The Whole is Greater than the Sum of its Parts: Linear Garbling and Applications

Tal Malkin[1] Valerio Pastro[1] abhi shelat[2]

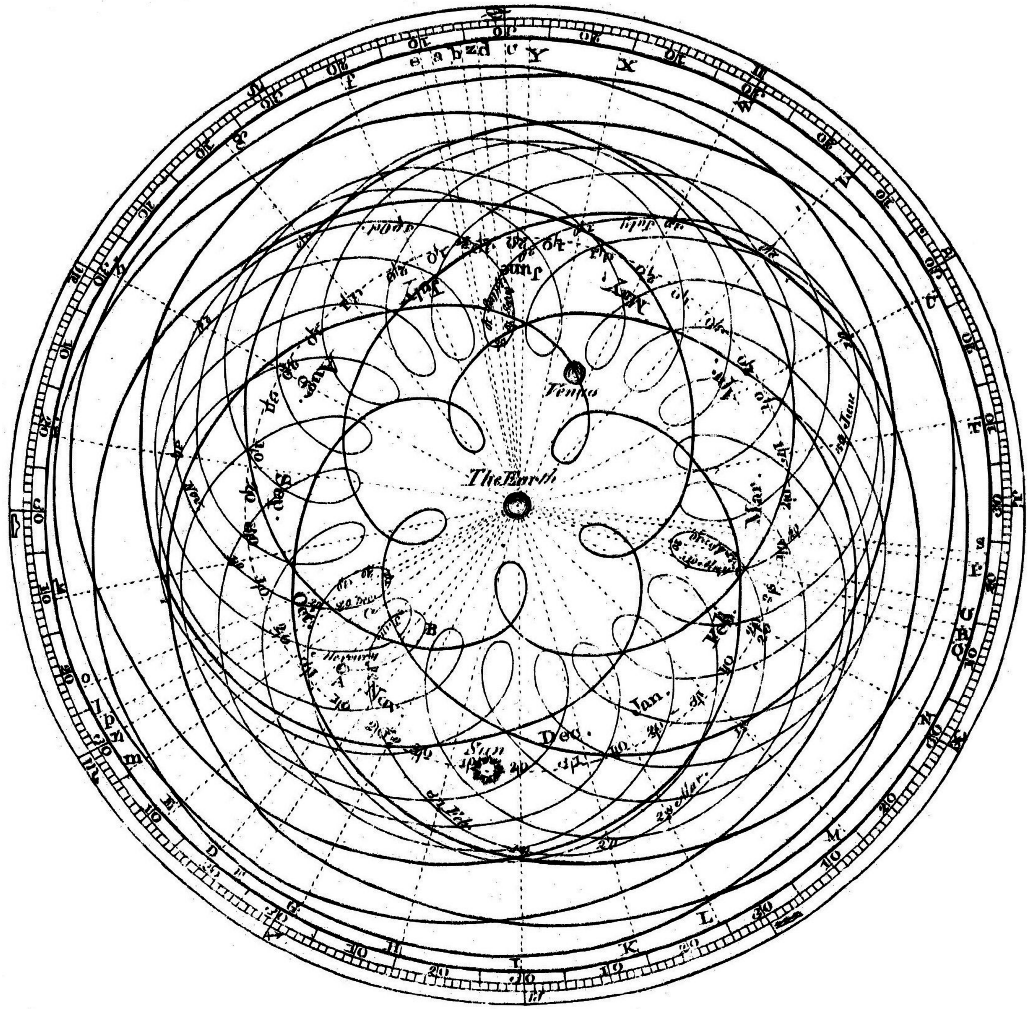
[1]Columbia University

[2]University of Virginia

June 10, 2015

Some complex system...

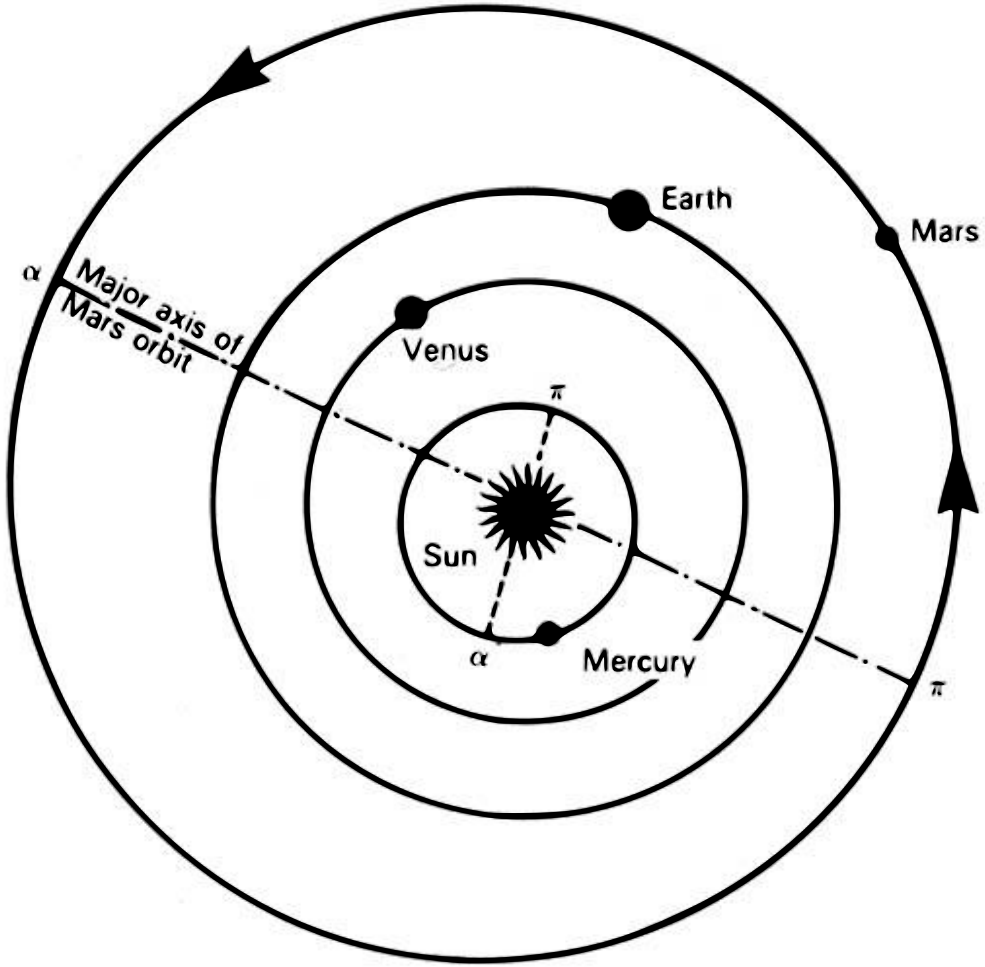
The solar system: Geocentric Model – 1400 AD



Credit: `http://en.wikipedia.org/wiki/Deferent_and_epicycle`

...can made simple, by changing perspective.

The solar system – today



Credit: http://history.nasa.gov/SP-4212/p427.html

More Context:

Our system: linear garbling

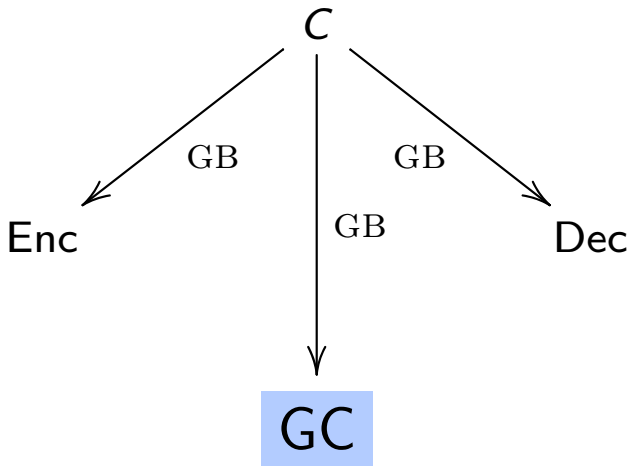
New perspective: linear garbling seen as linear secret sharing
simple properties $\Rightarrow$ simulation-based security

Why? simpler model $\Rightarrow$ more advanced schemes

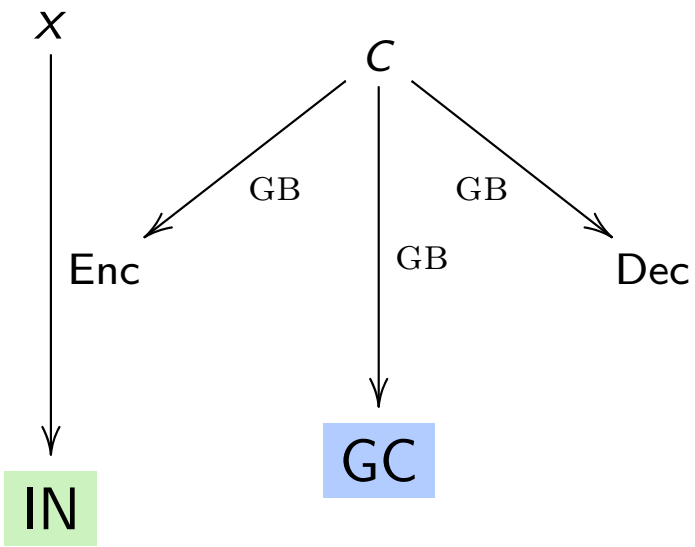
What is garbling? [BHR12]

C

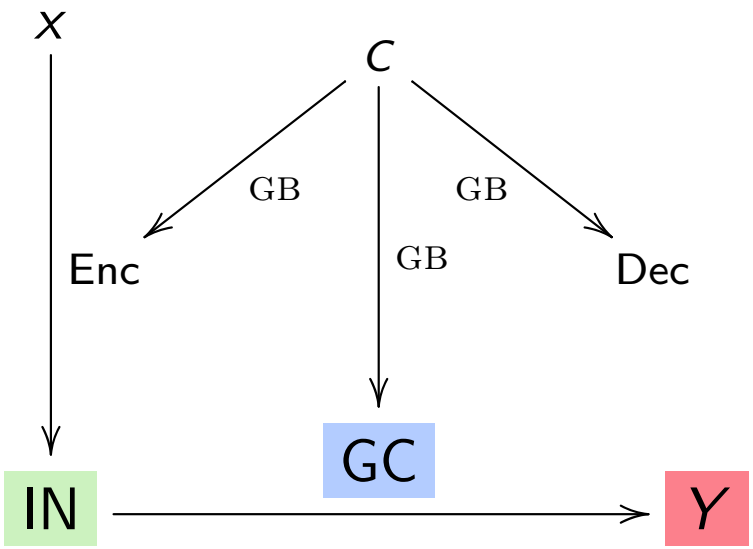
What is garbling? [BHR12]



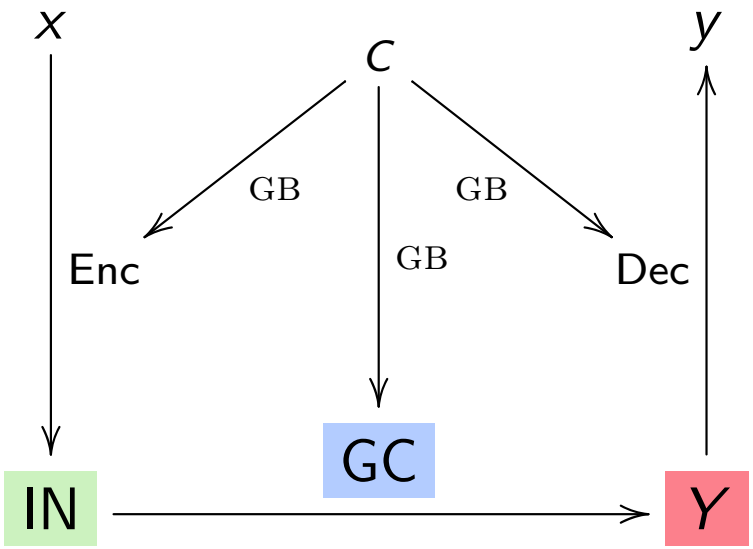
What is garbling? [BHR12]



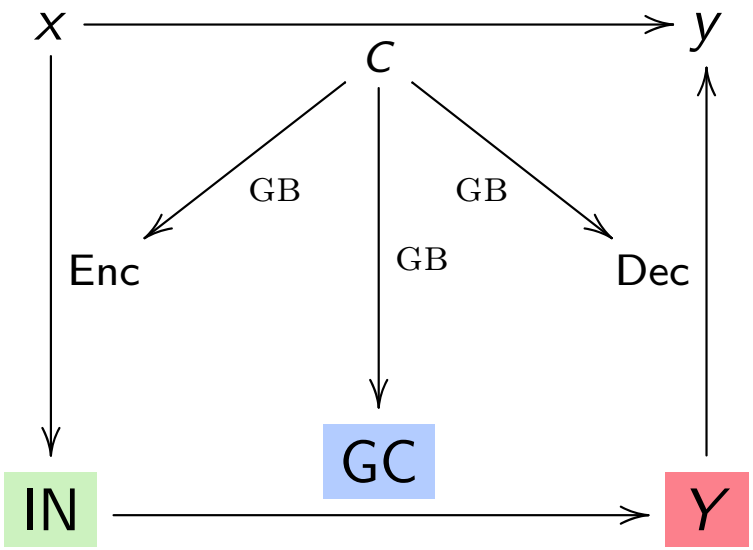
What is garbling? [BHR12]



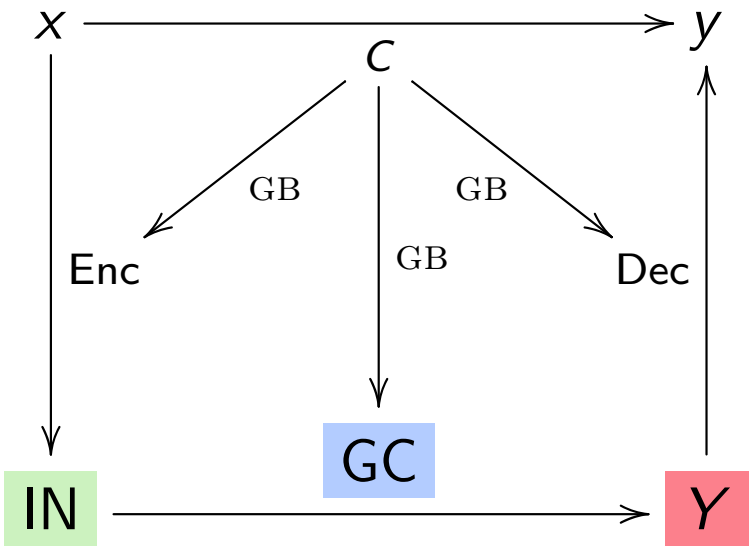
What is garbling? [BHR12]



What is garbling? [BHR12]



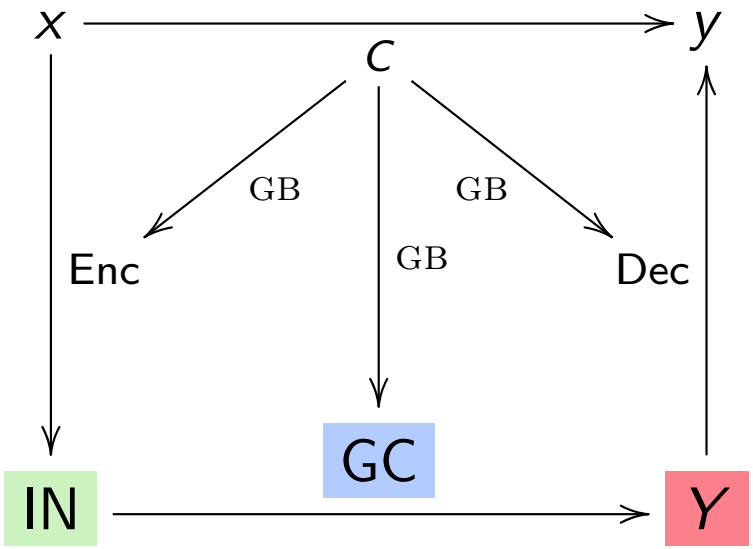
What is garbling? [BHR12]



Security:

$$\left\{\left(\boxed{\text{GC}}, \text{Enc}, \text{Dec}\right) \leftarrow \text{GB}(1^\lambda, C), \boxed{\text{IN}} \leftarrow \text{Enc}(x) : \left(\boxed{\text{GC}}, \boxed{\text{IN}}, \text{Dec}\right)\right\}_\lambda \approx_c \left\{S(1^\lambda, C, C(x))\right\}_\lambda$$

What is garbling? [BHR12]



Security:

$$\left\{\left(\boxed{\text{GC}}, \text{Enc}, \text{Dec}\right) \leftarrow \text{GB}(1^\lambda, C), \boxed{\text{IN}} \leftarrow \text{Enc}(x) : \left(\boxed{\text{GC}}, \boxed{\text{IN}}, \text{Dec}\right)\right\}_\lambda \approx_c \left\{S(1^\lambda, C, C(x))\right\}_\lambda$$

Focus on: boolean circuits, communication complexity (size of $\boxed{\text{GC}}$)

Can we do better?

	$\times\lambda$ bits	
Scheme	XOR	AND
Yao [Yao82]	4	4
GRR2 [PSSW09]	2	2
Free-XOR + GRR3 [KS08, NPS99]	0	3
FleXOR [KMR14]	2/1/0	2
Half-gates [ZRE15]	0	2

Table : Per-gate communication complexity.

Can we do better?

	$\times\lambda$ bits	
Scheme	XOR	AND
Yao [Yao82]	4	4
GRR2 [PSSW09]	2	2
Free-XOR + GRR3 [KS08, NPS99]	0	3
FleXOR [KMR14]	2/1/0	2
Half-gates [ZRE15]	0	2
[ZRE15]: any linear, gate-by-gate scheme		≥ 2

Table : Per-gate communication complexity.

How can we circumvent the lowerbound?

- linear, **not** gate-by-gate
- **not** linear, gate-by-gate

How can we circumvent the lowerbound?

- linear, **not** gate-by-gate $\Leftarrow$ this talk
- **not** linear, gate-by-gate

Approaching "**not** gate-by-gate" garbling:

- slice circuit in small "units"
- garble unit-by-unit

How can we circumvent the lowerbound?

- linear, **not** gate-by-gate $\Leftarrow$ this talk
- **not** linear, gate-by-gate

Approaching "**not** gate-by-gate" garbling:

- slice circuit in small "units"
- garble unit-by-unit

Note: if units are gates $\Rightarrow$ our scheme $=$ half-gates

How can we circumvent the lowerbound?

- linear, **not** gate-by-gate $\Leftarrow$ this talk
- **not** linear, gate-by-gate

Approaching "**not** gate-by-gate" garbling:

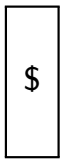
- slice circuit in small "units"
- garble unit-by-unit

Note: if units are gates $\Rightarrow$ our scheme $=$ half-gates

Large units $\Rightarrow$ hard proofs $\Rightarrow$ need for easier framework

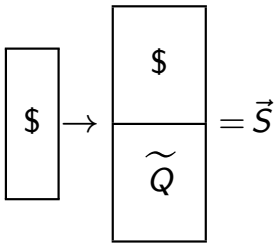
Linear garbling [ZRE15]

Intuition: garbler and evaluator: **RO calls** and **linear functions** only



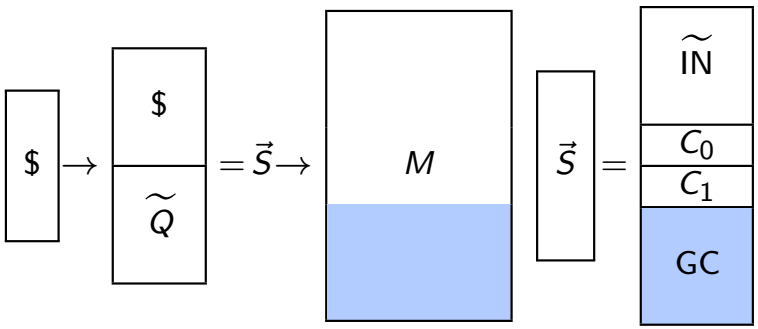
Linear garbling [ZRE15]

Intuition: garbler and evaluator: **RO calls** and **linear functions** only



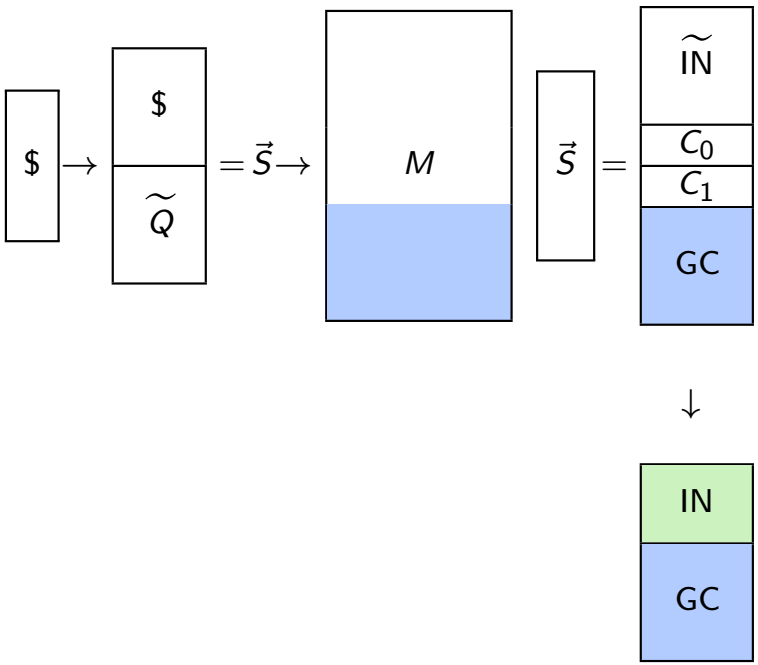
Linear garbling [ZRE15]

Intuition: garbler and evaluator: **RO calls** and **linear functions** only



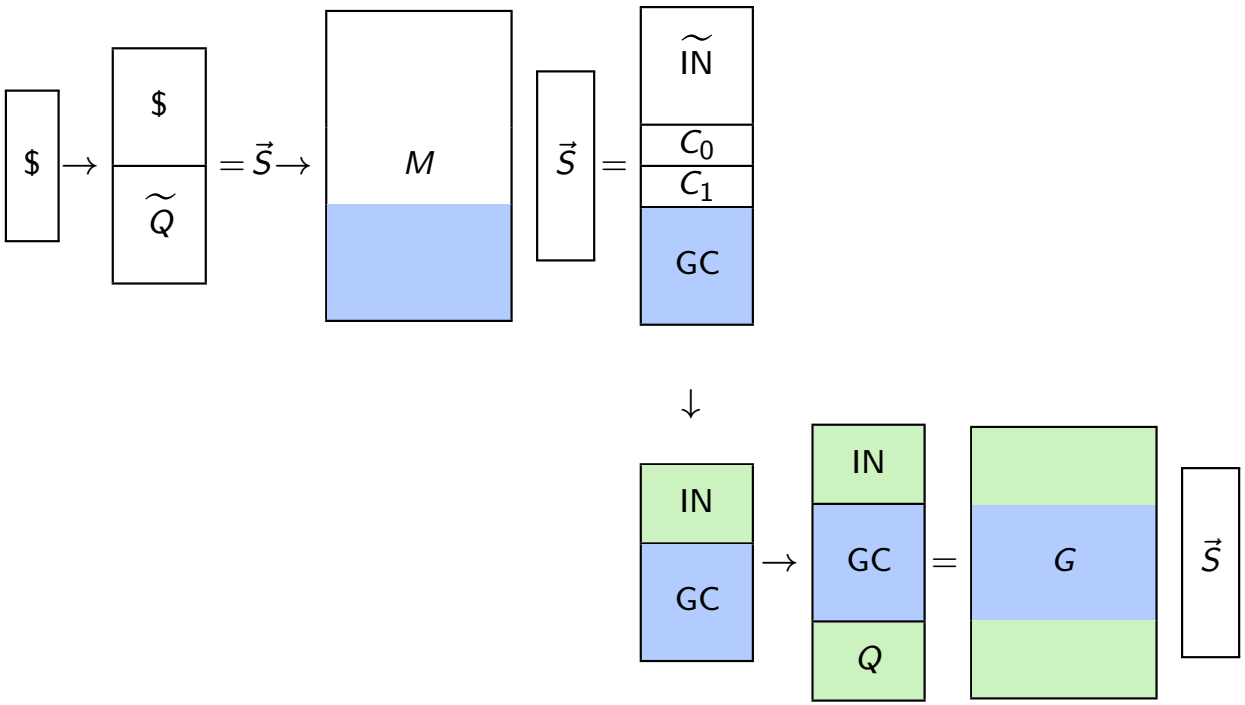
Linear garbling [ZRE15]

Intuition: garbler and evaluator: **RO calls** and **linear functions** only



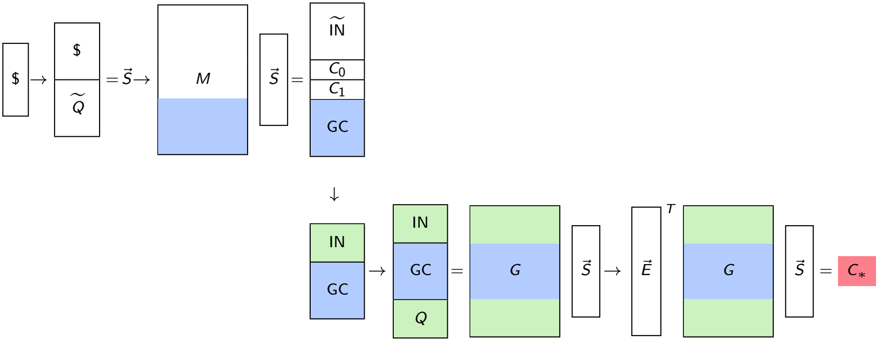
Linear garbling [ZRE15]

Intuition: garbler and evaluator: **RO calls** and **linear functions** only



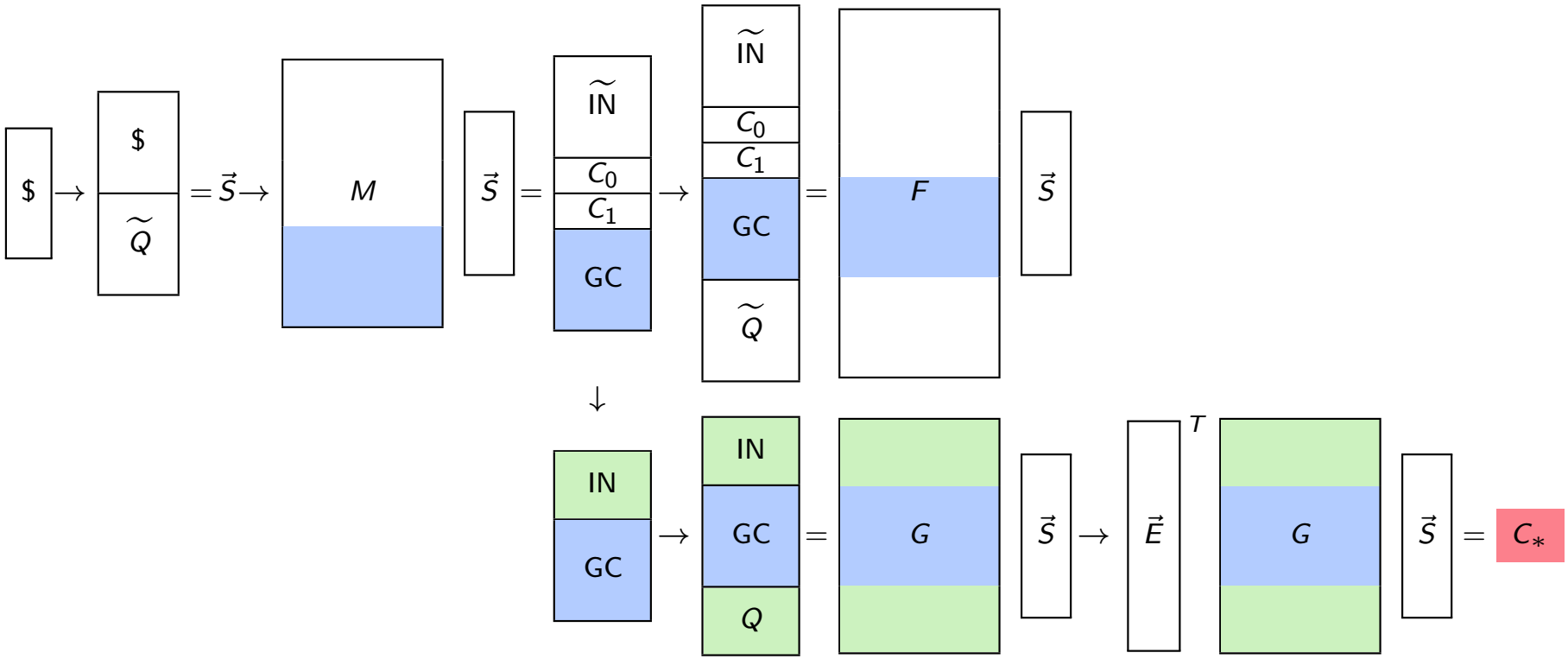
Linear garbling [ZRE15]

Intuition: garbler and evaluator: **RO calls** and **linear functions** only



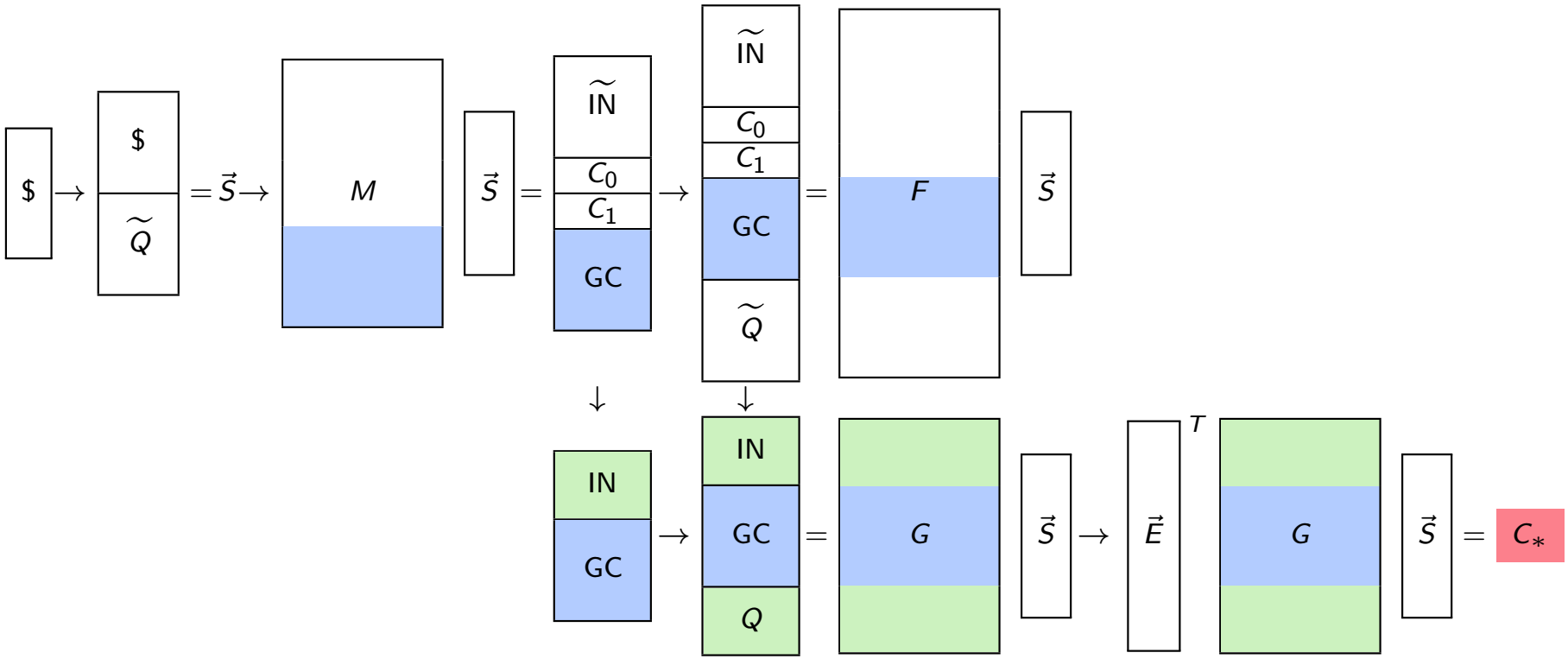
Linear garbling [ZRE15]

Intuition: garbler and evaluator: **RO calls** and **linear functions** only



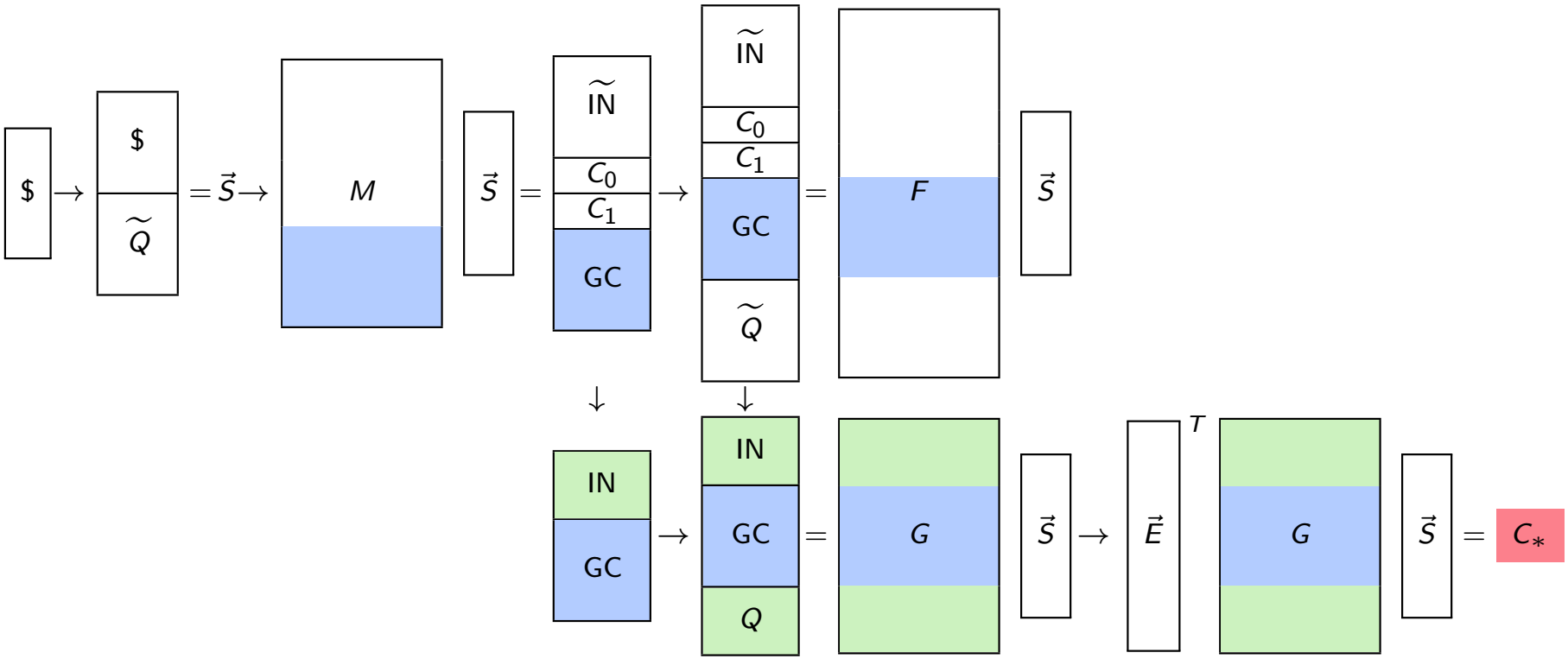
Linear garbling [ZRE15]

Intuition: garbler and evaluator: **RO calls** and **linear functions** only



Linear garbling [ZRE15]

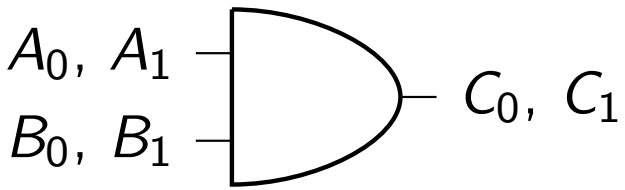
Intuition: garbler and evaluator: **RO calls** and **linear functions** only



Possible interpretation:

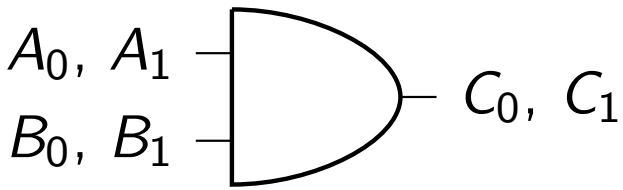
- F: secret sharing scheme for both C_0, C_1
- G: rows corresponding to shares given to evaluator

Yao Garbling – GB (M matrix)



$$
\begin{aligned}
G_{0,0} &= H(A_0 \| B_0) \oplus C_0 = \mathrm{Enc}_{A_0,B_0}(C_0) \\
G_{0,1} &= H(A_0 \| B_1) \oplus C_0 = \mathrm{Enc}_{A_0,B_1}(C_0) \\
G_{1,0} &= H(A_1 \| B_0) \oplus C_0 = \mathrm{Enc}_{A_1,B_0}(C_0) \\
G_{1,1} &= H(A_1 \| B_1) \oplus C_1 = \mathrm{Enc}_{A_1,B_1}(C_1)
\end{aligned}
$$

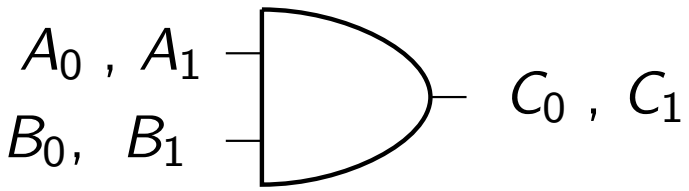
Yao Garbling – GB (M matrix)



$$
\begin{aligned}
G_{0,0} &= H(A_0 \| B_0) \oplus C_0 = \mathrm{Enc}_{A_0,B_0}(C_0) \\
G_{0,1} &= H(A_0 \| B_1) \oplus C_0 = \mathrm{Enc}_{A_0,B_1}(C_0) \\
G_{1,0} &= H(A_1 \| B_0) \oplus C_0 = \mathrm{Enc}_{A_1,B_0}(C_0) \\
G_{1,1} &= H(A_1 \| B_1) \oplus C_1 = \mathrm{Enc}_{A_1,B_1}(C_1)
\end{aligned}
$$

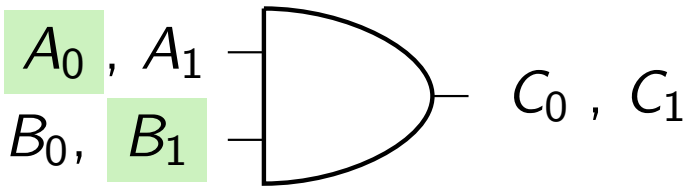
$$
\begin{pmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \\ C_0 \\ C_1 \\ G_{0,0} \\ G_{0,1} \\ G_{1,0} \\ G_{1,1} \end{pmatrix}
=
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \\ C_0 \\ C_1 \\ H(A_0 \| B_0) \\ H(A_0 \| B_1) \\ H(A_1 \| B_0) \\ H(A_1 \| B_1) \end{pmatrix}
$$

Yao Garbling – EN & EV (F, G, E matrices)



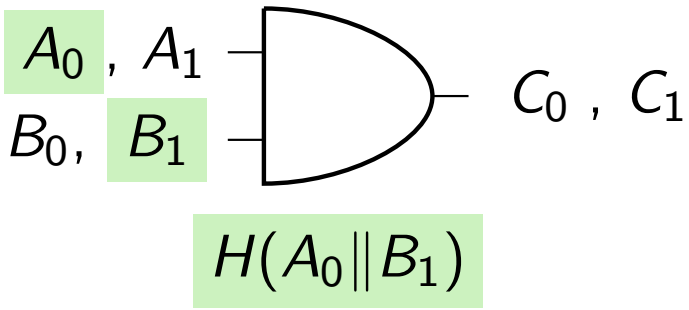
$$\begin{pmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \\ C_0 \\ C_1 \\ G_{0,0} \\ G_{0,1} \\ G_{1,0} \\ G_{1,1} \\ H(A_0\|B_0) \\ H(A_0\|B_1) \\ H(A_1\|B_0) \\ H(A_1\|B_1) \end{pmatrix} = \begin{pmatrix} 1&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0&1&0 \\ 0&0&0&0&0&1&0&0&0&1 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&1 \end{pmatrix} \begin{pmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \\ C_0 \\ C_1 \\ H(A_0\|B_0) \\ H(A_0\|B_1) \\ H(A_1\|B_0) \\ H(A_1\|B_1) \end{pmatrix}$$

Yao Garbling – EN & EV (F, G, E matrices)



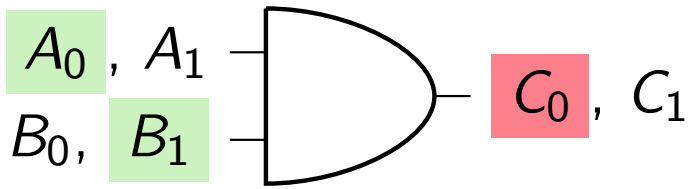
$$\begin{pmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \\ C_0 \\ C_1 \\ G_{0,0} \\ G_{0,1} \\ G_{1,0} \\ G_{1,1} \\ H(A_0\|B_0) \\ H(A_0\|B_1) \\ H(A_1\|B_0) \\ H(A_1\|B_1) \end{pmatrix} = \begin{pmatrix} 1&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0&1&0 \\ 0&0&0&0&0&1&0&0&0&1 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&1 \end{pmatrix} \begin{pmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \\ C_0 \\ C_1 \\ H(A_0\|B_0) \\ H(A_0\|B_1) \\ H(A_1\|B_0) \\ H(A_1\|B_1) \end{pmatrix}$$

Yao Garbling – EN & EV (F, G, E matrices)



$$\begin{pmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \\ C_0 \\ C_1 \\ G_{0,0} \\ G_{0,1} \\ G_{1,0} \\ G_{1,1} \\ H(A_0\|B_0) \\ H(A_0\|B_1) \\ H(A_1\|B_0) \\ H(A_1\|B_1) \end{pmatrix} = \begin{pmatrix} 1&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0&1&0 \\ 0&0&0&0&0&1&0&0&0&1 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&1 \end{pmatrix} \begin{pmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \\ C_0 \\ C_1 \\ H(A_0\|B_0) \\ H(A_0\|B_1) \\ H(A_1\|B_0) \\ H(A_1\|B_1) \end{pmatrix}$$

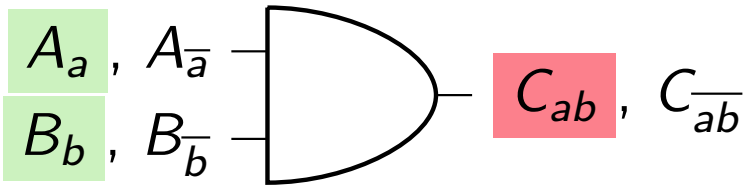
Yao Garbling – EN & EV (F, G, E matrices)



$$C_0 \leftarrow H(A_0 \| B_1) \oplus G_{0,1}$$

$$\begin{pmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \\ C_0 \\ C_1 \\ G_{0,0} \\ G_{0,1} \\ G_{1,0} \\ G_{1,1} \\ H(A_0\|B_0) \\ H(A_0\|B_1) \\ H(A_1\|B_0) \\ H(A_1\|B_1) \end{pmatrix} = \begin{pmatrix} 1&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0 \\ 0&0&0&0&1&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0&1&0 \\ 0&0&0&0&0&1&0&0&0&1 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&1 \end{pmatrix} \begin{pmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \\ C_0 \\ C_1 \\ H(A_0\|B_0) \\ H(A_0\|B_1) \\ H(A_1\|B_0) \\ H(A_1\|B_1) \end{pmatrix}$$

In general:



$$C_{ab} \leftarrow H(A_a \| B_b) \oplus G_{a,b}$$

$$\begin{pmatrix} A_a \\ B_b \\ C_{ab} \\ C_{\overline{ab}} \\ G_{0,0} \\ G_{0,1} \\ G_{1,0} \\ G_{1,1} \\ H(A_a \| B_b) \end{pmatrix} = \begin{pmatrix} \overline{a} & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \overline{b} & b & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \overline{ab} & ab & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & ab & \overline{ab} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \overline{ab} & \overline{a}b & a\overline{b} & ab \end{pmatrix} \begin{pmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \\ C_0 \\ C_1 \\ H(A_0 \| B_0) \\ H(A_0 \| B_1) \\ H(A_1 \| B_0) \\ H(A_1 \| B_1) \end{pmatrix}$$

A Different Interpretation of Correctness/Security

$$\begin{pmatrix} A_a \\ B_b \\ C_{ab} \\ C_{\overline{ab}} \\ G_{0,0} \\ G_{0,1} \\ G_{1,0} \\ G_{1,1} \\ H(A_a \| B_b) \end{pmatrix} = \begin{pmatrix} \overline{a} & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \overline{b} & b & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \overline{ab} & ab & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & ab & \overline{ab} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \overline{a}\overline{b} & \overline{a}b & a\overline{b} & ab \end{pmatrix} \begin{pmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \\ C_0 \\ C_1 \\ H(A_0 \| B_0) \\ H(A_0 \| B_1) \\ H(A_1 \| B_0) \\ H(A_1 \| B_1) \end{pmatrix}$$

■ $\in$ Span(■ $\cup$ ■): linear reconstruction

A Different Interpretation of Correctness/Security

$$\begin{pmatrix} A_a \\ B_b \\ C_{ab} \\ C_{\overline{ab}} \\ G_{0,0} \\ G_{0,1} \\ G_{1,0} \\ G_{1,1} \\ H(A_a \| B_b) \end{pmatrix} = \begin{pmatrix} \overline{a} & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \overline{b} & b & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \overline{ab} & ab & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & ab & \overline{ab} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \overline{a}\overline{b} & \overline{a}b & a\overline{b} & ab \end{pmatrix} \begin{pmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \\ C_0 \\ C_1 \\ H(A_0 \| B_0) \\ H(A_0 \| B_1) \\ H(A_1 \| B_0) \\ H(A_1 \| B_1) \end{pmatrix}$$

$\in$ Span($\cup$): linear reconstruction

$\notin$ Span($\cup$): linear privacy

A Different Interpretation of Correctness/Security

$$\begin{pmatrix} A_a \\ B_b \\ C_{ab} \\ C_{\overline{ab}} \\ G_{0,0} \\ G_{0,1} \\ G_{1,0} \\ G_{1,1} \\ H(A_a \| B_b) \end{pmatrix} = \begin{pmatrix} \overline{a} & a & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \overline{b} & b & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \overline{ab} & ab & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & ab & \overline{ab} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \overline{a}\overline{b} & \overline{a}b & a\overline{b} & ab \end{pmatrix} \begin{pmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \\ C_0 \\ C_1 \\ H(A_0 \| B_0) \\ H(A_0 \| B_1) \\ H(A_1 \| B_0) \\ H(A_1 \| B_1) \end{pmatrix}$$

$\in$ Span($\cup$): linear reconstruction

$\notin$ Span($\cup$): linear privacy

Theorem

Linear reconstruction & linear privacy $\Rightarrow$ simulation-based security

Warm up

Half-gate technique [ZRE15]:

$$\underbrace{v_A}_{\substack{\text{color bit,}\\ \text{known by evaluator}}} = \overbrace{a}^{\text{input}} + \underbrace{p_A}_{\substack{\text{permutation bit,}\\ \text{known by garbler}}}$$

Warm up

Half-gate technique [ZRE15]: $\underbrace{v_A}_{\substack{\text{color bit,}\\ \text{known by evaluator}}} = \overbrace{a}^{\text{input}} + \underbrace{p_A}_{\substack{\text{permutation bit,}\\ \text{known by garbler}}}$

$$\left(\begin{array}{ccccccc}
1 & 0 & v_A & 0 & 0 & 0 & 0 \\
0 & 1 & v_B & 0 & 0 & 0 & 0 \\
0 & 0 & ab + p_A p_B & 1 & 1 & 0 & 0 \\
0 & 0 & 1 + ab + p_A p_B & 1 & 1 & 0 & 0 \\
0 & 0 & p_B & 1 & 0 & 1 & 0 \\
1 & 0 & p_A & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 + v_A & 0 & v_A & 0 \\
0 & 0 & 0 & 0 & 1 + v_B & 0 & v_B
\end{array}\right)$$

Warm up

Half-gate technique [ZRE15]: $\underbrace{v_A}_{\substack{\text{color bit,}\\ \text{known by evaluator}}} = \overbrace{a}^{\text{input}} + \underbrace{p_A}_{\substack{\text{permutation bit,}\\ \text{known by garbler}}}$

$$\begin{pmatrix} \varnothing \\ \varnothing \end{pmatrix}^T \begin{pmatrix} 1 & 0 & v_A & 0 & 0 & 0 & 0 \\ 0 & 1 & v_B & 0 & 0 & 0 & 0 \\ 0 & 0 & ab + p_A p_B & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 + ab + p_A p_B & 1 & 1 & 0 & 0 \\ 0 & 0 & p_B & 1 & 0 & 1 & 0 \\ 1 & 0 & p_A & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 + v_A & 0 & v_A & 0 \\ 0 & 0 & 0 & 0 & 1 + v_B & 0 & v_B \end{pmatrix} =$$

Warm up

Half-gate technique [ZRE15]: $\underbrace{v_A}_{\substack{\text{color bit,}\\ \text{known by evaluator}}} = \overbrace{a}^{\text{input}} + \underbrace{p_A}_{\substack{\text{permutation bit,}\\ \text{known by garbler}}}$

$$\begin{pmatrix} 0 \\ \varnothing \\ \varnothing \end{pmatrix}^T \begin{pmatrix} 1 & 0 & v_A & 0 & 0 & 0 & 0 \\ 0 & 1 & v_B & 0 & 0 & 0 & 0 \\ 0 & 0 & ab + p_A p_B & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 + ab + p_A p_B & 1 & 1 & 0 & 0 \\ 0 & 0 & p_B & 1 & 0 & 1 & 0 \\ 1 & 0 & p_A & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 + v_A & 0 & v_A & 0 \\ 0 & 0 & 0 & 0 & 1 + v_B & 0 & v_B \end{pmatrix} =$$

$$\begin{matrix} 0 & ? \end{matrix}$$

$? = 0$

Warm up

Half-gate technique [ZRE15]: $\underbrace{v_A}_{\substack{\text{color bit,}\\ \text{known by evaluator}}} = \overbrace{a}^{\text{input}} + \underbrace{p_A}_{\substack{\text{permutation bit,}\\ \text{known by garbler}}}$

$$\begin{pmatrix} \\ 0 \\ \varnothing \\ \varnothing \\ v_A \\ \\ 1 \\ \\ \end{pmatrix}^T \begin{pmatrix} 1 & 0 & v_A & 0 & 0 & 0 & 0 \\ 0 & 1 & v_B & 0 & 0 & 0 & 0 \\ 0 & 0 & ab + p_A p_B & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 + ab + p_A p_B & 1 & 1 & 0 & 0 \\ 0 & 0 & p_B & 1 & 0 & 1 & 0 \\ 1 & 0 & p_A & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 + v_A & 0 & v_A & 0 \\ 0 & 0 & 0 & 0 & 1 + v_B & 0 & v_B \\ \\ & 0 & ? & 1 & & 0 & \end{pmatrix} =$$

$$? = v_A p_B$$

Warm up

Half-gate technique [ZRE15]: $\underbrace{v_A}_{\substack{\text{color bit,}\\ \text{known by evaluator}}} = \overbrace{a}^{\text{input}} + \underbrace{p_A}_{\substack{\text{permutation bit,}\\ \text{known by garbler}}}$

$$\begin{pmatrix} \\ 0 \\ \varnothing \\ \varnothing \\ v_A \\ v_B \\ 1 \\ 1 \end{pmatrix}^T \begin{pmatrix} 1 & 0 & v_A & 0 & 0 & 0 & 0 \\ 0 & 1 & v_B & 0 & 0 & 0 & 0 \\ 0 & 0 & ab + p_A p_B & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 + ab + p_A p_B & 1 & 1 & 0 & 0 \\ 0 & 0 & p_B & 1 & 0 & 1 & 0 \\ 1 & 0 & p_A & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 + v_A & 0 & v_A & 0 \\ 0 & 0 & 0 & 0 & 1 + v_B & 0 & v_B \end{pmatrix} =$$

v_B	0	?	1	1	0	0

$$? = v_A p_B + v_B p_A$$

Warm up

Half-gate technique [ZRE15]: $\underbrace{v_A}_{\substack{\text{color bit,}\\ \text{known by evaluator}}} = \overbrace{a}^{\text{input}} + \underbrace{p_A}_{\substack{\text{permutation bit,}\\ \text{known by garbler}}}$

$$\begin{pmatrix} v_B \\ 0 \\ \varnothing \\ \varnothing \\ v_A \\ v_B \\ 1 \\ 1 \end{pmatrix}^T \begin{pmatrix} 1 & 0 & v_A & 0 & 0 & 0 & 0 \\ 0 & 1 & v_B & 0 & 0 & 0 & 0 \\ 0 & 0 & ab + p_A p_B & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 + ab + p_A p_B & 1 & 1 & 0 & 0 \\ 0 & 0 & p_B & 1 & 0 & 1 & 0 \\ 1 & 0 & p_A & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 + v_A & 0 & v_A & 0 \\ 0 & 0 & 0 & 0 & 1 + v_B & 0 & v_B \end{pmatrix} =$$

$$\begin{array}{ccccccc} 0 & 0 & ? & 1 & 1 & 0 & 0 \end{array}$$

$$\begin{aligned} ? &= v_A p_B + v_B p_A + v_A v_B \\ &= (a + p_A) p_B + (b + p_B) p_A + (a + p_A)(b + p_B) \\ &= (a + p_A) b + (b + p_B) p_A \\ &= ab + p_A p_B \end{aligned}$$

QED

Our Scheme

Observation: [ZRE15] obtains $ab + p_A p_B$ in a clever way:

1. reveal one time pads (additive secret shares) of inputs ($v_A = a + p_A$)
2. reconstruct $ab + p_A p_B$ linearly in p_A, p_B

Very similar to Beaver's technique to compute MULT gates [Bea91].

This can be extended:
one product $\Rightarrow$ any polynomial of degree d in n variables

Example

$$f(a,b,c,d) = ab + ac + ad + bc + bd + cd$$

$$\left(\begin{array}{cccc|c|cccccccc}
1 & 0 & 0 & 0 & v_A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & v_B & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & v_C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & v_D & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & f(a,b,c,d) + f(p_A,p_B,p_C,p_D) & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 + f(a,b,c,d) + f(p_A,p_B,p_C,p_D) & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & p_B + p_C + p_D & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & p_C + p_D & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & p_D & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & \overline{v_A} & v_A & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \overline{v_B} & v_B & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \overline{v_C} & v_C & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \overline{v_D} & v_D
\end{array}\right)$$

Generalized half-gates

Theorem

Our scheme garbles any quadratic polynomial in n variables using n λ-bits.

Earlier example,

$$f(a, b, c, d) = ab + ac + ad + bc + bd + cd$$

can be garbled using 4 λ-bit strings.

Comparison 1

Trivial circuit C_1 for f:

$$ab + ac + ad + bc + bd + cd$$

6 AND gates $\Rightarrow$ 12 λ-bit strings required by [ZRE15] on C_1

Generalized half-gates

Theorem

Our scheme garbles any quadratic polynomial in n variables using n λ-bits.

Earlier example,

$$f(a, b, c, d) = ab + ac + ad + bc + bd + cd$$

can be garbled using 4 λ-bit strings.

Comparison 2

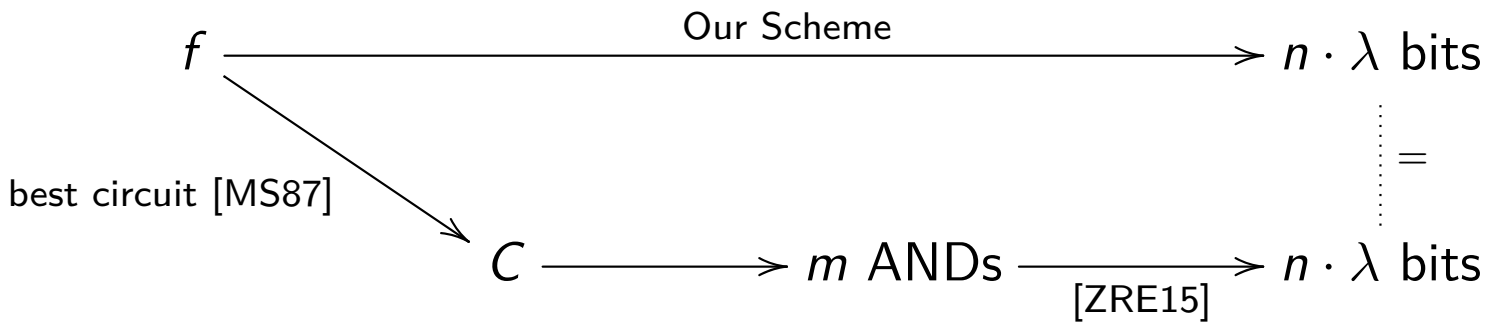
Best circuit C_2 for f:

$$(a + b + c)(a + d) + bc + a$$

2 AND gates $\Rightarrow$ 4 λ-bit strings required by [ZRE15] on C_2

Generalized half-gates

For quadratic polynomial f over $n = 2m$ variables:

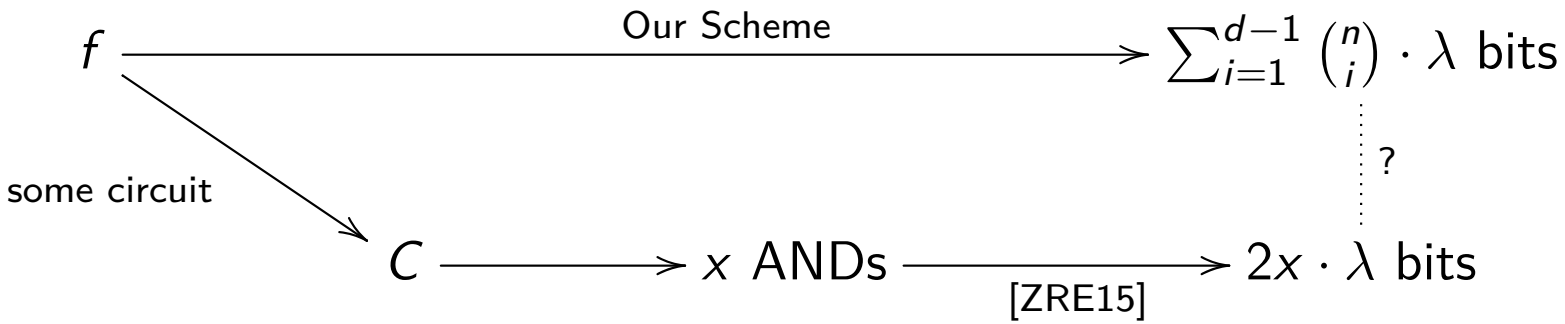


Generalized half-gates

Theorem

We garble any polynomial of degree d in n variables using $\sum_{i=1}^{d-1} \binom{n}{i}$ λ-bits.

In general? (f = degree d polynomial over n variables)



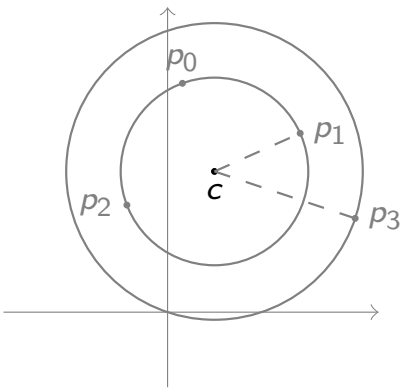
Random constant-degree d polynomial over n variables $\Rightarrow$
$\Rightarrow$ better communication complexity than [ZRE15], but
comparisons depend on C... generally hard to determine AND complexity

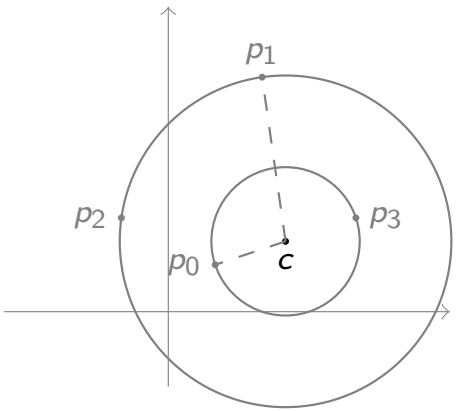
Summary, Sneek Peak, and Extras

- New framework: simple span properties $\Rightarrow$ sim-based security
- New boolean garbling scheme (proof in the above framework)
  - not gate-by-gate, garbles polynomials rather than circuits
  - can circumvent comm. complexity lowerbound for linear garbling
  - **calls to RO in each unit performed parallel (1 vs d)**
- New arithmetic garbling scheme (for small finite fields)
- Similar technique to improve Beaver-based MPC
- Non-linear garbling?

Summary, Sneek Peak, and Extras

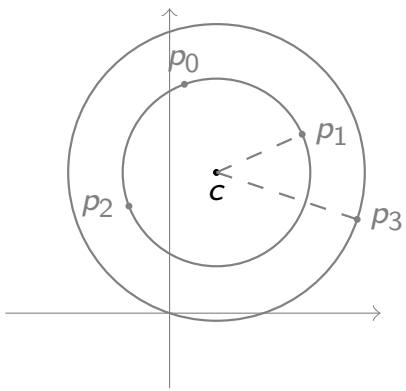
- New framework: simple span properties $\Rightarrow$ sim-based security
- New boolean garbling scheme (proof in the above framework)
  - not gate-by-gate, garbles polynomials rather than circuits
  - can circumvent comm. complexity lowerbound for linear garbling
  - **calls to RO in each unit performed parallel (1 vs d)**
- New arithmetic garbling scheme (for small finite fields)
- Similar technique to improve Beaver-based MPC
- Non-linear garbling?



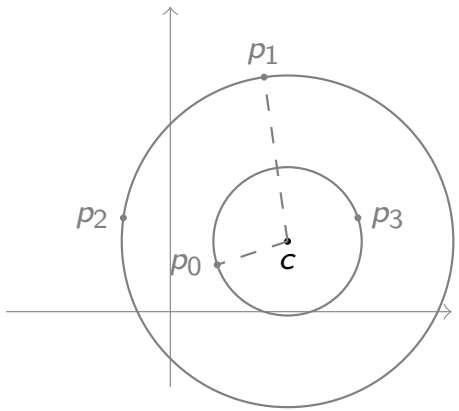


Summary, Sneek Peak, and Extras

- New framework: simple span properties $\Rightarrow$ sim-based security
- New boolean garbling scheme (proof in the above framework)
  - not gate-by-gate, garbles polynomials rather than circuits
  - can circumvent comm. complexity lowerbound for linear garbling
  - **calls to RO in each unit performed parallel (1 vs d)**
- New arithmetic garbling scheme (for small finite fields)
- Similar technique to improve Beaver-based MPC
- Non-linear garbling?



THANKS!



Donald Beaver.

Efficient multiparty protocols using circuit randomization.
In Joan Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 420–432. Springer, 1991.

Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway.

Foundations of garbled circuits.
In Ting Yu, George Danezis, and Virgil D. Gligor, editors, *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012*, pages 784–796. ACM, 2012.

Vladimir Kolesnikov, Payman Mohassel, and Mike Rosulek.

Flexor: Flexible garbling for XOR gates that beats free-xor.
In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, volume 8617 of *Lecture Notes in Computer Science*, pages 440–457. Springer, 2014.

Vladimir Kolesnikov and Thomas Schneider.

Improved garbled circuit: Free XOR gates and applications.
In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, volume 5126 of *Lecture Notes in Computer Science*, pages 486–498. Springer, 2008.

Roland Mirwald and Claus-Peter Schnorr.

The multiplicative complexity of quadratic boolean forms.
In *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987*, pages 141–150. IEEE Computer Society, 1987.

Moni Naor, Benny Pinkas, and Reuban Sumner.

Privacy preserving auctions and mechanism design.
In *EC*, pages 129–139, 1999.

Benny Pinkas, Thomas Schneider, Nigel P. Smart, and Stephen C. Williams.

Secure two-party computation is practical.

In Mitsuru Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, volume 5912 of *Lecture Notes in Computer Science*, pages 250–267. Springer, 2009.

Andrew Chi-Chih Yao.

Protocols for secure computations (extended abstract).
In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 160–164. IEEE Computer Society, 1982.

Samee Zahur, Mike Rosulek, and David Evans.

Two halves make a whole - reducing data transfer in garbled circuits using half gates.
In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 220–250. Springer, 2015.