

Secure Computing of Table with FLUTE

카이스트 전산학부 한승우

2024년 9월 6일

Lookup Table

LUT (Lookup Table)

δ -to- σ lookup table T is a function $T: \{0, 1\}^\delta \rightarrow \{0, 1\}^\sigma$.

x_1	x_2	x_3	y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Goal Given secret shares of x_1 , x_2 , and x_3 , we want to compute the secret shares of $T(x_1, x_2, x_3)$.

Question What secret sharing scheme should we use?

Question How can we optimize the online and setup cost? (Up to **2σ bits per LUT calculation** in the online phase!)

Example 3-to-1 LUT T

[·] Secret Sharing

P_0, P_1 : parties

[·] Secret Sharing

A bit $v \in \mathbb{Z}_2$ is said to be $[\cdot]$ -shared between P_0 and P_1 if P_i holds $[v]_i$ such that

$$v = [v]_0 \oplus [v]_1.$$

[·] Secret Sharing

The [·]-sharing is *linear* in the sense that:

- 1 For [·]-shared $u, v \in \mathbb{Z}_2$, [·]-sharing of $u \oplus v$ can be locally computed by:

$$[u \oplus v]_i = [u]_i \oplus [v]_i \text{ for each } i \in \{0, 1\}.$$

- 2 For [·]-shared $v \in \mathbb{Z}_2$ and public $b \in \mathbb{Z}_2$, [·]-sharing of bv can be locally computed by:

$$[bv]_i = b[v]_i \text{ for each } i \in \{0, 1\}.$$

- 3 For [·]-shared $v \in \mathbb{Z}_2$ and public $b \in \mathbb{Z}_2$, [·]-sharing of $v \oplus b$ can be locally computed by:

$$[v \oplus b]_i = [v]_i \oplus ib \text{ for each } i \in \{0, 1\}.$$

In particular, [·]-sharing of $\bar{v} = v \oplus 1$ can be locally computed.

Multiplication of $[\cdot]$ -shared Values

Beaver's Multiplication Triple

Suppose P_0 and P_1 have $[\cdot]$ -shares of $u, v \in \mathbb{Z}_2$. Suppose additionally that they have $[\cdot]$ -shares of a, b, c where $c = ab$.

- 1 P_i computes $[u \oplus a]_i$ and $[v \oplus b]_i$, and sends them to P_{1-i} .
- 2 P_0 and P_1 now know $d := u \oplus a$ and $e := v \oplus b$.
- 3 P_i computes $[z]_i = i \cdot de \oplus d[b]_i \oplus e[a]_i \oplus [c]_i$.
- 4 Then, $[z]_0 \oplus [z]_1 = z = uv$.

With the prepared *multiplication triple*, a $[\cdot]$ -share of uv is calculated with the cost of four bits in a single round.

$$\begin{aligned} z &= uv = (d \oplus a)(e \oplus b) \\ &= \underbrace{de}_{\text{public}} \oplus db \oplus ea \oplus \underbrace{ab}_{=c} \end{aligned}$$

$\langle \cdot \rangle$ Secret Sharing

$\langle \cdot \rangle$ Secret Sharing

A bit $v \in \mathbb{Z}_2$ is said to be $\langle \cdot \rangle$ -shared if:

- 1 A value $\lambda_v \in \mathbb{Z}_2$ is $[\cdot]$ -shared between P_0 and P_1 .
 - $[\lambda_v]_i$'s are locally set arbitrarily.
- 2 A value $m_v \in \mathbb{Z}_2$ is known to both parties.
- 3 $v = m_v \oplus \lambda_v$

The $\langle \cdot \rangle$ -share of v is denoted $\langle v \rangle_i = (m_v, [\lambda_v]_i)$.

⟨·⟩ Secret Sharing

The ⟨·⟩-sharing is *linear* in the sense that:

- 1 For ⟨·⟩-shared $u, v \in \mathbb{Z}_2$, ⟨·⟩-sharing of $u \oplus v$ can be locally computed by:

$$m_{u \oplus v} = m_u \oplus m_v \text{ and } [\lambda_{u \oplus v}]_i = [\lambda_u]_i \oplus [\lambda_v]_i$$

- 2 For ⟨·⟩-shared $v \in \mathbb{Z}_2$ and public $b \in \mathbb{Z}_2$, ⟨·⟩-sharing of bv can be locally computed by:

$$m_{bv} = bm_v \text{ and } [\lambda_{bv}]_i = b[\lambda_v]_i.$$

- 3 For ⟨·⟩-shared $v \in \mathbb{Z}_2$ and public $b \in \mathbb{Z}_2$, ⟨·⟩-sharing of $v \oplus b$ can be locally computed by:

$$m_{v \oplus b} = m_v \oplus b.$$

In particular, ⟨·⟩-sharing of $\bar{v} = v \oplus 1$ can be locally computed.

LUT Revisited

x_1	x_2	x_3	y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Example 3-to-1 LUT T

$$(\overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3})$$

$$\vee (\overline{x_1} \wedge x_2 \wedge x_3)$$

$$\vee (x_1 \wedge \overline{x_2} \wedge x_3)$$

DNF representation of T

$$(\overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3})$$

$$\oplus (\overline{x_1} \wedge x_2 \wedge x_3)$$

$$\oplus (x_1 \wedge \overline{x_2} \wedge x_3)$$

An equivalent circuit.

$$y = \begin{pmatrix} \overline{x_1} \\ \overline{x_1} \\ x_1 \end{pmatrix} \odot \begin{pmatrix} \overline{x_2} \\ x_2 \\ \overline{x_2} \end{pmatrix} \odot \begin{pmatrix} \overline{x_3} \\ x_3 \\ x_3 \end{pmatrix}$$

“Multi-Fan-In Inner Product”

Multiplication of $\langle \cdot \rangle$ -shared Values

$$\begin{aligned} z = uv &= (m_u \oplus \lambda_u)(m_v \oplus \lambda_v) \\ &= m_u m_v \oplus m_u \lambda_v \oplus m_v \lambda_u \oplus \lambda_u \lambda_v \end{aligned}$$

Multiplication of $\langle \cdot \rangle$ -shared Values

Input $\langle \cdot \rangle$ -shares of $u, v \in \mathbb{Z}_2$, $[\cdot]$ -shares of $\lambda_u \lambda_v$, and $[\cdot]$ -shares of λ_z .

Output $\langle \cdot \rangle$ -shares of $z := uv$.

- 1 P_i computes $[z]_i = i \cdot m_u m_v \oplus m_u [\lambda_v]_i \oplus m_v [\lambda_u]_i \oplus [\lambda_u \lambda_v]_i$.
- 2 P_i computes $[m_z]_i = [z]_i \oplus [\lambda_z]_i$ and sends it to P_{i-1} .
- 3 Then, P_0 and P_1 have $\langle \cdot \rangle$ -shares of $z = uv$.

With the prepared multiplication triple, a $\langle \cdot \rangle$ -share of uv is calculated with the cost of two bits in a single round.

If z is an operand of multiplication in following steps through the circuit, λ_z must be settled in the setup phase to optimize the online communication cost.

Inner Product of $\langle \cdot \rangle$ -shared Values

Inner Product of $\langle \cdot \rangle$ -shared Values

Input $\langle \cdot \rangle$ -shares of $u^1, \dots, u^N, v^1, \dots, v^N \in \mathbb{Z}_2$, $[\cdot]$ -shares of $\lambda_{u^1} \lambda_{v^1}, \dots, \lambda_{u^N} \lambda_{v^N}$, and $[\cdot]$ -shares of λ_z .

Output $\langle \cdot \rangle$ -shares of $z := \bigoplus_{k=1}^N u^k v^k$.

- 1 P_i computes $[u^k v^k]_i$ for $k \in [N]$ as before.
- 2 P_i computes $[m_z]_i = [\lambda_z]_i \oplus \bigoplus_{k=1}^N [u^k v^k]_i$ and sends it to P_{1-i} .
- 3 Then, P_0 and P_1 have $\langle \cdot \rangle$ -shares of $z = \bigoplus_{k=1}^N u^k v^k$.

With the prepared N multiplication triples, a $\langle \cdot \rangle$ -share of $\bigoplus_{k=1}^N u^k v^k$ is calculated with the cost of two bits in a single round.

Multi-Fan-In Product of $\langle \cdot \rangle$ -shared Values

Multi-Fan-In Product of $\langle \cdot \rangle$ -shared Values

Input $\langle \cdot \rangle$ -shares of $u^1, \dots, u^M \in \mathbb{Z}_2$, ???, and $[\cdot]$ -shares of λ_z .

Output $\langle \cdot \rangle$ -shares of $z := \bigwedge_{j=1}^M u^j$.

Let \mathcal{Q} be a set of $\langle \cdot \rangle$ -shared values (wires). We use the following notation:

$$m_{\mathcal{Q}} \triangleq \bigwedge_{u \in \mathcal{Q}} m_u \quad \text{and} \quad \lambda_{\mathcal{Q}} \triangleq \bigwedge_{u \in \mathcal{Q}} \lambda_u.$$

Observation

$$\bigwedge_{j=1}^M u^j = \bigwedge_{j=1}^M (m_{u^j} \oplus \lambda_{u^j}) = \bigoplus_{\mathcal{Q} \subseteq \mathcal{I}} (m_{\mathcal{Q}} \cdot \lambda_{\mathcal{I} \setminus \mathcal{Q}})$$

where $\mathcal{I} := \{u^1, u^2, \dots, u^M\}$.

Multi-Fan-In Product of $\langle \cdot \rangle$ -shared Values

Multi-Fan-In Product of $\langle \cdot \rangle$ -shared Values

Input $\langle \cdot \rangle$ -shares of $u^1, \dots, u^M \in \mathbb{Z}_2$, $[\cdot]$ -shares of λ_Q for each $Q \subseteq \mathcal{I}$ with $|Q| > 1$ where $\mathcal{I} = \{u^1, \dots, u^M\}$, and $[\cdot]$ -shares of λ_z .

Output $\langle \cdot \rangle$ -shares of $\bigwedge_{j=1}^M u^j$.

① P_i computes

$$[z]_i = i \cdot m_{\mathcal{I}} \oplus \bigoplus_{Q \not\subseteq \mathcal{I}} (m_Q \cdot [\lambda_{\mathcal{I} \setminus Q}]_i).$$

② P_i computes $[m_z]_i = [z]_i \oplus [\lambda_z]_i$ and sends it to P_{1-i} .

③ Then, P_0 and P_1 have $\langle \cdot \rangle$ -shares of $\bigwedge_{j=1}^M u^j$.

With the prepared $2^M - M - 1$ multiplication triples, a $\langle \cdot \rangle$ -share of $\bigwedge_{j=1}^M u^j$ is calculated with the cost of two bits in a single round.

Multi-Fan-In Inner Product of $\langle \cdot \rangle$ -shared Values

Multi-Fan-In Inner Product of $\langle \cdot \rangle$ -shared Values

Input $\langle \cdot \rangle$ -shares of $\mathbf{u}^1, \dots, \mathbf{u}^M \in (\mathbb{Z}_2)^N$, $[\cdot]$ -shares of λ_Q for each $Q \subseteq \mathcal{I}_k$ with $|Q| > 1$ where $\mathcal{I}_k = \{\mathbf{u}_k^1, \dots, \mathbf{u}_k^M\}$ for $k \in [N]$, and $[\cdot]$ -sharing of λ_z .

Output $\langle \cdot \rangle$ -shares of $z := \mathbf{u}^1 \odot \dots \odot \mathbf{u}^M = \bigoplus_{k=1}^N \bigwedge_{j=1}^M \mathbf{u}_k^j$.

① P_i computes

$$[z]_i = i \cdot \bigoplus_{k=1}^N m_{\mathcal{I}_k} \oplus \bigoplus_{k=1}^N \bigoplus_{Q \subsetneq \mathcal{I}_k} (m_Q \cdot [\lambda_{\mathcal{I}_k \setminus Q}]_i).$$

② P_i computes $[m_z]_i = [z]_i \oplus [\lambda_z]_i$ and sends it to P_{1-i} .

③ Then, P_0 and P_1 have $\langle \cdot \rangle$ -shares of $\mathbf{u}^1 \odot \dots \odot \mathbf{u}^N$.

With the prepared $N(2^M - M - 1)$ multiplication triples, a $\langle \cdot \rangle$ -share of $\bigodot_{j=1}^N \mathbf{u}^j$ is calculated with the cost of two bits in a single round.

Calculating LUT

Input $\langle \cdot \rangle$ -shares of $\mathbf{u}^1, \dots, \mathbf{u}^M \in (\mathbb{Z}_2)^N$, $[\cdot]$ -shares of λ_Q for each $Q \subseteq \mathcal{I}_k$ with $|Q| > 1$ where $\mathcal{I}_k = \{\mathbf{u}_k^1, \dots, \mathbf{u}_k^M\}$ for $k \in [N]$, and $[\cdot]$ -sharing of λ_z .

Output $\langle \cdot \rangle$ -shares of $z := \mathbf{u}^1 \odot \dots \odot \mathbf{u}^M = \bigoplus_{k=1}^N \bigwedge_{j=1}^M \mathbf{u}_k^j$.

With the prepared $N(2^M - M - 1)$ multiplication triples, a $\langle \cdot \rangle$ -share of $\bigodot_{j=1}^N \mathbf{u}^j$ is calculated with the cost of two bits in a single round.

$$y = \begin{pmatrix} \overline{x_1} \\ \overline{x_1} \\ x_1 \end{pmatrix} \odot \begin{pmatrix} \overline{x_2} \\ x_2 \\ \overline{x_2} \end{pmatrix} \odot \begin{pmatrix} \overline{x_3} \\ x_3 \\ x_3 \end{pmatrix}$$

For δ -to-1 LUT,
 $N \leq 2^{\delta-1}$ and $M = \delta$??
 $2^{\delta-1}(2^\delta - \delta - 1)$ triples??

We now exploit the fact that elements of \mathbf{u}^j are closely related.

λ_v is unchanged when calculating shares of \bar{v} !

Some Definitions

Fix a δ -to- σ LUT T .

x^1	x^2	x^3	y^1	y^2
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	0
1	1	1	0	1
e^1	e^2	e^3	y^1	y^2

For instance, in this example,

$$\mathbf{e}^2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \quad \text{and} \quad \mathbf{y}^1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Expressing LUT with Multi-Fan-In Inner Product

Now, in the previous example, note that:

x^1	x^2	x^3	y^1
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

$$\begin{aligned}
 \mathbf{z}^1 &= \begin{pmatrix} \overline{x^1} \\ \overline{x^1} \\ x^1 \end{pmatrix} \odot \begin{pmatrix} \overline{x^2} \\ x^2 \\ \overline{x^2} \end{pmatrix} \odot \begin{pmatrix} \overline{x^3} \\ x^3 \\ x^3 \end{pmatrix} \\
 &= \begin{pmatrix} \overline{x^1} \\ \overline{x^1} \\ \overline{x^1} \\ \vdots \end{pmatrix} \odot \begin{pmatrix} \overline{x^2} \\ \overline{x^2} \\ x^2 \\ x^2 \\ \vdots \end{pmatrix} \odot \begin{pmatrix} \overline{x^3} \\ \overline{x^3} \\ \overline{x^3} \\ x^3 \\ x^3 \\ \vdots \end{pmatrix} \odot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ \vdots \end{pmatrix} \\
 &= (\overline{x^1} \oplus \mathbf{e}^1) \odot \dots \odot (\overline{x^3} \oplus \mathbf{e}^3) \odot \mathbf{y}^1
 \end{aligned}$$

$\overline{x^j} \oplus \mathbf{e}^j$ is an element-wise addition.

Expressing LUT with Multi-Fan-In Inner Product

Fixing a δ -to- σ LUT T , and given $\langle \cdot \rangle$ -shares of x^1, \dots, x^δ , let

$$\mathbf{u}^j := \overline{x^j} \oplus \mathbf{e}^j \in (\mathbb{Z}_2)^{2^\delta}$$

for $j \in [\delta]$ so that

$$\langle \mathbf{u}_k^j \rangle_i = (\overline{m_{x^j}} \oplus \mathbf{e}_k^j, [\lambda_{x^j}]_i)$$

gives a $\langle \cdot \rangle$ -sharing of \mathbf{u}_k^j for $j \in [\delta]$, $k \in [2^\delta]$.

With this, we have

$$\lambda_{\mathbf{u}_{k_1}^{j_1} \dots \mathbf{u}_{k_n}^{j_n}} = \lambda_{x^{j_1} \dots x^{j_n}} !$$

Expressing LUT with Multi-Fan-In Inner Product

As previously shown, the w -th output \mathbf{z}^w can be calculated by

$$\begin{aligned}
 \mathbf{z}^w &= (\overline{x^1} \oplus \mathbf{e}^1) \odot \cdots \odot (\overline{x^\delta} \oplus \mathbf{e}^\delta) \odot \mathbf{y}^w \\
 &= \mathbf{u}^1 \odot \cdots \odot \mathbf{u}^\delta \odot \mathbf{y}^w \\
 &= \bigoplus_{k=1}^{2^\delta} \left[\left(\bigwedge_{j=1}^{\delta} \mathbf{u}_k^j \right) \wedge \mathbf{y}_k^w \right] \\
 &= \bigoplus_{k=1}^{2^\delta} \left[\left(\bigwedge_{j=1}^{\delta} \left(m_{\mathbf{u}_k^j} \oplus \lambda_{\mathbf{u}_k^j} \right) \right) \wedge \mathbf{y}_k^w \right] \\
 &= \bigoplus_{k=1}^{2^\delta} \left[\mathbf{y}_k^w \cdot \bigoplus_{\mathcal{Q} \subseteq \mathcal{I}_k} (m_{\mathcal{Q}} \cdot \lambda_{\mathcal{I}_k \setminus \mathcal{Q}}) \right]
 \end{aligned}$$

where $\mathcal{I}_k := \{\mathbf{u}_k^1, \cdots, \mathbf{u}_k^\delta\}$ for each $k \in [2^\delta]$.

Calculating LUT

Calculating δ -to- σ LUT

Input A δ -to- σ LUT T , $\langle \cdot \rangle$ -shares of $x^1, \dots, x^\delta \in \mathbb{Z}_2$, $[\cdot]$ -shares of λ_Q for each $Q \subseteq \mathcal{I}$ with $|Q| > 1$ where $\mathcal{I} = \{x^1, \dots, x^\delta\}$, and $[\cdot]$ -shares of λ_z .

Output $\langle \cdot \rangle$ -shares of $z := T(x^1, \dots, x^\delta)$.

① P_i computes its share of $\mathbf{u}_k^j = \overline{x^j} \oplus \mathbf{e}_k^j$ for $j \in [\delta]$ and $k \in [2^\delta]$.

② P_i computes, for each $w \in [\sigma]$,

$$[\mathbf{z}_w]_i = i \cdot \bigoplus_{k=1}^{2^\delta} (\mathbf{y}_k^w \cdot m_{\mathcal{I}_k}) \oplus \bigoplus_{k=1}^{2^\delta} \left[\mathbf{y}_k^w \cdot \bigoplus_{Q \subsetneq \mathcal{I}_k} (m_Q \cdot [\lambda_{\mathcal{I}_k \setminus Q}]_i) \right].$$

③ P_i computes $[m_{\mathbf{z}_w}]_i = [\mathbf{z}_w]_i \oplus [\lambda_{\mathbf{z}_w}]_i$ for $w \in [\sigma]$ and sends them to P_{1-i} .

④ Then, P_0 and P_1 have $\langle \cdot \rangle$ -shares of $\mathbf{z} = T(x^1, \dots, x^\delta)$.

Calculating LUT

Calculating δ -to- σ LUT

Input A δ -to- σ LUT T , $\langle \cdot \rangle$ -shares of $x^1, \dots, x^\delta \in \mathbb{Z}_2$, $[\cdot]$ -shares of λ_Q for each $Q \subseteq \mathcal{I}$ with $|Q| > 1$ where $\mathcal{I} = \{x^1, \dots, x^\delta\}$, and $[\cdot]$ -shares of λ_z .

Output $\langle \cdot \rangle$ -shares of $z := T(x^1, \dots, x^\delta)$.

With the prepared $2^\delta - \delta - 1$ multiplication triples, a $\langle \cdot \rangle$ -share of $T(x^1, \dots, x^\delta)$ is calculated with the cost of 2σ bits in a single round.

Pros and Cons

Pros

- Any complex circuit can be represented by an *interconnection of small LUTs*.
- Evaluation does not depend on the internal logic.
- Setup does not depend on the number of outputs.

Cons

- Exponential setup time/communicaion.
- Exponential internal calculation.

Analytic Comparison with Other Protocols

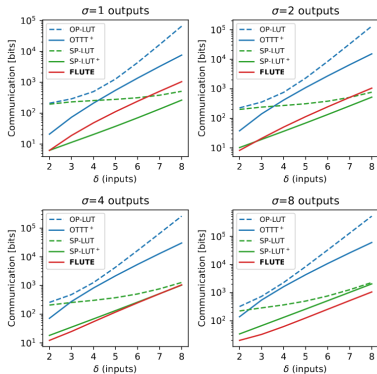


Figure 7: Total communication for different LUT sizes with $2 \leq \delta \leq 8$ inputs and $\sigma \in \{1, 2, 4, 8\}$ outputs.

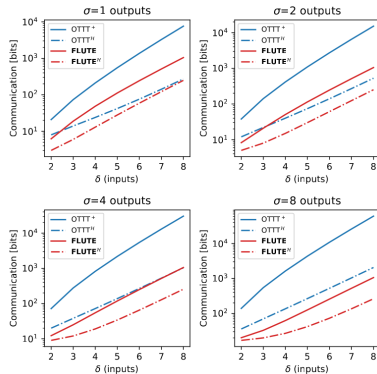
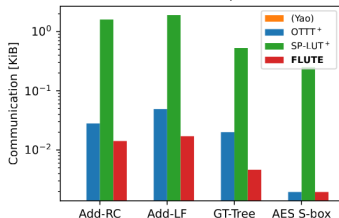


Figure 8: Total communication for different LUT sizes with $2 \leq \delta \leq 8$ inputs and $\sigma \in \{1, 2, 4, 8\}$ outputs using a helper server S_H (cf. §C).

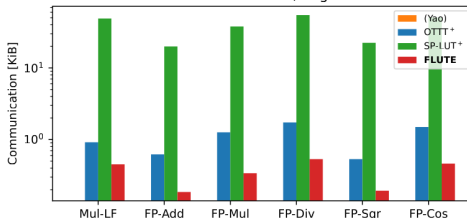
Benchmarks

Online/Total Communication Compared with Other Protocols

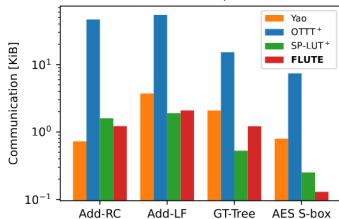
Online communication, small circuits



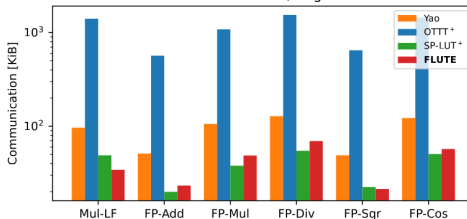
Online communication, large circuits



Total communication, small circuits



Total communication, large circuits



References

- Brüggemann, A., Hundt, R., Schneider, T., Suresh, A., & Yalame, H. (2023). Flute: Fast and secure lookup table evaluations. *2023 IEEE Symposium on Security and Privacy (SP)*.
<https://doi.org/10.1109/sp46215.2023.10179345>