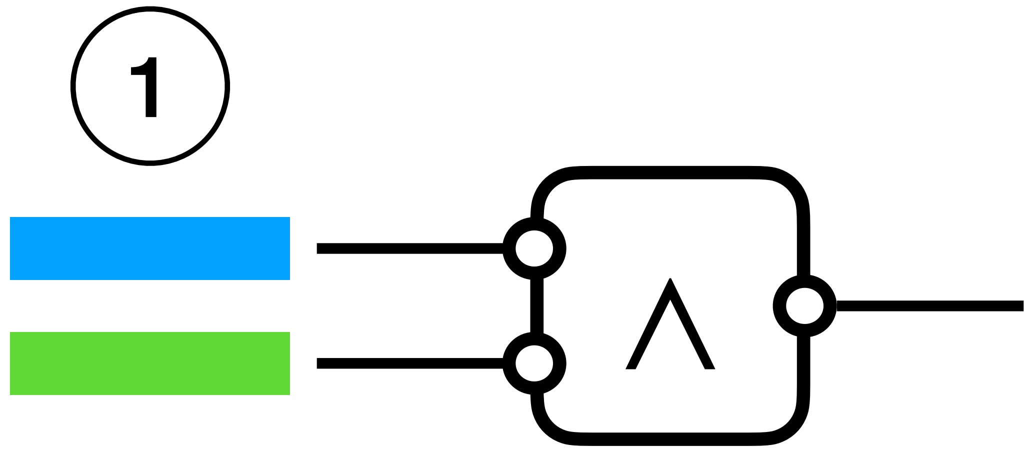
GC labels

Communication-free information-theoretic ADD/MUL



G

Boolean



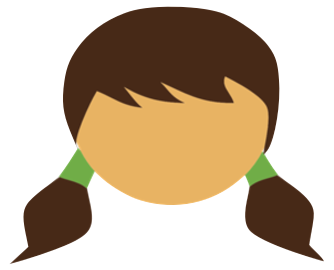
AIK

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

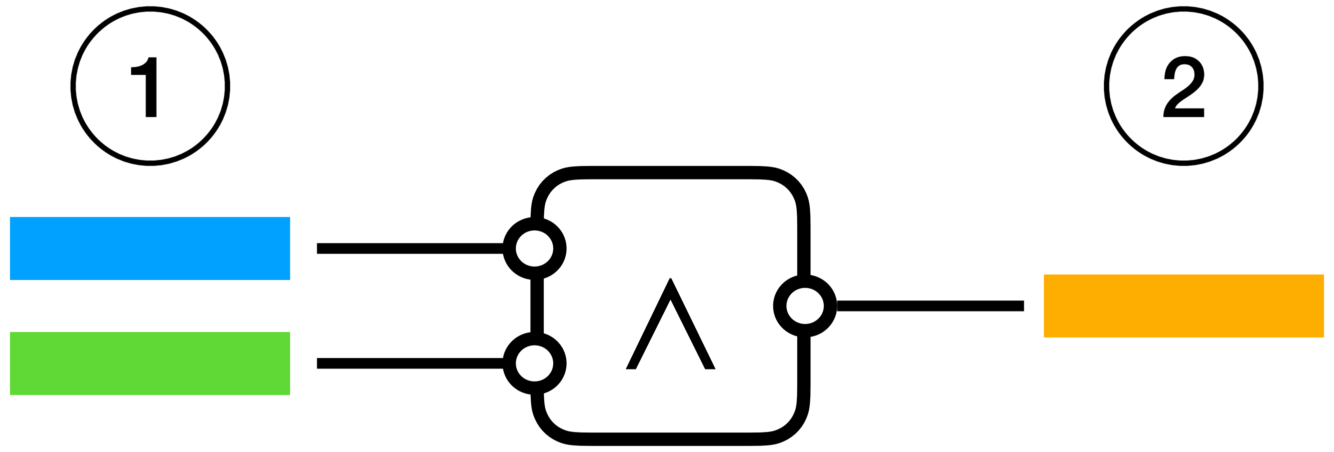
GC labels

Communication-free information-theoretic ADD/MUL



G

Boolean



garbled truth table

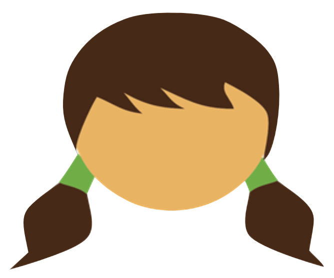
AIK

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

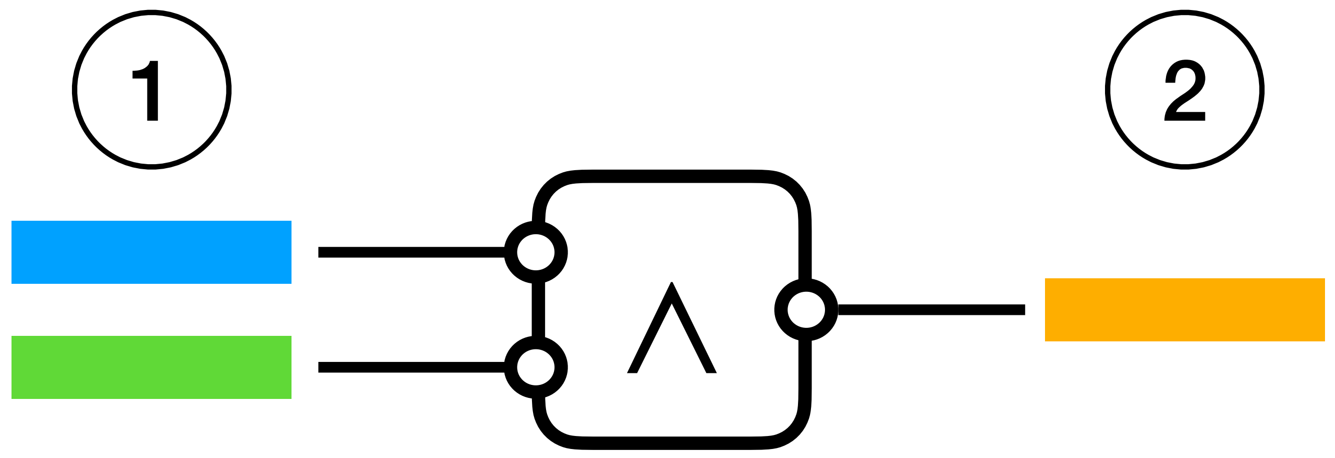
GC labels

Communication-free information-theoretic ADD/MUL

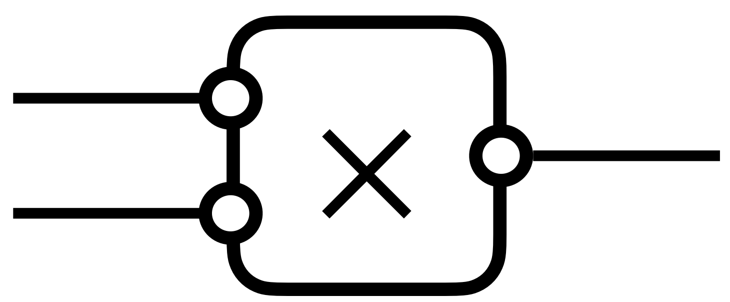


G

Boolean



AIK



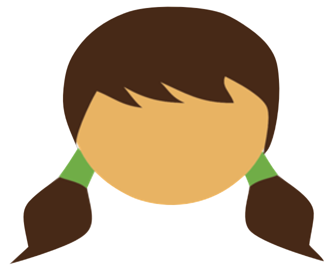
garbled truth table

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

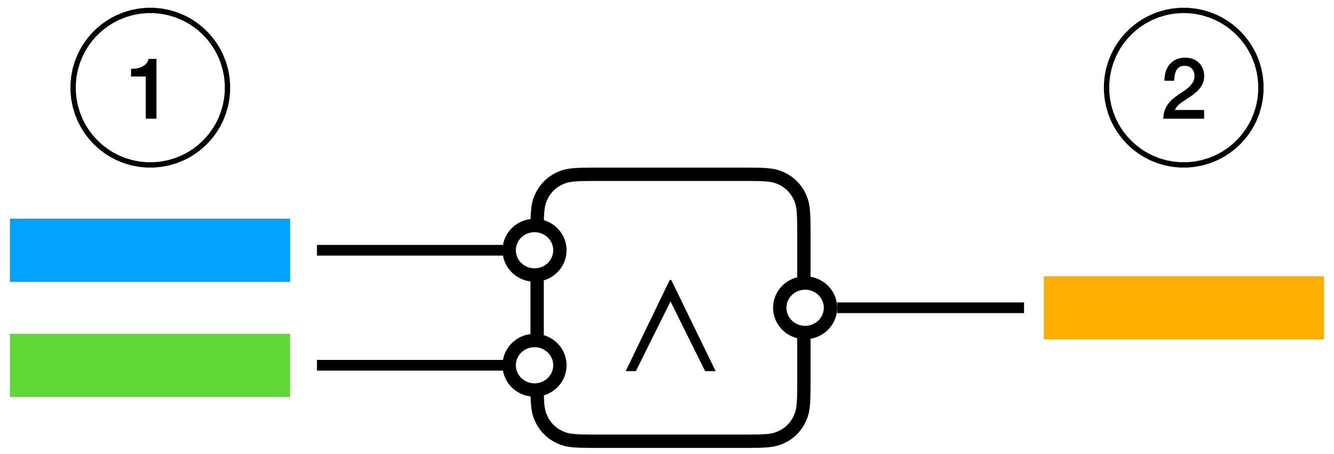
GC labels

Communication-free information-theoretic ADD/MUL



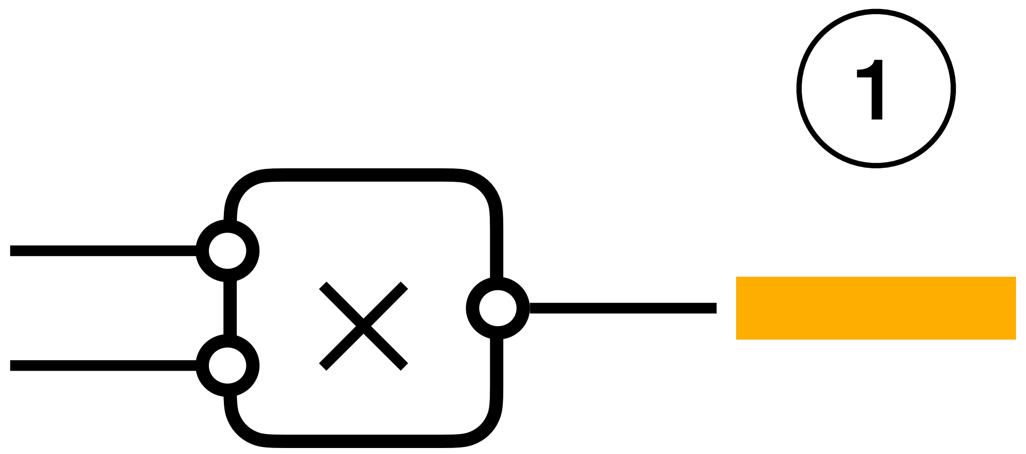
G

Boolean



garbled truth table

AIK

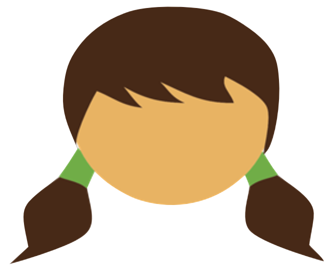


Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

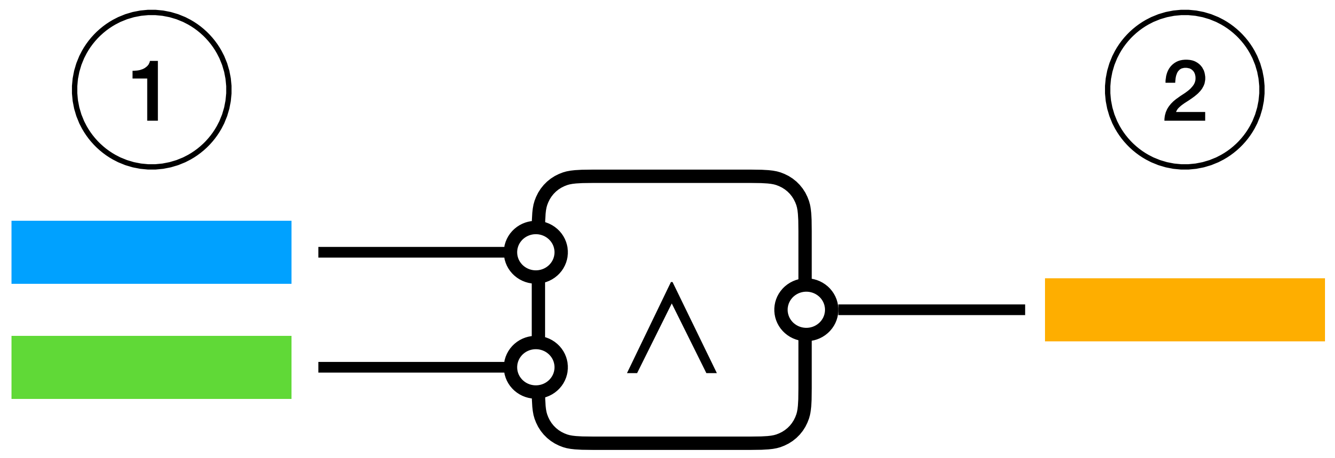
GC labels

Communication-free information-theoretic ADD/MUL



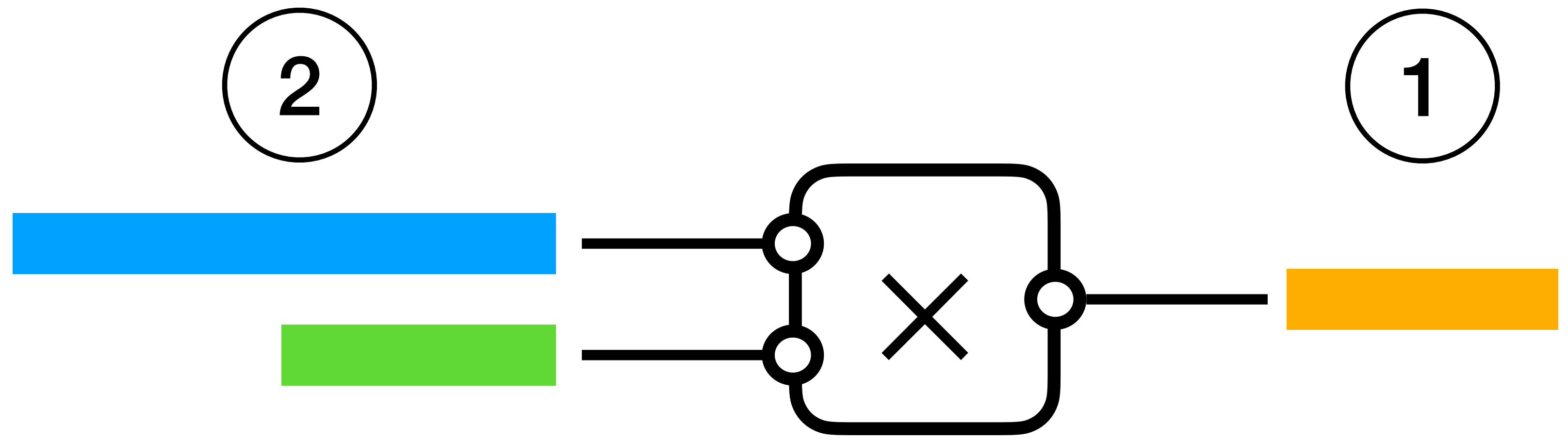
G

Boolean



garbled truth table

AIK



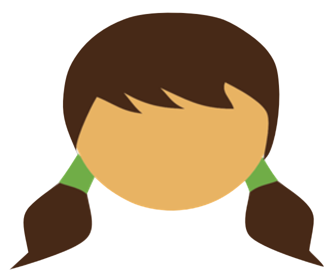
no communication

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

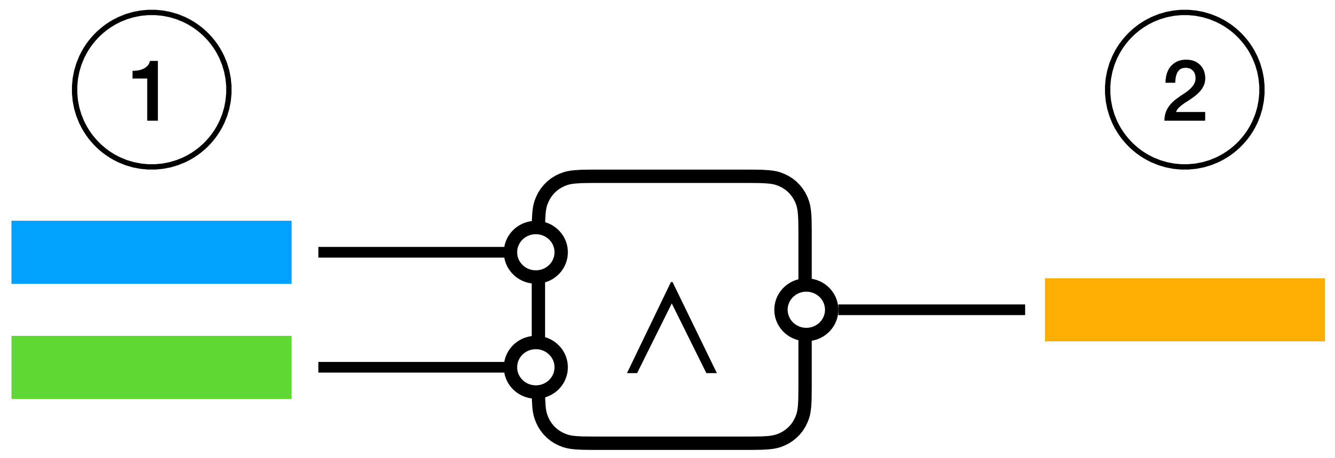
GC labels

Communication-free information-theoretic ADD/MUL



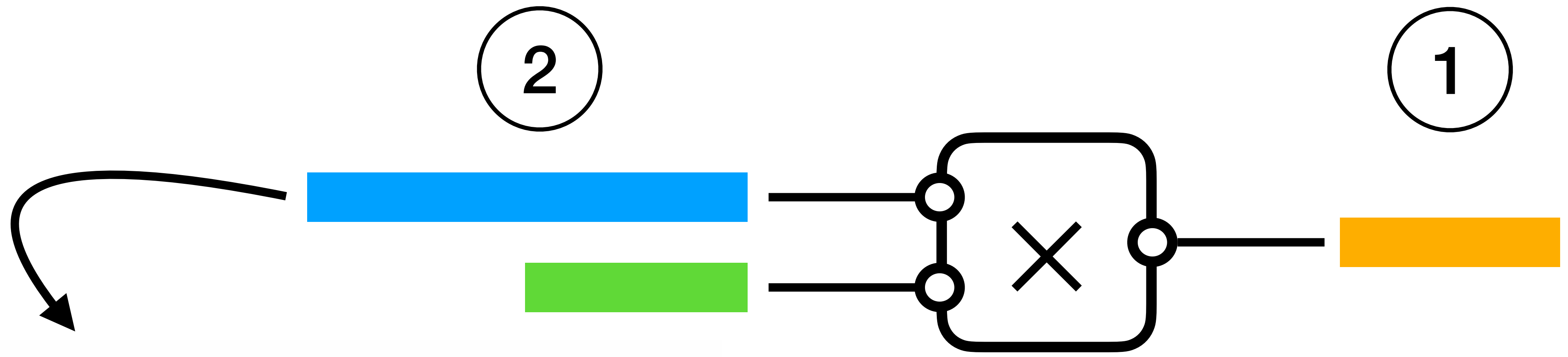
G

Boolean



garbled truth table

AIK



grows exponentially

no communication

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

GC labels

Communication-free information-theoretic ADD/MUL

KE gadgets

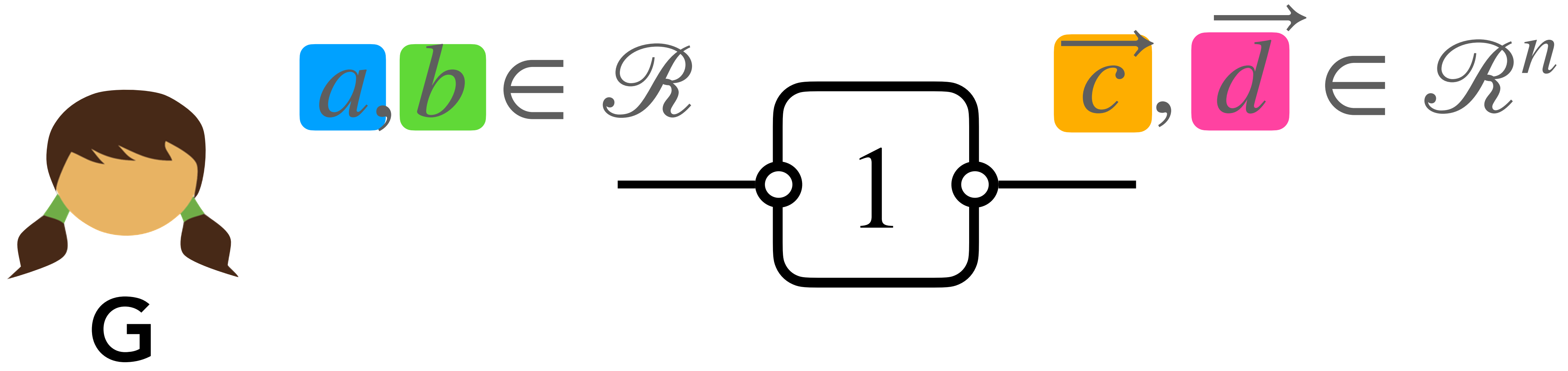
Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

GC labels

Communication-free information-theoretic ADD/MUL

KE gadgets



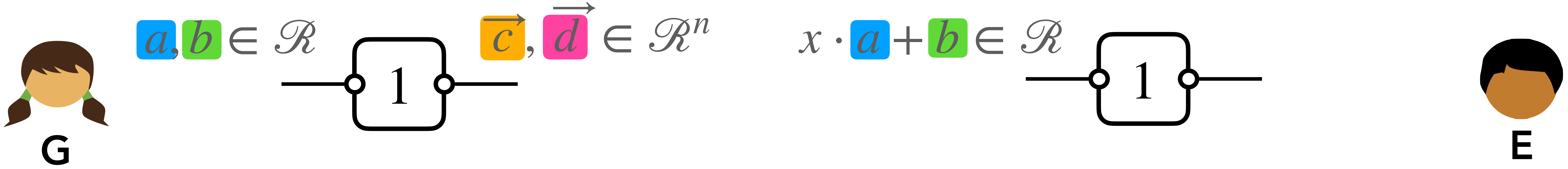
Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

GC labels

Communication-free information-theoretic ADD/MUL

KE gadgets



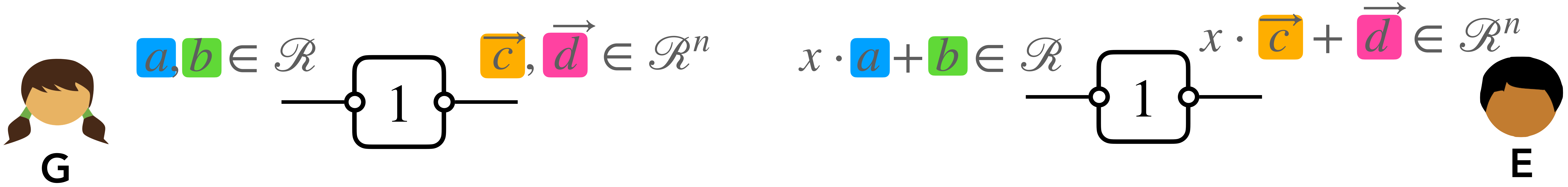
Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

GC labels

Communication-free information-theoretic ADD/MUL

KE gadgets



Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

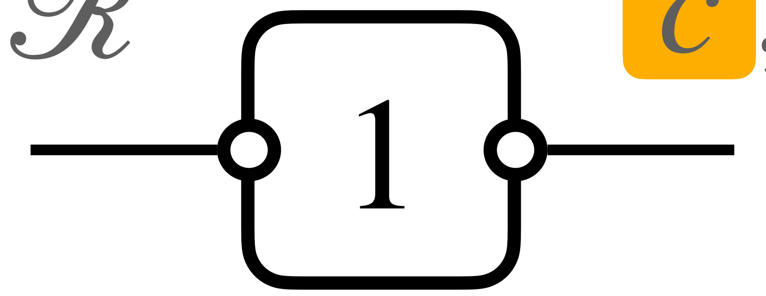
GC labels

Communication-free information-theoretic ADD/MUL

KE gadgets

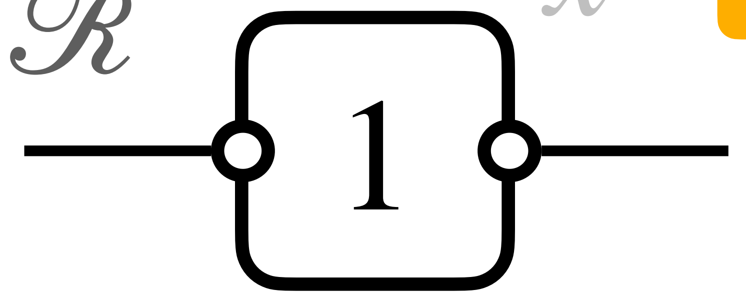
linear key/msg homomorphism

$a, b \in \mathscr{R}$

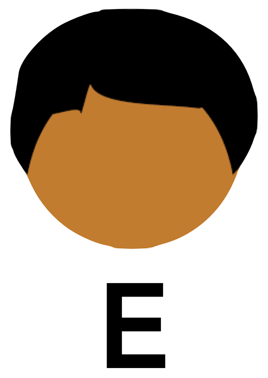


$\vec{c}, \vec{d} \in \mathscr{R}^n$

$x \cdot a + b \in \mathscr{R}$



$x \cdot \vec{c} + \vec{d} \in \mathscr{R}^n$



$\mathtt{Enc}_{b}(\vec{d})$ $\mathtt{Enc}_{a}(\vec{c})$

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

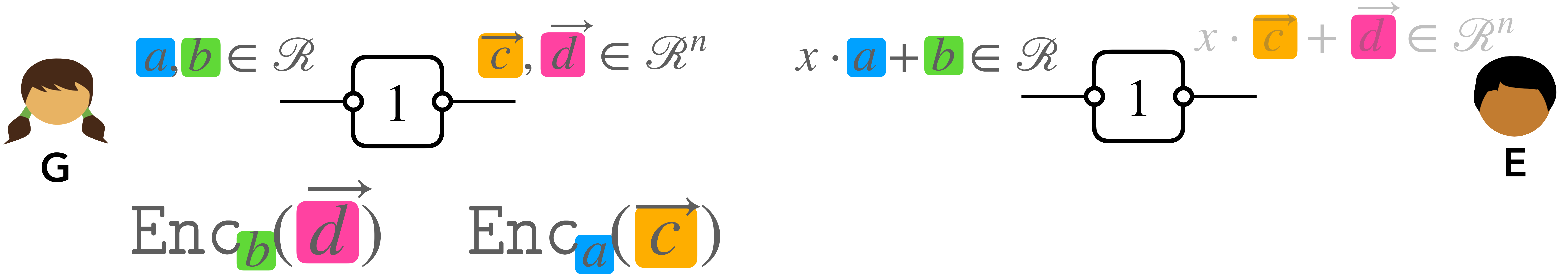
GC labels

Communication-free information-theoretic ADD/MUL

KE gadgets

linear key/msg homomorphism

x **is known by E**



Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

GC labels

Communication-free information-theoretic ADD/MUL

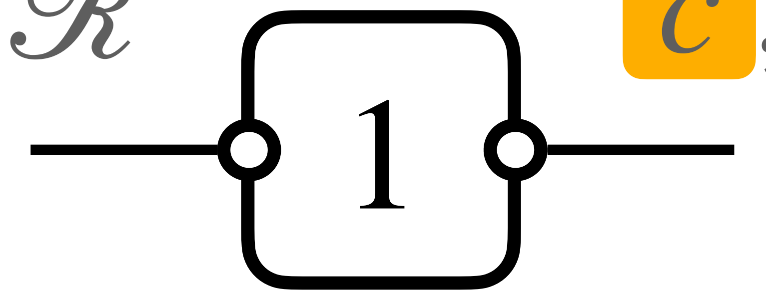
KE gadgets

linear key/msg homomorphism

x **is known by E**

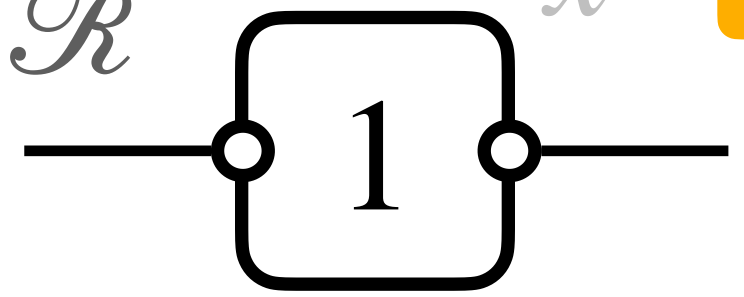
G

$a, b \in \mathscr{R}$



$\vec{c}, \vec{d} \in \mathscr{R}^n$

$x \cdot a + b \in \mathscr{R}$



$x \cdot \vec{c} + \vec{d} \in \mathscr{R}^n$

E

$\mathtt{Enc}_{b}(\vec{d})$ $\quad$ $\mathtt{Enc}_{a}(\vec{c})$ $\quad$ $\mathtt{Enc}_{x \cdot a + b}(x \cdot \vec{c} + \vec{d})$

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

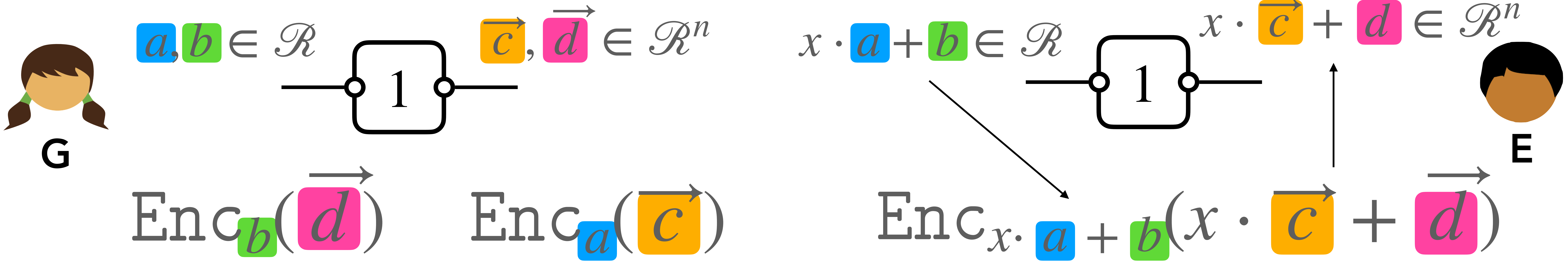
GC labels

Communication-free information-theoretic ADD/MUL

KE gadgets

linear key/msg homomorphism

x **is known by E**



Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

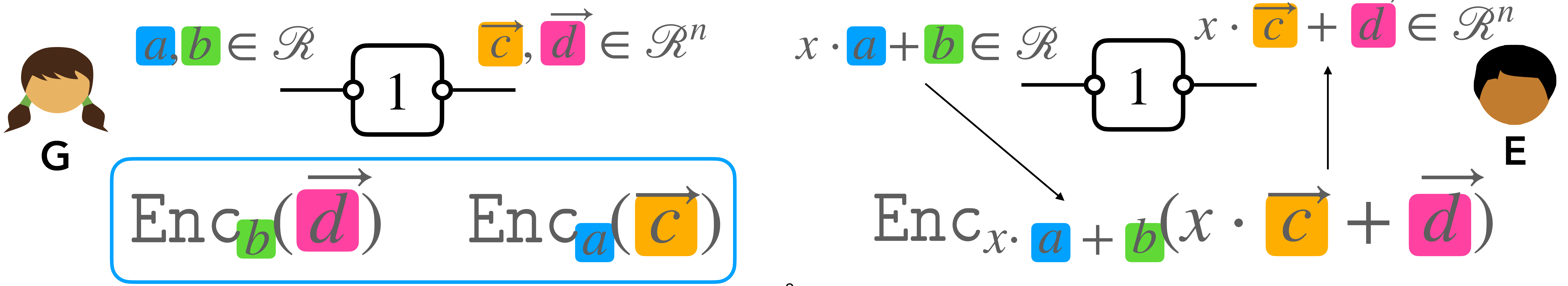
GC labels

Communication-free information-theoretic ADD/MUL

KE gadgets

linear key/msg homomorphism

x is known by E



Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

GC labels

Communication-free information-theoretic ADD/MUL

KE gadgets

BLLL constant-rate KE gadgets from DCR

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

GC labels

Communication-free information-theoretic ADD/MUL

KE gadgets

BLLL constant-rate KE gadgets from DCR

$\mathtt{Enc}_a(\overrightarrow{c})$

$$g_1^a(N+1)^{2c_1}, \cdots, g_n^a(N+1)^{2c_n} \mod N^{\zeta+1}$$

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

GC labels

Communication-free information-theoretic ADD/MUL

KE gadgets

BLLL constant-rate KE gadgets from DCR

Why BIC ?

$\mathtt{Enc}_a(\overrightarrow{c})$

$$g_1^a(N+1)^{2c_1}, \cdots, g_n^a(N+1)^{2c_n} \mod N^{\zeta+1}$$

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

GC labels

Communication-free information-theoretic ADD/MUL

KE gadgets

BLLL constant-rate KE gadgets from DCR

Why BIC ?

$\mathtt{Enc}_a(\overrightarrow{c})$

$$g_1^a(N+1)^{2c_1}, \cdots, g_n^a(N+1)^{2c_n} \mod N^{\zeta+1}$$

Group of order N^ζ

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

GC labels

Communication-free information-theoretic ADD/MUL

KE gadgets

BLLL constant-rate KE gadgets from DCR

$\mathtt{Enc}_a(\overrightarrow{c})$

Why BIC ?

$$g_1^a(N+1)^{2c_1}, \cdots, g_n^a(N+1)^{2c_n} \mod N^{\zeta+1}$$

Group of order N^ζ → **Use as key of next $\mathtt{Enc}_{\cdot}(\cdot)$**

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

GC labels

Communication-free information-theoretic ADD/MUL

KE gadgets

BLLL constant-rate KE gadgets from DCR

Why BIC ?

$\mathtt{Enc}_a(\overrightarrow{c})$

$$g_1^a(N+1)^{2c_1}, \cdots, g_n^a(N+1)^{2c_n} \mod N^{\zeta+1}$$

Group of order N^ζ → **Group of unknown order** **Use as key of next $\mathtt{Enc}.(\cdot)$**

Review of Constant-Rate BLLL GC

AIK GC paradigm over a ring

GC labels

Communication-free information-theoretic ADD/MUL

KE gadgets

BLLL constant-rate KE gadgets from DCR

Why BIC ?

$\mathtt{Enc}_a(\overrightarrow{c})$

$$g_1^a(N+1)^{2c_1}, \cdots, g_n^a(N+1)^{2c_n} \mod N^{\zeta+1}$$

The value can only be $[-B, B]$

Group of order N^ζ → **Use as key of next** $\mathtt{Enc}.(\cdot)$

Group of unknown order

$x \in \mathbb{Z}_{N^\zeta} \Rightarrow x \in \mathbb{Z}$

Roadmap

- Constant-rate BLLL GC
- **The overflow attack**
- Our protocol based-on BLLL GC

The Overflow Attack

Which part looks attackable?

AIK GC paradigm over a ring

GC labels

Communication-free information-theoretic ADD/MUL

KE gadgets

BLLL constant-rate KE gadgets from DCR

On BIC

The Overflow Attack

Maliciously change inputs

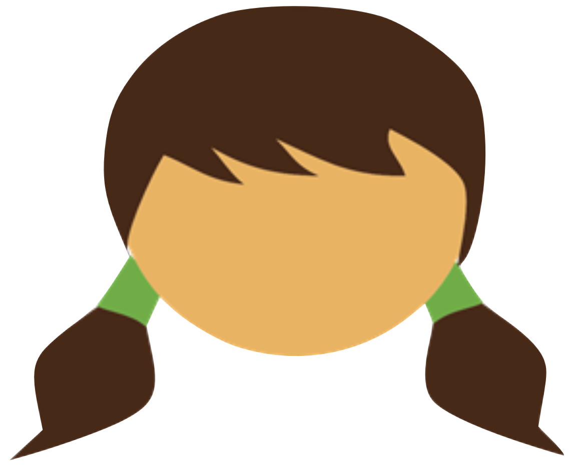
The Overflow Attack

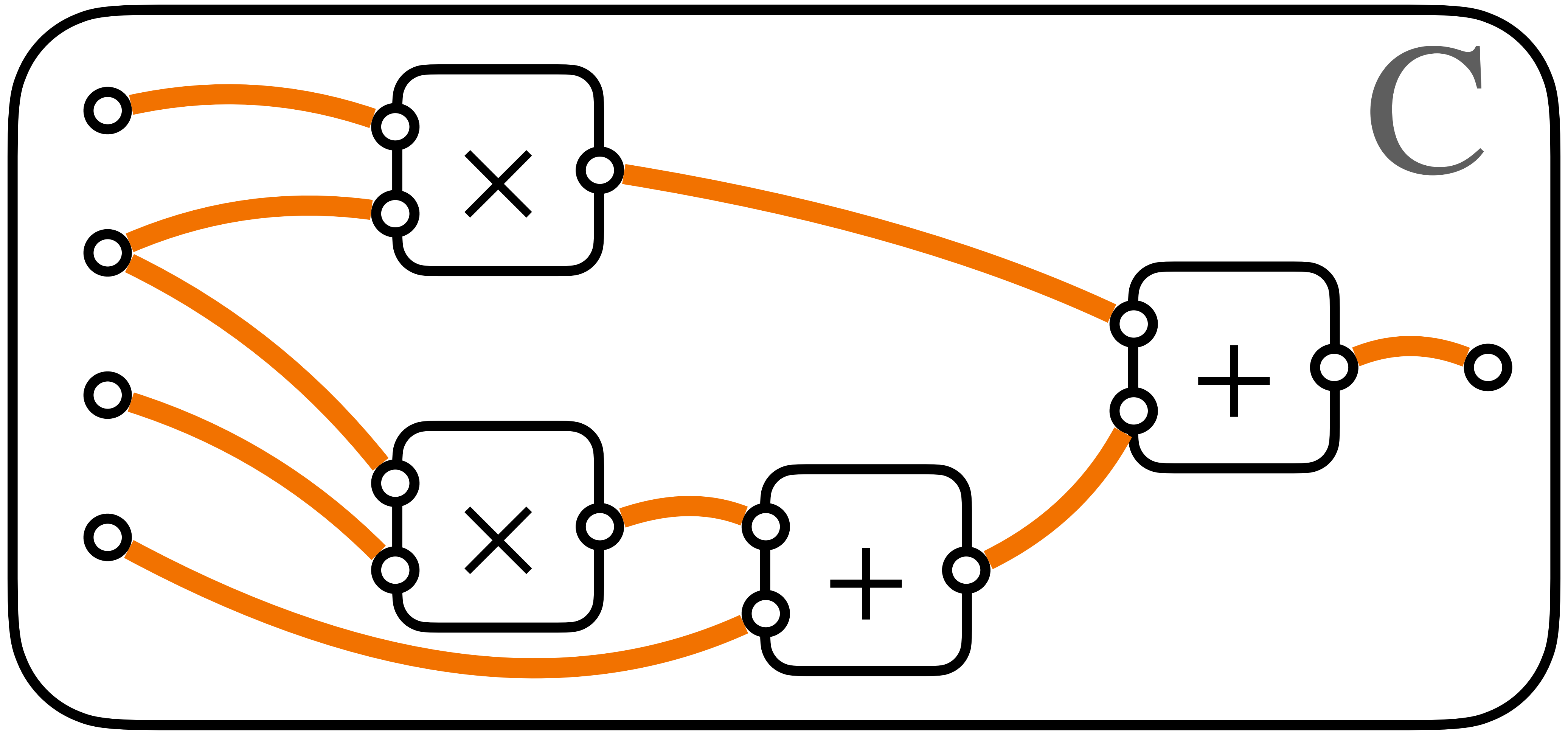
Maliciously change inputs

$\boxed{\text{Bounded-integer computation:}}$

$B \in \mathbb{Z}^+$

$\overrightarrow{x_0} \in \mathbb{Z}^*$





$B \in \mathbb{Z}^+$

$\overrightarrow{x_1} \in \mathbb{Z}^*$

Garbler

Every wire is B-bounded.

Evaluator

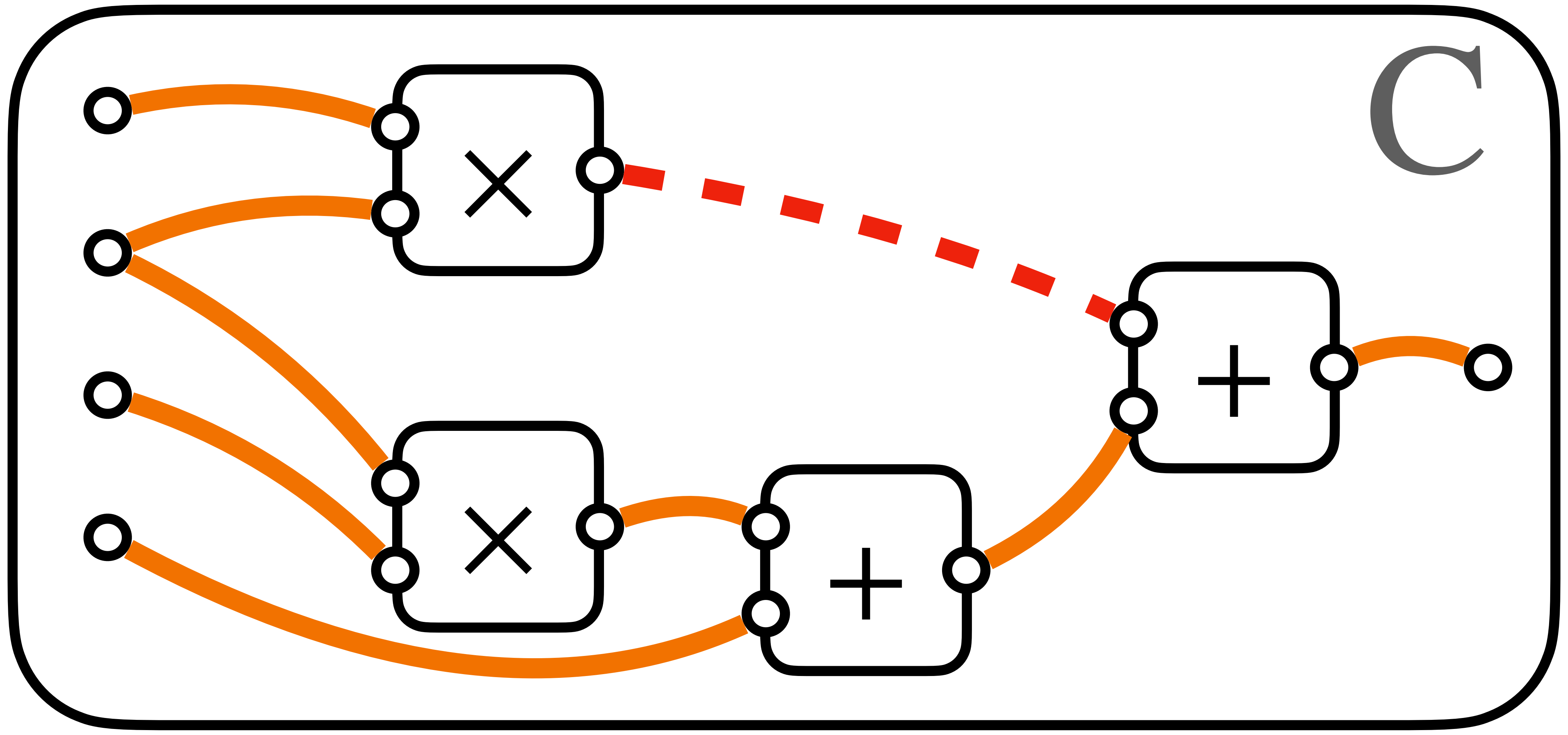
The Overflow Attack

Maliciously change inputs

Bounded-integer computation:

$B \in \mathbb{Z}^+$

$\overrightarrow{x_0'} \in \mathbb{Z}^*$



$B \in \mathbb{Z}^+$

$\overrightarrow{x_1} \in \mathbb{Z}^*$

Garbler

Some wire value may **not** be B-bounded.

Evaluator

The Overflow Attack

E could no longer evaluate

The Overflow Attack

E could no longer evaluate

Evaluator

$$x \in [-B, B]$$

$$x \in \mathbb{Z}_{N\zeta} \quad \rightarrow \quad x \in \mathbb{Z}$$

The Overflow Attack

E could no longer evaluate

Evaluator

$$x \notin [-B, B]$$

$$x \in \mathbb{Z}_{N\zeta} \quad \rightarrow \quad \tilde{x} \in \mathbb{Z}$$

The Overflow Attack

E could no longer evaluate

Evaluator

$$x \notin [-B, B]$$

$$x \in \mathbb{Z}_{N^{\zeta}} \rightarrow \tilde{x} \in \mathbb{Z}$$

E could no longer decrypt the ciphertext

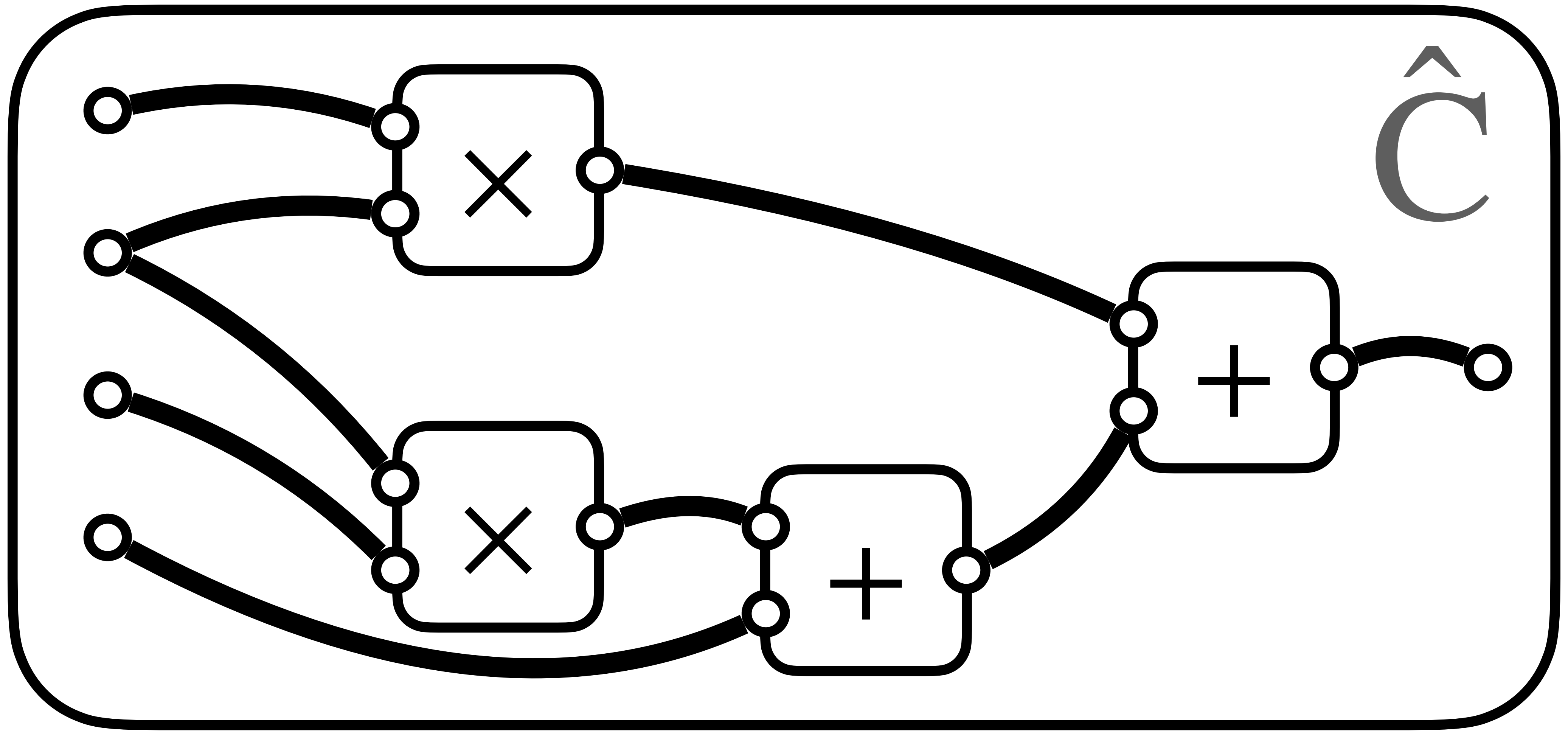
The Overflow Attack

A fully correct GC

$B \in \mathbb{Z}^+$

$\overrightarrow{x_0'} \in \mathbb{Z}^*$

Garbler



$B \in \mathbb{Z}^+$

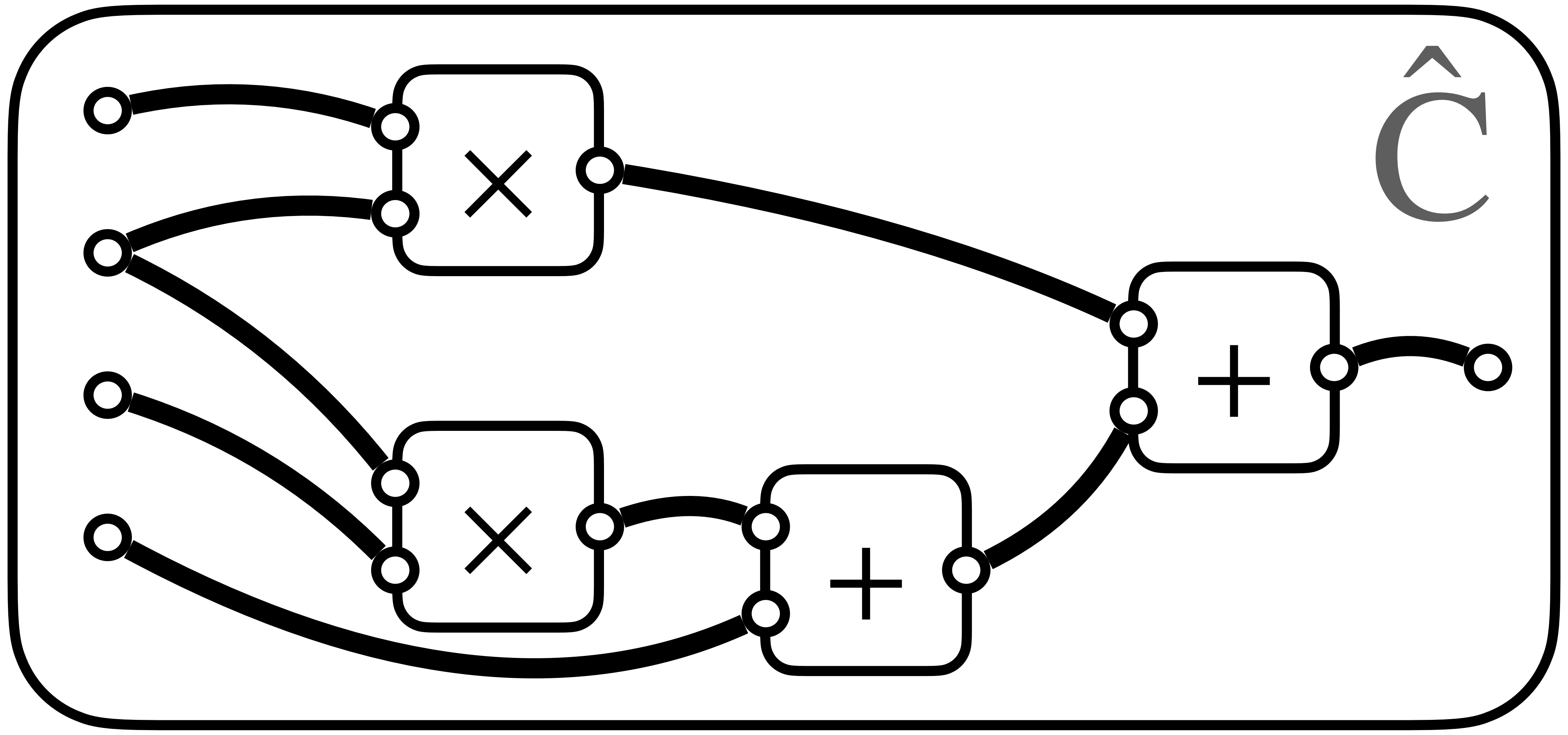
$\overrightarrow{x_1} \in \mathbb{Z}^*$

Evaluator

The Overflow Attack

$\mathtt{Dec} \to \perp$

A fully correct GC



$B \in \mathbb{Z}^+$

$\overrightarrow{x_0'} \in \mathbb{Z}^*$

Garbler

$B \in \mathbb{Z}^+$

$\overrightarrow{x_1} \in \mathbb{Z}^*$

Evaluator

The Overflow Attack

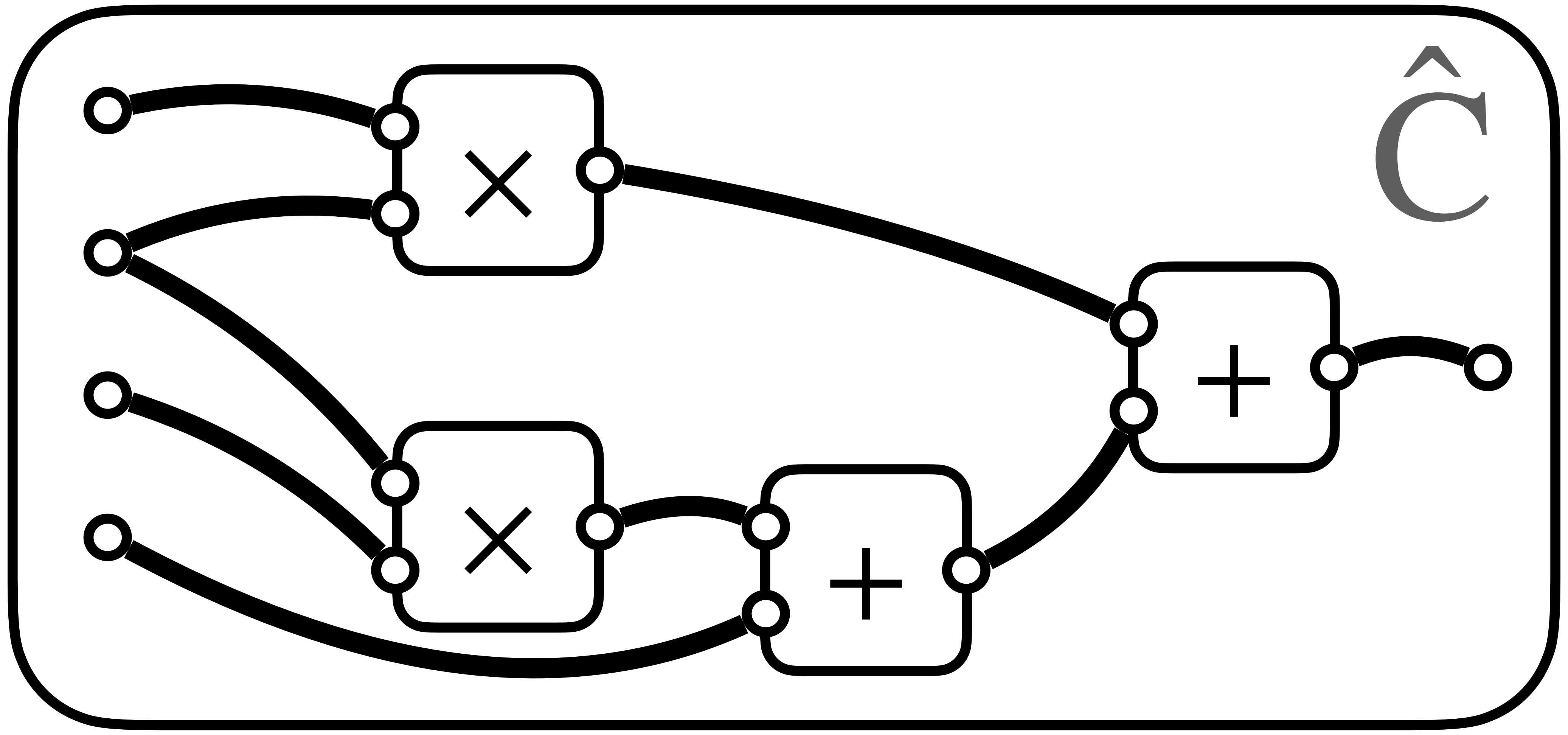
$\mathtt{Dec} \rightarrow \perp$

A fully correct GC

$B \in \mathbb{Z}^+$

$\overrightarrow{x_0'} \in \mathbb{Z}^*$

Garbler



selective attack

$B \in \mathbb{Z}^+$

$\overrightarrow{x_1} \in \mathbb{Z}^*$

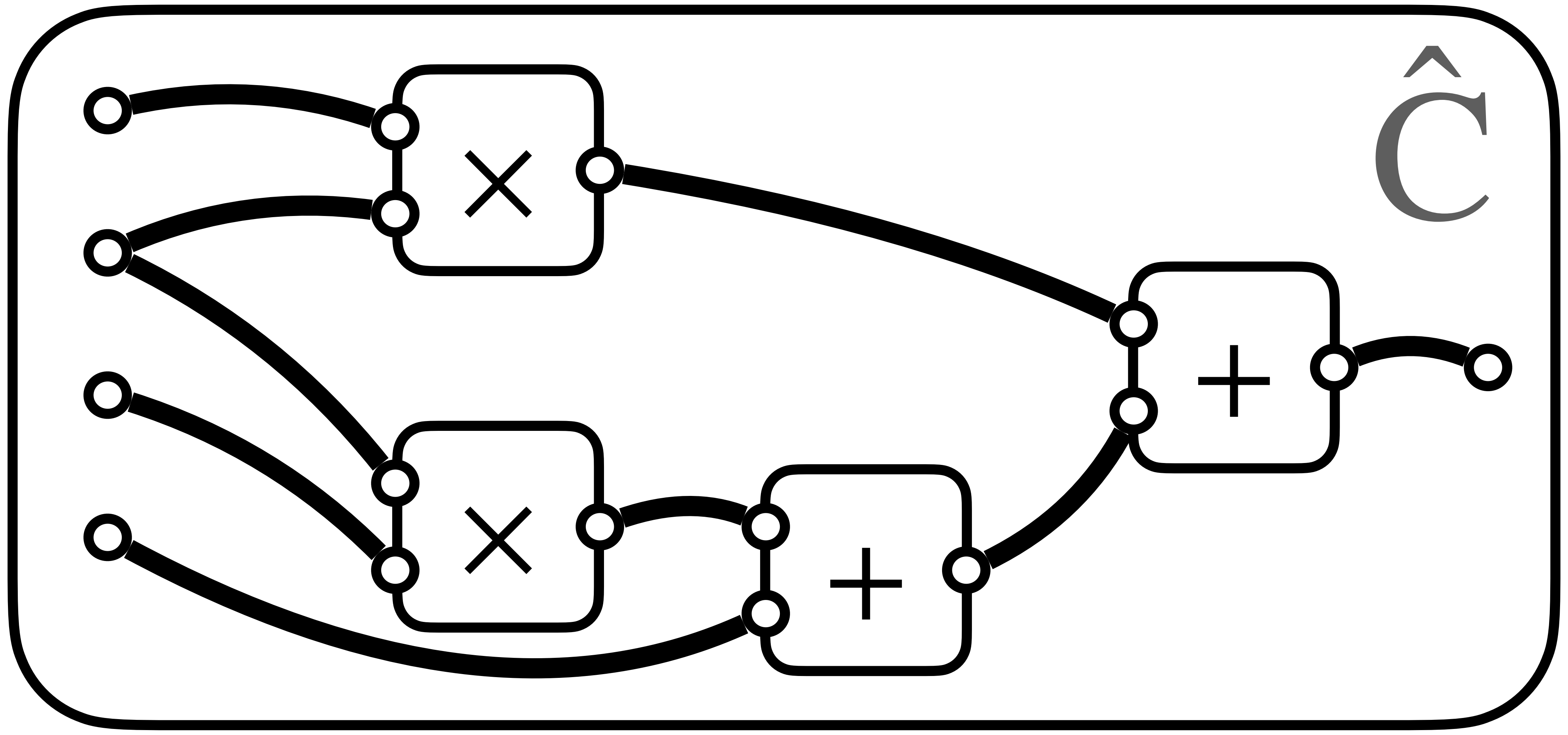
Evaluator

The Overflow Attack

A fully correct GC

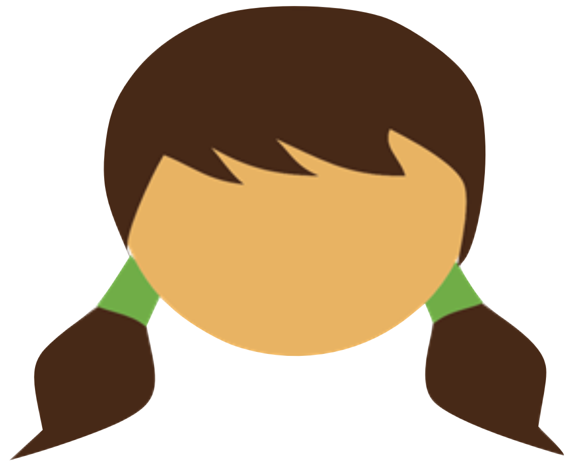
$B \in \mathbb{Z}^+$

$\overrightarrow{x_0} \in \mathbb{Z}^*$



$B \in \mathbb{Z}^+$

$\overrightarrow{x'_1} \in \mathbb{Z}^*$



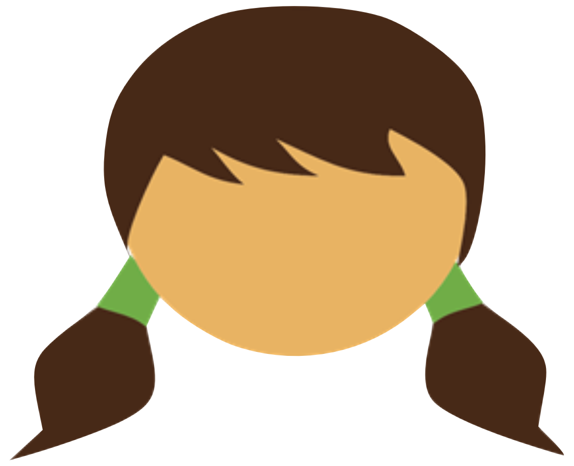
Garbler

Evaluator

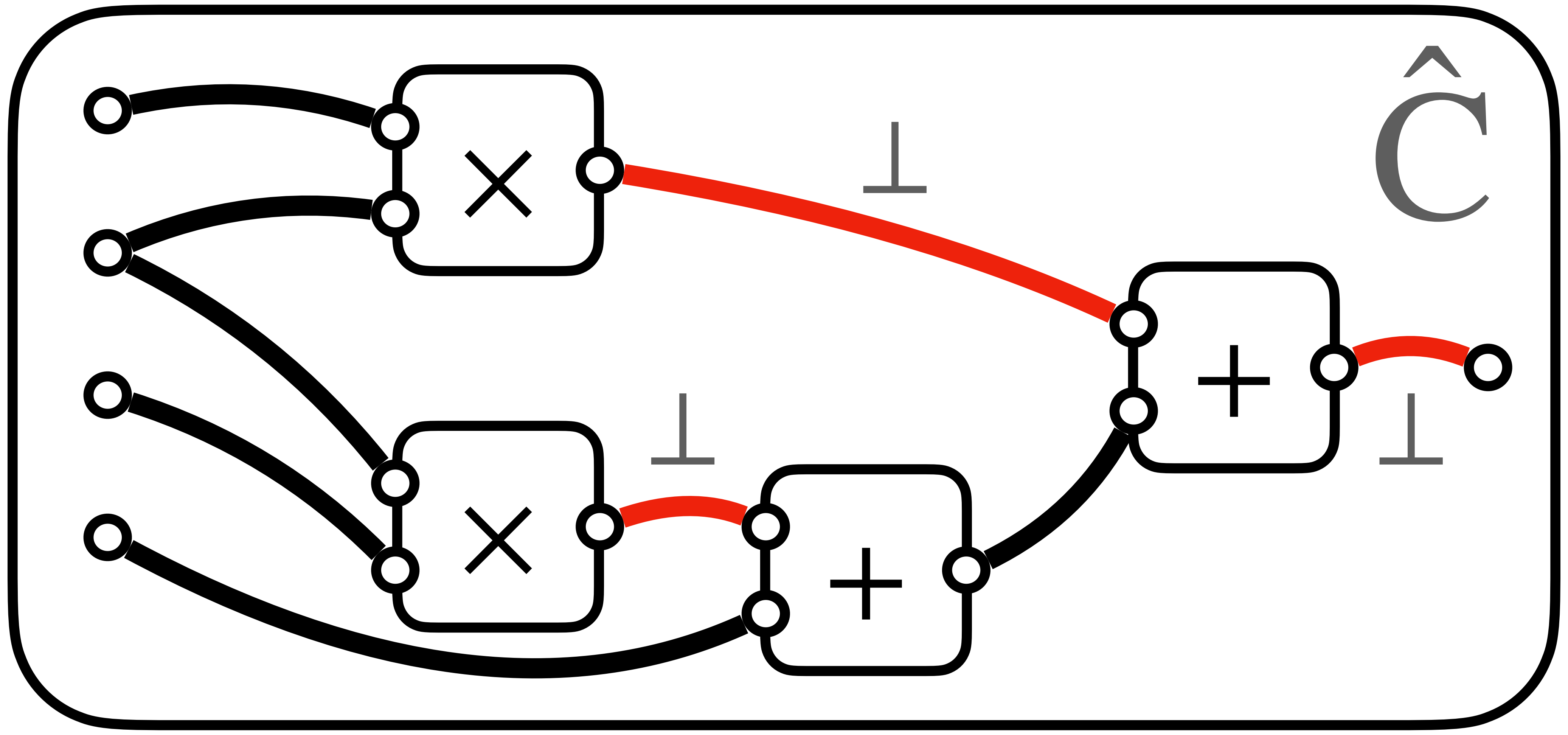
The Overflow Attack

$B \in \mathbb{Z}^+$

$\overrightarrow{x_0} \in \mathbb{Z}^*$



A fully correct GC



Where do overflows happen

$B \in \mathbb{Z}^+$

$\overrightarrow{x_1'} \in \mathbb{Z}^*$

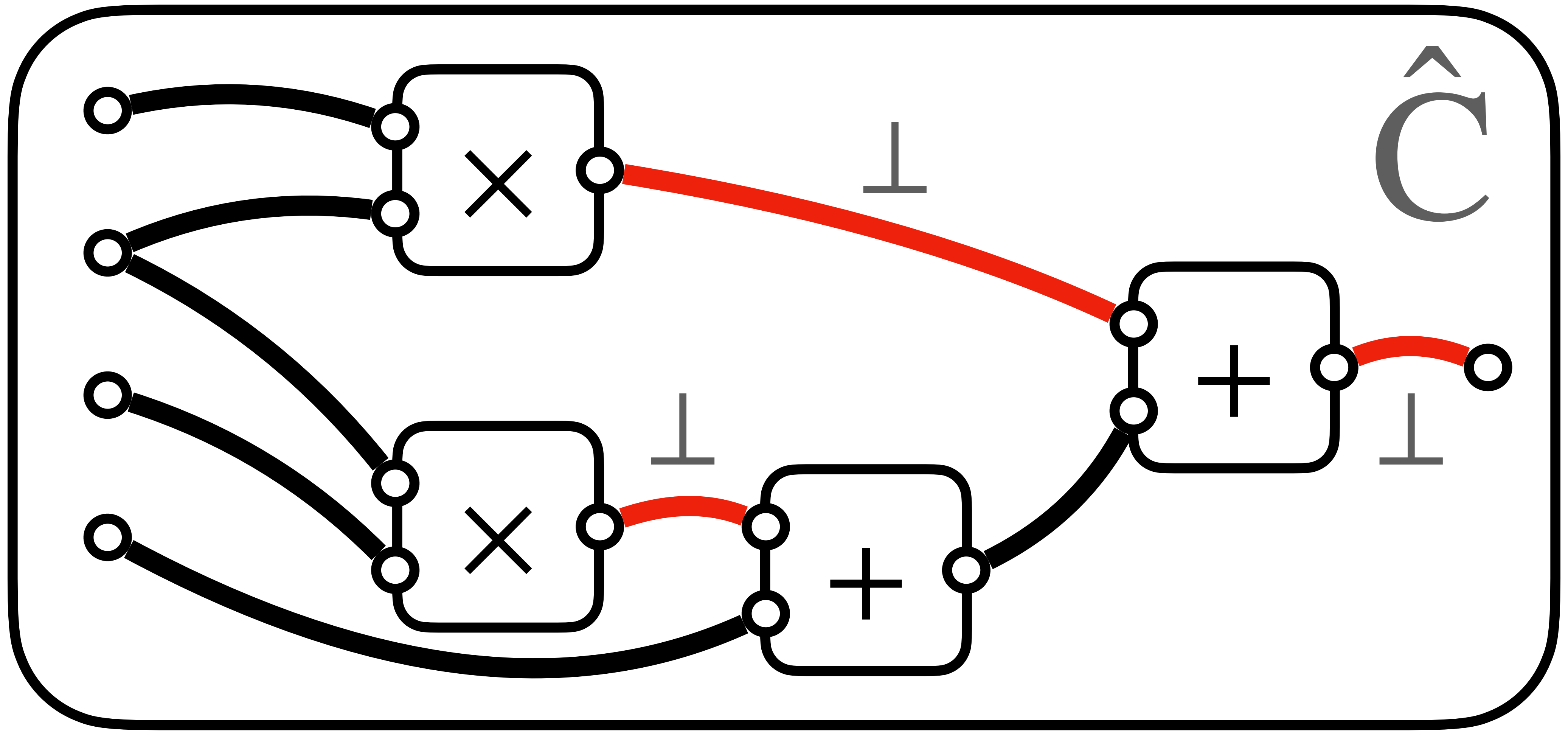
The Overflow Attack

Remark: Boolean GC is secure against malicious E

A fully correct GC

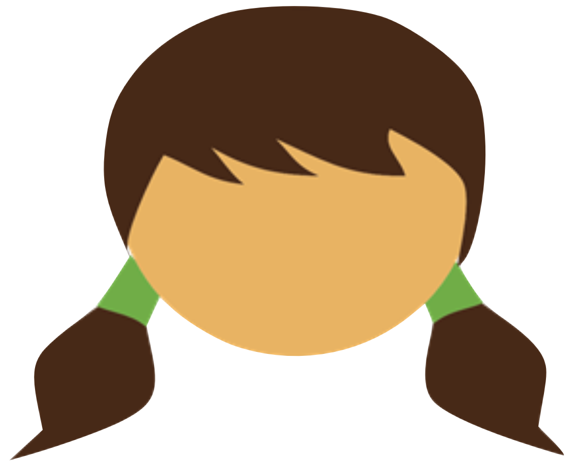
$B \in \mathbb{Z}^+$

$\overrightarrow{x_0} \in \mathbb{Z}^*$



$B \in \mathbb{Z}^+$

$\overrightarrow{x_1'} \in \mathbb{Z}^*$



Garbler

Where do overflows happen

Evaluator

The ***Best*** Achievable Security

malicious G with 1-bit leakage

and

semi-honest E

Roadmap

- Constant-rate BLLL GC
- The overflow attack
- **Our protocol based-on BLLL GC**

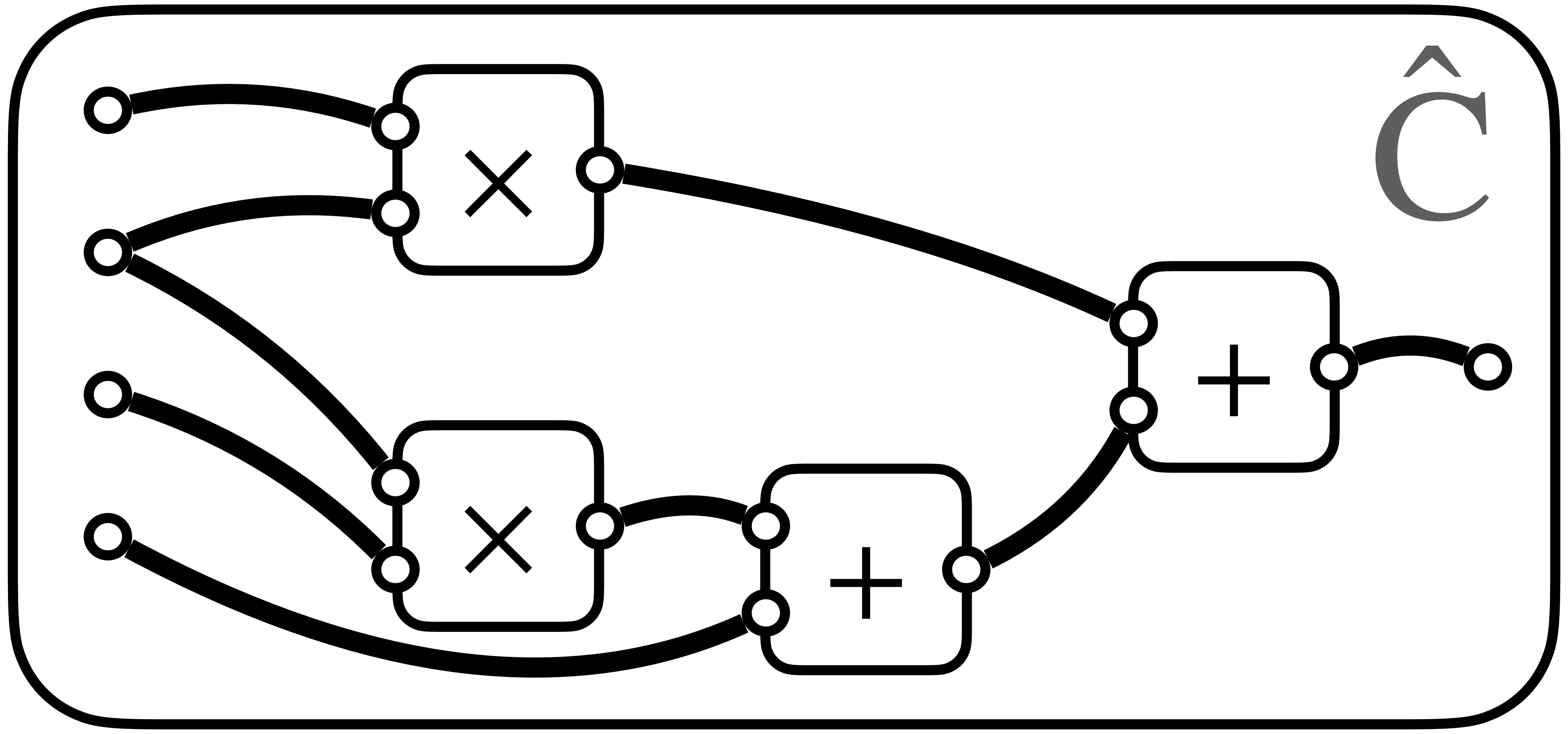
Our Protocol

How to achieve the best achievable security with constant-rate?

Our Protocol

How to achieve the best achievable security with constant-rate?

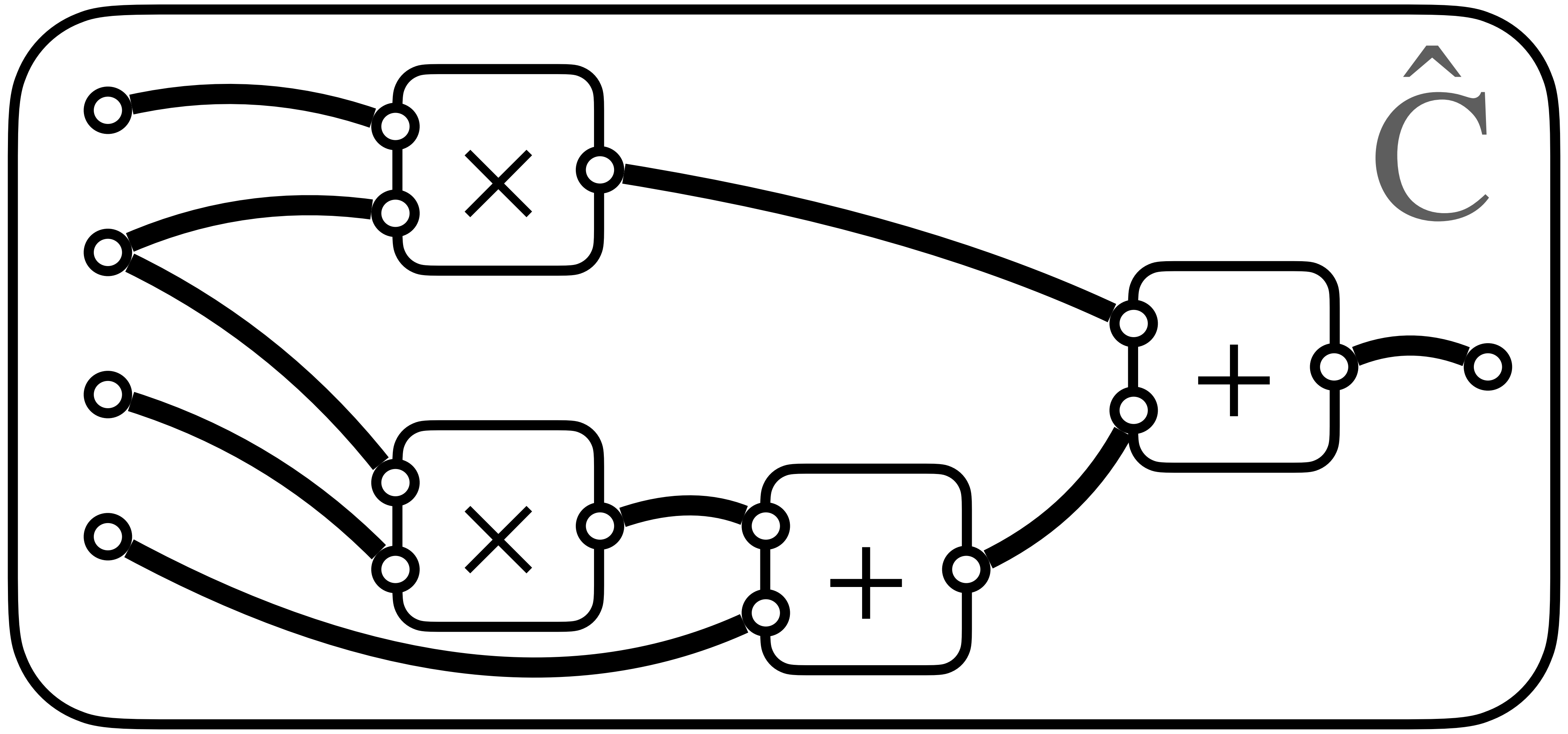
A fully correct GC



Our Protocol

How to achieve the best achievable security with constant-rate?

A fully correct GC



❌ **Cut-and-choose** $O(\lambda)$**-rate**

[LP07, …]

❓ **Authenticated garblings** $O(1)$**-rate**

[IKO+11, WRK17, …]

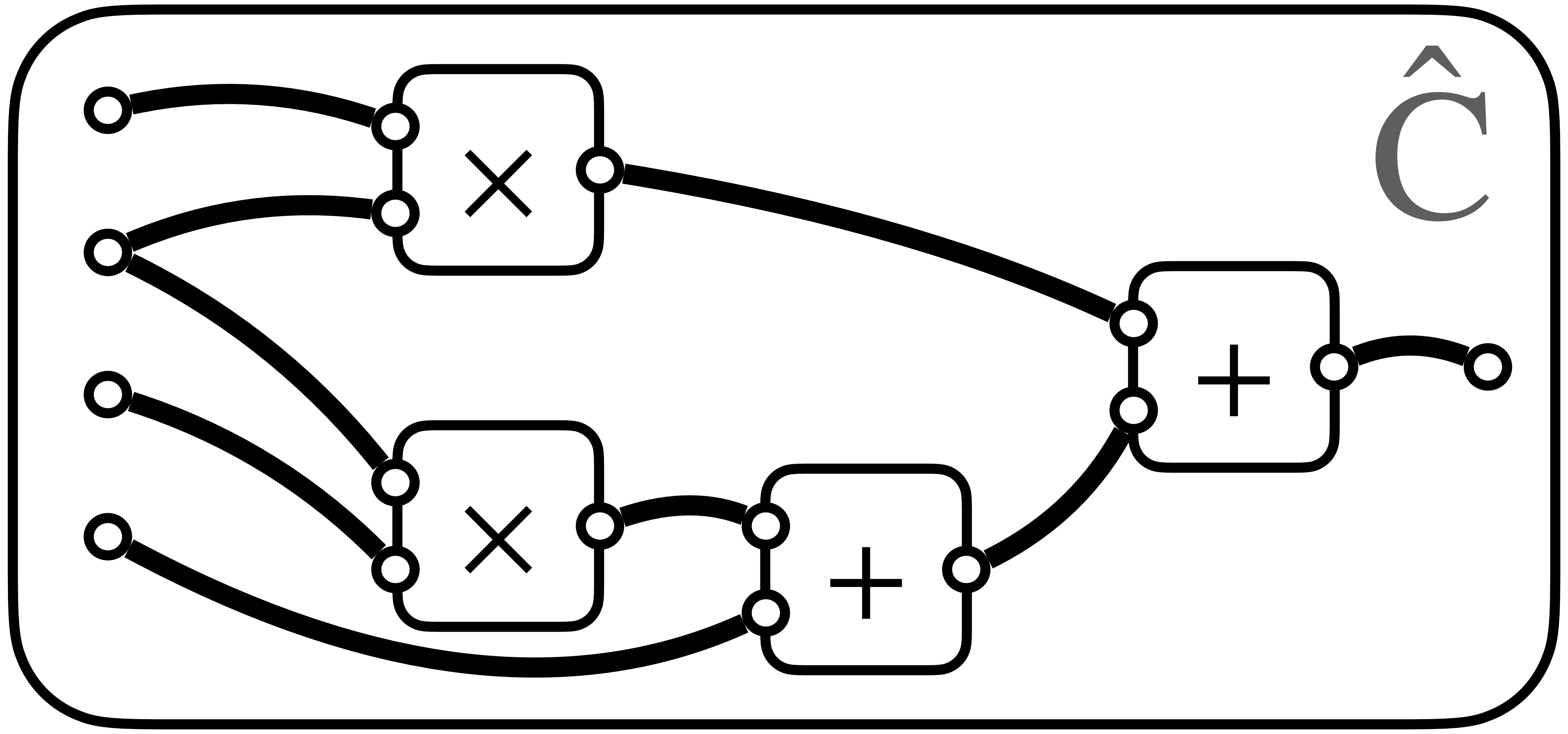
❌ **Dual execution** $O(1)$**-rate**

[MF06, …] ⚠ **1-bit leakage**

Our Protocol

How to achieve the best achievable security with constant-rate?

A fully correct GC



❌ **Cut-and-choose** $O(\lambda)$**-rate**

[LP07, …]

❓ **Authenticated garblings** $O(1)$**-rate**

[IKO+11, WRK17, …]

❌ **Dual execution** $O(1)$**-rate**

[MF06, …]

⚠ **1-bit leakage**

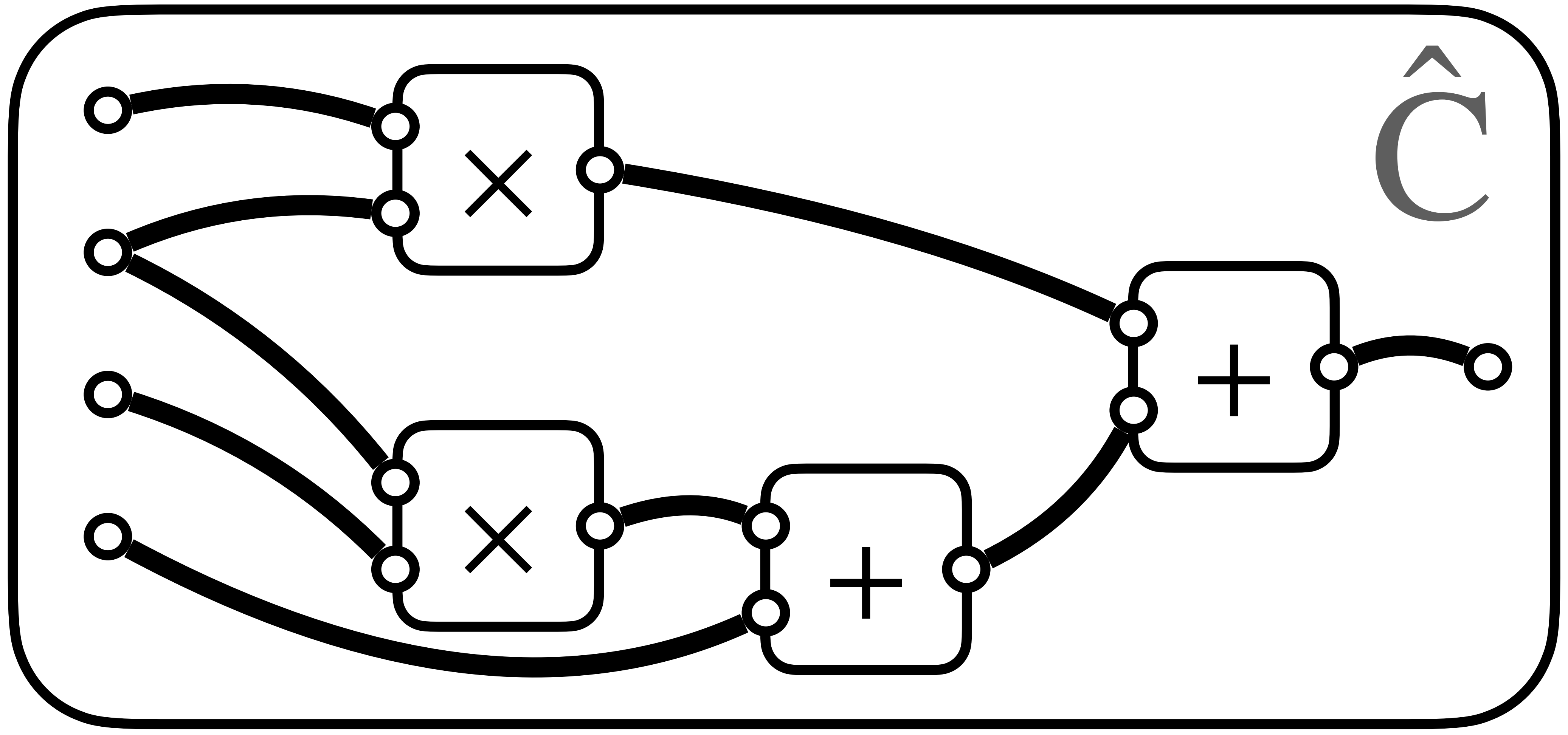
Feasibility:

GMW compiler with zkSNARK

Our Protocol

Can we do it more practical?

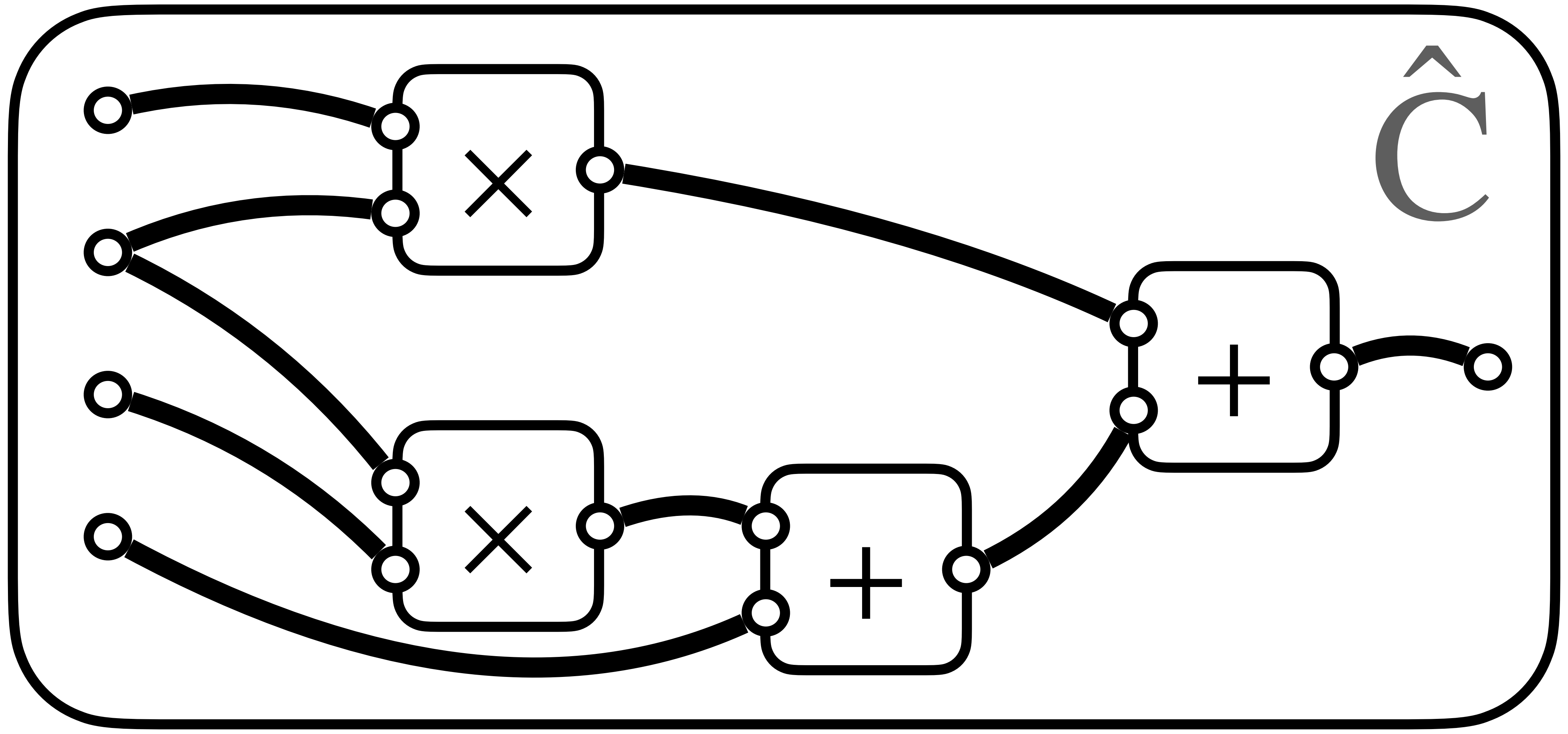
A fully correct GC



Our Protocol

Can we do it more practical?

A fully correct GC



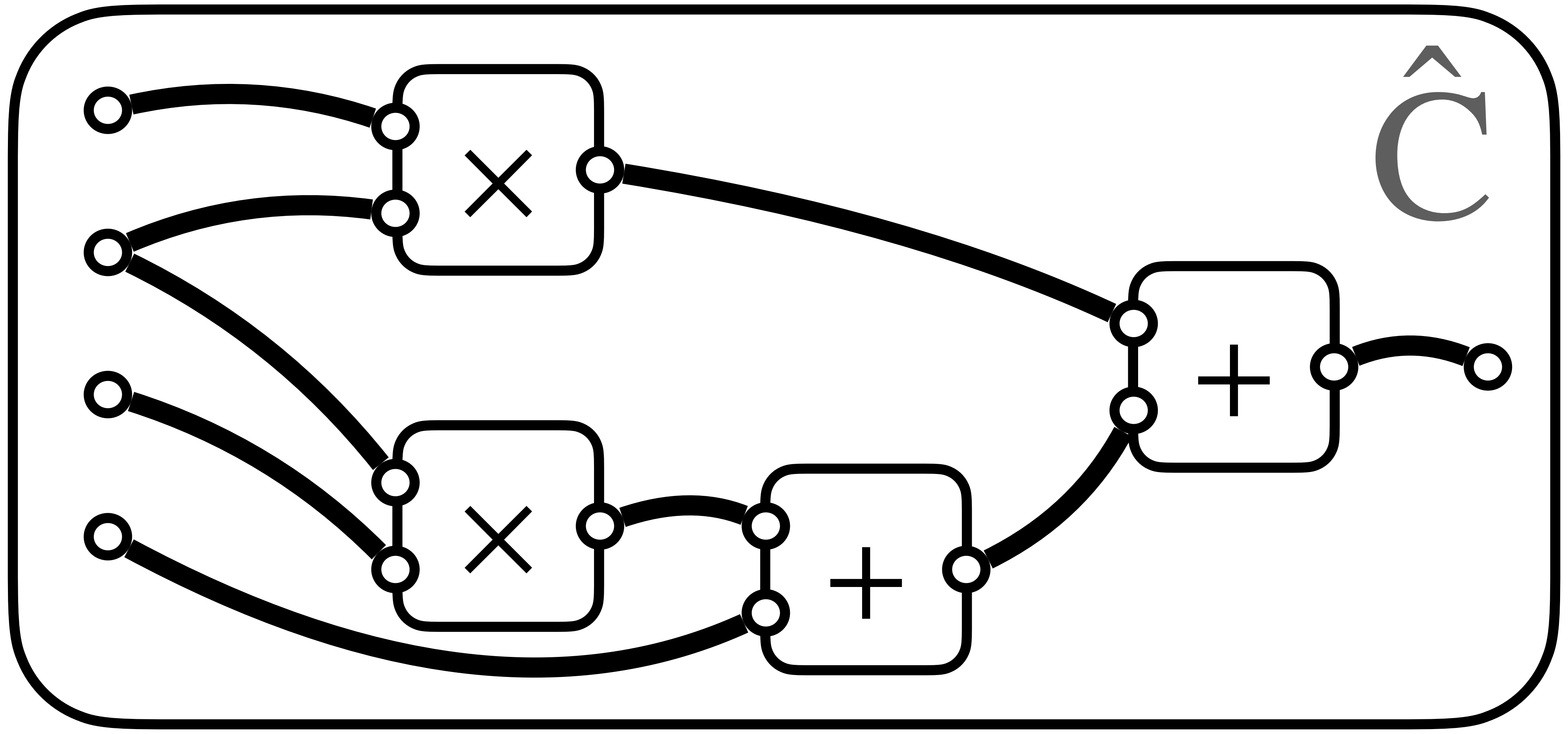
Key Insight:

If E does not abort, E will learn $C(\overrightarrow{x_0'}, \overrightarrow{x_1})$

Our Protocol

Can we do it more practical?

~~A fully~~ An almost correct GC



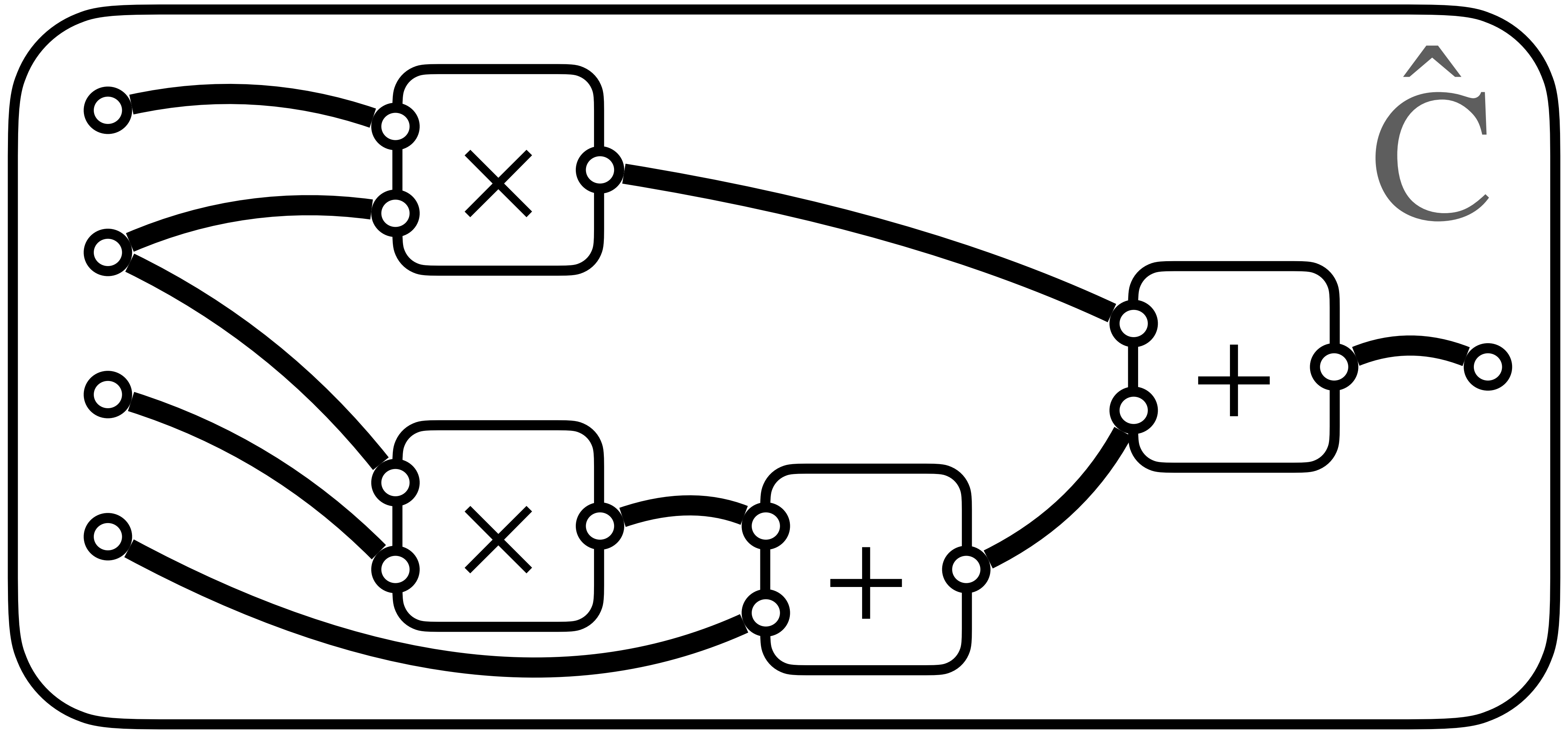
Key Insight:

If E does not abort, E will learn $C(\overrightarrow{x'_0}, \overrightarrow{x_1})$

Our Protocol

Can we do it more practical?

~~A fully~~ An almost correct GC



We design custom ZK to:

1. **Authenticate randomness**
   - **(Offline) VOLE over $\mathbb{Z}_{N^\zeta}$**
   - **(Online) ADD/MUL VOLEs**
2. **Get almost correct KE**
   - **(Online) Σ-protocol**

Key Insight:

If E does not abort, E will learn $\mathrm{C}(\overrightarrow{x_0'}, \overrightarrow{x_1})$

Conclusion

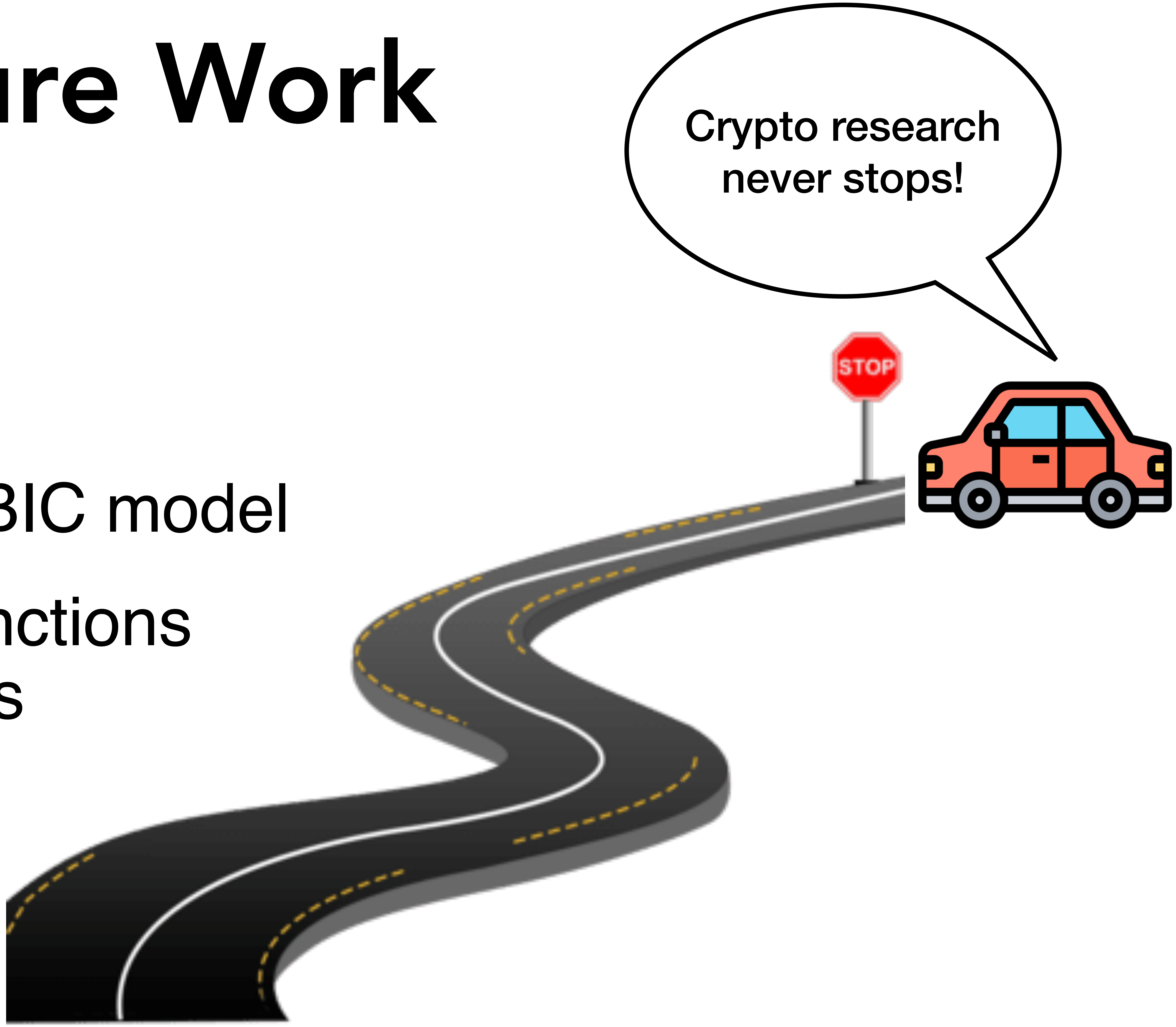
Conclusion

- Our overflow attack works for any arithmetic garbling in the BIC model, and there is no hope to get better security than 1-bit leakage.

- Our protocol achieves the first constant-rate constant-round 2PC with the best achievable security in the BIC model assuming DCR and LPN.

Future Work

- A fully correct GC
- Constant-rate GCs beyond BIC model
- Characterize the leakage functions learnt by E's overflow attacks



Q/A

ePrint: https://eprint.iacr.org/2024/283