Analysis on Sliced Garbling via Algebraic Approach

How to algebraically represent garbled circuits

Taechan Kim

ASIACRYPT 2024

Table of Contents

Table of Contents

Garbled Circuit

GC is a crypto primitive for secure 2-party computation.

- Constant Round.
- Mostly, based on symmetric primitives.
- But, communication cost scales with the circuit size.

Garbled Circuit

Basic approach relies on gate-by-gate construction

- We represent functions as Boolean circuits composed by AND, XOR gates.

Garbled Circuit

Basic approach relies on gate-by-gate construction

- We represent functions as Boolean circuits composed by AND, XOR gates.
- **Garbler** generates encrypted truth tables gate-by-gate.

Garbled Circuit

Basic approach relies on gate-by-gate construction

- We represent functions as Boolean circuits composed by AND, XOR gates.
- **Garbler** generates encrypted truth tables gate-by-gate.
- **Evaluator** can decrypt only the row corresponding to the inputs to the gates.

Table of Contents

Recent Improvements & Our Contributions

Mostly, devoted to reduce **communication costs**.

	AND	XOR
Yao's GC [Yao86]	4κ	4κ
Row reduction [NPS99]	3κ	3κ
Free-XOR [KS08]	3κ	0
Half-gate [ZRE15]	2κ	0
Three-halves [RR21]	$\approx (3/2)\kappa$	0
Sliced GC [AHS24]	$\approx (4/3)\kappa$?	0

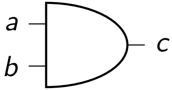
Table: κ: security parameter.

Our Contribution: We show that Sliced GC [AHS24] is insecure!

Table of Contents

Yao's GC

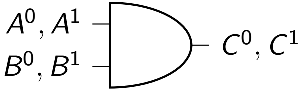
Consider a gate F with input wires a and b and output wire c.



Yao's GC

Consider a gate F with input wires a and b and output wire c.

- **Garbler** assigns κ-bit wire labels for each wires.
  - (A^0, A^1) correspond to $(0, 1)$ on the wire a.
  - Similarly for the others.



Yao's GC

Consider a gate F with input wires a and b and output wire c.

- **Basic Idea:** Garbler encrypts $C^{F(i,j)}$ using (A^i, B^j) as one-time pad keys.

a	b	c
A^0	B^0	C^0
A^0	B^1	C^0
A^1	B^0	C^0
A^1	B^1	C^1

$$\Rightarrow \quad \begin{aligned} G^{0,0} &= C^0 \oplus H(A^0, B^0) \\ G^{0,1} &= C^0 \oplus H(A^0, B^1) \\ G^{1,0} &= C^0 \oplus H(A^1, B^0) \\ G^{1,1} &= C^1 \oplus H(A^1, B^1) \end{aligned}$$

Yao's GC

Consider a gate F with input wires a and b and output wire c.

- **Basic Idea:** Given (A^1, B^0), Evaluator decrypts $G_{1,0}$ using (A^1, B^0) as one-time pad keys.

a	b	c
A^0	B^0	C^0
A^0	B^1	C^0
$\boldsymbol{A^1}$	$\boldsymbol{B^0}$	$\boldsymbol{C^0}$
A^1	B^1	C^1

$$\Rightarrow \quad \begin{aligned} G^{0,0} &= C^0 \oplus H(A^0, B^0) \\ G^{0,1} &= C^0 \oplus H(A^0, B^1) \\ \boldsymbol{G^{1,0}} &\boldsymbol{= C^0 \oplus H(A^1, B^0)} \\ G^{1,1} &= C^1 \oplus H(A^1, B^1) \end{aligned}$$

GC size – Yao's GC

The number of ctxts is 4.
Thus, the GC size is 4κ-bit.

Table of Contents

Notations

We assume the point-and-permute & Free-XOR techniques.
Garbler chooses[1]

- **(Global Offset)** $\Delta \in \{0,1\}^\kappa$ s.t. $lsb(\Delta) = 1$.
- **(Input Labels)** $A^0 \in \{0,1\}^\kappa$ corresponding to the truth value 0. Set $A^1 = A^0 + \Delta$.
    - A^u_x: where u: underlying truth value and x: least significant bit (lsb).
    - If **Evaluator** holds A^u_x, the *color bit* x is visible to **Evaluator**, where the underlying value is $u = x + \alpha$ for the *permute bit* $\alpha = lsb(A^0)$.

[1] Additions below are all over $\mathbf{F}_2$.

Garbling XOR, AND gates

Denote $u := x + \alpha$ and $v := y + \beta$.

(XOR gates) Garbling XOR gates requires no ciphertext.
Set $C := C^0 = A^0 + B^0$ corresponds to 0.
Given (A_x^u, B_y^v) for some $x, y \in \{0, 1\}$, Evaluator knows

$$C^{u+v} = C + (u + v)\Delta = A_x^u + B_y^v.$$

(AND gates) Given (A_x^u, B_y^v), Evaluator wants to know

$$C^{uv} = C + uv\Delta = C + (x + \alpha)(y + \beta)\Delta.$$

$\Rightarrow$ Recent works focus on improving garbling *AND gates*.

Table of Contents

Algebraic View on Garbling – Yao's GC

Motivated by linear-algebraic representation of GC equation [RR21], we provide our *algebraic view* on GC.

Examples (Yao's GC – Evaluator's view)

Given (A_x^u, B_y^v) for *some* $x, y \in \{0, 1\}$ and ciphertexts $G_{0,0}, \ldots, G_{1,1}$, obtain $C^{uv} = C + uv\Delta$ by computing

$$C + uv\Delta = G_{x,y} + H(A_x, B_y).$$

- Garbler's goal: set C and $G_{i,j}$'s so that the equation holds for *all* $x, y \in \mathbf{F}_2$.
- Rearrange the above equation so that C and G are on the left-hand side.

Algebraic View on Garbling – Yao's GC

Motivated by linear-algebraic representation of GC equation [RR21], we provide our *algebraic view* on GC.

Examples (Yao's GC – Garbler's view)

Given (A_x, B_y) for *all* $x, y \in \{0,1\}$, Garbler should set $G_{0,0}, \ldots, G_{1,1}$ and C such that

$$\underbrace{C + G_{x,y}}_{\text{should be determined by Garbler}} = H(A_x, B_y) + uv\Delta.$$

- Garbler determine the variables on the left-hand side from the values on the right-hand side.

Algebraic View on Garbling – Yao's GC

View the equation as a polynomial over $\mathbf{F}_2^\kappa[x,y]/(x^2+x, y^2+y)$ using the Lagrange polynomials.[2]

$$C + \underbrace{G_{x,y}}_{(x+1)(y+1)G_{0,0}+\cdots} = \underbrace{H(A_x, B_y)}_{(x+1)(y+1)H(A_0,B_0)+\cdots} + (x+\alpha)(y+\beta)\Delta$$

Observation:

- Both sides are *quadratic* polynomials.
- Garbler determines C and $G_{i,j}$ by comparing the coefficients of *four* monomials $1, x, y, xy$.
  - Ex. $C + G_{0,0} = H(A_0, B_0) + \alpha\beta\Delta$ from the constant term.
- (Row reduction) Only *four* variables are enough to determine the equation.
  $\Rightarrow$ One is for C, the other three is for ciphertexts.

[2] As $x, y \in \mathbf{F}_2$, it holds $x^2 = x$ and $y^2 = y$.

Algebraic View on Garbling – Half-gate

- Half-gate GC [ZRE15] only requires 2 ciphertexts for garbling AND gates.
- Garbler generates C and *two* ciphertexts G_1, G_2 as follows:

$$\begin{aligned} C &= H(A) + H(B) + \alpha\beta\Delta \\ G_1 &= H(A) + H(A+\Delta) + \beta\Delta \\ G_2 &= H(B) + H(B+\Delta) + A + \alpha\Delta \end{aligned}$$

- Given (A_x, B_y), Evaluator gets $C^{(x+\alpha)(y+\beta)} = C + (x+\alpha)(y+\beta)\Delta$ by computing

$$\underbrace{C + (x+\alpha)(y+\beta)\Delta}_{\text{correspond to } (x+\alpha)(y+\beta)} = \underbrace{xG_1 + yG_2 + H(A_x) + H(B_y) + yA_x}_{\text{computed by Ev}}.$$

Algebraic View on Garbling – Half-gate

How does it work?
Similarly, rearrange and rewrite the previous equation as a polynomial.

$$C + xG_1 + yG_2 = \underbrace{H(A_x)}_{(x+1)H(A_0)+xH(A_1)} + \underbrace{H(B_y)}_{\cdots} + \underbrace{yA_x}_{y(A+x\Delta)} + (x+\alpha)(y+\beta)\Delta.$$

Observation:

- $H(A_x) + H(B_y)$ is a linear polynomial.
- The quadratic term $(x+\alpha)(y+\beta)\Delta$ is inevitable if we aim to garble AND gates.
- To enforce the right-hand side to be linear, the term yA_x is introduced so that the quadratic term $xy\Delta$ cancels out.
- C and G_1, G_2 are determined by comparing the coefficients of the monomials $1, x, y$.

  $\Rightarrow$ *Two* ciphertexts are enough to garble AND gates!

Construction of GC

In a nutshell, construction of GC is to establish a suitable garbling equation, which mostly follows the direction:

1. **Determine the type** of random oracle queries on the input labels.
   - E.g. $H(A_x, B_y)$ in Yao's GC and $H(A_x) + H(B_y)$ in half-gate GC.
2. It will automatically determine **a polynomial subspace** spanned by these random oracle queries.
   - E.g. the space of quadratic/linear polynomials in Yao's GC/half-gate GC, resp.
3. **Adjust the term** $uv\Delta = (x + \alpha)(y + \beta)\Delta$ in order that it belongs to the same space.
   - E.g. Yao's GC requires no adjustment. In half-gate scheme, the term yA_x has been introduced to cancel out the quadratic term.
4. On the left-hand side, consider the same space generated with the variables C and G's, then **compare the both sides** to set the variables.

Algebraic View on Garbling – Three-halves

[RR21] further reduces GC size from 2κ to 1.5κ.

Clever Ideas: Slice wire labels into two parts, e.g. $A = (A^L \| A^R)$. Let Evaluator compute each half of the output label as[3]

$$\begin{aligned} C^L + (x+\alpha)(y+\beta)\Delta^L &= H(A_x) + H(A_x + B_y) + \cdots \\ C^R + (x+\alpha)(y+\beta)\Delta^R &= H(B_y) + H(A_x + B_y) + \cdots \end{aligned}$$

Note: With the Free-XOR,

$$A_0 + B_1 = A_1 + B_0 \text{ and } A_0 + B_0 = A_1 + B_1.$$

[3] H is now half-sized.

Algebraic View on Garbling – Three-halves

Consider the space spanned by (again using the Lagrange polynomials)

$$\begin{pmatrix} H(A_x) + H(A_x + B_y) \\ H(B_y) + H(A_x + B_y) \end{pmatrix} = \mathbf{M} \begin{pmatrix} H(A_0) \\ H(A_1) \\ H(B_0) \\ H(B_1) \\ H(A_0 + B_0) \\ H(A_0 + B_1) \end{pmatrix},$$

where

$$\mathbf{M} := \begin{pmatrix} x+1 & x & 0 & 0 & x+y+1 & x+y \\ 0 & 0 & y+1 & y & x+y+1 & x+y \end{pmatrix}.$$

Observation: $\dim(col.sp(\mathbf{M})) = 5$.
$\Rightarrow$ Two variables for C^L and C^R, and three variables for ciphertexts.
Each ciphertexts is of $\kappa/2$-bit. Thus the total GC size would be $(3/2)\kappa$.

Algebraic View on Garbling – Three-halves

Let the both sides of GC equation be in the same space as *span*(**M**). And write the GC equation (from the garbler's view) as follows:

$$\underbrace{\mathbf{W}\begin{pmatrix} C^L \\ C^R \\ G_1 \\ G_2 \\ G_3 \end{pmatrix}}_{span(\mathbf{W})=span(\mathbf{M})} = \underbrace{\mathbf{M}\begin{pmatrix} H(A_0) \\ H(A_1) \\ H(B_0) \\ H(B_1) \\ H(A_0+B_0) \\ H(A_0+B_1) \end{pmatrix}}_{\in span(\mathbf{M})} + \underbrace{\mathbf{R}_A\vec{A}_x + \mathbf{R}_B\vec{B}_y + (x+\alpha)(y+\beta)\vec{\Delta}}_{(*)}.$$

For the correctness,

- **W** is a column-reduced matrix of **M**, i.e. span(**W**) = span(**M**).
- The matrices $\mathbf{R}_A, \mathbf{R}_B$ control the term $(*)$ containing $(x+\alpha)(y+\beta)\vec{\Delta}$ belongs to *span*(**M**).

Algebraic View on Garbling – Three-halves

- We have

$$\mathbf{W} = \begin{pmatrix} 1 & 0 & x & 0 & x+y \\ 0 & 1 & 0 & y & x+y \end{pmatrix}$$

- The term $(*)$ also belongs to $span(\mathbf{M}) = span(\mathbf{W})$, i.e.

$$\mathbf{R}_A \vec{A}_x + \mathbf{R}_B \vec{B}_y + (x+\alpha)(y+\beta)\vec{\Delta} \in \text{span}(\mathbf{W}) \tag{1}$$

Completing the GC equation amounts to find $\mathbf{R}_A$ and $\mathbf{R}_B$.

Notes: With their linear-algebraic view in [RR21], they find $\mathbf{R}_A$ and $\mathbf{R}_B$ (which are 8×6 binary matrix for each) by exhaustive computer search. Our algebraic view simplifies this task.

Algebraic View on Garbling – Three-halves

Notes on span(**W**)**:** Observe that

$$\mathbf{W}\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{pmatrix} = \begin{pmatrix} z_1 + (z_3 + z_5)x + z_5 y \\ z_2 + z_5 x + (z_4 + z_5)y \end{pmatrix}.$$

We see that

- It consists of only *linear* polynomials.
- (y-coefficient on the top) = (x-coefficient on the bottom)

We shall use these relations to find a correct GC equation.

Algebraic View on Garbling – Three-halves

Let us write $\mathbf{R}_A = \mathbf{R}_{A,0} + x\mathbf{R}_{A,1} + y\mathbf{R}_{A,2}$ and similarly for $\mathbf{R}_B$.
Finding $\mathbf{R}_A$ and $\mathbf{R}_B$ satisfying Eq. (1) yields

$$\mathbf{R}_{A,1} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}, \quad \mathbf{R}_{A,2} = \begin{bmatrix} a_3 & a_4 \\ b_3 & b_4 \end{bmatrix}, \quad \mathbf{R}_{A,0} = \begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix}$$

$$\mathbf{R}_{B,1} = \begin{bmatrix} a_3 + 1 & a_4 \\ b_3 & b_4 + 1 \end{bmatrix}, \quad \mathbf{R}_{B,2} = \begin{bmatrix} b_3 & b_4 + 1 \\ e_3 & e_4 \end{bmatrix}, \quad \mathbf{R}_{B,0} = \begin{bmatrix} f_1 & f_2 \\ f_3 & f_4 \end{bmatrix},$$

where $f_1 = a_3 + b_3 + c_3 + \alpha$ and $f_2 = a_4 + b_4 + c_4 + \beta + 1$.

Note: We only need some algebra to solve $\mathbf{R}_A$ and $\mathbf{R}_B$ instead of the exhaustive computer search as in RR21.

Algebraic View on Garbling – Three-halves

Dicing Technique: Note that $\mathbf{R}_A$ and $\mathbf{R}_B$ contain information on α and β that leaks information on the private inputs!

What **Evaluator** indeed needs is the values of $\mathbf{R}_A$ and $\mathbf{R}_B$ at $(x, y) = (i, j)$ for her input (A_i, B_j).

Garbler generates *additional* ciphertexts in a way that Evaluator only obtains $\mathbf{R}_A(i, j)$ and $\mathbf{R}_B(i, j)$.
For instance, consider the first column of $\mathbf{R}_A$, the additional ciphertexts satisfy

$$\mathbf{W} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{pmatrix} = \mathbf{M}\vec{H} + \begin{pmatrix} c_1 + a_1 x + a_3 y \\ c_3 + a_3 x + b_3 y \end{pmatrix}.$$

Algebraic View on Garbling – Three-halves

Note: The dicing technique in RR21 does not leak information on the inputs:
I.e. from $\mathbf{R}_A(i,j)$ and $\mathbf{R}_B(i,j)$, Evaluator cannot infer information on α and β.

But, this is not the case for AHS24 construction.

Overview on AHS24 – Sliced Garbling

Sliced Garbling: A main feature of AHS24 is as follows:

- **(3-sliced)** It uses 3-sliced wire labels, i.e. $A = (A^1 \| A^2 \| A^3)$.
- **(target gates)** It targets garbling the 3-input gate $g(u, v, w) = u(v + w)$.
- **(oracle queries)** It uses

$$
\begin{aligned}
D^1 + g(u,v,w)\Delta^1 &= H(A_x) + H(B_y) + H(A_x + B_y + C_z) + \cdots \\
D^2 + g(u,v,w)\Delta^2 &= H(B_y) + H(C_z) + H(A_x + B_y + C_z) + \cdots \\
D^3 + g(u,v,w)\Delta^3 &= H(A_x) + H(C_z) + H(A_x + B_y + C_z) + \cdots ,
\end{aligned}
$$

where $(u, v, w) = (x + \alpha, y + \beta, z + \gamma)$.

Overview on AHS24 – Sliced Garbling

Sliced Garbling: A main feature of AHS24 is as follows:

- The $\vec{H}$ is defined as:

$$\vec{H} := \begin{pmatrix} H(A_0) & H(A_1) & H(B_0) & H(B_1) & H(C_0) & H(C_1) & H(A_0+B_0+C_0) & H(A_0+B_0+C_1) \end{pmatrix}^\top.$$

- Then the matrix $\mathbf{M}$ for AHS24 is of the form:

$$\mathbf{M} = \begin{pmatrix} x+1 & x & y+1 & y & 0 & 0 & x+y+z+1 & x+y+z \\ 0 & 0 & y+1 & y & z+1 & z & x+y+z+1 & x+y+z \\ x+1 & x & 0 & 0 & z+1 & z & x+y+z+1 & x+y+z \end{pmatrix}$$

- The column-reduced matrix $\mathbf{W}$ is of the form:

$$\mathbf{W} = \begin{pmatrix} 1 & 0 & 0 & x & y & 0 & x+y+z \\ 0 & 1 & 0 & 0 & y & z & x+y+z \\ 0 & 0 & 1 & x & 0 & z & x+y+z \end{pmatrix}$$

Overview on AHS24 – Sliced Garbling

Sliced Garbling: A main feature of AHS24 is as follows:

- Thus, $\dim(span(\mathbf{M})) = \dim(span(\mathbf{W})) = 7$.
- Among them, 4 will contribute to the ciphertexts, and each of ctxts is $\kappa/3$-bit.
- If the construction works, then its cost will be $(4/3)\kappa$-bit, smaller than $(3/2)\kappa$.
- But, AHS24 leaks information on α and β.

Table of Contents

Analysis on Sliced Garbling: Main Results

From our algebraic view, we write the GC equation for AHS24 as follows:

$$\mathbf{W}\vec{G} = \mathbf{M}\vec{H} + \underbrace{\mathbf{R}_A\vec{A}_x + \mathbf{R}_B\vec{B}_y + \mathbf{R}_C\vec{C}_z + g(x+\alpha, y+\beta, z+\gamma)\vec{\Delta}}_{\text{should belong to } span(\mathbf{M})},$$

where $\vec{G} := \left(D^1, D^2, D^3, G_1, G_2, G_3, G_4\right)^\top$.

Analysis on Sliced Garbling: Main Results

Notes on $span(\mathbf{W})$**:** Observe that

$$\mathbf{W}\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{pmatrix} = \begin{pmatrix} v_1 + x(v_4 + v_7) + y(v_5 + v_7) + z v_7 \\ v_2 + x v_7 + y(v_5 + v_7) + z(v_6 + v_7) \\ v_3 + x(v_4 + v_7) + y v_7 + z(v_6 + v_7) \end{pmatrix}$$

$$:= \vec{\nu}_0 + \vec{\nu}_1 x + \vec{\nu}_2 y + \vec{\nu}_3 z.$$

Analysis on Sliced Garbling: Main Results

More formally, we interpret this condition with linear algebra:

$$\vec{\nu} = \vec{\nu}_0 + \vec{\nu}_1 x + \vec{\nu}_2 y + \vec{\nu}_3 z \in span(\mathbf{W})$$

if and only if

$$\mathbf{P}_1\vec{\nu}_1 + \mathbf{P}_2\vec{\nu}_2 + \mathbf{P}_3\vec{\nu}_3 = 0,$$

where

$$(\mathbf{P}_1 \mid \mathbf{P}_2 \mid \mathbf{P}_3) = \left(\begin{array}{ccc|ccc|ccc} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{array}\right).$$

Analysis on Sliced Garbling: Main Results

We can provide explicit formulas for $\mathbf{R}_A, \mathbf{R}_B$ and $\mathbf{R}_C$.

$$
\begin{aligned}
\mathbf{R}_A = & \begin{pmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_0+\beta+\gamma & b_0 & c_0+\beta+\gamma \end{pmatrix} + \begin{pmatrix} a_3 & b_3 & c_3 \\ a_4 & b_4 & c_4 \\ a_3 & b_3 & c_3 \end{pmatrix} x \\
+ & \begin{pmatrix} a_4+1 & b_4 & c_4+1 \\ a_4+1 & b_4 & c_4+1 \\ a_4 & b_4 & c_4 \end{pmatrix} y + \begin{pmatrix} a_4 & b_4 & c_4 \\ a_4+1 & b_4 & c_4+1 \\ a_4+1 & b_4 & c_4+1 \end{pmatrix} z
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{R}_B = & \begin{pmatrix} d_0 & e_0 & f_0 \\ d_0+\alpha & e_0+\alpha & f_0 \\ a_1+1 & b_1+\beta+\gamma+1 & c_1+\alpha+1 \end{pmatrix} + \begin{pmatrix} a_4 & b_4 & c_4+1 \\ a_4+1 & b_4+1 & c_4+1 \\ a_4 & b_4 & c_4+1 \end{pmatrix} x \\
+ & \begin{pmatrix} d_5 & e_5 & f_5 \\ d_5 & e_5 & f_5 \\ a_4+1 & b_4+1 & c_4+1 \end{pmatrix} y + \begin{pmatrix} a_4+1 & b_4+1 & c_4+1 \\ a_4+1 & b_4+1 & c_4+1 \\ a_4+1 & b_4+1 & c_4+1 \end{pmatrix} z
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{R}_C = & \begin{pmatrix} a_1+\alpha+1 & b_1+\beta+\gamma+1 & c_1+1 \\ g_1 & h_1 & i_1 \\ g_1 & h_1+\alpha & i_1+\alpha \end{pmatrix} + \begin{pmatrix} a_4+1 & b_4 & c_4 \\ a_4+1 & b_4+1 & c_4+1 \\ a_4+1 & b_4 & c_4 \end{pmatrix} x \\
+ & \begin{pmatrix} a_4+1 & b_4+1 & c_4+1 \\ a_4+1 & b_4+1 & c_4+1 \\ a_4+1 & b_4+1 & c_4+1 \end{pmatrix} y + \begin{pmatrix} a_4+1 & b_4+1 & c_4+1 \\ g_6 & h_6 & i_6 \\ g_6 & h_6 & i_6 \end{pmatrix} z,
\end{aligned}
$$

Note: It leaks information on α, β and γ.
For instance, if Ev holds (A_0, B_0, C_0), then he will know the constant terms of each matrix. We can generalize this to arbitrary choice of inputs.

Analysis on Sliced Garbling [FLZ24]

Concurrent work: Recently, Fan, Lu, and Zhou also observed that Sliced Garbling is not secure. Their approach is based on a different methodology, and it only discusses the case when the color bits are $(0, 0, 0)$.

Thank you!
Any question?