Breaking the $1/\lambda$-Rate Barrier for Arithmetic Garbling

EUROCRYPT 2025



Geoffroy Couteau
CNRS, IRIF
Universitè Paris Citè

Carmit Hazay
Bar-Ilan University

Aditya Hegde
JHU

Naman Kumar
Oregon State University

Garbled Circuits

[Yao'86]

Garbler

Evaluator

Boolean Circuit C

Garbled Circuits

[Yao’86]

Garbler

Evaluator

Boolean Circuit C $\xrightarrow{\text{Garble}}$ Garbled Circuit $\widehat{C}$

🔑

Garbled Circuits

[Yao'86]

Garbler

Evaluator

Boolean Circuit C $\xrightarrow{\text{Garble}}$ Garbled Circuit $\widehat{C}$

Input x

Garbled Circuits

[Yao'86]

Garbler

Evaluator

Boolean Circuit C $\xrightarrow{\text{Garble}}$ Garbled Circuit $\widehat{C}$

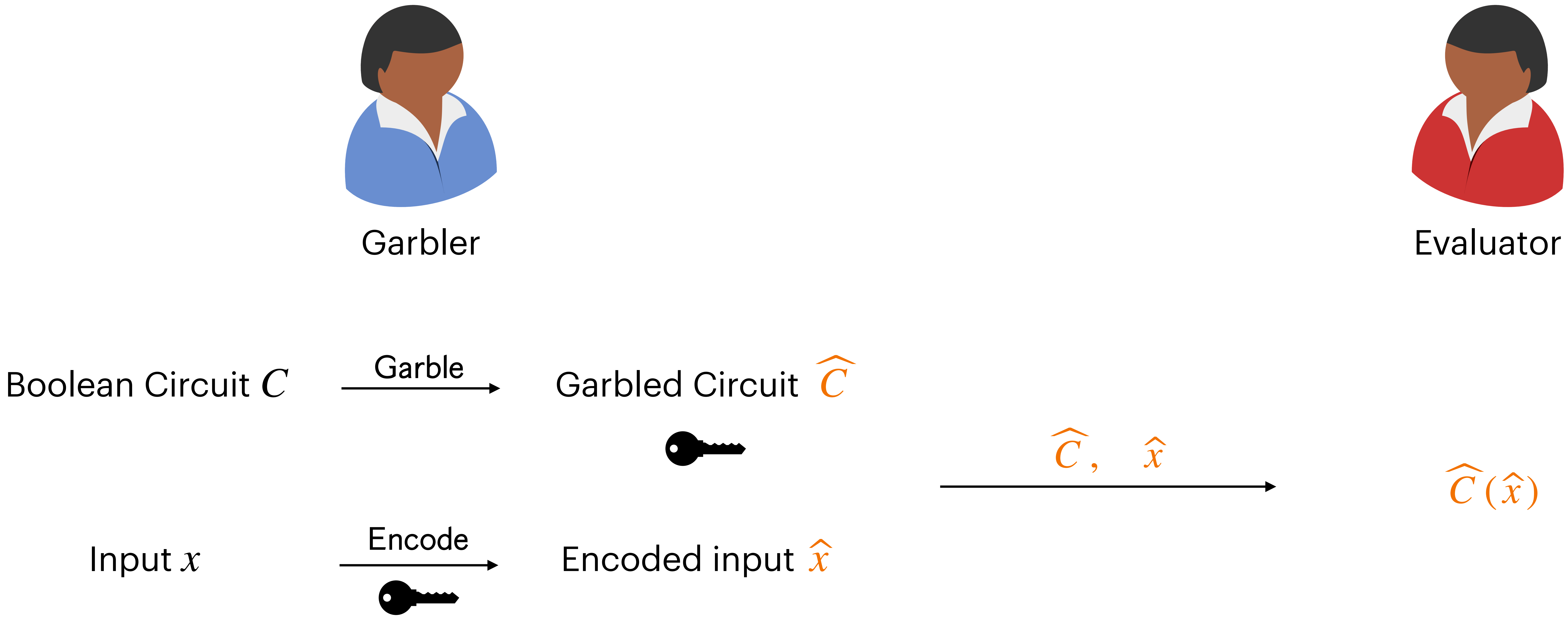
🔑

Input x $\xrightarrow{\text{Encode}}$ Encoded input $\hat{x}$

🔑

Garbled Circuits

[Yao’86]



Garbler

Evaluator

Boolean Circuit C $\xrightarrow{\text{Garble}}$ Garbled Circuit $\widehat{C}$

🔑

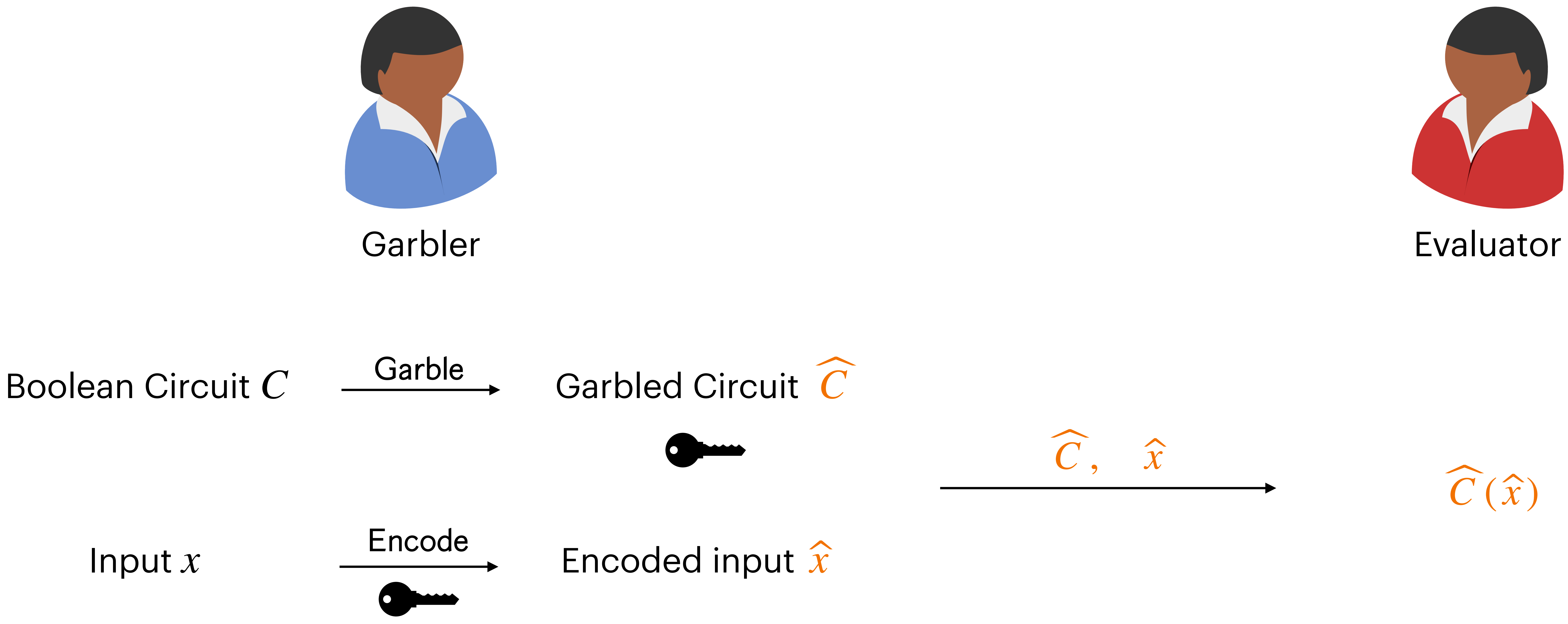
$\xrightarrow{\widehat{C},\quad \hat{x}}$ $\widehat{C}(\hat{x})$

Input x $\xrightarrow{\text{Encode}}$ Encoded input $\hat{x}$

🔑

Garbled Circuits

[Yao'86]



Garbler

Evaluator

Boolean Circuit C $\xrightarrow{\text{Garble}}$ Garbled Circuit $\widehat{C}$ 🔑

Input x $\xrightarrow[🔑]{\text{Encode}}$ Encoded input $\hat{x}$

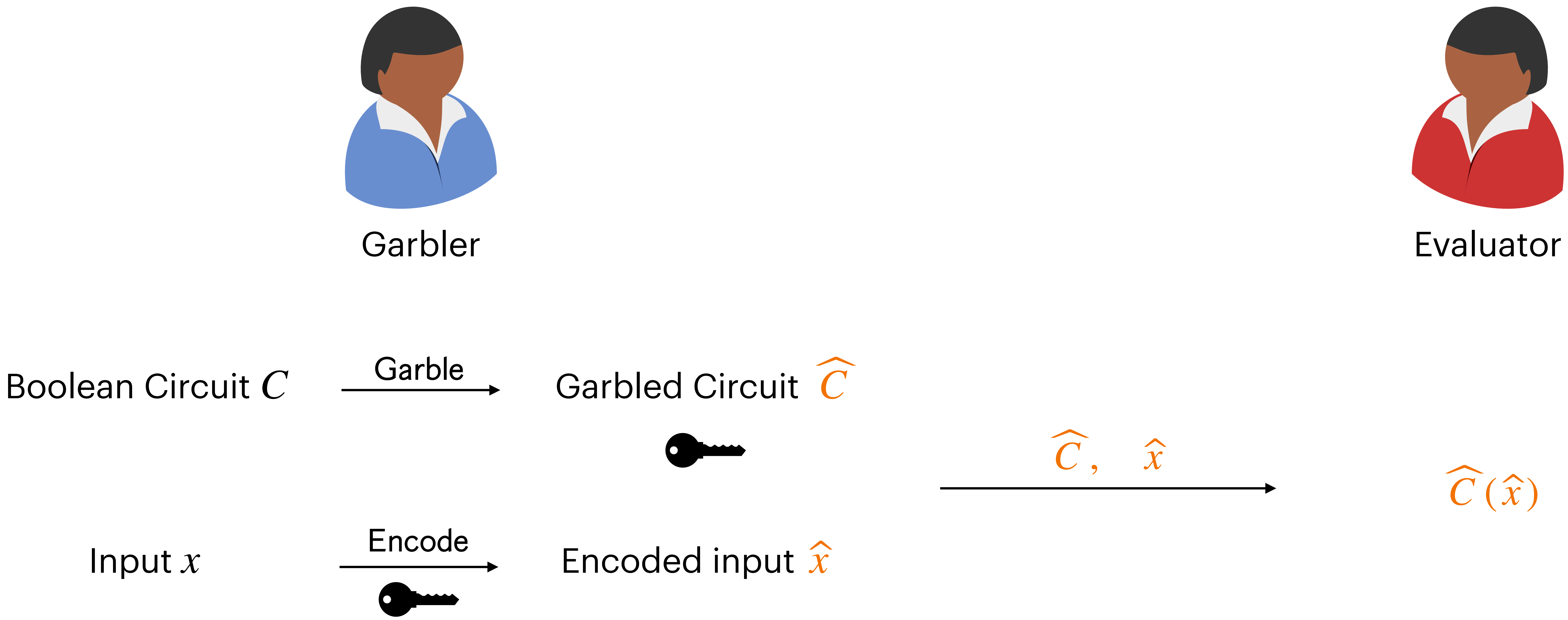
$\xrightarrow{\widehat{C},\ \hat{x}}$

$\widehat{C}(\hat{x})$

Correctness: $C(x) = \widehat{C}(\hat{x})$

Garbled Circuits

[Yao’86]



Garbler

Evaluator

Boolean Circuit C $\xrightarrow{\text{Garble}}$ Garbled Circuit $\widehat{C}$

Input x $\xrightarrow{\text{Encode}}$ Encoded input $\hat{x}$

$\xrightarrow{\widehat{C}, \quad \hat{x}}$

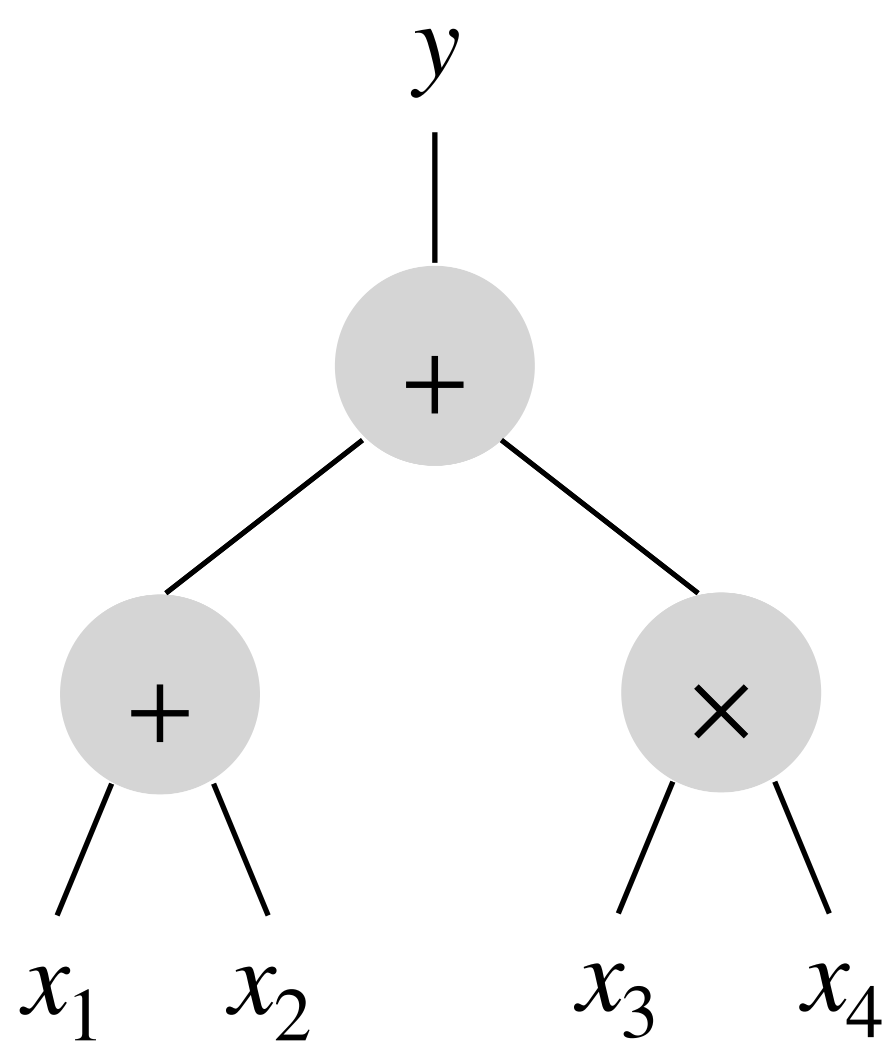
$\widehat{C}(\hat{x})$

Correctness: $C(x) = \widehat{C}(\hat{x})$

Privacy: $\widehat{C}$, $\hat{x}$ reveal nothing beyond the output $C(x)$

Arithmetic Garbling

[Applebaum-Ishai-Kushilevitz'11]



Arithmetic Circuit:

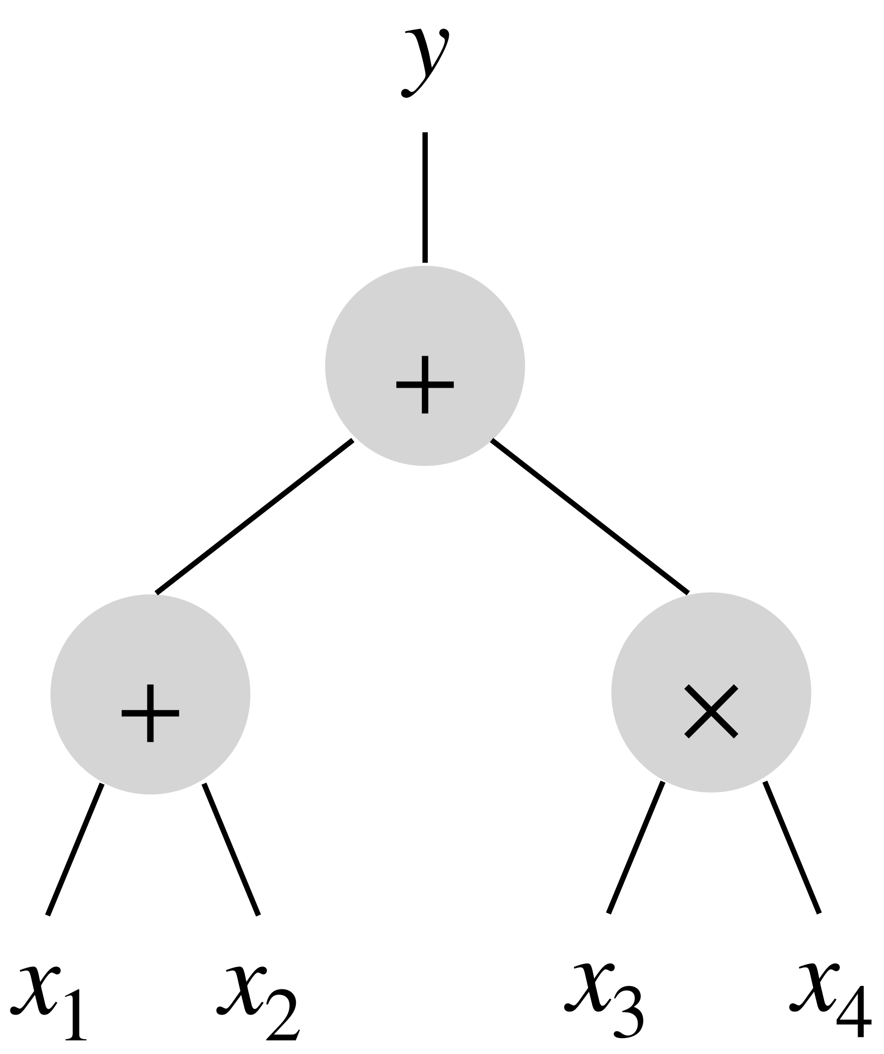
Computation over a ring R

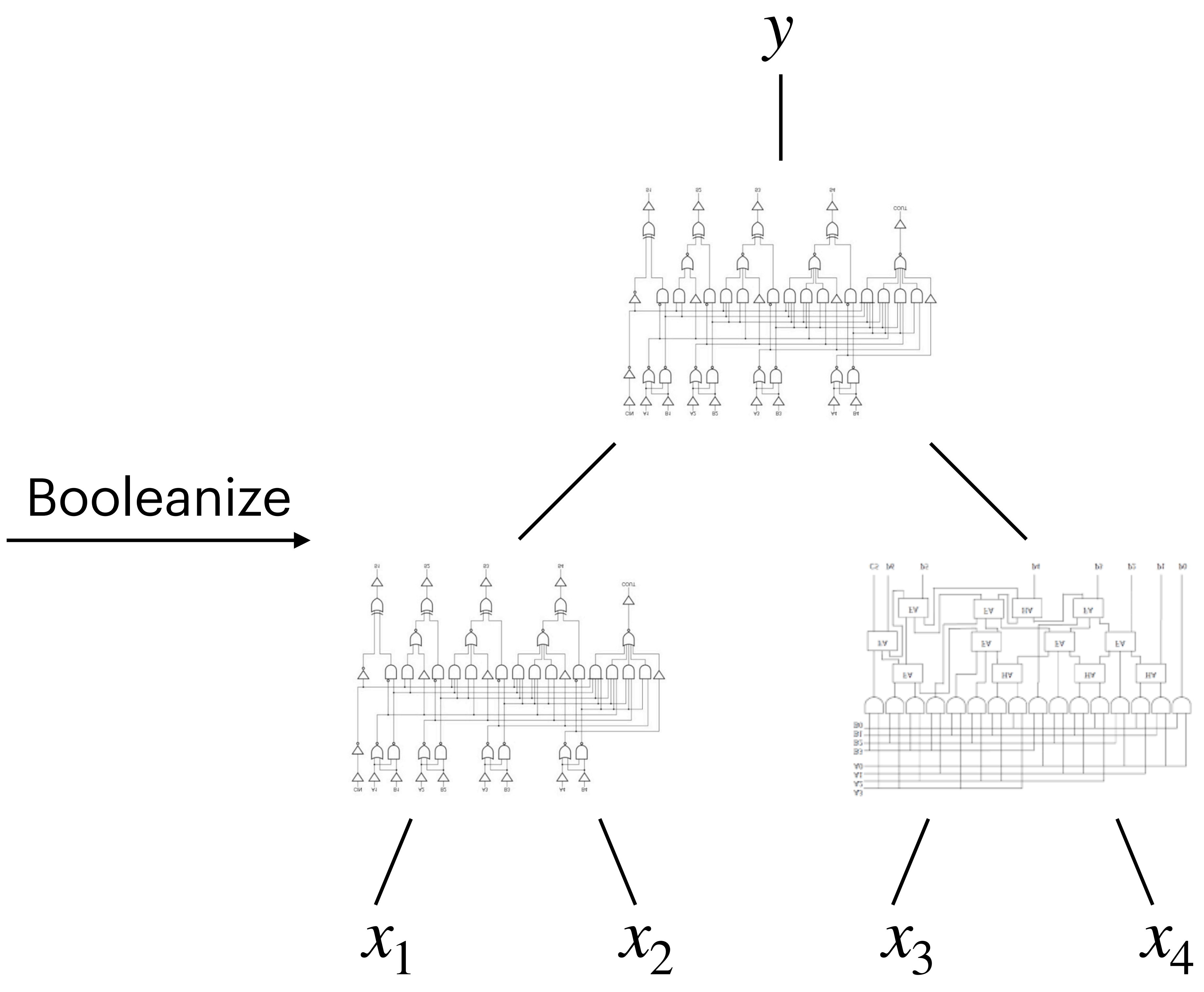
Common in

- Scientific computing
- Machine learning
- Cryptography

Arithmetic Garbling

[Applebaum-Ishai-Kushilevitz’11]





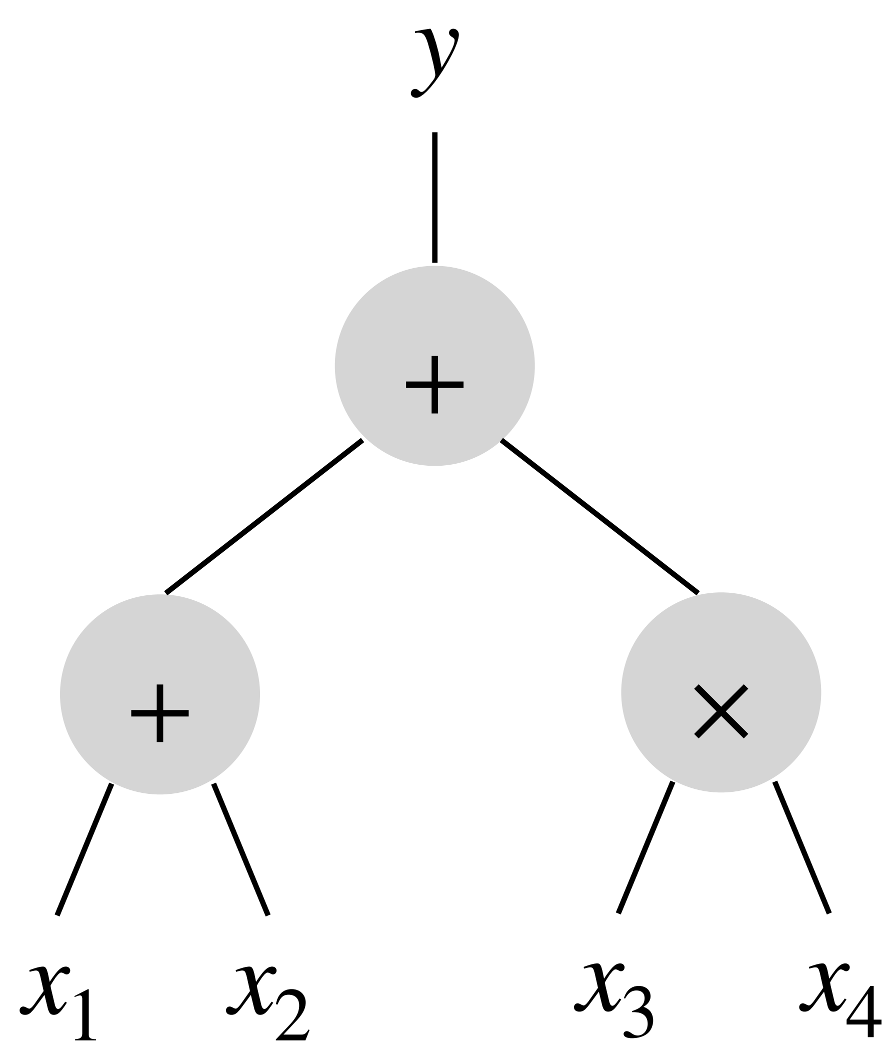
Arithmetic Circuit:
Computation over a ring R

Common in

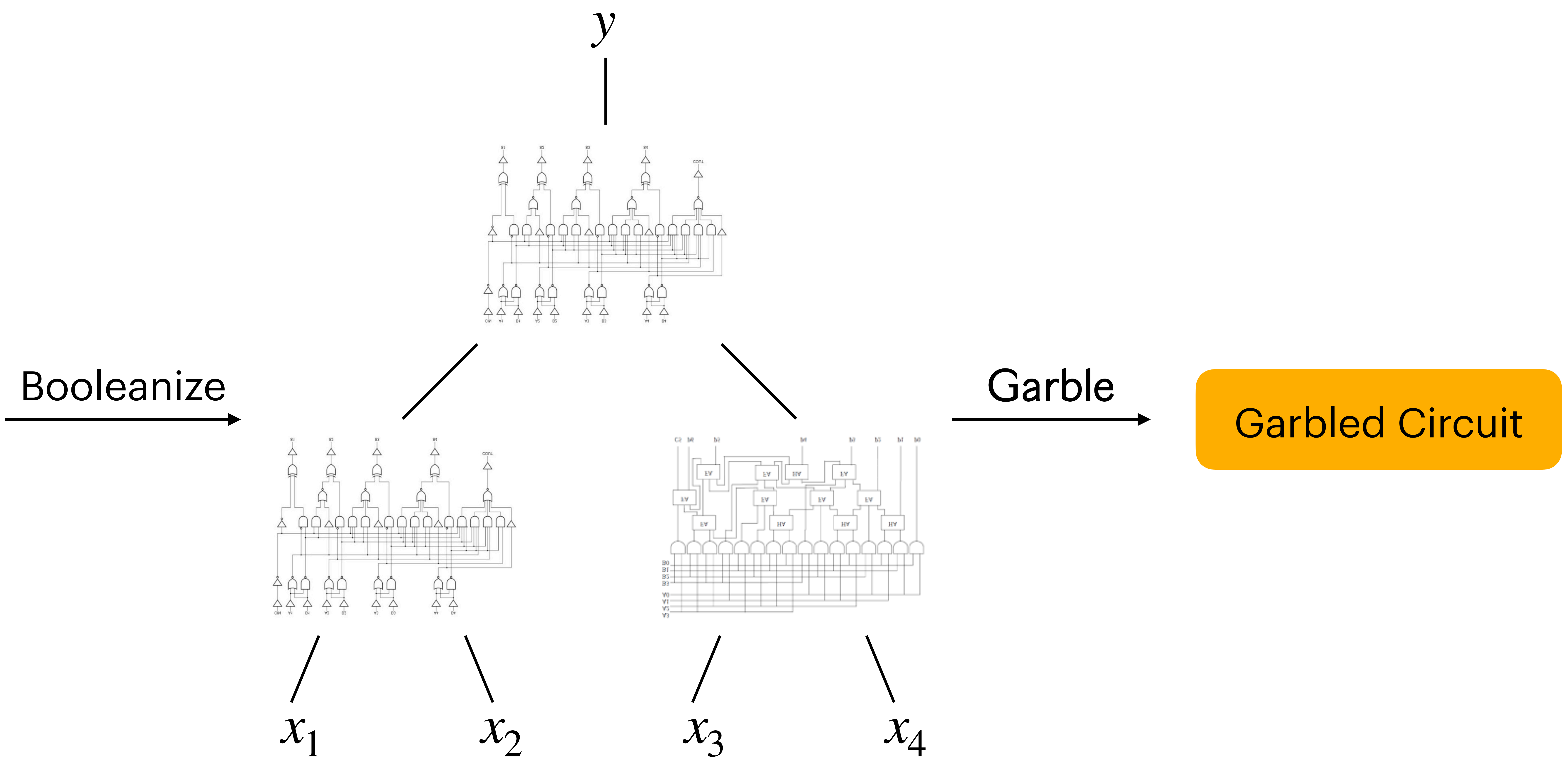
- Scientific computing
- Machine learning
- Cryptography

Arithmetic Garbling

[Applebaum-Ishai-Kushilevitz’11]



Arithmetic Circuit:
Computation over a ring R

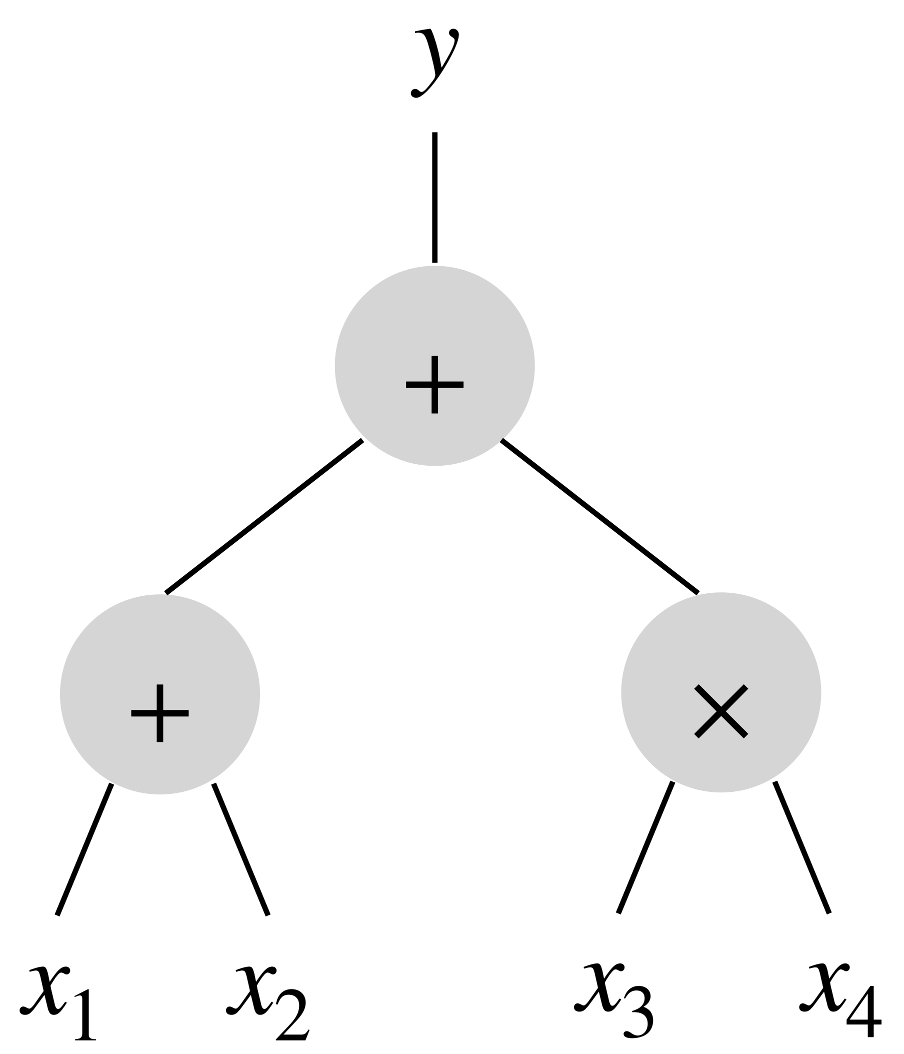


Common in

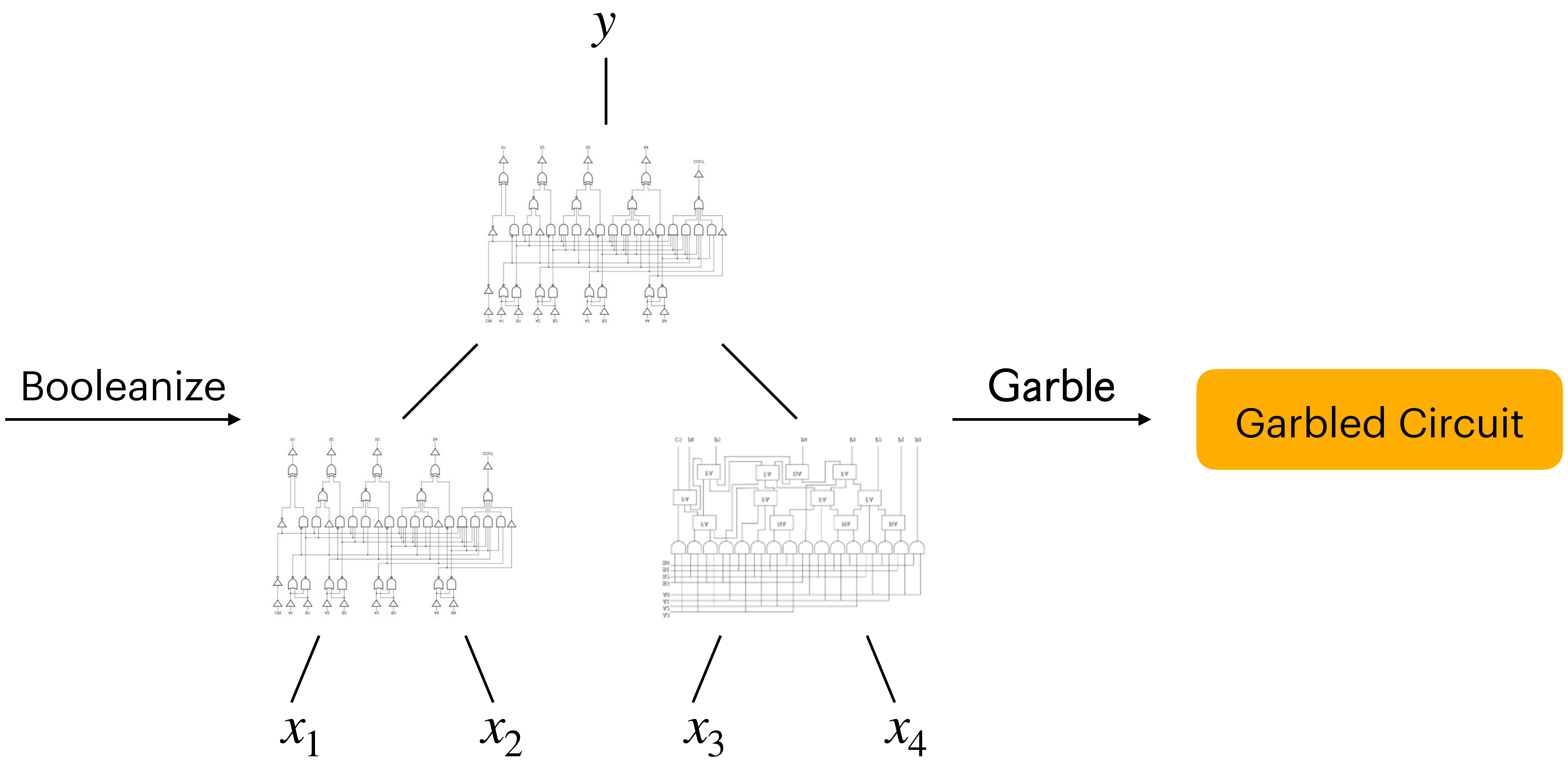
- Scientific computing
- Machine learning
- Cryptography

Arithmetic Garbling

[Applebaum-Ishai-Kushilevitz’11]



Arithmetic Circuit:
Computation over a ring R



Common in

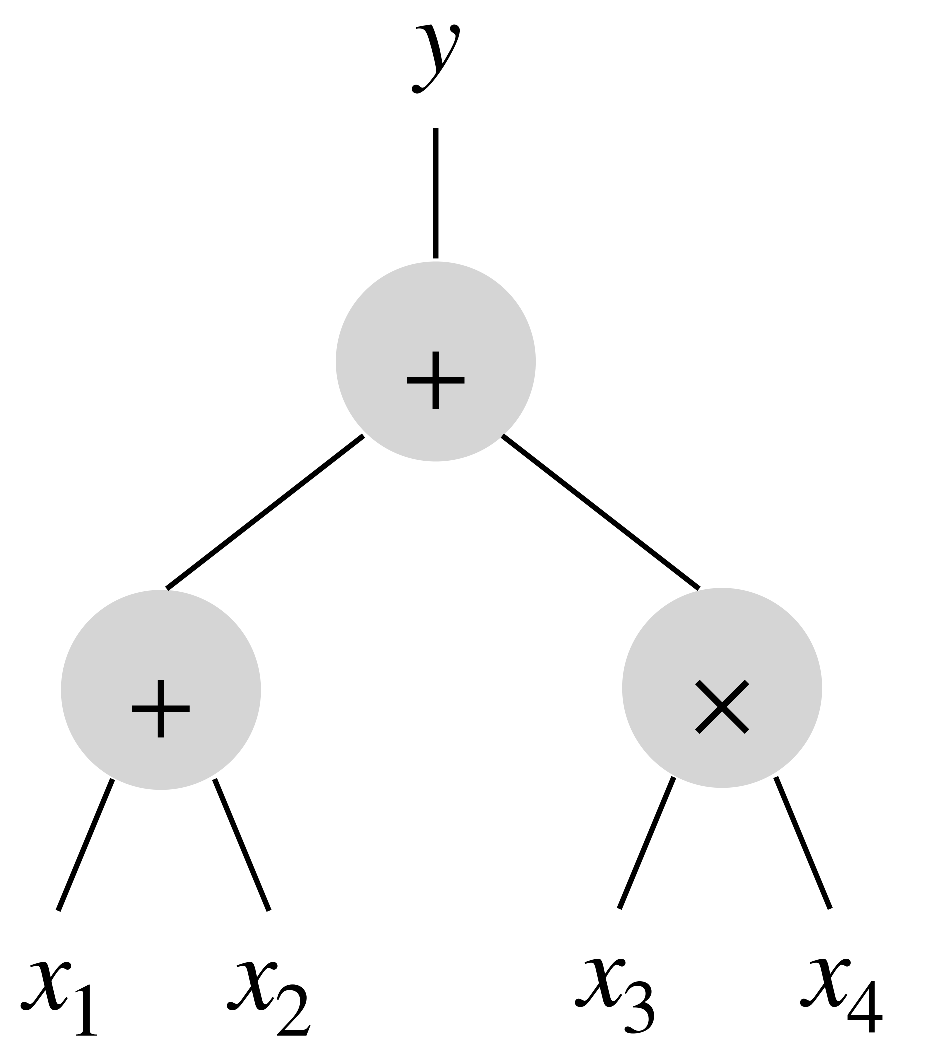
- Scientific computing
- Machine learning
- Cryptography

Equivalent boolean circuit has larger size

Requires bit-decomposition of inputs

Arithmetic Garbling

[Applebaum-Ishai-Kushilevitz’11]

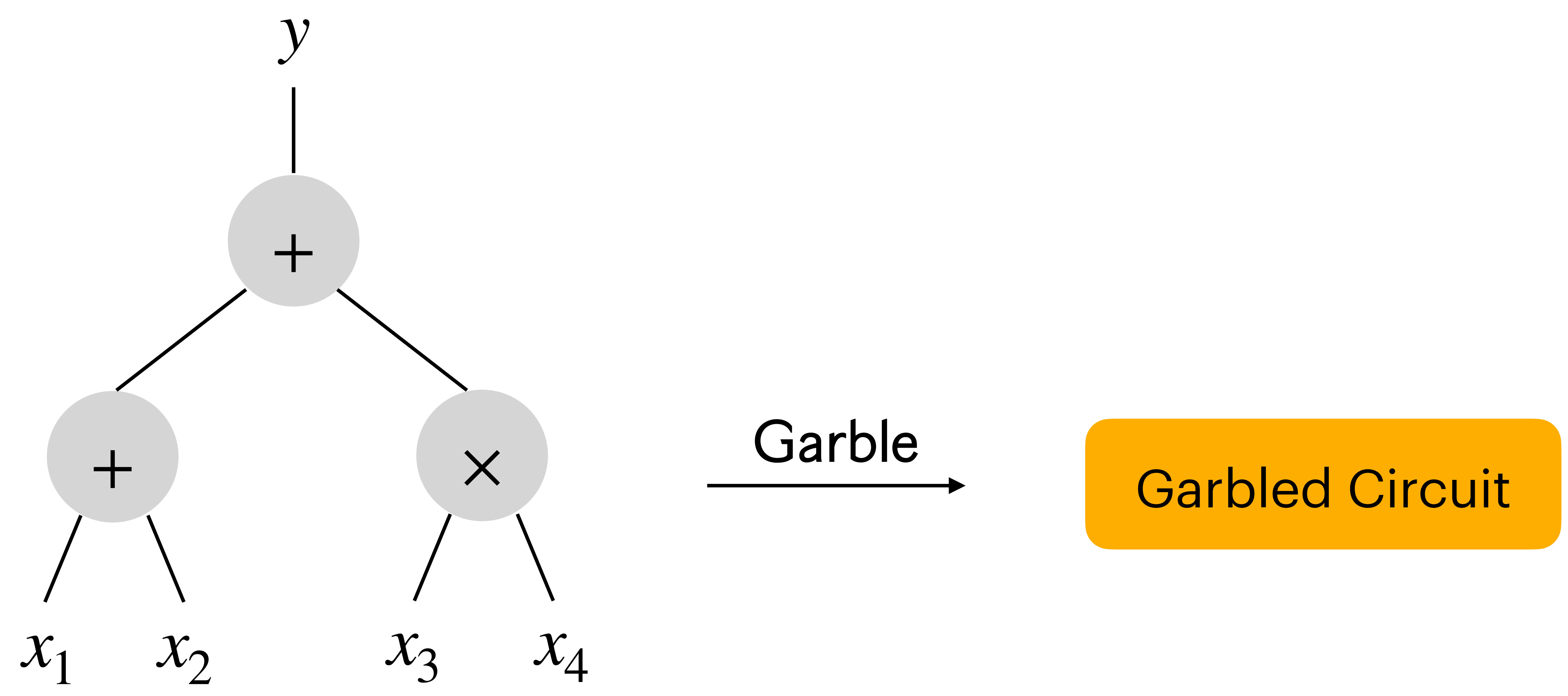


Arithmetic Circuit:
Computation over a ring R

$\xrightarrow{\text{Garble}}$ Garbled Circuit

Can we directly garble arithmetic circuits without booleanizing?

Rate of Arithmetic Garbling Schemes



Arithmetic Circuit:
Computation over a ring R

Higher rate $\Longrightarrow$ more efficient scheme

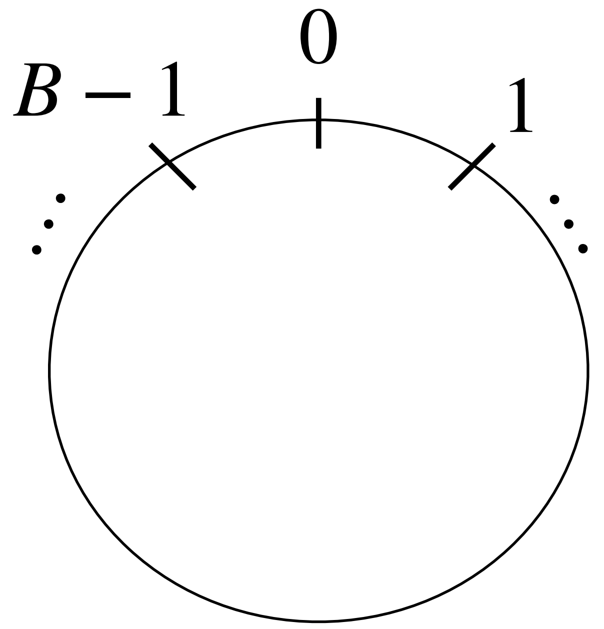
Rate:
$$\frac{\boxed{\text{Arithmetic Circuit}} \cdot \log|R|}{\boxed{\text{Garbled Circuit}}} = \frac{\text{Number of bits to represent all wire values}}{\boxed{\text{Garbled Circuit}}}$$

Landscape of Arithmetic Garbling

Bounded Integer Arithmetic

Modular Arithmetic

B-bounded computation over $\mathbb{Z}$



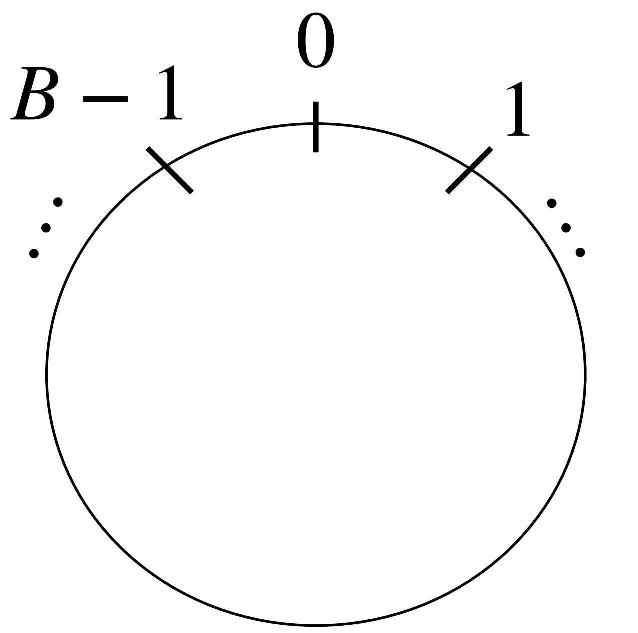
Computation over $\mathbb{Z}_B$

Landscape of Arithmetic Garbling

Bounded Integer Arithmetic

Modular Arithmetic

B-bounded computation over $\mathbb{Z}$



Computation over $\mathbb{Z}_B$

Rate-1 arithmetic garbling

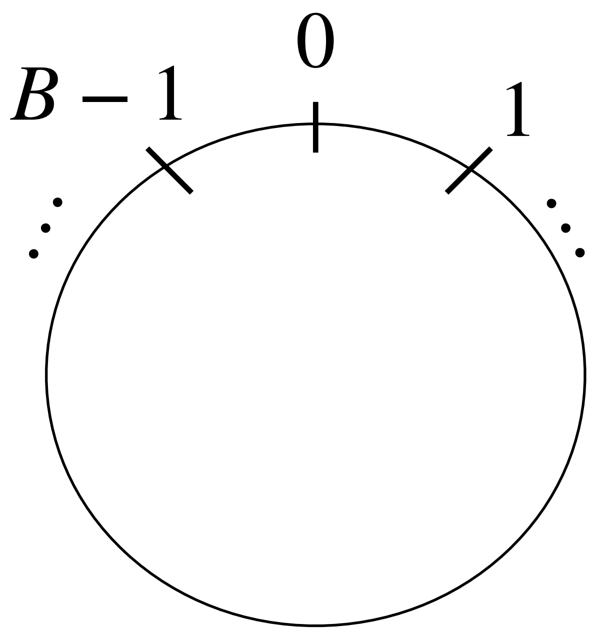
[Meyer-Orlandi-Roy-Scholl'24]

Landscape of Arithmetic Garbling

Bounded Integer Arithmetic

B-bounded computation over $\mathbb{Z}$

Modular Arithmetic



Computation over $\mathbb{Z}_B$

Rate-1 arithmetic garbling

[Meyer-Orlandi-Roy-Scholl'24]

Rate-$O(1/\lambda)$ arithmetic garbling

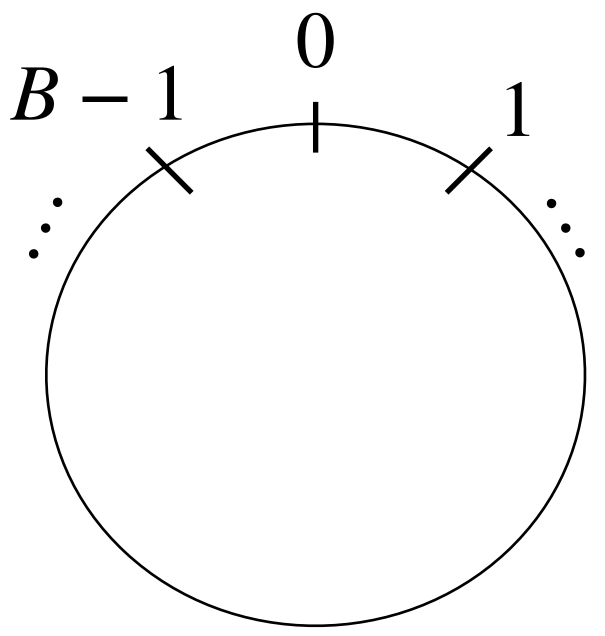
[Ball-Li-Lin-Liu'23] [Heath'24]

Landscape of Arithmetic Garbling

Bounded Integer Arithmetic

Modular Arithmetic

B-bounded computation over $\mathbb{Z}$



Computation over $\mathbb{Z}_B$

Rate-1 arithmetic garbling

[Meyer-Orlandi-Roy-Scholl'24]

Rate-$O(1/\lambda)$ arithmetic garbling

[Ball-Li-Lin-Liu'23] [Heath'24]

Can we break the $O(1/\lambda)$ rate barrier for modular arithmetic garbling?

Our Results

Assuming power-DDH and the existence of a tweakable correlation robust hash function, there exists an arithmetic garbling scheme for any polynomial size integer ring $\mathbb{Z}_B$ with rate

$$\Theta\left(\frac{\log B}{\lambda}\right)$$

Our Results

Assuming power-DDH and the existence of a tweakable correlation robust hash function, there exists an arithmetic garbling scheme for any polynomial size integer ring $\mathbb{Z}_B$ with rate

$$\Theta\left(\frac{\log B}{\lambda}\right)$$

Bounded Integer Arithmetic

Modular Arithmetic

Rate-1 arithmetic garbling
[Meyer-Orlandi-Roy-Scholl'24]

Rate-$O(1/\lambda)$ arithmetic garbling
[Ball-Li-Lin-Liu'23] [Heath'24]

Rate-$O(\log \lambda/\lambda)$ arithmetic garbling

Our Results

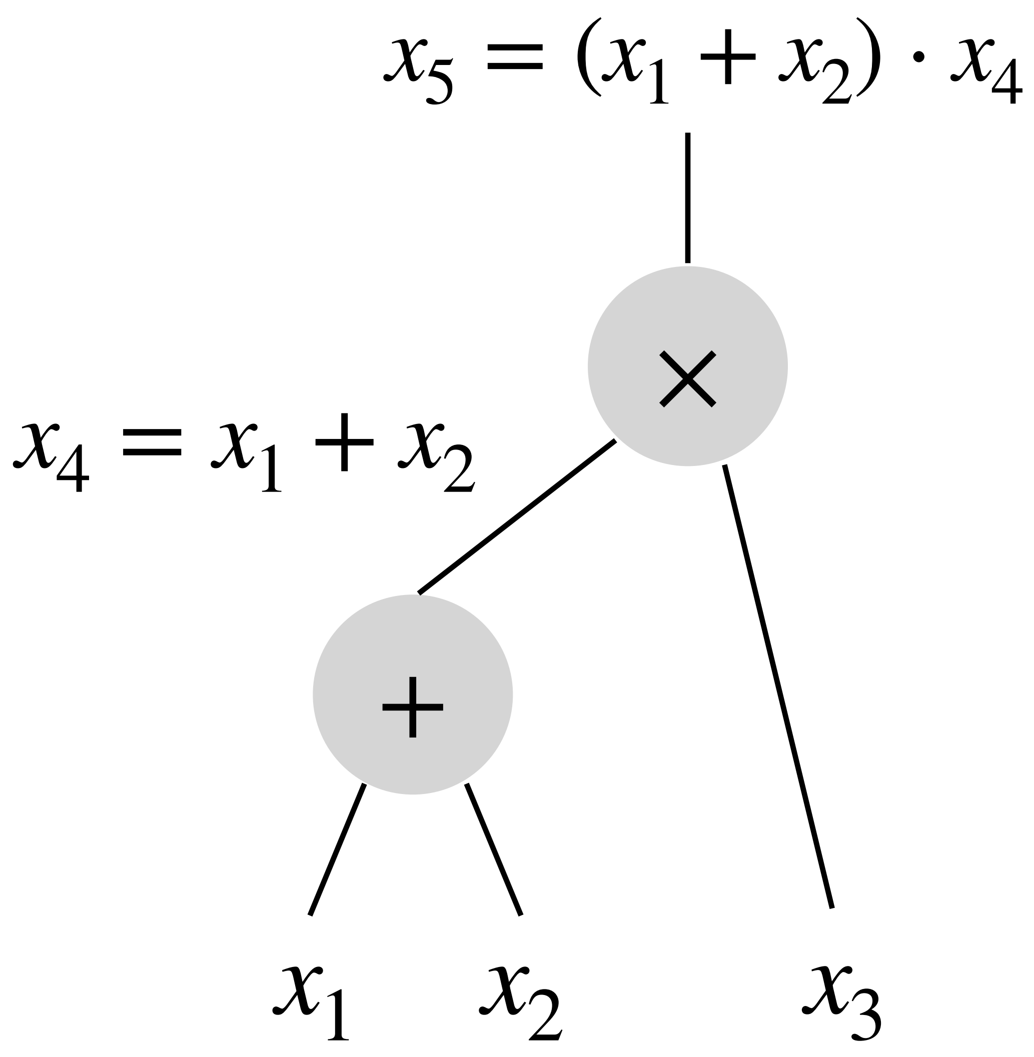
Assuming power-DDH and the existence of a tweakable correlation robust hash function, there exists an arithmetic garbling scheme for any polynomial size integer ring $\mathbb{Z}_B$ with rate

$$\Theta\left(\frac{\log B}{\lambda}\right)$$

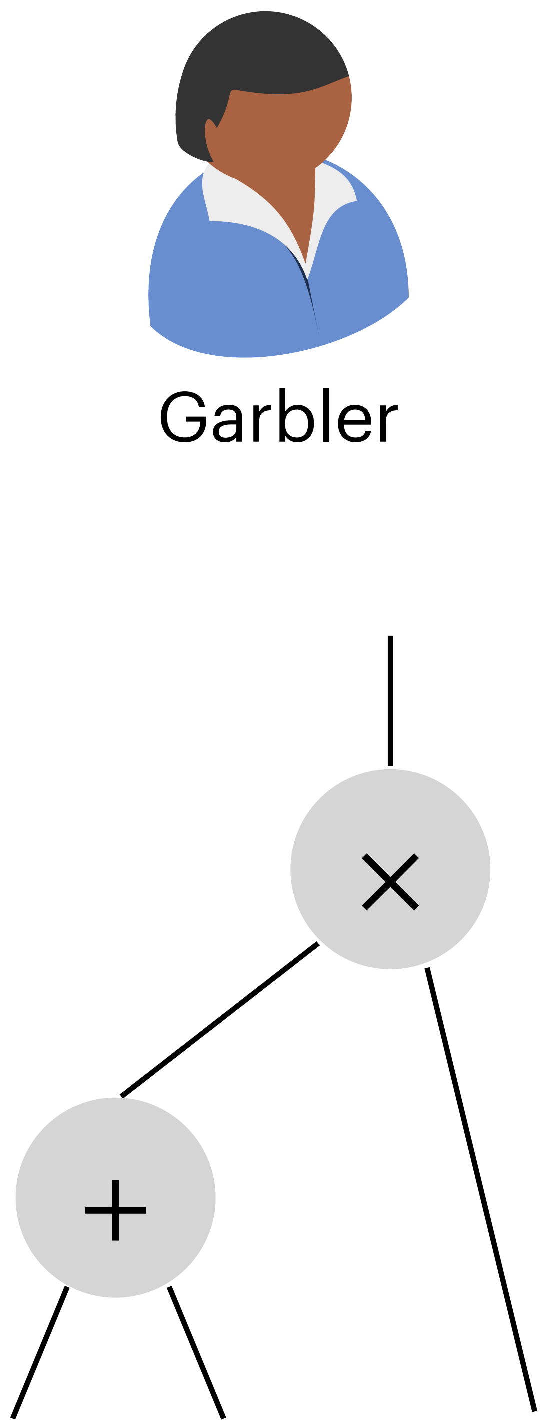
A key component of our construction is a new power-DDH based Punctured Pseudorandom Function (PPRF)

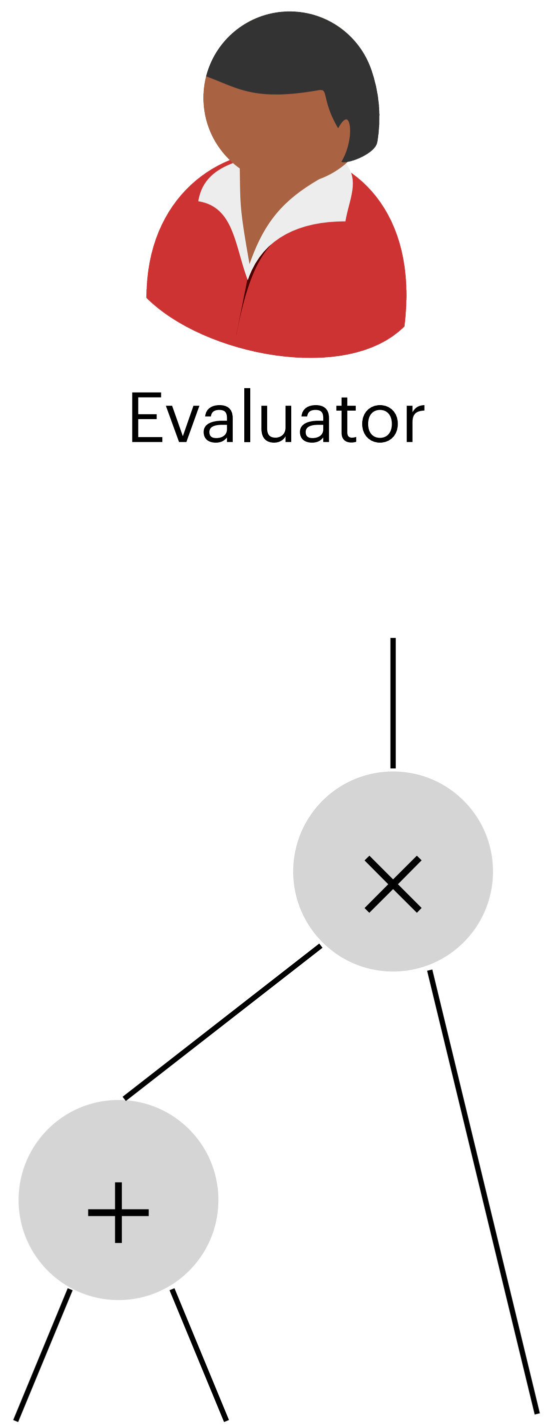
- **Reusable setup:** Common setup can be reused for multiple PRF keys
- **Small key size:** Single element in $\mathbb{Z}_q^*$

Template for Garbling Circuits

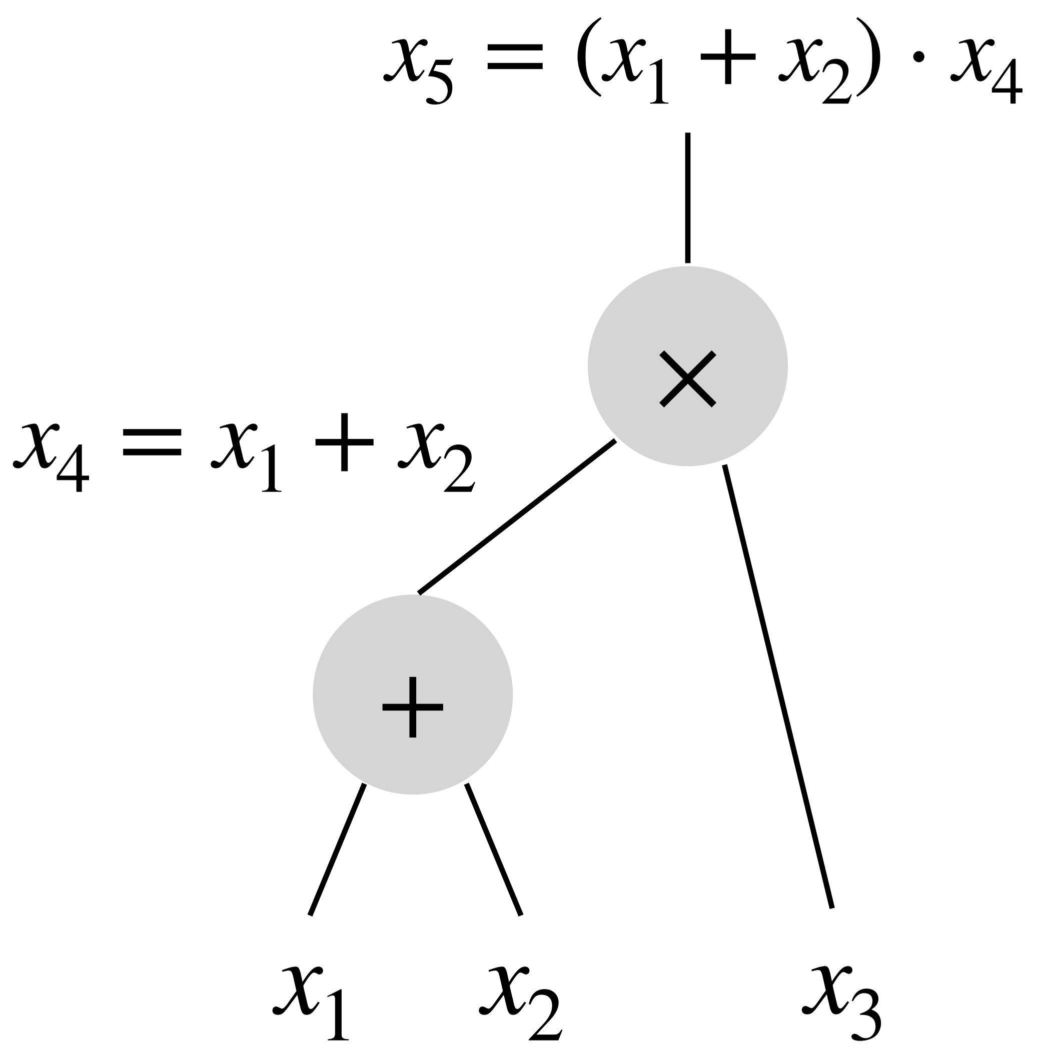


Arithmetic circuit over $\mathbb{Z}_B$

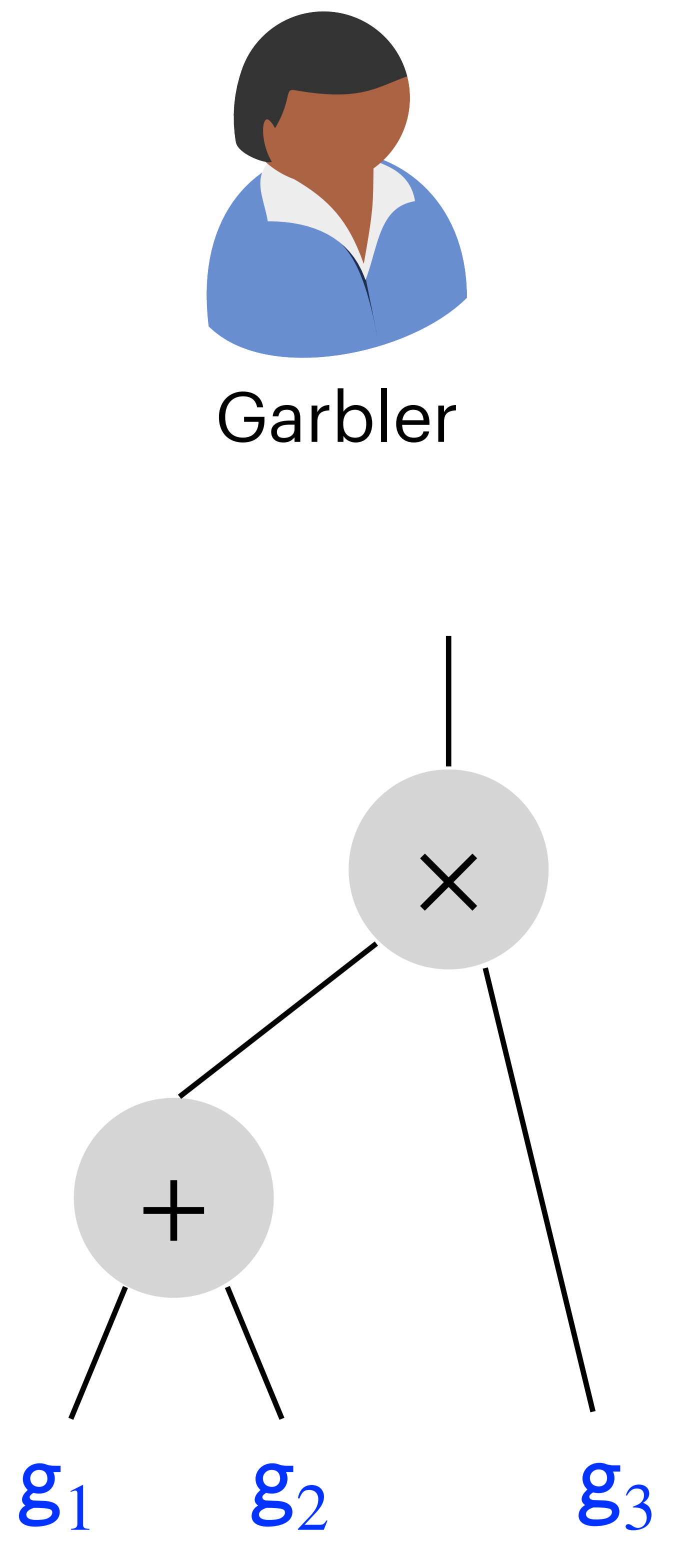




Template for Garbling Circuits

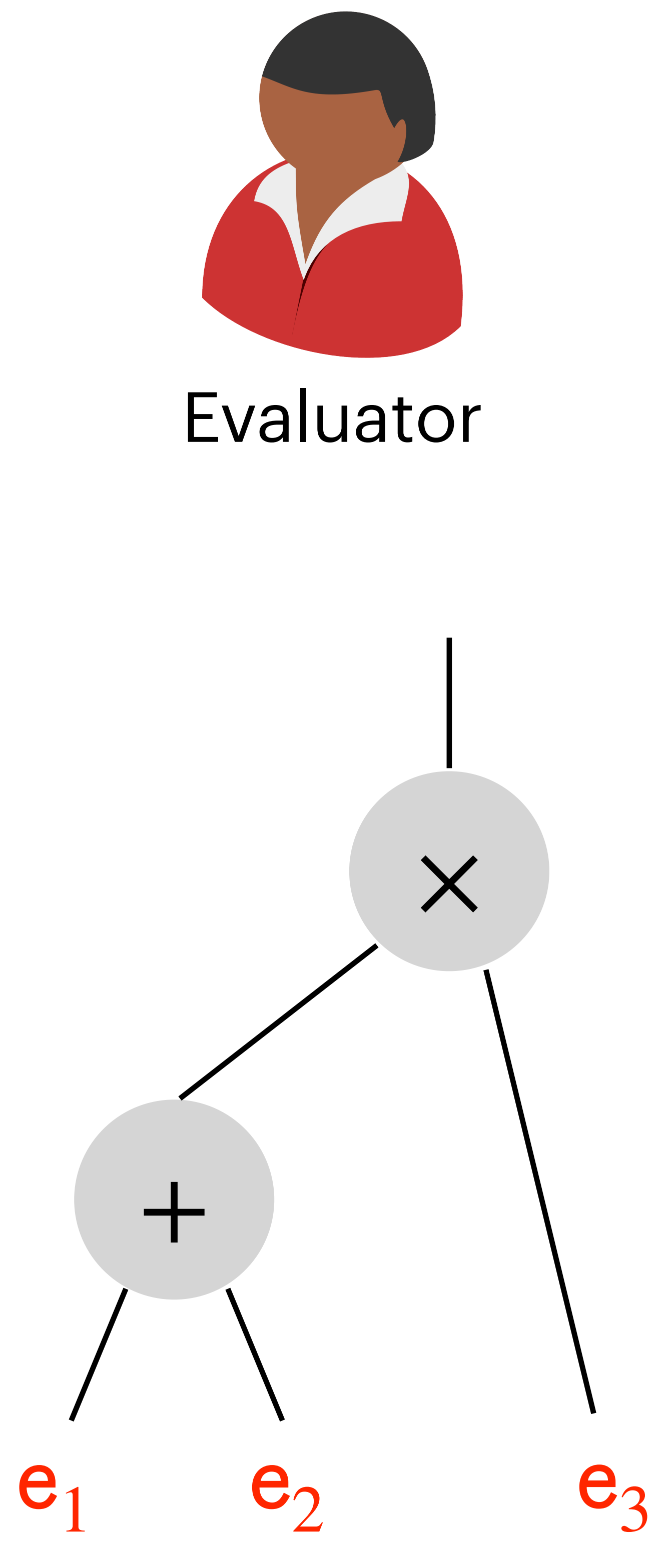


Arithmetic circuit over $\mathbb{Z}_B$

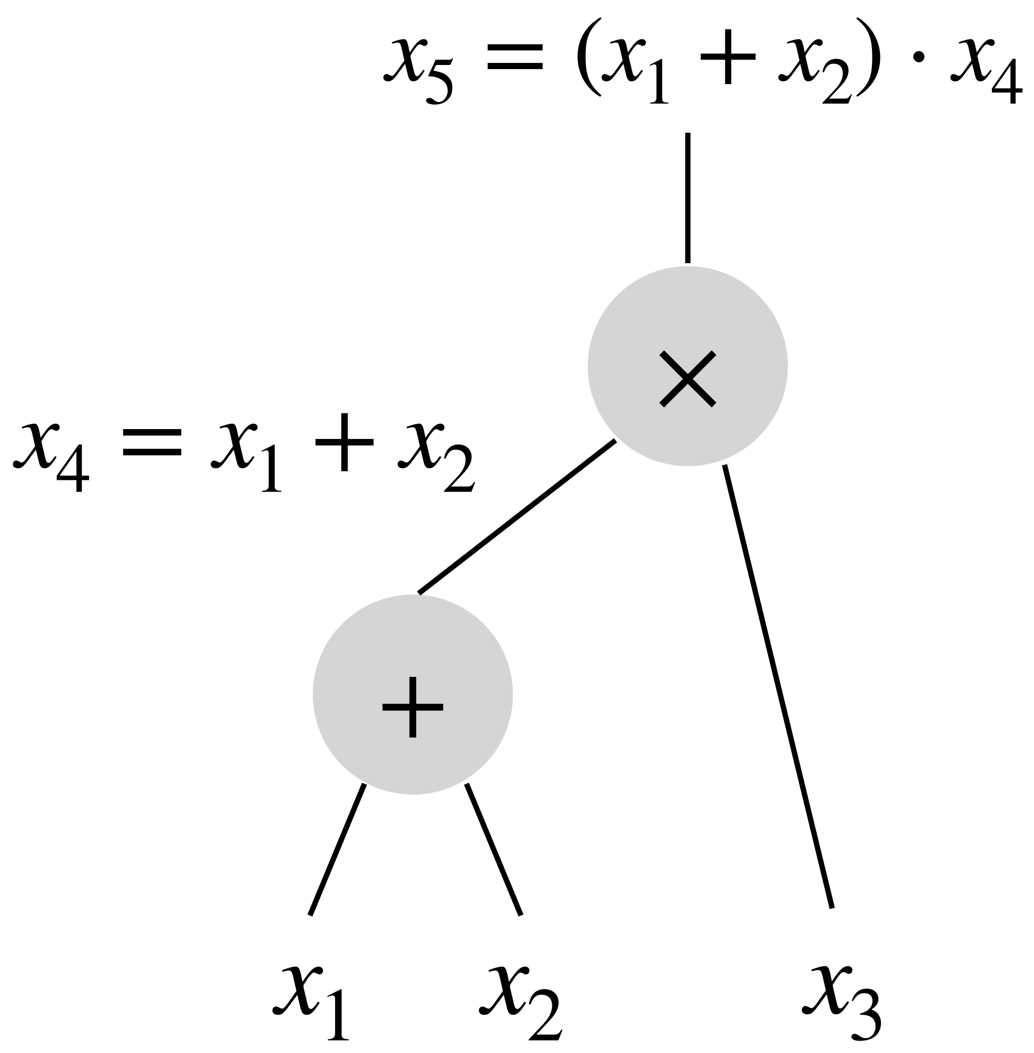


Invariant

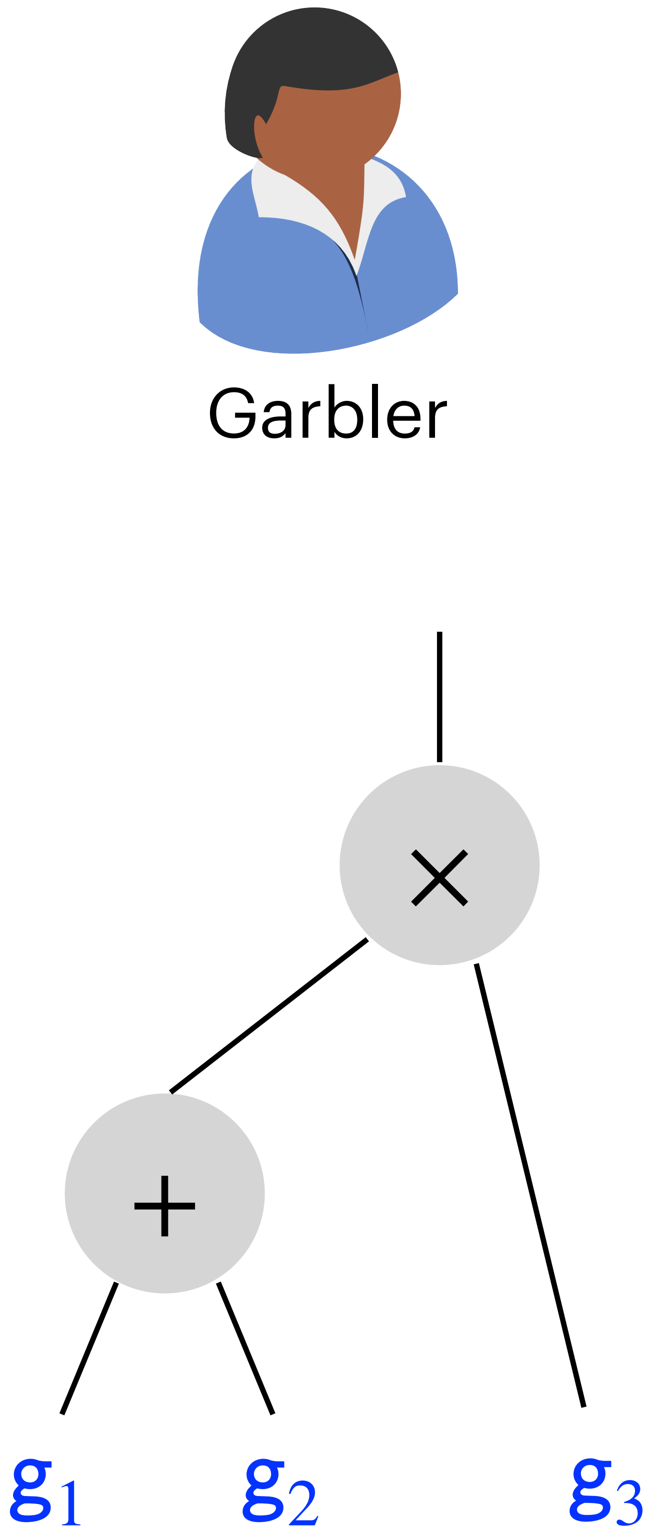
$g_i + e_i = x_i$



Template for Garbling Circuits

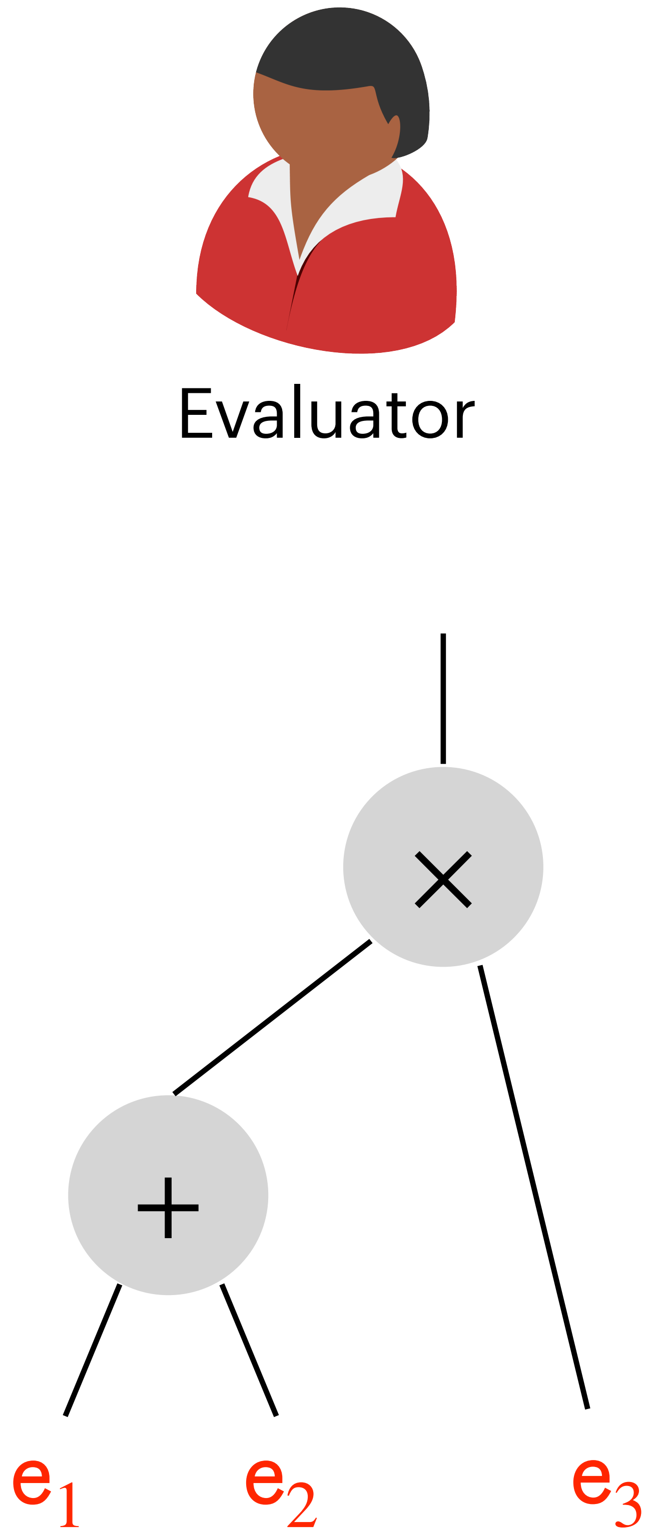


Arithmetic circuit over $\mathbb{Z}_B$



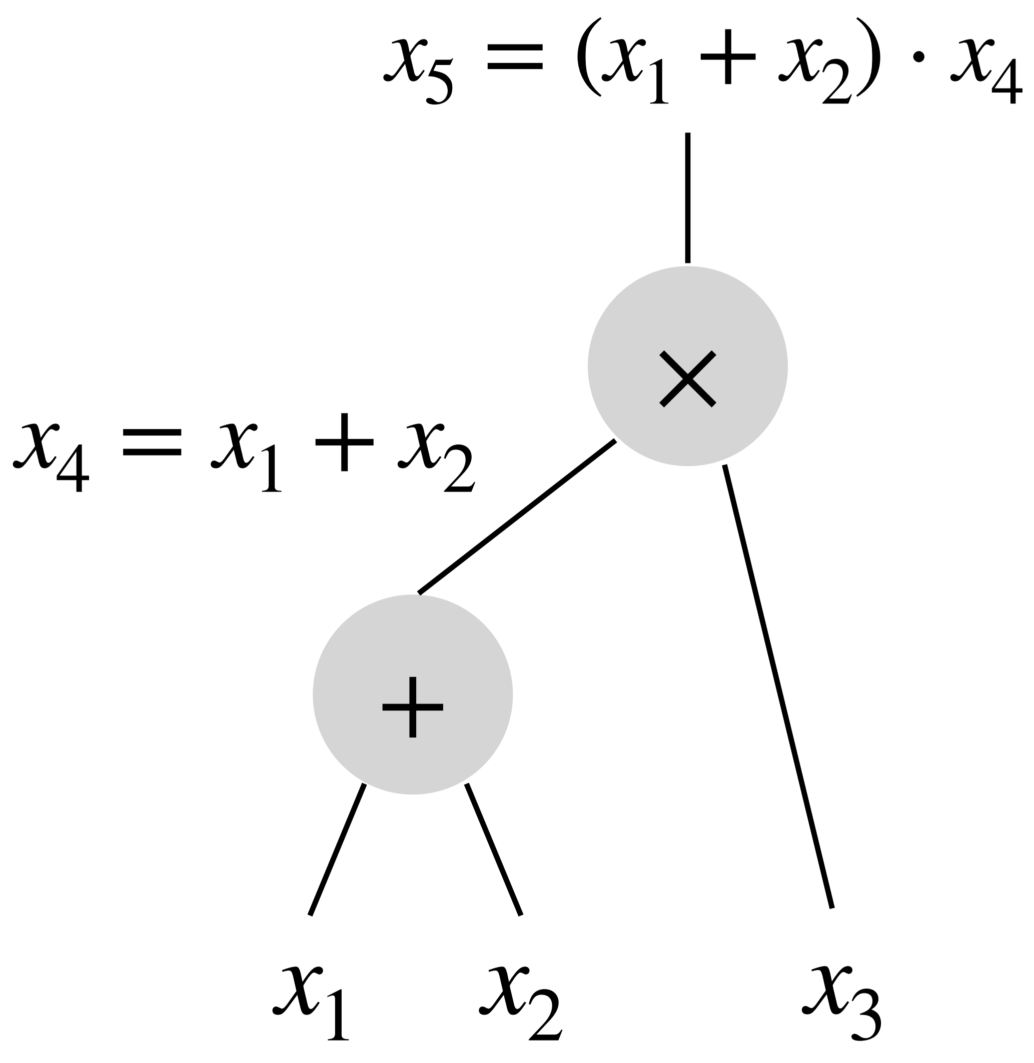
Invariant

$g_i + e_i = x_i$



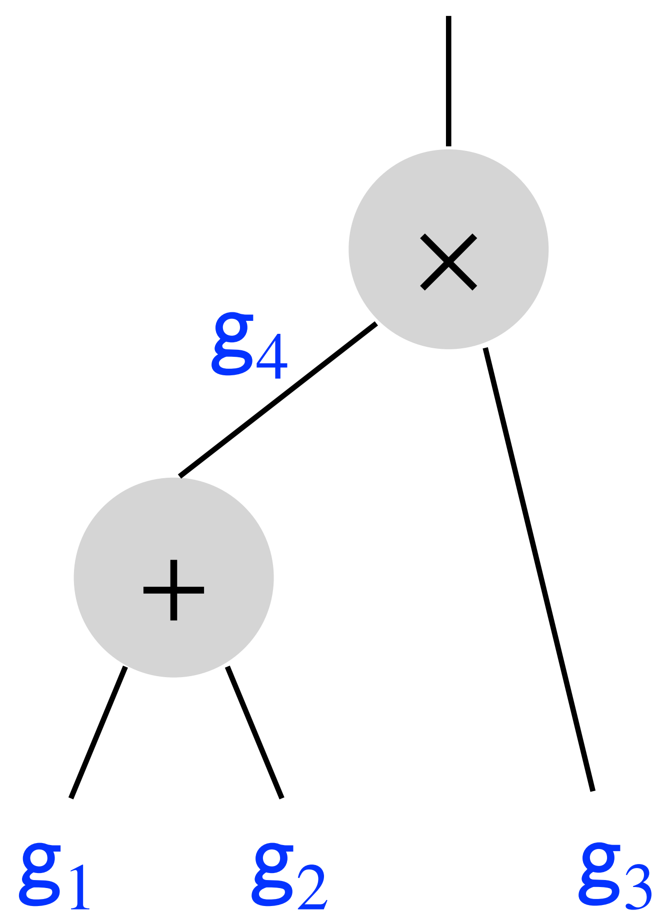
Garbled Circuit $\widehat{C}$

Template for Garbling Circuits



Arithmetic circuit over $\mathbb{Z}_B$

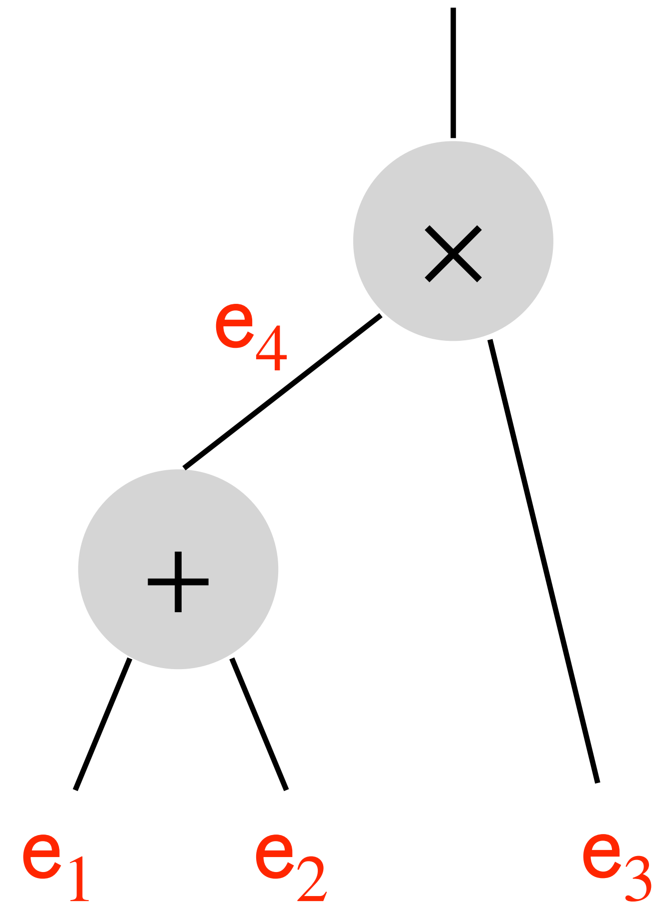
Garbler



Invariant

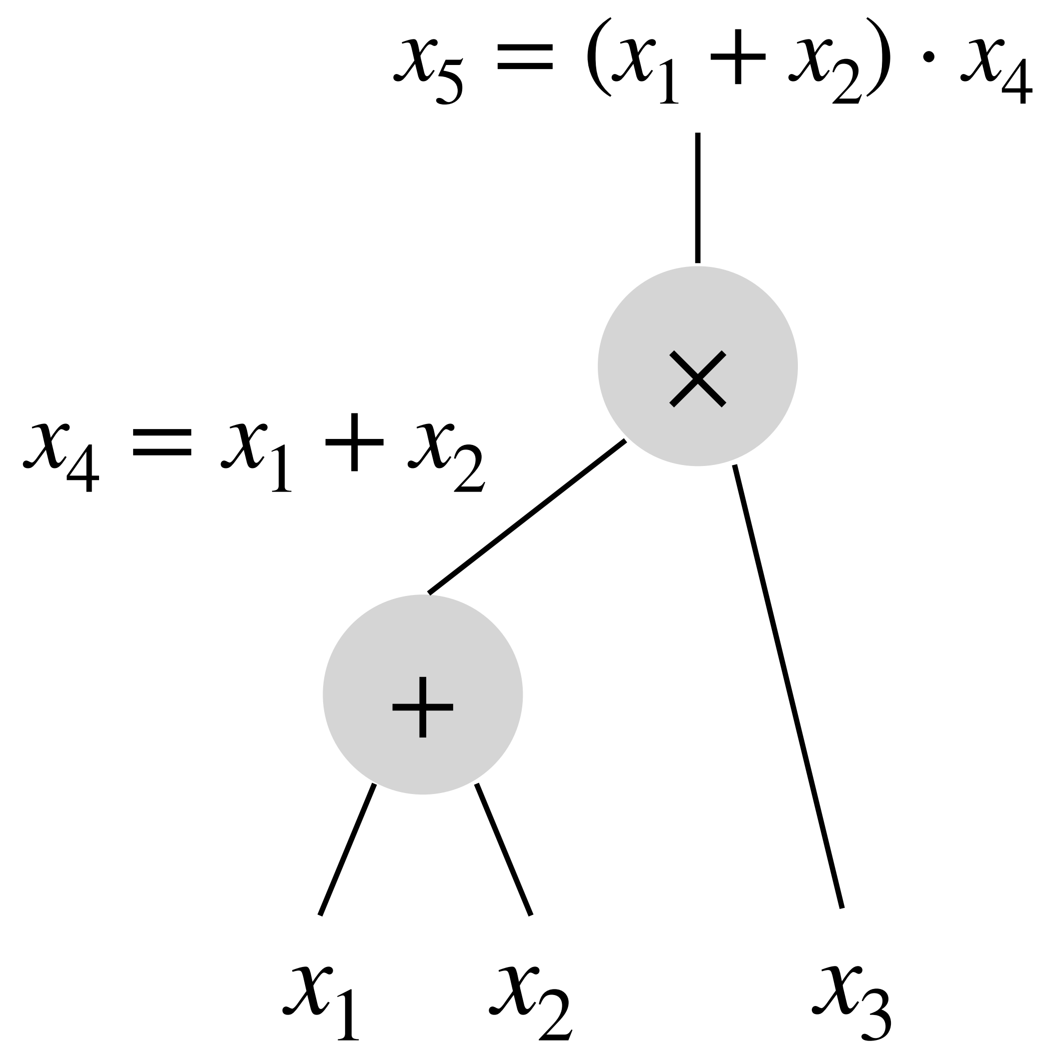
$g_i + e_i = x_i$

Evaluator

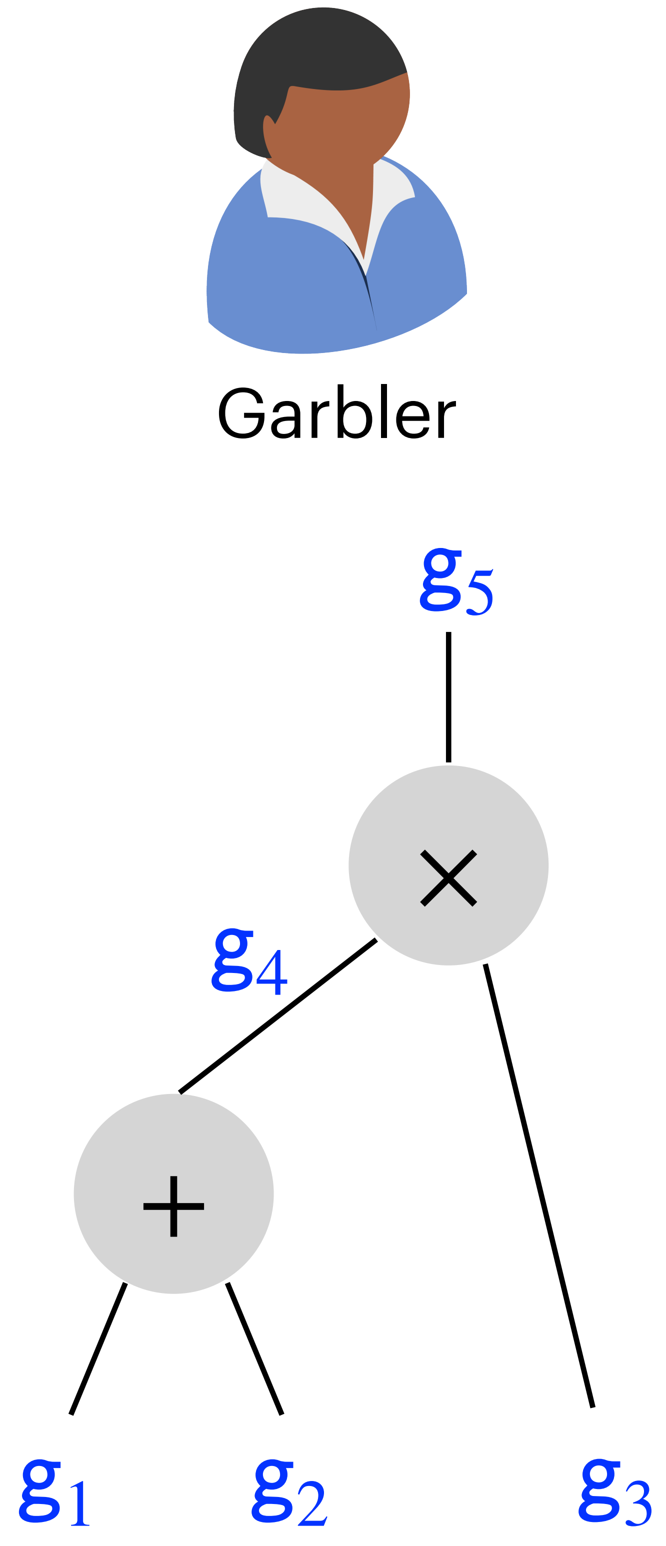


Garbled Circuit $\widehat{C}$

Template for Garbling Circuits

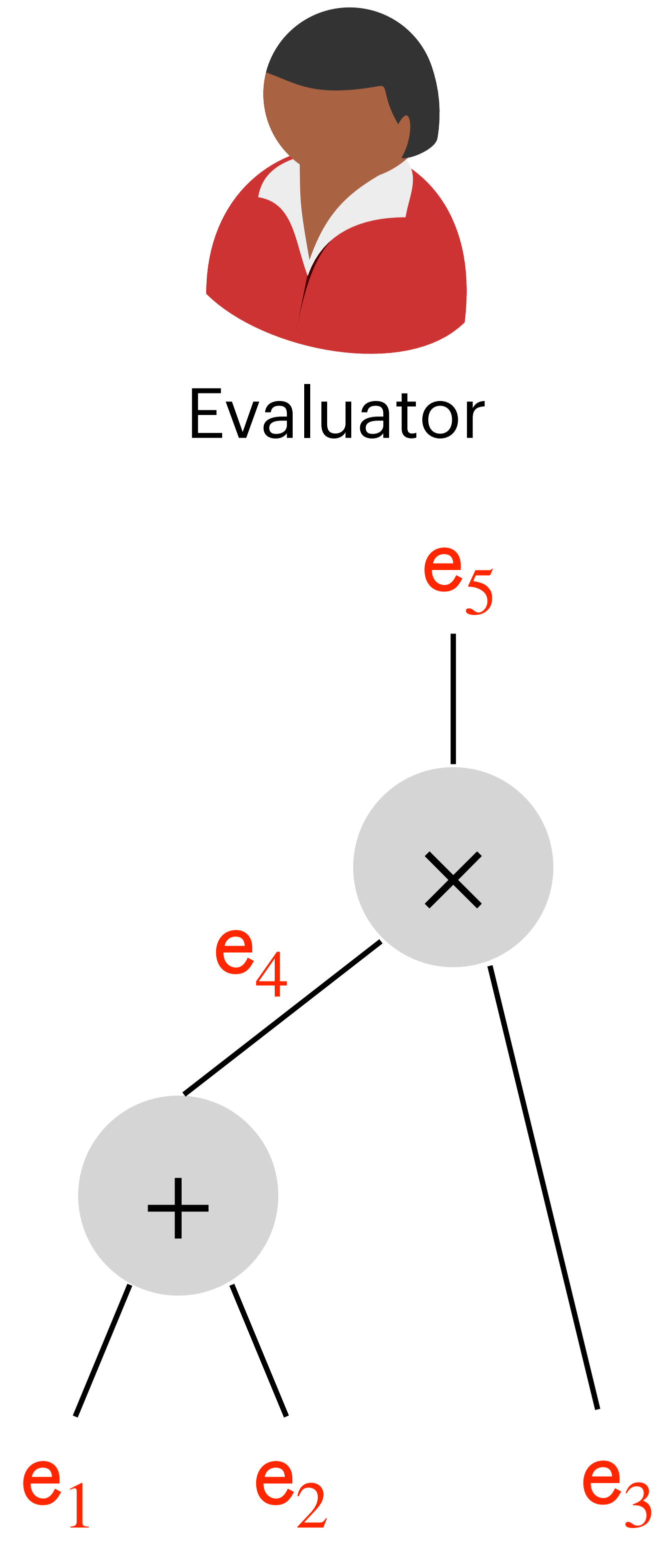


Arithmetic circuit over $\mathbb{Z}_B$



Invariant

$g_i + e_i = x_i$



Garbled Circuit $\widehat{C}$

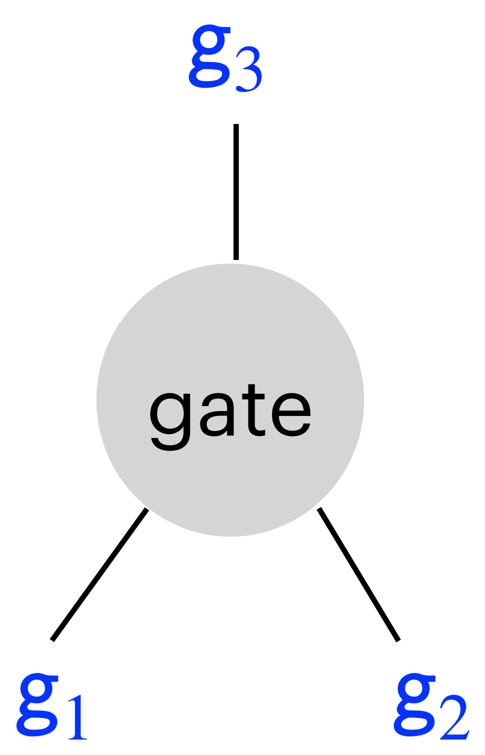
Template for Garbling Circuits

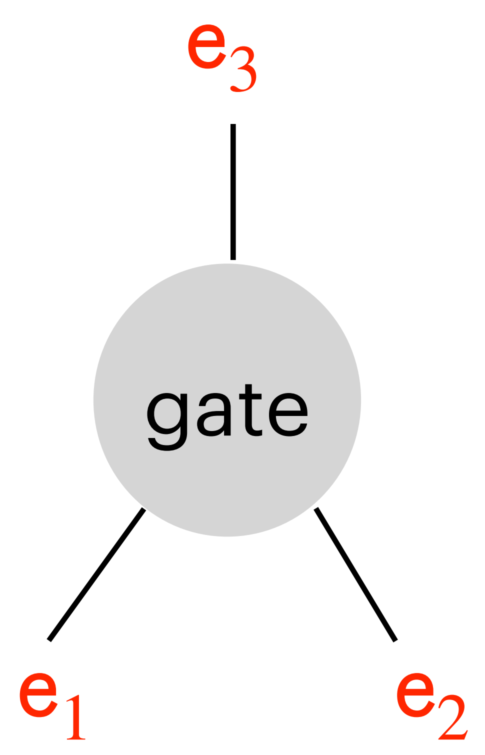
Garbler

Invariant

$$g_i + e_i = x_i$$

Evaluator





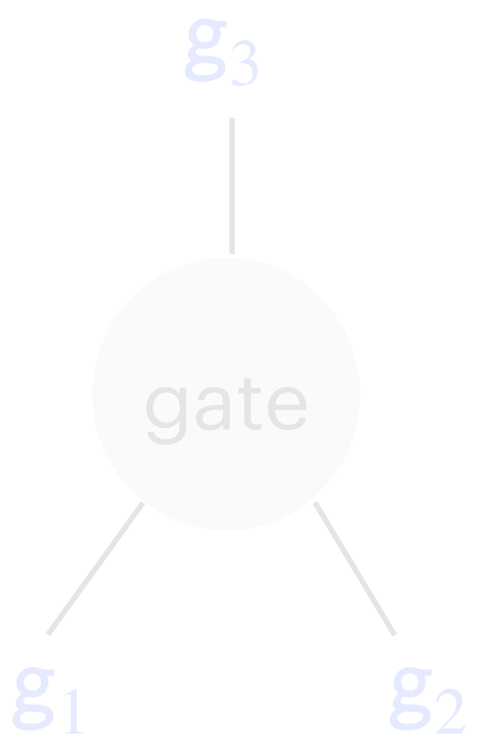
Template for Garbling Circuits

Garbler

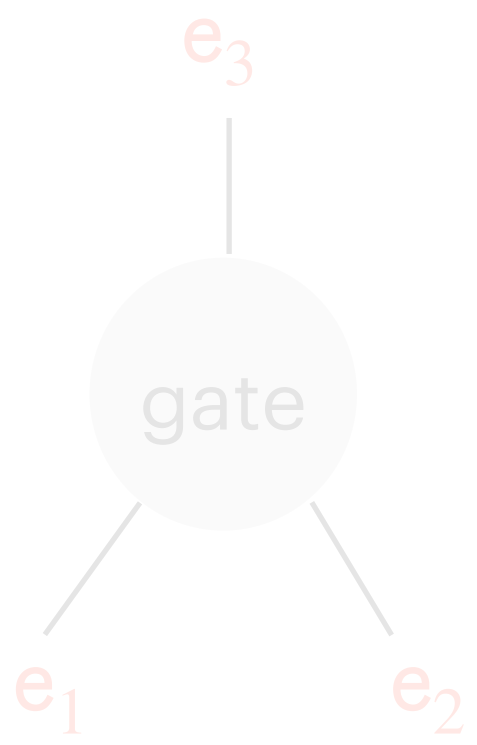
Invariant

$$g_i + e_i = x_i$$

Evaluator



$$z_G + z_E = xy$$



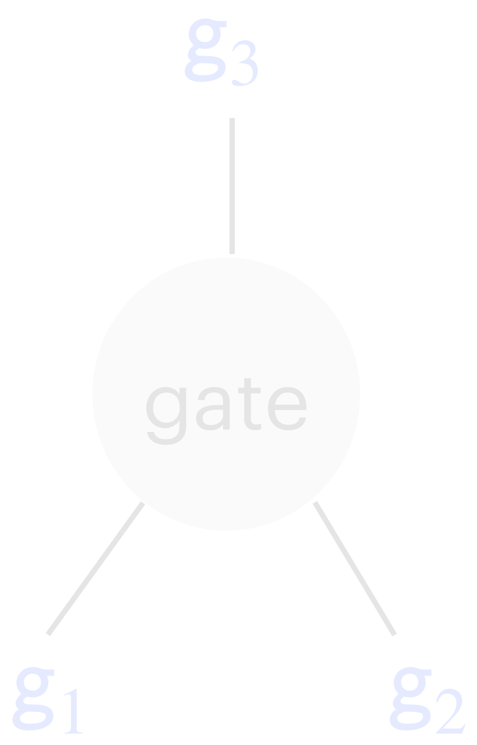
Template for Garbling Circuits

Garbler

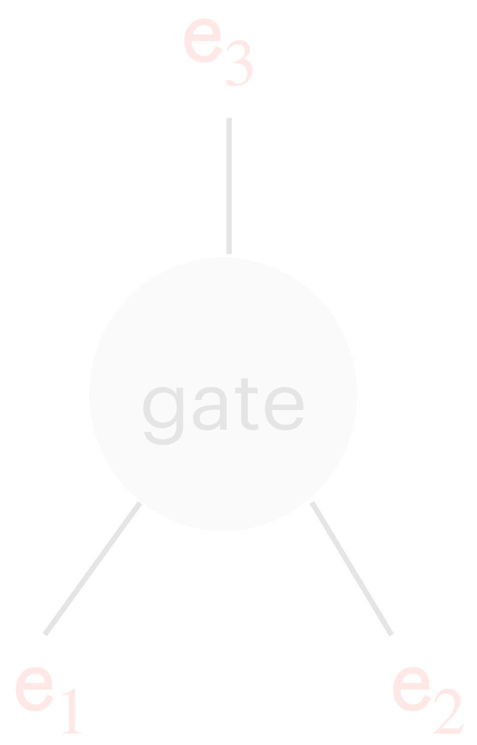
Invariant

$g_i + e_i = x_i$

Evaluator



$$z_G + z_E = xy$$



Instantiation:

Included as part of the garbled circuit

Template for Garbling Circuits

Known to the Garbler

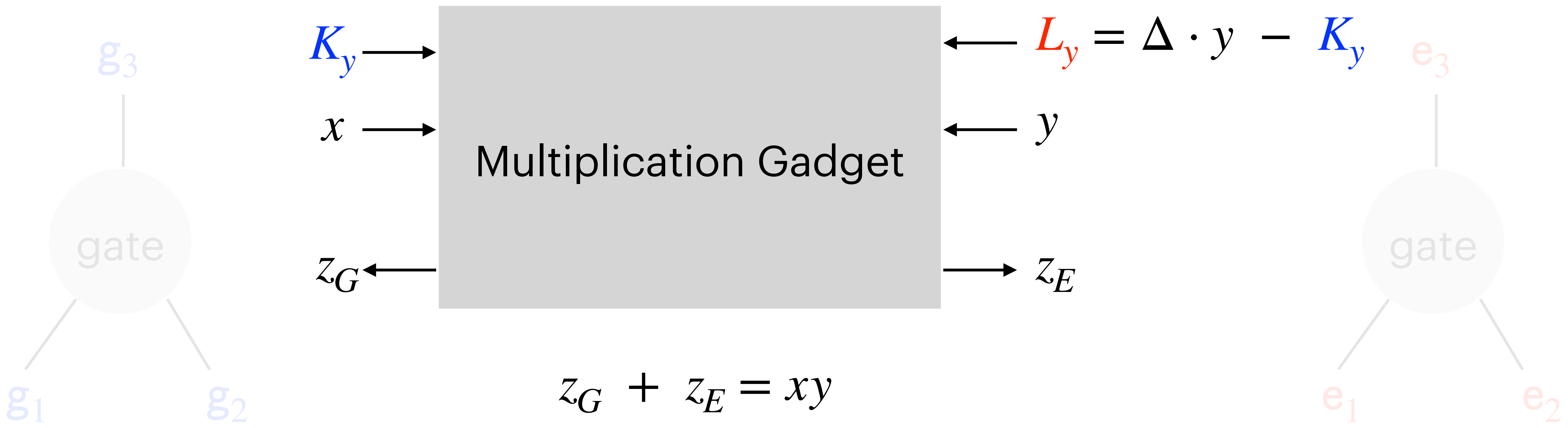
Δ

Garbler

Invariant

$g_i + e_i = x_i$

Evaluator



$$z_G + z_E = xy$$

Included as part of the garbled circuit

Instantiation:

Template for Garbling Circuits

Known to the Garbler

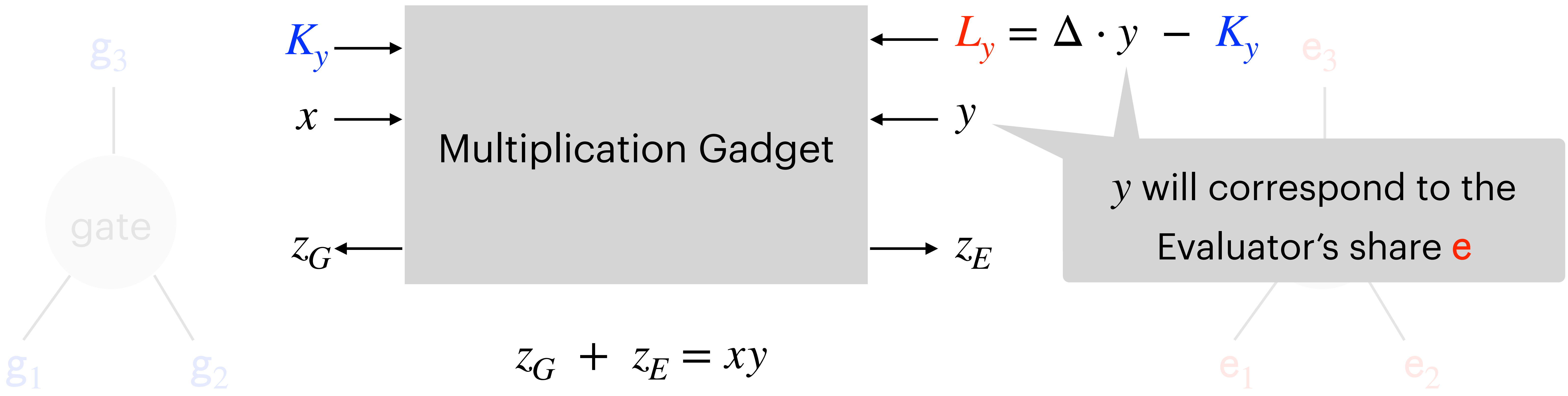
Δ

Garbler

Invariant

$g_i + e_i = x_i$

Evaluator



$L_y = \Delta \cdot y - K_y$

y will correspond to the Evaluator's share e

$$z_G + z_E = xy$$

Instantiation:

Included as part of the garbled circuit

Template for Garbling Circuits

Δ

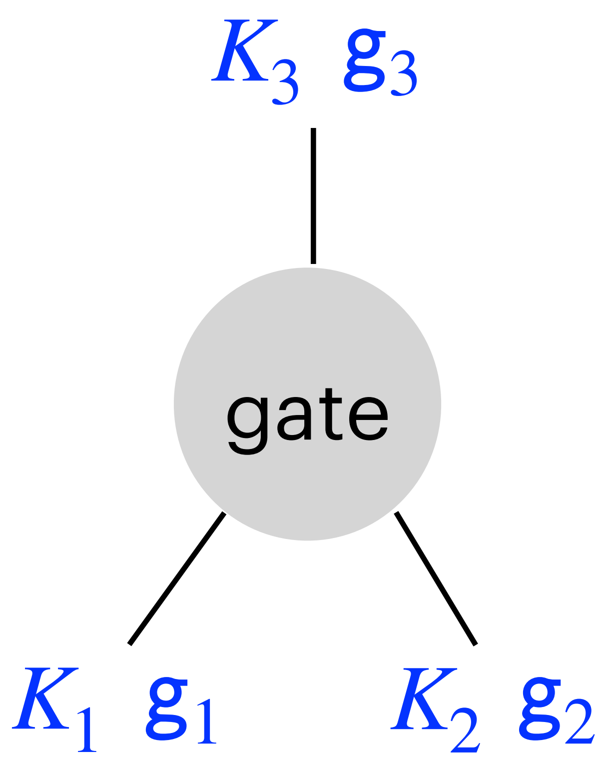
Garbler

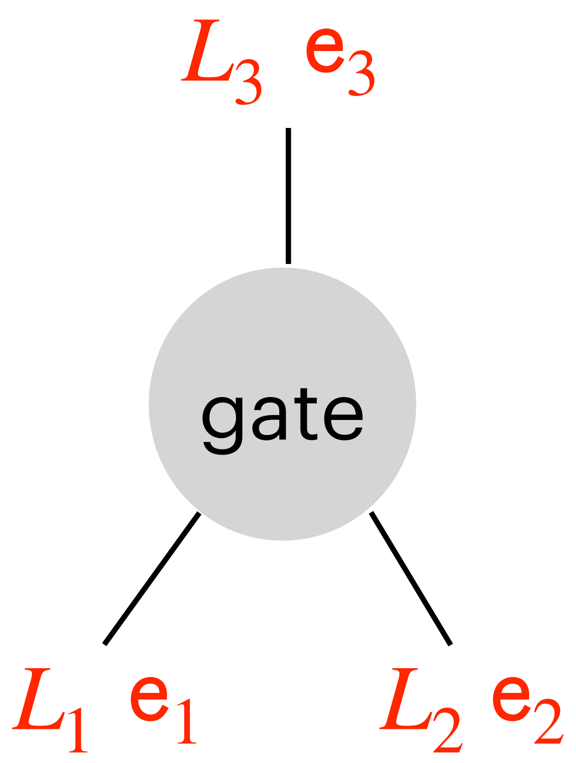
Invariant

$$\mathsf{g}_i + \mathsf{e}_i = x_i$$

$$K_i + L_i = \Delta \cdot \mathsf{e}_i$$

Evaluator





Towards Modular Arithmetic Garbling

Δ

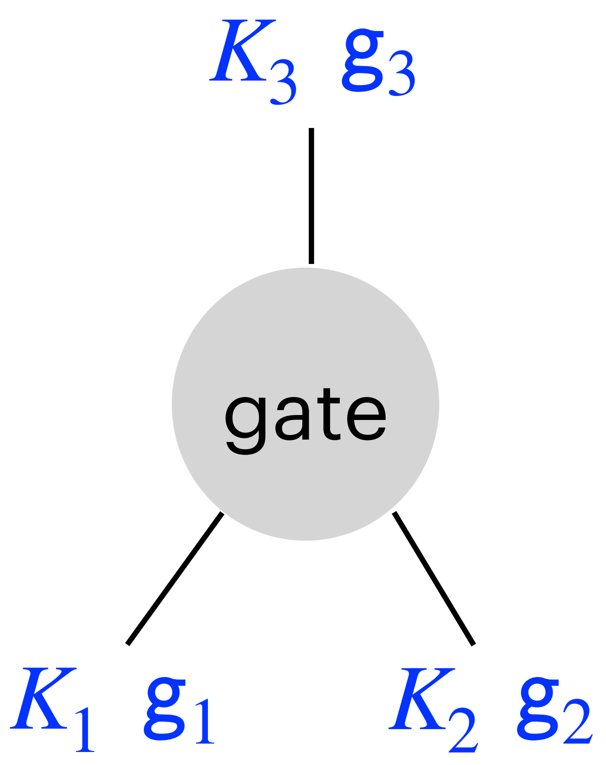
Garbler

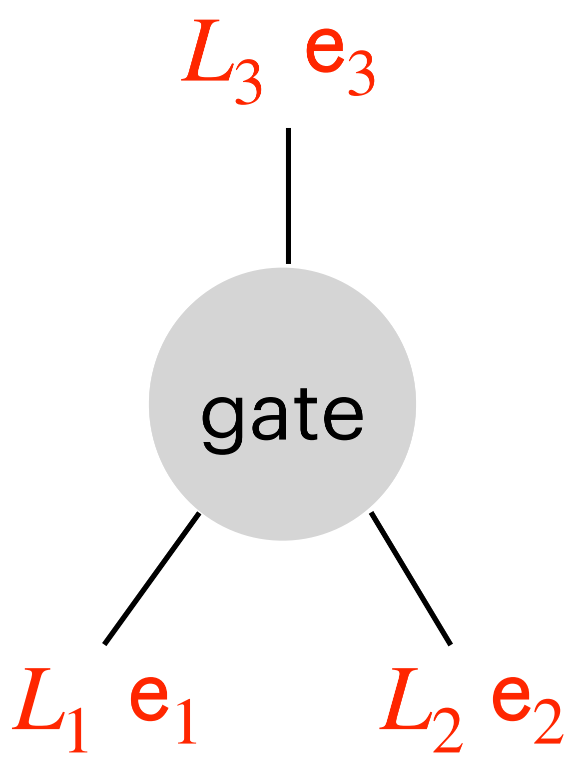
Invariant

$\mathsf{g}_i + \mathsf{e}_i = x_i$

$K_i + L_i = \Delta \cdot \mathsf{e}_i$

Evaluator





Towards Modular Arithmetic Garbling

Δ

Garbler

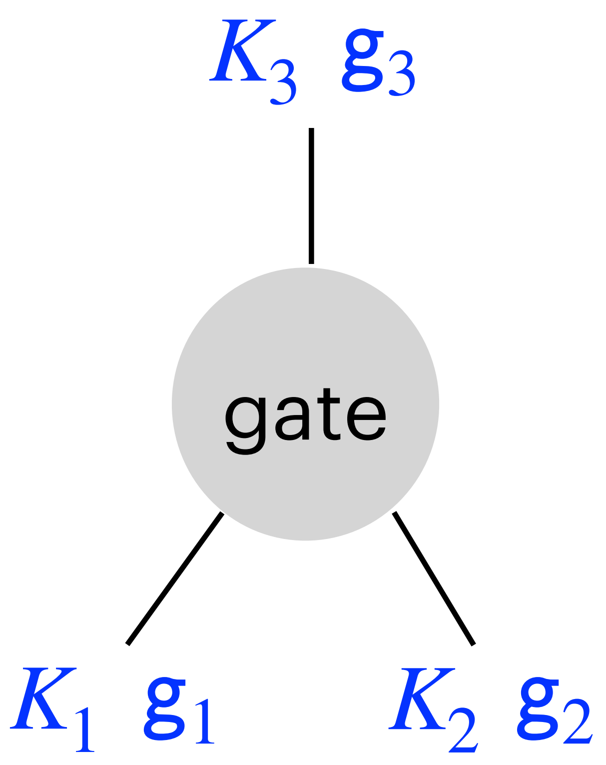
Invariant

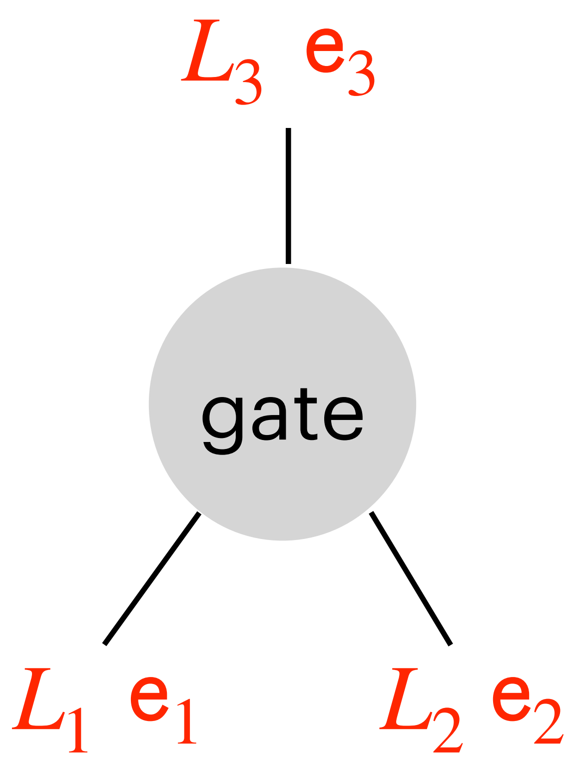
Circuit is computed over $\mathbb{Z}_B$

$$\mathsf{g}_i + \mathsf{e}_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot \mathsf{e}_i$$

Evaluator

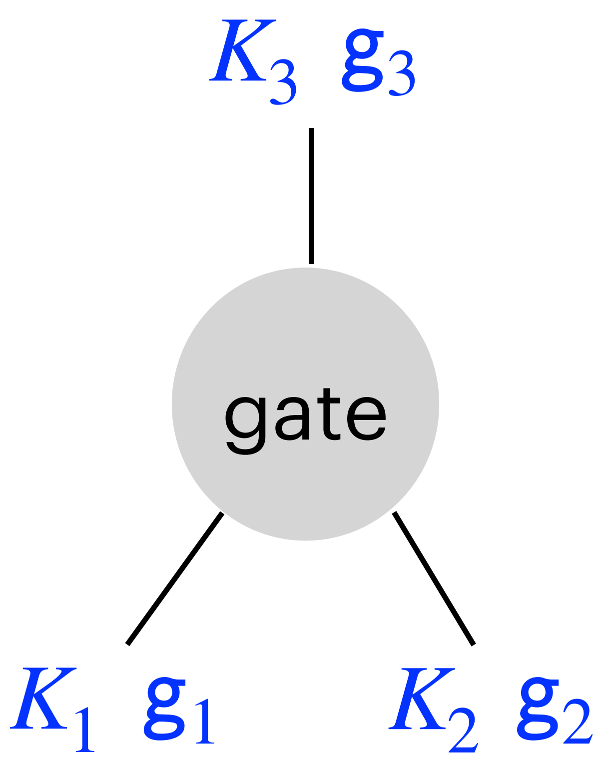




Towards Modular Arithmetic Garbling

Δ

Garbler



Invariant

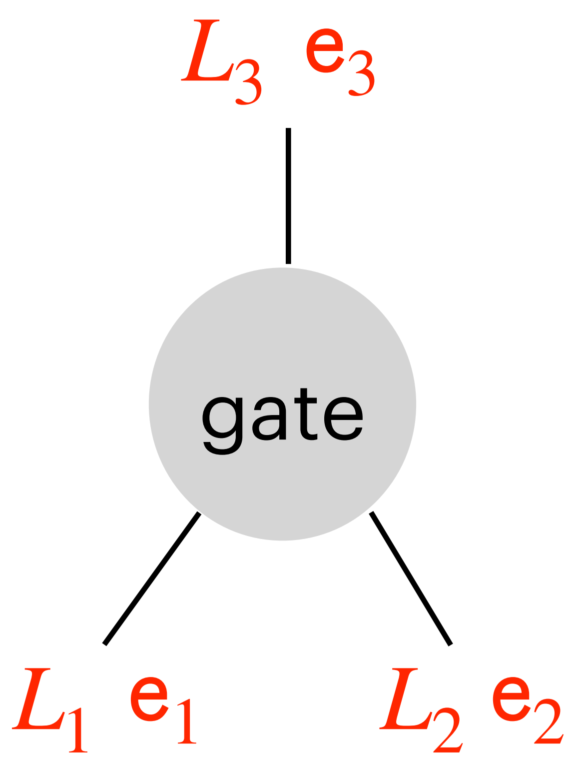
Circuit is computed over $\mathbb{Z}_B$

$$g_i + e_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot e_i \pmod{p}$$

Dictated by multiplication gadget

Evaluator



Towards Modular Arithmetic Garbling

Δ

Garbler

Invariant

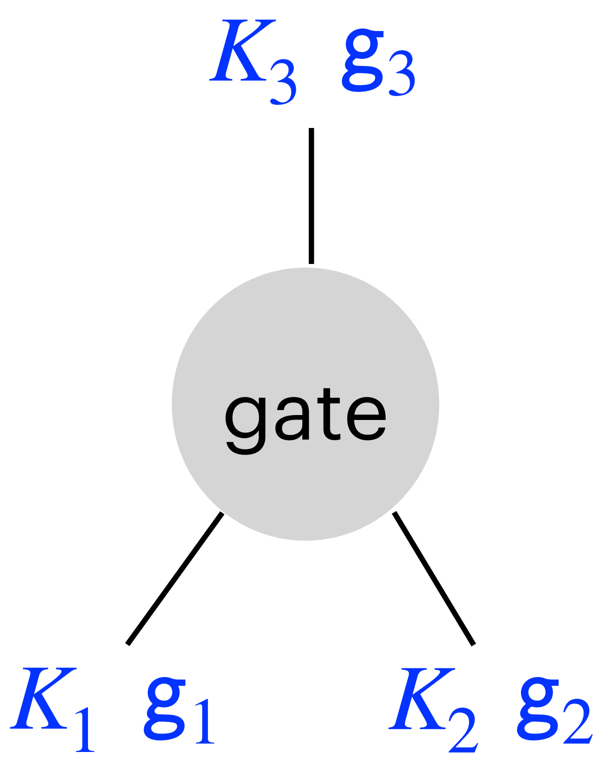
Circuit is computed over $\mathbb{Z}_B$

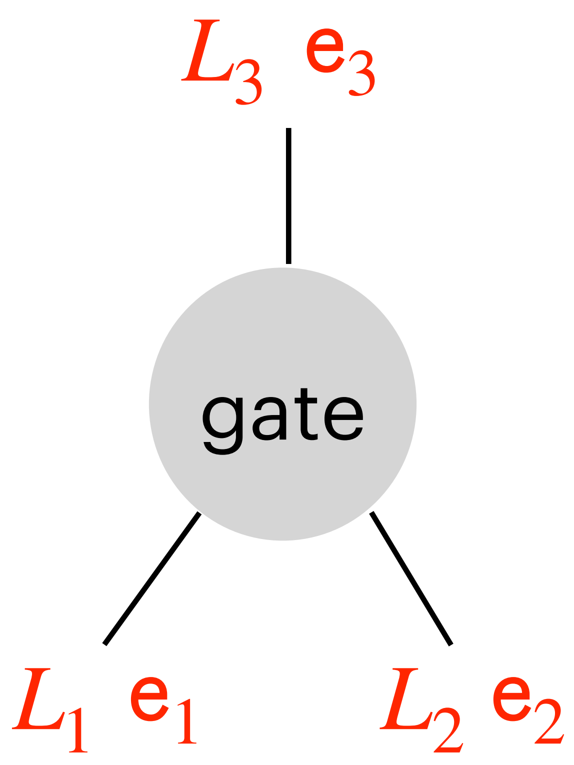
$$\mathsf{g}_i + \mathsf{e}_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot \mathsf{e}_i \pmod{p}$$

Dictated by multiplication gadget

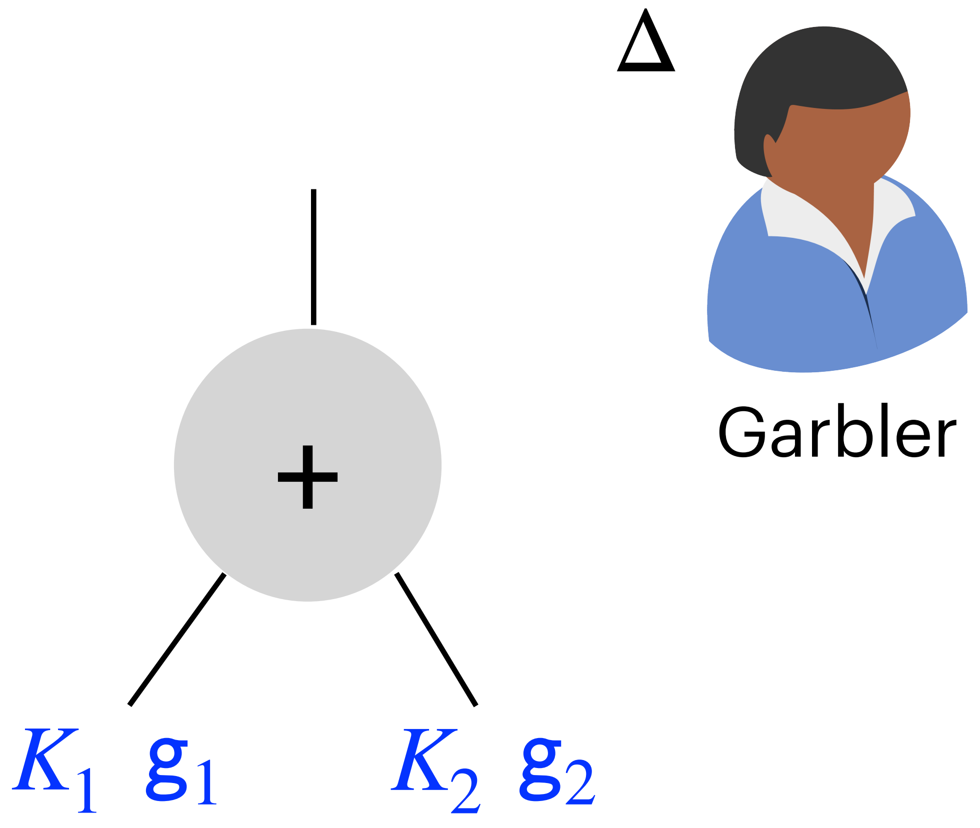
Evaluator





Modulus mismatch makes it non-trivial to maintain invariant

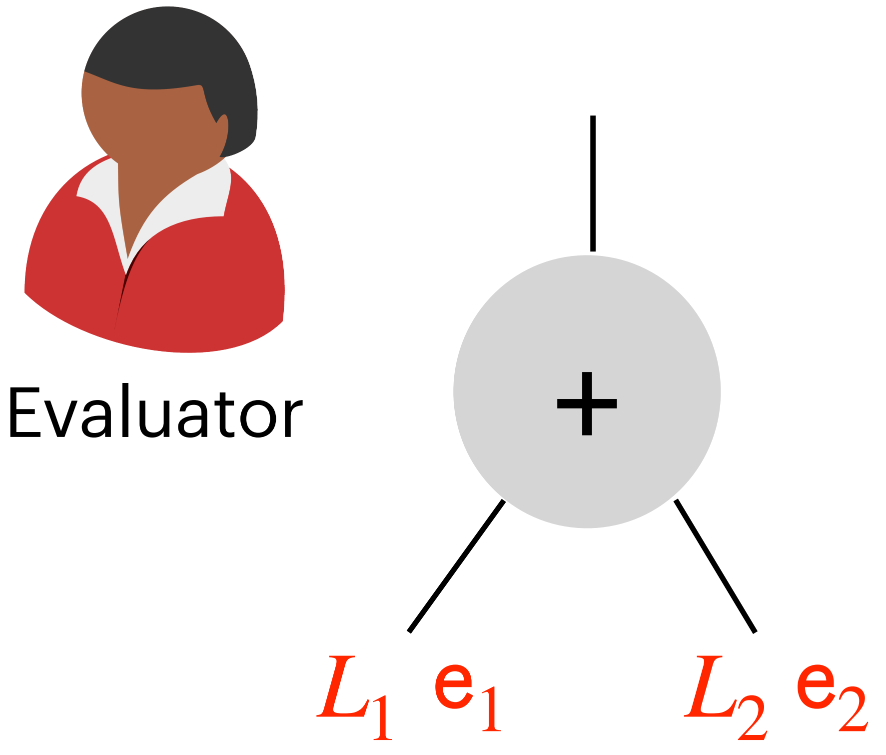
Towards Modular Arithmetic Garbling



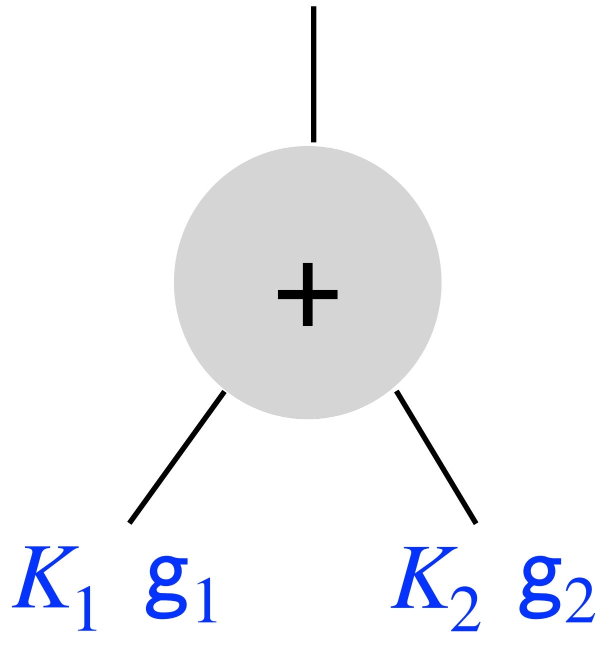
Invariant

$$\mathsf{g}_i + \mathsf{e}_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot \mathsf{e}_i \pmod{p}$$



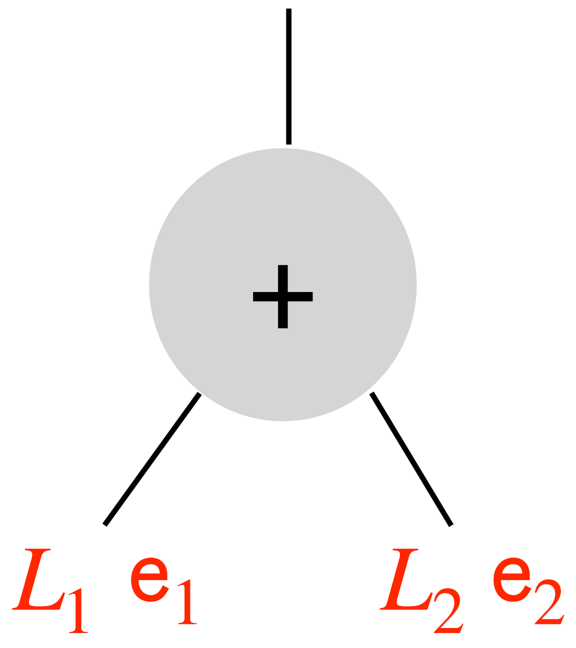
Towards Modular Arithmetic Garbling



Invariant

$$\mathsf{g}_i + \mathsf{e}_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot \mathsf{e}_i \pmod{p}$$



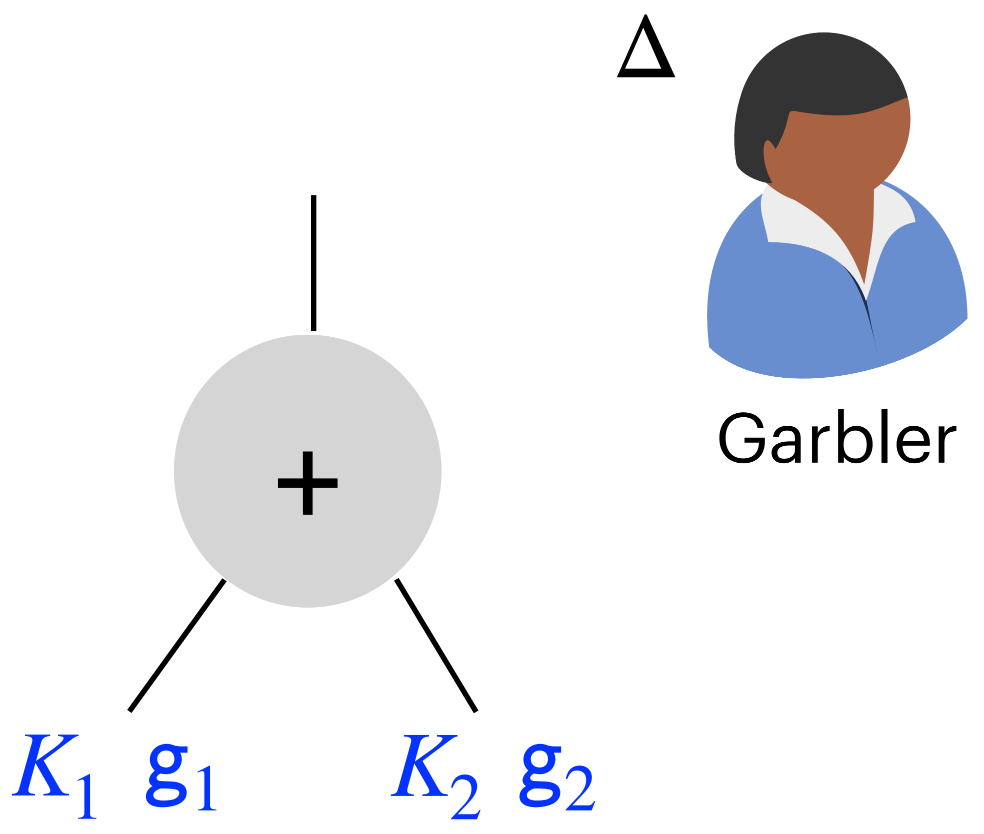
$$\mathsf{g}_3 = \mathsf{g}_1 + \mathsf{g}_2 \bmod B$$

$$K_3 = K_1 + K_2 \bmod p$$

$$\mathsf{e}_3 = \mathsf{e}_1 + \mathsf{e}_2 \bmod B$$

$$L_3 = L_1 + L_2 \bmod p$$

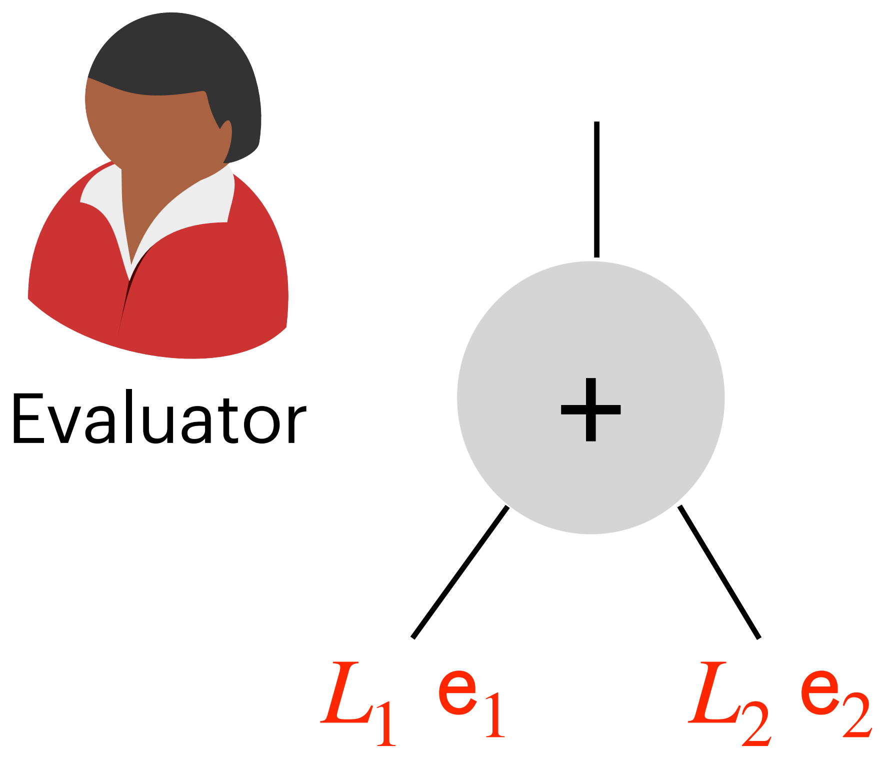
Towards Modular Arithmetic Garbling



Invariant

$$g_i + e_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot e_i \pmod{p}$$



$$g_3 = g_1 + g_2 \bmod B$$

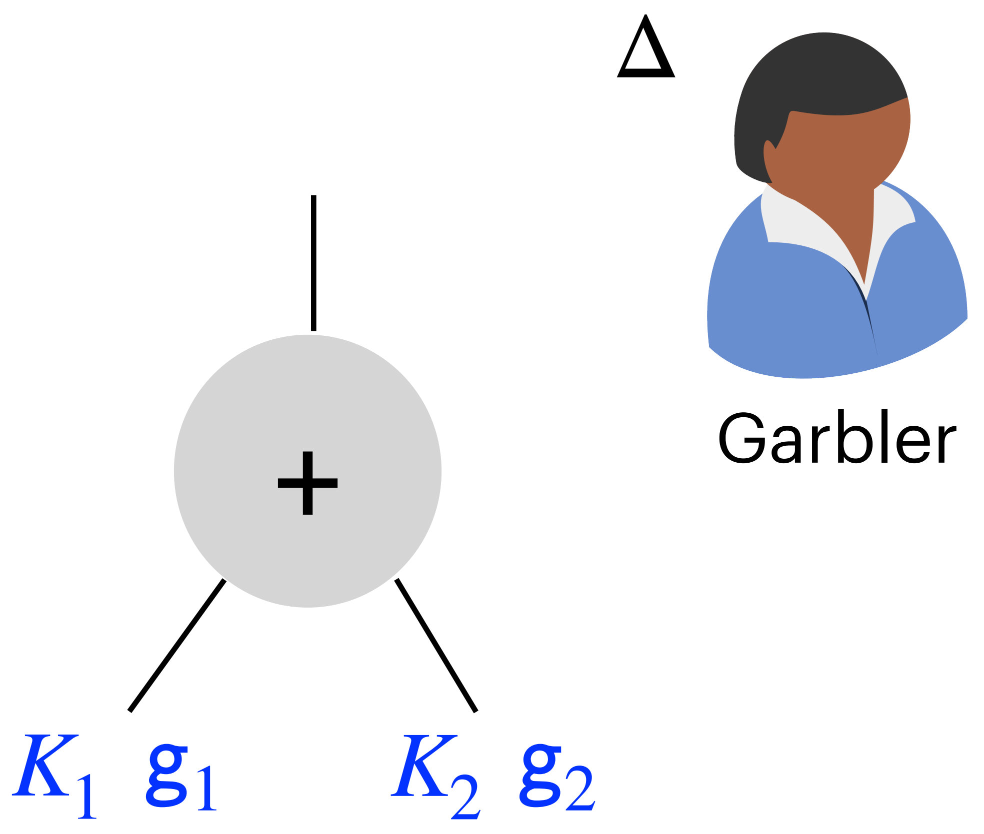
$$K_3 = K_1 + K_2 \bmod p$$

$$e_3 = e_1 + e_2 \bmod B$$

$$L_3 = L_1 + L_2 \bmod p$$

$$g_3 + e_3 = x_1 + x_2 \pmod{B}$$

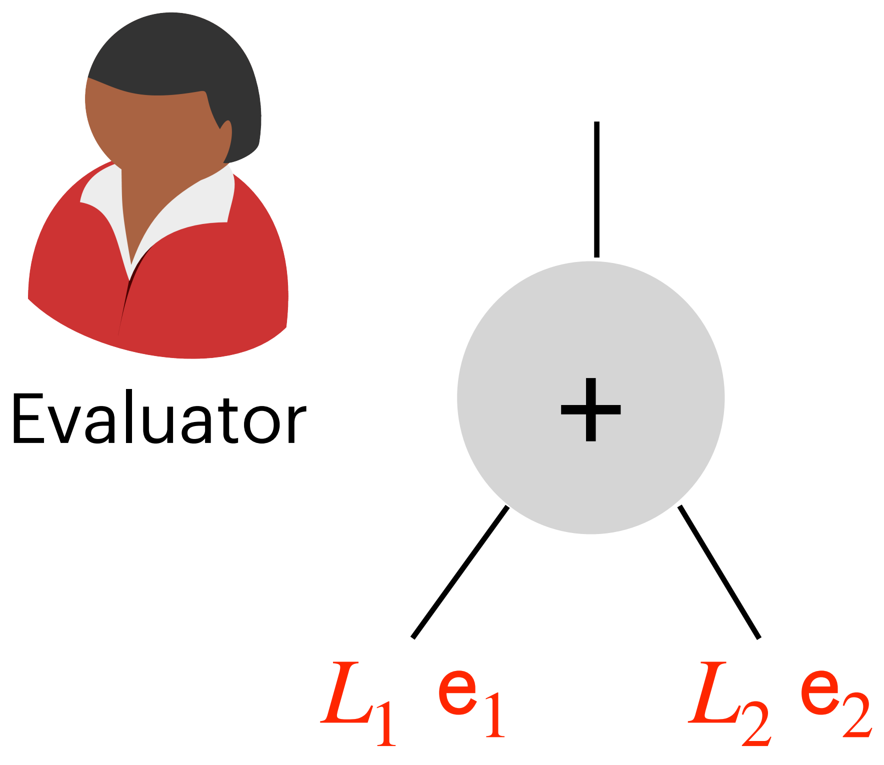
Towards Modular Arithmetic Garbling



Invariant

$$\mathsf{g}_i + \mathsf{e}_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot \mathsf{e}_i \pmod{p}$$



$$\mathsf{g}_3 = \mathsf{g}_1 + \mathsf{g}_2 \bmod B$$

$$K_3 = K_1 + K_2 \bmod p$$

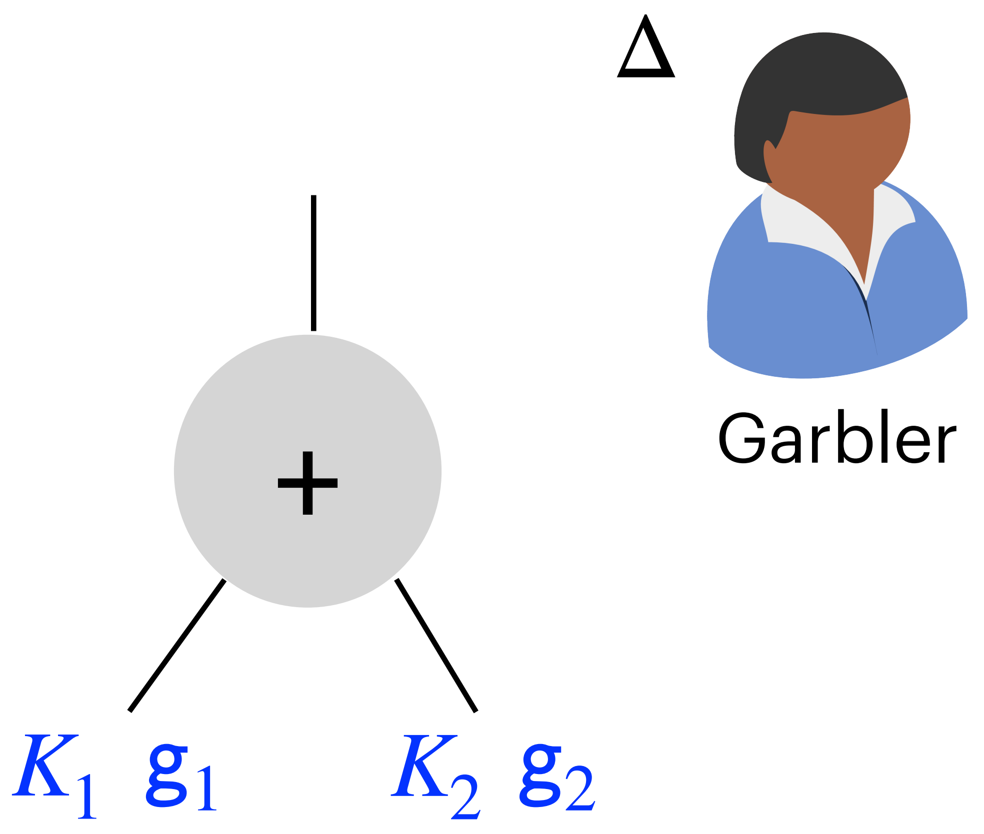
$$\mathsf{e}_3 = \mathsf{e}_1 + \mathsf{e}_2 \bmod B$$

$$L_3 = L_1 + L_2 \bmod p$$

$$\mathsf{g}_3 + \mathsf{e}_3 = x_1 + x_2 \pmod{B}$$

$$K_3 + L_3 \bmod p = \Delta \cdot (\mathsf{e}_1 + \mathsf{e}_2) \bmod p$$

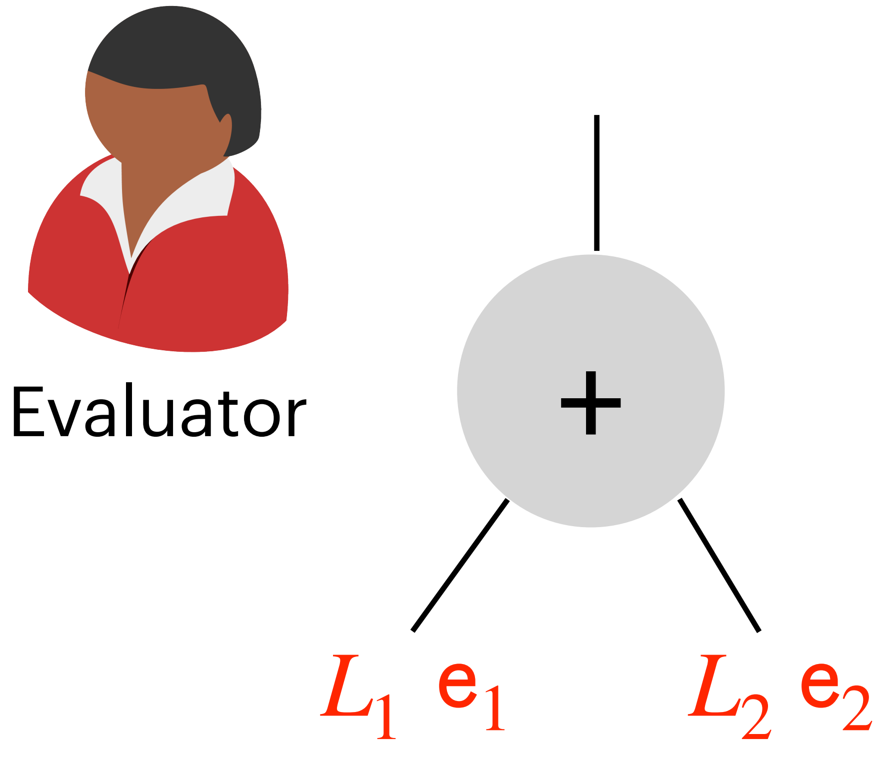
Towards Modular Arithmetic Garbling



Invariant

$$\mathsf{g}_i + \mathsf{e}_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot \mathsf{e}_i \pmod{p}$$



$$\mathsf{g}_3 = \mathsf{g}_1 + \mathsf{g}_2 \bmod B$$

$$K_3 = K_1 + K_2 \bmod p$$

$$\mathsf{e}_3 = \mathsf{e}_1 + \mathsf{e}_2 \bmod B$$

$$L_3 = L_1 + L_2 \bmod p$$

$$\mathsf{g}_3 + \mathsf{e}_3 = x_1 + x_2 \pmod{B}$$

$$K_3 + L_3 \bmod p = \Delta \cdot (\mathsf{e}_1 + \mathsf{e}_2) \bmod p \neq \Delta \cdot (\mathsf{e}_1 + \mathsf{e}_1 \bmod B) \bmod p$$

We need to compute over different moduli

Towards Modular Arithmetic Garbling

Δ

Garbler

Evaluator

Observation: An extension of the multiplication gadget can circumvent modulus mismatch

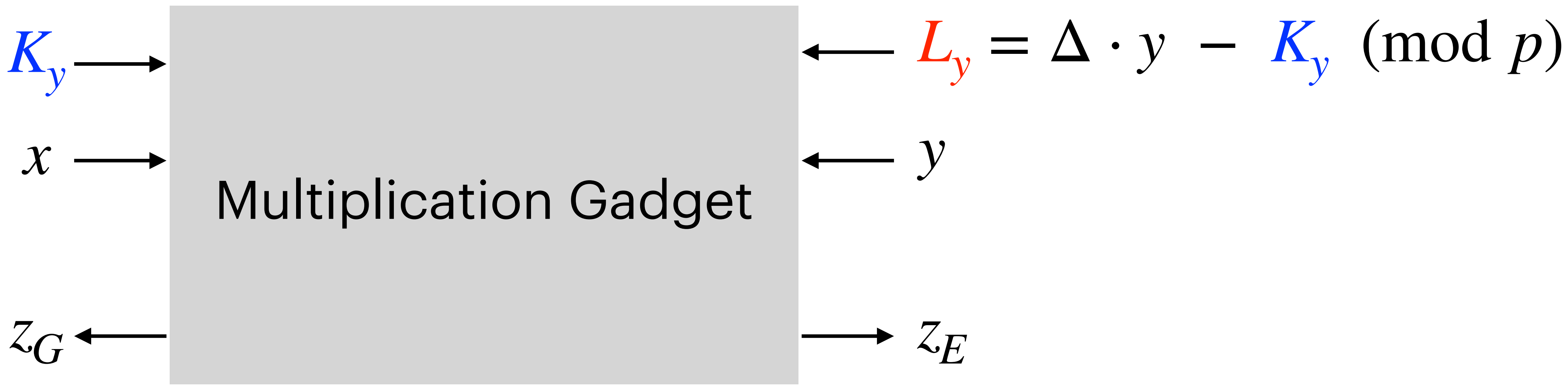
Towards Modular Arithmetic Garbling

Δ

Garbler

Evaluator

Observation: An extension of the multiplication gadget can circumvent modulus mismatch



Multiplication Gadget

$K_y \rightarrow$

$x \rightarrow$

$z_G \leftarrow$

$\leftarrow L_y = \Delta \cdot y - K_y \pmod{p}$

$\leftarrow y$

$\rightarrow z_E$

$$z_G + z_E = xy$$

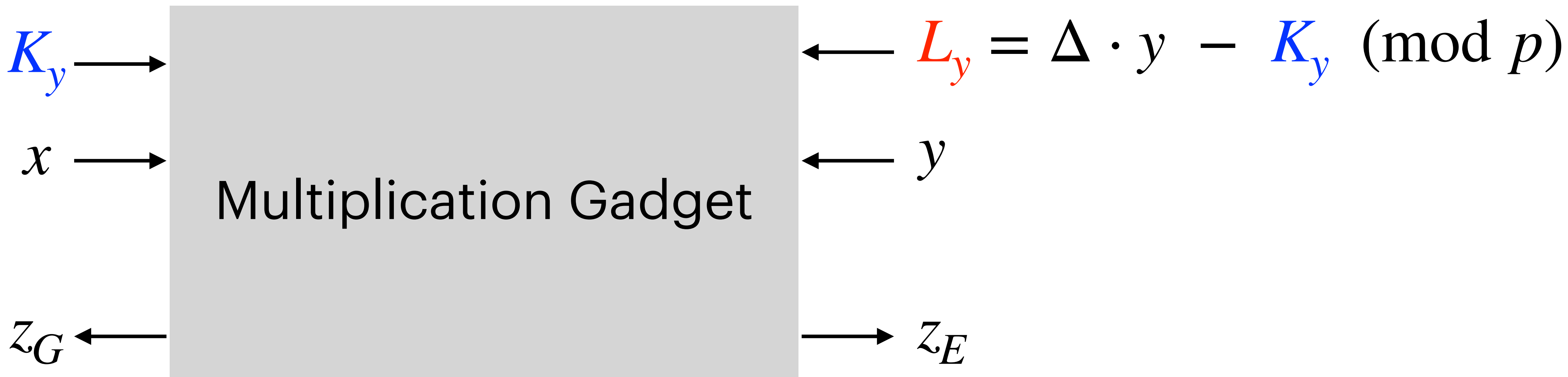
Towards Modular Arithmetic Garbling

Δ

Garbler

Evaluator

Observation: An extension of the multiplication gadget can circumvent modulus mismatch



Multiplication Gadget

K_y

x

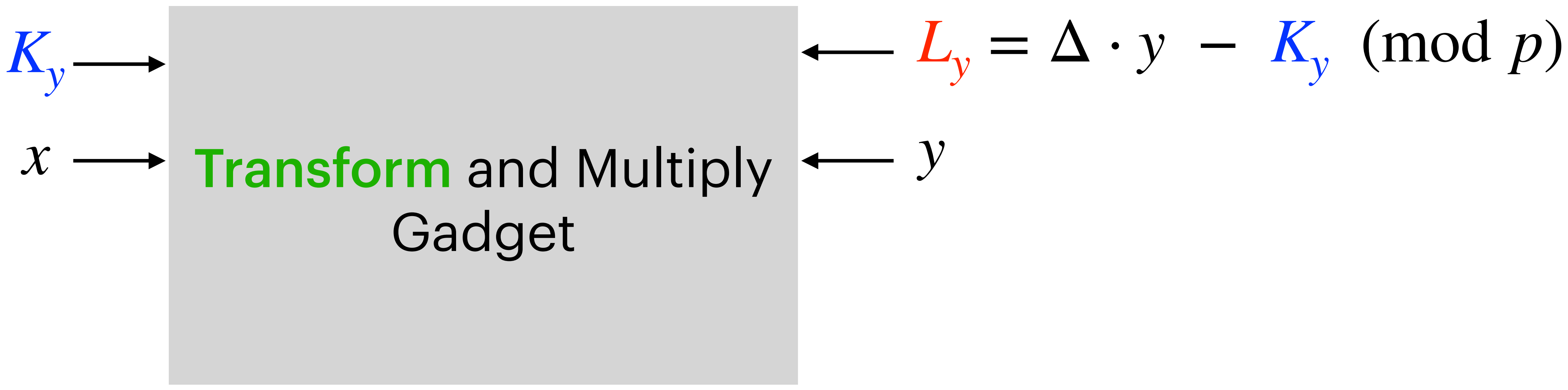
z_G

$L_y = \Delta \cdot y \; - \; K_y \;\; (\text{mod } p)$

y

z_E

$$z_G \; + \; z_E = xy$$



Transform and Multiply Gadget

K_y

x

$L_y = \Delta \cdot y \; - \; K_y \;\; (\text{mod } p)$

y

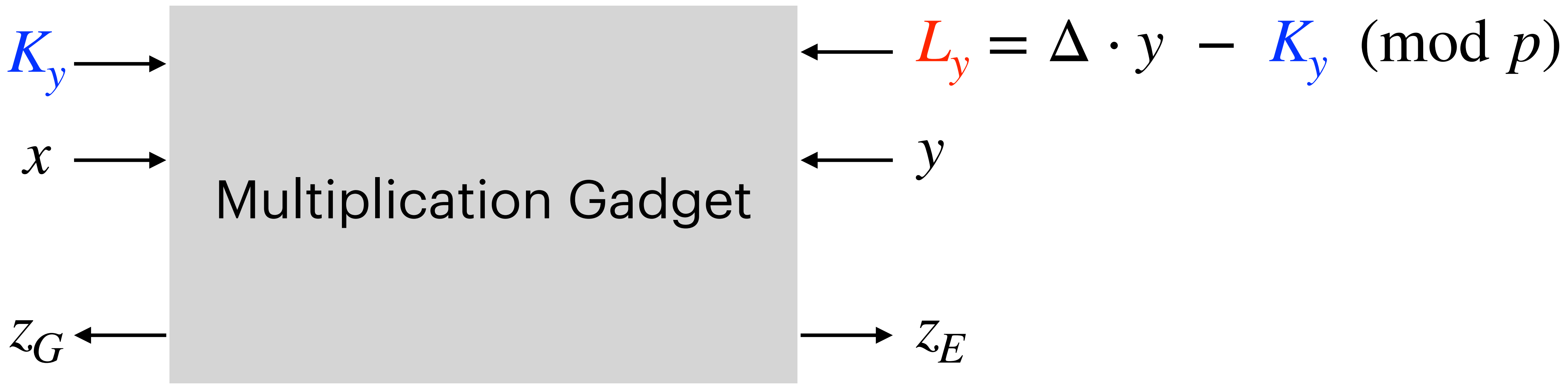
Towards Modular Arithmetic Garbling

Δ

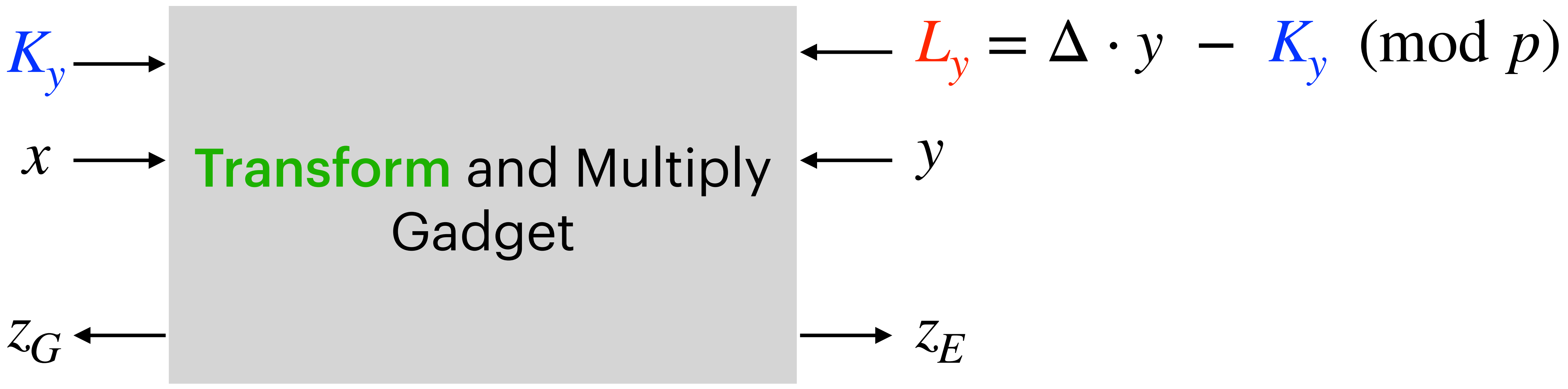
Garbler

Evaluator

Observation: An extension of the multiplication gadget can circumvent modulus mismatch

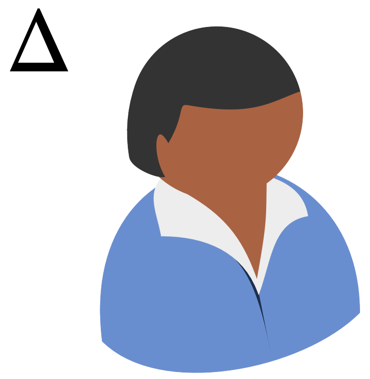


$$z_G + z_E = xy$$



$$z_G + z_E = x \cdot f(y)$$

Towards Modular Arithmetic Garbling

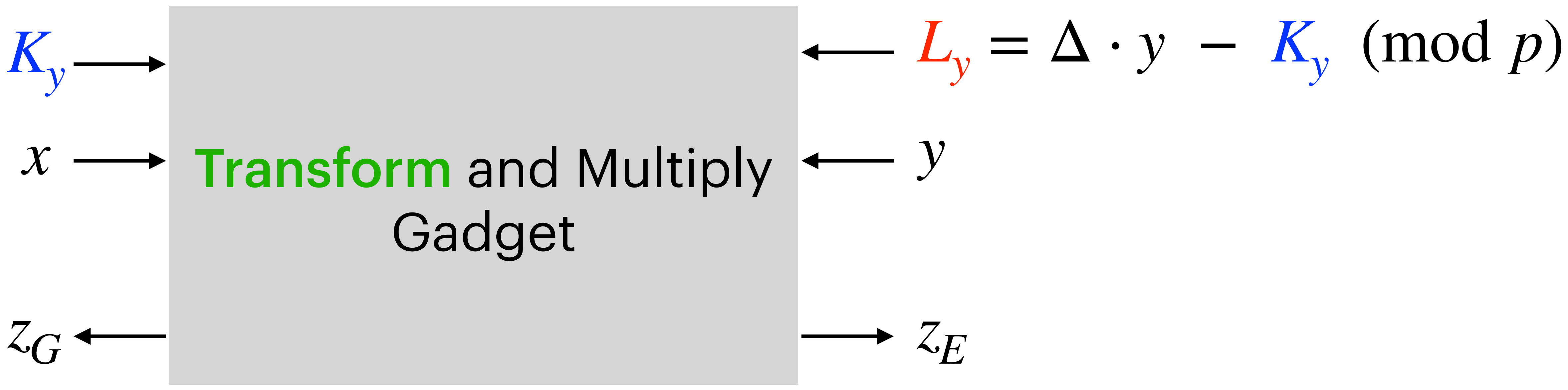


Garbler

Evaluator

Observation: An extension of the multiplication gadget can circumvent modulus mismatch

Transform-and-Multiply Gadget for any function f using the power-DDH based PPRF



K_y

x

z_G

Transform and Multiply Gadget

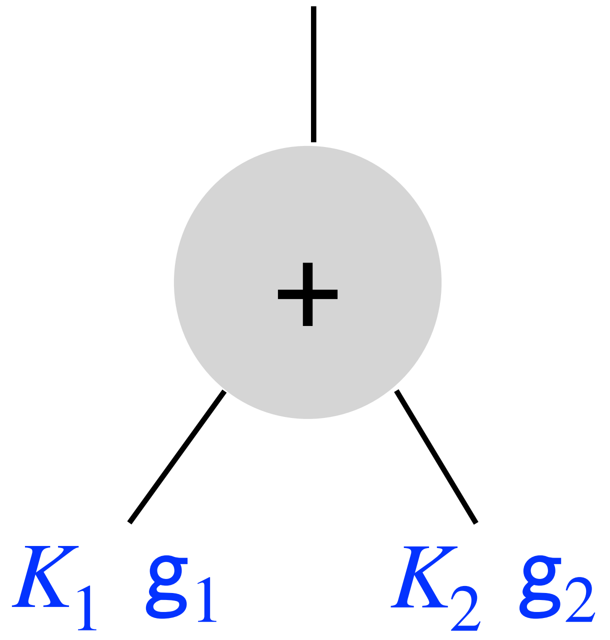
$L_y = \Delta \cdot y - K_y \pmod{p}$

y

z_E

$$z_G + z_E = x \cdot f(y)$$

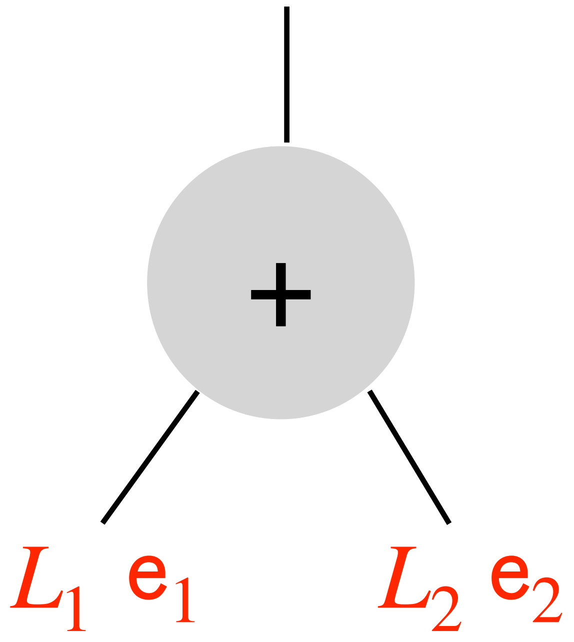
Modular Arithmetic Garbling



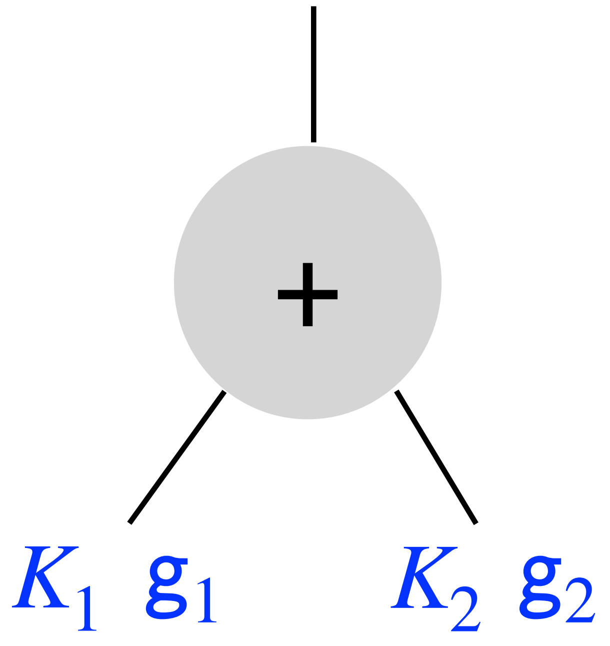
Invariant

$$\mathsf{g}_i + \mathsf{e}_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot \mathsf{e}_i \pmod{p}$$



Modular Arithmetic Garbling



Δ

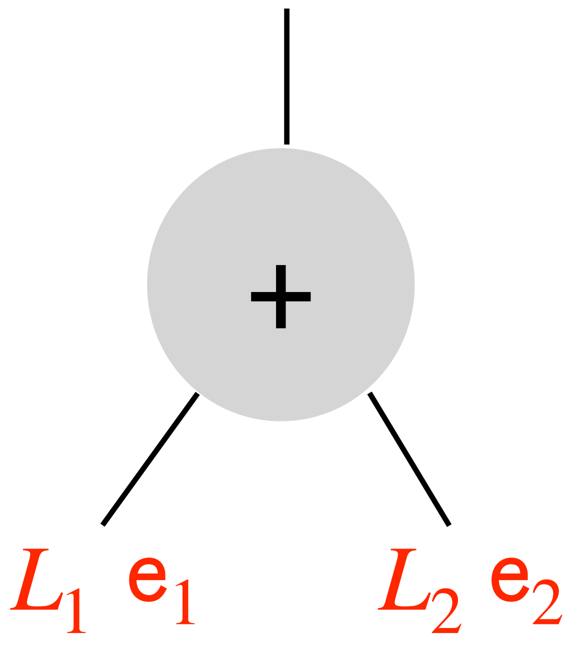
Garbler

Invariant

$$g_i + e_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot e_i \pmod{p}$$

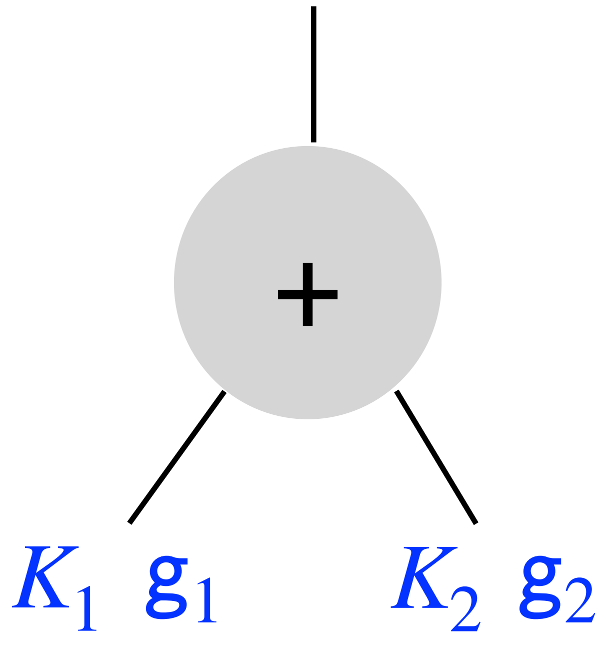
Evaluator

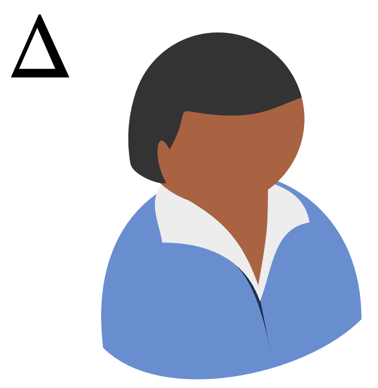


Computed over the integers

$$e^* = e_1 + e_2$$

Modular Arithmetic Garbling





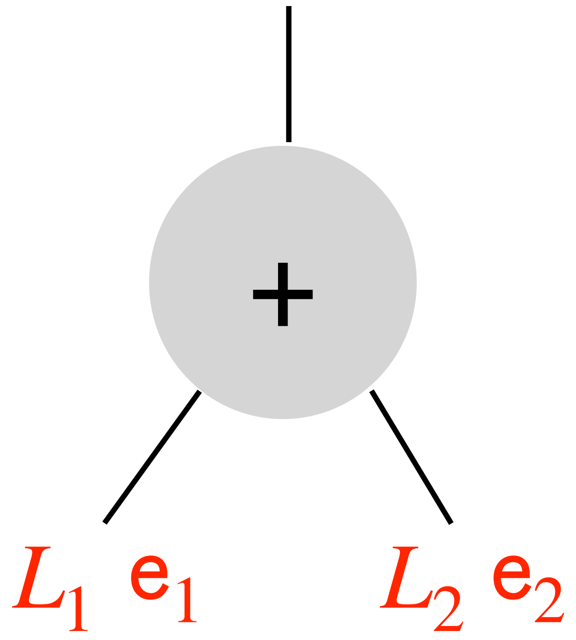
Garbler

Invariant

$$g_i + e_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot e_i \pmod{p}$$

Evaluator



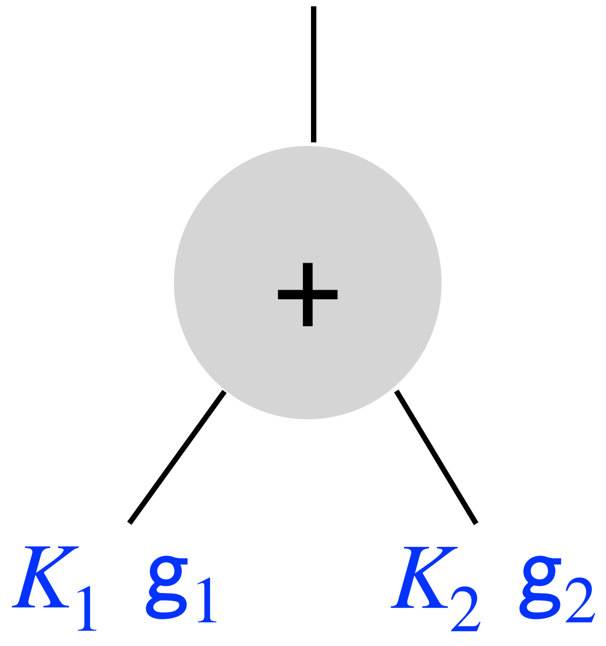
Computed over the integers

$$K^* = K_1 + K_2 \bmod p$$

$$e^* = e_1 + e_2$$

$$L^* = L_1 + L_2 \bmod p$$

Modular Arithmetic Garbling



Δ

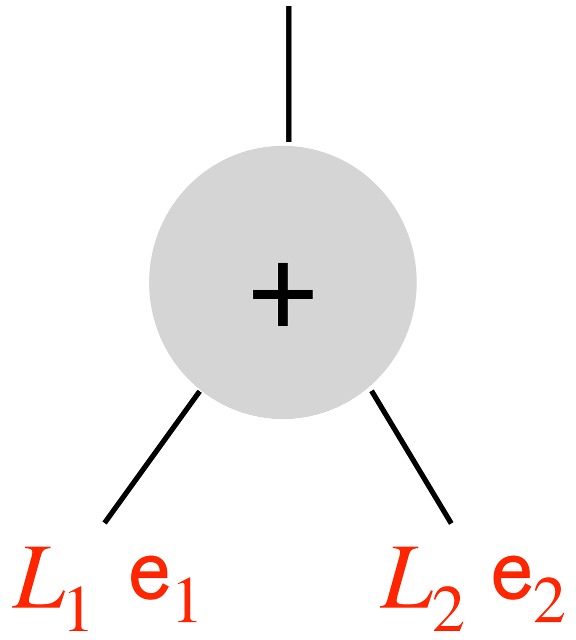
Garbler

Invariant

$$g_i + e_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot e_i \pmod{p}$$

Evaluator

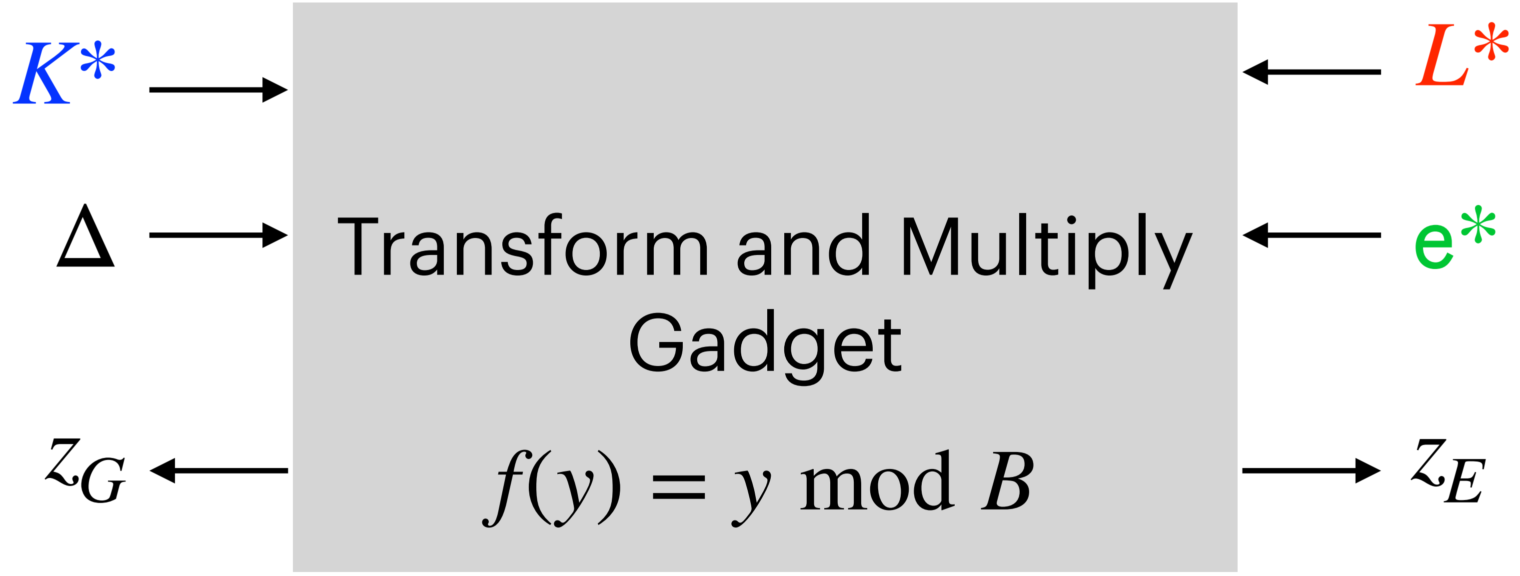


Computed over the integers

$$e^* = e_1 + e_2$$

$$L^* = L_1 + L_2 \bmod p$$

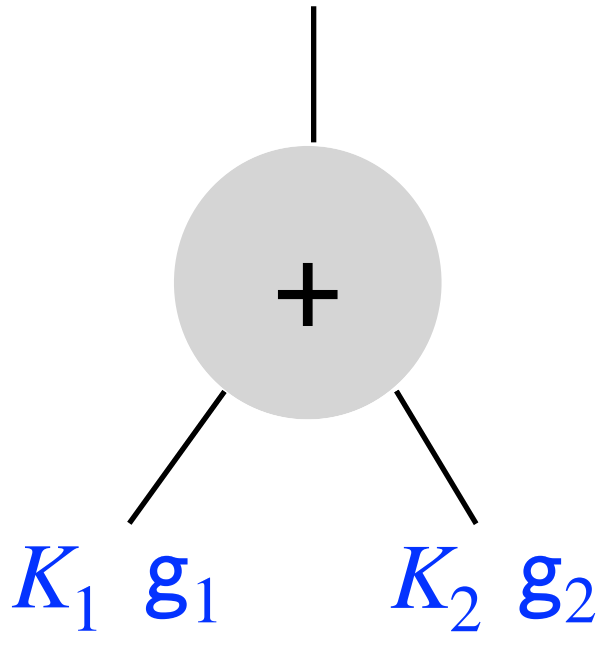
$$K^* = K_1 + K_2 \bmod p$$



Transform and Multiply Gadget

$f(y) = y \bmod B$

Modular Arithmetic Garbling



Δ

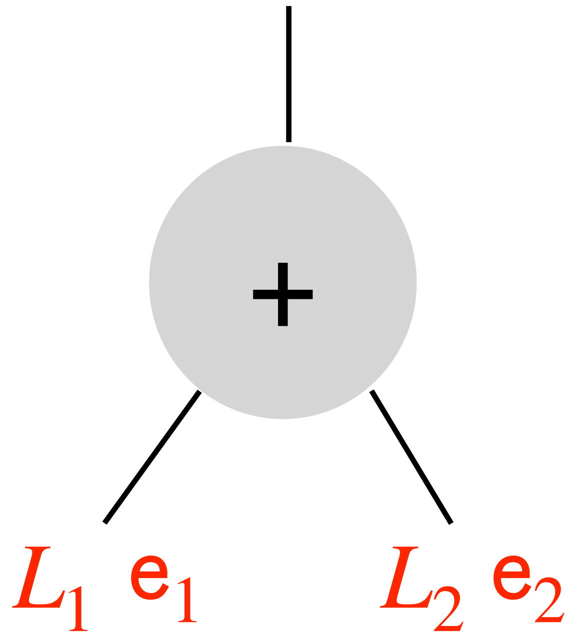
Garbler

Invariant

$$g_i + e_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot e_i \pmod{p}$$

Evaluator

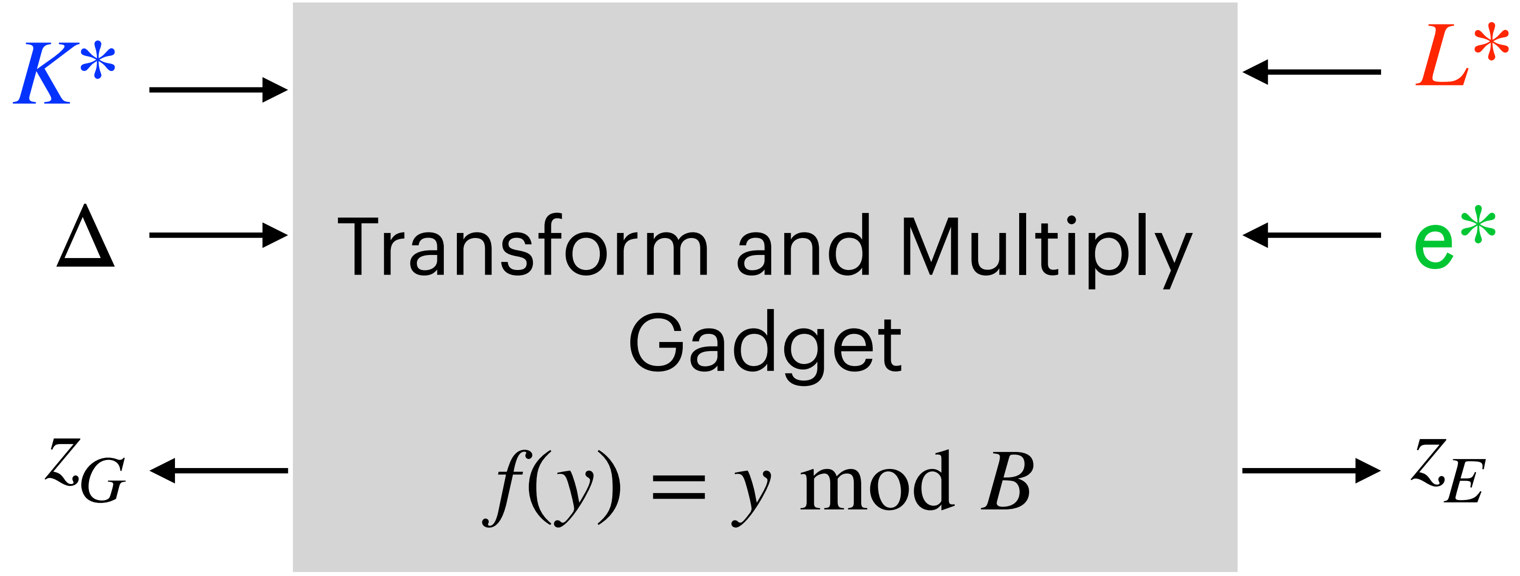


Computed over the integers

$$e^* = e_1 + e_2$$

$$L^* = L_1 + L_2 \bmod p$$

$$K^* = K_1 + K_2 \bmod p$$

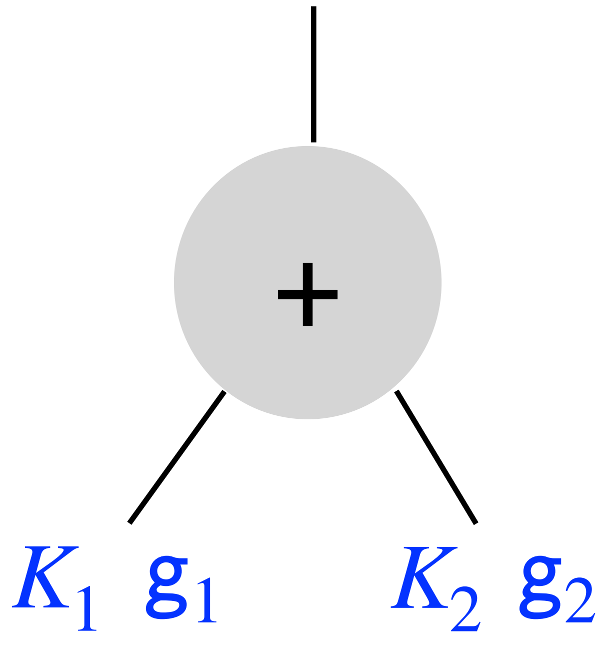


Transform and Multiply Gadget

$f(y) = y \bmod B$

$$K^* + L^* \bmod p = \Delta \cdot (e_1 + e_2) \bmod p$$
$$= \Delta \cdot e^* \bmod p$$

Modular Arithmetic Garbling



Δ

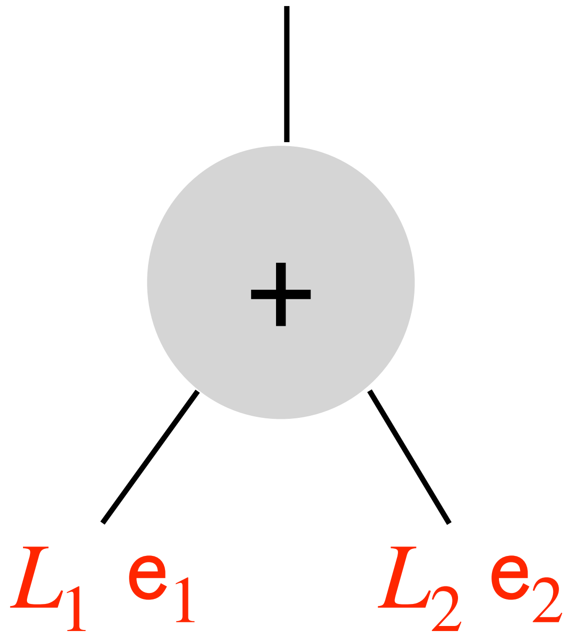
Garbler

Invariant

$$g_i + e_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot e_i \pmod{p}$$

Evaluator

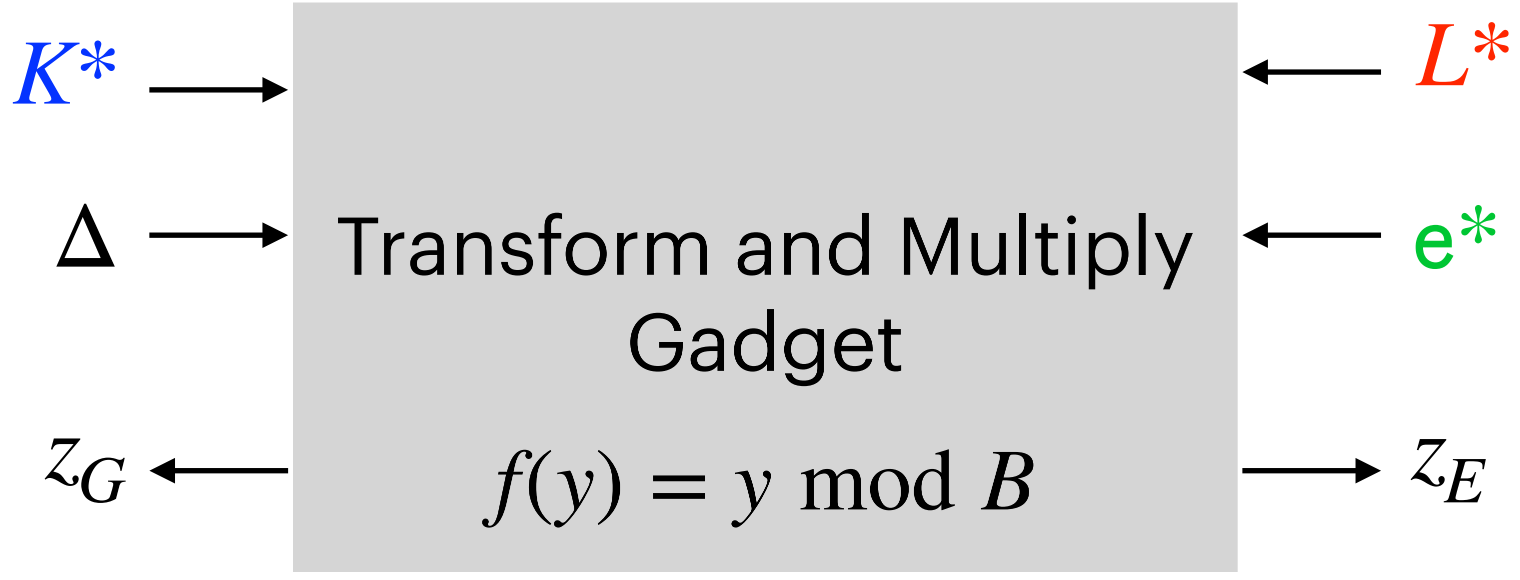


Computed over the integers

$$e^* = e_1 + e_2$$

$$L^* = L_1 + L_2 \bmod p$$

$$K^* = K_1 + K_2 \bmod p$$



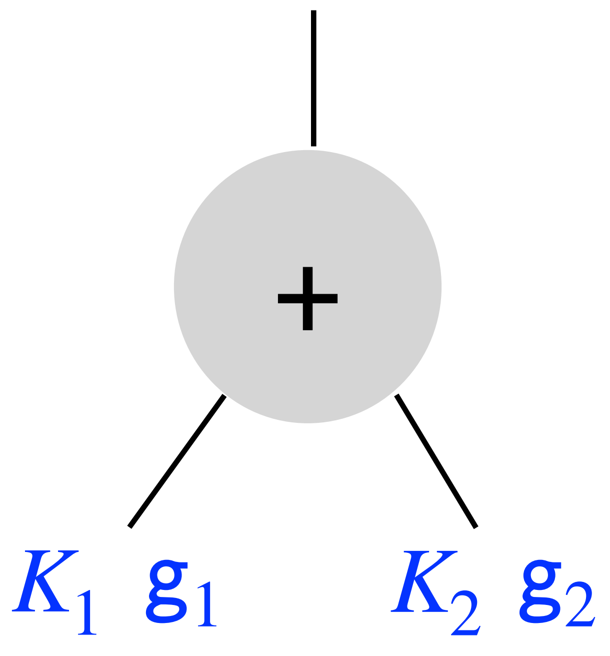
Transform and Multiply Gadget

$$f(y) = y \bmod B$$

$$K^* + L^* \bmod p = \Delta \cdot (e_1 + e_2) \bmod p$$
$$= \Delta \cdot e^* \bmod p$$

$$z_G + z_E = \Delta \cdot f(e^*)$$
$$= \Delta \cdot \big((e_1 + e_2) \bmod B\big)$$

Modular Arithmetic Garbling



Δ

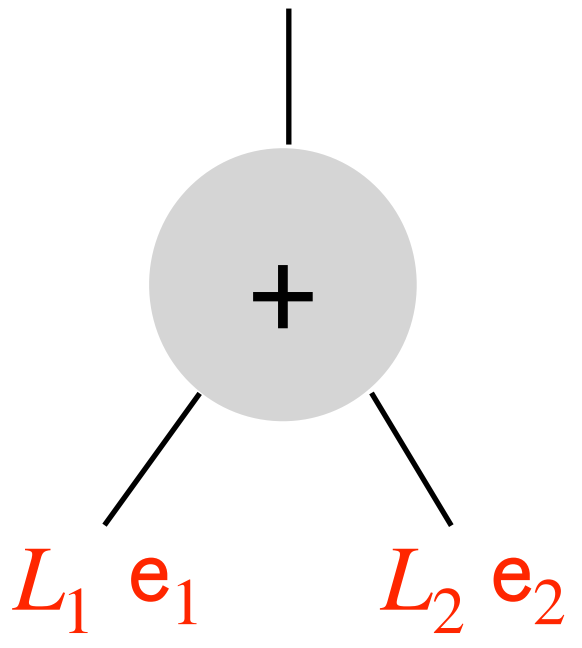
Garbler

Invariant

$$g_i + e_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot e_i \pmod{p}$$

Evaluator

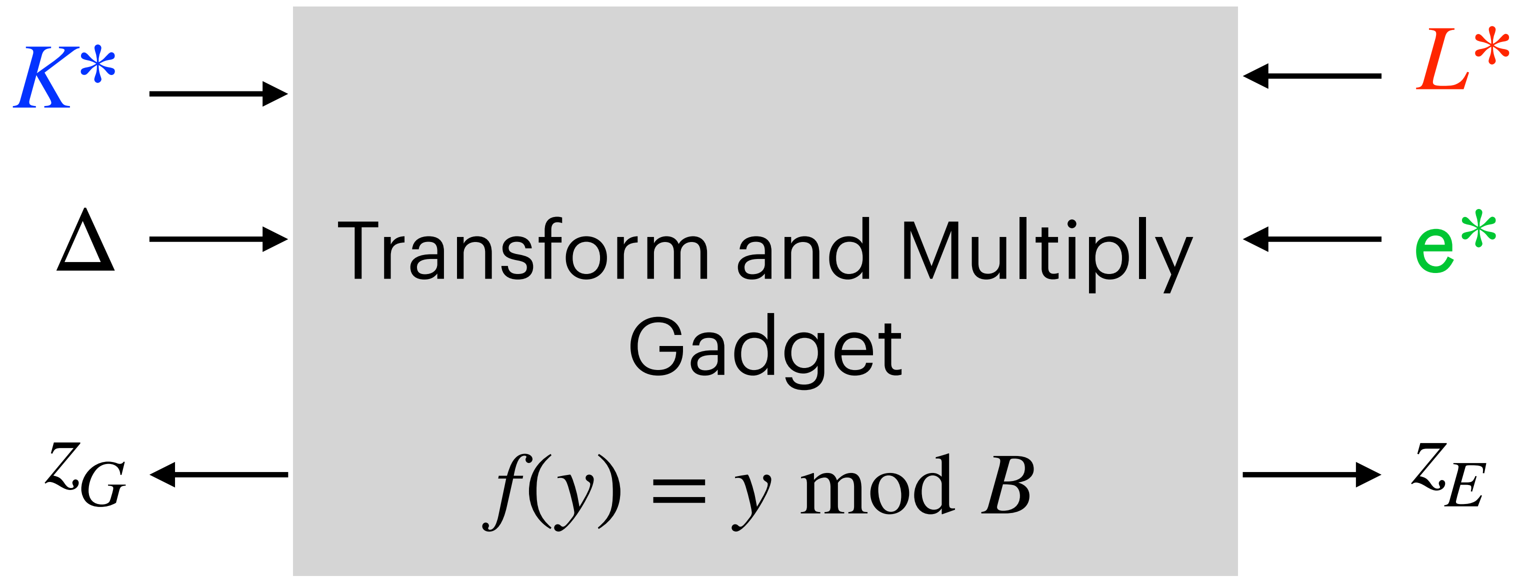


Computed over the integers

$$e^* = e_1 + e_2$$

$$L^* = L_1 + L_2 \bmod p$$

$$K^* = K_1 + K_2 \bmod p$$

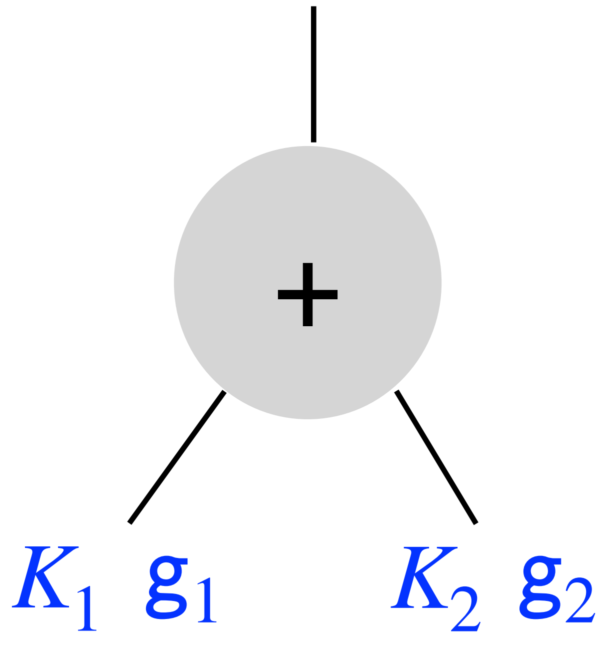


Transform and Multiply Gadget

$$f(y) = y \bmod B$$

$$z_G + z_E = \Delta \cdot \big((e_1 + e_2) \bmod B\big)$$

Modular Arithmetic Garbling



Δ

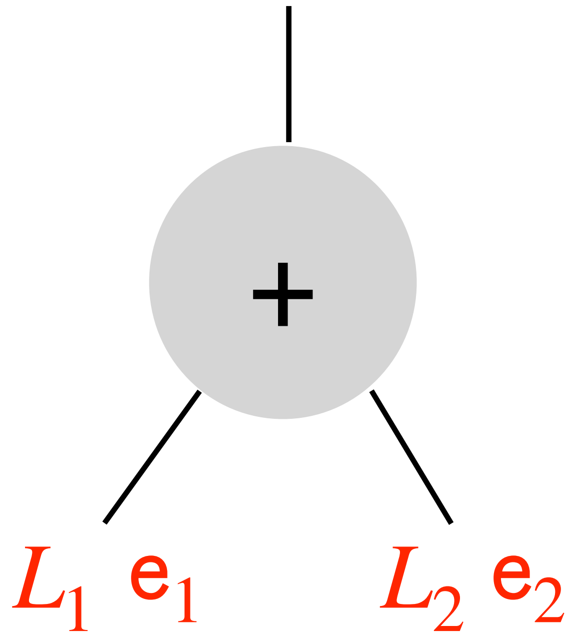
Garbler

Invariant

$$g_i + e_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot e_i \pmod{p}$$

Evaluator

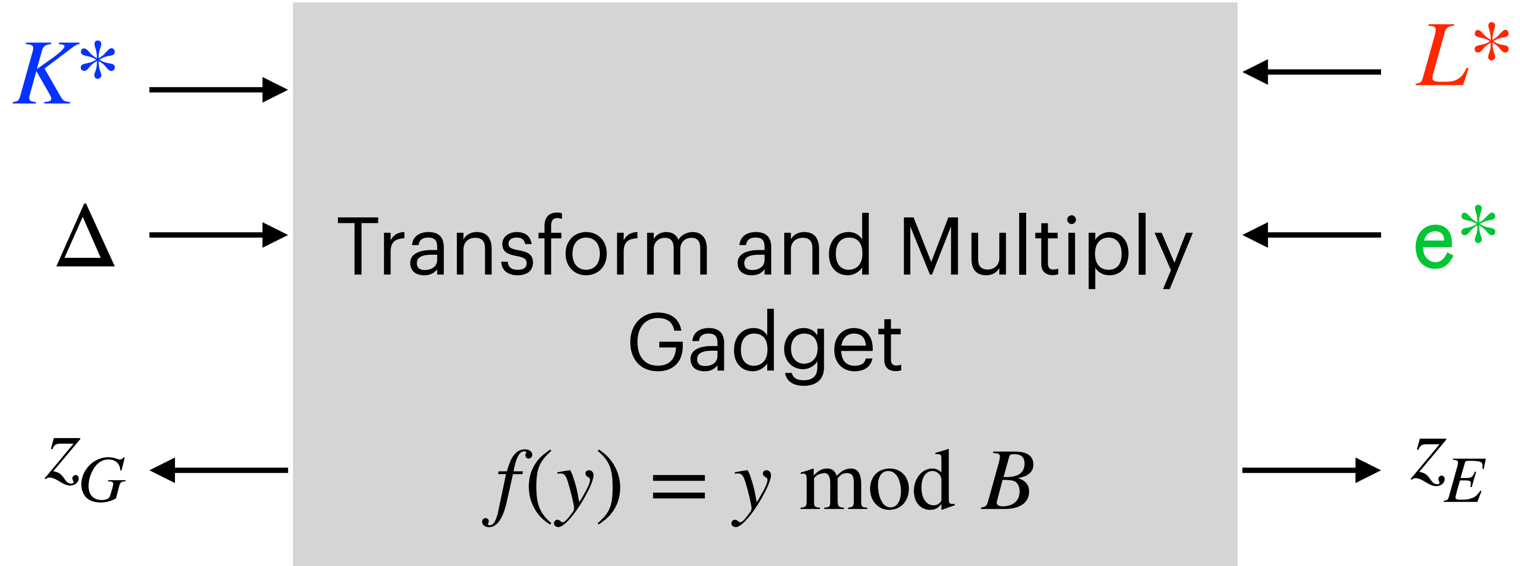


Computed over the integers

$$e^* = e_1 + e_2$$

$$K^* = K_1 + K_2 \bmod p$$

$$L^* = L_1 + L_2 \bmod p$$



Transform and Multiply Gadget

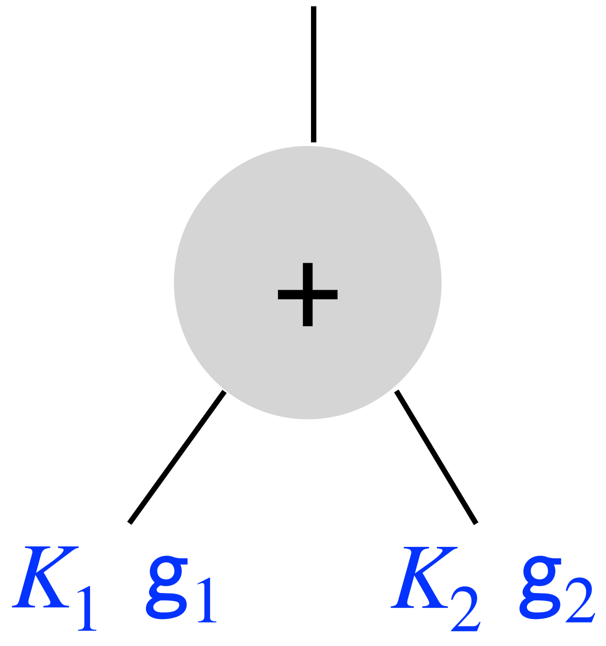
$$f(y) = y \bmod B$$

$$g_3 = g_1 + g_2 \bmod B$$

$$e_3 = e_1 + e_2 \bmod B$$

$$z_G + z_E = \Delta \cdot \big((e_1 + e_2) \bmod B\big)$$

Modular Arithmetic Garbling



Δ

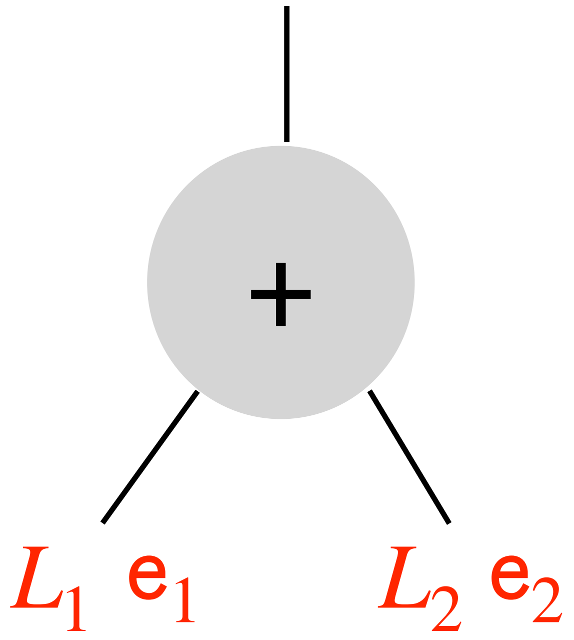
Garbler

Invariant

$$g_i + e_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot e_i \pmod{p}$$

Evaluator

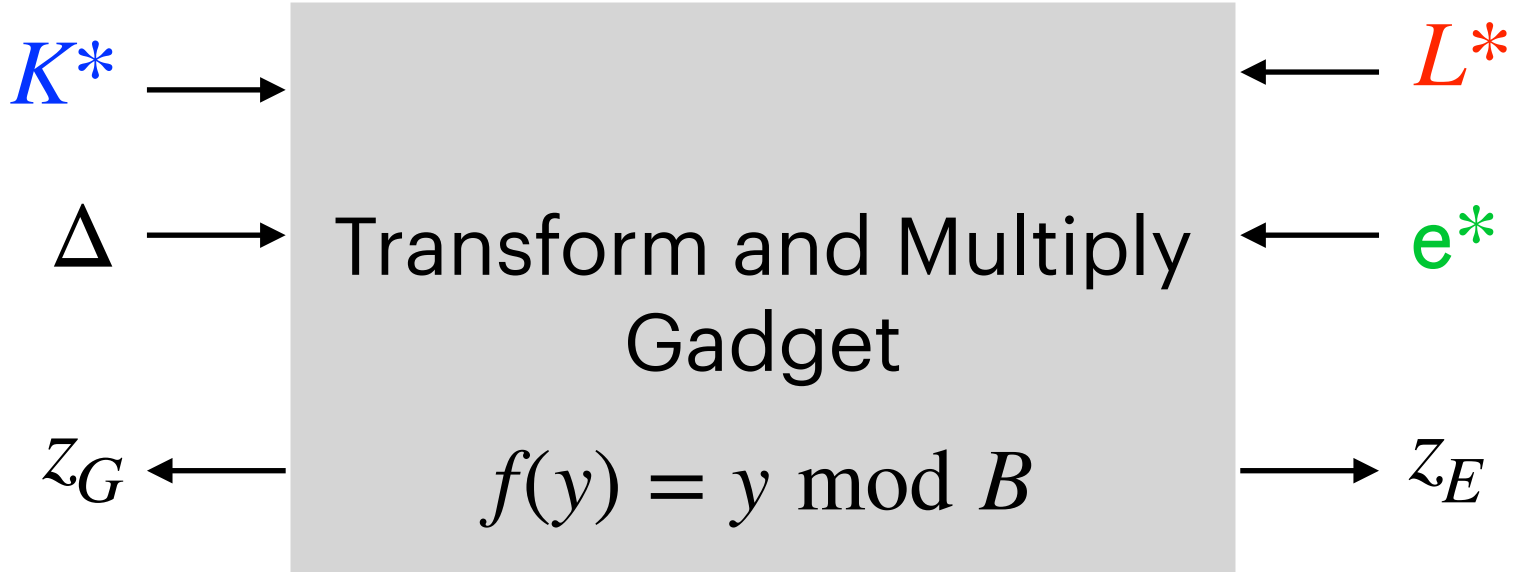


Computed over the integers

$$e^* = e_1 + e_2$$

$$L^* = L_1 + L_2 \bmod p$$

$$K^* = K_1 + K_2 \bmod p$$



Transform and Multiply Gadget

$f(y) = y \bmod B$

$$g_3 = g_1 + g_2 \bmod B$$

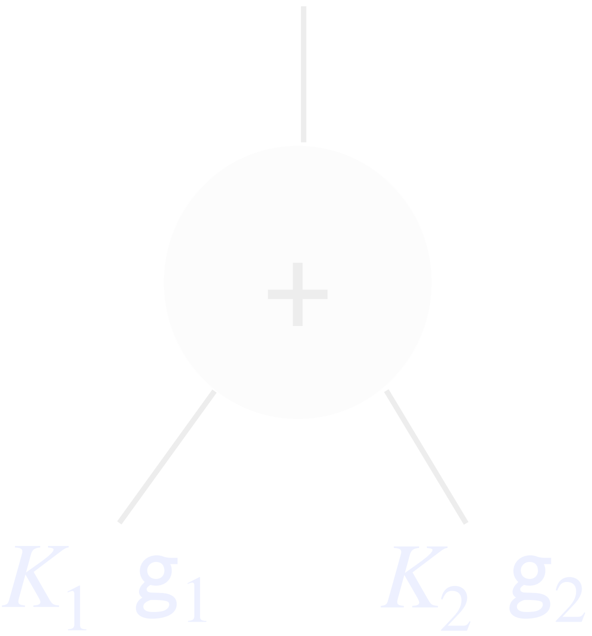
$$K_3 = z_G \bmod p$$

$$e_3 = e_1 + e_2 \bmod B$$

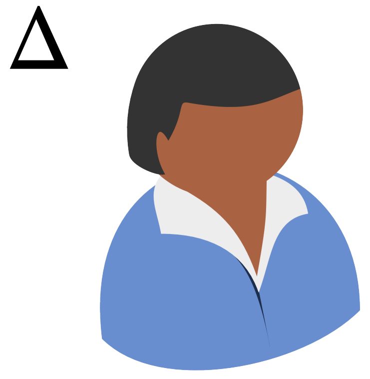
$$L_3 = z_E \bmod p$$

$$z_G + z_E = \Delta \cdot \big((e_1 + e_2) \bmod B\big)$$

Modular Arithmetic Garbling



Δ



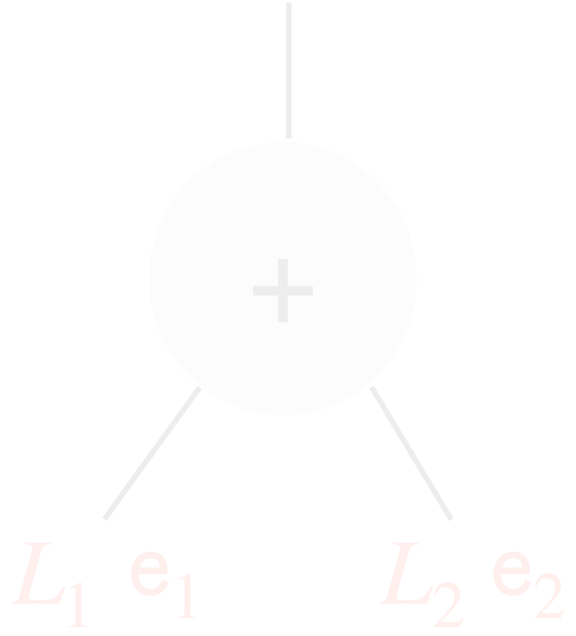
Garbler

Invariant

$$g_i + e_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot e_i \pmod{p}$$

Evaluator

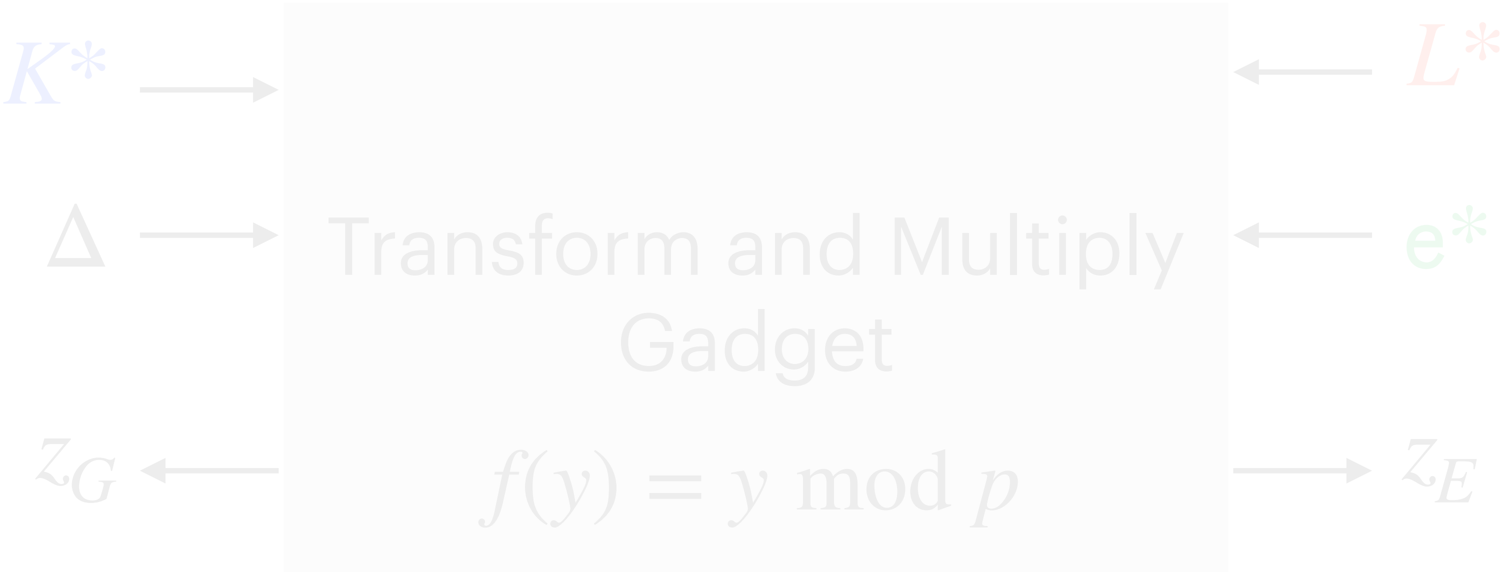


Computed over the integers

$e^* = e_1 + e_2$

$L^* = L_1 + L_2 \bmod p$

$K^* = K_1 + K_2 \bmod p$



Transform and Multiply Gadget

$f(y) = y \bmod p$

$g_3 = g_1 + g_2 \bmod B$

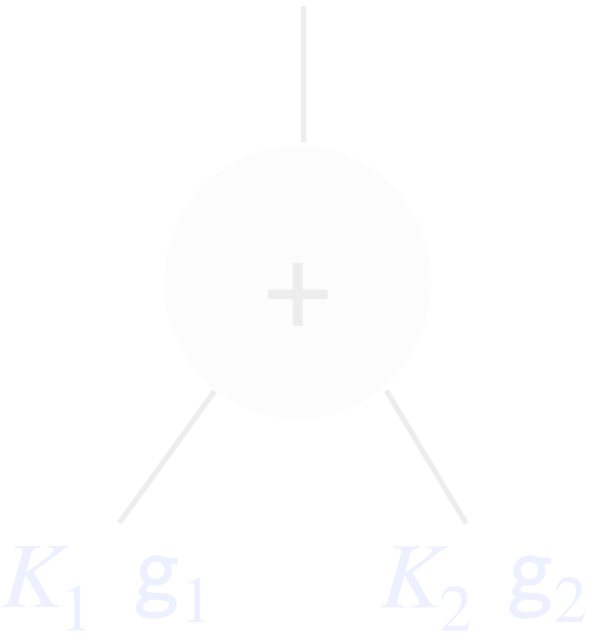
$K_3 = z_G \bmod p$

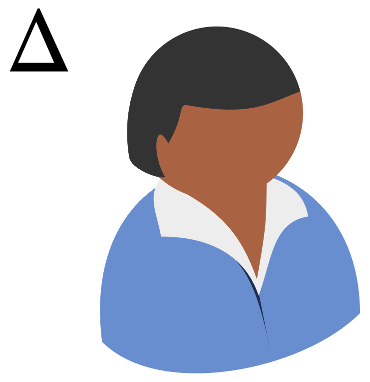
$e_3 = e_1 + e_2 \bmod B$

$L_3 = z_E \bmod p$

$$z_G + z_E = \Delta \cdot \big((e_1 + e_2) \bmod B\big)$$

Modular Arithmetic Garbling





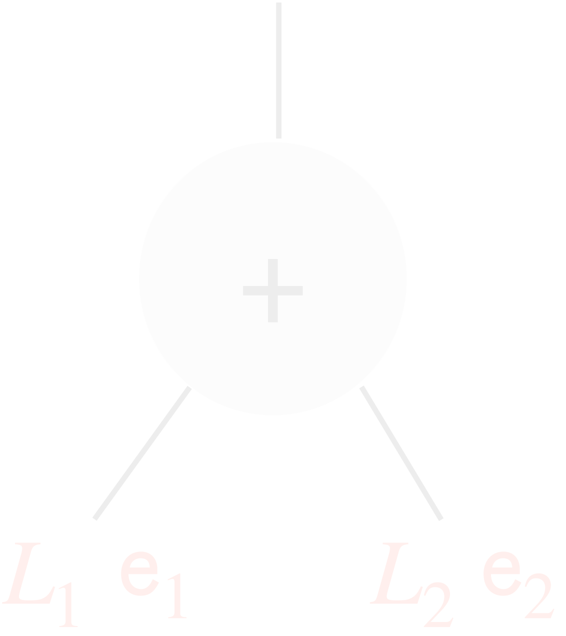
Garbler

Invariant

$$g_i + e_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot e_i \pmod{p}$$

Evaluator



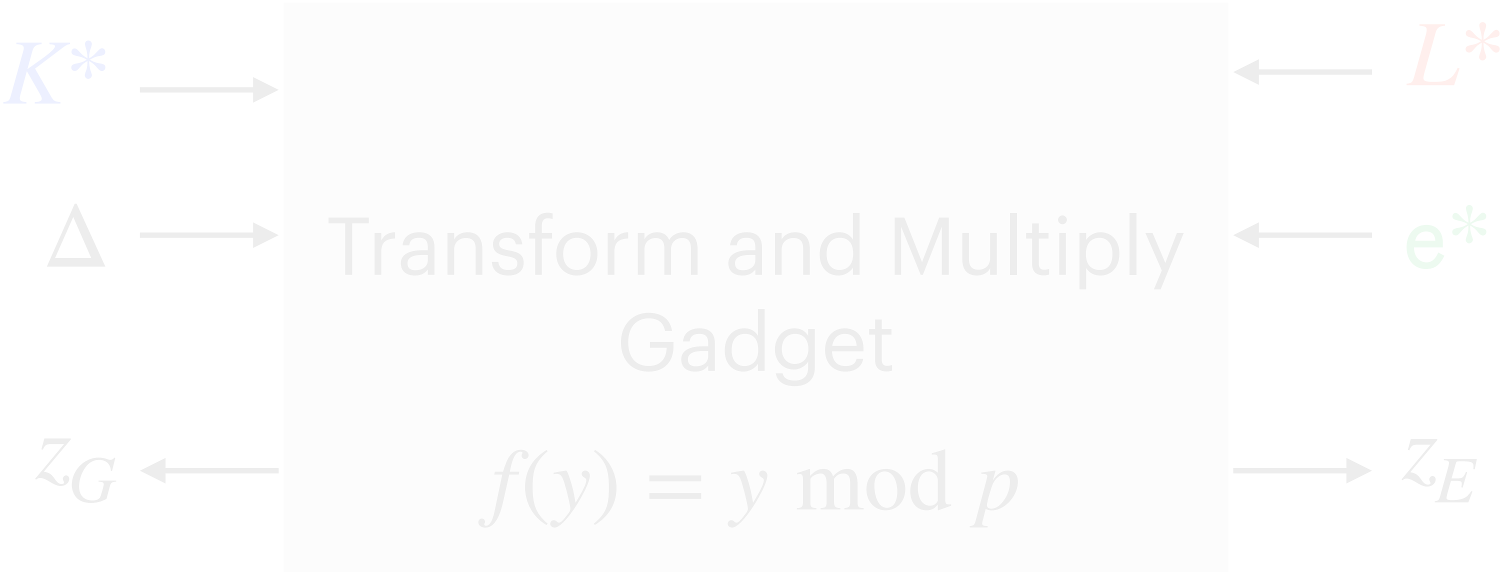
Computed over the integers

$$g_3 + e_3 = x_1 + x_2 \pmod{B}$$

$$e^* = e_1 + e_2$$

$$K^* = K_1 + K_2 \bmod p$$

$$L^* = L_1 + L_2 \bmod p$$



$$g_3 = g_1 + g_2 \bmod B$$

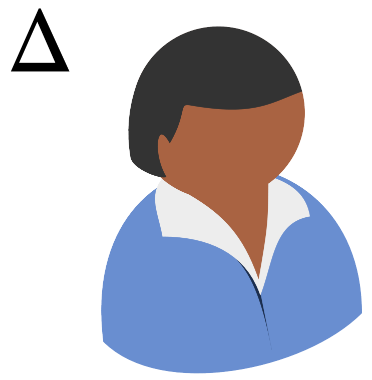
$$K_3 = z_G \bmod p$$

$$e_3 = e_1 + e_2 \bmod B$$

$$L_3 = z_E \bmod p$$

$$z_G + z_E = \Delta \cdot \big((e_1 + e_2) \bmod B\big)$$

Modular Arithmetic Garbling



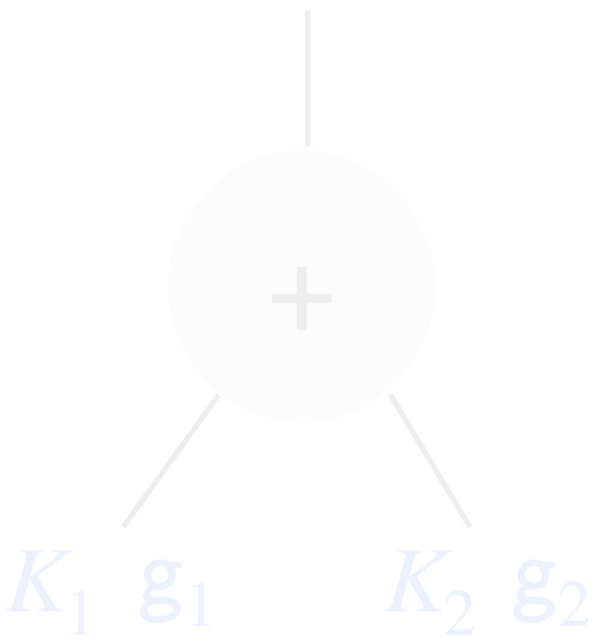
Garbler

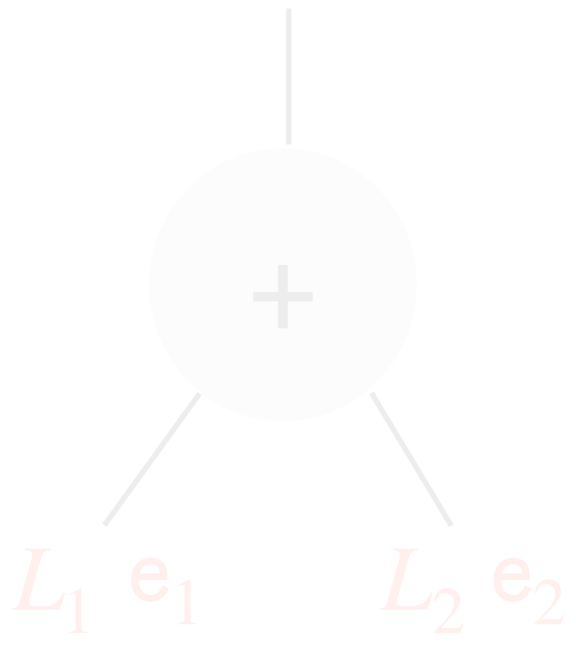
Invariant

$$g_i + e_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot e_i \pmod{p}$$

Evaluator





Computed over the integers

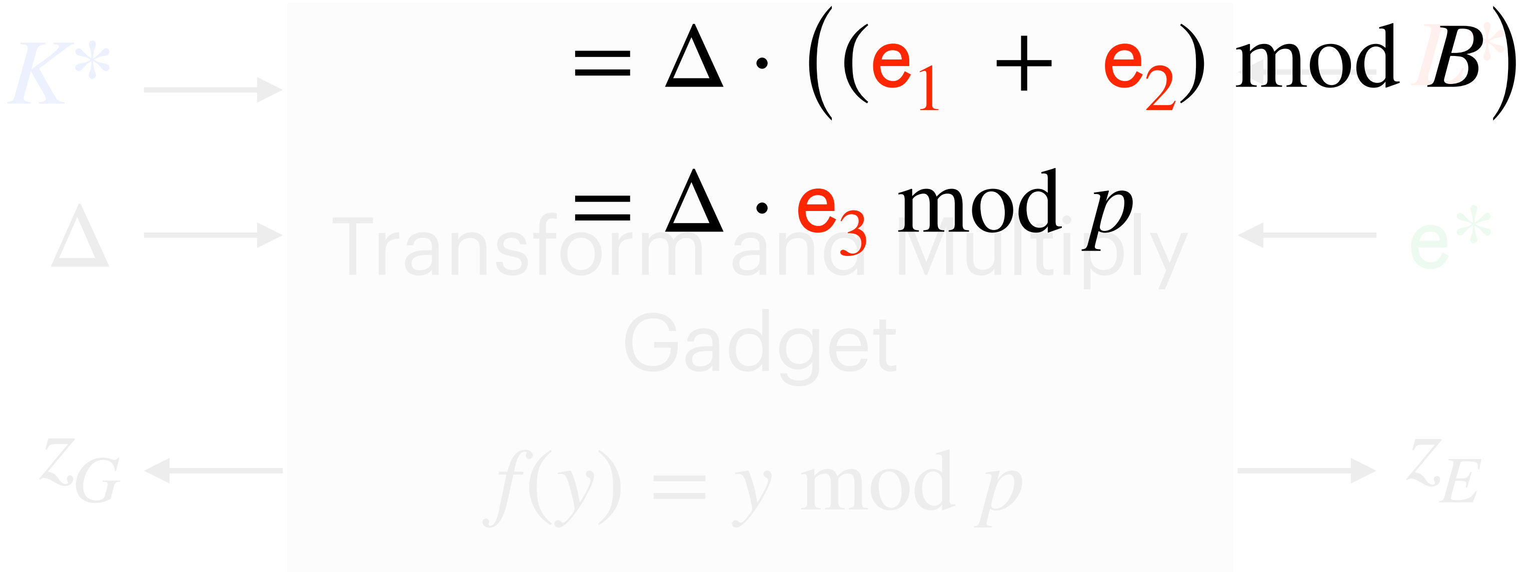
$$g_3 + e_3 = x_1 + x_2 \pmod{B}$$

$$K^* = K_1 + K_2 \bmod p$$

$$e^* = e_1 + e_2$$

$$L^* = L_1 + L_2 \bmod p$$

$$\begin{aligned} K_3 + L_3 &= z_G + z_E \bmod p \\ &= \Delta \cdot \big((e_1 + e_2) \bmod B\big) \bmod p \\ &= \Delta \cdot e_3 \bmod p \end{aligned}$$



$$g_3 = g_1 + g_2 \bmod B$$

$$K_3 = z_G \bmod p$$

$$e_3 = e_1 + e_2 \bmod B$$

$$L_3 = z_E \bmod p$$

$$z_G + z_E = \Delta \cdot \big((e_1 + e_2) \bmod B\big)$$

Modular Arithmetic Garbling

Δ

Garbler

Invariant

$$g_i + e_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot e_i \pmod{p}$$

Evaluator

$+$

K_1

Invariant maintained!

$+$

$L_1 \; e_1 \qquad L_2 \; e_2$

Computed over the integers

$$e^* = e_1 + e_2$$

$$L^* = L_1 + L_2 \bmod p$$

$$K^* = K_1 + K_2 \bmod p$$

$$g_3 + e_3 = x_1 + x_2 \pmod{B}$$

$$K_3 + L_3 = z_G + z_E \bmod p$$

$$= \Delta \cdot \big((e_1 + e_2) \bmod B\big) \bmod p$$

$$= \Delta \cdot e_3 \bmod p$$

$K^* \rightarrow$

$\Delta \rightarrow$

$z_G \leftarrow$

Transform and Multiply Gadget

$f(y) = y \bmod p$

$\leftarrow e^*$

$\rightarrow z_E$

$$g_3 = g_1 + g_2 \bmod B$$

$$K_3 = z_G \bmod p$$

$$e_3 = e_1 + e_2 \bmod B$$

$$L_3 = z_E \bmod p$$

$$z_G + z_E = \Delta \cdot \big((e_1 + e_2) \bmod B\big)$$

Modular Arithmetic Garbling

Δ

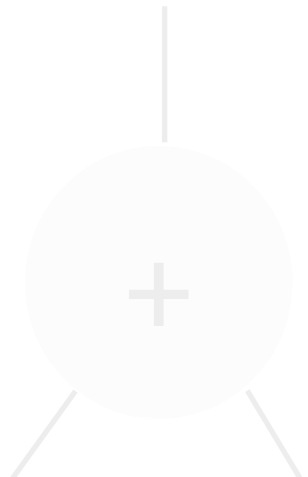
Garbler

Invariant

$$g_i + e_i = x_i \pmod{B}$$

$$K_i + L_i = \Delta \cdot e_i \pmod{p}$$

Evaluator



K_1

Invariant maintained!

L_1 e_1 L_2 e_2

Computed over the integers

$$g_3 + e_3 = x_1 + x_2 \pmod{B}$$

$e^* = e_1 + e_2$

$L^* = L_1 + L_2 \bmod p$

Transform-and-multiply suffices for multiplication gates too

$$K_3 + L_3 = z_G + z_E \bmod p$$

$$= \Delta \cdot \big((e_1 + e_2) \bmod B\big) \bmod p$$

$$= \Delta \cdot e_3 \bmod p$$

$K^* \rightarrow$

$\Delta \rightarrow$ Transform and Multiply Gadget $\leftarrow e^*$

$f(y) = y \bmod p$

$z_G \leftarrow$ $\rightarrow z_E$

$$g_3 = g_1 + g_2 \bmod B$$

$$K_3 = z_G \bmod p$$

$$e_3 = e_1 + e_2 \bmod B$$

$$L_3 = z_E \bmod p$$

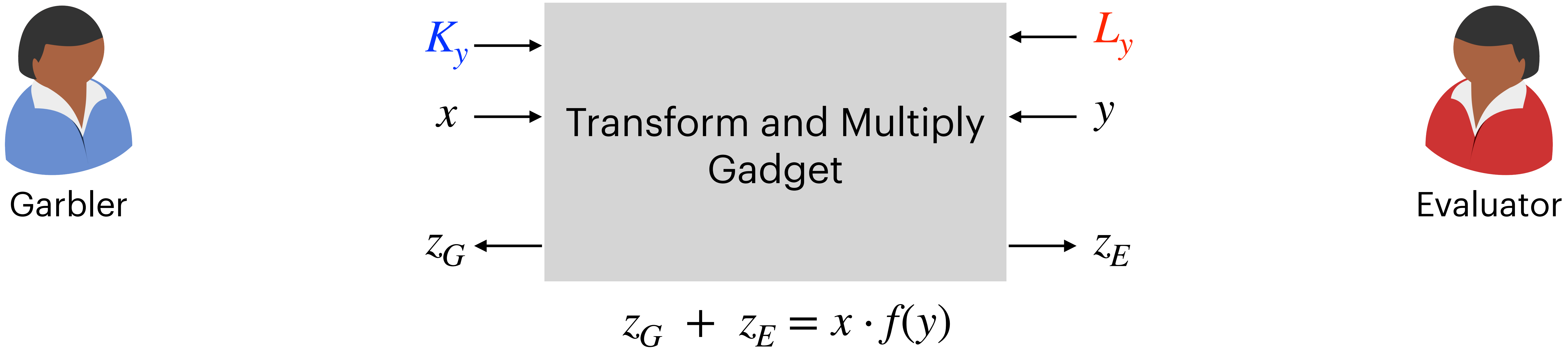
$$z_G + z_E = \Delta \cdot \big((e_1 + e_2) \bmod B\big)$$

Modular Arithmetic Garbling

- Each addition and multiplication gate makes constant number of Transform-and-Multiply invocations
- Each invocation to Transform-and-Multiply adds $\Theta(\lambda)$-bits to the garbled circuit

Rate: $$\frac{|C| \cdot \log B}{|C| \cdot \Theta(\lambda)} = \Theta\left(\frac{\log B}{\lambda}\right)$$

Constructing the Transform-and-Multiply Gadget



Garbler

$K_y \rightarrow$

$x \rightarrow$

$z_G \leftarrow$

Transform and Multiply Gadget

$\leftarrow L_y$

$\leftarrow y$

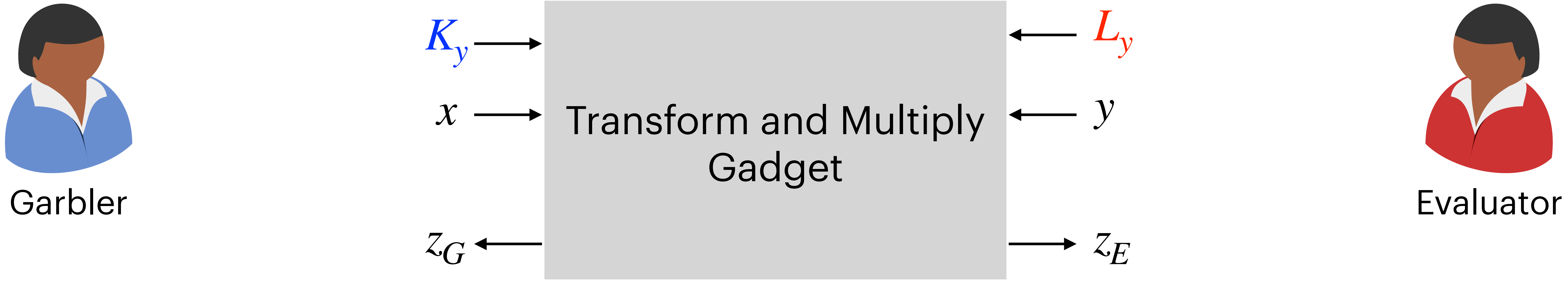
$\rightarrow z_E$

Evaluator

$$z_G + z_E = x \cdot f(y)$$

$$K_y + L_y = \Delta \cdot y \pmod p$$

Constructing the Transform-and-Multiply Gadget



Garbler

K_y → Transform and Multiply Gadget ← L_y

x → ← y

z_G ← → z_E

Evaluator

$$z_G + z_E = x \cdot f(y)$$

$$K_y + L_y = \Delta \cdot y \pmod{p}$$

Property of power-DDH based PPRF

Parties can non-interactively convert their shares

Garbler: Convert K_y to PRF key k

Evaluator: Convert L_y to PRF key k_y punctured at y

Constructing the Transform-and-Multiply Gadget

[Roy'22] [Heath-Kolesnikov-Ng'24]

Garbler

$$K_y + L_y = \Delta \cdot y \pmod{p}$$

K_y

x

z_G

Transform and Multiply Gadget

L_y

y

z_E

$$z_G + z_E = x \cdot f(y)$$

Evaluator

Constructing the Transform-and-Multiply Gadget

[Roy’22] [Heath-Kolesnikov-Ng’24]

Garbler

$$K_y + L_y = \Delta \cdot y \pmod{p}$$

$K_y \rightarrow$

$x \rightarrow$

$z_G \leftarrow$

Transform and Multiply Gadget

$\leftarrow L_y$

$\leftarrow y$

$\rightarrow z_E$

$$z_G + z_E = x \cdot f(y)$$

Evaluator

$$K_y \xrightarrow{\text{Convert}} k$$

Constructing the Transform-and-Multiply Gadget

[Roy'22] [Heath-Kolesnikov-Ng'24]

$$K_y + L_y = \Delta \cdot y \pmod{p}$$

Garbler

Evaluator

$$z_G + z_E = x \cdot f(y)$$

$$K_y \xrightarrow{\text{Convert}} k$$

$$A = \sum_{i=0}^{B-1} \text{PRF}(k, i)$$

Constructing the Transform-and-Multiply Gadget

[Roy'22] [Heath-Kolesnikov-Ng'24]

Garbler

$$K_y + L_y = \Delta \cdot y \pmod{p}$$

$K_y \rightarrow$

$x \rightarrow$

$z_G \leftarrow$

Transform and Multiply Gadget

$\leftarrow L_y$

$\leftarrow y$

$\rightarrow z_E$

$$z_G + z_E = x \cdot f(y)$$

Evaluator

$$K_y \xrightarrow{\text{Convert}} k$$

$$A = \sum_{i=0}^{B-1} \mathsf{PRF}(k, i)$$

$$z_G = \sum_{i=0}^{B-1} f(i) \cdot \mathsf{PRF}(k, i)$$

Constructing the Transform-and-Multiply Gadget

[Roy'22] [Heath-Kolesnikov-Ng'24]

$$K_y + L_y = \Delta \cdot y \pmod{p}$$

Garbler

Evaluator

$$z_G + z_E = x \cdot f(y)$$

$$K_y \xrightarrow{\text{Convert}} k$$

$$A = \sum_{i=0}^{B-1} \text{PRF}(k, i)$$

$$z_G = \sum_{i=0}^{B-1} f(i) \cdot \text{PRF}(k, i) \qquad \xrightarrow{\quad x - A \quad}$$

Constructing the Transform-and-Multiply Gadget

[Roy’22] [Heath-Kolesnikov-Ng’24]

$$K_y + L_y = \Delta \cdot y \pmod{p}$$

Garbler

Evaluator

$$z_G + z_E = x \cdot f(y)$$

$$K_y \xrightarrow{\text{Convert}} k$$

$$L_y \xrightarrow{\text{Convert}} k_y$$

$$A = \sum_{i=0}^{B-1} \mathsf{PRF}(k, i)$$

$$z_E = \sum_{i \neq y} \big(f(y) - f(i)\big) \cdot \mathsf{PRF}(k_y, i) + f(y) \cdot (x - A)$$

$$z_G = \sum_{i=0}^{B-1} f(i) \cdot \mathsf{PRF}(k, i)$$

$$\xrightarrow{x - A}$$

Constructing the Transform-and-Multiply Gadget

[Roy'22] [Heath-Kolesnikov-Ng'24]

Garbler

$$K_y + L_y = \Delta \cdot y \pmod p$$

$K_y \rightarrow$

$x \rightarrow$

$z_G \leftarrow$

Transform and Multiply Gadget

$\leftarrow L_y$

$\leftarrow y$

$\rightarrow z_E$

$$z_G + z_E = x \cdot f(y)$$

Evaluator

$$K_y \xrightarrow{\text{Convert}} k$$

$$L_y \xrightarrow{\text{Convert}} k_y$$

$$A = \sum_{i=0}^{B-1} \mathsf{PRF}(k, i)$$

$$z_E = \sum_{i \neq y} \big(f(y) - f(i)\big) \cdot \mathsf{PRF}(k_y, i) + f(y) \cdot (x - A)$$

$$z_G = \sum_{i=0}^{B-1} f(i) \cdot \mathsf{PRF}(k, i)$$

$$\xrightarrow{x - A}$$

$$= \sum_{i=0}^{B-1} \big(f(y) - f(i)\big) \cdot \mathsf{PRF}(k, i) + f(y) \cdot (x - A)$$

Constructing the Transform-and-Multiply Gadget

[Roy’22] [Heath-Kolesnikov-Ng’24]

$K_y + L_y = \Delta \cdot y \pmod{p}$

$z_G + z_E = x \cdot f(y)$

$K_y \xrightarrow{\text{Convert}} k$

$L_y \xrightarrow{\text{Convert}} k_y$

$$A = \sum_{i=0}^{B-1} \text{PRF}(k, i)$$

$$z_G = \sum_{i=0}^{B-1} f(i) \cdot \text{PRF}(k, i)$$

When $i \neq y$, PRF(k_y, i) = PRF(k, i)

When $i = y$, both terms are 0

$$= \sum_{i \neq y} \big(f(y) - f(i)\big) \cdot \text{PRF}(k_y, i) + f(y) \cdot (x - A)$$

$$= \sum_{i=0}^{B-1} \big(f(y) - f(i)\big) \cdot \text{PRF}(k, i) + f(y) \cdot (x - A)$$

Constructing the Transform-and-Multiply Gadget

[Roy’22] [Heath-Kolesnikov-Ng’24]

Garbler

$$K_y + L_y = \Delta \cdot y \pmod{p}$$

K_y

x

Transform and Multiply Gadget

z_G

L_y

y

z_E

$$z_G + z_E = x \cdot f(y)$$

Evaluator

$$K_y \xrightarrow{\text{Convert}} k$$

$$L_y \xrightarrow{\text{Convert}} k_y$$

$$A = \sum_{i=0}^{B-1} \mathsf{PRF}(k, i)$$

$$z_G = \sum_{i=0}^{B-1} f(i) \cdot \mathsf{PRF}(k, i)$$

$$\xrightarrow{\quad x - A \quad}$$

$$z_E = \sum_{i \neq y} \big(f(y) - f(i)\big) \cdot \mathsf{PRF}(k_y, i) + f(y) \cdot (x - A)$$

$$= \sum_{i=0}^{B-1} \big(f(y) - f(i)\big) \cdot \mathsf{PRF}(k, i) + f(y) \cdot (x - A)$$

$$= f(y) \cdot A - z_G + x \cdot f(y) - f(y) \cdot A$$

$$= x \cdot f(y) - z_G$$

Constructing the Transform-and-Multiply Gadget

[Roy'22] [Heath-Kolesnikov-Ng'24]

$$K_y + L_y = \Delta \cdot y \pmod p$$

Garbler

$K_y \rightarrow$

$x \rightarrow$

$z_G \leftarrow$

Transform and Multiply Gadget

$\leftarrow L_y$

$\leftarrow y$

$\rightarrow z_E$

Evaluator

$$z_G + z_E = x \cdot f(y)$$

$$K_y \xrightarrow{\text{Convert}} k$$

$$L_y \xrightarrow{\text{Convert}} k_y$$

$$A = \sum_{i=0}^{B-1} \mathsf{PRF}(k, i)$$

$$z_G = \sum_{i=0}^{B-1} f(i) \cdot \mathsf{PRF}(k, i)$$

x remains private since it is masked by PRF(k, y)

$$\xrightarrow{x - A}$$

$$z_E = \sum_{i \neq y} \big(f(y) - f(i)\big) \cdot \mathsf{PRF}(k_y, i) + f(y) \cdot (x - A)$$

$$= \sum_{i=0}^{B-1} \big(f(y) - f(i)\big) \cdot \mathsf{PRF}(k, i) + f(y) \cdot (x - A)$$

$$= f(y) \cdot A - z_G + x \cdot f(y) - f(y) \cdot A$$

$$= x \cdot f(y) - z_G$$

Power-DDH Based Punctured PRF

Power-DDH Based Punctured PRF

msk $= \alpha$

h is not given as part of CRS; pseudorandom from power-DDH

Domain size of PRF

crs $= \quad h^{\alpha^{-B}} \quad h^{\alpha^{-B+1}} \quad \ldots \quad h^{\alpha^{-4}} \quad h^{\alpha^{-3}} \quad h^{\alpha^{-2}} \quad h^{\alpha^{-1}} \quad \blacksquare \quad h^{\alpha^{1}} \quad h^{\alpha^{2}} \quad h^{\alpha^{3}} \quad h^{\alpha^{4}} \quad \ldots \quad h^{\alpha^{B-1}} \quad h^{\alpha^{B}}$

Power-DDH Based Punctured PRF

$\mathsf{msk} = \alpha$

KeyGen: $k \leftarrow \mathbb{Z}_q$

crs = $h^{\alpha^{-B}}$ $h^{\alpha^{-B+1}}$ $\ldots$ $h^{\alpha^{-4}}$ $h^{\alpha^{-3}}$ $h^{\alpha^{-2}}$ $h^{\alpha^{-1}}$ ■ $h^{\alpha^{1}}$ $h^{\alpha^{2}}$ $h^{\alpha^{3}}$ $h^{\alpha^{4}}$ $\ldots$ $h^{\alpha^{B-1}}$ $h^{\alpha^{B}}$

Power-DDH Based Punctured PRF

$\mathsf{msk} = \alpha$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Eval(msk, k, x) : $h^{k \cdot \alpha^x}$

crs = $h^{\alpha^{-B}}$ $h^{\alpha^{-B+1}}$ … $h^{\alpha^{-4}}$ $h^{\alpha^{-3}}$ $h^{\alpha^{-2}}$ $h^{\alpha^{-1}}$ $\blacksquare$ $h^{\alpha^{1}}$ $h^{\alpha^{2}}$ $h^{\alpha^{3}}$ $h^{\alpha^{4}}$ … $h^{\alpha^{B-1}}$ $h^{\alpha^{B}}$

Power-DDH Based Punctured PRF

msk $= \alpha$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Eval(msk, k, x) : $h^{k \cdot \alpha^x}$

PRF Inputs:	0	1	2	3	4		$B-1$	B
	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$		$\downarrow$	$\downarrow$
PRF Outputs:	h^{k}	$h^{k \cdot \alpha^1}$	$h^{k \cdot \alpha^2}$	$h^{k \cdot \alpha^3}$	$h^{k \cdot \alpha^4}$	$\ldots$	$h^{k \cdot \alpha^{B-1}}$	$h^{k \cdot \alpha^B}$

crs $= \quad h^{\alpha^{-B}} \quad h^{\alpha^{-B+1}} \quad \ldots \quad h^{\alpha^{-4}} \quad h^{\alpha^{-3}} \quad h^{\alpha^{-2}} \quad h^{\alpha^{-1}}$ ■ $h^{\alpha^1} \quad h^{\alpha^2} \quad h^{\alpha^3} \quad h^{\alpha^4} \quad \ldots \quad h^{\alpha^{B-1}} \quad h^{\alpha^B}$

Power-DDH Based Punctured PRF

msk $= \alpha$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Eval(msk, k, x) : $h^{k \cdot \alpha^x}$

Puncture(msk, k, y): $k_y = \alpha^y \cdot k$

PRF Inputs:	0	1	2	3	4		$B-1$	B
	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$		$\downarrow$	$\downarrow$
PRF Outputs:	h^{k}	$h^{k \cdot \alpha^1}$	$h^{k \cdot \alpha^2}$	$h^{k \cdot \alpha^3}$	$h^{k \cdot \alpha^4}$	$\ldots$	$h^{k \cdot \alpha^{B-1}}$	$h^{k \cdot \alpha^B}$

crs $= \quad h^{\alpha^{-B}} \quad h^{\alpha^{-B+1}} \quad \ldots \quad h^{\alpha^{-4}} \quad h^{\alpha^{-3}} \quad h^{\alpha^{-2}} \quad h^{\alpha^{-1}} \quad \blacksquare \quad h^{\alpha^{1}} \quad h^{\alpha^{2}} \quad h^{\alpha^{3}} \quad h^{\alpha^{4}} \quad \ldots \quad h^{\alpha^{B-1}} \quad h^{\alpha^{B}}$

Power-DDH Based Punctured PRF

msk $= \alpha$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Eval(msk, k, x) : $h^{k \cdot \alpha^x}$

Puncture(msk, k, y): $k_y = \alpha^y \cdot k$

PunctEval(k_y, x): $\mathsf{crs}_{x-z}^{k_y}$

PRF Inputs:	0	1	2	3	4		$B-1$	B
	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$		$\downarrow$	$\downarrow$
PRF Outputs:	h^k	$h^{k \cdot \alpha^1}$	$h^{k \cdot \alpha^2}$	$h^{k \cdot \alpha^3}$	$h^{k \cdot \alpha^4}$	$\ldots$	$h^{k \cdot \alpha^{B-1}}$	$h^{k \cdot \alpha^B}$

crs $=$ $h^{\alpha^{-B}}$ $h^{\alpha^{-B+1}}$ $\ldots$ $h^{\alpha^{-4}}$ $h^{\alpha^{-3}}$ $h^{\alpha^{-2}}$ $h^{\alpha^{-1}}$ ■ h^{α^1} h^{α^2} h^{α^3} h^{α^4} $\ldots$ $h^{\alpha^{B-1}}$ h^{α^B}

Power-DDH Based Punctured PRF

msk $= \alpha$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Eval(msk, k, x) : $h^{k \cdot \alpha^x}$

Puncture(msk, k, y): $k_y = \alpha^y \cdot k$

PunctEval(k_y, x): crs$_{x-z}^{k_y}$

$$\begin{array}{lcccccccc} \text{PRF Inputs:} & 0 & 1 & 2 & 3 & 4 & & B-1 & B \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ \text{PRF Outputs:} & h^{k} & h^{k \cdot \alpha^1} & h^{k \cdot \alpha^2} & h^{k \cdot \alpha^3} & h^{k \cdot \alpha^4} & \dots & h^{k \cdot \alpha^{B-1}} & h^{k \cdot \alpha^B} \end{array}$$

crs $= \quad h^{\alpha^{-B}} \quad h^{\alpha^{-B+1}} \quad \dots \quad h^{\alpha^{-4}} \quad h^{\alpha^{-3}} \quad h^{\alpha^{-2}} \quad h^{\alpha^{-1}} \quad$ ■ $\quad h^{\alpha^1} \quad h^{\alpha^2} \quad h^{\alpha^3} \quad h^{\alpha^4} \quad \dots \quad h^{\alpha^{B-1}} \quad h^{\alpha^B}$

$k_3 = \alpha^3 \cdot k$

Power-DDH Based Punctured PRF

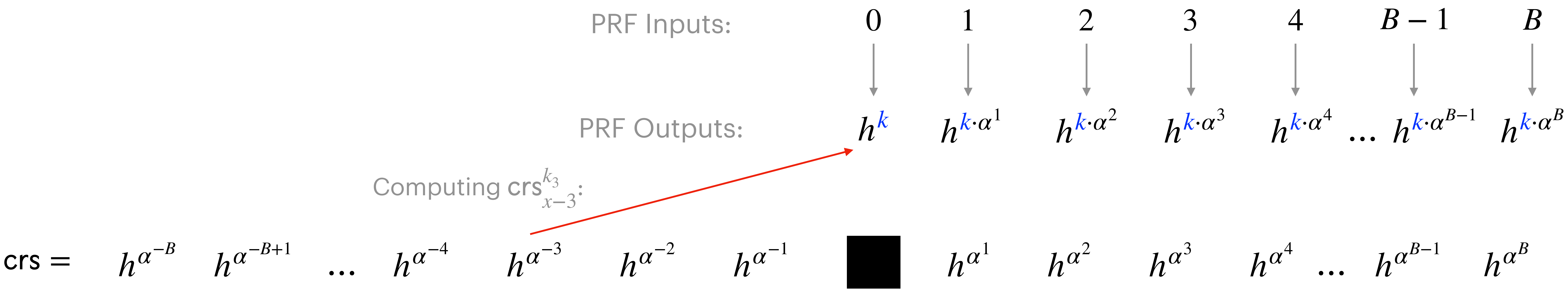
msk $= \alpha$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Eval(msk, k, x) : $h^{k \cdot \alpha^x}$

Puncture(msk, k, y): $k_y = \alpha^y \cdot k$

PunctEval(k_y, x): $\mathsf{crs}_{x-z}^{k_y}$



PRF Inputs: $0 \quad 1 \quad 2 \quad 3 \quad 4 \quad B-1 \quad B$

PRF Outputs: $h^{k} \quad h^{k\cdot\alpha^1} \quad h^{k\cdot\alpha^2} \quad h^{k\cdot\alpha^3} \quad h^{k\cdot\alpha^4} \ \dots \ h^{k\cdot\alpha^{B-1}} \quad h^{k\cdot\alpha^B}$

Computing $\mathsf{crs}_{x-3}^{k_3}$:

$\mathsf{crs} = \quad h^{\alpha^{-B}} \quad h^{\alpha^{-B+1}} \ \dots \ h^{\alpha^{-4}} \quad h^{\alpha^{-3}} \quad h^{\alpha^{-2}} \quad h^{\alpha^{-1}} \quad \blacksquare \quad h^{\alpha^1} \quad h^{\alpha^2} \quad h^{\alpha^3} \quad h^{\alpha^4} \ \dots \ h^{\alpha^{B-1}} \quad h^{\alpha^B}$

$k_3 = \alpha^3 \cdot k$

Power-DDH Based Punctured PRF

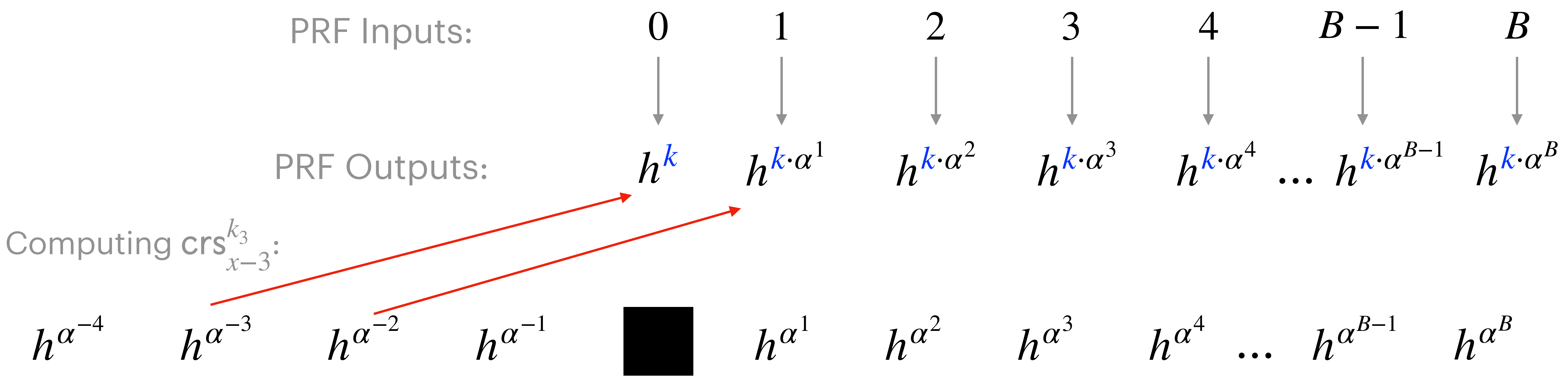
msk $= \alpha$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Eval(msk, k, x) : $h^{k \cdot \alpha^x}$

Puncture(msk, k, y): $k_y = \alpha^y \cdot k$

PunctEval(k_y, x): $\text{crs}_{x-z}^{k_y}$



PRF Inputs: $0 \quad 1 \quad 2 \quad 3 \quad 4 \quad B-1 \quad B$

PRF Outputs: $h^{k} \quad h^{k \cdot \alpha^1} \quad h^{k \cdot \alpha^2} \quad h^{k \cdot \alpha^3} \quad h^{k \cdot \alpha^4} \; \ldots \; h^{k \cdot \alpha^{B-1}} \quad h^{k \cdot \alpha^B}$

Computing $\text{crs}_{x-3}^{k_3}$:

crs $= \quad h^{\alpha^{-B}} \quad h^{\alpha^{-B+1}} \; \ldots \; h^{\alpha^{-4}} \quad h^{\alpha^{-3}} \quad h^{\alpha^{-2}} \quad h^{\alpha^{-1}} \quad ■ \quad h^{\alpha^1} \quad h^{\alpha^2} \quad h^{\alpha^3} \quad h^{\alpha^4} \; \ldots \; h^{\alpha^{B-1}} \quad h^{\alpha^B}$

$$k_3 = \alpha^3 \cdot k$$

Power-DDH Based Punctured PRF

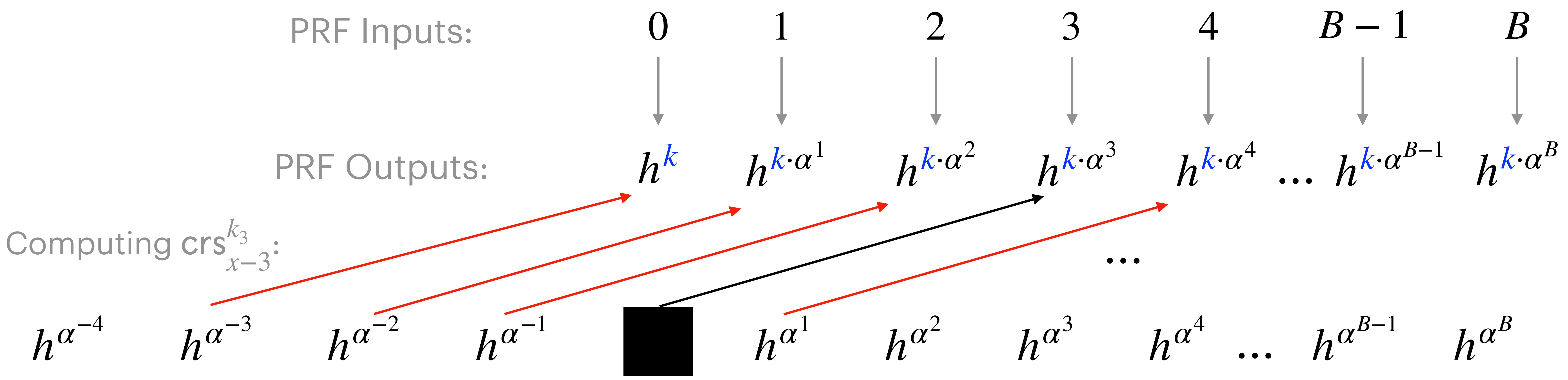
msk $= \alpha$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Eval(msk, k, x) : $h^{k\cdot\alpha^{x}}$

Puncture(msk, k, y): $k_y = \alpha^y \cdot k$

PunctEval(k_y, x): $\mathsf{crs}^{k_y}_{x-z}$



PRF Inputs: $0 \quad 1 \quad 2 \quad 3 \quad 4 \quad B-1 \quad B$

PRF Outputs: $h^{k} \quad h^{k\cdot\alpha^{1}} \quad h^{k\cdot\alpha^{2}} \quad h^{k\cdot\alpha^{3}} \quad h^{k\cdot\alpha^{4}} \ \dots \ h^{k\cdot\alpha^{B-1}} \quad h^{k\cdot\alpha^{B}}$

Computing $\mathsf{crs}^{k_3}_{x-3}$:

$\dots$

$\mathsf{crs} = \quad h^{\alpha^{-B}} \quad h^{\alpha^{-B+1}} \quad \dots \quad h^{\alpha^{-4}} \quad h^{\alpha^{-3}} \quad h^{\alpha^{-2}} \quad h^{\alpha^{-1}} \quad \blacksquare \quad h^{\alpha^{1}} \quad h^{\alpha^{2}} \quad h^{\alpha^{3}} \quad h^{\alpha^{4}} \ \dots \ h^{\alpha^{B-1}} \quad h^{\alpha^{B}}$

$k_3 = \alpha^3 \cdot k$

Power-DDH Based Punctured PRF

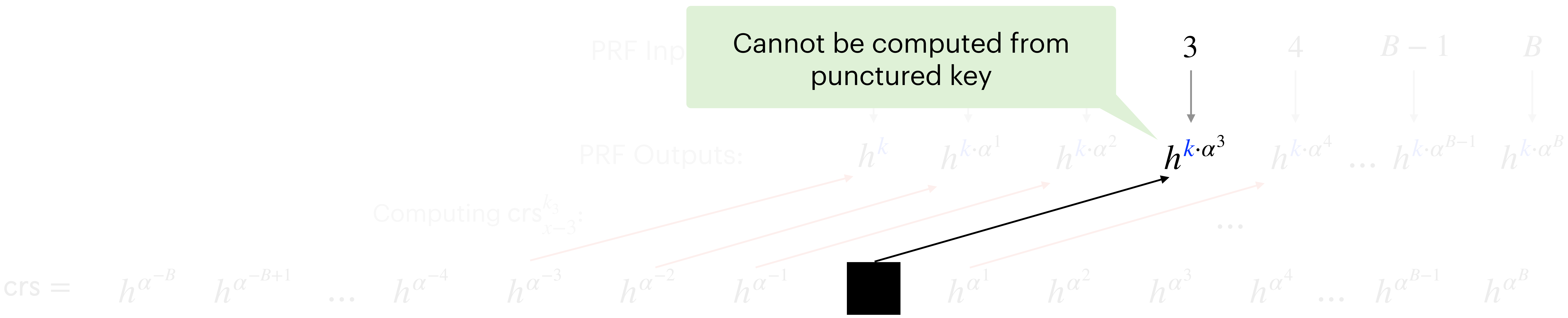
msk $= \alpha$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Eval(msk, k, x) : $h^{k \cdot \alpha^x}$

Puncture(msk, k, y): $k_y = \alpha^y \cdot k$

PunctEval(k_y, x): $\mathsf{crs}_{x-z}^{k_y}$



Cannot be computed from punctured key

$$k_3 = \alpha^3 \cdot k$$

Power-DDH Based Punctured PRF

$\text{msk} = \alpha$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Eval(msk, k, x) : $h^{k \cdot \alpha^x}$

$\text{crs} = \{h^{\alpha^i}\}_{i \neq 0}$

Puncture(msk, k, y): $k_y = \alpha^y \cdot k$

PunctEval(k_y, x): $\text{crs}_{x-z}^{k_y}$

Power-DDH Based Punctured PRF

msk $= \alpha$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Eval(msk, k, x) : $h^{k \cdot \alpha^x}$

crs $= \{h^{\alpha^i}\}_{i \neq 0}$

Puncture(msk, k, y): $k_y = \alpha^y \cdot k$

PunctEval(k_y, x): crs$^{k_y}_{x-z}$


Garbler

$$K_y + L_y = \Delta \cdot y \pmod p$$


Evaluator

Power-DDH Based Punctured PRF

msk $= \alpha$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Eval(msk, k, x) : $h^{k \cdot \alpha^x}$

crs $= \{h^{\alpha^i}\}_{i \neq 0}$

Puncture(msk, k, y): $k_y = \alpha^y \cdot k$

PunctEval(k_y, x): crs$^{k_y}_{x-z}$


Garbler

Let G be a generator of $\mathbb{Z}_q^*$

$$\alpha = G^{\Delta}$$

$$K_y + L_y = \Delta \cdot y \pmod{p}$$


Evaluator

Power-DDH Based Punctured PRF

msk $= \alpha$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Eval(msk, k, x) : $h^{k\cdot\alpha^x}$

crs $= \{h^{\alpha^i}\}_{i\neq 0}$

Puncture(msk, k, y): $k_y = \alpha^y \cdot k$

PunctEval(k_y, x): $\mathsf{crs}_{x-z}^{k_y}$

Garbler

Let G be a generator of $\mathbb{Z}_q^*$

$$\alpha = G^{\Delta}$$

$$K_y + L_y = \Delta \cdot y \pmod{q-1}$$

Evaluator

Power-DDH Based Punctured PRF

msk $= \alpha$

crs $= \{h^{\alpha^i}\}_{i \neq 0}$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Puncture(msk, k, y): $k_y = \alpha^y \cdot k$

Eval(msk, k, x) : $h^{k \cdot \alpha^x}$

PunctEval(k_y, x): $\text{crs}_{x-z}^{k_y}$

Garbler

Let G be a generator of $\mathbb{Z}_q^*$

$$\alpha = G^{\Delta}$$

$$K_y + L_y = \Delta \cdot y \pmod{q-1}$$

Evaluator

$$K_y \xrightarrow{\text{Convert}} k$$

$$k = G^{-K_y}$$

Power-DDH Based Punctured PRF

$\text{msk} = \alpha$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Eval(msk, k, x) : $h^{k \cdot \alpha^x}$

$\text{crs} = \{h^{\alpha^i}\}_{i \neq 0}$

Puncture(msk, k, y): $k_y = \alpha^y \cdot k$

PunctEval(k_y, x): $\text{crs}_{x-z}^{k_y}$

Garbler

Let G be a generator of $\mathbb{Z}_q^*$

$$\alpha = G^{\Delta}$$

$$K_y + L_y = \Delta \cdot y \pmod{q-1}$$

Evaluator

$$K_y \xrightarrow{\text{Convert}} k$$

$$k = G^{-K_y}$$

$$L_y \xrightarrow{\text{Convert}} k_y$$

$$k_y = G^{L_y}$$

Power-DDH Based Punctured PRF

$\mathsf{msk} = \alpha$

$\mathsf{crs} = \{h^{\alpha^i}\}_{i \neq 0}$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Puncture(msk, k, y): $k_y = \alpha^y \cdot k$

Eval(msk, k, x) : $h^{k \cdot \alpha^x}$

PunctEval(k_y, x): $\mathsf{crs}_{x-z}^{k_y}$

Garbler

Let G be a generator of $\mathbb{Z}_q^*$

$$\alpha = G^{\Delta}$$

$$K_y + L_y = \Delta \cdot y \pmod{q-1}$$

Evaluator

$K_y \xrightarrow{\text{Convert}} k$

$k = G^{-K_y}$

$L_y \xrightarrow{\text{Convert}} k_y$

$k_y = G^{L_y} = G^{\Delta \cdot y - K_y} = \alpha^y \cdot k$

Power-DDH Based Punctured PRF

msk $= \alpha$

KeyGen: $k \leftarrow \mathbb{Z}_q$

Eval(msk, k, x) : $h^{k \cdot \alpha^x}$

crs $= \{h^{\alpha^i}\}_{i \neq 0}$

Puncture(msk, k, y): $k_y = \alpha^y \cdot k$

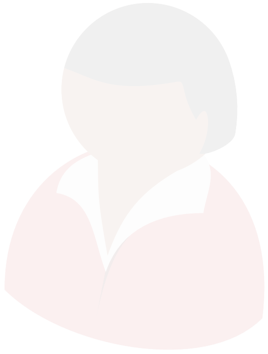
PunctEval(k_y, x): crs$_{x-z}^{k_y}$

Garbler

Let G be a generator of $\mathbb{Z}_q^*$

$$\alpha = G^{\Delta}$$

$$K_y + L_y = \Delta \cdot y \pmod{q-1}$$



Evaluator

$$K_y \xrightarrow{\text{Convert}} k$$

$$k = G^{-K_y}$$

$$L_y \xrightarrow{\text{Convert}} k_y$$

$$k_y = G^{L_y} = G^{\Delta \cdot y - K_y} = \alpha^y \cdot k$$

Conclusion

- Handling multiplication gates requires composing Transform-and-multiply and requires developing new techniques to bounds the output share size while ensuring privacy
- Using the same Δ for all wires requires circular security $\implies$ switch Δ at each level
- Extends via CRT to $\mathbb{Z}_N$ of arbitrary size as long as N is smooth
- **Open Questions**
  - Can we do better than rate-$O(1/\lambda)$ modular arithmetic garbling even for super-polynomial size rings?
  - Can we improve the rate beyond $O(\log \lambda / \lambda)$?

Conclusion

- Handling multiplication gates requires composing Transform-and-multiply and requires developing new techniques to bounds the output share size while ensuring privacy
- Using the same Δ for all wires requires circular security $\implies$ switch Δ at each level
- Extends via CRT to $\mathbb{Z}_N$ of arbitrary size as long as N is smooth
- **Open Questions**
    - Can we do better than rate-$O(1/\lambda)$ modular arithmetic garbling even for super-polynomial size rings?
    - Can we improve the rate beyond $O(\log \lambda/\lambda)$?

Thank You