通用 MPC 协议构造

混淆电路的优化

张聪
zhangcong@iie.ac.cn

中国科学院信息工程研究所国家重点实验室

2023 年 3 月 11 日

1. GC 构造优化
2. 参考文献

GC 优化

经过了 20 多年的发展，GC 的构造效率已经得到极大提升：

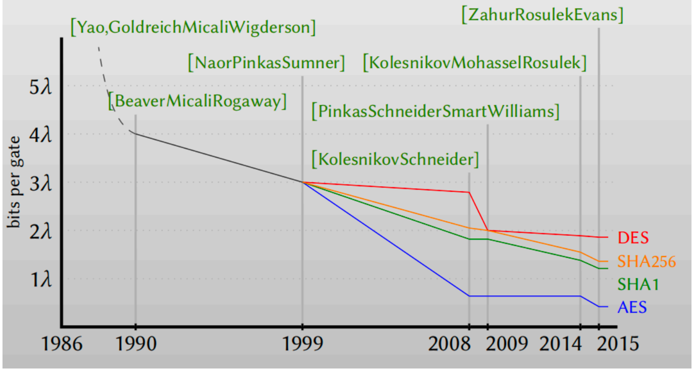


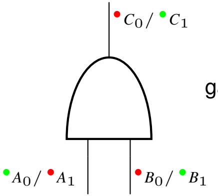
图 1: GC 发展

基本对比

GC schemes	GC size (κ bits per gates)		calls to H per gate: Garbler		calls to H per gate: Evaluator		Assumption
	AND	XOR	AND	XOR	AND	XOR	
Unoptimized textbook Yao [Yao86]	8	8	4	4	2.5	2.5	PRF
Yao with point-and-permute [BMR90]	4	4	4	4	1	1	PRF
GRR3 [NPS99]	3	3	4	4	1	1	PRF
GRR2 [Pin+09]	2	2	4	4	1	1	PRF
Free-XOR [KS08]	3	0	4	0	1	0	CCR
Fle-XOR [KMR14]	2	$\{0,1,2\}$	4	$\{0,2,4\}$	1	$\{0,1,2\}$	CCR
Fast-GRR2[Gue+15]	2	1	4	3	2	1.5	PRF
Half-Gate [ZRE15]	2	0	4	0	2	0	CCR
Slicing-and-Dicing [RR21]	1.5	0	≤ 6	0	≤ 3	0	CCR

表 1: 不同 GC 方案对比。PRF = pseudorandom function, CCR = circular correlation robust hash function.

Point-and-Permute



$\xrightarrow{\text{garbles}}$

a	b	Garbled value
${\color{green}\bullet}A_0$	${\color{red}\bullet}B_0$	$\mathbb{E}_{A_0,B_0}({\color{red}\bullet}C_0)$
${\color{green}\bullet}A_0$	${\color{green}\bullet}B_1$	$\mathbb{E}_{A_0,B_1}({\color{red}\bullet}C_0)$
${\color{red}\bullet}A_1$	${\color{red}\bullet}B_0$	$\mathbb{E}_{A_1,B_0}({\color{red}\bullet}C_0)$
${\color{red}\bullet}A_1$	${\color{green}\bullet}B_1$	$\mathbb{E}_{A_1,B_1}({\color{green}\bullet}C_1)$

$\xrightarrow{\text{permutes}}$

Order	Garbled value
${\color{red}\bullet}{\color{red}\bullet}$	$\mathbb{E}_{A_1,B_0}({\color{red}\bullet}C_0)$
${\color{red}\bullet}{\color{green}\bullet}$	$\mathbb{E}_{A_1,B_1}({\color{green}\bullet}C_1)$
${\color{green}\bullet}{\color{red}\bullet}$	$\mathbb{E}_{A_0,B_0}({\color{red}\bullet}C_0)$
${\color{green}\bullet}{\color{green}\bullet}$	$\mathbb{E}_{A_0,B_1}({\color{red}\bullet}C_0)$

$$\mathbb{E}_{A,B}(C) := H(A,B) \oplus C$$

Point-and-Permute

$\mathrm{Gb}(1^\kappa, f)$:

1. $(n, m, q, in_1, in_2, G) \leftarrow f$
2. 对每条线 $i \in [n+q]$: $\lambda_i \leftarrow \{0,1\}, W_i^0 \leftarrow \{0,1\}^{\kappa-1}\lambda_i, W_i^1 \leftarrow \{0,1\}^{\kappa-1}\bar{\lambda}_i$
3. 对每个门 $c \in [n+1, n+q]$: $a \leftarrow in_1(c), b \leftarrow in_2(c), A_0, A_1, B_0, B_1, C_0, C_1 \leftarrow W_a^0, W_a^1, W_b^0, W_b^1, W_c^0, W_c^1, \lambda_a, \lambda_b \leftarrow lsb(A_0), lsb(B_0)$, 计算

$$
\begin{aligned}
G_{0,0} &= H(A_{\lambda_a}, B_{\lambda_b}, c) \oplus C_{G_c(\lambda_a,\lambda_b)} \\
G_{0,1} &= H(A_{\lambda a}, B_{\bar{\lambda}_b}, c) \oplus C_{G_c(\lambda_a,\bar{\lambda}_b)} \\
G_{1,0} &= H(A_{\bar{\lambda}_a}, B_{\lambda_b}, c) \oplus C_{G_c(\bar{\lambda}_a,\lambda_b)} \\
G_{1,1} &= H(A_{\bar{\lambda}_a}, B_{\bar{\lambda}_b}, c) \oplus C_{G_c(\bar{\lambda}_a,\bar{\lambda}_b)}
\end{aligned}
$$

$$\vec{G}_c := (G_{0,0}, G_{0,1}, G_{1,0}, G_{1,1})$$

4. $F := (n, m, q, in_1, in_2, \vec{G}), e := \{W_i^0, W_i^1\}_{i\in[n]}, d := \{\lambda_{n+q-m+i}\}_{i\in[m]}$
5. 输出 (F, e, d)

Point-and-Permute

$\mathrm{Ev}(F, X)$:

1. $(n, m, q, in_1, in_2, \vec{G}) \leftarrow F$
2. $(W_1, \ldots, W_n) \leftarrow X$
3. 对每个门 $c \in [n+1, n+q]$:
   $a \leftarrow in_1(c), b \leftarrow in_2(c), \hat{a}, \hat{b} \leftarrow lsb(W_a), lsb(W_b), (G_{0,0}, G_{0,1}, G_{1,0}, G_{1,1}) \leftarrow \vec{G}_c$.
   $W_c := H(W_a, W_b, c) \oplus G_{\hat{a},\hat{b}}$
4. 输出 $Y := \{W_{n+q-m+i}\}_{i\in[m]}$

Point-and-Permute

$\mathrm{De}(d, Y)$:

1. $\{d_i\}_{i\in[m]} \leftarrow d$
2. $\{Y_i\}_{i\in[m]} \leftarrow Y$
3. 对 $i \in [m]$: 令 $y_i := d_i \oplus lsb(Y_i)$.
4. 输出 $y := y_1 \ldots y_m$

Free-XOR

$\mathrm{Gb}(1^\kappa, f)$:

1. $(n, m, q, in_1, in_2, G) \leftarrow f$
2. $\Delta \leftarrow \{0,1\}^{\kappa-1}1$
3. 对每条输入线 $i \in [n]$: $\lambda_i \leftarrow \{0,1\}, W_i^0 \leftarrow \{0,1\}^{\kappa-1}\lambda_i, W_i^1 := W_i^0 \oplus \Delta$
4. 对每个门 $c \in [n+1, n+q]$:

   $a \leftarrow in_1(c), b \leftarrow in_2(c), A_0, A_1, B_0, B_1 \leftarrow W_a^0, W_a^1, W_b^0, W_b^1, \lambda_a, \lambda_b \leftarrow lsb(A_0), lsb(B_0)$,

   如果 $G = XOR$:

   $$W_c^0 = C_0 := A_0 \oplus B_0, W_c^1 = C_1 := W_c^0 \oplus \Delta$$

   如果 $G = AND$:

   $$W_c^0 = C_0 \leftarrow \{0,1\}^\kappa, W_c^1 = C_1 := W_c^0 \oplus \Delta$$
   $$G_{0,0} = H(A_{\lambda_a}, B_{\lambda_b}, c) \oplus C_{\lambda_a \wedge \lambda_b}$$
   $$G_{0,1} = H(A_{\lambda a}, B_{\bar{\lambda}_b}, c) \oplus C_{\lambda_a \wedge \bar{\lambda}_b}$$
   $$G_{1,0} = H(A_{\bar{\lambda}_a}, B_{\lambda_b}, c) \oplus C_{\bar{\lambda}_a \wedge \lambda_b}$$
   $$G_{1,1} = H(A_{\bar{\lambda}_a}, B_{\bar{\lambda}_b}, c) \oplus C_{\bar{\lambda}_a \wedge \bar{\lambda}_b}$$

   $\vec{G}_c := (G_{0,0}, G_{0,1}, G_{1,0}, G_{1,1})$
5. $F := (n, m, q, in_1, in_2, \vec{G}), e := \{W_i^0, W_i^1\}_{i \in [n]}, d := \{\lambda_{n+q-m+i}\}_{i \in [m]}$
6. 输出 (F, e, d)

Free-XOR

$\mathrm{Ev}(F, X)$:

1. $(n, m, q, in_1, in_2, \vec{G}) \leftarrow F$
2. $(W_1, \ldots, W_n) \leftarrow X$
3. 对每个门 $c \in [n+1, n+q]$:
   $a \leftarrow in_1(c), b \leftarrow in_2(c), \hat{a}, \hat{b} \leftarrow lsb(W_a), lsb(W_b), (G_{0,0}, G_{0,1}, G_{1,0}, G_{1,1}) \leftarrow \vec{G}_c$.
   如果 $G = XOR$:
   $$W_c := W_a \oplus W_b$$
   如果 $G = AND$:
   $$W_c := H(W_a, W_b, c) \oplus G_{\hat{a},\hat{b}}$$
4. 输出 $Y := \{W_{n+q-m+i}\}_{i\in[m]}$

Garbled Row Reduction $4 \to 3$ (GRR3)

$\mathrm{Gb}(1^\kappa, f)$:

1. $(n, m, q, in_1, in_2, G) \leftarrow f$
2. $\Delta \leftarrow \{0,1\}^{\kappa-1}1$
3. 对每条输入线 $i \in [n]$: $\lambda_i \leftarrow \{0,1\}, W_i^0 \leftarrow \{0,1\}^{\kappa-1}\lambda_i, W_i^1 := W_i^0 \oplus \Delta$
4. 对每个门 $c \in [n+1, n+q]$:
   $a \leftarrow in_1(c), b \leftarrow in_2(c), A_0, A_1, B_0, B_1 \leftarrow W_a^0, W_a^1, W_b^0, W_b^1, \lambda_a, \lambda_b \leftarrow lsb(A_0), lsb(B_0),$
   如果 $G = XOR$:
   $$W_c^0 = C_0 := A_0 \oplus B_0, W_c^1 = C_1 := W_c^0 \oplus \Delta$$
   如果 $G = AND$:
   $$W_c^{\lambda_a \wedge \lambda_b} = C_{\lambda_a \wedge \lambda_b} := H(A_{\lambda_a}, B_{\lambda_b}, c), W_c^{\lambda_a \wedge \lambda_b \oplus 1} = C_{\lambda_a \wedge \lambda_b \oplus 1} := W_c^{\lambda_a \wedge \lambda_b} \oplus \Delta$$
   $$G_{0,1} = H(A_{\lambda a}, B_{\bar{\lambda}_b}, c) \oplus C_{\lambda_a \wedge \bar{\lambda}_b}$$
   $$G_{1,0} = H(A_{\bar{\lambda}_a}, B_{\lambda_b}, c) \oplus C_{\bar{\lambda}_a \wedge \lambda_b}$$
   $$G_{1,1} = H(A_{\bar{\lambda}_a}, B_{\bar{\lambda}_b}, c) \oplus C_{\bar{\lambda}_a \wedge \bar{\lambda}_b}$$
   $\vec{G}_c := (G_{0,1}, G_{1,0}, G_{1,1})$
5. $F := (n, m, q, in_1, in_2, \vec{G}), e := \{W_i^0, W_i^1\}_{i \in [n]}, d := \{\lambda_{n+q-m+i}\}_{i \in [m]}$
6. 输出 (F, e, d)

GRR3

$\mathrm{Ev}(F, X)$:

1. $(n, m, q, in_1, in_2, \vec{G}) \leftarrow F$
2. $(W_1, \ldots, W_n) \leftarrow X$
3. 对每个门 $c \in [n+1, n+q]$:
   $a \leftarrow in_1(c), b \leftarrow in_2(c), \hat{a}, \hat{b} \leftarrow lsb(W_a), lsb(W_b), (G_{0,1}, G_{1,0}, G_{1,1}) \leftarrow \vec{G}_c$.
   如果 $G = XOR$:
   $$W_c := W_a \oplus W_b$$
   如果 $G = AND$:
   如果 $\hat{a} = \hat{b} = 0$:
   $$W_c := H(W_a, W_b, c)$$
   $$W_c := H(W_a, W_b, c) \oplus G_{\hat{a},\hat{b}}$$
4. 输出 $Y := \{W_{n+q-m+i}\}_{i \in [m]}$

GRR2

当不需要与 Free-XOR 技术兼容之后，我们对每条线的两个 label 都有了独立选择的“权利”，利用这一点可以扩展 GRR 技术。构造分为奇门和偶门两种情况，这里奇门指的是真值表中 1 的个数为奇数，如 AND 门，OR 门；偶门指的是真值表中 1 的个数为偶数，如 XOR 门。除此之外，我们把所有密钥看成有限域 $\mathbb{F}_{2^\kappa}$ 中的元素。

GRR2

奇门：以 AND 门为例，当真值 $z_a = z_b = 1$ 时，我们希望求值者能求得输出密钥 C_1。也就是说，希望 garbled table 的第 $(\bar{\lambda}_a, \bar{\lambda}_b)$ 这一行对应的密钥是 C_1，而对于其他三行，希望求值者得到 C_0。现在为每行定义一个密钥和一个掩盖比特：

$K_1||M_1 := H(A_{\lambda_a}||B_{\lambda_b}||c) \in \{0,1\}^{\kappa+1}$

$K_2||M_2 := H(A_{\lambda_a}||B_{\bar{\lambda}_b}||c) \in \{0,1\}^{\kappa+1}$

$K_3||M_3 := H(A_{\bar{\lambda}_a}||B_{\lambda_b}||c) \in \{0,1\}^{\kappa+1}$

$K_4||M_4 := H(A_{\bar{\lambda}_a}||B_{\bar{\lambda}_b}||c) \in \{0,1\}^{\kappa+1}$

注：这 4 个密钥其实就是前面方案构造 garbled table 时的密钥值。现把它们看成有限域中的元素。我们把行数的二进制转换记为 $r(\lambda_1, \lambda_2) = 2\lambda_1 + \lambda_2 + 1$。

GRR2

定义一个 $\mathbb{F}_{2^\kappa}$ 上的二次多项式 $p(x)$，利用拉格朗日插值，使这个多项式过点 $(r(\lambda_a,\lambda_b),K_{r(\lambda_a,\lambda_b)}),(r(\lambda_a,\bar{\lambda}_b),K_{r(\lambda_a,\bar{\lambda}_b)}),(r(\bar{\lambda}_a,\lambda_b),K_{r(\bar{\lambda}_a,\lambda_b)})$，然后定义 $C_0 := p(0)$。除此之外，计算 $K_5 := p(5)$，$K_6 := p(6)$。接下来定义另一个 $\mathbb{F}_{2^\kappa}$ 上的二次多项式 $q(x)$，同样利用拉格朗日插值，过点 $(r(\bar{\lambda}_a,\bar{\lambda}_b),K_{r(\bar{\lambda}_a,\bar{\lambda}_b)}),(5,K_5),(6,K_6)$。此时定义 $C_1 := q(0)$。此时我们把新的 garbled table 表示为：(K_5,K_6)，且原始的 garbled table 中的置换比特在由掩盖比特掩盖后也按原来的顺序放入新 garbled table 中，即令 $\{\lambda_1,\lambda_2,\lambda_3,\lambda_4\} :=$
$\{M_1 \oplus \lambda_c \oplus \lambda_a \wedge \lambda_b, M_2 \oplus \lambda_c \oplus \lambda_a \wedge \bar{\lambda}_b, M_3 \oplus \lambda_c \oplus \bar{\lambda}_a \wedge \lambda_b, M_4 \oplus \lambda_c \oplus \bar{\lambda}_a \wedge \bar{\lambda}_b\}$。也就是说，新的 garbled table 总共 $2n+4$ 比特。

GRR2

当求值者想要求 garbled table 时，首先得到了输入 label 为 A, B，令 $\hat{a} := lsb(A) = \lambda_a \oplus z_a, \hat{b} := lsb(B) = \lambda_b \oplus z_b$。然后求值者计算行数 $row := 2\hat{a} + \hat{b} + 1$ 以及对应密钥和掩盖比特 $K_{row}||M_{row} := H(A||B||c)$。然后进行拉格朗日插值 $(row, K_{row}), (5, K_5), (6, K_6)$。可以发现，如果 $z_a = z_b = 1$，则 $row = r(\bar{\lambda}_a, \bar{\lambda}_b)$，$K_{row} = K_{r(\bar{\lambda}_a, \bar{\lambda}_b)}$，此时插值得到的多项式是 $q(x)$，则求值者令 $C := q(0) = C_1$。而对于其他三种情况，显然有
$(row, K_{row}) \in \{(r(\lambda_a, \lambda_b), K_{r(\lambda_a, \lambda_b)}), (r(\lambda_a, \bar{\lambda}_b), K_{r(\lambda_a, \bar{\lambda}_b)}), (r(\bar{\lambda}_a, \lambda_b), K_{r(\bar{\lambda}_a, \lambda_b)})\}$
因此插值得到的多项式是 $p(x)$，此时求值者令 $C := p(0) = C_0$。
对于置换比特，求值者令 $\hat{c} := M_{row} \oplus \lambda_{row} = \lambda_c \oplus (\lambda_a \oplus \hat{a}) \wedge (\lambda_b \oplus \hat{b}) = \lambda_c \oplus z_c$
以上就是奇门的 garbled table 构造和求值方法。

GRR2

偶门：以 XOR 门为例，当真值 $z_a = z_b = 0$、1 时，我们希望求值者能求得输出密钥 C_0。也就是说，希望 garbled table 的第 (λ_a, λ_b)，$(\bar{\lambda}_a, \bar{\lambda}_b)$ 这两行对应的密钥是 C_0，而对于其他两行，希望求值者得到 C_1。每行的密钥和掩盖比特和奇门相同：

$K_1||M_1 := H(A_{\lambda_a}||B_{\lambda_b}||c) \in \{0,1\}^{\kappa+1}$

$K_2||M_2 := H(A_{\lambda_a}||B_{\bar{\lambda}_b}||c) \in \{0,1\}^{\kappa+1}$

$K_3||M_3 := H(A_{\bar{\lambda}_a}||B_{\lambda_b}||c) \in \{0,1\}^{\kappa+1}$

$K_4||M_4 := H(A_{\bar{\lambda}_a}||B_{\bar{\lambda}_b}||c) \in \{0,1\}^{\kappa+1}$

GRR2

此时定义一个线性多项式 $p(x)$，利用拉格朗日插值，过点 $(r(\lambda_a,\lambda_b),K_{r(\lambda_a,\lambda_b)})$，$(r(\bar{\lambda}_a,\bar{\lambda}_b),K_{r(\bar{\lambda}_a,\bar{\lambda}_b)})$，令 $C_0 := p(0)$，并计算 $p(5)$，当 $\lambda_c = 0$ 时，把 $p(5)$ 放在第一行，否则放在第二行。同样定义另一个线性多项式 $q(x)$，利用拉格朗日插值，过点 $(r(\lambda_a,\bar{\lambda}_b),K_{r(\lambda_a,\bar{\lambda}_b)}),(r(\bar{\lambda}_a,\lambda_b),K_{r(\bar{\lambda}_a,\lambda_b)})$，令 $C_1 := q(0)$，同样计算 $q(5)$，并放到另一行。除此之外新的 garbled table 也包含 4 个掩盖过的置换比特，和奇门相同，即 $\{\lambda_1,\lambda_2,\lambda_3,\lambda_4\} :=$ $\{M_1\oplus\lambda_c\oplus(\lambda_a\oplus\lambda_b),M_2\oplus\lambda_c\oplus(\lambda_a\oplus\bar{\lambda}_b),M_3\oplus\lambda_c\oplus(\bar{\lambda}_a\oplus\lambda_b),M_4\oplus\lambda_c\oplus(\bar{\lambda}_a\oplus\bar{\lambda}_b)\}$。新的 garbled table 共 $2n+4$ 比特。即新的 garbled table 为 $\{K_5,K_5',\lambda_1,\lambda_2,\lambda_3,\lambda_4\}$。这里

$$K_5 = \begin{cases} p(5) & \lambda_c = 0 \\ q(5) & \lambda_c = 1 \end{cases} \qquad K_5' = \begin{cases} q(5) & \lambda_c = 0 \\ p(5) & \lambda_c = 1 \end{cases}$$

GRR2

当求值者想要求 garbled table 时，首先得到了输入 label 为 A, B，令 $\hat{a} := lsb(A) = \lambda_a \oplus z_a, \hat{b} := lsb(B) = \lambda_b \oplus z_b$。然后求值者计算行数 $row := 2\hat{a} + \hat{b} + 1$ 以及对应密钥和掩盖比特 $K_{row}||M_{row} := H(A||B||c)$。接着求输出线的置换比特 $\hat{c} := M_{row} \oplus \lambda_{row} = \lambda_c \oplus \lambda_a \oplus \hat{a} \oplus \lambda_b \oplus \hat{b} = \lambda_c \oplus z_c$。当 $\hat{c} = 0$ 时，求值者对取 garbled table 的第一行进行插值 $(row, K_{row}), (5, K_5)$，得到多项式 $s(x)$，并令 $C := s(0)$。当 $\hat{c} = 1$ 时，求值者对取 garbled table 的第二行进行插值 $(row, K_{row}), (5, K_5')$，得到多项式 $s(x)$，并令 $C := s(0)$。

这是因为，当 $z_a = z_b = 0$、1 时，有 $(row, K_{row}) \in \{(r(\lambda_a, \lambda_b), K_{r(\lambda_a,\lambda_b)}), (r(\bar{\lambda}_a, \bar{\lambda}_b), K_{r(\bar{\lambda}_a,\bar{\lambda}_b)})\}$，此时 $\hat{c} = \lambda_a \oplus z_a \oplus \lambda_b \oplus z_b = \lambda_a \oplus \lambda_b = \lambda_c$。而 $\lambda_c \oplus z_c = 0$ 时，$K_5 = p(5)$，$\lambda_c \oplus z_c = 1$ 时，$K_5' = p(5)$，所以插值得到的多项式是 $p(x)$，最终得到的密钥是 C_0。同理，当 $z_a \neq z_b$ 时，最终得到的密钥是 C_1。

Half Gate

半门的思想是将一个 AND 门分成两个半门构造 garbled table，每个半门 2 行密文，通过 GRR 技术约减为 1 个，再把两个半门的 garbled table 合成一个，总共需要 2 行密文。这里半门的意思是这一个门的两个输入，其中一个值已经知道了，只有另一个值是未知的。根据知道值的人不同，将半门分成两类，即生成者半门 (generator half-gate) 和求值者半门 (evaluator half-gate)。

Generator Half Gate

生成者半门的 **garbled table** 构造：
对于 AND 门 $G=(a,b,c,\wedge)$，真值关系为 $z_a \wedge z_b = z_c$。
假设生成者知道 z_a，那么它就可以根据 z_a 来构造 garbled table。思路是如果 $z_a=0$，则要让求值者 (有线 b 上的 label B) 只能得到 C_0，如果 $z_a=1$，则要让求值者得到 C_{z_b}。因此令 garbled table 为：
$H(B_0) \oplus C_0$
$H(B_1) \oplus C_0 \oplus z_a\Delta$
接下来对这两行密文根据 λ_b 做一个置换，即令：
$G_0^g := H(B_{\lambda_b}) \oplus C_0 \oplus \lambda_b z_a \Delta$
$G_1^g := H(B_{\bar{\lambda}_b}) \oplus C_0 \oplus \bar{\lambda}_b z_a \Delta$

Generator Half Gate

求值者求值时，得到 $B = B_0 \oplus z_b\Delta$，令 $\hat{b} = lsb(B) = \lambda_b \oplus z_b$，选择第 $\hat{b}$ 行解密，即令 $C := H(B) \oplus G^g_{\hat{b}} = H(B) \oplus H(B_{\lambda_b \oplus \hat{b}}) \oplus C_0 \oplus (\lambda_b \oplus \hat{b}) z_a\Delta = C_0 \oplus z_a z_b \Delta = C_0 \oplus z_c\Delta$

接下来运用 GRR 技术，令 $G^g_0 = 0$，即令 $C_0 := H(B_{\lambda_b}) \oplus \lambda_b z_a \Delta$

此时生成者半门的密文即为：

$G^g := H(B_{\bar{\lambda}_b}) \oplus H(B_{\lambda_b}) \oplus \lambda_b z_a \Delta \oplus \bar{\lambda}_b z_a \Delta = H(B_0) \oplus H(B_1) \oplus z_a\Delta$

Evaluator Half Gate

求值者半门的 garbled table 构造：

对于 AND 门 $G=(a,b,c,\wedge)$，真值关系为 $z_a \wedge z_b = z_c$。

现在假设求值者知道 z_a，那么生成者可以根据求值者知道 z_a 这一信息来构造 garbled table。思路是当求值者知道 z_a 时，如果 $z_a=0$，则求值者只能得到 C_0；如果 $z_a=1$，求值者只需得到 $C_0 \oplus B_0$ 就够了，这是因为，求值者知道线 b 上的 label，B_{z_b}，与这个值异或后即可得到 $B_{z_b} \oplus C_0 \oplus B_0 = C_{z_b}$。因此，生成者令 garbled table 为：

$G_0^e := H(A_0) \oplus C_0$

$G_1^e := H(A_1) \oplus C_0 \oplus B_0$

Evaluator Half Gate

求值者求值时，得到 $A = A_{z_a}$，$B = B_{z_b}$ 时，选择第 z_a 行用 A 解密，再把结果异或 $z_a B$，即令 $C := H(A) \oplus G^e_{z_a} \oplus z_a B = H(A_{z_a}) \oplus H(A_{z_a}) \oplus C_0 \oplus z_a B_0 \oplus z_a B_{z_b} = C_0 \oplus z_a z_b \Delta$。
接下来运用 GRR 技术，令 $c^e_0 = 0$，即令 $C_0 := H(A_0)$。
此时求值者半门的密文即为：
$c^e := H(A_1) \oplus H(A_0) \oplus B_0$

Two Half Make a Whole

两个半门生成完整 AND 门的 garbled table 构造：

实际构造 AND 门的时候，有关系 $z_a \wedge z_a = z_c$。这里的 z_a 和 z_b 都是未知的。那么如何构造两个半门让两方正好知道其中一个值呢？回忆点置换技术，生成者在生成 garbled circuit 时，会为每条线选一个置换比特，而求值者求值时会知道每个 label 的最后一比特 (即 $z \oplus \lambda$)，此时我们把等式写成这样：

$$
\begin{aligned}
z_c &= z_a \wedge z_b \\
&= z_a \wedge (z_b \oplus \lambda_b \oplus \lambda_b) \\
&= (z_a \wedge \lambda_b) \oplus (z_a \wedge (z_b \oplus \lambda_b))
\end{aligned}
$$

可以发现，前半部分生成者知道 λ_b，后半部分求值者知道 $z_b \oplus \lambda_b$(即 $\hat{b}$)，然后利用 Free-XOR 技术把这两个半门连接起来即可。

Two Half Make a Whole

对生成者半门 $z_a \wedge \lambda_b$ 的构造为:
$G^g := H(A_0) \oplus H(A_1) \oplus \lambda_b \Delta$
$C1_0 := H(A_{\lambda_a}) \oplus \lambda_a \lambda_b \Delta$
对求值者半门 $z_a \wedge (z_b \oplus \lambda_b)$ 的构造为:
$G^e := H(B_1) \oplus H(B_0) \oplus A_0$
$C2_0 := H(B_{\lambda_b})$
利用 Free-XOR 技术，令最终的 $C_0 := C1_0 \oplus C2_0 = H(A_{\lambda_a}) \oplus H(B_{\lambda_b}) \oplus \lambda_a \lambda_b \Delta$
因此半门技术的 garbled table 构造为:
$G_0 := H(A_0) \oplus H(A_1) \oplus \lambda_b \Delta$
$G_1 := H(B_1) \oplus H(B_0) \oplus A_0$
$C_0 := H(A_{\lambda_a}) \oplus H(B_{\lambda_b}) \oplus \lambda_a \lambda_b \Delta$

Two Half Make a Whole

求值者求值时，得到 $A = A_{z_a}, B = B_{z_b}$，令 $\hat{a} := lsb(A) = z_a \oplus \lambda_a, \hat{b} := lsb(B) = z_b \oplus \lambda_b$。求值公式为：

$$\begin{aligned}
C :&= H(A) \oplus H(B) \oplus \hat{a}G_0 \oplus \hat{b}(G_1 \oplus A) \\
&= H(A_{z_a}) \oplus H(B_{z_b}) \oplus \hat{a}(H(A_0) \oplus H(A_1) \oplus \lambda_b \Delta) \oplus \hat{b}(H(B_1) \oplus H(B_0) \oplus A_0 \oplus A_{z_a}) \\
&= H(A_{\lambda_a}) \oplus H(B_{\lambda_b}) \oplus \hat{a}\lambda_b \Delta \oplus \hat{b} z_\alpha \Delta \\
&= H(A_{\lambda_a}) \oplus H(B_{\lambda_b}) \oplus \lambda_a \lambda_b \Delta \oplus z_a z_b \Delta \\
&= C_0 \oplus z_c \Delta
\end{aligned}$$

Half Gate

$\mathrm{Gb}(1^\kappa, f)$:

1. $(n, m, q, in_1, in_2, G) \leftarrow f$
2. $\Delta \leftarrow \{0,1\}^{\kappa-1}1$
3. 对每条输入线 $i \in [n]$: $\lambda_i \leftarrow \{0,1\}, W_i^0 \leftarrow \{0,1\}^{\kappa-1}\lambda_i, W_i^1 := W_i^0 \oplus \Delta$
4. 对每个门 $c \in [n+1, n+q]$:
   $a \leftarrow in_1(c), b \leftarrow in_2(c), A_0, A_1, B_0, B_1 \leftarrow W_a^0, W_a^1, W_b^0, W_b^1, \lambda_a, \lambda_b \leftarrow lsb(A_0), lsb(B_0)$,
   如果 $G = XOR$:
   $$W_c^0 = C_0 := A_0 \oplus B_0, W_c^1 = C_1 := W_c^0 \oplus \Delta$$
   如果 $G = AND$:
   $$W_c^0 = C_0 := H(A_{\lambda_a}) \oplus H(B_{\lambda_b}) \oplus \lambda_a\lambda_b\Delta, W_c^1 = C_1 := C_0 \oplus \Delta$$
   $$G_0 = H(A_0) \oplus H(A_1) \oplus \lambda_b\Delta$$
   $$G_1 = H(B_0) \oplus H(B_1) \oplus \lambda_a\lambda_b\Delta$$
   $\vec{G}_c := (G_0, G_1)$
5. $F := (n, m, q, in_1, in_2, \vec{G}), e := \{W_i^0, W_i^1\}_{i\in[n]}, d := \{\lambda_{n+q-m+i}\}_{i\in[m]}$
6. 输出 (F, e, d)

Half Gate

$\mathrm{Ev}(F, X)$:

1. $(n, m, q, in_1, in_2, \vec{G}) \leftarrow F$
2. $(W_1, \ldots, W_n) \leftarrow X$
3. 对每个门 $c \in [n+1, n+q]$:
   $a \leftarrow in_1(c), b \leftarrow in_2(c), \hat{a}, \hat{b} \leftarrow lsb(W_a), lsb(W_b), (G_0, G_1) \leftarrow \vec{G}_c$.
   如果 $G = XOR$:
   $$W_c := W_a \oplus W_b$$
   如果 $G = AND$:
   $$W_c := H(W_a) \oplus H(W_b) \oplus \hat{a}G_0 \oplus \hat{b}(G_1 \oplus A)$$
4. 输出 $Y := \{W_{n+q-m+i}\}_{i \in [m]}$

Linear Garbling Scheme

这里描述一下 [ZRE15] 给出的线性 garbled 方案的抽象。以上所有的改进都可以看成一般的线性 garble 方案的一个特例。这里线性 garble 方案是指所有的构造 garbled table 和求值的操作都是线性操作，都可以用矩阵乘法表示。

为了方便描述，这里转换一下符号的含义，用 A_0, A_1, B_0, B_1 表示最后一比特分别是 0/1 的输入 label。此时，$A_{\lambda_a}, B_{\lambda_b}$ 表示真值为 0 的 label。而为了不引入 λ_c，输出 label 还是用下标表示真值，即 C_0 表示真值为 0 的输出 label。（如果也用 C_{λ_c} 表示 0 的 label，则 $C_0 = C_{\lambda_c} \oplus \lambda_c \Delta$ 会引入 λ_c 这个系数，表示真值 0 的输出 label 是不需要 λ_c 这个参数的。）

Linear Garbling Scheme

以 AND 门为例，最基本的 garbled circuit 加点置换技术的方案表示如下（红字代表变量）：

Gb：

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

Linear Garbling Scheme

Ev：有 $A_{\hat{a}} = A_{z_a \oplus \lambda_a}, B_{\hat{b}} = B_{z_b \oplus \lambda_b}$，求 C_{z_c}

$$C_{z_c} = \begin{pmatrix} 0 & 0 & 1 & \bar{\hat{a}} \cdot \bar{\hat{b}} & \bar{\hat{a}} \cdot \hat{b} & \hat{a} \cdot \bar{\hat{b}} & \hat{a} \cdot \hat{b} \end{pmatrix} \cdot \begin{pmatrix} A_{\hat{a}} \\ B_{\hat{b}} \\ H(A_{\hat{a}} || B_{\hat{b}}) \\ G_{0,0} \\ G_{0,1} \\ G_{1,0} \\ G_{1,1} \end{pmatrix}$$

Linear Garbling Scheme

遍历 $\hat{a}\hat{b} = 00, 01, 10, 11$，得：

$$
\begin{aligned}
C_{\lambda_a\lambda_b} &= H(A_0||B_0) \oplus G_{0,0} \\
C_{\lambda_a\overline{\lambda_b}} &= H(A_0||B_1) \oplus G_{0,1} \\
C_{\overline{\lambda_a}\lambda_b} &= H(A_1||B_0) \oplus G_{1,0} \\
C_{\overline{\lambda_a\lambda_b}} &= H(A_1||B_1) \oplus G_{1,1}
\end{aligned}
$$

Linear Garbling Scheme

写成矩阵形式

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

Linear Garbling Scheme

Free-XOR：
采用 Free-XOR 的 AND 门表示如下：
Gb：

$$\begin{pmatrix} {\color{red}C_0} \\ {\color{red}C_1} \\ {\color{red}G_{0,0}} \\ {\color{red}G_{0,1}} \\ {\color{red}G_{1,0}} \\ {\color{red}G_{1,1}} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & \lambda_a\lambda_b & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & \lambda_a\overline{\lambda_b} & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & \overline{\lambda_a}\lambda_\beta & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & \overline{\lambda_a\lambda_b} & 1 \end{pmatrix} \cdot \begin{pmatrix} H(A_0||B_0) \\ H(A_0||B_1) \\ H(A_1||B_0) \\ H(A_1||B_1) \\ A_0 \\ B_0 \\ \Delta \\ {\color{red}C_0} \end{pmatrix}$$

Linear Garbling Scheme

Ev：有 $A_{\hat{a}} = A_{z_a \oplus \lambda_a}, B_{\hat{b}} = B_{z_b \oplus \lambda_b}$，求 C_{z_c}

$$C_{z_c} = \begin{pmatrix} 0 & 0 & 1 & \bar{\hat{a}} \cdot \bar{\hat{b}} & \bar{\hat{a}} \cdot \hat{b} & \hat{a} \cdot \bar{\hat{b}} & \hat{a} \cdot \hat{b} \end{pmatrix} \cdot \begin{pmatrix} A_{\hat{a}} \\ B_{\hat{b}} \\ H(A_{\hat{a}} || B_{\hat{b}}) \\ G_{0,0} \\ G_{0,1} \\ G_{1,0} \\ G_{1,1} \end{pmatrix}$$

Linear Garbling Scheme

遍历 $\hat{a}\hat{b} = 00, 01, 10, 11$，得：

$$
\begin{aligned}
C_0 \oplus \lambda_a \lambda_b \Delta &= H(A_0 || B_0) \oplus G_{0,0} \\
C_0 \oplus \lambda_a \overline{\lambda_b} \Delta &= H(A_0 || B_1) \oplus G_{0,1} \\
C_0 \oplus \overline{\lambda_a} \lambda_b \Delta &= H(A_1 || B_0) \oplus G_{1,0} \\
C_0 \oplus \overline{\lambda_a \lambda_b} \Delta &= H(A_1 || B_1) \oplus G_{1,1}
\end{aligned}
$$

Linear Garbling Scheme

写成矩阵形式

$$
\begin{pmatrix}
1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
C_0 \\ G_{0,0} \\ G_{0,1} \\ G_{1,0} \\ G_{1,1}
\end{pmatrix}
=
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0
\end{pmatrix}
\cdot
\begin{pmatrix}
H(A_0||B_0) \\ H(A_0||B_1) \\ H(A_1||B_0) \\ H(A_1||B_1) \\ A_0 \\ B_0 \\ \Delta
\end{pmatrix}
\oplus
\begin{pmatrix}
\lambda_a\lambda_b \\ \lambda_a\overline{\lambda_b} \\ \overline{\lambda_a}\lambda_b \\ \overline{\lambda_a\lambda_b}
\end{pmatrix}.
$$

Linear Garbling Scheme

Free-XOR+GRR3：
采用 Free-XOR 和 GRR3 的 AND 门表示如下：
Gb：

$$\begin{pmatrix} C_0 \\ C_1 \\ G_{0,1} \\ G_{1,0} \\ G_{1,1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \lambda_a\lambda_b \\ 1 & 0 & 0 & 0 & 0 & 0 & \overline{\lambda_a\lambda_b} \\ 1 & 1 & 0 & 0 & 0 & 0 & \lambda_a \\ 1 & 0 & 1 & 0 & 0 & 0 & \lambda_b \\ 1 & 0 & 0 & 1 & 0 & 0 & \lambda_a \oplus \overline{\lambda_b} \end{pmatrix} \cdot \begin{pmatrix} H(A_0||B_0) \\ H(A_0||B_1) \\ H(A_1||B_0) \\ H(A_1||B_1) \\ A_0 \\ B_0 \\ \Delta \end{pmatrix}$$

Linear Garbling Scheme

Ev：有 $A_{\hat{a}} = A_{z_a \oplus \lambda_a}, B_{\hat{b}} = B_{z_b \oplus \lambda_b}$，求 C_{z_c}

$$C_{z_c} = \begin{pmatrix} 0 & 0 & 1 & \bar{\hat{a}} \cdot \hat{b} & \hat{a} \cdot \bar{\hat{b}} & \hat{a} \cdot \hat{b} \end{pmatrix} \cdot \begin{pmatrix} A_{\hat{a}} \\ B_{\hat{b}} \\ H(A_{\hat{a}} || B_{\hat{b}}) \\ G_{0,1} \\ G_{1,0} \\ G_{1,1} \end{pmatrix}$$

Linear Garbling Scheme

遍历 $\hat{a}\hat{b} = 00, 01, 10, 11$，得：

$$
\begin{aligned}
C_0 \oplus \lambda_a \lambda_b \Delta &= H(A_0 || B_0) \\
C_0 \oplus \lambda_a \overline{\lambda_b} \Delta &= H(A_0 || B_1) \oplus G_{0,1} \\
C_0 \oplus \overline{\lambda_a} \lambda_b \Delta &= H(A_1 || B_0) \oplus G_{1,0} \\
C_0 \oplus \overline{\lambda_a \lambda_b} \Delta &= H(A_1 || B_1) \oplus G_{1,1}
\end{aligned}
$$

 参考文献

Linear Garbling Scheme

写成矩阵形式

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \color{red}{C_0} \\ \color{red}{G_{0,1}} \\ \color{red}{G_{1,0}} \\ \color{red}{G_{1,1}} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} H(A_0||B_0) \\ H(A_0||B_1) \\ H(A_1||B_0) \\ H(A_1||B_1) \\ A_0 \\ B_0 \\ \Delta \end{pmatrix} \oplus \begin{pmatrix} \lambda_a\lambda_b \\ \lambda_a\overline{\lambda_b} \\ \overline{\lambda_a}\lambda_b \\ \overline{\lambda_a\lambda_b} \end{pmatrix} \cdot \Delta$$

Linear Garbling Scheme

Half-gate：
采用半门的 AND 门表示如下：
Gb：

$$\begin{pmatrix} C_0 \\ C_1 \\ G_0 \\ G_1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & \lambda_a\lambda_b \\ 1 & 0 & 1 & 0 & 0 & 0 & \overline{\lambda_a\lambda_b} \\ 1 & 1 & 0 & 0 & 0 & 0 & \lambda_b \\ 0 & 0 & 1 & 1 & 1 & 0 & \lambda_a \end{pmatrix} \cdot \begin{pmatrix} H(A_0) \\ H(A_1) \\ H(B_0) \\ H(B_1) \\ A_0 \\ B_0 \\ \Delta \end{pmatrix}$$

Linear Garbling Scheme

Ev： 有 $A_{\hat{a}} = A_{z_a \oplus \lambda_a}, B_{\hat{b}} = B_{z_b \oplus \lambda_b}$, 求 C_{z_c}

$$C_{z_c} = \begin{pmatrix} \hat{b} & 0 & 1 & 1 & \hat{a} & \hat{b} \end{pmatrix} \cdot \begin{pmatrix} A_{\hat{a}} \\ B_{\hat{b}} \\ H(A_{\hat{a}}) \\ H(B_{\hat{b}}) \\ G_0 \\ G_1 \end{pmatrix}$$

Linear Garbling Scheme

遍历 $\hat{a}\hat{b} = 00, 01, 10, 11$，得：

$$
\begin{aligned}
C_0 \oplus \lambda_a \lambda_b \Delta &= H(A_0) \oplus H(B_0) \\
C_0 \oplus \lambda_a \overline{\lambda_b} \Delta &= H(A_0) \oplus H(B_1) \oplus A_0 \oplus G_1 \\
C_0 \oplus \overline{\lambda_a} \lambda_b \Delta &= H(A_1) \oplus H(B_0) \oplus G_0 \\
C_0 \oplus \overline{\lambda_a \lambda_b} \Delta &= H(A_1) \oplus H(B_1) \oplus A_0 \oplus \Delta \oplus G_0 \oplus G_1
\end{aligned}
$$

Linear Garbling Scheme

写成矩阵形式

$$
\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}
\begin{pmatrix} \color{red}{C_0} \\ \color{red}{G_0} \\ \color{red}{G_1} \end{pmatrix}
=
\left(\begin{array}{cccc|ccc}
1 & 0 & 1 & 0 & 0 & 0 & \lambda_a\lambda_b \\
1 & 0 & 0 & 1 & 1 & 0 & \lambda_a\overline{\lambda_b} \\
0 & 1 & 1 & 0 & 0 & 0 & \overline{\lambda_a}\lambda_b \\
0 & 1 & 0 & 1 & 1 & 0 & 1 \oplus \overline{\lambda_a\lambda_b}
\end{array}\right)
\cdot
\begin{pmatrix} H(A_0) \\ H(A_1) \\ H(B_0) \\ H(B_1) \\ A_0 \\ B_0 \\ \Delta \end{pmatrix}
$$

Slicing & Dicing

Rosulek 和 Roy[RR21] 提出了 Slicing & Dicing 技术，进一步将 AND 门的通信量降低到 1.5κ，且与 Free-XOR 兼容，是目前通信最低的 GC 方案。

Slicing & Dicing

考虑半门构造:

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \color{red}{C_0} \\ \color{red}{G_0} \\ \color{red}{G_1} \end{pmatrix} = \left(\begin{array}{cccc|ccc} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{array}\right) \cdot \begin{pmatrix} H(A_0) \\ H(A_1) \\ H(B_0) \\ H(B_1) \\ A_0 \\ B_0 \\ \Delta \end{pmatrix} \oplus \begin{pmatrix} \lambda_a \lambda_b \\ \lambda_a \overline{\lambda_b} \\ \overline{\lambda_a} \lambda_b \\ \overline{\lambda_a}\,\overline{\lambda_b} \end{pmatrix} \cdot \Delta$$

Slicing & Dicing

$$
\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}
\begin{pmatrix} \color{red}{C_0} \\ \color{red}{G_0} \\ \color{red}{G_1} \end{pmatrix}
=
\left(
\left(\begin{array}{cccc|ccc} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{array}\right)
\oplus
\left(\begin{array}{cccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & \lambda_a\lambda_b \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_a\overline{\lambda_b} \\ 0 & 0 & 0 & 0 & 0 & 0 & \overline{\lambda_a}\lambda_b \\ 0 & 0 & 0 & 0 & 0 & 0 & \overline{\lambda_a\lambda_b} \end{array}\right)
\right)
\cdot
\begin{pmatrix} H(A_0) \\ H(A_1) \\ H(B_0) \\ H(B_1) \\ A_0 \\ B_0 \\ \Delta \end{pmatrix}
$$

Slicing & Dicing

观察 1: 可以询问的不只有 $H(A), H(B)$，还有 $H(A \oplus B)$

	$H(A_0)$	$H(A_1)$	$H(B_0)$	$H(B_1)$	$H(A_0 \oplus B_0)$	$H(A_0 \oplus B_1)$
gate input (0,0)	✓		✓		✓	
gate input (0,1)	✓			✓		✓
gate input (1,0)		✓	✓			✓
gate input (1,1)		✓		✓	✓	

图 2: 观察 1

Slicing & Dicing

观察 2: 可以将 label 划分为左半和右半，求值者对每一半使用不同的方程求解（此时每个门的求值由四个方程变成八个方程）：

	$H(A_0)$	$H(A_1)$	$H(B_0)$	$H(B_1)$	$H(A_0 \oplus B_0)$	$H(A_0 \oplus B_1)$
(0,0) left	✓				✓	
(0,0) right			✓		✓	
(0,1) left	✓					✓
(0,1) right				✓		✓
(1,0) left		✓				✓
(1,0) right			✓			✓
(1,1) left		✓			✓	
(1,1) right				✓	✓	

图 3: 观察 2

Slicing & Dicing

观察 3: 随机化并隐藏求值者系数

$$
\left[\begin{array}{ccccc}
1&0&?&?&?\\
0&1&?&?&?\\ \hline
1&0&?&?&?\\
0&1&?&?&?\\ \hline
1&0&?&?&?\\
0&1&?&?&?\\ \hline
1&0&?&?&?\\
0&1&?&?&?
\end{array}\right]
\begin{bmatrix} C_L\\ C_R\\ G_0\\ G_1\\ G_2 \end{bmatrix}
=
\left(
\left[\begin{array}{cccccccccccc}
1&0&0&0&1&0&?&?&?&?&?&?\\
0&0&1&0&1&0&?&?&?&?&?&?\\ \hline
1&0&0&0&0&1&?&?&?&?&?&?\\
0&0&0&1&0&1&?&?&?&?&?&?\\ \hline
0&1&0&0&0&1&?&?&?&?&?&?\\
0&0&1&0&0&1&?&?&?&?&?&?\\ \hline
0&1&0&0&1&0&?&?&?&?&?&?\\
0&0&0&1&1&0&?&?&?&?&?&?
\end{array}\right]
\oplus
\left[\begin{array}{c|c}
\begin{matrix}0&\cdots&0\\ \vdots&&\vdots\\ \vdots&&\vdots\\ \vdots&&\vdots\\ \vdots&&\vdots\\ \vdots&&\vdots\\ \vdots&&\vdots\\ 0&\cdots&0\end{matrix}
&
\underbrace{\begin{array}{cc}
0&0\\
0&0\\ \hline
0&0\\
0&0\\ \hline
1&0\\
0&1\\ \hline
0&0\\
0&0
\end{array}}_{t}
\end{array}\right]
\right)
\begin{bmatrix}
H(A_0)\\ H(A_1)\\ H(B_0)\\ H(B_1)\\ H(A_0\oplus B_0)\\ H(A_0\oplus B_1)\\ A_{0L}\\ A_{0R}\\ B_{0L}\\ B_{0R}\\ \Delta_L\\ \Delta_R
\end{bmatrix}
$$

图 4: 目前的结构

Slicing & Dicing

需要确定"?" 处的值，使得不管 t 怎么选，两边的矩阵具有相同的列空间：

- 右侧的列（表示 H 输出）已经张成了 5 维的空间，因此只能将左侧矩阵扩展到该空间的基
- 右侧的"?" 项受其他约束，因此它们反映了求值者在每个输入组合中实际可以做什么。例如，在输入 A_0, B_1 上，求值者不能在其线性组合中包含 B_{0R}，只能包含 $B_{1R} = B_{0R} \oplus \Delta_R$。

Slicing & Dicing

对每个 t 的取值，都可以找到对应的"?" 项，但是不存在对任意 t 都满足限制的"?" 项。

- Garbler 先根据 t 生成对应的矩阵 (此时这个矩阵依赖 t)，再对这个矩阵进行加密
- Garbler 令矩阵的未知值服从某一分布，使得在只知道一组 label 的情况下，其他三组值与 t 无关
- 此时 Evaluator 求值之前需要先根据自己的 label 计算需要做的线性组合比特，称为控制比特 (control bits)，再根据控制比特做线性组合求解 garbled table

Slicing & Dicing
矩阵求解

使用符号表示如下：

$$V\begin{bmatrix} C \\ \vec{G} \end{bmatrix} := M\vec{H} \oplus \left((R \oplus [0 \ \ldots \ |t]) \begin{bmatrix} A_0 \\ B_0 \\ \Delta \end{bmatrix} \right)$$

其中 $C = [C_L, C_R]^T, \vec{G} = [G_0, G_1, G_2]^T$,
$\vec{H} = [H(A_0), H(A_1), H(B_0), H(B_1), H(A_0 \oplus B_0), H(A_0 \oplus B_1)]^T$
$R_{8\times 6}$：控制矩阵 (control matrix)

Slicing & Dicing

矩阵求解

两边的系数矩阵需要具有相同的列空间，而 M 的列秩已经是 5，有如下关系成立：

$$\mathcal{G} = colspace(V) = colspace(M) \supseteq colspace(R \oplus [0 \ \ldots \ |t|])$$

方便起见，将 $\mathcal{G}$ 看作一组线性限制条件，而不是 M 的列空间。用 K 表示 M 的 kernel 的一组基，使得任意 $\mathcal{G}$ 中的向量 v 满足 $v \in \mathcal{G} \iff Kv = 0$，则 V 满足 $rank(V) = 5$ 且 $KV = 0$。任意满足上述条件的 K, V 均可，使用如下矩阵：

$$K = \left(\begin{array}{cc|cc|cc|cc} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{array}\right) V = \left(\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{array}\right)$$

Slicing & Dicing

矩阵求解

下面求控制矩阵 R:

根据关系 $KR = K[0\dots0|t] = \left(\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & p & 0 \\ 0 & 0 & 0 & 0 & 0 & p \\ 0 & 0 & 0 & 0 & a & b \end{array}\right)$

- p 表示门的奇偶性 (AND:$p = 1$; XOR:$p = 0$)
- a, b 泄露 t 中 1 的位置 (AND:$a = \bar{\lambda}_a, b = \bar{\lambda}_b$;XOR:$a = b = 1$)

要隐藏 a, b 的值，需要随机化控制矩阵 R

Slicing & Dicing
矩阵求解

此外，由于求值需要，R 具有以下分块形式:

$$R = \left(\begin{array}{c|c|c} R_{00A} & R_{00B} & 0 \\ R_{01A} & R_{01B} & R_{01B} \\ R_{10A} & R_{10B} & R_{10A} \\ R_{11A} & R_{11B} & R_{11A} \oplus R_{11bB} \end{array}\right)_{8\times 6}$$

当求值者持有输入 label A_i, B_j 时，子矩阵 $R_{ij} = [R_{ijA} R_{ijB}]$ 足以完全确定应使用的线性组合。称 R_{ij} 为该输入组合的边际视图 (marginal view)。

Slicing & Dicing

矩阵求解

目标：

- 根据上面列出的约束条件随机化 R 的选择
- 任何单个边际视图都不会泄漏关于 t 的任何信息

即，希望找到一个分布 $\mathcal{R}(t)$，使得当 $R \leftarrow \mathcal{R}(t)$，$KR = K[0 \ldots 0|t]$ 概率为 1，但对于每个 $i,j \in \{0,1\}$，如果 $t \leftarrow T, R \leftarrow \mathcal{R}(t)$，则 t 和 R_{ij} 分布互相独立。

Slicing & Dicing
矩阵求解

求 $\mathcal{R}(t)$ 思路：

1. 先找分布 $\mathcal{R}_0$，满足：
   - 对 $R_\$ \leftarrow \mathcal{R}_0, KR_\$ = 0$
   - 对每个 $i,j \in \{0,1\}$，如果 $R_\$ \leftarrow \mathcal{R}_0$，则 $(R_\$)_{ij}$ 是均匀的 (uniform)
2. 确定常数矩阵 R_p, R_a, R_b，满足

$$KR_p = \left(\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) KR_a = \left(\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right) KR_b = \left(\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

3. 定义 $\mathcal{R}(t)$：先采样 $R_\$ \leftarrow \mathcal{R}_0$，令 $R := pR_p \oplus aR_a \oplus bR_b \oplus R_\$$

$$R_{\$} \leftarrow \mathrm{span}\left\{\begin{matrix}
\left[\begin{array}{cc|cc|cc} 1&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ \hline 0&0&1&0&1&0 \\ 1&0&0&0&0&0 \\ \hline 0&0&1&0&0&0 \\ 1&0&1&0&1&0 \\ \hline 1&0&0&0&1&0 \\ 0&0&1&0&1&0 \end{array}\right], &
\left[\begin{array}{cc|cc|cc} 0&1&0&0&0&0 \\ 0&0&0&0&0&0 \\ \hline 0&0&0&1&0&1 \\ 0&1&0&0&0&0 \\ \hline 0&0&0&1&0&0 \\ 0&1&0&1&0&1 \\ \hline 0&1&0&0&0&1 \\ 0&0&0&1&0&1 \end{array}\right], &
\left[\begin{array}{cc|cc|cc} 0&0&1&0&0&0 \\ 0&0&0&0&0&0 \\ \hline 0&0&0&0&0&0 \\ 1&0&0&0&0&0 \\ \hline 0&0&1&0&0&0 \\ 0&0&1&0&0&0 \\ \hline 0&0&0&0&0&0 \\ 1&0&1&0&0&0 \end{array}\right], &
\left[\begin{array}{cc|cc|cc} 0&0&0&1&0&0 \\ 0&0&0&0&0&0 \\ \hline 0&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ \hline 0&0&0&1&0&0 \\ 0&0&0&1&0&0 \\ \hline 0&0&0&0&0&0 \\ 0&1&0&1&0&0 \end{array}\right], \\
\left[\begin{array}{cc|cc|cc} 0&0&0&0&0&0 \\ 1&0&0&0&0&0 \\ \hline 0&0&1&0&1&0 \\ 0&0&0&0&0&0 \\ \hline 0&0&0&0&0&0 \\ 1&0&1&0&1&0 \\ \hline 0&0&1&0&1&0 \\ 0&0&1&0&1&0 \end{array}\right], &
\left[\begin{array}{cc|cc|cc} 0&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ \hline 0&0&0&1&0&1 \\ 0&0&0&0&0&0 \\ \hline 0&0&0&0&0&0 \\ 0&1&0&1&0&1 \\ \hline 0&0&0&1&0&1 \\ 0&0&0&1&0&1 \end{array}\right], &
\left[\begin{array}{cc|cc|cc} 0&0&0&0&0&0 \\ 0&0&1&0&0&0 \\ \hline 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ \hline 0&0&0&0&0&0 \\ 0&0&1&0&0&0 \\ \hline 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \end{array}\right], &
\left[\begin{array}{cc|cc|cc} 0&0&0&0&0&0 \\ 0&0&0&1&0&0 \\ \hline 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ \hline 0&0&0&0&0&0 \\ 0&0&0&1&0&0 \\ \hline 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \end{array}\right], \\
\left[\begin{array}{cc|cc|cc} 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ \hline 1&0&1&0&1&0 \\ 1&0&0&0&0&0 \\ \hline 0&0&1&0&0&0 \\ 1&0&1&0&1&0 \\ \hline 1&0&0&0&1&0 \\ 0&0&1&0&1&0 \end{array}\right], &
\left[\begin{array}{cc|cc|cc} 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ \hline 0&1&0&1&0&1 \\ 0&1&0&0&0&0 \\ \hline 0&0&0&1&0&0 \\ 0&1&0&1&0&1 \\ \hline 0&1&0&0&0&1 \\ 0&0&0&1&0&1 \end{array}\right], &
\left[\begin{array}{cc|cc|cc} 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ \hline 0&0&0&0&0&0 \\ 0&0&1&0&1&0 \\ \hline 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ \hline 0&0&0&0&0&0 \\ 0&0&1&0&1&0 \end{array}\right], &
\left[\begin{array}{cc|cc|cc} 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ \hline 0&0&0&0&0&0 \\ 0&0&0&1&0&1 \\ \hline 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ \hline 0&0&0&0&0&0 \\ 0&0&0&1&0&1 \end{array}\right], \\
\left[\begin{array}{cc|cc|cc} 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ \hline 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ \hline 1&0&0&0&1&0 \\ 0&0&0&0&0&0 \\ \hline 1&0&0&0&1&0 \\ 0&0&0&0&0&0 \end{array}\right], &
\left[\begin{array}{cc|cc|cc} 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ \hline 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ \hline 0&1&0&0&0&1 \\ 0&0&0&0&0&0 \\ \hline 0&1&0&0&0&1 \\ 0&0&0&0&0&0 \end{array}\right] & &
\end{matrix}\right\}$$

Slicing & Dicing

矩阵求解

$$R_p = \left(\begin{array}{cc|cc|cc} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) R_a = \left(\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{array}\right) R_b = \left(\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ \hline 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{array}\right)$$

Slicing & Dicing

矩阵压缩

如果 $\mathcal{R}_0$ 是满足 $KR = 0$ 的所有矩阵上的均匀分布，则 garbler 必须对每个完整的 8 比特边际视图 R_{ij} 进行加密。为了降低通信，可以使用一个更好的分布，使得 garbler 能够使用更少的比特加密边际视图 R_{ij}.

主要观察：每个边际视图 R_{ij} 都是一个 2×4 的矩阵，可以限制 R_{ij} 到某个线性子空间 $S = span\{S_1, \ldots, S_d\}$ 中 $(d < 8)$ 且维持原来的性质不变。可以用计算机穷搜的方法找到这个线性子空间。

Slicing & Dicing

矩阵求解

$$S_1 = \left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{array}\right] \qquad S_2 = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{array}\right]$$

$$\bar{R}_a = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \bar{R}_b = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad \bar{R}_\$ \leftarrow \operatorname{span}\left\{ \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \right\}$$

$$R_a = \left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{array}\right] \qquad R_b = \left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ \hline 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{array}\right]$$

$$R_\$ \leftarrow \operatorname{span}\left\{ \left[\begin{array}{cc|cc|cc} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ \hline 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ \hline 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{array}\right], \left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ \hline 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{array}\right] \right\}$$

图 6: 不需要对求值者隐藏门 (Parity-leaking gates) 时的压缩表示

Slicing & Dicing

矩阵求解

$$S_1 = \left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{array}\right] \quad S_2 = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{array}\right] \quad S_3 = \left[\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \quad S_4 = \left[\begin{array}{cc|cc} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right]$$

$$\bar{R}_p = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \bar{R}_\$ \leftarrow \mathrm{span}\left\{ \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \ldots \right\}$$

$$R_p = \left[\begin{array}{cc|cc|cc} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \qquad R_\$ \leftarrow \mathrm{span}\left\{ \left[\begin{array}{cc|cc|cc} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ \hline 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{array}\right], \left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right], \ldots \right\}$$

图 7: 需要对求值者隐藏门 (Parity-hiding gates) 时的压缩表示

Slicing & Dicing
混淆控制比特

目标：garbler 已知控制矩阵的压缩表示 $\bar{R}_{4\times d}$，需要像加密 garbled table 一样加密/混淆 $\bar{R}$，使得当求值者有输入 A_i, B_j 时，只能解密 $\bar{R}_{ij}$

方法：同样使用 Slicing & Dicing 技术，将 $\bar{R}$ 看作线上的 label，把 $\bar{R}_{ij}$ 切分成奇数和偶数两部分 $\bar{r}_{ijL}, \bar{r}_{ijR} \in \mathbb{F}_{2^{d/2}}$，定义 $\vec{r} := [\bar{r}_{00L}\ \bar{r}_{00R}\ \bar{r}_{01L}\ \bar{r}_{01R}\ \bar{r}_{10L}\ \bar{r}_{10R}\ \bar{r}_{11L}\ \bar{r}_{11R}]^T$(这里 $\vec{r}$ 实际上是对 $\bar{R}$ 奇偶列比特的一个重新排列)。观察发现，对于所有的 Parity-leaking gates/Parity-hiding gates 构造，这个向量均在门子空间 $\mathcal{G}$ 上，即满足 $K\vec{r} = 0$，因此可以使用如下等式混淆 $\vec{r}$:

$$V\vec{z} \oplus Mlsb_{d/2}(\vec{H}) = \vec{r}$$

注：这里假设 H 的输出为 $\kappa/2 + d/2$ 比特，前 $msb_{\kappa/2}(\vec{H})$ 比特用于混淆 label，后 $lsb_{d/2}(\vec{H})$ 用于混淆控制比特。$\vec{z}$ 是 $5 \times d/2$ 的矩阵。

Slicing & Dicing
混淆控制比特

综合之前混淆 label 的等式，有：

$$V\left(\vec{z}\,\|\begin{bmatrix} C \\ \vec{G} \end{bmatrix}\right) = M\vec{H}\oplus\left(\vec{r}\,\|(R\oplus[0\ \ldots 0\ |t])\begin{bmatrix} A_0 \\ B_0 \\ \Delta \end{bmatrix}\right)$$

结合前面的分析，使得方程有解的 $V, R, \vec{r}$ 已知，目标是求 $\vec{z}, C, \vec{G}$。每个门的 garbled table 包括 $\vec{G}, \vec{z}$，共 $3\kappa/2 + 5d/2$ 比特。

Slicing & Dicing

$\mathrm{Gb}(1^\kappa, f)$:

1. $(n, m, q, in_1, in_2, G) \leftarrow f$
2. $\Delta \leftarrow \begin{bmatrix} 1\|\mathsf{GF}(2^{\kappa/2-1}) \\ \mathsf{GF}(2^{\kappa/2}) \end{bmatrix}$
3. 对每条输入线 $i \in [n]$: $\lambda_i \leftarrow \{0,1\}, W_i^0 \leftarrow \begin{bmatrix} \lambda_i\|\mathsf{GF}(2^{\kappa/2-1}) \\ \mathsf{GF}(2^{\kappa/2}) \end{bmatrix}, W_i^1 := W_i^0 \oplus \Delta$
4. 对每个门 $c \in [n+1, n+q]$:
   $a \leftarrow in_1(c), b \leftarrow in_2(c), A_0, A_1, B_0, B_1 \leftarrow W_a^0, W_a^1, W_b^0, W_b^1, \lambda_a, \lambda_b \leftarrow lsb(A_0), lsb(B_0)$,
   如果 $G = XOR$:
   $W_c^0 = C_0 := A_0 \oplus B_0, W_c^1 = C_1 := W_c^0 \oplus \Delta$
   如果 $G = AND$:
   $t := [\lambda_a\lambda_b \cdot I_2 \quad \lambda_a\bar{\lambda}_b \cdot I_2 \quad \bar{\lambda}_a\lambda_b \cdot I_2 \quad \bar{\lambda}_a\bar{\lambda}_b \cdot I_2]^T, (R, \vec{r}) \leftarrow \mathrm{SampleR}(t, \mathrm{leak}(c))$
   $\vec{z}_c\| \begin{bmatrix} C_0 \\ \vec{G}_c \end{bmatrix} := V^{-1}M\vec{H} \oplus V^{-1}\left(\vec{r}\|(R \oplus [0 \ \ldots 0 \ |t]) \begin{bmatrix} A_{\lambda_a} \\ B_{\lambda_b} \\ \Delta \end{bmatrix}\right)$
   $W_c^0 := C_0, W_c^1 = W_c^0 \oplus \Delta$
5. $F := (n, m, q, in_1, in_2, \vec{G}, \vec{z}), e := \{W_i^0, W_i^1\}_{i\in[n]}, d := \{\lambda_{n+q-m+i}\}_{i\in[m]}$
6. 输出 (F, e, d)

Slicing & Dicing

$\mathrm{Ev}(F, X)$:

1. $(n, m, q, in_1, in_2, \vec{G}, \vec{z}) \leftarrow F$
2. $(W_1, \ldots, W_n) \leftarrow X$
3. 对每个门 $c \in [n+1, n+q]$: $a \leftarrow in_1(c), b \leftarrow in_2(c), \hat{a}, \hat{b} \leftarrow lsb(W_a), lsb(W_b)$.
   如果 $G = XOR$:
   $$W_c := W_a \oplus W_b$$
   如果 $G = AND$:
   $$\vec{r} || X_{\hat{a}\hat{b}} := V_{\hat{a}\hat{b}}\left(\vec{z}_c || \begin{bmatrix} 0 \\ \vec{G}_c \end{bmatrix}\right) \oplus \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} H(W_a) \\ H(W_b) \\ H(W_a \oplus W_b) \end{bmatrix}$$
   $$R_{\hat{a}\hat{b}} := \mathrm{DecodeR}(\vec{r}, \mathrm{leak}, \hat{a}, \hat{b})$$
   $$W_c := X_{\hat{a}\hat{b}} \oplus R_{\hat{a}\hat{b}} \begin{bmatrix} W_a \\ W_b \end{bmatrix}$$
4. 输出 $Y := \{W_{n+q-m+i}\}_{i \in [m]}$

主要参考文献

1. [KS08] Vladimir Kolesnikov and Thomas Schneider. Improved Garbled Circuit: Free XOR Gates and Applications. ICALP 2008.
2. [Pin+09] Benny Pinkas and Thomas Schneider and Nigel P. Smart and Stephen C. Williams. Secure Two-Party Computation Is Practical. ASIACRYPT 2009.
3. [Bel+13] Mihir Bellare and Viet Tung Hoang and Sriram Keelveedhi and Phillip Rogaway. Efficient Garbling from a Fixed-Key Blockcipher. S & P 2013.
4. [ZRE15] Samee Zahur and Mike Rosulek and David Evans. Two Halves Make a Whole - Reducing Data Transfer in Garbled Circuits Using Half Gates. EUROCRYPT 2015.
5. [RR21] Mike Rosulek and Lawrence Roy. Secure Three Halves Make a Whole? Beating the Half-Gates Lower Bound for Garbled Circuits. CRYPTO 2021.

参考文献 I

[BMR90] Donald Beaver, Silvio Micali, and Phillip Rogaway. "The Round Complexity of Secure Protocols (Extended Abstract)". In: *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*. 1990, pp. 503–513. DOI: 10.1145/100216.100287. URL: https://doi.org/10.1145/100216.100287.

[Bel+13] Mihir Bellare et al. "Efficient Garbling from a Fixed-Key Blockcipher". In: *2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013*. 2013, pp. 478–492. DOI: 10.1109/SP.2013.39. URL: https://doi.org/10.1109/SP.2013.39.

参考文献 II

[Gue+15] Shay Gueron et al. "Fast Garbling of Circuits Under Standard Assumptions". In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*. Ed. by Indrajit Ray, Ninghui Li, and Christopher Kruegel. ACM, 2015, pp. 567–578. DOI: 10.1145/2810103.2813619. URL: https://doi.org/10.1145/2810103.2813619.

参考文献 III

[KMR14] Vladimir Kolesnikov, Payman Mohassel, and Mike Rosulek. “FleXOR: Flexible Garbling for XOR Gates That Beats Free-XOR”. In: *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*. Ed. by Juan A. Garay and Rosario Gennaro. Vol. 8617. Lecture Notes in Computer Science. Springer, 2014, pp. 440–457. DOI: 10.1007/978-3-662-44381-1_25. URL: https://doi.org/10.1007/978-3-662-44381-1_25.

参考文献 IV

[KS08] Vladimir Kolesnikov and Thomas Schneider. “Improved Garbled Circuit: Free XOR Gates and Applications”. In: *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*. 2008, pp. 486–498. DOI: 10.1007/978-3-540-70583-3_40. URL: https://doi.org/10.1007/978-3-540-70583-3_40.

参考文献 V

[NPS99] Moni Naor, Benny Pinkas, and Reuban Sumner. “Privacy preserving auctions and mechanism design”. In: *Proceedings of the First ACM Conference on Electronic Commerce (EC-99), Denver, CO, USA, November 3-5, 1999*. 1999, pp. 129–139. DOI: 10.1145/336992.337028. URL: https://doi.org/10.1145/336992.337028.

参考文献 VI

[Pin+09] Benny Pinkas et al. "Secure Two-Party Computation Is Practical". In: *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*. 2009, pp. 250–267. DOI: 10.1007/978-3-642-10366-7_15. URL: https://doi.org/10.1007/978-3-642-10366-7_15.

参考文献 VII

[RR21] Mike Rosulek and Lawrence Roy. “Three Halves Make a Whole? Beating the Half-Gates Lower Bound for Garbled Circuits”. In: *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*. 2021, pp. 94–124. DOI: 10.1007/978-3-030-84242-0_5. URL: https://doi.org/10.1007/978-3-030-84242-0_5.

[Yao86] Andrew Chi-Chih Yao. “How to Generate and Exchange Secrets (Extended Abstract)”. In: *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*. 1986, pp. 162–167. DOI: 10.1109/SFCS.1986.25. URL: https://doi.org/10.1109/SFCS.1986.25.

参考文献 VIII

[ZRE15] Samee Zahur, Mike Rosulek, and David Evans. "Two Halves Make a Whole - Reducing Data Transfer in Garbled Circuits Using Half Gates". In: *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*. 2015, pp. 220–250. DOI: 10.1007/978-3-662-46803-6_8. URL: https://doi.org/10.1007/978-3-662-46803-6_8.

Thanks!