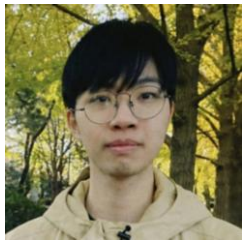
Efficient Garbled PRFs and Lookup Tables from Minimal Assumption

Wei-Kai Lin



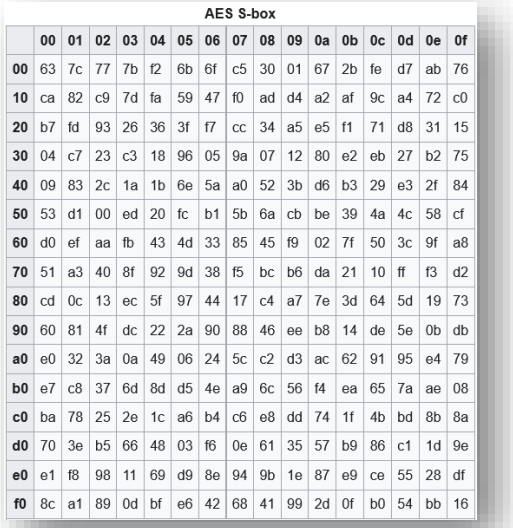
Zhenghao Lu

Hong-Sheng Zhou

Lookup Tables: Everywhere

Retrieving a value from memory is often faster than carrying out an "expensive" computation. [Wiki]

AES:
“Substitution-box”



AES S-box

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Multiplication Table

1 × 1 = 1	1 × 2 = 2
2 × 1 = 2	2 × 2 = 4
3 × 1 = 3	3 × 2 = 6
4 × 1 = 4	4 × 2 = 8
5 × 1 = 5	5 × 2 = 10
6 × 1 = 6	6 × 2 = 12
7 × 1 = 7	7 × 2 = 14
8 × 1 = 8	8 × 2 = 16
9 × 1 = 9	9 × 2 = 18
10 × 1 = 10	10 × 2 = 20
11 × 1 = 11	11 × 2 = 22
12 × 1 = 12	12 × 2 = 24

This talk: Table is N-bit string f, lookup by $f(x)$,
$x \in [N]$, $|x| = \log N = n$ bits

$f(0), f(1), \quad \ldots \quad f(x), \quad \ldots$

$f(x): \{0,1\}^n \to \{0,1\}$
$x \in [N = 2^n]$, $|x| = n$ bits

Garbled Lookup Tables

Garble functionality f:
$(\tilde{f}, Lb_{out}) \leftarrow \text{Garb}(f, Lb_{in})$

Garble input x:
$\tilde{x} := Lb_{in}(x)$

Garbler

$\tilde{f}$ →

$\tilde{x}$ →

Evaluate $f(x)$:
$\tilde{y} \leftarrow \text{Eval}(\tilde{f}, \tilde{x})$

Evaluator

Correctness: $\tilde{y} = Lb_{out}(f(x))$

Security: $(\tilde{f}, \tilde{x}, Lb_{out})$ only reveals $f(x)$

$Lb_{in} =$ $\begin{matrix} \tilde{x}_0^0 \\ \tilde{x}_0^1 \end{matrix}$ $\begin{matrix} \tilde{x}_1^0 \\ \tilde{x}_1^1 \end{matrix}$ $\begin{matrix} \tilde{x}_2^0 \\ \tilde{x}_2^1 \end{matrix}$

$Lb_{out} =$ $\begin{matrix} \tilde{y}_0^0 \\ \tilde{y}_0^1 \end{matrix}$

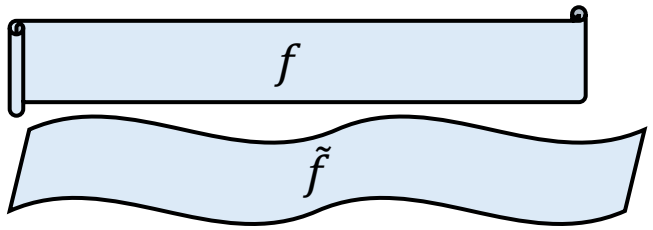
Efficiency metric: Communication, $\tilde{f}$ and $\tilde{x}$, in bits.

$f(x): \{0,1\}^n \to \{0,1\}$
$x \in [N = 2^n]$, $|x| = n$ bits

Previous: As good as sending table

Naive way, based on PRF: communication $|\tilde{f}| = O(N \cdot \kappa)$
By correctness, $|\tilde{f}| \geq |f| = N$, baseline.

[Heath-Kolesnikov-Ng'24]:
based on Random Oracle or Circular Correlation Robustness Hash (RO/CCRH)
communication $|\tilde{f}| = N + (2n - 1) \cdot \kappa$
Huge because $\kappa > 100$



Main question: Get the best of both worlds?

Our Result: Min Assumption and Communication

	Assump	Commu
Naïve	PRF	$O(N \cdot \kappa)$
[HKN24]	RO/CCRH	$N + (2n - 1) \cdot \kappa$
This work	PRF	$N + (5n + 9) \cdot \kappa + n$

All preserve composability, all takes $O(N \cdot \kappa)$ computation

This work:
- Based on [HKN24]
- Also construct Garbled PRF for small-domain

Framework:

Garbled PRF ⇒ GLUT

[HKN24]

Define function $f'(z) \coloneqq f(z \oplus \boldsymbol{a}) \oplus PRF(z)$, for random $\boldsymbol{a}$ and PRF

Let $z = x \oplus \boldsymbol{a}$:

$$f(x) = f'(x \oplus \boldsymbol{a}) \oplus PRF(x \oplus \boldsymbol{a})$$

Garble f'
Reveal f' and $x \oplus \boldsymbol{a}$ to evaluator

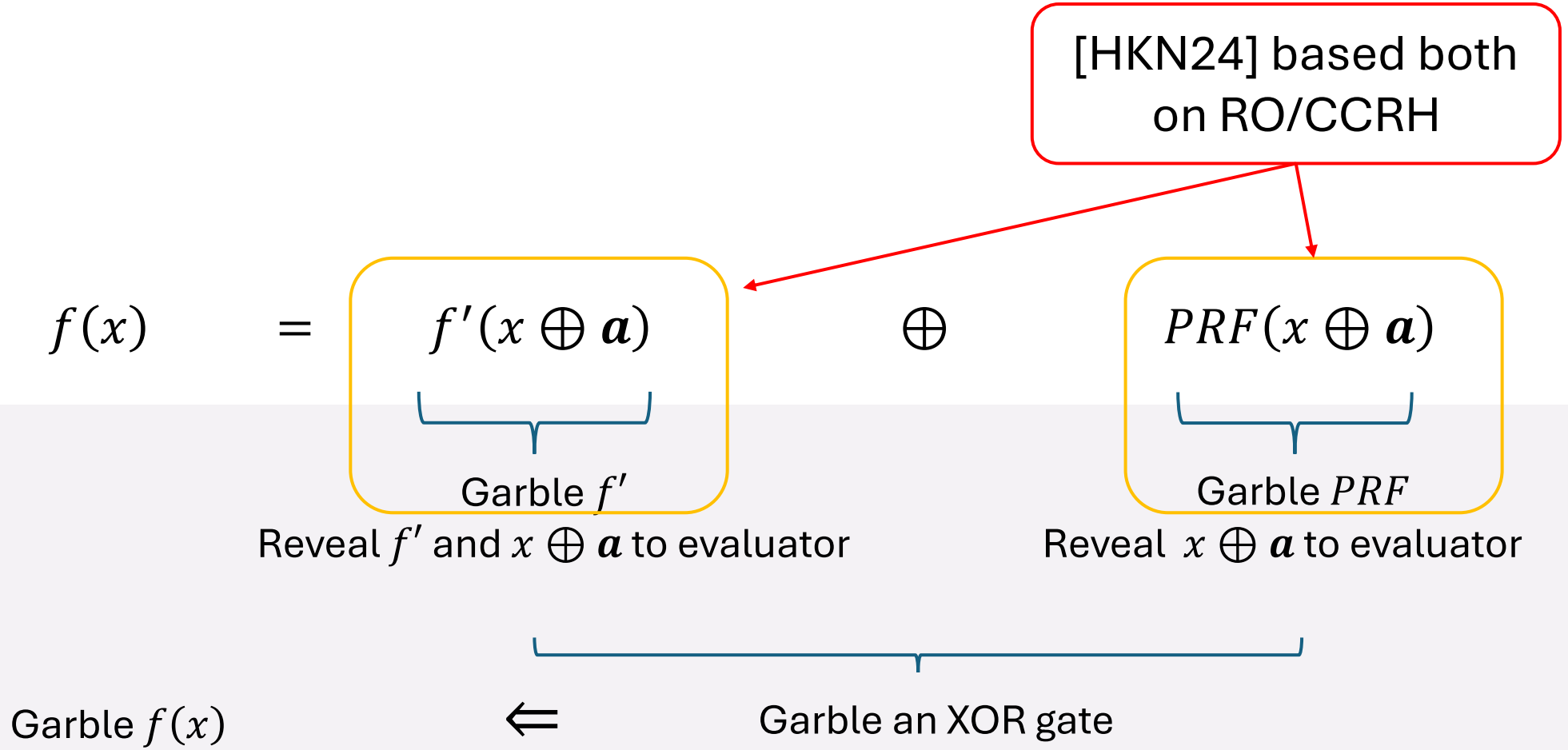
Garble PRF
Reveal $x \oplus \boldsymbol{a}$ to evaluator

Garble $f(x)$ ⇐ Garble an XOR gate

Framework:

Garbled PRF ⇒ GLUT

[HKN24]



Our technical observation to break circularity and correlation:
sample independent wire labels &
introduce fresh random seeds

RO/CCRH

This talk

$f(x) = f'(x \oplus \boldsymbol{a}) \oplus PRF(x \oplus \boldsymbol{a})$

Garble f'

Reveal f' and $x \oplus \boldsymbol{a}$ to evaluator

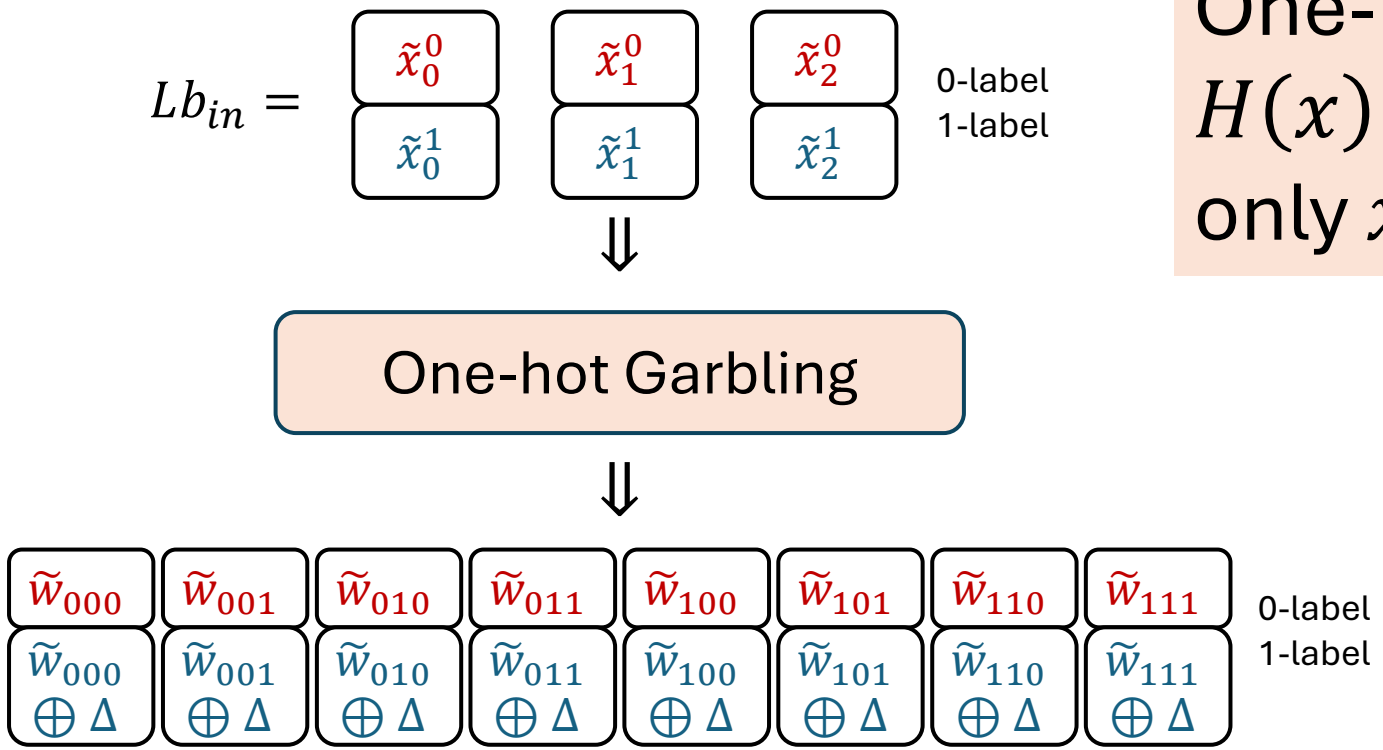
Garble PRF

Reveal $x \oplus \boldsymbol{a}$ to evaluator

Garble $f(x)$ $\Leftarrow$ Garble an XOR gate

Garble $f(x)$, f and x are revealed From One-hot Garbling

[Heath-Kolesnikov'21]

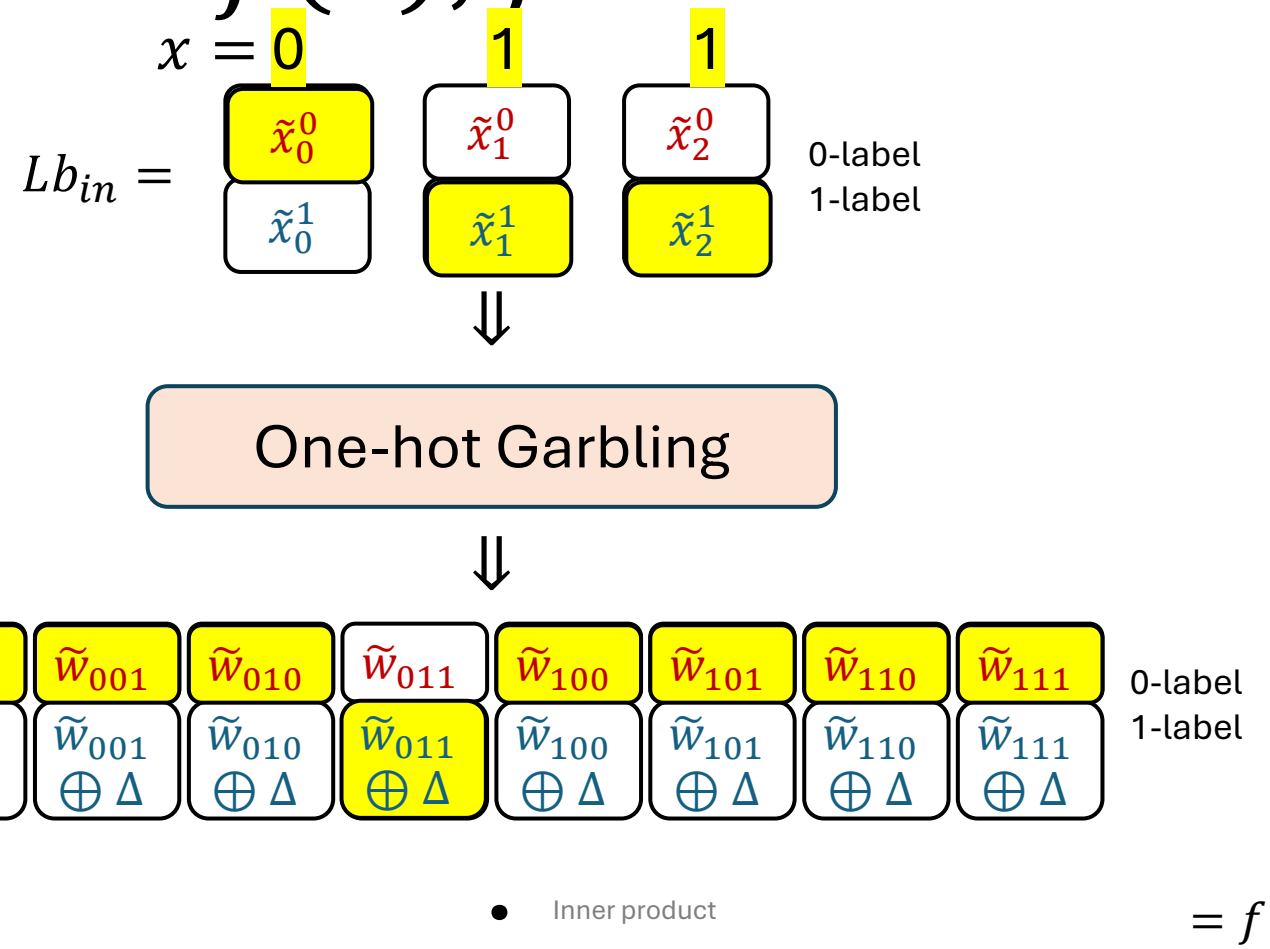


One-hot encoding of x:
$H(x) = (0,0,\dots,0,1,0,0,\dots,0)$,
only x-th position is 1 .

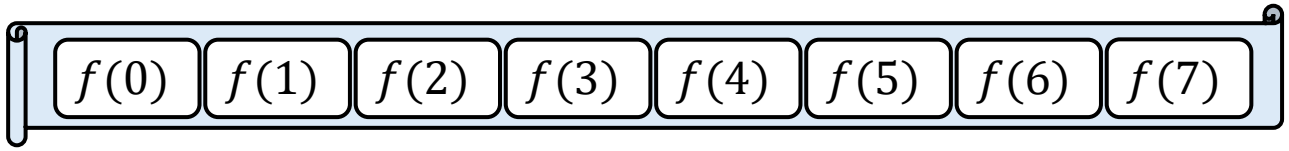
Garble $f(x)$, f and x are revealed

[Heath-Kolesnikov'21]

One-hot encoding of x:
$H(x) = (0,0,\dots,0,1,0,0,\dots,0)$,
only x-th position is 1 .



- Inner product

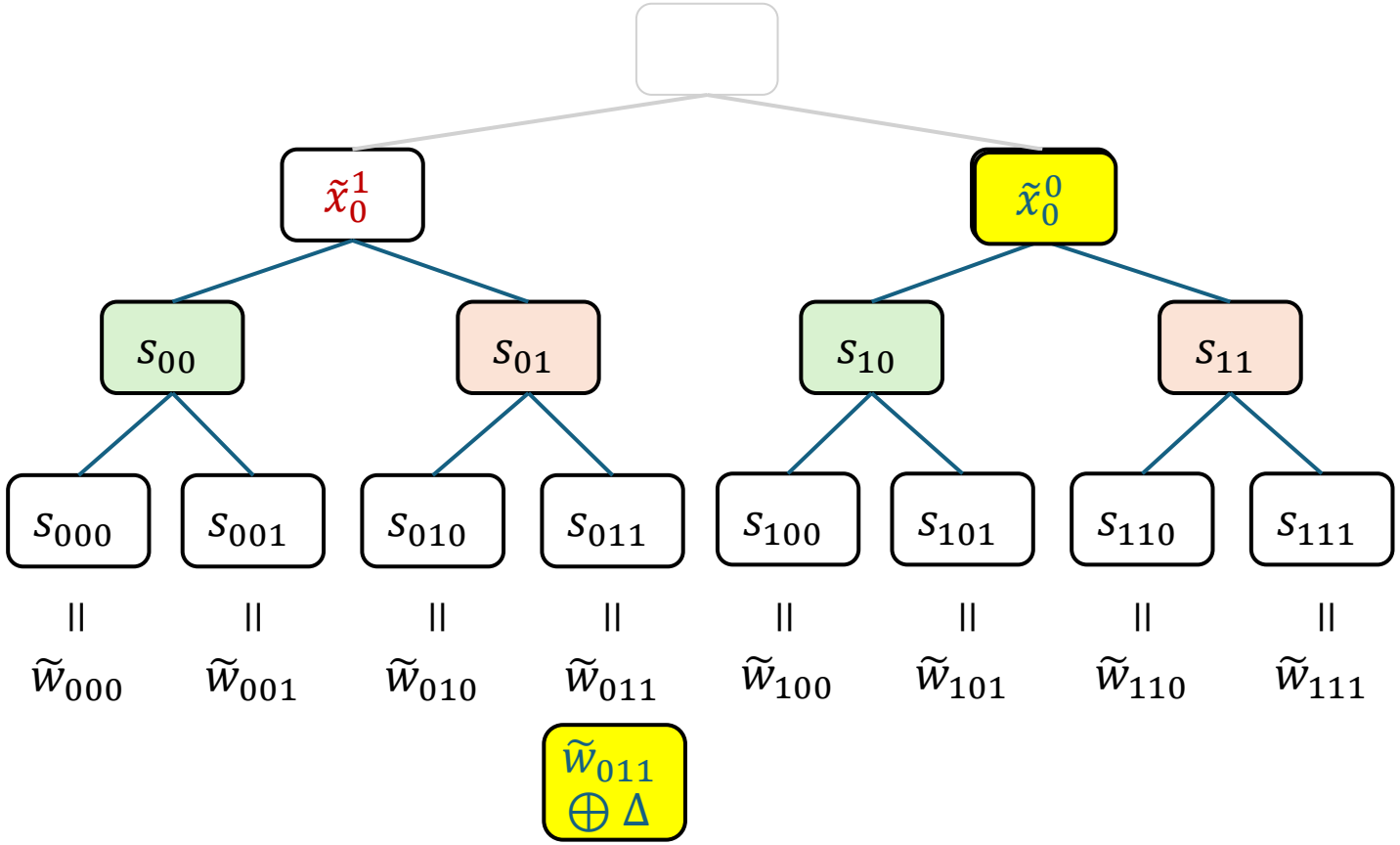


$$= f(3) \cdot \Delta \oplus \sum_{i \in \{0,1\}^3} \tilde{w}_i \cdot f(i)$$

$=$ output label of $f(3)$

Previous One-hot Garbling [Heath-Kolesnikov'21]

Goldreich-Goldwasser-Micali tree: Double seed length



One-hot garbling

$Enc_{\{\tilde{x}_1^1\}}(s_{00} \oplus s_{10})$ $Enc_{\{\tilde{x}_1^0\}}(s_{01} \oplus s_{11})$

$Enc_{\{\tilde{x}_2^1\}}(s_{000} \oplus s_{010} \oplus s_{100} \oplus s_{110} \oplus)$

$Enc_{\{\tilde{x}_2^0\}}(s_{001} \oplus s_{011} \oplus s_{101} \oplus s_{111} \oplus)$

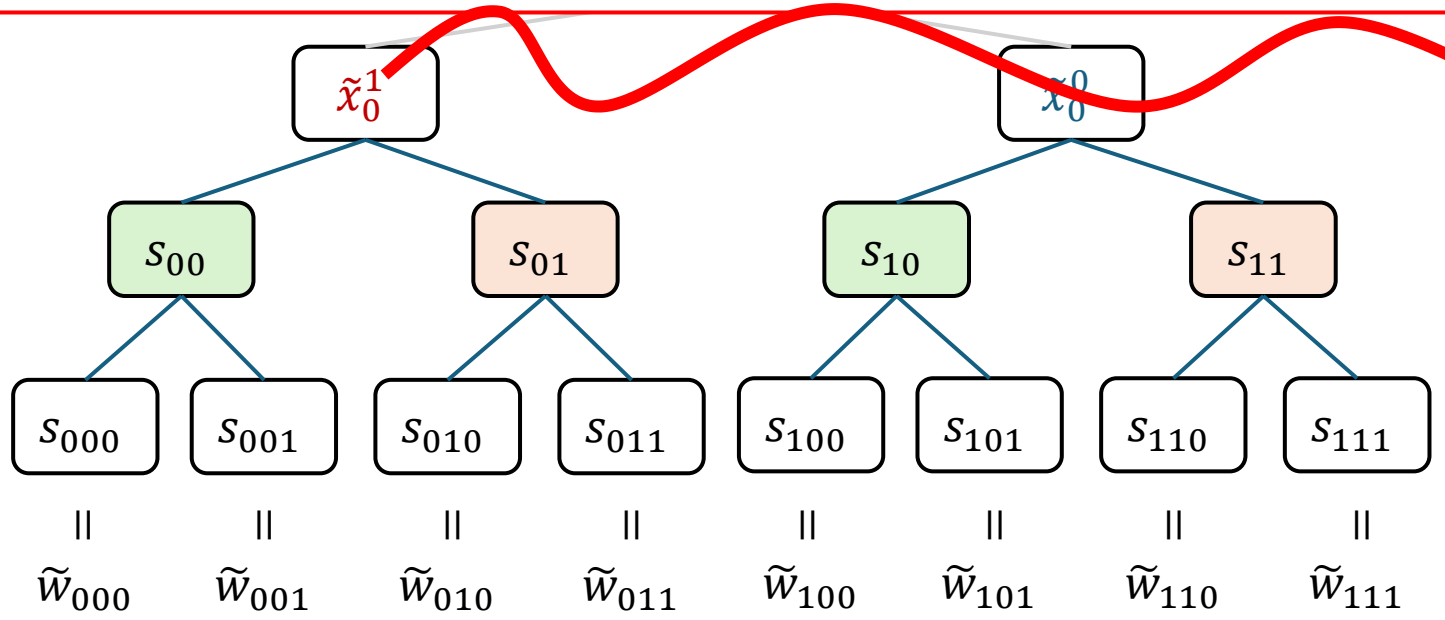
$\text{ct} := (\Delta \oplus s_{000} \oplus s_{001} \oplus s_{010} \oplus s_{011} \oplus s_{100} \oplus s_{101} \oplus s_{110} \oplus s_{111})$

Previous One-hot Garbling

[Heath-Kolesnikov'21]

Dependency:
When using free-XOR for input labels,
there are circularity and correlation.

$$\tilde{x}_0^1 = \tilde{x}_0^0 + \Delta$$
$$\tilde{x}_1^1 = \tilde{x}_1^0 + \Delta$$



hot garbling

$Enc_{\{\tilde{x}_1^1\}}$ $s_{00} \oplus$ s_{10} $Enc_{\{\tilde{x}_1^0\}}$ $s_{01} \oplus$ s_{11}

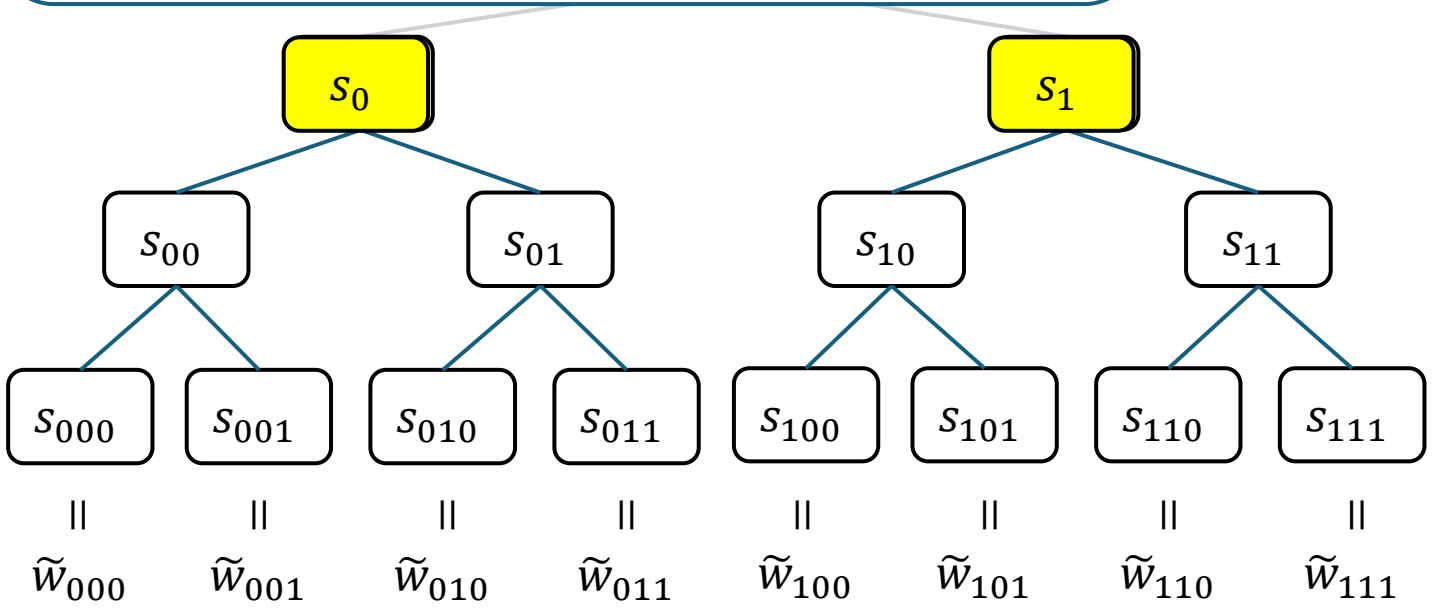
$Enc_{\{\tilde{x}_2^1\}}$ $s_{000} \oplus$ $s_{010} \oplus$ $s_{100} \oplus$ $s_{110} \oplus$

$Enc_{\{\tilde{x}_2^0\}}$ $s_{001} \oplus$ $s_{011} \oplus$ $s_{101} \oplus$ $s_{111} \oplus$

ct := Δ $\oplus s_{000}$ $\oplus s_{001}$ $\oplus s_{010}$ $\oplus s_{011}$
$\oplus s_{100}$ $\oplus s_{101}$ $\oplus s_{110}$ $\oplus s_{111}$

Our One-hot Garbling

Introduce random seeds (s_0, s_1)
+ indep labels (indep $\tilde{x}_0^0$ and $\tilde{x}_0^1$)
⇒ Break circularity



One-hot garbling

$Enc_{\{\tilde{x}_0^1\}}(s_0)$ $Enc_{\{\tilde{x}_0^0\}}(s_1)$

$Enc_{\{\tilde{x}_1^1\}}(s_{00} \oplus s_{10})$ $Enc_{\{\tilde{x}_1^0\}}(s_{01} \oplus s_{11})$

$Enc_{\{\tilde{x}_2^1\}}(s_{000} \oplus s_{010} \oplus s_{100} \oplus s_{110} \oplus)$

$Enc_{\{\tilde{x}_2^0\}}(s_{001} \oplus s_{011} \oplus s_{101} \oplus s_{111} \oplus)$

$\mathrm{ct} := \Delta \oplus s_{000} \oplus s_{001} \oplus s_{010} \oplus s_{011} \oplus s_{100} \oplus s_{101} \oplus s_{110} \oplus s_{111}$

* In fact, introducing random seeds is not required to break circularity in one-hot garbling, but it is necessary for PRF garbling. We include it here to illustrate our main technique.

Omitted Step: Garbled PRF

sample independent wire labels

+

introduce independent random seeds

+

address global dependency

Procedure 8. $\mathsf{PRF.Gb}\left((k_i^0, k_i^1)_{i\in[n]}\right)$:

1. Extract the permute bits: for all $i \in [n]$, $p_i := \mathsf{lsb}(k_i^0)$.
2. Prepare one-hot garbling (Procedure 5):

$$\left(\hat{\mathbf{h}} := (\hat{h}_z)_{z\in\{0,1\}^n}, \Delta_g, c^{\mathrm{hot}}\right) \leftarrow \mathsf{OneHot.Gb}\left((k_i^0, k_i^1)_{i\in[n]}\right).$$

3. Sample the PRF seeds and truth table (Procedure 7):

$$\left((s_i^{\mathrm{even}}, s_i^{\mathrm{odd}})_{i\in[n]}, r_n, \mathcal{T}_r\right) \leftarrow \mathsf{SeedGen}\left(1^\lambda, 1^n\right).$$

4. Prepare garbled PRF for each $i \in [n]$ as below:

   (a) Prepare "half" inner products:

$$t_i^{\mathrm{even}} := \bigoplus_{z\in\mathsf{Even}(n,i)} \mathsf{F}_{s_i^{\mathrm{even}}}(z)\cdot\hat{h}_z \quad \text{and}$$

$$t_i^{\mathrm{odd}} := \bigoplus_{z\in\mathsf{Odd}(n,i)} \mathsf{F}_{s_i^{\mathrm{odd}}}(z)\cdot\hat{h}_z.$$

   (b) Sample label $k_i^{\mathrm{msk}} \leftarrow \{0,1\}^\lambda$.

   (c) Compute $\begin{cases} c_i^{\mathrm{even}} := \mathsf{F}'_{k_i^0}(g) \oplus \left(s_i^{\mathrm{even}} \| (k_i^{\mathrm{msk}} \oplus t_i^{\mathrm{odd}})\right), \\ c_i^{\mathrm{odd}} := \mathsf{F}'_{k_i^1}(g) \oplus \left(s_i^{\mathrm{odd}} \| (k_i^{\mathrm{msk}} \oplus t_i^{\mathrm{even}})\right). \end{cases}$

5. Garbled the bit r_n: sample $k_n^{\mathrm{msk}} \leftarrow \{0,1\}^\lambda$, and then compute $c_n := k_n^{\mathrm{msk}} \oplus (r_n \cdot \Delta_g)$.
6. Solder the output wires: Compute

$$k^0 := \langle \mathcal{T}_r, \hat{\mathbf{h}} \rangle \oplus k_n^{\mathrm{msk}} \oplus \bigoplus_{i\in[n]} k_i^{\mathrm{msk}}, \quad \text{and} \quad k^1 := k^0 \oplus \Delta_g,$$

   and then garble the identity function $(k^{0,\mathrm{out}}, k^{1,\mathrm{out}}, c^{\mathrm{out}}) \leftarrow \mathsf{IDF.Gb}(k^0, k^1)$.
7. Output the following:
   - Truth table $\mathcal{T}_r$,
   - Garbled ciphertexts

$$c := \left(c^{\mathrm{hot}}, (c_i^{\mathrm{even}}, c_i^{\mathrm{odd}})_{i\in[n]}, c_n, c^{\mathrm{out}}, p := (p_0, \cdots, p_{n-1})\right),$$

   - Output labels $(k^{0,\mathrm{out}}, k^{1,\mathrm{out}})$.

Eprint: https://eprint.iacr.org/2025/578.pdf

Future Directions

- This work: Remove circularity from One-hot Garbling, Garbled PRF, Garbled LUT
  - Conjecture: PRF-based and RO-based differ by const factor?!
- Linear-commu arithmetic garbling [Heath'24], based on PRF?
- Garbled LUT with public table?
  - [Ng-Kolesnikov'25] achieves $O\left(\sqrt{N\kappa}\right)$ commu and $O(N\kappa)$ comp
- Large domain Garbled PRF, better efficiency?