恶意安全性
切分选择

《安全多方计算——可证明安全视角》第九章第二节

2026 年 2 月 12 日

目录

切分选择：动机

问题：

- 姚氏混淆电路在半诚实模型下安全.

切分选择：动机

问题：

- 姚氏混淆电路在半诚实模型下安全.
- 但在恶意模型中，电路生成方 P_1 可能发送错误电路 C'，导致隐私泄露甚至直接暴露 P_2 的输入.

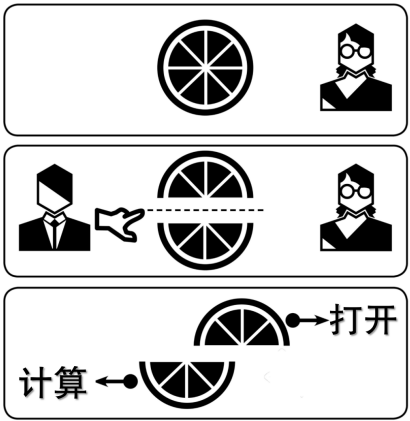
切分选择：动机

问题：

- 姚氏混淆电路在半诚实模型下安全.
- 但在恶意模型中，电路生成方 P_1 可能发送错误电路 C'，导致隐私泄露甚至直接暴露 P_2 的输入.

切分选择 (Cut-and-Choose)

- P_1 生成 s 个混淆电路副本.
- P_2 随机选择 $\frac{s}{2}$ 个电路要求打开并验证.
- 若打开电路全部正确，则对剩余电路求值并取多数作为输出.



欺骗成功概率：事件建模

设 P_1 生成 s 个电路，P_2 打开其中 $\frac{s}{2}$ 个.

欺骗成功概率：事件建模

设 P_1 生成 s 个电路，P_2 打开其中 $\frac{s}{2}$ 个.

- badTotal：错误电路总数.
- noAbort：打开的 $\frac{s}{2}$ 个电路全部正确.
- badMaj：未打开的 $\frac{s}{2}$ 个电路中，错误电路不少于一半.

欺骗成功概率：事件建模

设 P_1 生成 s 个电路，P_2 打开其中 $\frac{s}{2}$ 个.

- badTotal：错误电路总数.
- noAbort：打开的 $\frac{s}{2}$ 个电路全部正确.
- badMaj：未打开的 $\frac{s}{2}$ 个电路中，错误电路不少于一半.

我们关心

$$\Pr[\mathsf{noAbort} \wedge \mathsf{badMaj}] = \sum_{i=s/4}^{s/2} \Pr[\mathsf{noAbort} \wedge \mathsf{badTotal} = i].$$

欺骗成功概率：组合计算

当 badTotal $= i$ 时，noAbort 发生需从 $s - i$ 个正确电路中选出 $\frac{s}{2}$ 个：

$$\Pr[\text{noAbort} \mid \text{badTotal} = i] = \frac{\binom{s-i}{s/2}}{\binom{s}{s/2}}.$$

欺骗成功概率：组合计算

当 $\mathsf{badTotal} = i$ 时，noAbort 发生需从 $s - i$ 个正确电路中选出 $\frac{s}{2}$ 个：

$$\Pr[\mathsf{noAbort} \mid \mathsf{badTotal} = i] = \frac{\binom{s-i}{s/2}}{\binom{s}{s/2}}.$$

因此

$$\begin{aligned}\Pr[\mathsf{noAbort} \wedge \mathsf{badMaj}] &\leq \sum_{i=s/4}^{s/2} \frac{\binom{s-i}{s/2}}{\binom{s}{s/2}} \\ &= \frac{1}{\binom{s}{s/2}} \sum_{i=s/4}^{s/2} \binom{s-i}{s/2}.\end{aligned}$$

欺骗成功概率：组合计算

当 $\mathsf{badTotal} = i$ 时，$\mathsf{noAbort}$ 发生需从 $s - i$ 个正确电路中选出 $\frac{s}{2}$ 个：

$$\Pr[\mathsf{noAbort} \mid \mathsf{badTotal} = i] = \frac{\binom{s-i}{s/2}}{\binom{s}{s/2}}.$$

因此

$$\begin{aligned}\Pr[\mathsf{noAbort} \wedge \mathsf{badMaj}] &\leq \sum_{i=s/4}^{s/2} \frac{\binom{s-i}{s/2}}{\binom{s}{s/2}} \\ &= \frac{1}{\binom{s}{s/2}} \sum_{i=s/4}^{s/2} \binom{s-i}{s/2}.\end{aligned}$$

令 $j = s - i$，并使用恒等式 $\sum_{k=0}^{n} \binom{k}{m} = \binom{n+1}{m+1}$，得

$$\Pr[\mathsf{noAbort} \wedge \mathsf{badMaj}] = \frac{\binom{3s/4+1}{s/2+1}}{\binom{s}{s/2}}.$$

欺骗成功概率：严格上界

$$\frac{\binom{3s/4+1}{s/2+1}}{\binom{s}{s/2}} = \frac{(\frac{3s}{4}+1)!}{(\frac{s}{2}+1)!(\frac{s}{4})!} \cdot \frac{(\frac{s}{2})!(\frac{s}{2})!}{s!}$$
$$= \frac{(\frac{s}{2})(\frac{s}{2}-1)\cdots(\frac{s}{4}+1)}{s(s-1)\cdots(\frac{3s}{4}+2)} \cdot \frac{1}{\frac{s}{2}+1}.$$

欺骗成功概率：严格上界

$$\frac{\binom{3s/4+1}{s/2+1}}{\binom{s}{s/2}} = \frac{(\frac{3s}{4}+1)!}{(\frac{s}{2}+1)!(\frac{s}{4})!} \cdot \frac{(\frac{s}{2})!(\frac{s}{2})!}{s!}$$
$$= \frac{(\frac{s}{2})(\frac{s}{2}-1)\cdots(\frac{s}{4}+1)}{s(s-1)\cdots(\frac{3s}{4}+2)} \cdot \frac{1}{\frac{s}{2}+1}.$$

令 $t = s/4$，则

$$\frac{2t}{4t} \cdot \frac{2t-1}{4t-1} \cdots \frac{t+2}{3t+2} \cdot \frac{t+1}{2t+1} = \left(\prod_{i=2}^{t} \frac{t+i}{3t+i}\right) \cdot \frac{t+1}{2t+1}.$$

欺骗成功概率：严格上界

$$\frac{\binom{3s/4+1}{s/2+1}}{\binom{s}{s/2}} = \frac{(\frac{3s}{4}+1)!}{(\frac{s}{2}+1)!(\frac{s}{4})!} \cdot \frac{(\frac{s}{2})!(\frac{s}{2})!}{s!}$$
$$= \frac{(\frac{s}{2})(\frac{s}{2}-1)\cdots(\frac{s}{4}+1)}{s(s-1)\cdots(\frac{3s}{4}+2)} \cdot \frac{1}{\frac{s}{2}+1}.$$

令 $t = s/4$，则

$$\frac{2t}{4t} \cdot \frac{2t-1}{4t-1} \cdots \frac{t+2}{3t+2} \cdot \frac{t+1}{2t+1} = \left(\prod_{i=2}^{t} \frac{t+i}{3t+i}\right) \cdot \frac{t+1}{2t+1}.$$

因 $\frac{t+i}{3t+i} < \frac{1}{2}$（对所有 $i < t$），故

$$\Pr[\mathsf{noAbort} \wedge \mathsf{badMaj}] < \frac{1}{2^{t-1}} = \frac{1}{2^{s/4-1}}.$$

欺骗成功概率：严格上界

$$\frac{\binom{3s/4+1}{s/2+1}}{\binom{s}{s/2}} = \frac{(\frac{3s}{4}+1)!}{(\frac{s}{2}+1)!(\frac{s}{4})!} \cdot \frac{(\frac{s}{2})!(\frac{s}{2})!}{s!}$$
$$= \frac{(\frac{s}{2})(\frac{s}{2}-1)\cdots(\frac{s}{4}+1)}{s(s-1)\cdots(\frac{3s}{4}+2)} \cdot \frac{1}{\frac{s}{2}+1}.$$

令 $t = s/4$，则

$$\frac{2t}{4t} \cdot \frac{2t-1}{4t-1} \cdots \frac{t+2}{3t+2} \cdot \frac{t+1}{2t+1} = \left(\prod_{i=2}^{t} \frac{t+i}{3t+i}\right) \cdot \frac{t+1}{2t+1}.$$

因 $\frac{t+i}{3t+i} < \frac{1}{2}$（对所有 $i < t$），故

$$\Pr[\mathsf{noAbort} \wedge \mathsf{badMaj}] < \frac{1}{2^{t-1}} = \frac{1}{2^{s/4-1}}.$$

例如取 $s = 164$，则该概率 $< 2^{-40}$.

不一致输出与多数票原则

注意：即便出现不一致，P_2 也应输出**多数电路**的结果.

不一致输出与多数票原则

注意：即便出现不一致，P_2 也应输出**多数电路**的结果.

Q: 为什么不能直接中止？

不一致输出与多数票原则

注意：即便出现不一致，P_2 也应输出**多数电路**的结果.

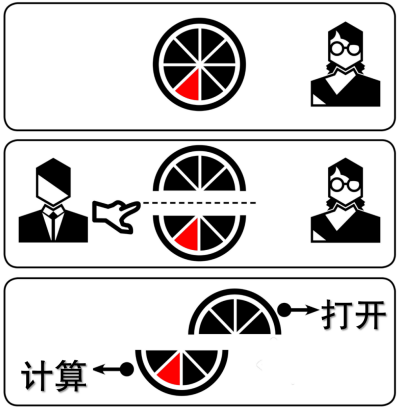
Q: 为什么不能直接中止？

为什么不能直接中止？

若 P_2 在输出不一致时立刻中止，P_1 可实施如下攻击：（y_1 为 P_2 输入的第一位）

- P_1 构造一个“特殊”电路：当 $y_1 = 1$ 输出错误，当 $y_1 = 0$ 输出正确.
- 其余 $s - 1$ 个电路正确.
- 该错误电路被抽检概率为 $1/2$.

若未被抽检，P_1 通过 P_2 是否中止学习 y_1.



输入一致性 (Input Consistency)

问题：P_1 的输入可能在不同电路不一致

输入一致性 (Input Consistency)

问题：P_1 的输入可能在不同电路不一致

- P_2 需要在多个未打开电路上计算，必须保证这些电路使用**同一份输入**.

输入一致性 (Input Consistency)

问题：P_1 的输入可能在不同电路不一致

- P_2 需要在多个未打开电路上计算，必须保证这些电路使用**同一**份输入.
- 否则 P_1 可跨电路“编程”输入以泄露信息.

输入一致性 (Input Consistency)

问题：P_1 的输入可能在不同电路不一致

- P_2 需要在多个未打开电路上计算，必须保证这些电路使用**同一份输入**.
- 否则 P_1 可跨电路“编程”输入以泄露信息.

示例：内积泄露

设 $x, y \in \{0,1\}^n$，功能 $f(x,y) = \langle x, y\rangle = \sum_{i=1}^{n} x_i y_i$. 若有 n 个电路，P_1 在第 i 个电路中令 $x_i = 1$ 且其余为 0，则 P_2 对第 i 个电路的输出即为 y_i. 多数输出将暴露 y 中多数位为 0 还是 1.

输入一致性：典型做法

承诺 + 零知识证明

输入一致性：典型做法

承诺 + 零知识证明

- 对每条输入导线，P_1 承诺所有电路中该导线的 $0/1$ 密钥，形成“0 集合”和“1 集合”，大小均为 s.

输入一致性：典型做法

承诺 + 零知识证明

- 对每条输入导线，P_1 承诺所有电路中该导线的 $0/1$ 密钥，形成“0 集合”和“1 集合”，大小均为 s.
- 抽检电路时，P_1 同时打开相应承诺，P_2 验证一致性.

输入一致性：典型做法

承诺 + 零知识证明

- 对每条输入导线，P_1 承诺所有电路中该导线的 $0/1$ 密钥，形成“0 集合”和“1 集合”，大小均为 s.
- 抽检电路时，P_1 同时打开相应承诺，P_2 验证一致性.
- 在未打开电路上，P_1 需零知识证明：所有发送的输入密钥**都在“0 集合”中或都在“1 集合”中**.

要点：确保用于计算的大多数电路中 P_1 的输入一致.

选择性中止攻击 (Selective Abort/Failure)

攻击方式

选择性中止攻击 (Selective Abort/Failure)

攻击方式

- 在 OT 中，P_1 应提供 (k_i^0, k_i^1).

选择性中止攻击 (Selective Abort/Failure)

攻击方式

- 在 OT 中，P_1 应提供 (k_i^0, k_i^1).
- 若恶意 P_1 发送 $(k_i^0, 0)$，则 P_2 在输入位为 0 时正常，输入位为 1 时中止.

选择性中止攻击 (Selective Abort/Failure)

攻击方式

- 在 OT 中，P_1 应提供 (k_i^0, k_i^1).
- 若恶意 P_1 发送 $(k_i^0, 0)$，则 P_2 在输入位为 0 时正常，输入位为 1 时中止.
- P_1 通过观察是否中止，学习 P_2 的输入比特.

选择性中止攻击 (Selective Abort/Failure)

攻击方式

- 在 OT 中，P_1 应提供 (k_i^0, k_i^1).
- 若恶意 P_1 发送 $(k_i^0, 0)$，则 P_2 在输入位为 0 时正常，输入位为 1 时中止.
- P_1 通过观察是否中止，学习 P_2 的输入比特.

防御

选择性中止攻击 (Selective Abort/Failure)

攻击方式

- 在 OT 中，P_1 应提供 (k_i^0, k_i^1).
- 若恶意 P_1 发送 $(k_i^0, 0)$，则 P_2 在输入位为 0 时正常，输入位为 1 时中止.
- P_1 通过观察是否中止，学习 P_2 的输入比特.

防御

- 使用 committing OT：OT 同时绑定 P_1 的输入，抽检电路时一并打开并验证.

选择性中止攻击 (Selective Abort/Failure)

攻击方式

- 在 OT 中，P_1 应提供 (k_i^0, k_i^1).
- 若恶意 P_1 发送 $(k_i^0, 0)$，则 P_2 在输入位为 0 时正常，输入位为 1 时中止.
- P_1 通过观察是否中止，学习 P_2 的输入比特.

防御

- 使用 committing OT：OT 同时绑定 P_1 的输入，抽检电路时一并打开并验证.
- 或使用 cut-and-choose OT：抽检索引直接暴露两条 OT 输入，迫使 P_1 在大多数 OT 中诚实.

输出真实性 (Output Authenticity)

- 在恶意模型下，P_2 可能向 P_1 发送任意结果.

输出真实性 (Output Authenticity)

- 在恶意模型下，P_2 可能向 P_1 发送任意结果.
- **公平性**无法实现，但可以保证：若 P_2 发送输出，P_1 可验证其真实性.

输出真实性 (Output Authenticity)

- 在恶意模型下，P_2 可能向 P_1 发送任意结果.
- **公平性**无法实现，但可以保证：若 P_2 发送输出，P_1 可验证其真实性.

基于 MAC 的构造

设功能 $f = (f_1, f_2)$，只允许 P_2 先获得输出. 取域 $\mathbb{F}$，随机选 $p, a, b \leftarrow_{\$} \mathbb{F}$，定义单输出功能 $g((p, a, b, x), y) = (\alpha, \beta, f_2(x, y))$，其中

$$\alpha = p + f_1(x, y), \quad \beta = a \cdot \alpha + b.$$

P_2 将 (α, β) 发送给 P_1，P_1 验证 $\beta \stackrel{?}{=} a \cdot \alpha + b$，通过则输出 $\alpha - p$.

输出真实性 (Output Authenticity)

- 在恶意模型下，P_2 可能向 P_1 发送任意结果.
- **公平性**无法实现，但可以保证：若 P_2 发送输出，P_1 可验证其真实性.

基于 MAC 的构造

设功能 $f = (f_1, f_2)$，只允许 P_2 先获得输出. 取域 $\mathbb{F}$，随机选 $p, a, b \leftarrow_{\$} \mathbb{F}$，定义单输出功能 $g((p, a, b, x), y) = (\alpha, \beta, f_2(x, y))$，其中

$$\alpha = p + f_1(x, y), \quad \beta = a \cdot \alpha + b.$$

P_2 将 (α, β) 发送给 P_1，P_1 验证 $\beta \stackrel{?}{=} a \cdot \alpha + b$，通过则输出 $\alpha - p$.

安全性：一次一密隐藏 f_1，伪造 MAC 的成功率为 $1/|\mathbb{F}|$.

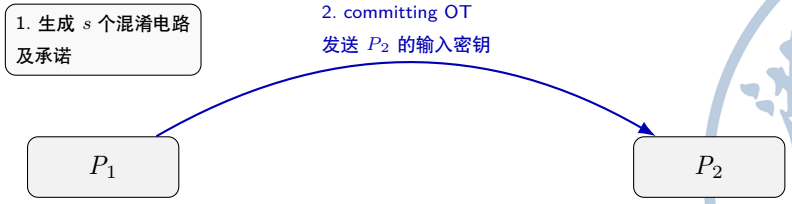
切分选择协议流程

1. 生成 s 个混淆电路及承诺

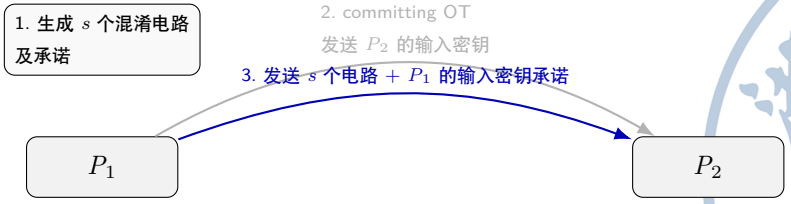
P_1

P_2

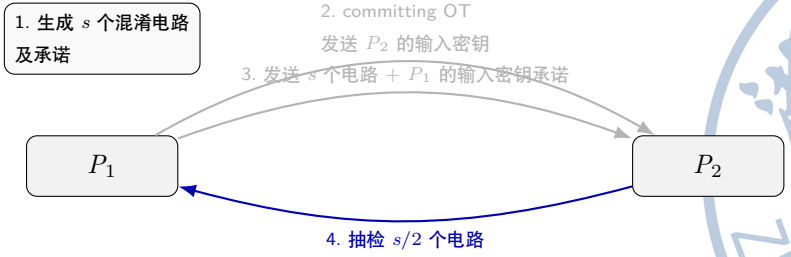
切分选择协议流程



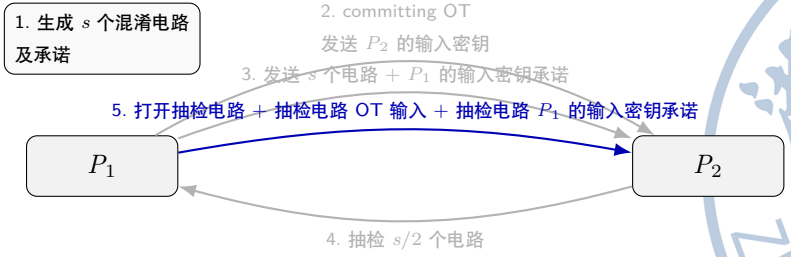
切分选择协议流程


1. 生成 s 个混淆电路及承诺
2. committing OT
发送 P_2 的输入密钥
3. 发送 s 个电路 + P_1 的输入密钥承诺
P_1
P_2

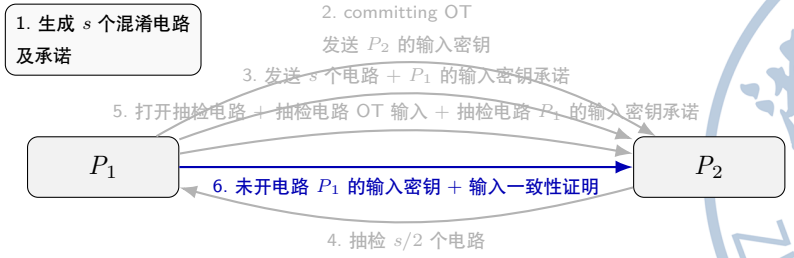
切分选择协议流程


1. 生成 s 个混淆电路及承诺
2. committing OT
发送 P_2 的输入密钥
3. 发送 s 个电路 + P_1 的输入密钥承诺
P_1
P_2
4. 抽检 $s/2$ 个电路

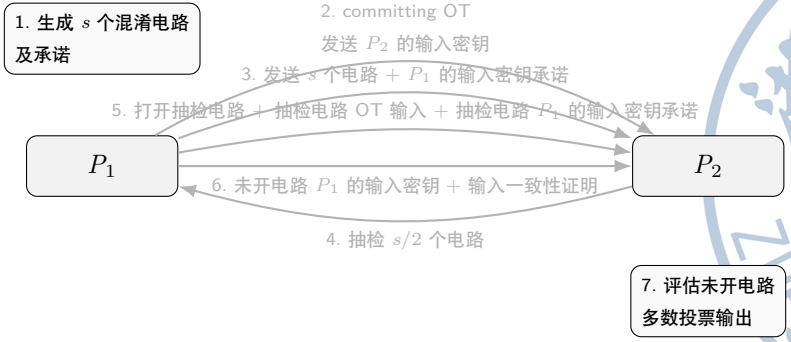
切分选择协议流程


1. 生成 s 个混淆电路及承诺
2. committing OT
发送 P_2 的输入密钥
3. 发送 s 个电路 + P_1 的输入密钥承诺
5. 打开抽检电路 + 抽检电路 OT 输入 + 抽检电路 P_1 的输入密钥承诺
P_1
P_2
4. 抽检 $s/2$ 个电路

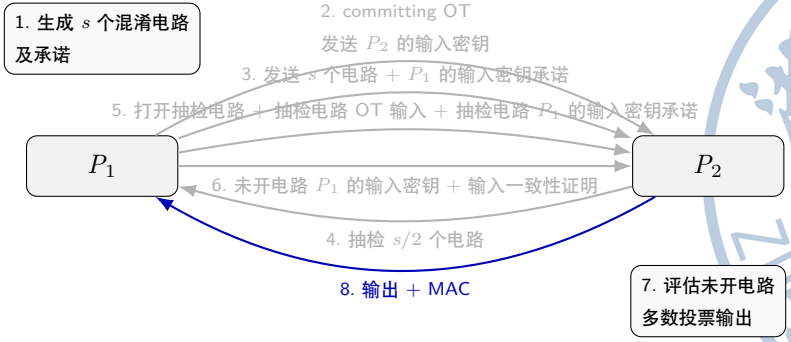
切分选择协议流程


1. 生成 s 个混淆电路及承诺
2. committing OT
发送 P_2 的输入密钥
3. 发送 s 个电路 + P_1 的输入密钥承诺
5. 打开抽检电路 + 抽检电路 OT 输入 + 抽检电路 P_1 的输入密钥承诺
P_1
P_2
6. 未开电路 P_1 的输入密钥 + 输入一致性证明
4. 抽检 $s/2$ 个电路

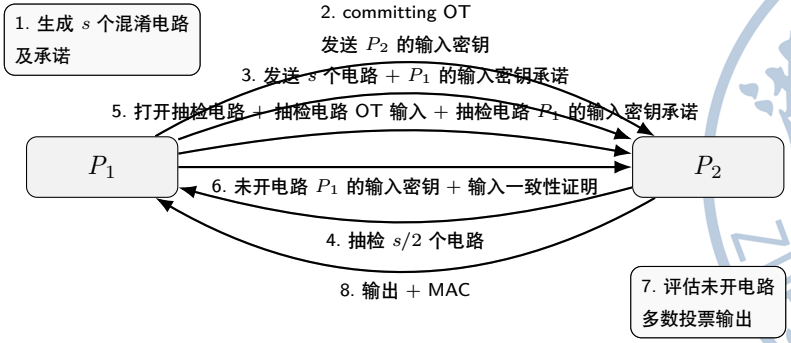
切分选择协议流程


1. 生成 s 个混淆电路及承诺
2. committing OT
发送 P_2 的输入密钥
3. 发送 s 个电路 + P_1 的输入密钥承诺
5. 打开抽检电路 + 抽检电路 OT 输入 + 抽检电路 P_1 的输入密钥承诺
P_1
P_2
6. 未开电路 P_1 的输入密钥 + 输入一致性证明
4. 抽检 $s/2$ 个电路
7. 评估未开电路多数投票输出

切分选择协议流程


1. 生成 s 个混淆电路及承诺
2. committing OT
发送 P_2 的输入密钥
3. 发送 s 个电路 + P_1 的输入密钥承诺
5. 打开抽检电路 + 抽检电路 OT 输入 + 抽检电路 P_1 的输入密钥承诺
P_1
P_2
6. 未开电路 P_1 的输入密钥 + 输入一致性证明
4. 抽检 $s/2$ 个电路
8. 输出 + MAC
7. 评估未开电路多数投票输出

切分选择协议流程


1. 生成 s 个混淆电路及承诺
2. committing OT
发送 P_2 的输入密钥
3. 发送 s 个电路 + P_1 的输入密钥承诺
5. 打开抽检电路 + 抽检电路 OT 输入 + 抽检电路 P_1 的输入密钥承诺
P_1
P_2
6. 未开电路 P_1 的输入密钥 + 输入一致性证明
4. 抽检 $s/2$ 个电路
8. 输出 + MAC
7. 评估未开电路多数投票输出

LP11 协议

Lindell 和 Pinkas 在 2011 年提出的切分选择协议进行了以下改进：

LP11 协议

Lindell 和 Pinkas 在 2011 年提出的切分选择协议进行了以下改进：

1. **以 CCOT 代替 committing OT**：
   - 接收方输入选择比特 $\sigma_1, \ldots, \sigma_s$ 与集合 $\mathcal{J} \subset [s], |\mathcal{J}| = s/2$；对 $i \in [s]$ 得到 $x_i^{\sigma_i}$，对 $j \in \mathcal{J}$ 得到 (x_j^0, x_j^1).
   - $\mathcal{J}$ 即要打开的电路索引，CCOT 将 OT 与切分选择检查合并，有效抵御选择性中止攻击.

LP11 协议

Lindell 和 Pinkas 在 2011 年提出的切分选择协议进行了以下改进：

1. **以 CCOT 代替 committing OT**：
   - 接收方输入选择比特 $\sigma_1, \ldots, \sigma_s$ 与集合 $\mathcal{J} \subset [s], |\mathcal{J}| = s/2$；对 $i \in [s]$ 得到 $x_i^{\sigma_i}$，对 $j \in \mathcal{J}$ 得到 (x_j^0, x_j^1).
   - $\mathcal{J}$ 即要打开的电路索引，CCOT 将 OT 与切分选择检查合并，有效抵御选择性中止攻击.
2. **高效输入一致性证明**：
   - 设输入长度为 ℓ，选取 $\{g^{a_i^0}, g^{a_i^1}\}_{i=1}^{\ell}$ 与 $\{g^{r_j}\}_{j=1}^{s}$，设置第 j 个电路第 i 位密钥为 $g^{a_i^0 r_j}, g^{a_i^1 r_j}$.
   - 在 DDH 假设下，已知 $\{g^{a_i^0}, g^{a_i^1}, g^{r_j}\}$ 与实际输入对应密钥时，其余密钥仍伪随机，从而使一致性 ZK 证明更高效.

切分选择 OT

切分选择 OT 理想功能 $\mathcal{F}_{\mathsf{CCOT}}$

- **输入**：
  - 发送方 P_s 的输入是 s 对消息 $\bar{x} = \{(x_0^i, x_1^i)\}_{i=1}^s$.
  - 接收方 P_r 的输入是 s 个选择比特 $\sigma_1, \ldots, \sigma_s \in \{0,1\}$ 和大小为 $s/2$ 的集合 $\mathcal{J} \in [s]$.
- **输出**：如果 $\mathcal{J}$ 的大小不是 $s/2$，P_s 和 P_r 的输出为 $\perp$. 否则，
  - 对于 $j \in \mathcal{J}$，接收方 P_r 获得 (x_0^j, x_1^j).
  - 对于 $j \notin \mathcal{J}$，接收方 P_r 获得 $x_{\sigma_j}^j$.

切分选择 OT

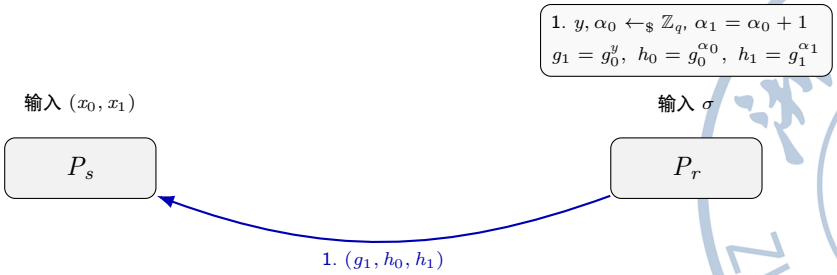
切分选择 OT 理想功能 $\mathcal{F}_{\mathsf{CCOT}}$

- **输入**：
  - 发送方 P_s 的输入是 s 对消息 $\bar{x} = \{(x_0^i, x_1^i)\}_{i=1}^s$.
  - 接收方 P_r 的输入是 s 个选择比特 $\sigma_1, \ldots, \sigma_s \in \{0,1\}$ 和大小为 $s/2$ 的集合 $\mathcal{J} \in [s]$.
- **输出**：如果 $\mathcal{J}$ 的大小不是 $s/2$，P_s 和 P_r 的输出为 $\perp$. 否则，
  - 对于 $j \in \mathcal{J}$，接收方 P_r 获得 (x_0^j, x_1^j).
  - 对于 $j \notin \mathcal{J}$，接收方 P_r 获得 $x_{\sigma_j}^j$.

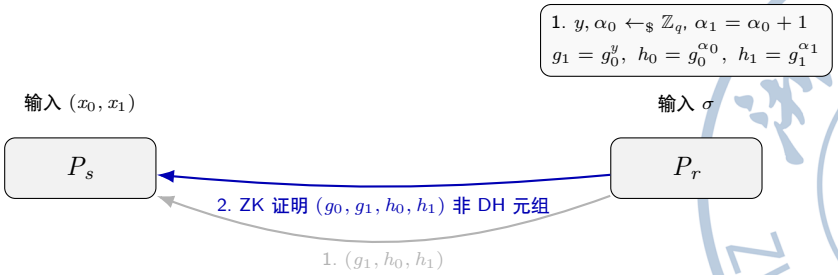
单一选择切分选择 OT 理想功能 $\mathcal{F}_{\mathsf{CCOT}}^S$

- **输入**：与 $\mathcal{F}_{\mathsf{CCOT}}$ 相同，但 P_r 只有一个选择比特 σ.
- **输出**：与 $\mathcal{F}_{\mathsf{CCOT}}$ 相同，但对于 $j \notin \mathcal{J}$，P_r 获得 x_σ^j.

PVW 协议

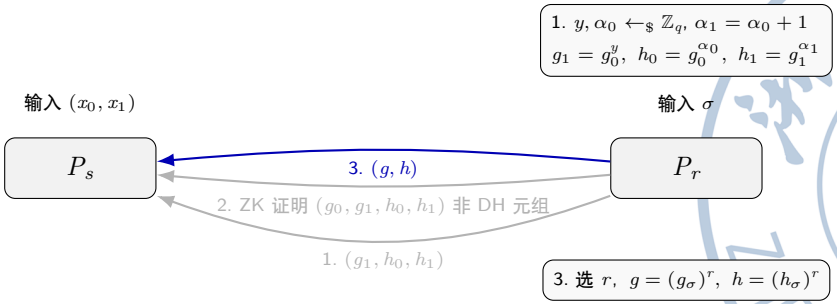

1. $y, \alpha_0 \leftarrow_{\$} \mathbb{Z}_q,\ \alpha_1 = \alpha_0 + 1$
$g_1 = g_0^y,\ h_0 = g_0^{\alpha_0},\ h_1 = g_1^{\alpha_1}$
输入 (x_0, x_1)
输入 σ
P_s
P_r
1. (g_1, h_0, h_1)

PVW 协议



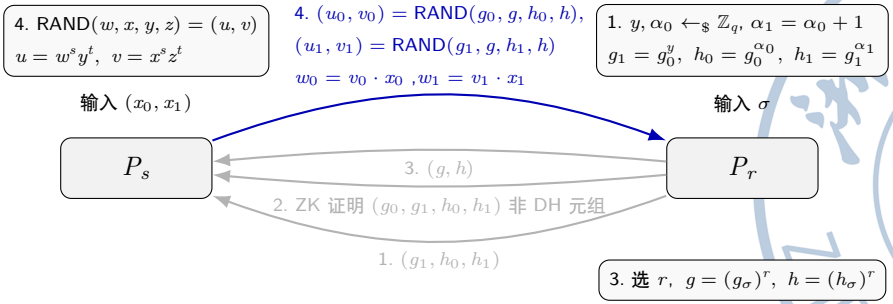
注：P_r 实际上证明 $(g_0, g_1, h_0, \frac{h_1}{g_1})$ 是 DH 元组.

PVW 协议



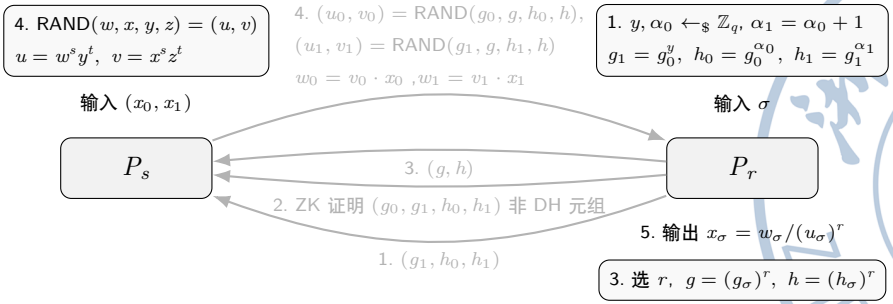
注：P_r 实际上证明 $(g_0, g_1, h_0, \frac{h_1}{g_1})$ 是 DH 元组.

PVW 协议



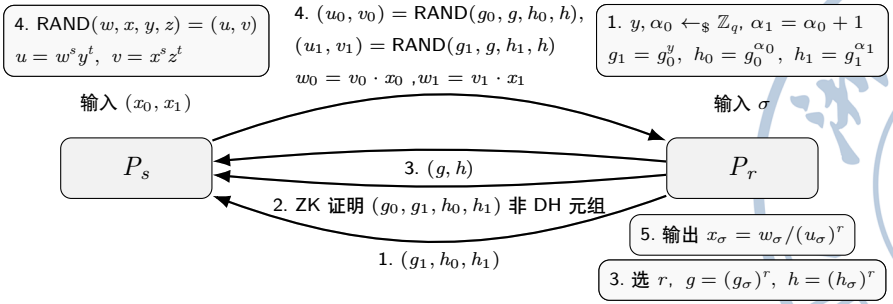
注: P_r 实际上证明 $(g_0,g_1,h_0,\frac{h_1}{g_1})$ 是 DH 元组.

PVW 协议



注：P_r 实际上证明 $(g_0,g_1,h_0,\frac{h_1}{g_1})$ 是 DH 元组.

PVW 协议



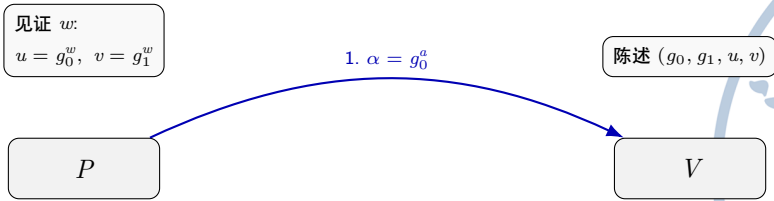
注：P_r 实际上证明 $(g_0,g_1,h_0,\frac{h_1}{g_1})$ 是 DH 元组.

PVW 协议的正确性

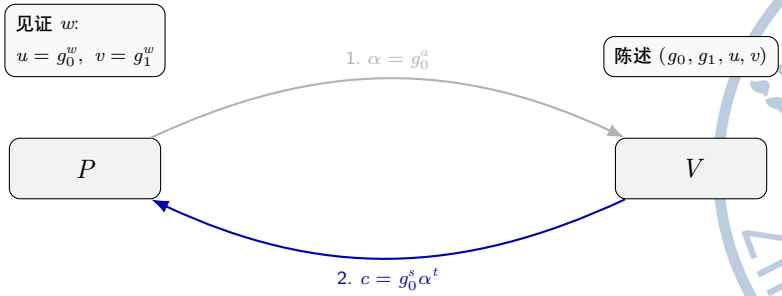
正确性:

$$\frac{w_\sigma}{(u_\sigma)^r} = \frac{v_\sigma \cdot x_\sigma}{(u_\sigma)^r} = \frac{g^s \cdot h^t \cdot x_\sigma}{((g_\sigma)^s \cdot (h_\sigma)^t)^r} = \frac{g^s \cdot h^t \cdot x_\sigma}{g^s \cdot h^t} = x_\sigma$$

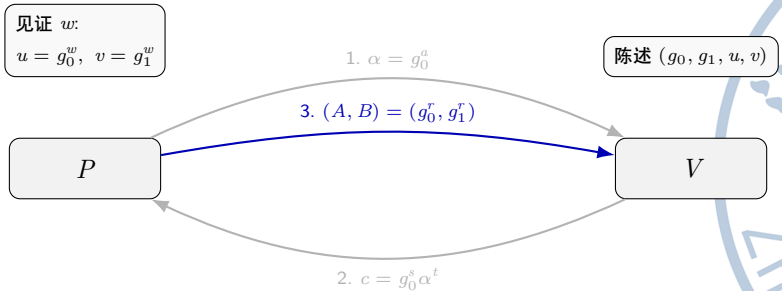
DH 元组的零知识证明协议



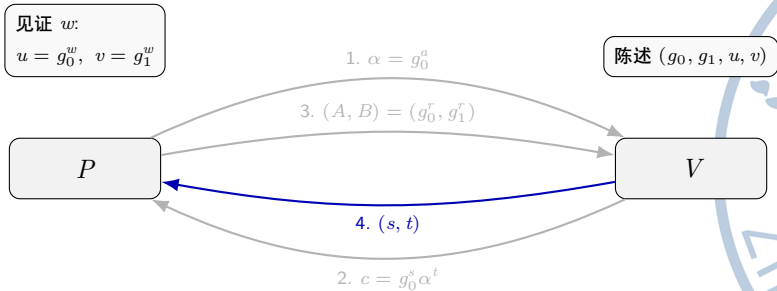
DH 元组的零知识证明协议



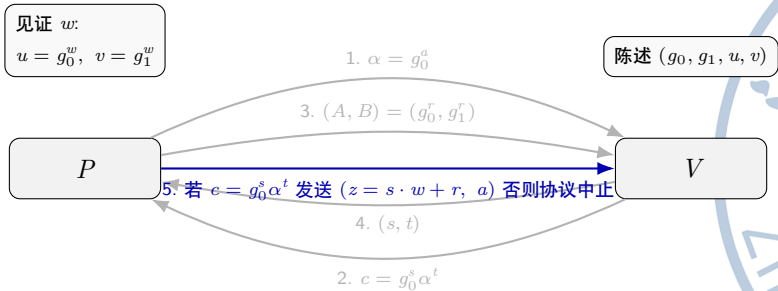
DH 元组的零知识证明协议



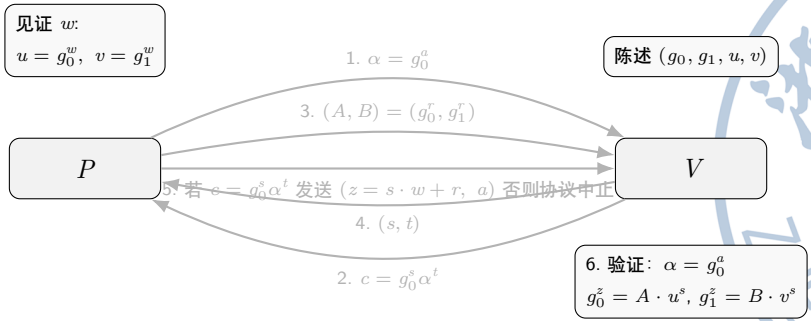
DH 元组的零知识证明协议



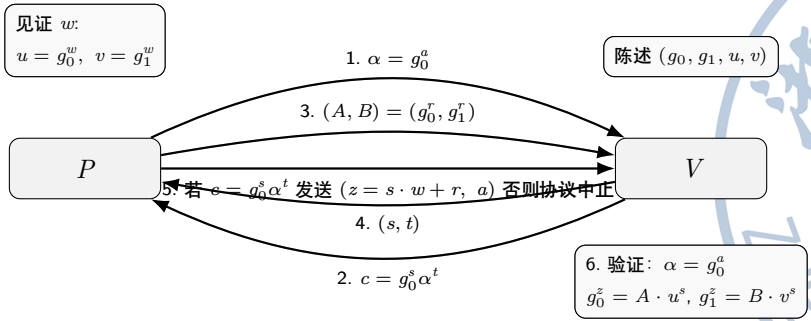
DH 元组的零知识证明协议



DH 元组的零知识证明协议



DH 元组的零知识证明协议



PVW 协议的安全性

安全性:（基于 RAND 的性质）

PVW 协议的安全性

安全性:（基于 RAND 的性质）

- 若 (g_0, g_1, h_0, h_1) 不是 DH 元组：
  - 则 RAND 的输出为**独立均匀**的群元素.
  - 因而 (u_0, w_0) 与 (u_1, w_1) 中**至少一个**是均匀随机分布，接收方至多得到一个输入.

PVW 协议的安全性

安全性:（基于 RAND 的性质）

- 若 (g_0, g_1, h_0, h_1) 不是 DH 元组：
  - 则 RAND 的输出为**独立均匀**的群元素.
  - 因而 (u_0, w_0) 与 (u_1, w_1) 中**至少一个**是均匀随机分布，接收方至多得到一个输入.
- 若 (g_0, g_1, h_0, h_1) 为 DH 元组且接收方知道 $y = \log_{g_0} g_1$：
  - 对 $\sigma = 0$，接收方已得 x_0，并可计算

$$x_1 = \frac{w_1}{(u_1)^{ry^{-1}}} = \frac{g^s h^t x_1}{((g_1)^s (h_1)^t)^{ry^{-1}}} = \frac{g^s h^t x_1}{((g_0)^s (h_0)^t)^r} = x_1.$$

  - 同理 $\sigma = 1$ 时，可由 $w_0/(u_0)^{ry}$ 得到 x_0.

PVW 协议的安全性

安全性:（基于 RAND 的性质）

- 若 (g_0, g_1, h_0, h_1) 不是 DH 元组：
  - 则 RAND 的输出为**独立均匀**的群元素.
  - 因而 (u_0, w_0) 与 (u_1, w_1) 中**至少一个**是均匀随机分布，接收方至多得到一个输入.
- 若 (g_0, g_1, h_0, h_1) 为 DH 元组且接收方知道 $y = \log_{g_0} g_1$：
  - 对 $\sigma = 0$，接收方已得 x_0，并可计算

$$x_1 = \frac{w_1}{(u_1)^{ry^{-1}}} = \frac{g^s h^t x_1}{((g_1)^s (h_1)^t)^{ry^{-1}}} = \frac{g^s h^t x_1}{((g_0)^s (h_0)^t)^r} = x_1.$$

  - 同理 $\sigma = 1$ 时，可由 $w_0/(u_0)^{ry}$ 得到 x_0.

因此协议要求 P_r 证明 $(g_0, g_1, h_0, \frac{h_1}{g_1})$ 为 DH 元组，从而保证其**只能获得一个输入**.

模拟器的构造

发送方 P_s 被攻陷

模拟器 $\mathcal{S}$ 令 $\alpha_1 = \alpha_0$（使元组为 DH），**并伪造**零知识证明，从而提取 P_s 的两个输入.

模拟器的构造

发送方 P_s 被攻陷

模拟器 $\mathcal{S}$ 令 $\alpha_1 = \alpha_0$（使元组为 DH），**并伪造**零知识证明，从而提取 P_s 的两个输入.

接收方 P_r 被攻陷

模拟器 $\mathcal{S}$ 通过零知识证明**提取见证** α_0，计算 $\alpha_1 = \alpha_0 + 1$. 比较 $h = g^{\alpha_0}$ 或 $h = g^{\alpha_1}$，即可恢复选择比特 σ.

模拟器的构造

发送方 P_s 被攻陷

模拟器 $\mathcal{S}$ 令 $\alpha_1 = \alpha_0$（使元组为 DH），**并伪造**零知识证明，从而提取 P_s 的两个输入.

接收方 P_r 被攻陷

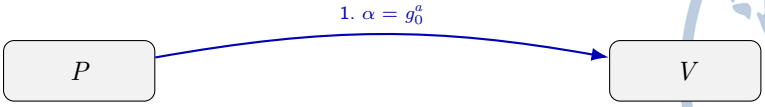
模拟器 $\mathcal{S}$ 通过零知识证明**提取见证** α_0，计算 $\alpha_1 = \alpha_0 + 1$. 比较 $h = g^{\alpha_0}$ 或 $h = g^{\alpha_1}$，即可恢复选择比特 σ.

备注：伪造证明与见证提取依赖 Sigma 协议的可模拟性与 2-特殊可靠性.

子集是 DH 元组的零知识证明协议

见证：$I = \{i_j\}, W = \{(i_j, w_{i_j})\}$
$|I| = s/2,\ h_0^i = g_0^{w_i},\ h_1^i = g_1^{w_i}$

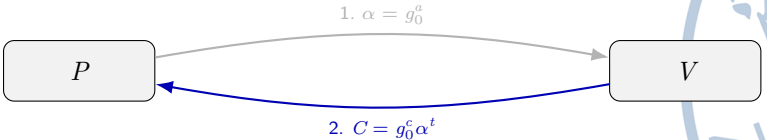
陈述：$(g_0, g_1, (h_0^i, h_1^i)_{i\in[s]})$



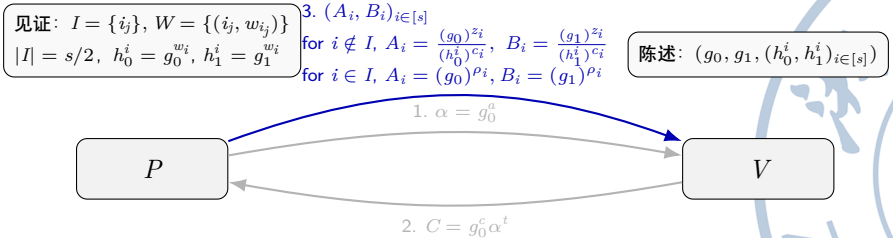
子集是 DH 元组的零知识证明协议

见证：$I = \{i_j\}, W = \{(i_j, w_{i_j})\}$
$|I| = s/2,\ h_0^i = g_0^{w_i},\ h_1^i = g_1^{w_i}$

陈述：$(g_0, g_1, (h_0^i, h_1^i)_{i\in[s]})$



子集是 DH 元组的零知识证明协议



子集是 DH 元组的零知识证明协议

见证：$I = \{i_j\}, W = \{(i_j, w_{i_j})\}$
$|I| = s/2,\ h_0^i = g_0^{w_i},\ h_1^i = g_1^{w_i}$

3. $(A_i, B_i)_{i\in[s]}$
for $i \notin I$, $A_i = \frac{(g_0)^{z_i}}{(h_0^i)^{c_i}},\ B_i = \frac{(g_1)^{z_i}}{(h_1^i)^{c_i}}$
for $i \in I$, $A_i = (g_0)^{\rho_i}, B_i = (g_1)^{\rho_i}$

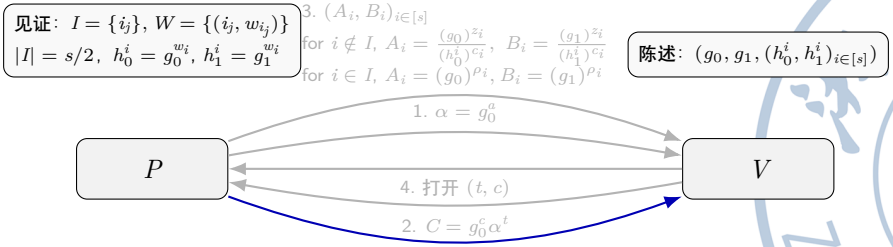
陈述：$(g_0, g_1, (h_0^i, h_1^i)_{i\in[s]})$

1. $\alpha = g_0^a$

P ⟵ 4. 打开 (t, c) ⟵ V

2. $C = g_0^c \alpha^t$

子集是 DH 元组的零知识证明协议



子集是 DH 元组的零知识证明协议

见证：$I=\{i_j\}, W=\{(i_j, w_{i_j})\}$
$|I|=s/2,\ h_0^i=g_0^{w_i},\ h_1^i=g_1^{w_i}$

陈述：$(g_0, g_1, (h_0^i, h_1^i)_{i\in[s]})$

3. $(A_i, B_i)_{i\in[s]}$
for $i\notin I$, $A_i=\frac{(g_0)^{z_i}}{(h_0^i)^{c_i}},\ B_i=\frac{(g_1)^{z_i}}{(h_1^i)^{c_i}}$
for $i\in I$, $A_i=(g_0)^{\rho_i}, B_i=(g_1)^{\rho_i}$

1. $\alpha=g_0^a$

P

V

4. 打开 (t, c)

2. $C=g_0^c\alpha^t$

5. 验证承诺 C，发送 $(c_i, z_i)_{i\in[s]}$ 和 a
其中 c_i 为 c 的 $(s/2, s)$-门限秘密份额
对于 $i\in I$，$z_i=c_i\cdot w_i+\rho_i$

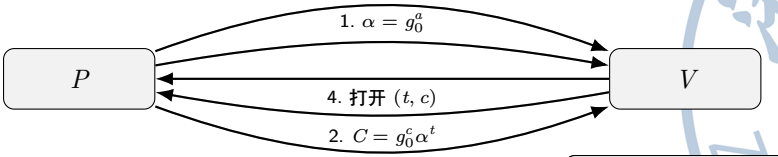
6. 验证 $\alpha=g_0^a$；c_i 合法
$A_i=\frac{g_0^{z_i}}{(h_0^i)^{c_i}},\ B_i=\frac{g_1^{z_i}}{(h_1^i)^{c_i}}$

子集是 DH 元组的零知识证明协议

见证：$I = \{i_j\}, W = \{(i_j, w_{i_j})\}$
$|I| = s/2,\ h_0^i = g_0^{w_i},\ h_1^i = g_1^{w_i}$

3. $(A_i, B_i)_{i\in[s]}$
for $i \notin I$, $A_i = \frac{(g_0)^{z_i}}{(h_0^i)^{c_i}},\ B_i = \frac{(g_1)^{z_i}}{(h_1^i)^{c_i}}$
for $i \in I$, $A_i = (g_0)^{\rho_i}, B_i = (g_1)^{\rho_i}$

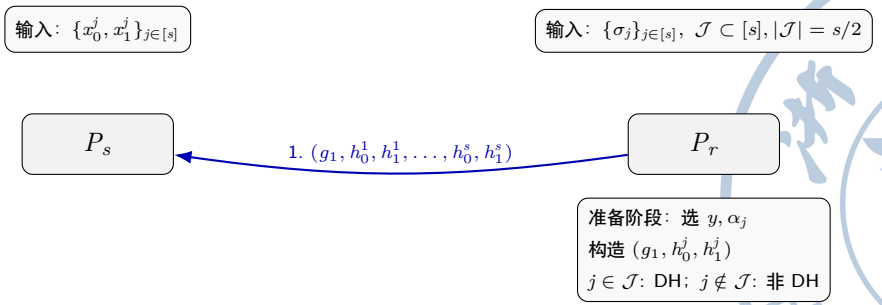
陈述：$(g_0, g_1, (h_0^i, h_1^i)_{i\in[s]})$



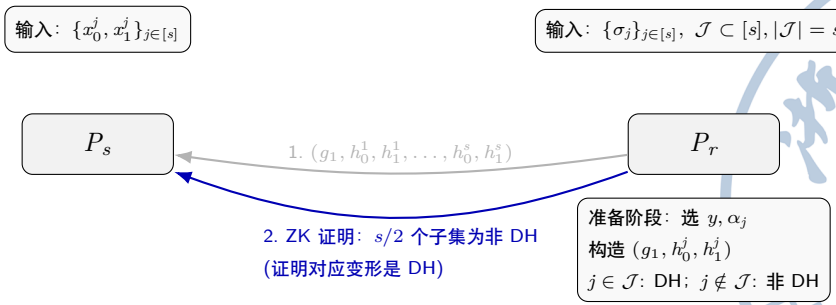
5. 验证承诺 C，发送 $(c_i, z_i)_{i\in[s]}$ 和 a
其中 c_i 为 c 的 $(s/2, s)$-门限秘密份额
对于 $i \in I$，$z_i = c_i \cdot w_i + \rho_i$

6. 验证 $\alpha = g_0^a$；c_i 合法
$A_i = \frac{g_0^{z_i}}{(h_0^i)^{c_i}},\ B_i = \frac{g_1^{z_i}}{(h_1^i)^{c_i}}$

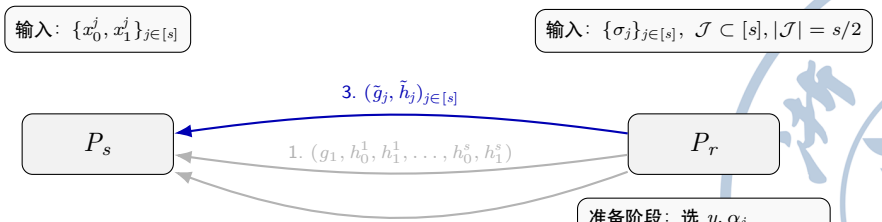
切分选择 OT 协议


输入：$\{x_0^j, x_1^j\}_{j\in[s]}$
输入：$\{\sigma_j\}_{j\in[s]},\ \mathcal{J}\subset[s], |\mathcal{J}| = s/2$
P_s
1. $(g_1, h_0^1, h_1^1, \ldots, h_0^s, h_1^s)$
P_r
准备阶段：选 y, α_j
构造 (g_1, h_0^j, h_1^j)
$j \in \mathcal{J}$：DH；$j \notin \mathcal{J}$：非 DH

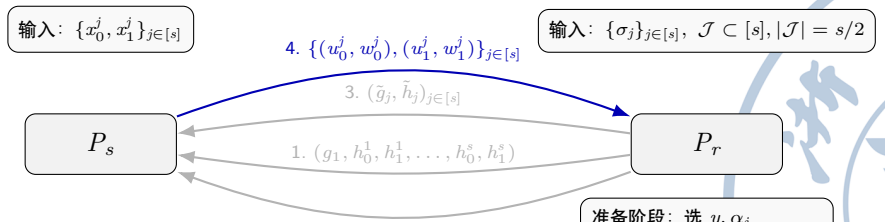
切分选择 OT 协议


输入：$\{x_0^j, x_1^j\}_{j\in[s]}$
输入：$\{\sigma_j\}_{j\in[s]},\ \mathcal{J}\subset[s], |\mathcal{J}| = s/2$
P_s
P_r
1. $(g_1, h_0^1, h_1^1, \ldots, h_0^s, h_1^s)$
2. ZK 证明：$s/2$ 个子集为非 DH
(证明对应变形是 DH)
准备阶段：选 y, α_j
构造 (g_1, h_0^j, h_1^j)
$j\in\mathcal{J}$：DH；$j\notin\mathcal{J}$：非 DH

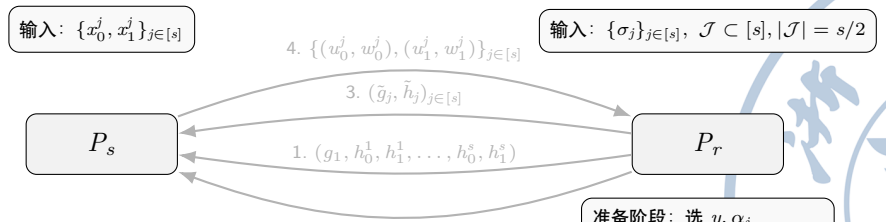
切分选择 OT 协议


输入：$\{x_0^j, x_1^j\}_{j\in[s]}$
输入：$\{\sigma_j\}_{j\in[s]}$，$\mathcal{J}\subset[s], |\mathcal{J}|=s/2$
3. $(\tilde{g}_j, \tilde{h}_j)_{j\in[s]}$
P_s
1. $(g_1, h_0^1, h_1^1, \ldots, h_0^s, h_1^s)$
P_r
2. ZK 证明：$s/2$ 个子集为非 DH
(证明对应变形是 DH)
准备阶段：选 y, α_j
构造 (g_1, h_0^j, h_1^j)
$j\in\mathcal{J}$：DH；$j\notin\mathcal{J}$：非 DH
传输阶段（并行 $j\in[s]$）：
选 r_j，$\tilde{g}_j=(g_{\sigma_j})^{r_j}$，$\tilde{h}_j=(h_{\sigma_j}^j)^{r_j}$

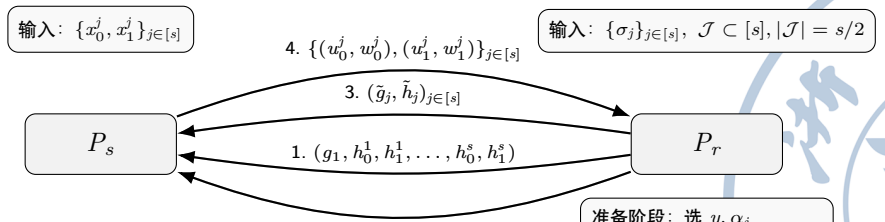
切分选择 OT 协议


输入：$\{x_0^j, x_1^j\}_{j\in[s]}$
输入：$\{\sigma_j\}_{j\in[s]}$，$\mathcal{J}\subset[s], |\mathcal{J}|=s/2$
4. $\{(u_0^j, w_0^j), (u_1^j, w_1^j)\}_{j\in[s]}$
3. $(\tilde{g}_j, \tilde{h}_j)_{j\in[s]}$
1. $(g_1, h_0^1, h_1^1, \ldots, h_0^s, h_1^s)$
P_s
P_r
2. ZK 证明：$s/2$ 个子集为非 DH
(证明对应变形是 DH)
传输阶段：对每个 j 计算
$(u_b^j, v_b^j) = \mathsf{RAND}(g_b, \tilde{g}_j, h_b^j, \tilde{h}_j)$
$w_b^j = v_b^j \cdot x_b^j,\ b\in\{0,1\}$
准备阶段：选 y, α_j
构造 (g_1, h_0^j, h_1^j)
$j\in\mathcal{J}$：DH；$j\notin\mathcal{J}$：非 DH
传输阶段（并行 $j\in[s]$）：
选 r_j，$\tilde{g}_j = (g_{\sigma_j})^{r_j}$，$\tilde{h}_j = (h_{\sigma_j}^j)^{r_j}$

切分选择 OT 协议


输入：$\{x_0^j, x_1^j\}_{j\in[s]}$
输入：$\{\sigma_j\}_{j\in[s]}$，$\mathcal{J}\subset[s], |\mathcal{J}|=s/2$
4. $\{(w_0^j, w_0^j), (w_1^j, w_1^j)\}_{j\in[s]}$
3. $(\tilde{g}_j, \tilde{h}_j)_{j\in[s]}$
P_s
P_r
1. $(g_1, h_0^1, h_1^1, \ldots, h_0^s, h_1^s)$
2. ZK 证明：$s/2$ 个子集为非 DH
传输阶段：对每个 j 计算
$(w_b^j, v_b^j) = \mathrm{RAND}(g_b, \tilde{g}_j, h_b^j, \tilde{h}_j)$
$w_b^j = v_b^j \cdot x_b^j,\ b\in\{0,1\}$
输出阶段：$\forall j,\ x_{\sigma_j}^j = \frac{w_{\sigma_j}^j}{(w_{\sigma_j}^j)^{r_j}}$
$\forall j\in\mathcal{J},\ x_{1-\sigma_j}^j = \frac{w_{1-\sigma_j}^j}{(w_{1-\sigma_j}^j)^{r_j z_j}}$
准备阶段：选 y, α_j
构造 (g_1, h_0^j, h_1^j)
$j\in\mathcal{J}$：DH；$j\notin\mathcal{J}$：非 DH
传输阶段（并行 $j\in[s]$）：
选 $r_j,\ \tilde{g}_j = (g_{\sigma_j})^{r_j},\ \tilde{h}_j = (h_{\sigma_j}^j)^{r_j}$

切分选择 OT 协议


输入：$\{x_0^j, x_1^j\}_{j\in[s]}$
输入：$\{\sigma_j\}_{j\in[s]}$，$\mathcal{J}\subset[s], |\mathcal{J}|=s/2$
4. $\{(w_0^j, w_0^j), (w_1^j, w_1^j)\}_{j\in[s]}$
3. $(\tilde{g}_j, \tilde{h}_j)_{j\in[s]}$
P_s
1. $(g_1, h_0^1, h_1^1, \ldots, h_0^s, h_1^s)$
P_r
2. ZK 证明：$s/2$ 个子集为非 DH
传输阶段：对每个 j 计算
$(w_b^j, v_b^j) = \mathrm{RAND}(g_b, \tilde{g}_j, h_b^j, \tilde{h}_j)$
$w_b^j = v_b^j \cdot x_b^j,\ b\in\{0,1\}$
输出阶段：$\forall j,\ x_{\sigma_j}^j = \frac{w_{\sigma_j}^j}{(w_{\sigma_j}^j)^{r_j}}$
$\forall j\in\mathcal{J},\ x_{1-\sigma_j}^j = \frac{w_{1-\sigma_j}^j}{(w_{1-\sigma_j}^j)^{r_j z_j}}$
准备阶段：选 y, α_j
构造 (g_1, h_0^j, h_1^j)
$j\in\mathcal{J}$：DH；$j\notin\mathcal{J}$：非 DH
传输阶段（并行 $j\in[s]$）：
选 $r_j,\ \tilde{g}_j = (g_{\sigma_j})^{r_j},\ \tilde{h}_j = (h_{\sigma_j}^j)^{r_j}$

批处理单一选择切分选择 OT

批处理单一选择切分选择 OT 理想功能 $\mathcal{F}_{\mathsf{CCOT}}^{S,B}$

- **输入**：
  - 发送方 P_s 的输入是 ℓ 个长度为 s 的向量 $\vec{x}_i,\ i \in [\ell]$（每个向量由 s 对消息构成）.
  - 接收方 P_r 的输入是 ℓ 个选择比特 $\sigma_1, \ldots, \sigma_\ell \in \{0,1\}$ 和大小为 $s/2$ 的集合 $\mathcal{J} \in [s]$.
- **输出**：如果 $\mathcal{J}$ 的大小不是 $s/2$，P_s 和 P_r 的输出为 $\perp$. 否则，
  - 对于 $i \in [\ell]$ 和 $j \in \mathcal{J}$，接收方 P_r 获得 $\vec{x}_i$ 的第 j 对消息.
  - 对于 $i \in [\ell]$ 和 $j \notin \mathcal{J}$，接收方 P_r 获得 $\vec{x}_i$ 的第 j 对消息的第 σ_i 个值（$\sigma_i \in \{0,1\}$，这里的第 σ_i 个值从 0 开始计数）.

批处理单一选择切分选择 OT

批处理单一选择切分选择 OT 理想功能 $\mathcal{F}_{\mathsf{CCOT}}^{S,B}$

- **输入**：
  - 发送方 P_s 的输入是 ℓ 个长度为 s 的向量 $\vec{x}_i,\ i \in [\ell]$（每个向量由 s 对消息构成）.
  - 接收方 P_r 的输入是 ℓ 个选择比特 $\sigma_1, \ldots, \sigma_\ell \in \{0,1\}$ 和大小为 $s/2$ 的集合 $\mathcal{J} \in [s]$.
- **输出**：如果 $\mathcal{J}$ 的大小不是 $s/2$，P_s 和 P_r 的输出为 $\perp$. 否则，
  - 对于 $i \in [\ell]$ 和 $j \in \mathcal{J}$，接收方 P_r 获得 $\vec{x}_i$ 的第 j 对消息.
  - 对于 $i \in [\ell]$ 和 $j \notin \mathcal{J}$，接收方 P_r 获得 $\vec{x}_i$ 的第 j 对消息的第 σ_i 个值（$\sigma_i \in \{0,1\}$，这里的第 σ_i 个值从 0 开始计数）.

构造方法：

批处理单一选择切分选择 OT

批处理单一选择切分选择 OT 理想功能 $\mathcal{F}_{\mathsf{CCOT}}^{S,B}$

- **输入**：
  - 发送方 P_s 的输入是 ℓ 个长度为 s 的向量 $\vec{x}_i,\ i \in [\ell]$（每个向量由 s 对消息构成）.
  - 接收方 P_r 的输入是 ℓ 个选择比特 $\sigma_1, \ldots, \sigma_\ell \in \{0,1\}$ 和大小为 $s/2$ 的集合 $\mathcal{J} \in [s]$.
- **输出**：如果 $\mathcal{J}$ 的大小不是 $s/2$，P_s 和 P_r 的输出为 $\perp$. 否则，
  - 对于 $i \in [\ell]$ 和 $j \in \mathcal{J}$，接收方 P_r 获得 $\vec{x}_i$ 的第 j 对消息.
  - 对于 $i \in [\ell]$ 和 $j \notin \mathcal{J}$，接收方 P_r 获得 $\vec{x}_i$ 的第 j 对消息的第 σ_i 个值（$\sigma_i \in \{0,1\}$，这里的第 σ_i 个值从 0 开始计数）.

构造方法：

- 只运行一次切分选择 OT 协议的准备阶段；

批处理单一选择切分选择 OT

批处理单一选择切分选择 OT 理想功能 $\mathcal{F}_{\mathsf{CCOT}}^{S,B}$

- **输入**：
  - 发送方 P_s 的输入是 ℓ 个长度为 s 的向量 $\vec{x}_i,\ i \in [\ell]$（每个向量由 s 对消息构成）.
  - 接收方 P_r 的输入是 ℓ 个选择比特 $\sigma_1, \ldots, \sigma_\ell \in \{0,1\}$ 和大小为 $s/2$ 的集合 $\mathcal{J} \in [s]$.
- **输出**：如果 $\mathcal{J}$ 的大小不是 $s/2$，P_s 和 P_r 的输出为 $\perp$. 否则，
  - 对于 $i \in [\ell]$ 和 $j \in \mathcal{J}$，接收方 P_r 获得 $\vec{x}_i$ 的第 j 对消息.
  - 对于 $i \in [\ell]$ 和 $j \notin \mathcal{J}$，接收方 P_r 获得 $\vec{x}_i$ 的第 j 对消息的第 σ_i 个值（$\sigma_i \in \{0,1\}$，这里的第 σ_i 个值从 0 开始计数）.

构造方法：

- 只运行一次切分选择 OT 协议的准备阶段；
- 对于每个 $i \in [\ell]$，运行**单一选择切分选择 OT 协议**的传输阶段（可以并行地运行）；

批处理单一选择切分选择 OT

批处理单一选择切分选择 OT 理想功能 $\mathcal{F}_{\mathsf{CCOT}}^{S,B}$

- **输入**：
  - 发送方 P_s 的输入是 ℓ 个长度为 s 的向量 $\vec{x}_i,\ i \in [\ell]$（每个向量由 s 对消息构成）.
  - 接收方 P_r 的输入是 ℓ 个选择比特 $\sigma_1, \ldots, \sigma_\ell \in \{0,1\}$ 和大小为 $s/2$ 的集合 $\mathcal{J} \in [s]$.
- **输出**：如果 $\mathcal{J}$ 的大小不是 $s/2$，P_s 和 P_r 的输出为 $\perp$. 否则，
  - 对于 $i \in [\ell]$ 和 $j \in \mathcal{J}$，接收方 P_r 获得 $\vec{x}_i$ 的第 j 对消息.
  - 对于 $i \in [\ell]$ 和 $j \notin \mathcal{J}$，接收方 P_r 获得 $\vec{x}_i$ 的第 j 对消息的第 σ_i 个值（$\sigma_i \in \{0,1\}$，这里的第 σ_i 个值从 0 开始计数）.

构造方法：

- 只运行一次切分选择 OT 协议的准备阶段；
- 对于每个 $i \in [\ell]$，运行**单一选择切分选择 OT 协议**的传输阶段（可以并行地运行）：
  - 切分选择 OT 协议的第一步改为：P_r 随机选择 $r \leftarrow_{\$} \mathbb{Z}_q$ 并计算 $g' = (g_\sigma)^r$. 对于 $j \in [s]$，计算 $h_j = (h_\sigma^j)^r$. P_r 发送 $(g', h_1, \ldots, h_s)$ 给 P_s 并零知识地证明其正确性.

输入一致性零知识证明

陈述：$(\mathbb{G}, g_0, g_1, g', u_1, v_1, \ldots, u_s, v_s, h_1, \ldots, h_s)$，其中 $\mathbb{G}$ 是一个阶为 q 的群，g_0, g_1 是其生成元.

输入一致性零知识证明

陈述：$(\mathbb{G}, g_0, g_1, g', u_1, v_1, \ldots, u_s, v_s, h_1, \ldots, h_s)$，其中 $\mathbb{G}$ 是一个阶为 q 的群，g_0, g_1 是其生成元.

见证：r s.t. “$g' = (g_0)^r$ 且对于 $j \in [s]$ 有 $h_j = (u_j)^r$ ” 或 “$g' = (g_1)^r$ 且对于 $j \in [s]$ 有 $h_j = (v_j)^r$”

输入一致性零知识证明

陈述：$(\mathbb{G}, g_0, g_1, g', u_1, v_1, \ldots, u_s, v_s, h_1, \ldots, h_s)$，其中 $\mathbb{G}$ 是一个阶为 q 的群，g_0, g_1 是其生成元.

见证：r s.t. “$g' = (g_0)^r$ 且对于 $j \in [s]$ 有 $h_j = (u_j)^r$ ” 或 “$g' = (g_1)^r$ 且对于 $j \in [s]$ 有 $h_j = (v_j)^r$”

协议：

输入一致性零知识证明

陈述：$(\mathbb{G}, g_0, g_1, g', u_1, v_1, \ldots, u_s, v_s, h_1, \ldots, h_s)$，其中 $\mathbb{G}$ 是一个阶为 q 的群，g_0, g_1 是其生成元.

见证：r s.t. “$g' = (g_0)^r$ 且对于 $j \in [s]$ 有 $h_j = (u_j)^r$ ” 或 “$g' = (g_1)^r$ 且对于 $j \in [s]$ 有 $h_j = (v_j)^r$”

协议：

1. 验证者 V 随机选择 $\gamma_1, \ldots, \gamma_s \leftarrow_\$ \mathbb{Z}_q$ 并发送给证明者 P.

输入一致性零知识证明

陈述：$(\mathbb{G}, g_0, g_1, g', u_1, v_1, \ldots, u_s, v_s, h_1, \ldots, h_s)$，其中 $\mathbb{G}$ 是一个阶为 q 的群，g_0, g_1 是其生成元.

见证：r s.t. “$g' = (g_0)^r$ 且对于 $j \in [s]$ 有 $h_j = (u_j)^r$ ” 或 “$g' = (g_1)^r$ 且对于 $j \in [s]$ 有 $h_j = (v_j)^r$”

协议：

1. 验证者 V 随机选择 $\gamma_1, \ldots, \gamma_s \leftarrow_\$ \mathbb{Z}_q$ 并发送给证明者 P.
2. P 和 V 本地计算

$$\tilde{u} = \prod_{i=1}^{s} (u_i)^{\gamma_i},\ \tilde{v} = \prod_{i=1}^{s} (v_i)^{\gamma_i},\ \tilde{h} = \prod_{i=1}^{s} (h_i)^{\gamma_i}$$

输入一致性零知识证明

陈述：$(\mathbb{G}, g_0, g_1, g', u_1, v_1, \ldots, u_s, v_s, h_1, \ldots, h_s)$，其中 $\mathbb{G}$ 是一个阶为 q 的群，g_0, g_1 是其生成元.

见证：r s.t. “$g' = (g_0)^r$ 且对于 $j \in [s]$ 有 $h_j = (u_j)^r$ ” 或 “$g' = (g_1)^r$ 且对于 $j \in [s]$ 有 $h_j = (v_j)^r$”

协议：

1. 验证者 V 随机选择 $\gamma_1, \ldots, \gamma_s \leftarrow_\$ \mathbb{Z}_q$ 并发送给证明者 P.
2. P 和 V 本地计算

$$\tilde{u} = \prod_{i=1}^{s} (u_i)^{\gamma_i},\ \tilde{v} = \prod_{i=1}^{s} (v_i)^{\gamma_i},\ \tilde{h} = \prod_{i=1}^{s} (h_i)^{\gamma_i}$$

3. P 向 V 零知识地证明 $(g_0, g', \tilde{u}, \tilde{h})$ 或 $(g_1, g', \tilde{v}, \tilde{h})$ 是 DH 元组.

LP11 协议

输入 $x \in \{0,1\}^{\ell}$

1. 构造 s 个混淆电路 GC_j

设置密钥 $k_{i,j}^{b} = H(g^{a_i^b r_j})$

输入 $y \in \{0,1\}^{\ell}$

选随机检查集 $\mathcal{J} \subset [s]$

P_1

2. 批处理单一选择 CCOT

(P_2 得到检查/计算电路输入密钥)

P_2

LP11 协议

输入 $x \in \{0,1\}^\ell$

1. 构造 s 个混淆电路 GC_j

设置密钥 $k_{i,j}^b = H(g^{a_i^b r_j})$

3. 发送 $GC_1, \ldots, GC_s$

$+$ P_1 的输入导线密钥的承诺

$\{(i, 0, g^{a_i^0}), (i, 1, g^{a_i^1})\}_{i=1}^{\ell}$, $\{(j, g^{r_j})\}_{j=1}^{s}$.

输入 $y \in \{0,1\}^\ell$

选随机检查集 $\mathcal{J} \subset [s]$

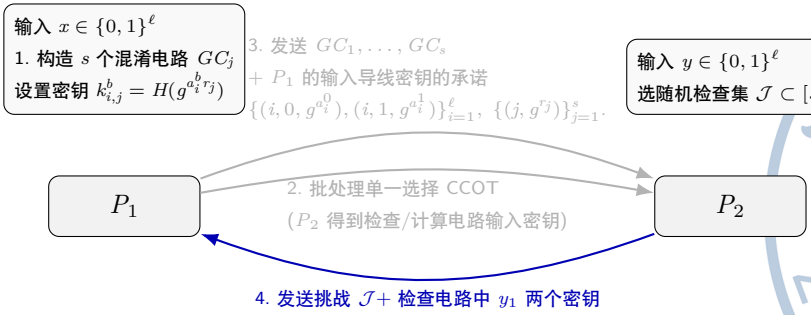
P_1

2. 批处理单一选择 CCOT

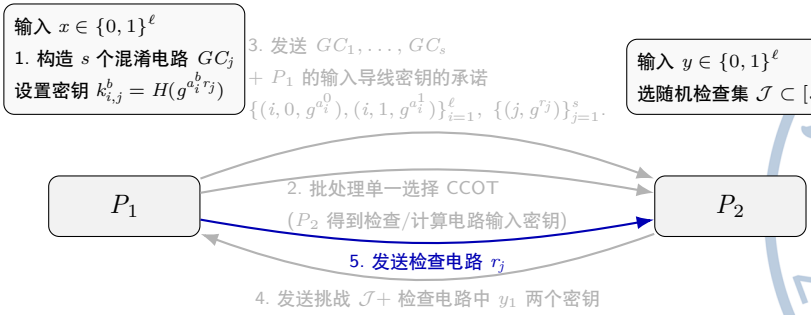
(P_2 得到检查/计算电路输入密钥)

P_2

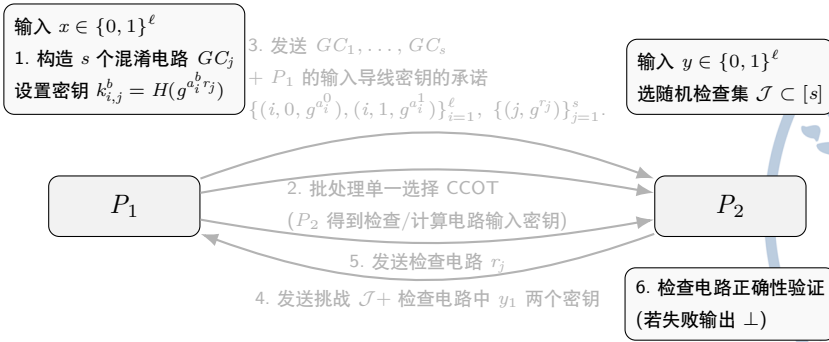
LP11 协议


输入 $x \in \{0,1\}^\ell$
1. 构造 s 个混淆电路 GC_j
设置密钥 $k_{i,j}^b = H(g^{a_i^b r_j})$
3. 发送 $GC_1, \ldots, GC_s$
+ P_1 的输入导线密钥的承诺
$\{(i,0,g^{a_i^0}),(i,1,g^{a_i^1})\}_{i=1}^{\ell},\ \{(j,g^{r_j})\}_{j=1}^{s}$.
输入 $y \in \{0,1\}^\ell$
选随机检查集 $\mathcal{J} \subset [s]$
P_1
2. 批处理单一选择 CCOT
(P_2 得到检查/计算电路输入密钥)
P_2
4. 发送挑战 $\mathcal{J}+$ 检查电路中 y_1 两个密钥

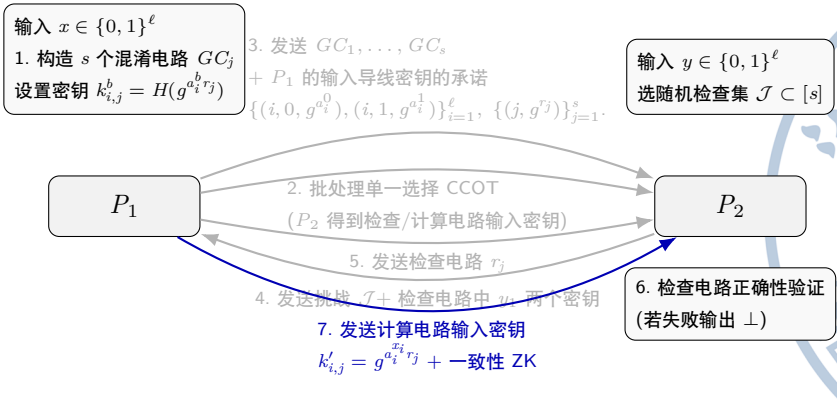
LP11 协议


输入 $x \in \{0,1\}^\ell$
1. 构造 s 个混淆电路 GC_j
设置密钥 $k_{i,j}^b = H(g^{a_i^b r_j})$
3. 发送 $GC_1, \ldots, GC_s$
+ P_1 的输入导线密钥的承诺
$\{(i,0,g^{a_i^0}),(i,1,g^{a_i^1})\}_{i=1}^{\ell},\ \{(j,g^{r_j})\}_{j=1}^{s}$.
输入 $y \in \{0,1\}^\ell$
选随机检查集 $\mathcal{J} \subset [s]$
P_1
P_2
2. 批处理单一选择 CCOT
(P_2 得到检查/计算电路输入密钥)
5. 发送检查电路 r_j
4. 发送挑战 $\mathcal{J}$+ 检查电路中 y_1 两个密钥

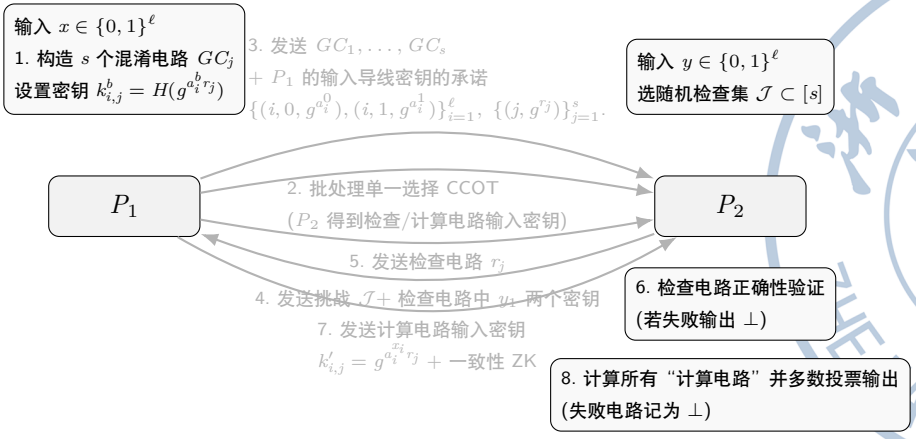
LP11 协议


输入 $x \in \{0,1\}^\ell$
1. 构造 s 个混淆电路 GC_j
设置密钥 $k_{i,j}^b = H(g^{a_i^b r_j})$
3. 发送 $GC_1, \ldots, GC_s$
+ P_1 的输入导线密钥的承诺
$\{(i,0,g^{a_i^0}),(i,1,g^{a_i^1})\}_{i=1}^{\ell},\ \{(j,g^{r_j})\}_{j=1}^{s}$.
输入 $y \in \{0,1\}^\ell$
选随机检查集 $\mathcal{J} \subset [s]$
P_1
2. 批处理单一选择 CCOT
(P_2 得到检查/计算电路输入密钥)
P_2
5. 发送检查电路 r_j
4. 发送挑战 $\mathcal{J}$+ 检查电路中 y_1 两个密钥
6. 检查电路正确性验证
(若失败输出 $\perp$)

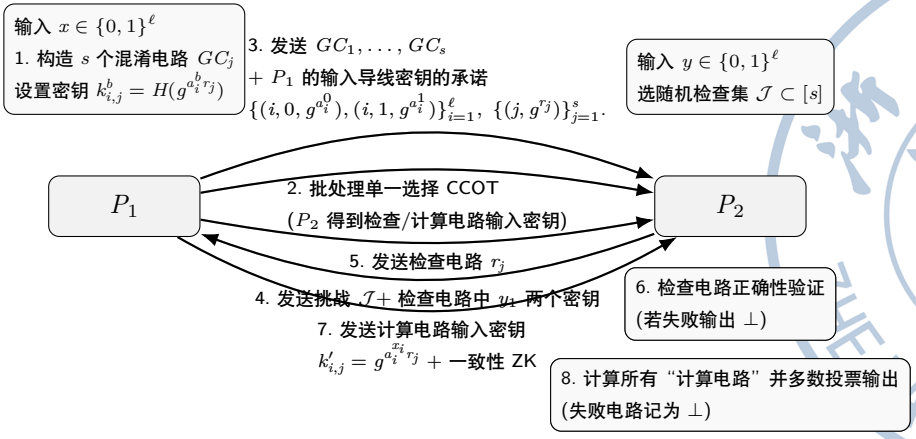
LP11 协议


输入 $x \in \{0,1\}^\ell$
1. 构造 s 个混淆电路 GC_j
设置密钥 $k_{i,j}^b = H(g^{a_i^b r_j})$
3. 发送 $GC_1, \ldots, GC_s$
+ P_1 的输入导线密钥的承诺
$\{(i,0,g^{a_i^0}),(i,1,g^{a_i^1})\}_{i=1}^{\ell}$, $\{(j,g^{r_j})\}_{j=1}^{s}$.
输入 $y \in \{0,1\}^\ell$
选随机检查集 $\mathcal{J} \subset [s]$
P_1
2. 批处理单一选择 CCOT
(P_2 得到检查/计算电路输入密钥)
P_2
5. 发送检查电路 r_j
4. 发送挑战 $\mathcal{J}$+ 检查电路中 y_i 两个密钥
6. 检查电路正确性验证
(若失败输出 $\perp$)
7. 发送计算电路输入密钥
$k'_{i,j} = g^{a_i^{x_i} r_j}$ + 一致性 ZK

LP11 协议


输入 $x \in \{0,1\}^\ell$
1. 构造 s 个混淆电路 GC_j
设置密钥 $k_{i,j}^b = H(g^{a_i^b r_j})$
3. 发送 $GC_1, \ldots, GC_s$
+ P_1 的输入导线密钥的承诺
$\{(i,0,g^{a_i^0}),(i,1,g^{a_i^1})\}_{i=1}^{\ell}$, $\{(j,g^{r_j})\}_{j=1}^{s}$.
输入 $y \in \{0,1\}^\ell$
选随机检查集 $\mathcal{J} \subset [s]$
P_1
2. 批处理单一选择 CCOT
(P_2 得到检查/计算电路输入密钥)
P_2
5. 发送检查电路 r_j
4. 发送挑战 $\mathcal{J}$ + 检查电路中 y_i 两个密钥
7. 发送计算电路输入密钥
$k'_{i,j} = g^{a_i^{x_i} r_j}$ + 一致性 ZK
6. 检查电路正确性验证
(若失败输出 $\perp$)
8. 计算所有“计算电路”并多数投票输出
(失败电路记为 $\perp$)

LP11 协议


输入 $x \in \{0,1\}^\ell$
1. 构造 s 个混淆电路 GC_j
设置密钥 $k_{i,j}^b = H(g^{a_i^b r_j})$
3. 发送 $GC_1, \ldots, GC_s$
+ P_1 的输入导线密钥的承诺
$\{(i,0,g^{a_i^0}),(i,1,g^{a_i^1})\}_{i=1}^{\ell}$, $\{(j,g^{r_j})\}_{j=1}^{s}$.
输入 $y \in \{0,1\}^\ell$
选随机检查集 $\mathcal{J} \subset [s]$
P_1
P_2
2. 批处理单一选择 CCOT
(P_2 得到检查/计算电路输入密钥)
5. 发送检查电路 r_j
4. 发送挑战 $\mathcal{J}$+ 检查电路中 y_i 两个密钥
6. 检查电路正确性验证
(若失败输出 $\perp$)
7. 发送计算电路输入密钥
$k'_{i,j} = g^{a_i^{x_i} r_j}$ + 一致性 ZK
8. 计算所有“计算电路”并多数投票输出
(失败电路记为 $\perp$)

LP11 协议

公共输入：统计安全参数 s，函数 f 等价的布尔电路 $\mathcal{C}$，$(\mathbb{G}, q, g)$，其中 $\mathbb{G}$ 是阶为 q，以 g 为生成元的群．$|q| = n$.

LP11 协议

公共输入：统计安全参数 s，函数 f 等价的布尔电路 $\mathcal{C}$，$(\mathbb{G}, q, g)$，其中 $\mathbb{G}$ 是阶为 q，以 g 为生成元的群．$|q| = n$.

输入：P_1 的输入是 $x \in \{0,1\}^\ell$，P_2 的输入是 $y \in \{0,1\}^\ell$.

LP11 协议

公共输入：统计安全参数 s，函数 f 等价的布尔电路 $\mathcal{C}$，$(\mathbb{G}, q, g)$，其中 $\mathbb{G}$ 是阶为 q，以 g 为生成元的群．$|q| = n$.

输入：P_1 的输入是 $x \in \{0,1\}^\ell$，P_2 的输入是 $y \in \{0,1\}^\ell$.

协议：

- **1. 构造混淆电路**：
  1. P_1 选择随机值 $a_1^0, a_1^1, \ldots, a_\ell^0, a_\ell^1 \leftarrow_\$ \mathbb{Z}_q$ 和 $r_1, \ldots, r_s \leftarrow_\$ \mathbb{Z}_q$.
  2. 设 $w_1, \ldots, w_\ell$ 是 P_1 在电路 $\mathcal{C}$ 中的输入导线．用 $w_{i,j}$ 表示第 j 个混淆电路中的导线 w_i，用 $k_{i,j}^b$ 表示导线 $w_{i,j}$ 对应于比特 b 的密钥．P_1 将这些输入导线的密钥设置为

     $$k_{i,j}^0 = H(g^{a_i^0 \cdot r_j}),\ k_{i,j}^1 = H(g^{a_i^1 \cdot r_j})$$

     其中，H 是一个随机数提取器.
  3. P_1 按照第 8.1.2 节中所述的方式构造 s 个混淆电路，记为 $GC_1, \ldots, GC_s$，其中导线 $w_1, \ldots, w_\ell$ 的密钥按照上述方式确定.

LP11 协议

- **2. 茫然传输**：P_1 和 P_2 以参数 ℓ 和 s 执行批处理单一选择切分选择 OT：
  1. P_1 定义向量 $\vec{z}_1, \ldots, \vec{z}_\ell$，其中 $\vec{z}_i$ 包含了 P_2 的输入比特 y_i 在 $GC_1, \ldots, GC_s$ 中的 s 对密钥.
  2. P_2 的输入是大小为 $s/2$ 的随机集合 $\mathcal{J} \subset [s]$ 和 $\sigma_1, \ldots, \sigma_\ell \in \{0,1\}$，其中 $\sigma_i = y_i$，$i \in [\ell]$.
  3. 对于 GC_j，$j \in \mathcal{J}$，P_2 获得他的输入导线的两个密钥；对于其他电路，P_2 获得他的输入对应的密钥.

LP11 协议

- **3. 发送电路和承诺**：P_1 向 P_2 发送 s 个混淆电路，随机数提取器 H 的定义，以及 P_1 的输入导线密钥的承诺 $\{(i,0,g^{a_i^0}),(i,1,g^{a_i^1})\}_{i=1}^{\ell}$，$\{(j,g^{r_j})\}_{j=1}^{s}$.

LP11 协议

- **3. 发送电路和承诺**：P_1 向 P_2 发送 s 个混淆电路，随机数提取器 H 的定义，以及 P_1 的输入导线密钥的承诺 $\{(i,0,g^{a_i^0}),(i,1,g^{a_i^1})\}_{i=1}^{\ell}$，$\{(j,g^{r_j})\}_{j=1}^{s}$.
- **4. 发送切分选择挑战**：P_2 向 P_1 发送集合 $\mathcal{J}$ 以及电路 $\{GC_j\}_{j\in\mathcal{J}}$ 中 P_2 的第一个输入比特 y_1 对应的两个密钥. 如果密钥不正确，P_1 输出 $\perp$ 并终止协议. $\{GC_j\}_{j\in\mathcal{J}}$ 称为“检查电路”，$\{GC_j\}_{j\notin\mathcal{J}}$ 称为“计算电路”.

LP11 协议

- **3. 发送电路和承诺**：P_1 向 P_2 发送 s 个混淆电路，随机数提取器 H 的定义，以及 P_1 的输入导线密钥的承诺 $\{(i,0,g^{a_i^0}),(i,1,g^{a_i^1})\}_{i=1}^{\ell}$，$\{(j,g^{r_j})\}_{j=1}^{s}$.
- **4. 发送切分选择挑战**：P_2 向 P_1 发送集合 $\mathcal{J}$ 以及电路 $\{GC_j\}_{j\in\mathcal{J}}$ 中 P_2 的第一个输入比特 y_1 对应的两个密钥. 如果密钥不正确，P_1 输出 $\perp$ 并终止协议. $\{GC_j\}_{j\in\mathcal{J}}$ 称为“检查电路”，$\{GC_j\}_{j\notin\mathcal{J}}$ 称为“计算电路”.
- **5. 发送“检查电路”中所有输入密钥**：对于每个“检查电路” GC_j，P_1 向 P_2 发送 r_j. P_2 检查该值与第 3 步中的 (j,g^{r_j}) 是否一致. 如果不一致，P_2 输出 $\perp$ 并终止协议.

LP11 协议

- **3. 发送电路和承诺**：P_1 向 P_2 发送 s 个混淆电路，随机数提取器 H 的定义，以及 P_1 的输入导线密钥的承诺 $\{(i,0,g^{a_i^0}),(i,1,g^{a_i^1})\}_{i=1}^{\ell}$，$\{(j,g^{r_j})\}_{j=1}^{s}$.
- **4. 发送切分选择挑战**：P_2 向 P_1 发送集合 $\mathcal{J}$ 以及电路 $\{GC_j\}_{j\in\mathcal{J}}$ 中 P_2 的第一个输入比特 y_1 对应的两个密钥. 如果密钥不正确，P_1 输出 $\perp$ 并终止协议. $\{GC_j\}_{j\in\mathcal{J}}$ 称为“检查电路”，$\{GC_j\}_{j\notin\mathcal{J}}$ 称为“计算电路”.
- **5. 发送“检查电路”中所有输入密钥**：对于每个“检查电路” GC_j，P_1 向 P_2 发送 r_j. P_2 检查该值与第 3 步中的 (j,g^{r_j}) 是否一致. 如果不一致，P_2 输出 $\perp$ 并终止协议.
- **6. “检查电路”正确性检验**：
  1. 对于 $j\in\mathcal{J}$，P_2 使用第 3 步得到的 $g^{a_i^0}, g^{a_i^1}$ 和第 5 步得到的 r_j 计算 GC_j 中 P_1 的输入导线密钥 $k_{i,j}^0 = H(g^{a_i^0\cdot r_j}), k_{i,j}^1 = H(g^{a_i^1\cdot r_j})$.
  2. P_2 将自己的输入导线密钥设置为切分选择 OT 中获得的值.
  3. 已知所有的输入导线密钥，P_2 解密所有的混淆表来检查混淆电路的正确性.
  4. 如果存在不正确的混淆电路，P_2 输出 $\perp$ 并终止协议.

LP11 协议

- **7. 发送“计算电路”中的密钥：**
  1. 对于 $j \notin \mathcal{J}$，$i \in [\ell]$，P_1 发送 $k'_{i,j} = g^{a_i^{x_i} \cdot r_j}$ 给 P_2．P_2 设置 $k_{i,j} = H(k'_{i,j})$.
  2. P_1 证明输入密钥的一致性：对于 $i \in [\ell]$，P_1 使用图 ??的协议并行地证明存在 $\sigma_i \in \{0,1\}$ 使得 $k'_{i,j} = g^{a_i^{\sigma_i} \cdot r_j}$ 对 $j \notin \mathcal{J}$ 都成立．如果证明不通过，P_2 输出 $\perp$ 并终止协议.

LP11 协议

- **7. 发送“计算电路”中的密钥：**
  1. 对于 $j \notin \mathcal{J}$，$i \in [\ell]$，P_1 发送 $k'_{i,j} = g^{a_i^{x_i} \cdot r_j}$ 给 P_2．P_2 设置 $k_{i,j} = H(k'_{i,j})$.
  2. P_1 证明输入密钥的一致性：对于 $i \in [\ell]$，P_1 使用图 ??的协议并行地证明存在 $\sigma_i \in \{0,1\}$ 使得 $k'_{i,j} = g^{a_i^{\sigma_i} \cdot r_j}$ 对 $j \notin \mathcal{J}$ 都成立．如果证明不通过，P_2 输出 $\perp$ 并终止协议.
- **8. 对电路计算：**
  1. P_2 用第 7 步获得的 P_1 的输入密钥和第 2 步获得的 P_2 的输入密钥计算所有的“计算电路” $\{GC_j\}_{j \notin \mathcal{J}}$.
  2. 如果某个电路的计算失败了，P_2 将其输入记为 $\perp$.
  3. 最后，P_2 取大多数电路的输出作为协议的输出.

安全性证明目标（情况 1：P_1 被攻陷）

证明目标

在 $\mathcal{F}_{\mathsf{CCOT}}^{S,B}$-混合模型下，构造模拟器 $\mathcal{S}$，使得

$$\left\{\mathrm{IDEAL}_{f,\mathcal{S}}(x,y,n,s)\right\}_{x,y\in\{0,1\}^{\ell};\, n,s\in\mathbb{N}} \overset{\mathrm{comp}}{\approx} \left\{\mathrm{REAL}_{\Pi,\mathcal{A}}(x,y,n,s)\right\}_{x,y\in\{0,1\}^{\ell};\, n,s\in\mathbb{N}}.$$

安全性证明目标（情况 1：P_1 被攻陷）

证明目标

在 $\mathcal{F}_{\mathsf{CCOT}}^{S,B}$-混合模型下，构造模拟器 $\mathcal{S}$，使得

$$\{\mathrm{IDEAL}_{f,\mathcal{S}}(x,y,n,s)\}_{x,y\in\{0,1\}^\ell;\, n,s\in\mathbb{N}} \overset{\mathrm{comp}}{\approx} \{\mathrm{REAL}_{\Pi,\mathcal{A}}(x,y,n,s)\}_{x,y\in\{0,1\}^\ell;\, n,s\in\mathbb{N}}.$$

核心困难

- P_1 可能构造错误混淆电路.
- 需要证明：检查电路都正确但计算电路多数错误的概率可忽略.
- 需要从步骤 7(b) 的零知识证明中提取 P_1 的一致输入 x.

模拟器 $\mathcal{S}$（P_1 被攻陷）I

1. $\mathcal{S}$ 内部运行敌手 $\mathcal{A}$（控制 P_1），记录其在 CCOT 中输入 $\{(z_0^{i,j}, z_1^{i,j})\}_{i\in[\ell],j\in[s]}$.

模拟器 $\mathcal{S}$（P_1 被攻陷）I

1. $\mathcal{S}$ 内部运行敌手 $\mathcal{A}$（控制 P_1），记录其在 CCOT 中输入 $\{(z_0^{i,j}, z_1^{i,j})\}_{i\in[\ell],j\in[s]}$.
2. 记录 P_1 发送的 $GC_1, \ldots, GC_s$ 及承诺 $\{(i,0,u_i^0),(i,1,u_i^1)\}_{i=1}^{\ell}, \{(j,h_j)\}_{j=1}^{s}$.

模拟器 $\mathcal{S}$（P_1 被攻陷）I

1. $\mathcal{S}$ 内部运行敌手 $\mathcal{A}$（控制 P_1），记录其在 CCOT 中输入 $\{(z_0^{i,j}, z_1^{i,j})\}_{i\in[\ell],j\in[s]}$.
2. 记录 P_1 发送的 $GC_1, \ldots, GC_s$ 及承诺 $\{(i,0,u_i^0),(i,1,u_i^1)\}_{i=1}^{\ell}, \{(j,h_j)\}_{j=1}^{s}$.
3. $\mathcal{S}$ 均匀随机选取 $|\mathcal{J}| = s/2$，并模拟 P_2 发送 $\mathcal{J}$ 与 $\{(z_0^{1,j}, z_1^{1,j})\}_{j\in\mathcal{J}}$（协议第 4 步）.

模拟器 $\mathcal{S}$（P_1 被攻陷）I

1. $\mathcal{S}$ 内部运行敌手 $\mathcal{A}$（控制 P_1），记录其在 CCOT 中输入 $\{(z_0^{i,j}, z_1^{i,j})\}_{i\in[\ell],j\in[s]}$.
2. 记录 P_1 发送的 $GC_1, \ldots, GC_s$ 及承诺 $\{(i,0,u_i^0),(i,1,u_i^1)\}_{i=1}^{\ell}, \{(j,h_j)\}_{j=1}^{s}$.
3. $\mathcal{S}$ 均匀随机选取 $|\mathcal{J}| = s/2$，并模拟 P_2 发送 $\mathcal{J}$ 与 $\{(z_0^{1,j}, z_1^{1,j})\}_{j\in\mathcal{J}}$（协议第 4 步）.
4. 记录 $\{r_j\}_{j\in\mathcal{J}}$，检查 $h_j \stackrel{?}{=} g^{r_j}$. 若失败，向理想功能发送 $\perp$ 并终止.

模拟器 $\mathcal{S}$（P_1 被攻陷）II

5. **按协议检查 $\{GC_j\}_{j\in\mathcal{J}}$ 正确性；若失败则发送 $\perp$ 并终止.**

模拟器 $\mathcal{S}$（P_1 被攻陷）II

5. 按协议检查 $\{GC_j\}_{j\in\mathcal{J}}$ 正确性；若失败则发送 $\perp$ 并终止.
6. 记录第 7(a) 步消息 $\{k'_{i,j}\}_{i\in[\ell],\, j\notin\mathcal{J}}$.

模拟器 $\mathcal{S}$（P_1 被攻陷）II

5. 按协议检查 $\{GC_j\}_{j\in\mathcal{J}}$ 正确性；若失败则发送 $\perp$ 并终止.
6. 记录第 7(a) 步消息 $\{k'_{i,j}\}_{i\in[\ell],\, j\notin\mathcal{J}}$.
7. 从第 7(b) 步 ZK 证明提取见证 a_i，满足

$$k'_{i,j} = (h_j)^{a_i}\ (j\notin\mathcal{J}), \quad u_i^0 = g^{a_i} \ \vee\ u_i^1 = g^{a_i}.$$

若任一 i 提取失败，则发送 $\perp$ 并终止.

模拟器 $\mathcal{S}$（P_1 被攻陷）II

5. 按协议检查 $\{GC_j\}_{j\in\mathcal{J}}$ 正确性；若失败则发送 $\perp$ 并终止.
6. 记录第 7(a) 步消息 $\{k'_{i,j}\}_{i\in[\ell],\, j\notin\mathcal{J}}$.
7. 从第 7(b) 步 ZK 证明提取见证 a_i，满足

$$k'_{i,j} = (h_j)^{a_i}\ (j\notin\mathcal{J}),\quad u_i^0 = g^{a_i}\ \vee\ u_i^1 = g^{a_i}.$$

   若任一 i 提取失败，则发送 $\perp$ 并终止.
8. 对每个 $i\in[\ell]$：若 $u_i^0 = g^{a_i}$ 置 $x_i = 0$，若 $u_i^1 = g^{a_i}$ 置 $x_i = 1$. 输出提取到的 $x = x_1\cdots x_\ell$ 给可信方，并输出 $\mathcal{A}$ 的输出.

坏电路

坏电路

对每个电路 GC_j:

- P_1 输入导线密钥: $(g^{a_1^0 r_j}, g^{a_1^1 r_j}), \ldots, (g^{a_\ell^0 r_j}, g^{a_\ell^1 r_j})$;
- P_2 输入导线密钥: 来自 CCOT 输入 $(z_0^{i,j}, z_1^{i,j})$.

若这两部分密钥不能将 GC_j 打开为正确布尔电路 $\mathcal{C}$, 则称 GC_j 为坏电路.

坏电路

坏电路

对每个电路 GC_j:

- P_1 输入导线密钥: $(g^{a_1^0 r_j}, g^{a_1^1 r_j}), \ldots, (g^{a_\ell^0 r_j}, g^{a_\ell^1 r_j})$;
- P_2 输入导线密钥: 来自 CCOT 输入 $(z_0^{i,j}, z_1^{i,j})$.

若这两部分密钥不能将 GC_j 打开为正确布尔电路 $\mathcal{C}$, 则称 GC_j 为坏电路.

noAbort : 检查电路全正确, badMaj : 计算电路中坏电路至少一半, badTotal : 坏电路总数.

坏电路

坏电路

对每个电路 GC_j:

- P_1 输入导线密钥: $(g^{a_1^0 r_j}, g^{a_1^1 r_j}), \ldots, (g^{a_\ell^0 r_j}, g^{a_\ell^1 r_j})$;
- P_2 输入导线密钥: 来自 CCOT 输入 $(z_0^{i,j}, z_1^{i,j})$.

若这两部分密钥不能将 GC_j 打开为正确布尔电路 $\mathcal{C}$, 则称 GC_j 为坏电路.

noAbort : 检查电路全正确, badMaj : 计算电路中坏电路至少一半, badTotal : 坏电路总数.

$$\Pr[\text{noAbort} \wedge \text{badMaj}] \leq \sum_{i=s/4}^{s/2} \frac{\binom{s-i}{s/2}}{\binom{s}{s/2}} = \frac{\binom{3s/4+1}{s/2+1}}{\binom{s}{s/2}} < 2^{-(s/4-1)}.$$

坏电路

坏电路

对每个电路 GC_j：

- P_1 输入导线密钥：$(g^{a_1^0 r_j}, g^{a_1^1 r_j}), \ldots, (g^{a_\ell^0 r_j}, g^{a_\ell^1 r_j})$；
- P_2 输入导线密钥：来自 CCOT 输入 $(z_0^{i,j}, z_1^{i,j})$.

若这两部分密钥不能将 GC_j 打开为正确布尔电路 $\mathcal{C}$，则称 GC_j 为坏电路.

noAbort：检查电路全正确， badMaj：计算电路中坏电路至少一半， badTotal：坏电路总数.

$$\Pr[\mathsf{noAbort} \wedge \mathsf{badMaj}] \leq \sum_{i=s/4}^{s/2} \frac{\binom{s-i}{s/2}}{\binom{s}{s/2}} = \frac{\binom{3s/4+1}{s/2+1}}{\binom{s}{s/2}} < 2^{-(s/4-1)}.$$

- 该项仅依赖切分选择抽样，属于**统计安全**项（由参数 s 控制）.
- 即使 P_1 自适应构造坏电路，在承诺先于 $\mathcal{J}$ 的前提下，“检验通过且计算多数错误”概率仍指数小.

不可区分结论（情况 1）

条件于好事件

当 $\neg(\text{noAbort} \wedge \text{badMaj})$ 时：

- 由 ZK 证明的可靠性，提取失败概率为 $\text{negl}(n)$；
- 提取到的 x 与真实 P_1 输入一致；
- 真实执行中多数计算电路输出等于 $f(x, y)$；
- $\mathcal{S}$ 对 $\mathcal{J}$ 与挑战消息的模拟分布与真实相同.

不可区分结论（情况 1）

条件于好事件

当 $\neg(\mathsf{noAbort} \wedge \mathsf{badMaj})$ 时：

- 由 ZK 证明的可靠性，提取失败概率为 $\mathrm{negl}(n)$；
- 提取到的 x 与真实 P_1 输入一致；
- 真实执行中多数计算电路输出等于 $f(x, y)$；
- $\mathcal{S}$ 对 $\mathcal{J}$ 与挑战消息的模拟分布与真实相同.

$$\mathrm{Adv}_{\mathcal{D}} \leq \Pr[\mathsf{noAbort} \wedge \mathsf{badMaj}] + \mathrm{negl}(n) \leq 2^{-(s/4-1)} + \mathrm{negl}(n) = \mathrm{negl}(n, s).$$

因而“情况 1：P_1 被攻陷”下，REAL 与 IDEAL 计算不可区分.

模拟器 $\mathcal{S}$（情况 2：P_2 被攻陷）I

1. $\mathcal{S}$ 内部运行 $\mathcal{A}$（控制 P_2），记录其向 $\mathcal{F}_{\mathsf{CCOT}}^{S,B}$ 的输入：$\mathcal{J}$ 与 $\sigma_1, \ldots, \sigma_\ell$. 若 $|\mathcal{J}| \neq s/2$，则发送 $\perp$ 并终止.

模拟器 $\mathcal{S}$（情况 2：P_2 被攻陷）I

1. $\mathcal{S}$ 内部运行 $\mathcal{A}$（控制 P_2），记录其向 $\mathcal{F}_{\mathsf{CCOT}}^{S,B}$ 的输入：$\mathcal{J}$ 与 $\sigma_1, \ldots, \sigma_\ell$. 若 $|\mathcal{J}| \neq s/2$，则发送 $\perp$ 并终止.
2. 随机选 $a_i^0, a_i^1 \leftarrow_\$ \mathbb{Z}_q$ $(i \in [\ell])$，$r_j \leftarrow_\$ \mathbb{Z}_q$ $(j \in [s])$，并构造密钥矩阵：

$$(x_{i,j}^0, x_{i,j}^1) = \left(H(g^{a_i^0 r_j}),\ H(g^{a_i^1 r_j})\right)$$

（对应 P_1 输入导线）；P_2 输入导线密钥随机生成.

模拟器 $\mathcal{S}$（情况 2：P_2 被攻陷）I

1. $\mathcal{S}$ 内部运行 $\mathcal{A}$（控制 P_2），记录其向 $\mathcal{F}_{\mathsf{CCOT}}^{S,B}$ 的输入：$\mathcal{J}$ 与 $\sigma_1, \ldots, \sigma_\ell$. 若 $|\mathcal{J}| \neq s/2$，则发送 $\perp$ 并终止.
2. 随机选 $a_i^0, a_i^1 \leftarrow_\$ \mathbb{Z}_q$ $(i \in [\ell])$，$r_j \leftarrow_\$ \mathbb{Z}_q$ $(j \in [s])$，并构造密钥矩阵：

$$(x_{i,j}^0, x_{i,j}^1) = \left(H(g^{a_i^0 r_j}),\ H(g^{a_i^1 r_j})\right)$$

   （对应 P_1 输入导线）；P_2 输入导线密钥随机生成.
3. 将该矩阵作为发送方输入喂给理想 CCOT，令 $\mathcal{A}$ 获得一致输出：$j \in \mathcal{J}$ 得到两把密钥，$j \notin \mathcal{J}$ 得到 $\{x_{i,j}^{\sigma_i}\}_{i \in [\ell]}$.

模拟器 $\mathcal{S}$（情况 2：P_2 被攻陷）I

1. $\mathcal{S}$ 内部运行 $\mathcal{A}$（控制 P_2），记录其向 $\mathcal{F}_{\mathsf{CCOT}}^{S,B}$ 的输入：$\mathcal{J}$ 与 $\sigma_1, \ldots, \sigma_\ell$. 若 $|\mathcal{J}| \neq s/2$，则发送 $\perp$ 并终止.
2. 随机选 $a_i^0, a_i^1 \leftarrow_{\$} \mathbb{Z}_q$ $(i \in [\ell])$，$r_j \leftarrow_{\$} \mathbb{Z}_q$ $(j \in [s])$，并构造密钥矩阵：

$$(x_{i,j}^0, x_{i,j}^1) = \left(H(g^{a_i^0 r_j}),\ H(g^{a_i^1 r_j})\right)$$

   （对应 P_1 输入导线）；P_2 输入导线密钥随机生成.
3. 将该矩阵作为发送方输入喂给理想 CCOT，令 $\mathcal{A}$ 获得一致输出：$j \in \mathcal{J}$ 得到两把密钥，$j \notin \mathcal{J}$ 得到 $\{x_{i,j}^{\sigma_i}\}_{i \in [\ell]}$.
4. $\mathcal{S}$ 发送 $y = \sigma_1 \cdots \sigma_\ell$ 给可信方，得到功能输出 $z = f(x, y)$.

模拟器 $\mathcal{S}$（情况 2：P_2 被攻陷）II

5. 对 $j \in \mathcal{J}$，诚实构造检查电路 GC_j.

模拟器 $\mathcal{S}$（情况 2：P_2 被攻陷）II

5. 对 $j \in \mathcal{J}$，诚实构造检查电路 GC_j.
6. 对 $j \notin \mathcal{J}$，构造“假”电路 $\widetilde{GC}_j$，其输出恒为 z，输入导线密钥沿用前述矩阵.

模拟器 $\mathcal{S}$（情况 2：P_2 被攻陷）II

5. 对 $j \in \mathcal{J}$，诚实构造检查电路 GC_j.
6. 对 $j \notin \mathcal{J}$，构造“假”电路 $\widetilde{GC}_j$，其输出恒为 z，输入导线密钥沿用前述矩阵.
7. 模拟发送电路与承诺 $\{(i,0,g^{a_i^0}),(i,1,g^{a_i^1})\}_{i=1}^{\ell},\{(j,g^{r_j})\}_{j=1}^{s}$.

模拟器 $\mathcal{S}$（情况 2：P_2 被攻陷）II

5. 对 $j \in \mathcal{J}$，诚实构造检查电路 GC_j.
6. 对 $j \notin \mathcal{J}$，构造“假”电路 $\widetilde{GC}_j$，其输出恒为 z，输入导线密钥沿用前述矩阵.
7. 模拟发送电路与承诺 $\{(i,0,g^{a_i^0}),(i,1,g^{a_i^1})\}_{i=1}^{\ell},\{(j,g^{r_j})\}_{j=1}^{s}$.
8. 记录 $\mathcal{A}$ 发来的 $\mathcal{J}'$ 与密钥，检查：

$$\mathcal{J}' \neq \mathcal{J} \wedge \text{两把密钥都正确} \Rightarrow \text{fail},$$

密钥不正确则发送 $\perp$ 并终止；否则继续.

模拟器 $\mathcal{S}$（情况 2：P_2 被攻陷）II

5. 对 $j \in \mathcal{J}$，诚实构造检查电路 GC_j.
6. 对 $j \notin \mathcal{J}$，构造“假”电路 $\widetilde{GC}_j$，其输出恒为 z，输入导线密钥沿用前述矩阵.
7. 模拟发送电路与承诺 $\{(i,0,g^{a_i^0}),(i,1,g^{a_i^1})\}_{i=1}^{\ell},\{(j,g^{r_j})\}_{j=1}^{s}$.
8. 记录 $\mathcal{A}$ 发来的 $\mathcal{J}'$ 与密钥，检查：

$$\mathcal{J}' \neq \mathcal{J} \wedge \text{两把密钥都正确} \Rightarrow \text{fail},$$

   密钥不正确则发送 $\perp$ 并终止；否则继续.
9. 发送 $\{r_j\}_{j\in\mathcal{J}}$；对 $j \notin \mathcal{J}$ 发送 $k'_{i,j} = g^{a_i^0 r_j}$，并诚实执行输入一致性 ZK（取 $x = 0^{\ell}$）.

情况 2 的关键概率界

失败事件 fail

fail 仅在 $\mathcal{J}' \neq \mathcal{J}$ 且敌手仍给出某检查电路输入位的两把正确密钥时发生. 这要求敌手猜中未获得的随机密钥，故

$$\Pr[\mathsf{fail}] \leq s \cdot 2^{-n} = \mathrm{negl}(n).$$

情况 2 的关键概率界

失败事件 fail

fail 仅在 $\mathcal{J}' \neq \mathcal{J}$ 且敌手仍给出某检查电路输入位的两把正确密钥时发生. 这要求敌手猜中未获得的随机密钥，故

$$\Pr[\mathsf{fail}] \leq s \cdot 2^{-n} = \mathrm{negl}(n).$$

直观含义

- CCOT 的单一选择约束保证 P_2 无法在不同电路使用不一致输入.
- 因而 P_2 视图中，主要差异仅是“计算电路”由真实电路替换为假电路.

不可区分结论（情况 2）

设混合序列 $\mathsf{H}_0, \ldots, \mathsf{H}_{s/2}$：$\mathsf{H}_t$ 中前 t 个计算电路被替换为假电路，其余保持真实.

$$\left|\Pr[\mathcal{D}(\mathsf{H}_0) = 1] - \Pr[\mathcal{D}(\mathsf{H}_{s/2}) = 1]\right| \leq \sum_{t=1}^{s/2} \left|\Pr[\mathcal{D}(\mathsf{H}_{t-1}) = 1] - \Pr[\mathcal{D}(\mathsf{H}_t) = 1]\right|.$$

每一项由姚电路安全性给出 $\mathrm{negl}(n)$，故

$$\left|\Pr[\mathcal{D}(\mathsf{H}_0) = 1] - \Pr[\mathcal{D}(\mathsf{H}_{s/2}) = 1]\right| \leq \frac{s}{2} \cdot \mathrm{negl}(n) = \mathrm{negl}(n, s).$$

不可区分结论（情况 2）

设混合序列 $\mathsf{H}_0, \ldots, \mathsf{H}_{s/2}$：$\mathsf{H}_t$ 中前 t 个计算电路被替换为假电路，其余保持真实.

$$\left|\Pr[\mathcal{D}(\mathsf{H}_0)=1] - \Pr[\mathcal{D}(\mathsf{H}_{s/2})=1]\right| \le \sum_{t=1}^{s/2} \left|\Pr[\mathcal{D}(\mathsf{H}_{t-1})=1] - \Pr[\mathcal{D}(\mathsf{H}_t)=1]\right|.$$

每一项由姚电路安全性给出 $\mathrm{negl}(n)$，故

$$\left|\Pr[\mathcal{D}(\mathsf{H}_0)=1] - \Pr[\mathcal{D}(\mathsf{H}_{s/2})=1]\right| \le \frac{s}{2}\cdot \mathrm{negl}(n) = \mathrm{negl}(n,s).$$

再结合 $\Pr[\mathsf{fail}] = \mathrm{negl}(n)$，得到

$$\{\mathrm{IDEAL}_{f,\mathcal{S}}(x,y,n,s)\}_{x,y\in\{0,1\}^\ell;\, n,s\in\mathbb{N}} \overset{\mathrm{comp}}{\approx} \{\mathrm{REAL}_{\Pi,\mathcal{A}}(x,y,n,s)\}_{x,y\in\{0,1\}^\ell;\, n,s\in\mathbb{N}}.$$

因而“情况 2：P_2 被攻陷”成立.

UC 安全性

- 回顾前述证明过程，只有“从零知识证明中提取见证”这一步使用了倒带技术，

UC 安全性

- 回顾前述证明过程，只有“从零知识证明中提取见证”这一步使用了倒带技术，
- 若将协议中的 Sigma 协议替换为 **UC 安全的零知识证明协议**，则整个协议可提升到 UC 安全.

UC 安全性

- 回顾前述证明过程，只有“从零知识证明中提取见证”这一步使用了倒带技术，
- 若将协议中的 Sigma 协议替换为 **UC 安全的零知识证明协议**，则整个协议可提升到 UC 安全.
- 在公共参考字符串（CRS）模型下，存在将 Sigma 协议高效转换为 UC 安全零知识证明的构造.

UC 安全性

- 回顾前述证明过程，只有“从零知识证明中提取见证”这一步使用了倒带技术，
- 若将协议中的 Sigma 协议替换为 **UC 安全的零知识证明协议**，则整个协议可提升到 UC 安全.
- 在公共参考字符串（CRS）模型下，存在将 Sigma 协议高效转换为 UC 安全零知识证明的构造.

UC 安全性

对任意输入长度为 ℓ 的两方功能 f，在 CRS 模型与 DDH 假设下，存在协议 Π，可在静态恶意敌手下 UC-安全实现功能 f.

Q & A