基于混淆电路的协议

姚氏混淆电路与 BMR 协议

《安全多方计算——可证明安全视角》第八章

2026 年 1 月 30 日

目录

1 姚氏混淆电路

2 独立模型

3 混淆电路优化技术

4 BMR 多方混淆电路

计算一个与门……

表: 与门的真值表

x	y	$z = x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

安全地计算一个与门

表: 与门加密后的真值表

x 的对应密钥	y 的对应密钥	加密的 z(表 T)
k_X^0	k_Y^0	$\mathsf{Enc}_{k_X^0}(\mathsf{Enc}_{k_Y^0}(0))$
k_X^0	k_Y^1	$\mathsf{Enc}_{k_X^0}(\mathsf{Enc}_{k_Y^1}(0))$
k_X^1	k_Y^0	$\mathsf{Enc}_{k_X^1}(\mathsf{Enc}_{k_Y^0}(0))$
k_X^1	k_Y^1	$\mathsf{Enc}_{k_X^1}(\mathsf{Enc}_{k_Y^1}(1))$

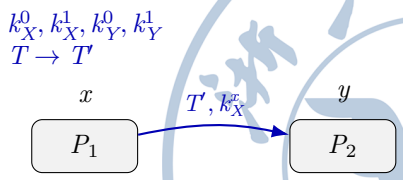
* 注: 实际传输时需将表 T 随机打乱为 T'.

安全地计算一个与门

表: 与门加密后的真值表

x 的对应密钥	y 的对应密钥	加密的 z(表 T)
k_X^0	k_Y^0	$\mathsf{Enc}_{k_X^0}(\mathsf{Enc}_{k_Y^0}(0))$
k_X^0	k_Y^1	$\mathsf{Enc}_{k_X^0}(\mathsf{Enc}_{k_Y^1}(0))$
k_X^1	k_Y^0	$\mathsf{Enc}_{k_X^1}(\mathsf{Enc}_{k_Y^0}(0))$
k_X^1	k_Y^1	$\mathsf{Enc}_{k_X^1}(\mathsf{Enc}_{k_Y^1}(1))$

* 注: 实际传输时需将表 T 随机打乱为 T'.

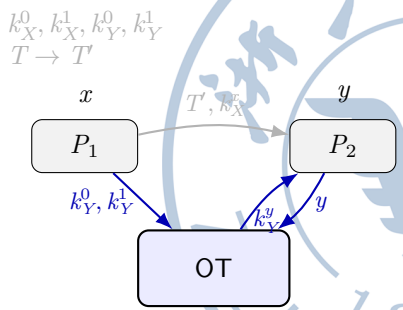


安全地计算一个与门

表: 与门加密后的真值表

x 的对应密钥	y 的对应密钥	加密的 z(表 T)
k_X^0	k_Y^0	$\mathsf{Enc}_{k_X^0}(\mathsf{Enc}_{k_Y^0}(0))$
k_X^0	k_Y^1	$\mathsf{Enc}_{k_X^0}(\mathsf{Enc}_{k_Y^1}(0))$
k_X^1	k_Y^0	$\mathsf{Enc}_{k_X^1}(\mathsf{Enc}_{k_Y^0}(0))$
k_X^1	k_Y^1	$\mathsf{Enc}_{k_X^1}(\mathsf{Enc}_{k_Y^1}(1))$

* 注: 实际传输时需将表 T 随机打乱为 T'.

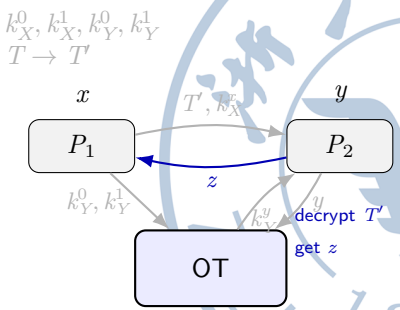


安全地计算一个与门

表: 与门加密后的真值表

x 的对应密钥	y 的对应密钥	加密的 z(表 T)
k_X^0	k_Y^0	$\mathsf{Enc}_{k_X^0}(\mathsf{Enc}_{k_Y^0}(0))$
k_X^0	k_Y^1	$\mathsf{Enc}_{k_X^0}(\mathsf{Enc}_{k_Y^1}(0))$
k_X^1	k_Y^0	$\mathsf{Enc}_{k_X^1}(\mathsf{Enc}_{k_Y^0}(0))$
k_X^1	k_Y^1	$\mathsf{Enc}_{k_X^1}(\mathsf{Enc}_{k_Y^1}(1))$

* 注: 实际传输时需将表 T 随机打乱为 T'.



将函数表示为查找表

表: 函数 $f(x,y)$ 加密后的真值表 (大小为 $|X|\cdot|Y|$)

x 的对应密钥	y 的对应密钥	加密的 z(表 T)
$k_X^{00\ldots00}$	$k_Y^{00\ldots00}$	$\mathsf{Enc}_{k_X^{00\ldots00}}(\mathsf{Enc}_{k_Y^{00\ldots00}}(f(00\ldots00,00\ldots00)))$
$k_X^{00\ldots00}$	$k_Y^{00\ldots01}$	$\mathsf{Enc}_{k_X^{00\ldots00}}(\mathsf{Enc}_{k_Y^{00\ldots01}}(f(00\ldots00,00\ldots01)))$
…	…	…
$k_X^{00\ldots00}$	$k_Y^{11\ldots11}$	$\mathsf{Enc}_{k_X^{00\ldots00}}(\mathsf{Enc}_{k_Y^{11\ldots11}}(f(00\ldots00,11\ldots11)))$
$k_X^{00\ldots01}$	$k_Y^{00\ldots00}$	$\mathsf{Enc}_{k_X^{00\ldots01}}(\mathsf{Enc}_{k_Y^{00\ldots00}}(f(00\ldots01,00\ldots00)))$
$k_X^{00\ldots01}$	$k_Y^{00\ldots01}$	$\mathsf{Enc}_{k_X^{00\ldots01}}(\mathsf{Enc}_{k_Y^{00\ldots01}}(f(00\ldots01,00\ldots01)))$
…	…	…
$k_X^{00\ldots01}$	$k_Y^{11\ldots11}$	$\mathsf{Enc}_{k_X^{00\ldots01}}(\mathsf{Enc}_{k_Y^{11\ldots11}}(f(00\ldots01,11\ldots11)))$
…	…	…
$k_X^{11\ldots11}$	$k_Y^{11\ldots11}$	$\mathsf{Enc}_{k_X^{11\ldots11}}(\mathsf{Enc}_{k_Y^{11\ldots11}}(f(11\ldots11,11\ldots11)))$

降低查找表的大小

- 为布尔电路的每一根导线选取两个密钥.
- 逐门计算，依次解密各门的密钥.

表: 与门的查找表

w_a 的对应密钥	w_b 的对应密钥	w_c 的对应密钥	查找表
k_a^0	k_b^0	k_c^0	$\mathsf{Enc}_{k_a^0}(\mathsf{Enc}_{k_b^0}(k_c^0))$
k_a^0	k_b^1	k_c^0	$\mathsf{Enc}_{k_a^0}(\mathsf{Enc}_{k_b^1}(k_c^0))$
k_a^1	k_b^0	k_c^0	$\mathsf{Enc}_{k_a^1}(\mathsf{Enc}_{k_b^0}(k_c^0))$
k_a^1	k_b^1	k_c^1	$\mathsf{Enc}_{k_a^1}(\mathsf{Enc}_{k_b^1}(k_c^1))$

哪个解密结果是正确的?

- 密钥后面添加 n 个 0 再进行加密.
- 解密结果的最后 n 位都是 0 时得到的就是正确的密钥.

表: 与门的查找表（添加 n 个 0）

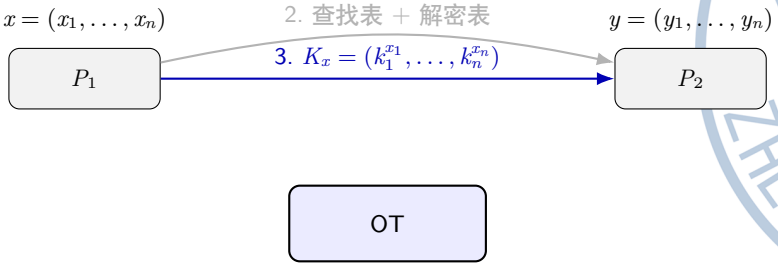
w_a 的对应密钥	w_b 的对应密钥	w_c 的对应密钥	查找表
k_a^0	k_b^0	k_c^0	$\mathsf{Enc}_{k_a^0}(\mathsf{Enc}_{k_b^0}(k_c^0 \| \| 0^n))$
k_a^0	k_b^1	k_c^0	$\mathsf{Enc}_{k_a^0}(\mathsf{Enc}_{k_b^1}(k_c^0 \| \| 0^n))$
k_a^1	k_b^0	k_c^0	$\mathsf{Enc}_{k_a^1}(\mathsf{Enc}_{k_b^0}(k_c^0 \| \| 0^n))$
k_a^1	k_b^1	k_c^1	$\mathsf{Enc}_{k_a^1}(\mathsf{Enc}_{k_b^1}(k_c^1 \| \| 0^n))$

姚氏混淆电路协议流程

1.
为每根导线 w_i 随机选取密钥 k_i^0, k_i^1
为每根导线构造并置换查找表
为每根输出导线构造解密表 $(0, k_i^0)$ 和 $(1, k_i^1)$

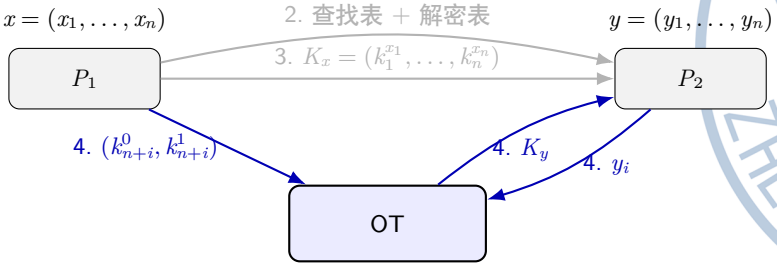
姚氏混淆电路协议流程

1.
为每根导线 w_i 随机选取密钥 k_i^0, k_i^1
为每根导线构造并置换查找表
为每根输出导线构造解密表 $(0, k_i^0)$ 和 $(1, k_i^1)$



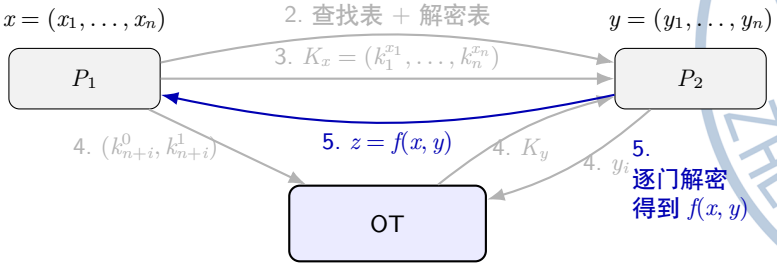
姚氏混淆电路协议流程

1.
为每根导线 w_i 随机选取密钥 k_i^0, k_i^1
为每根导线构造并置换查找表
为每根输出导线构造解密表 $(0, k_i^0)$ 和 $(1, k_i^1)$



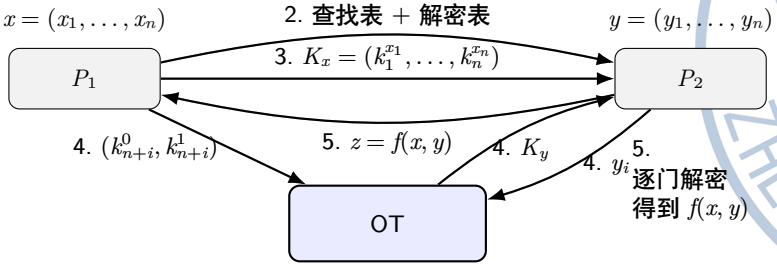
姚氏混淆电路协议流程

1.
为每根导线 w_i 随机选取密钥 k_i^0, k_i^1
为每根导线构造并置换查找表
为每根输出导线构造解密表 $(0, k_i^0)$ 和 $(1, k_i^1)$



姚氏混淆电路协议流程

1.
为每根导线 w_i 随机选取密钥 k_i^0, k_i^1
为每根导线构造并置换查找表
为每根输出导线构造解密表 $(0, k_i^0)$ 和 $(1, k_i^1)$



“特殊的”加密方案

“特殊的”加密方案

- Q: 如果（用错误的密钥）解密查找表其他几行会发生什么？

“特殊的”加密方案

- Q: 如果（用错误的密钥）解密查找表其他几行会发生什么？
- A: 我们希望**高效地发现**得到的是**错误的解密结果**.

“特殊的”加密方案

- Q: 如果（用错误的密钥）解密查找表其他几行会发生什么？
- A: 我们希望**高效地发现**得到的是**错误的解密结果**.

不可知、可验证密文空间

令 $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ 是一个对称加密方案.
用 $\mathsf{Range}_n(k) \stackrel{\mathsf{def}}{=} \{\mathsf{Enc}_k(x)\}_{x\in\{0,1\}^n}$ 表示密钥 k 的密文空间.

“特殊的”加密方案

- Q: 如果（用错误的密钥）解密查找表其他几行会发生什么？
- A: 我们希望**高效地发现**得到的是**错误的解密结果**.

不可知、可验证密文空间

令 $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ 是一个对称加密方案.

用 $\mathsf{Range}_n(k) \stackrel{\text{def}}{=} \{\mathsf{Enc}_k(x)\}_{x\in\{0,1\}^n}$ 表示密钥 k 的密文空间.

1. 如果对任意 PPT 的算法 $\mathcal{A}$，任意多项式 p，以及足够大的 n 都有

$$\Pr[k \leftarrow \mathsf{Gen}(1^n), \mathcal{A}(1^n) \in \mathsf{Range}_n(k)] < \frac{1}{p(n)}$$

那么，我们称 $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ 有不可知的密文空间 (elusive range).

“特殊的”加密方案

- Q: 如果（用错误的密钥）解密查找表其他几行会发生什么？
- A: 我们希望**高效地发现**得到的是**错误的解密结果**.

不可知、可验证密文空间

令 $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ 是一个对称加密方案.
用 $\mathsf{Range}_n(k) \stackrel{\mathsf{def}}{=} \{\mathsf{Enc}_k(x)\}_{x\in\{0,1\}^n}$ 表示密钥 k 的密文空间.

1. 如果对任意 PPT 的算法 $\mathcal{A}$，任意多项式 p，以及足够大的 n 都有

$$\Pr[k \leftarrow \mathsf{Gen}(1^n), \mathcal{A}(1^n) \in \mathsf{Range}_n(k)] < \frac{1}{p(n)}$$

   那么，我们称 $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ 有不可知的密文空间 (elusive range).

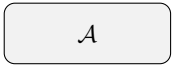
2. 如果存在 PPT 的算法 M 满足 $M(1^n, k, c) = 1$ 当且仅当 $c \in \mathsf{Range}_n(k)$，那么，我们称 $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ 有可验证的密文空间 (efficiently verifiable range).

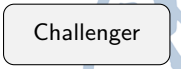
双重加密安全

- Q: 如果（用错误的密钥）解密查找表其他几行会发生什么？
- A: 我们希望得到的错误的解密结果**不会泄露明文信息**.

双重加密安全

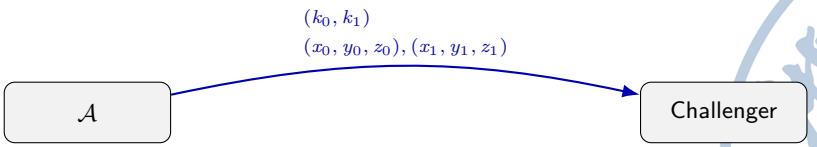
- Q: 如果（用错误的密钥）解密查找表其他几行会发生什么？
- A: 我们希望得到的错误的解密结果**不会泄露明文信息**.





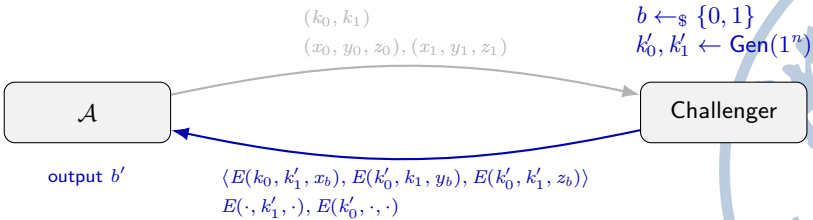
双重加密安全

- Q: 如果（用错误的密钥）解密查找表其他几行会发生什么？
- A: 我们希望得到的错误的解密结果**不会泄露明文信息**.



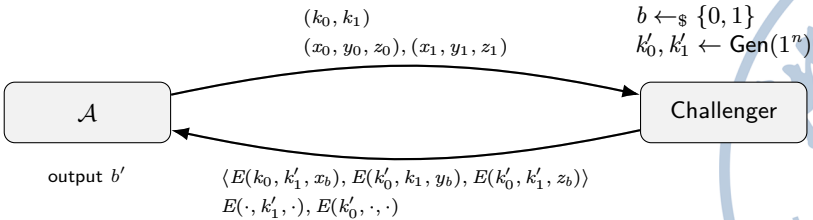
双重加密安全

- Q: 如果（用错误的密钥）解密查找表其他几行会发生什么？
- A: 我们希望得到的错误的解密结果**不会泄露明文信息**.



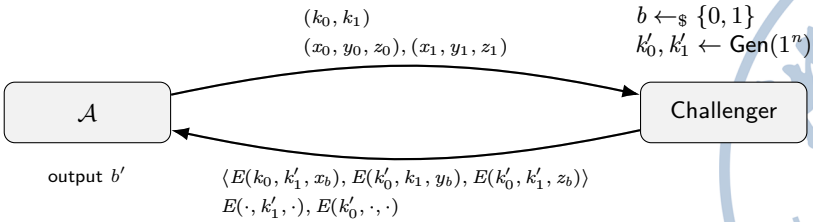
双重加密安全

- Q: 如果（用错误的密钥）解密查找表其他几行会发生什么？
- A: 我们希望得到的错误的解密结果**不会泄露明文信息**.



双重加密安全

- Q: 如果（用错误的密钥）解密查找表其他几行会发生什么？
- A: 我们希望得到的错误的解密结果**不会泄露明文信息**.



选择明文攻击下的双重加密安全 (Secure under Chosen Double Encryption)

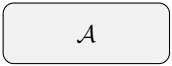
如果对任意概率多项式时间的敌手 $\mathcal{A}$，都存在一个可忽略函数 negl，使得

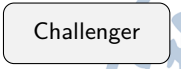
$$\Pr[\mathsf{Game}_{\mathcal{A}}^{\mathsf{double}}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n)$$

那么，我们说对称加密方案 $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ 在选择明文攻击下具有双重加密安全.

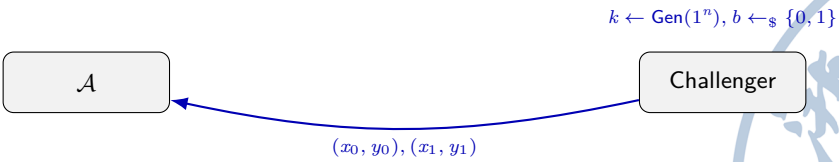
选择明文攻击安全

选择明文攻击安全

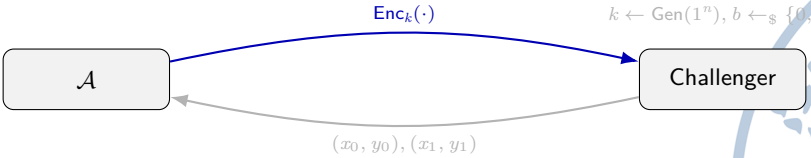




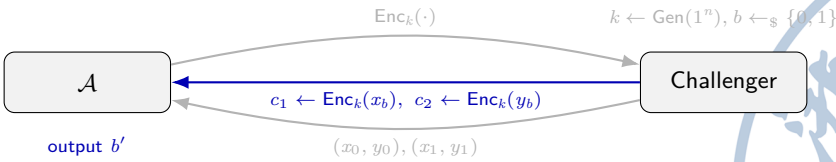
选择明文攻击安全


$k \leftarrow \mathsf{Gen}(1^n), b \leftarrow_{\$} \{0,1\}$
$\mathcal{A}$
Challenger
$(x_0, y_0), (x_1, y_1)$

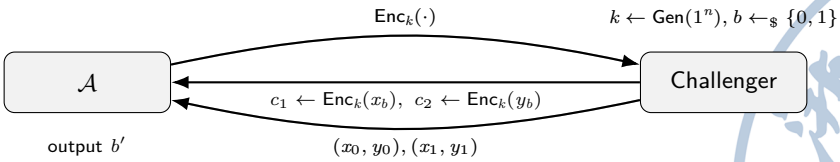
选择明文攻击安全


$\mathsf{Enc}_k(\cdot)$
$k \leftarrow \mathsf{Gen}(1^n), b \leftarrow_{\$} \{0,1\}$
$\mathcal{A}$
Challenger
$(x_0, y_0), (x_1, y_1)$

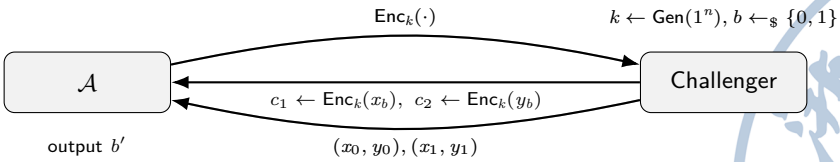
选择明文攻击安全


$\mathsf{Enc}_k(\cdot)$
$k \leftarrow \mathsf{Gen}(1^n), b \leftarrow_{\$} \{0,1\}$
$\mathcal{A}$
Challenger
$c_1 \leftarrow \mathsf{Enc}_k(x_b),\ c_2 \leftarrow \mathsf{Enc}_k(y_b)$
output b'
$(x_0, y_0), (x_1, y_1)$

选择明文攻击安全


$\mathsf{Enc}_k(\cdot)$
$k \leftarrow \mathsf{Gen}(1^n), b \leftarrow_{\$} \{0,1\}$
$\mathcal{A}$
Challenger
$c_1 \leftarrow \mathsf{Enc}_k(x_b),\ c_2 \leftarrow \mathsf{Enc}_k(y_b)$
output b'
$(x_0, y_0), (x_1, y_1)$

选择明文攻击安全



选择明文攻击下的不可区分性 (IND-CPA Secure)

如果对任意概率多项式时间的敌手 $\mathcal{A}$，都存在一个可忽略函数 negl，使得

$$\Pr[\mathsf{Game}_{\mathcal{A}}^{\mathsf{CPA}}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n)$$

那么，我们说对称加密方案 $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ 在选择明文攻击下具有不可区分性.

不可区分性 ⇒ 双重加密安全

- 目标：证明当 $b \in \{0,1\}$ 时，$\mathsf{Game}_{\mathcal{A}}^{\mathsf{double}}(n)$ 视图不可区分.
- 构造中间游戏 $\mathsf{Game}_{\mathcal{A}}^{\mathsf{mod}}(n)$ 并比较相邻游戏.

不可区分性 ⇒ 双重加密安全

- 目标：证明当 $b \in \{0,1\}$ 时，$\mathsf{Game}_{\mathcal{A}}^{\mathsf{double}}(n)$ 视图不可区分.
- 构造中间游戏 $\mathsf{Game}_{\mathcal{A}}^{\mathsf{mod}}(n)$ 并比较相邻游戏.

步骤 1：double ≈ mod

在 mod 中总是加密 y_0：
$\langle E(k_0, k_1', x_b), E(k_0', k_1, y_0), E(k_0', k_1', z_b)\rangle$.

- 当 $b = 0$ 两游戏相同；
- 当 $b = 1$ 区分它们等价于区分 $E(k_0', k_1, y_0)$ 与 $E(k_0', k_1, y_1)$，

不可区分性 ⇒ 双重加密安全

- 目标：证明当 $b \in \{0,1\}$ 时，$\mathsf{Game}_{\mathcal{A}}^{\mathsf{double}}(n)$ 视图不可区分.
- 构造中间游戏 $\mathsf{Game}_{\mathcal{A}}^{\mathsf{mod}}(n)$ 并比较相邻游戏.

步骤 1：double ≈ mod

在 mod 中总是加密 y_0：
$\langle E(k_0, k_1', x_b), E(k_0', k_1, y_0), E(k_0', k_1', z_b)\rangle$.

- 当 $b = 0$ 两游戏相同；
- 当 $b = 1$ 区分它们等价于区分 $E(k_0', k_1, y_0)$ 与 $E(k_0', k_1, y_1)$，

步骤 2：$\Pr[\mathsf{mod} = 1] \le \frac{1}{2} + \mathsf{negl}(n)$

假设 $\mathcal{A}$ 已知 k_0'. 收到的 $E(k_0', k_1, y_0)$ 与 b 无关，可移除.
若 $\mathcal{A}$ 在 mod 中获胜，则可构造在 $\mathsf{Game}_{\mathcal{A}}^{\mathsf{CPA}}(n)$ 中获胜的算法：
对 $\langle E(k_0, k_1', x_b), E(k_0', k_1', z_b)\rangle$ 去一层解密得到 CPA 视图.

不可区分性 ⇒ 双重加密安全

- 目标：证明当 $b \in \{0,1\}$ 时，$\mathsf{Game}_{\mathcal{A}}^{\mathsf{double}}(n)$ 视图不可区分.
- 构造中间游戏 $\mathsf{Game}_{\mathcal{A}}^{\mathsf{mod}}(n)$ 并比较相邻游戏.

步骤 1：double ≈ mod

在 mod 中总是加密 y_0：
$\langle E(k_0, k_1', x_b), E(k_0', k_1, y_0), E(k_0', k_1', z_b)\rangle$.

- 当 $b = 0$ 两游戏相同；
- 当 $b = 1$ 区分它们等价于区分 $E(k_0', k_1, y_0)$ 与 $E(k_0', k_1, y_1)$，

步骤 2：$\Pr[\mathsf{mod} = 1] \le \frac{1}{2} + \mathsf{negl}(n)$

假设 $\mathcal{A}$ 已知 k_0'. 收到的 $E(k_0', k_1, y_0)$ 与 b 无关，可移除.
若 $\mathcal{A}$ 在 mod 中获胜，则可构造在 $\mathsf{Game}_{\mathcal{A}}^{\mathsf{CPA}}(n)$ 中获胜的算法：
对 $\langle E(k_0, k_1', x_b), E(k_0', k_1', z_b)\rangle$ 去一层解密得到 CPA 视图.

由混合游戏与 IND-CPA 安全性可知 $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ 具有双重加密安全. □

混淆电路的正确性

混淆电路的正确性

设 $\mathcal{C}$ 为布尔电路，输入为 $x = x_1 \dots x_n$ 与 $y = y_1 \dots y_n$. 若构造混淆电路 $G(\mathcal{C})$ 的加密方案满足**不可知的密文空间与可验证的密文空间**，则给定 $G(\mathcal{C})$ 及输入导线标签 $k_{\mathsf{in}_1}^{x_1}, \dots, k_{\mathsf{in}_n}^{x_n}, k_{\mathsf{in}_{n+1}}^{y_1}, \dots, k_{\mathsf{in}_{2n}}^{y_n}$，可计算出 $\mathcal{C}(x, y)$，失败概率为可忽略函数.

混淆电路的正确性

混淆电路的正确性

设 $\mathcal{C}$ 为布尔电路，输入为 $x = x_1 \dots x_n$ 与 $y = y_1 \dots y_n$. 若构造混淆电路 $G(\mathcal{C})$ 的加密方案满足**不可知的密文空间与可验证的密文空间**，则给定 $G(\mathcal{C})$ 及输入导线标签 $k_{\mathsf{in}_1}^{x_1}, \dots, k_{\mathsf{in}_n}^{x_n}, k_{\mathsf{in}_{n+1}}^{y_1}, \dots, k_{\mathsf{in}_{2n}}^{y_n}$，可计算出 $\mathcal{C}(x, y)$，失败概率为可忽略函数.

证明思路：先证明每个门的混淆表仅有唯一一项可被正确解密；再按电路拓扑顺序归纳得到所有导线的正确标签，最后解码输出.

混淆电路的正确性证明

正确性证明思路：

混淆电路的正确性证明

正确性证明思路：

1. 每个门的混淆表仅有唯一一项可被正确解密.

混淆电路的正确性证明

正确性证明思路：

1. 每个门的混淆表仅有唯一一项可被正确解密.
2. 整个电路出错的概率不超过每个门出错的概率之和.

混淆电路的正确性证明

正确性证明思路：

1. 每个门的混淆表仅有唯一一项可被正确解密.
2. 整个电路出错的概率不超过每个门出错的概率之和.

设门 g 输入导线为 w_1, w_2，输出导线为 w_3，输入比特为 (α, β). 混淆表包含

$$c_{\alpha,\beta} = \mathsf{Enc}_{k_1^\alpha}\big(\mathsf{Enc}_{k_2^\beta}(k_3^{g(\alpha,\beta)})\big),$$

并随机打乱为 c_0, c_1, c_2, c_3.

混淆电路的正确性证明

正确性证明思路：

1. 每个门的混淆表仅有唯一一项可被正确解密.
2. 整个电路出错的概率不超过每个门出错的概率之和.

设门 g 输入导线为 w_1, w_2，输出导线为 w_3，输入比特为 (α, β). 混淆表包含

$$c_{\alpha,\beta} = \mathsf{Enc}_{k_1^\alpha}\big(\mathsf{Enc}_{k_2^\beta}(k_3^{g(\alpha,\beta)})\big),$$

并随机打乱为 c_0, c_1, c_2, c_3.

目标：给定 k_1^α, k_2^β，唯一存在 c_i 使得

$$c_i \in \mathsf{Range}_n(k_1^\alpha) \ \wedge \ \mathsf{Dec}_{k_1^\alpha}(c_i) \in \mathsf{Range}_n(k_2^\beta),$$

且解密得到 $k_3^{g(\alpha,\beta)}$.

混淆电路的正确性证明

正确性证明思路：

1. 每个门的混淆表仅有唯一一项可被正确解密.
2. 整个电路出错的概率不超过每个门出错的概率之和.

设门 g 输入导线为 w_1, w_2，输出导线为 w_3，输入比特为 (α, β). 混淆表包含

$$c_{\alpha,\beta} = \mathsf{Enc}_{k_1^\alpha}\big(\mathsf{Enc}_{k_2^\beta}(k_3^{g(\alpha,\beta)})\big),$$

并随机打乱为 c_0, c_1, c_2, c_3.

目标：给定 k_1^α, k_2^β，唯一存在 c_i 使得

$$c_i \in \mathsf{Range}_n(k_1^\alpha) \ \wedge \ \mathsf{Dec}_{k_1^\alpha}(c_i) \in \mathsf{Range}_n(k_2^\beta),$$

且解密得到 $k_3^{g(\alpha,\beta)}$.

- **可验证密文空间**保证：若解密成功（非 $\perp$），则输入密文属于对应密文空间.

混淆电路的正确性证明

正确性证明思路：

1. 每个门的混淆表仅有唯一一项可被正确解密.
2. 整个电路出错的概率不超过每个门出错的概率之和.

设门 g 输入导线为 w_1, w_2，输出导线为 w_3，输入比特为 (α, β). 混淆表包含

$$c_{\alpha,\beta} = \mathsf{Enc}_{k_1^\alpha}\big(\mathsf{Enc}_{k_2^\beta}(k_3^{g(\alpha,\beta)})\big),$$

并随机打乱为 c_0, c_1, c_2, c_3.

目标：给定 k_1^α, k_2^β，唯一存在 c_i 使得

$$c_i \in \mathsf{Range}_n(k_1^\alpha) \ \wedge \ \mathsf{Dec}_{k_1^\alpha}(c_i) \in \mathsf{Range}_n(k_2^\beta),$$

且解密得到 $k_3^{g(\alpha,\beta)}$.

- **可验证密文空间**保证：若解密成功（非 $\perp$），则输入密文属于对应密文空间.
- **不可知密文空间**保证：若解密失败（$\perp$），则输入密文属于对应密文空间的概率可以忽略.

混淆电路的正确性证明

正确性证明思路：

1. 每个门的混淆表仅有唯一一项可被正确解密.
2. 整个电路出错的概率不超过每个门出错的概率之和.

设门 g 输入导线为 w_1, w_2，输出导线为 w_3，输入比特为 (α, β). 混淆表包含

$$c_{\alpha,\beta} = \mathsf{Enc}_{k_1^\alpha}\big(\mathsf{Enc}_{k_2^\beta}(k_3^{g(\alpha,\beta)})\big),$$

并随机打乱为 c_0, c_1, c_2, c_3.

目标：给定 k_1^α, k_2^β，唯一存在 c_i 使得

$$c_i \in \mathsf{Range}_n(k_1^\alpha) \ \wedge \ \mathsf{Dec}_{k_1^\alpha}(c_i) \in \mathsf{Range}_n(k_2^\beta),$$

且解密得到 $k_3^{g(\alpha,\beta)}$.

- **可验证密文空间**保证：若解密成功（非 $\perp$），则输入密文属于对应密文空间.
- **不可知密文空间**保证：若解密失败（$\perp$），则输入密文属于对应密文空间的概率可以忽略.

在证明**安全性**之前，我们先来了解一下……

独立模型（半诚实敌手）

独立模型下对于半诚实敌手的安全性

令 $f=(f_1,f_2)$ 是一个功能．如果存在 PPT 的算法 $\mathcal{S}_1$ 和 $\mathcal{S}_2$ 使得

$$\{(\mathcal{S}_1(x,f_1(x,y)),f(x,y))\}_{x,y\in\{0,1\}^*} \overset{\text{comp}}{\approx} \{(\mathsf{view}_1^{\Pi}(x,y),\mathsf{output}^{\Pi}(x,y))\}_{x,y\in\{0,1\}^*}$$

$$\{(\mathcal{S}_2(y,f_2(x,y)),f(x,y))\}_{x,y\in\{0,1\}^*} \overset{\text{comp}}{\approx} \{(\mathsf{view}_2^{\Pi}(x,y),\mathsf{output}^{\Pi}(x,y))\}_{x,y\in\{0,1\}^*}$$

其中 $|x|=|y|$，那么，我们说协议 Π 对于静态半诚实敌手安全实现了功能 f.

独立模型（半诚实敌手）

独立模型下对于半诚实敌手的安全性

令 $f=(f_1,f_2)$ 是一个功能．如果存在 PPT 的算法 $\mathcal{S}_1$ 和 $\mathcal{S}_2$ 使得

$$\{(\mathcal{S}_1(x,f_1(x,y)),f(x,y))\}_{x,y\in\{0,1\}^*} \overset{\text{comp}}{\approx} \{(\text{view}_1^{\Pi}(x,y),\text{output}^{\Pi}(x,y))\}_{x,y\in\{0,1\}^*}$$

$$\{(\mathcal{S}_2(y,f_2(x,y)),f(x,y))\}_{x,y\in\{0,1\}^*} \overset{\text{comp}}{\approx} \{(\text{view}_2^{\Pi}(x,y),\text{output}^{\Pi}(x,y))\}_{x,y\in\{0,1\}^*}$$

其中 $|x|=|y|$，那么，我们说协议 Π 对于静态半诚实敌手安全实现了功能 f.

注：对于确定性功能，我们无需考虑联合分布，只需要 $\mathcal{S}_i$ 的输出与 $\text{view}_i^{\Pi}(x,y)$ 不可区分.

并发组合的（不）安全性

独立模型下安全的协议，与其他协议**并发**运行时可能不安全!

并发组合的（不）安全性

独立模型下安全的协议，与其他协议**并发**运行时可能不安全!

例 (密钥交换)

设 Π 为独立模型下安全的密钥交换协议，双方生成共享密钥 k.
构造 Π'：在 Π 基础上增加指令“若收到消息等于 k 则回复 `yes`，否则 `no`”.

并发组合的（不）安全性

独立模型下安全的协议，与其他协议**并发**运行时可能不安全!

例 (密钥交换)

设 Π 为独立模型下安全的密钥交换协议，双方生成共享密钥 k.
构造 Π'：在 Π 基础上增加指令“若收到消息等于 k 则回复 `yes`，否则 `no`”.

并发组合攻击（一次一密 + Π'）

加密消息 $m \in \{\text{“buy”}, \text{“sell”}\}$ 采用一次一密 $c = k \oplus m$. 敌手看到 c 后发送 $c' = c \oplus \text{“buy”}$ 给 Π':

- 若 $m = \text{“buy”}$，则 $c' = k$，Π' 回复 `yes`.
- 若 $m = \text{“sell”}$，则 $c' \neq k$，Π' 回复 `no`.

因此敌手可判别明文，破坏隐私.

并发组合的（不）安全性

独立模型下安全的协议，与其他协议**并发**运行时可能不安全!

例 (密钥交换)

设 Π 为独立模型下安全的密钥交换协议，双方生成共享密钥 k.
构造 Π'：在 Π 基础上增加指令“若收到消息等于 k 则回复 `yes`，否则 `no`”.

并发组合攻击（一次一密 + Π'）

加密消息 $m \in \{\text{“buy”}, \text{“sell”}\}$ 采用一次一密 $c = k \oplus m$. 敌手看到 c 后发送 $c' = c \oplus \text{“buy”}$ 给 Π':

- 若 $m = \text{“buy”}$，则 $c' = k$，Π' 回复 `yes`.
- 若 $m = \text{“sell”}$，则 $c' \neq k$，Π' 回复 `no`.

因此敌手可判别明文，破坏隐私.

独立模型下证明安全的协议在顺序组合时仍然是安全的.

独立模型（恶意敌手）

独立模型下对于恶意敌手的安全性

令 $f\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ 是一个功能，其中 $f = (f_1, f_2)$. 令 Π 是一个计算 f 的两方协议. 如果对于每一对真实世界中可允许的非统一的概率多项式时间的机器 $\overline{A} = (A_1, A_2)$，都存在一对理想世界中可允许的非统一的概率期望多项式时间的机器 $\overline{B} = (B_1, B_2)$，使得

$$\{\mathrm{IDEAL}_{f,\overline{B}}(x,y)\}_{x,y \text{ s.t. } |x|=|y|} \overset{\text{comp}}{\approx} \{\mathrm{REAL}_{\Pi,\overline{A}}(x,y)\}_{x,y \text{ s.t. } |x|=|y|}$$

那么，我们说协议 Π 对于静态恶意敌手安全实现了 f.

注：非统一 (non-uniform) 指的是一个算法无需支持任意输入，而是可以对每个输入定义单独的算法.
注：可允许 (admissible) 指的是至少有一个参与方是诚实的.

独立模型（恶意敌手）：理想世界的运行

独立模型（恶意敌手）：理想世界的运行

- **输入**：每个参与方获得一个输入，记为 w（对 P_1 而言 $w = x$，对 P_2 而言 $w = y$）.

独立模型（恶意敌手）：理想世界的运行

- **输入**：每个参与方获得一个输入，记为 w（对 P_1 而言 $w = x$，对 P_2 而言 $w = y$）.
- **向可信方 (trusted party) 发送输入**：诚实的参与方总是将 w 发给可信方．恶意的参与方可以根据 w 选择结束协议或发送 $w' \in \{0,1\}^{|w|}$ 给可信方.

独立模型（恶意敌手）：理想世界的运行

- **输入**：每个参与方获得一个输入，记为 w（对 P_1 而言 $w = x$，对 P_2 而言 $w = y$）.
- **向可信方 (trusted party) 发送输入**：诚实的参与方总是将 w 发给可信方．恶意的参与方可以根据 w 选择结束协议或发送 $w' \in \{0,1\}^{|w|}$ 给可信方.
- **可信方回复第一个参与方**．当可信方收到输入对 (x, y) 后，他首先向第一个参与方发送 $f_1(x, y)$. 如果可信方只收到一个合法的输入，他向两个参与方都发送 $\perp$.

独立模型（恶意敌手）：理想世界的运行

- **输入**：每个参与方获得一个输入，记为 w（对 P_1 而言 $w = x$，对 P_2 而言 $w = y$）.
- **向可信方 (trusted party) 发送输入**：诚实的参与方总是将 w 发给可信方．恶意的参与方可以根据 w 选择结束协议或发送 $w' \in \{0,1\}^{|w|}$ 给可信方.
- **可信方回复第一个参与方**．当可信方收到输入对 (x, y) 后，他首先向第一个参与方发送 $f_1(x, y)$. 如果可信方只收到一个合法的输入，他向两个参与方都发送 $\perp$.
- **可信方回复第二个参与方**．如果第一个参与方是恶意的，他可以基于自己的输入和可信方的回复，决定是否让可信方终止．如果第一个参与方选择终止，可信方向第二个参与方发送 $\perp$ 并终止；否则，可信方向第二个参与方发送 $f_2(x, y)$.

独立模型（恶意敌手）：理想世界的运行

- **输入**：每个参与方获得一个输入，记为 w（对 P_1 而言 $w = x$，对 P_2 而言 $w = y$）.
- **向可信方 (trusted party) 发送输入**：诚实的参与方总是将 w 发给可信方．恶意的参与方可以根据 w 选择结束协议或发送 $w' \in \{0,1\}^{|w|}$ 给可信方.
- **可信方回复第一个参与方**．当可信方收到输入对 (x, y) 后，他首先向第一个参与方发送 $f_1(x, y)$. 如果可信方只收到一个合法的输入，他向两个参与方都发送 $\perp$.
- **可信方回复第二个参与方**．如果第一个参与方是恶意的，他可以基于自己的输入和可信方的回复，决定是否让可信方终止．如果第一个参与方选择终止，可信方向第二个参与方发送 $\perp$ 并终止；否则，可信方向第二个参与方发送 $f_2(x, y)$.
- **输出**：诚实的参与方总是将他从可信方收到的消息作为输出．恶意的参与方可以输出一个关于他的输入和来自可信方消息的（概率多项式时间可计算的）函数.

独立模型与通用可组合模型的关系

通用可组合 (UC) 模型

- 存在环境 $\mathcal{Z}$ 与协议**实时交互**.
- $\mathcal{Z}$ 可控制消息顺序，代表任意外部环境.
- 模拟器不可倒带 (rewind).
- 支持**任意并发组合**的安全性保证.

独立模型与通用可组合模型的关系

通用可组合 (UC) 模型

- 存在环境 $\mathcal{Z}$ 与协议**实时交互**.
- $\mathcal{Z}$ 可控制消息顺序，代表任意外部环境.
- 模拟器不可倒带 (rewind).
- 支持**任意并发组合**的安全性保证.

独立 (Stand-alone) 模型

- 无环境 $\mathcal{Z}$ 的实时交互.
- 输入在协议开始时确定，更“静态”.
- 恶意情形下可使用倒带进行模拟/提取.
- 仅保证**单次执行/顺序组合**安全.

独立模型与通用可组合模型的关系

通用可组合 (UC) 模型

- 存在环境 $\mathcal{Z}$ 与协议**实时交互**.
- $\mathcal{Z}$ 可控制消息顺序，代表任意外部环境.
- 模拟器不可倒带 (rewind).
- 支持**任意并发组合**的安全性保证.

独立 (Stand-alone) 模型

- 无环境 $\mathcal{Z}$ 的实时交互.
- 输入在协议开始时确定，更“静态”.
- 恶意情形下可使用倒带进行模拟/提取.
- 仅保证**单次执行/顺序组合**安全.

- 协议在**非并发环境（只允许在协议开始前或结束后通信）**下 UC-安全 $\Longleftrightarrow$ 在独立模型下安全.
- 但独立模型**不保证**并发组合安全（见上一页反例）.

姚氏混淆电路协议的安全性（情况 1）

姚氏混淆电路协议的安全性

令 f 是单一输出确定性功能．假设使用的加密方案满足选择明文攻击安全 (IND-CPA secure)，且有不可知的密文空间、可验证的密文空间．那么，的姚氏混淆电路协议 Π 在 $\mathcal{F}_{\text{OT}}$-混合模型下对于静态半诚实敌手安全实现了 f.

姚氏混淆电路协议的安全性（情况 1）

姚氏混淆电路协议的安全性

令 f 是单一输出确定性功能．假设使用的加密方案满足选择明文攻击安全 (IND-CPA secure)，且有不可知的密文空间、可验证的密文空间．那么，的姚氏混淆电路协议 Π 在 $\mathcal{F}_{\text{OT}}$-混合模型下对于静态半诚实敌手安全实现了 f.

情况 1：P_1 **被攻陷**.

$\mathcal{S}_1$ 的构造非常简单：$\mathcal{S}_1$ 的输入是 $(x, f(x,y))$，它随机选择 $r_{\mathcal{C}}$（P_1 以此为随机数生成混淆电路 $G(\mathcal{C})$），然后输出

$$(x, r_{\mathcal{C}}, f(x,y))$$

显然，$\mathcal{S}_1$ 选择的随机数与 P_1 诚实运行选择的随机数有相同的分布．因此，

$$\{\mathcal{S}_1(x, f(x,y))\}_{x,y\in\{0,1\}^n} \overset{\text{comp}}{\approx} \{\text{view}_1^{\Pi}(x,y)\}_{x,y\in\{0,1\}^n}$$

姚氏混淆电路协议的安全性（情况 2）

情况 2：P_2 **被攻陷**. 构造模拟器 $\mathcal{S}_2$，输入 $(y, f(x,y))$，输出 view_2^{Π}.

姚氏混淆电路协议的安全性（情况 2）

情况 2：P_2 **被攻陷**. 构造模拟器 $\mathcal{S}_2$，输入 $(y, f(x,y))$，输出 view_2^{Π}.

- P_2 在步骤 1 先收到混淆电路，$\mathcal{S}_2$ 需模拟一个**假的**混淆电路 $\tilde{G}(C)$.

姚氏混淆电路协议的安全性（情况 2）

情况 2：P_2 **被攻陷**. 构造模拟器 $\mathcal{S}_2$，输入 $(y, f(x,y))$，输出 view_2^{Π}.

- P_2 在步骤 1 先收到混淆电路，$\mathcal{S}_2$ 需模拟一个**假的**混淆电路 $\tilde{G}(\mathcal{C})$.
- $\mathcal{S}_2$ 不知道 x，因此要求：**无论输入密钥是哪一组，电路输出都等于** $f(x,y)$.

姚氏混淆电路协议的安全性（$\mathcal{S}_2$ 的构造）

1. 构造 $\tilde{G}(\mathcal{C})$：对每条导线 w_i 取两把随机密钥 k_i, k_i'. 对于门 $g:(w_i,w_j)\to w_\ell$，混淆表的四项都加密同一密钥 k_ℓ：

$$\mathsf{Enc}_{k_i}(\mathsf{Enc}_{k_j}(k_\ell)),\ \mathsf{Enc}_{k_i}(\mathsf{Enc}_{k_j'}(k_\ell)),\ \mathsf{Enc}_{k_i'}(\mathsf{Enc}_{k_j}(k_\ell)),\ \mathsf{Enc}_{k_i'}(\mathsf{Enc}_{k_j'}(k_\ell)).$$

随机排列得到该门的混淆表.

姚氏混淆电路协议的安全性（$\mathcal{S}_2$ 的构造）

1. 构造 $\tilde{G}(\mathcal{C})$：对每条导线 w_i 取两把随机密钥 k_i, k_i'. 对于门 $g:(w_i,w_j)\to w_\ell$，混淆表的四项都加密同一密钥 k_ℓ：

$$\mathsf{Enc}_{k_i}(\mathsf{Enc}_{k_j}(k_\ell)),\ \mathsf{Enc}_{k_i}(\mathsf{Enc}_{k_j'}(k_\ell)),\ \mathsf{Enc}_{k_i'}(\mathsf{Enc}_{k_j}(k_\ell)),\ \mathsf{Enc}_{k_i'}(\mathsf{Enc}_{k_j'}(k_\ell)).$$

随机排列得到该门的混淆表.

2. 输出解密表：令输出 $f(x,y)=z_1\dots z_n$，输出导线为 $w_{m-n+1},\dots,w_m$. 若 $z_i=0$，解密表为 $[(0,k_{m-n+i}),(1,k'_{m-n+i})]$；若 $z_i=1$，解密表为 $[(0,k'_{m-n+i}),(1,k_{m-n+i})]$.

姚氏混淆电路协议的安全性（$\mathcal{S}_2$ 的构造）

1. 构造 $\tilde{G}(\mathcal{C})$：对每条导线 w_i 取两把随机密钥 k_i, k_i'. 对于门 $g:(w_i,w_j)\to w_\ell$，混淆表的四项都加密同一密钥 k_ℓ：

$$\mathsf{Enc}_{k_i}(\mathsf{Enc}_{k_j}(k_\ell)),\ \mathsf{Enc}_{k_i}(\mathsf{Enc}_{k_j'}(k_\ell)),\ \mathsf{Enc}_{k_i'}(\mathsf{Enc}_{k_j}(k_\ell)),\ \mathsf{Enc}_{k_i'}(\mathsf{Enc}_{k_j'}(k_\ell)).$$

随机排列得到该门的混淆表.

2. 输出解密表：令输出 $f(x,y)=z_1\dots z_n$，输出导线为 $w_{m-n+1},\dots,w_m$. 若 $z_i=0$，解密表为 $[(0,k_{m-n+i}),(1,k'_{m-n+i})]$；若 $z_i=1$，解密表为 $[(0,k'_{m-n+i}),(1,k_{m-n+i})]$.

3. 模拟其余消息：

- P_2 从 P_1 收到输入密钥：设为 $k_1,\dots,k_n$.
- 在 $\mathcal{F}_{\mathsf{OT}}$-混合模型下，$\mathcal{S}_2$ 令 P_2 获得 $k_{n+1},\dots,k_{2n}$.

综上，$\mathcal{S}_2$ 输出

$$(y,\tilde{G}(\mathcal{C}),k_1,\dots,k_n,k_{n+1},\dots,k_{2n}).$$

姚氏混淆电路协议的安全性（情况 2）

目标:

$$\{\mathcal{S}_2(y, f(x,y))\}_{x,y} \overset{\text{comp}}{\approx} \{\mathsf{view}_2^{\Pi}(x,y)\}_{x,y}.$$

姚氏混淆电路协议的安全性（情况 2）

目标：

$$\{\mathcal{S}_2(y, f(x,y))\}_{x,y} \overset{\text{comp}}{\approx} \{\text{view}_2^{\Pi}(x,y)\}_{x,y}.$$

混合论证. 将电路门按拓扑顺序记为 $g_1, \ldots, g_{|\mathcal{C}|}$. 定义混合实验 $H_i(x,y)$：

- 先用输入 (x,y) 计算电路，标记**活跃密钥**与**不活跃密钥**.
- 对前 i 个门 $g_1, \ldots, g_i$，将其混淆表替换为只加密输出导线**活跃密钥**的版本；
- 其余门保持真实混淆表.

则 $H_0(x,y) = \text{view}_2^{\Pi}(x,y)$，而 $H_{|\mathcal{C}|}(x,y)$ 与 $\mathcal{S}_2(y, f(x,y))$ 分布相同.

姚氏混淆电路协议的安全性（情况 2）

关键一步：对任意 i，证明 $H_{i-1}(x,y) \overset{\text{comp}}{\approx} H_i(x,y)$. 若存在区分器 D 使两者可区分，则可构造敌手 $\mathcal{A}_E$ 攻破**双重加密安全性**：

- H_{i-1} 与 H_i 的唯一区别在于门 g_i 的混淆表中 **某一项的明文**由 $k_c^{g(\alpha,\beta)}$ 替换为同一密钥.
- 区分这两种门表等价于区分双重加密下的两条明文，进而违反双重加密安全性（由 IND-CPA 推得）.

由混合论证：

$$H_0(x,y) \overset{\text{comp}}{\approx} H_{|\mathcal{C}|}(x,y),$$

从而 $\{\mathcal{S}_2(y, f(x,y))\}_{x,y} \overset{\text{comp}}{\approx} \{\text{view}_2^{\Pi}(x,y)\}_{x,y}$.

单一输出确定性功能 $\Rightarrow$ 单一输出概率功能

目标：实现单一输出概率功能 f.

单一输出确定性功能 $\Rightarrow$ 单一输出概率功能

目标：实现单一输出概率功能 f.

定义确定性功能

$$f'((x,r),(y,s)) = f(x,y,r \oplus s),$$

其中 $r, s \in \{0,1\}^{q(n)}$ **为随机掩码**.

单一输出确定性功能 ⇒ 单一输出概率功能

目标：实现单一输出概率功能 f.

定义确定性功能

$$f'((x,r),(y,s)) = f(x,y,r \oplus s),$$

其中 $r, s \in \{0,1\}^{q(n)}$ **为随机掩码**.

协议构造：

- P_1 **采样** $r \leftarrow_{\$} \{0,1\}^{q(n)}$，$P_2$ **采样** $s \leftarrow_{\$} \{0,1\}^{q(n)}$；
- **双方以** (x,r) **与** (y,s) **运行安全实现** f' **的协议** Π'；
- **输出为** $f'((x,r),(y,s)) = f(x,y,r \oplus s)$.

结论：单一输出确定性功能可实现任意单一输出概率功能.

单一输出确定性功能 $\Rightarrow$ 概率功能

当 $f = (f_1, f_2)$ **且** $f_1 \neq f_2$ **时，定义确定性功能**

$$f'((x, r), (y, s)) = (f_1(x, y) \oplus r \,||\, f_2(x, y) \oplus s).$$

单一输出确定性功能 ⇒ 概率功能

当 $f=(f_1,f_2)$ **且** $f_1\neq f_2$ **时，定义确定性功能**

$$f'((x,r),(y,s))=(f_1(x,y)\oplus r\,||\,f_2(x,y)\oplus s).$$

假设 Π' **安全实现** f'**，协议** Π**：**

- P_1 **采样** $r\leftarrow_{\$}\{0,1\}^{q(n)}$，$P_2$ **采样** $s\leftarrow_{\$}\{0,1\}^{q(n)}$；
- **运行** Π' **得到** $(v,w)=f'((x,r),(y,s))$；
- P_1 **输出** $v\oplus r$，P_2 **输出** $w\oplus s$.

单一输出确定性功能 $\Rightarrow$ 概率功能

当 $f=(f_1,f_2)$ **且** $f_1\neq f_2$ **时，定义确定性功能**

$$f'((x,r),(y,s))=(f_1(x,y)\oplus r\,||\,f_2(x,y)\oplus s).$$

假设 Π' **安全实现** f'**，协议** Π**：**

- P_1 **采样** $r\leftarrow_{\$}\{0,1\}^{q(n)}$，$P_2$ **采样** $s\leftarrow_{\$}\{0,1\}^{q(n)}$；
- **运行** Π' **得到** $(v,w)=f'((x,r),(y,s))$；
- P_1 **输出** $v\oplus r$，P_2 **输出** $w\oplus s$.

结论：单一输出确定性功能可实现任意概率功能.

单一输出确定性功能 $\Rightarrow$ 概率功能

当 $f=(f_1,f_2)$ 且 $f_1\neq f_2$ 时，定义确定性功能

$$f'((x,r),(y,s))=(f_1(x,y)\oplus r\,||\,f_2(x,y)\oplus s).$$

假设 Π' 安全实现 f'，协议 Π：

- P_1 采样 $r\leftarrow_{\$}\{0,1\}^{q(n)}$，$P_2$ 采样 $s\leftarrow_{\$}\{0,1\}^{q(n)}$；
- 运行 Π' 得到 $(v,w)=f'((x,r),(y,s))$；
- P_1 输出 $v\oplus r$，P_2 输出 $w\oplus s$.

结论：单一输出确定性功能可实现任意概率功能.

姚氏混淆电路协议的安全性

令 $f=(f_1,f_2)$ 是任意的概率功能. 假设存在加密方案满足选择明文攻击安全 (IND-CPA secure)，且有不可知的密文空间、可验证的密文空间. 那么，存在协议 Π 在 $\mathcal{F}_{\text{OT}}$-混合模型下对于半诚实敌手安全实现了 f.

混淆电路优化技术

表: 混淆电路优化方案及其开销. 列在标识置换之后的方案同样采用了标识置换技术. 混淆表大小以表中密文数量为度量，忽略额外的常数项；计算开销以调用加密/解密算法的次数为度量.

方案	混淆表大小		计算开销			
			生成		计算	
	异或门	与门	异或门	与门	异或门	与门
经典方案	4	4	4	4	2.5	2.5
标识置换	4	4	4	4	1	1
$4 \to 3$ 混淆表行缩减 (GRR3)	3	3	4	4	1	1
$4 \to 2$ 混淆表行缩减 (GRR2)	2	2	4	4	1	1
free-XOR + GRR3	0	3	0	4	0	1
半门	0	2	0	4	0	2

标识置换 (Point-and-Permute)

- 在经典方案中，计算方需要对混淆表每一行都进行解密尝试.
- 标识置换方案为每一个密钥附加了一个标识比特，指示了应该解密混淆表的哪一行.

标识置换 (Point-and-Permute)

- 在经典方案中，计算方需要对混淆表每一行都进行解密尝试.
- 标识置换方案为每一个密钥附加了一个标识比特，指示了应该解密混淆表的哪一行.

表: 标识置换方案，与门的混淆表

w_a 的对应标签	w_b 的对应标签	混淆表
$k_a^0 \|\| (p_a^0 = 1)$	$k_b^0 \|\| (p_b^0 = 0)$	$\mathsf{Enc}_{k_a^1}(\mathsf{Enc}_{k_b^0}(k_c^0 \|\| p_c^0))$
$k_a^0 \|\| (p_a^0 = 1)$	$k_b^1 \|\| (p_b^1 = 1)$	$\mathsf{Enc}_{k_a^1}(\mathsf{Enc}_{k_b^1}(k_c^1 \|\| p_c^1))$
$k_a^1 \|\| (p_a^1 = 0)$	$k_b^0 \|\| (p_b^0 = 0)$	$\mathsf{Enc}_{k_a^0}(\mathsf{Enc}_{k_b^0}(k_c^0 \|\| p_c^0))$
$k_a^1 \|\| (p_a^1 = 0)$	$k_b^1 \|\| (p_b^1 = 1)$	$\mathsf{Enc}_{k_a^0}(\mathsf{Enc}_{k_b^1}(k_c^0 \|\| p_c^0))$

混淆表行缩减 (Garbled Row Reduction, GRR)

GRR3 $(4 \to 3)$：由于混淆表中被加密的标签是随机的，可通过**巧妙选择**使**第一行固定为全 0**，从而只需传输 3 条密文.

混淆表行缩减 (Garbled Row Reduction, GRR)

GRR3 $(4 \to 3)$：由于混淆表中被加密的标签是随机的，可通过**巧妙选择**使**第一行固定为全 0**，从而只需传输 3 条密文.

对门 g_i（输入导线 w_a, w_b，输出 w_c），其双重加密可写为

$$\mathsf{Enc}^{i}_{k_a^{\alpha},k_b^{\beta}}(k_c^{g_i(\alpha,\beta)} \mid\mid p_c^{g_i(\alpha,\beta)}) = H(k_a^{\alpha}||k_b^{\beta}||i) \oplus (k_c^{g_i(\alpha,\beta)}||p_c^{g_i(\alpha,\beta)}),$$

其中 H 可理想化为随机谕示机.

混淆表行缩减 (Garbled Row Reduction, GRR)

GRR3 $(4 \to 3)$：由于混淆表中被加密的标签是随机的，可通过**巧妙选择**使**第一行固定为全 0**，从而只需传输 3 条密文.

对门 g_i（输入导线 w_a, w_b，输出 w_c），其双重加密可写为

$$\mathsf{Enc}^{i}_{k_a^{\alpha},k_b^{\beta}}(k_c^{g_i(\alpha,\beta)} \,||\, p_c^{g_i(\alpha,\beta)}) = H(k_a^{\alpha}||k_b^{\beta}||i) \oplus (k_c^{g_i(\alpha,\beta)}||p_c^{g_i(\alpha,\beta)}),$$

其中 H 可理想化为随机谕示机.

经过**标识置换**排列后，设第一行对应 (α, β)，则令

$$k_c^{g_i(\alpha,\beta)}||p_c^{g_i(\alpha,\beta)} = H(k_a^{\alpha}||k_b^{\beta}||i),$$

使得该行密文为全 0. 于是每个门仅需发送其余 **3 行**密文.

混淆表行缩减：GRR2

GRR2（$4 \to 2$）：通过**多项式插值**将混淆表缩减为 2 行.

混淆表行缩减：GRR2

GRR2（$4 \to 2$）：通过**多项式插值**将混淆表缩减为 2 行.

按真值表输出中 $0/1$ 的个数将门分为：

- **奇门**：0 和 1 的个数为 $1/3$（如 AND/OR）.
- **偶门**：0 和 1 的个数为 $2/2$（如 XOR）.

混淆表行缩减：GRR2

GRR2（$4 \to 2$）：通过**多项式插值**将混淆表缩减为 2 行.

按真值表输出中 $0/1$ 的个数将门分为：

- **奇门**：0 和 1 的个数为 $1/3$（如 AND/OR）.
- **偶门**：0 和 1 的个数为 $2/2$（如 XOR）.

统一记

$$\mathsf{Enc}^{i}_{k_a^\alpha, k_b^\beta}(m) = H(k_a^\alpha || k_b^\beta || i) \oplus m,$$

并令

$$K_r || M_r = H(k_a^\alpha || k_b^\beta || i), \quad r = 2p_a^\alpha + p_b^\beta + 1.$$

其中 K_r 为 κ 比特掩码，M_r 为 1 比特掩码.

GRR2：奇门的构造

奇门（AND/OR）示意：设 $p_a^0 = p_b^0 = 0$，混淆表四行中有三行对应同一输出密钥.

- 对 $(1, K_1), (2, K_2), (3, K_3)$ 插值得 $P(X)$，设 $K_5 = P(5), K_6 = P(6)$；
- 对 $(4, K_4), (5, K_5), (6, K_6)$ 插值得 $Q(X)$；
- 设 $k_c^0 = P(0)$，$k_c^1 = Q(0)$，混淆表仅发送 (K_5, K_6) 与 $c_r = M_r \oplus p_c^{(\cdot)}$（4 个加密标识比特）.
- 计算方通过多项式插值解密混淆表.

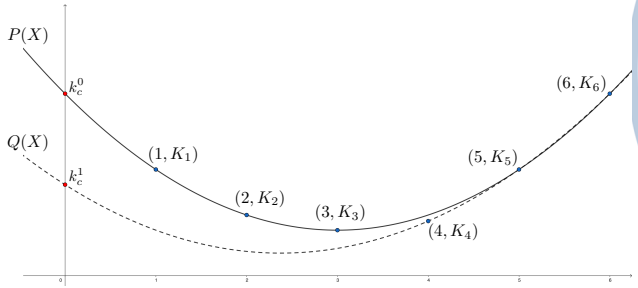


图: GRR2 示意图（奇门）

GRR2：偶门的构造

偶门（XOR）示意：

- 对 $(1, K_1), (4, K_4)$ 插值得 $P(X)$，设 $k_c^0 = P(0)$；
- 对 $(2, K_2), (3, K_3)$ 插值得 $Q(X)$，设 $k_c^1 = Q(0)$；
- 混淆表仅发送两行 $(P(5), Q(5))$ 或 $(Q(5), P(5))$ 与 4 个加密标识比特.
- 计算方通过多项式插值解密混淆表.

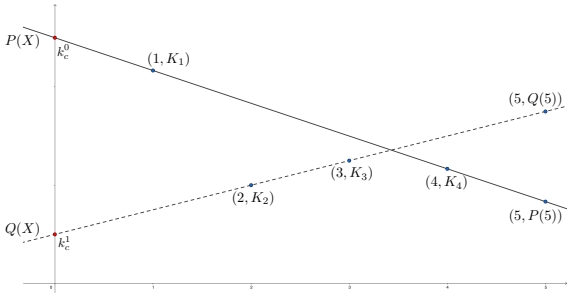


图: GRR2 示意图（偶门）

Free-XOR

目标: 异或门计算变为密钥异或，无需加密表，**通信量为 0**.

Free-XOR

目标: 异或门计算变为密钥异或，无需加密表，**通信量为 0**.

思想:

- **选取全局偏移量** $\Delta \leftarrow_{\$} \{0,1\}^{\kappa}$.
- **标签关联**: $k_i^1 = k_i^0 \oplus \Delta$，$p_i^1 = p_i^0 \oplus 1$.

Free-XOR

目标: 异或门计算变为密钥异或，无需加密表，**通信量为 0**.

思想:

- 选取全局偏移量 $\Delta \leftarrow_{\$} \{0,1\}^{\kappa}$.
- 标签关联：$k_i^1 = k_i^0 \oplus \Delta$，$p_i^1 = p_i^0 \oplus 1$.

XOR 门

$$k_c^0 = k_a^0 \oplus k_b^0, \quad k_c^1 = k_c^0 \oplus \Delta$$
$$p_c^0 = p_a^0 \oplus p_b^0, \quad p_c^1 = p_c^0 \oplus 1$$

Free-XOR

目标：异或门计算变为密钥异或，无需加密表，**通信量为 0**.

思想：

- 选取全局偏移量 $\Delta \leftarrow_{\$} \{0,1\}^{\kappa}$.
- 标签关联：$k_i^1 = k_i^0 \oplus \Delta$，$p_i^1 = p_i^0 \oplus 1$.

XOR 门

$$k_c^0 = k_a^0 \oplus k_b^0, \quad k_c^1 = k_c^0 \oplus \Delta$$
$$p_c^0 = p_a^0 \oplus p_b^0, \quad p_c^1 = p_c^0 \oplus 1$$

非 XOR 门

仍用标识置换 + GRR3：将第一行设为 $H(k_a^{\alpha}||k_b^{\beta}||i)$，并令 $k_c^{1-g} = k_c^g \oplus \Delta$.

Free-XOR

目标：异或门计算变为密钥异或，无需加密表，**通信量为 0**.

思想：

- 选取全局偏移量 $\Delta \leftarrow_{\$} \{0,1\}^{\kappa}$.
- 标签关联：$k_i^1 = k_i^0 \oplus \Delta$，$p_i^1 = p_i^0 \oplus 1$.

XOR 门

$$k_c^0 = k_a^0 \oplus k_b^0, \quad k_c^1 = k_c^0 \oplus \Delta$$
$$p_c^0 = p_a^0 \oplus p_b^0, \quad p_c^1 = p_c^0 \oplus 1$$

解密表：$(0, H(k_i^0)), (1, H(k_i^1))$.

非 XOR 门

仍用标识置换 + GRR3：将第一行设为 $H(k_a^{\alpha}||k_b^{\beta}||i)$，并令 $k_c^{1-g} = k_c^g \oplus \Delta$.

Free-XOR

目标：异或门计算变为密钥异或，无需加密表，**通信量为 0**.

思想：

- 选取全局偏移量 $\Delta \leftarrow_{\$} \{0,1\}^{\kappa}$.
- 标签关联：$k_i^1 = k_i^0 \oplus \Delta$，$p_i^1 = p_i^0 \oplus 1$.

XOR 门

$$k_c^0 = k_a^0 \oplus k_b^0, \quad k_c^1 = k_c^0 \oplus \Delta$$
$$p_c^0 = p_a^0 \oplus p_b^0, \quad p_c^1 = p_c^0 \oplus 1$$

解密表：$(0, H(k_i^0)), (1, H(k_i^1))$.

注意：Free-XOR 与 GRR2 不兼容！

非 XOR 门

仍用标识置换 + GRR3：将第一行设为 $H(k_a^{\alpha}||k_b^{\beta}||i)$，并令 $k_c^{1-g} = k_c^g \oplus \Delta$.

半门 (Half-Gates)

目标: 在兼容 Free-XOR 的前提下，将奇门开销降至 2 个密文.

半门 (Half-Gates)

目标: 在兼容 Free-XOR 的前提下，将奇门开销降至 2 个密文.

原理: 将奇门分解为两个“半门”的异或（$\alpha_a, \alpha_b, \alpha_c$ 由门的类型决定）:

$$g(v_a, v_b) = (\alpha_a \oplus v_a) \wedge (\alpha_b \oplus v_b) \oplus \alpha_c$$

$$= ((v_a \oplus \alpha_a) \wedge (r \oplus \alpha_b) \oplus \alpha_c) \oplus ((v_a \oplus \alpha_a) \wedge (r \oplus v_b))$$

半门 (Half-Gates)

目标: 在兼容 Free-XOR 的前提下，将奇门开销降至 2 个密文.

原理: 将奇门分解为两个“半门”的异或（$\alpha_a, \alpha_b, \alpha_c$ 由门的类型决定）:

$$g(v_a, v_b) = (\alpha_a \oplus v_a) \wedge (\alpha_b \oplus v_b) \oplus \alpha_c$$

$$= ((v_a \oplus \alpha_a) \wedge (r \oplus \alpha_b) \oplus \alpha_c) \oplus ((v_a \oplus \alpha_a) \wedge (r \oplus v_b))$$

- **生成方半门（左）**：生成方已知 $(r \oplus \alpha_b)$ 的明文值.
- **计算方半门（右）**：计算方已知 $(v_a \oplus \alpha_a)$ 的明文值.
- **GRR 技术**: 每个半门仅需 2-1=1 个密文 $\Longrightarrow$ 总共 2 个密文.

半门：以与门为例

令 $v_c = v_a \wedge v_b$. 取随机比特 r，有

$$v_c = (r \wedge v_a) \oplus ((r \oplus v_b) \wedge v_a).$$

取 $r = p_b^0$（w_b 的标识比特），则计算方从 $p_b^{v_b}$ 得到 $r \oplus v_b$.

半门：以与门为例

令 $v_c = v_a \wedge v_b$. 取随机比特 r，有

$$v_c = (r \wedge v_a) \oplus ((r \oplus v_b) \wedge v_a).$$

取 $r = p_b^0$（w_b 的标识比特），则计算方从 $p_b^{v_b}$ 得到 $r \oplus v_b$.

生成方半门（已知 r）：

$$H(W_b^0) \oplus W_G^0$$
$$H(W_b^1) \oplus W_G^0 \oplus r\Delta$$

依据标识比特排列并用 GRR 缩减一条密文.

半门：以与门为例

令 $v_c = v_a \wedge v_b$. 取随机比特 r，有

$$v_c = (r \wedge v_a) \oplus ((r \oplus v_b) \wedge v_a).$$

取 $r = p_b^0$（w_b 的标识比特），则计算方从 $p_b^{v_b}$ 得到 $r \oplus v_b$.

生成方半门（已知 r）：

$$H(W_b^0) \oplus W_G^0$$
$$H(W_b^1) \oplus W_G^0 \oplus r\Delta$$

依据标识比特排列并用 GRR 缩减一条密文.
计算方持有 $W_b^{v_b}$：解密得 W_G^0 或 $W_G^0 \oplus r\Delta$.

半门：以与门为例

令 $v_c = v_a \wedge v_b$. 取随机比特 r，有

$$v_c = (r \wedge v_a) \oplus ((r \oplus v_b) \wedge v_a).$$

取 $r = p_b^0$（w_b 的标识比特），则计算方从 $p_b^{v_b}$ 得到 $r \oplus v_b$.

生成方半门（已知 r）：

$$H(W_b^0) \oplus W_G^0$$
$$H(W_b^1) \oplus W_G^0 \oplus r\Delta$$

依据标识比特排列并用 GRR 缩减一条密文.

计算方半门（已知 v_a）：

$$H(W_a^0) \oplus W_E^0$$
$$H(W_a^1) \oplus W_E^0 \oplus W_b^0$$

不打乱顺序并用 GRR 缩减一条密文.

计算方持有 $W_b^{v_b}$：解密得 W_G^0 或 $W_G^0 \oplus r\Delta$.

半门：以与门为例

令 $v_c = v_a \wedge v_b$. 取随机比特 r，有

$$v_c = (r \wedge v_a) \oplus ((r \oplus v_b) \wedge v_a).$$

取 $r = p_b^0$（w_b 的标识比特），则计算方从 $p_b^{v_b}$ 得到 $r \oplus v_b$.

生成方半门（已知 r）：

$$H(W_b^0) \oplus W_G^0$$
$$H(W_b^1) \oplus W_G^0 \oplus r\Delta$$

依据标识比特排列并用 GRR 缩减一条密文.
计算方持有 $W_b^{v_b}$：解密得 W_G^0 或 $W_G^0 \oplus r\Delta$.

计算方半门（已知 v_a）：

$$H(W_a^0) \oplus W_E^0$$
$$H(W_a^1) \oplus W_E^0 \oplus W_b^0$$

不打乱顺序并用 GRR 缩减一条密文.
若 $v_a = 0$ 解密第 1 行得 W_E^0;
若 $v_a = 1$ 解密第 2 行并再异或 $W_b^{v_b}$.

半门：以与门为例

令 $v_c = v_a \wedge v_b$. 取随机比特 r，有

$$v_c = (r \wedge v_a) \oplus ((r \oplus v_b) \wedge v_a).$$

取 $r = p_b^0$（w_b 的标识比特），则计算方从 $p_b^{v_b}$ 得到 $r \oplus v_b$.

生成方半门（已知 r）：

$$H(W_b^0) \oplus W_G^0$$
$$H(W_b^1) \oplus W_G^0 \oplus r\Delta$$

依据标识比特排列并用 GRR 缩减一条密文.
计算方持有 $W_b^{v_b}$：解密得 W_G^0 或 $W_G^0 \oplus r\Delta$.

计算方半门（已知 v_a）：

$$H(W_a^0) \oplus W_E^0$$
$$H(W_a^1) \oplus W_E^0 \oplus W_b^0$$

不打乱顺序并用 GRR 缩减一条密文.
若 $v_a = 0$ 解密第 1 行得 W_E^0；
若 $v_a = 1$ 解密第 2 行并再异或 $W_b^{v_b}$.

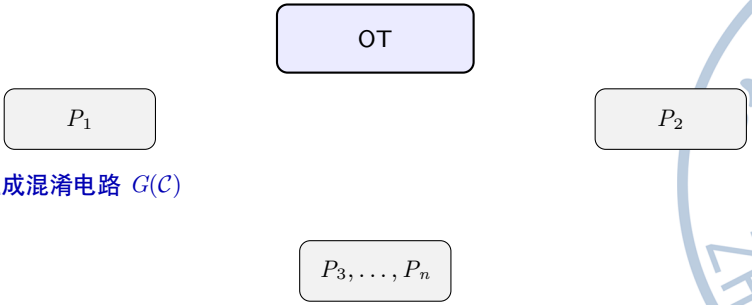
合并输出：计算方取 $W_c = W_G \oplus W_E$，得到与门输出标签.

混淆电路优化技术

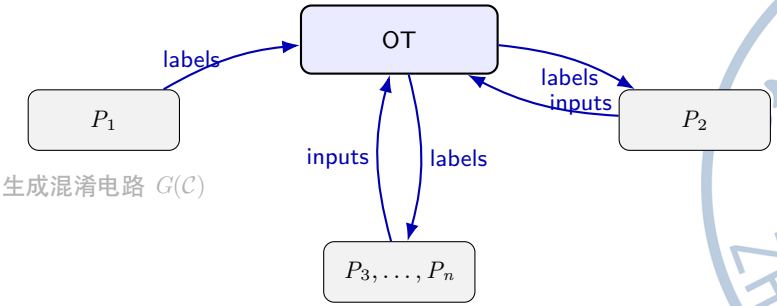
表: 混淆电路优化方案及其开销. 列在标识置换之后的方案同样采用了标识置换技术. 混淆表大小以表中密文数量为度量，忽略额外的常数项；计算开销以调用加密/解密算法的次数为度量.

方案	混淆表大小		计算开销			
			生成		计算	
	异或门	与门	异或门	与门	异或门	与门
经典方案	4	4	4	4	2.5	2.5
标识置换	4	4	4	4	1	1
$4 \to 3$ 混淆表行缩减 (GRR3)	3	3	4	4	1	1
$4 \to 2$ 混淆表行缩减 (GRR2)	2	2	4	4	1	1
free-XOR + GRR3	0	3	0	4	0	1
半门	0	2	0	4	0	2

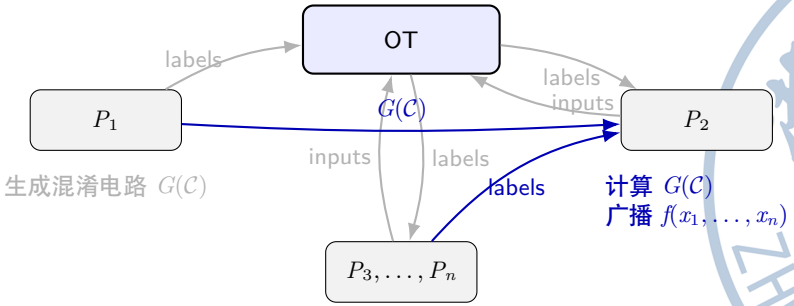
n 方混淆电路协议（尝试）



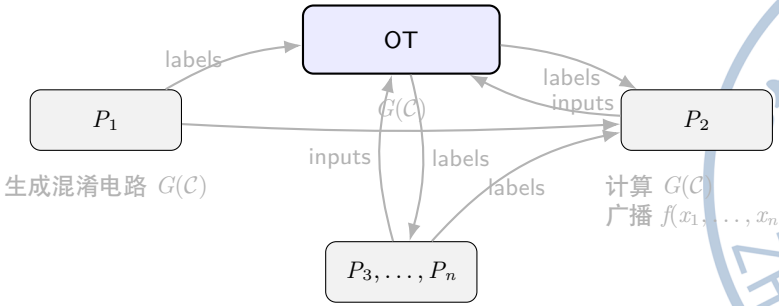
n 方混淆电路协议（尝试）


OT
labels
labels
inputs
P_1
P_2
inputs
labels
$P_3, \ldots, P_n$
生成混淆电路 $G(C)$

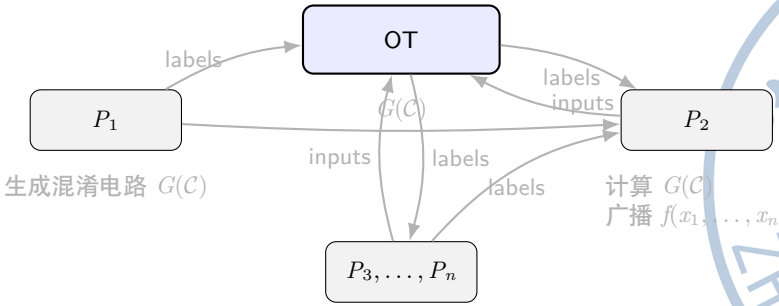
n 方混淆电路协议（尝试）


OT
labels
labels
inputs
P_1
$G(C)$
P_2
inputs
labels
labels
生成混淆电路 $G(C)$
计算 $G(C)$
广播 $f(x_1, \ldots, x_n)$
$P_3, \ldots, P_n$

n 方混淆电路协议（尝试）

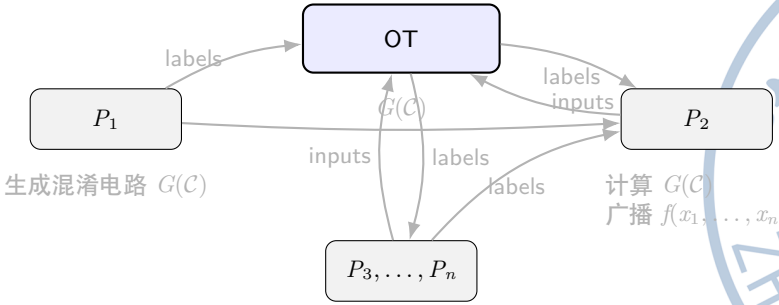

OT
labels
labels
inputs
P_1
$G(C)$
P_2
inputs
labels
labels
生成混淆电路 $G(C)$
计算 $G(C)$
广播 $f(x_1, \ldots, x_n)$
$P_3, \ldots, P_n$

n 方混淆电路协议（尝试）



如果 P_1 和 P_2 同时被攻陷，那么敌手可以获得所有参与方的输入！

n 方混淆电路协议（尝试）



如果 P_1 和 P_2 同时被攻陷，那么敌手可以获得所有参与方的输入！
应该让所有参与方**调用 MPC 协议**生成混淆电路.

BMR(Beaver-Micali-Rogaway) 协议

目标: 将姚氏 GC 扩展到 n 方，抵抗至多 t 个被攻陷方.

BMR(Beaver-Micali-Rogaway) 协议

目标: 将姚氏 GC 扩展到 n 方，抵抗至多 t 个被攻陷方.

- 记电路 $\mathcal{C}$ 计算 $f(x_1, \ldots, x_n)$，导线真实值 v_i.

BMR(Beaver-Micali-Rogaway) 协议

目标: 将姚氏 GC 扩展到 n 方，抵抗至多 t 个被攻陷方.

- 记电路 $\mathcal{C}$ 计算 $f(x_1, \ldots, x_n)$，导线真实值 v_i.
- **掩码比特**: 参与方 P_j 选取 $\lambda_{i,j} \leftarrow_{\$} \{0,1\}$，导线 w_i 的掩码比特为 $\lambda_i = \bigoplus_{j=1}^{n} \lambda_{i,j}$.

BMR(Beaver-Micali-Rogaway) 协议

目标: 将姚氏 GC 扩展到 n 方，抵抗至多 t 个被攻陷方.

- 记电路 $\mathcal{C}$ 计算 $f(x_1, \ldots, x_n)$，导线真实值 v_i.
- **掩码比特**: 参与方 P_j 选取 $\lambda_{i,j} \leftarrow_{\$} \{0,1\}$，导线 w_i 的掩码比特为 $\lambda_i = \bigoplus_{j=1}^{n} \lambda_{i,j}$.
- **外部值**: $p_i = v_i \oplus \lambda_i$.

BMR(Beaver-Micali-Rogaway) 协议

目标: 将姚氏 GC 扩展到 n 方，抵抗至多 t 个被攻陷方.

- 记电路 $\mathcal{C}$ 计算 $f(x_1, \ldots, x_n)$，导线真实值 v_i.
- **掩码比特**: 参与方 P_j 选取 $\lambda_{i,j} \leftarrow_{\$} \{0,1\}$，导线 w_i 的掩码比特为 $\lambda_i = \bigoplus_{j=1}^{n} \lambda_{i,j}$.
- **外部值**: $p_i = v_i \oplus \lambda_i$.
- **标签（上标为外部值）**:
  - $S_i^0 = s_{i,1}^0 || \cdots || s_{i,n}^0 || 0$
  - $S_i^1 = s_{i,1}^1 || \cdots || s_{i,n}^1 || 1$

  $s_{i,j}^0, s_{i,j}^1 \leftarrow_{\$} \{0,1\}^{\kappa}$ 为参与方 P_j 为 w_i 选取的**子标签**.

BMR(Beaver-Micali-Rogaway) 协议

目标: 将姚氏 GC 扩展到 n 方，抵抗至多 t 个被攻陷方.

- 记电路 $\mathcal{C}$ 计算 $f(x_1, \ldots, x_n)$，导线真实值 v_i.
- **掩码比特**: 参与方 P_j 选取 $\lambda_{i,j} \leftarrow_{\$} \{0,1\}$，导线 w_i 的掩码比特为 $\lambda_i = \bigoplus_{j=1}^{n} \lambda_{i,j}$.
- **外部值**: $p_i = v_i \oplus \lambda_i$.
- **标签（上标为外部值）**:
  - $S_i^0 = s_{i,1}^0 || \cdots || s_{i,n}^0 || 0$
  - $S_i^1 = s_{i,1}^1 || \cdots || s_{i,n}^1 || 1$

  $s_{i,j}^0, s_{i,j}^1 \leftarrow_{\$} \{0,1\}^\kappa$ 为参与方 P_j 为 w_i 选取的**子标签**.
- PRG $\mathcal{G} : \{0,1\}^\kappa \to \{0,1\}^{2n\kappa+2}$，定义
  - $g_{i,j}^p = \mathcal{G}(s_{i,j}^p)[1 : n\kappa + 1]$
  - $h_{i,j}^p = \mathcal{G}(s_{i,j}^p)[n\kappa + 2 : 2n\kappa + 2]$

BMR：混淆表的分布式生成

对门 g，输入导线 w_a, w_b，输出导线 w_c. 参与方用底层 MPC 协议安全计算混淆表 (A_g, B_g, C_g, D_g)：

$$A_g = \bigoplus_{j=1}^{n} g_{a,j}^{0} \oplus \bigoplus_{j=1}^{n} g_{b,j}^{0} \oplus \begin{cases} S_c^0 & \text{if } g(\lambda_a, \lambda_b) = \lambda_c \\ S_c^1 & \text{otherwise} \end{cases}$$

$$B_g = \bigoplus_{j=1}^{n} h_{a,j}^{0} \oplus \bigoplus_{j=1}^{n} g_{b,j}^{1} \oplus \begin{cases} S_c^0 & \text{if } g(\lambda_a, \overline{\lambda_b}) = \lambda_c \\ S_c^1 & \text{otherwise} \end{cases}$$

$$C_g = \bigoplus_{j=1}^{n} g_{a,j}^{1} \oplus \bigoplus_{j=1}^{n} h_{b,j}^{0} \oplus \begin{cases} S_c^0 & \text{if } g(\overline{\lambda_a}, \lambda_b) = \lambda_c \\ S_c^1 & \text{otherwise} \end{cases}$$

$$D_g = \bigoplus_{j=1}^{n} h_{a,j}^{1} \oplus \bigoplus_{j=1}^{n} h_{b,j}^{1} \oplus \begin{cases} S_c^0 & \text{if } g(\overline{\lambda_a}, \overline{\lambda_b}) = \lambda_c \\ S_c^1 & \text{otherwise} \end{cases}$$

BMR：输入标签与混淆输入

对每条输入导线 w_i：

BMR：输入标签与混淆输入

对每条输入导线 w_i：

1. 输入值 v_i 先被秘密分享；

BMR：输入标签与混淆输入

对每条输入导线 w_i：

1. 输入值 v_i 先被秘密分享；
2. 各方安全计算并公开外部值 $p_i = v_i \oplus \lambda_i$；

BMR：输入标签与混淆输入

对每条输入导线 w_i：

1. 输入值 v_i 先被秘密分享；
2. 各方安全计算并公开外部值 $p_i = v_i \oplus \lambda_i$；
3. 各方广播 $s_{i,j}^{p_i}$，得到输入标签

$$S_i^{p_i} = s_{i,1}^{p_i} || \cdots || s_{i,n}^{p_i} || p_i.$$

BMR：输入标签与混淆输入

对每条输入导线 w_i：

1. 输入值 v_i 先被秘密分享；
2. 各方安全计算并公开外部值 $p_i = v_i \oplus \lambda_i$；
3. 各方广播 $s_{i,j}^{p_i}$，得到输入标签

$$S_i^{p_i} = s_{i,1}^{p_i} || \cdots || s_{i,n}^{p_i} || p_i.$$

混淆输入：所有输入导线的 $S_i^{p_i}$ 组成 garbled inputs.

BMR：计算与输出恢复

给定输入标签 $S_a^{p_a}, S_b^{p_b}$：

BMR：计算与输出恢复

给定输入标签 $S_a^{p_a}, S_b^{p_b}$：

1. 各方本地计算 $g_{a,j}^{p_a}, h_{a,j}^{p_a}, g_{b,j}^{p_b}, h_{b,j}^{p_b}$；

BMR：计算与输出恢复

给定输入标签 $S_a^{p_a}, S_b^{p_b}$：

1. 各方本地计算 $g_{a,j}^{p_a}, h_{a,j}^{p_a}, g_{b,j}^{p_b}, h_{b,j}^{p_b}$；
2. 按 p_a, p_b 选择混淆表项，计算输出标签

$$S_c^{p_c} = \begin{cases} \bigoplus_j g_{a,j}^{p_a} \oplus \bigoplus_j g_{b,j}^{p_b} \oplus A_g & (p_a, p_b) = (0,0) \\ \bigoplus_j h_{a,j}^{p_a} \oplus \bigoplus_j g_{b,j}^{p_b} \oplus B_g & (p_a, p_b) = (0,1) \\ \bigoplus_j g_{a,j}^{p_a} \oplus \bigoplus_j h_{b,j}^{p_b} \oplus C_g & (p_a, p_b) = (1,0) \\ \bigoplus_j h_{a,j}^{p_a} \oplus \bigoplus_j h_{b,j}^{p_b} \oplus D_g & (p_a, p_b) = (1,1) \end{cases}$$

BMR：计算与输出恢复

给定输入标签 $S_a^{p_a}, S_b^{p_b}$：

1. 各方本地计算 $g_{a,j}^{p_a}, h_{a,j}^{p_a}, g_{b,j}^{p_b}, h_{b,j}^{p_b}$；
2. 按 p_a, p_b 选择混淆表项，计算输出标签

$$S_c^{p_c} = \begin{cases} \bigoplus_j g_{a,j}^{p_a} \oplus \bigoplus_j g_{b,j}^{p_b} \oplus A_g & (p_a, p_b) = (0,0) \\ \bigoplus_j h_{a,j}^{p_a} \oplus \bigoplus_j g_{b,j}^{p_b} \oplus B_g & (p_a, p_b) = (0,1) \\ \bigoplus_j g_{a,j}^{p_a} \oplus \bigoplus_j h_{b,j}^{p_b} \oplus C_g & (p_a, p_b) = (1,0) \\ \bigoplus_j h_{a,j}^{p_a} \oplus \bigoplus_j h_{b,j}^{p_b} \oplus D_g & (p_a, p_b) = (1,1) \end{cases}$$

输出导线 w_j 的外部值为 p_j（标签末位），公开 λ_j 后恢复

$$v_j = p_j \oplus \lambda_j.$$

BMR 协议的安全性和效率

BMR 协议的安全性和效率

- **安全性来源**：通过调用底层 MPC 协议（如 BGW）的模拟器，模拟器可生成“假的”混淆电路，安全性继承底层协议，门限与底层协议相同.

BMR 协议的安全性和效率

- **安全性来源**：通过调用底层 MPC 协议（如 BGW）的模拟器，模拟器可生成“假的”混淆电路，安全性继承底层协议，门限与底层协议相同.
- **轮数复杂度**：输入分享 + 并行计算混淆表 + 并行计算输入标签，轮数为常数，与电路深度无关. 混淆表公开后，各方本地解密评估电路.

BMR 协议的安全性和效率

- **安全性来源**：通过调用底层 MPC 协议（如 BGW）的模拟器，模拟器可生成“假的”混淆电路，安全性继承底层协议，门限与底层协议相同.
- **轮数复杂度**：输入分享 + 并行计算混淆表 + 并行计算输入标签，轮数为常数，与电路深度无关. 混淆表公开后，各方本地解密评估电路.

我们实现了**常数轮**的 MPC 协议！

本章总结

1. **姚氏混淆电路**: 2PC 的基石，将计算转化为解密.
2. **安全性**: 在独立模型下对于半诚实敌手安全实现了任意概率功能.
3. **混淆电路优化**: 减小混淆电路大小，提高计算效率.
4. **BMR**: 将 GC 思想推广至多方，实现了常数轮 MPC.

本章总结

1. **姚氏混淆电路**: 2PC 的基石，将计算转化为解密.
2. **安全性**: 在独立模型下对于半诚实敌手安全实现了任意概率功能.
3. **混淆电路优化**: 减小混淆电路大小，提高计算效率.
4. **BMR**: 将 GC 思想推广至多方，实现了常数轮 MPC.

下一章，我们将学习如何将半诚实安全的 *GC* 升级为恶意安全*!*

思考题

1. 姚氏混淆电路协议中的对称加密算法和随机谕示机有哪些高效的实现方式？这些实现方法在应对大规模（亿级）门电路时是否引入了额外的安全风险？

思考题

1. 姚氏混淆电路协议中的对称加密算法和随机谕示机有哪些高效的实现方式？这些实现方法在应对大规模（亿级）门电路时是否引入了额外的安全风险？
2. 姚氏混淆电路协议和 BMR 协议都是常数轮复杂度的多方安全计算协议．请问常数轮协议在哪些实际应用场景具有优势？

思考题

1. 姚氏混淆电路协议中的对称加密算法和随机谕示机有哪些高效的实现方式？这些实现方法在应对大规模（亿级）门电路时是否引入了额外的安全风险？
2. 姚氏混淆电路协议和 BMR 协议都是常数轮复杂度的多方安全计算协议．请问常数轮协议在哪些实际应用场景具有优势？
3. 如果一个安全多方计算协议将会在虚拟机上运行，假设敌手可以任意暂停或者倒带 (rewind) 此虚拟机，这对协议的安全性提出了哪些要求？

Q & A