基于线性秘密分享的协议

BGW 协议与 GMW 协议 (半诚实安全)

《安全多方计算——可证明安全视角》第五章

2026 年 1 月 25 日

目录

BGW 协议概述 (Ben-Or, Goldwasser, Wigderson '88)

基本设定

- **基础**: Shamir (t, n)-门限秘密分享 (基于多项式插值).
- **安全门限**: $t < n/2$ (诚实多数).
- **模型**: 算术电路 (加法门、乘法门)，半诚实敌手.
- **符号**: $[a; f_a]_t$ 表示秘密 a 由 t 次多项式 f_a 分享.

BGW 协议流程：计算阶段

1. 线性运算 (本地计算):

- **加法**: $[a+b; f_a+f_b]_t = [a; f_a]_t + [b; f_b]_t$.
- **标量乘法**: $[c \cdot a; c \cdot f_a]_t = c \cdot [a; f_a]_t$.

2. 乘法运算 (交互计算):

1. **本地相乘**: 各方计算 $v_i = f_a(\alpha_i) \cdot f_b(\alpha_i)$.
   - 此时 v_i 是 $2t$ 次多项式 $h(x) = f_a(x) f_b(x)$ 的点.
2. **降维 (Degree Reduction)**:
   - 利用重组向量 $\boldsymbol{r}$ (满足 $h(0) = \sum r_i h(\alpha_i)$).
   - 各方将 v_i 作为新秘密进行分享 $[v_i]_t$.
   - 计算 $[ab]_t = \sum_{i=1}^{n} r_i \cdot [v_i]_t$.

BGW 协议的安全性

理想功能 $\mathcal{F}_{\mathsf{sfe}}$

该理想功能 $\mathcal{F}_{\mathsf{sfe}}$ 与参与方 $P_1, \ldots, P_n$ 以及敌手 $\mathcal{S}$ 交互. 它的参数是函数 f 满足 $(y_1, \ldots, y_n) = f(x_1, \ldots, x_n)$.

- 当收到来自 P_i 的消息 $(\textsc{Input}, \mathsf{sid}, x_i)$ 时，该理想功能 $\mathcal{F}_{\mathsf{sfe}}$ 做如下操作：
  - 记录 $(\textsc{Input}, \mathsf{sid}, x_i)$，并发送 $(\textsc{Input}, \mathsf{sid}, P_i)$ 给敌手 $\mathcal{S}$.
- 当收到来自 $\mathcal{S}$ 的 $(\textsc{Output}, \mathsf{sid}, P_i)$ 时，如果所有参与方均已提供输入，该理想功能 $\mathcal{F}_{\mathsf{sfe}}$ 做如下操作：
  - 发送 $(\textsc{Output}, \mathsf{sid}, y_i)$ 给接收方 P_i.

BGW 协议的安全性

理想功能 $\mathcal{F}_{\text{sfe}}$

该理想功能 $\mathcal{F}_{\text{sfe}}$ 与参与方 $P_1, \ldots, P_n$ 以及敌手 $\mathcal{S}$ 交互. 它的参数是函数 f 满足 $(y_1, \ldots, y_n) = f(x_1, \ldots, x_n)$.

- 当收到来自 P_i 的消息 $(\text{INPUT}, \text{sid}, x_i)$ 时，该理想功能 $\mathcal{F}_{\text{sfe}}$ 做如下操作：
  - 记录 $(\text{INPUT}, \text{sid}, x_i)$，并发送 $(\text{INPUT}, \text{sid}, P_i)$ 给敌手 $\mathcal{S}$.
- 当收到来自 $\mathcal{S}$ 的 $(\text{OUTPUT}, \text{sid}, P_i)$ 时，如果所有参与方均已提供输入，该理想功能 $\mathcal{F}_{\text{sfe}}$ 做如下操作：
  - 发送 $(\text{OUTPUT}, \text{sid}, y_i)$ 给接收方 P_i.

定理 (BGW 半诚实安全)

假设参与方之间存在点对点安全信道，且被攻陷方数量 $|C| \leq t$，其中 $t < n/2$. 协议 Π_{BGW} 对于静态半诚实敌手 *UC*-安全实现了理想功能 $\mathcal{F}_{\text{sfe}}$.

为简化证明，假设每个输出门前紧跟一个乘法门；必要时可增加常数个乘法门而不影响安全性.

证明目标与记号

- 目标：构造模拟器 $\mathcal{S}$，使得对任意环境 $\mathcal{Z}$，

$$\mathrm{EXEC}_{\Pi_{\mathsf{BGW}},\mathcal{A},\mathcal{Z}} \approx \mathrm{EXEC}_{\mathcal{F}_{\mathsf{sfe}},\mathcal{S},\mathcal{Z}}$$

- $\mathcal{A}$ 为平凡敌手；协议信息论安全，不需要限制 $\mathcal{Z}$ 为 PPT.
- 令 $C \subset \{P_1, \ldots, P_n\}$ 为被攻陷方集合，$|C| \leq t$.

模拟器 $\mathcal{S}$

- **输入分享阶段**：
  - 若收到 $(\text{INPUT}, \text{sid}, P_i)$ 且 $P_i \in C$，$\mathcal{S}$ 以真实输入 x_i 模拟 P_i 执行输入分享.
  - 若 $P_j \notin C$，设定 $x_j = 0$，模拟 P_j 执行输入分享，并将份额发送给被攻陷方.

模拟器 $\mathcal{S}$

- **输入分享阶段**：
  - 若收到 $(\text{INPUT}, \text{sid}, P_i)$ 且 $P_i \in C$，$\mathcal{S}$ 以真实输入 x_i 模拟 P_i 执行输入分享.
  - 若 $P_j \notin C$，设定 $x_j = 0$，模拟 P_j 执行输入分享，并将份额发送给被攻陷方.
- **加法门/常数乘法门**：$\mathcal{S}$ 仅模拟被攻陷方遵循协议执行.

模拟器 $\mathcal{S}$

- **输入分享阶段**：
  - 若收到 $(\text{INPUT}, \text{sid}, P_i)$ 且 $P_i \in C$，$\mathcal{S}$ 以真实输入 x_i 模拟 P_i 执行输入分享.
  - 若 $P_j \notin C$，设定 $x_j = 0$，模拟 P_j 执行输入分享，并将份额发送给被攻陷方.
- **加法门/常数乘法门**：$\mathcal{S}$ 仅模拟被攻陷方遵循协议执行.
- **乘法门**：模拟被攻陷方执行；对每个诚实方模拟其向被攻陷方分发 $[0]_t$.

模拟器 $\mathcal{S}$

- **输入分享阶段**：
  - 若收到 $(\textsc{Input}, \mathsf{sid}, P_i)$ 且 $P_i \in C$，$\mathcal{S}$ 以真实输入 x_i 模拟 P_i 执行输入分享.
  - 若 $P_j \notin C$，设定 $x_j = 0$，模拟 P_j 执行输入分享，并将份额发送给被攻陷方.
- **加法门/常数乘法门**：$\mathcal{S}$ 仅模拟被攻陷方遵循协议执行.
- **乘法门**：模拟被攻陷方执行；对每个诚实方模拟其向被攻陷方分发 $[0]_t$.
- **输出重建**：当需要重建 P_i 的输出时，$\mathcal{S}$ 向 $\mathcal{F}_{\mathsf{sfe}}$ 发送 $(\textsc{Output}, \mathsf{sid}, P_i)$.

模拟器 $\mathcal{S}$

- **输入分享阶段**：
  - 若收到 $(\textsc{Input}, \mathsf{sid}, P_i)$ 且 $P_i \in C$，$\mathcal{S}$ 以真实输入 x_i 模拟 P_i 执行输入分享.
  - 若 $P_j \notin C$，设定 $x_j = 0$，模拟 P_j 执行输入分享，并将份额发送给被攻陷方.
- **加法门/常数乘法门**：$\mathcal{S}$ 仅模拟被攻陷方遵循协议执行.
- **乘法门**：模拟被攻陷方执行；对每个诚实方模拟其向被攻陷方分发 $[0]_t$.
- **输出重建**：当需要重建 P_i 的输出时，$\mathcal{S}$ 向 $\mathcal{F}_{\mathsf{sfe}}$ 发送 $(\textsc{Output}, \mathsf{sid}, P_i)$.
  - 若 $P_i \in C$，得到 y_i，构造 t 次多项式 f_{y_i} 满足 $f_{y_i}(0) = y_i$ 且与已知的 $|C|$ 个份额一致:

模拟器 $\mathcal{S}$

- **输入分享阶段**：
  - 若收到 $(\textsc{Input}, \mathsf{sid}, P_i)$ 且 $P_i \in C$，$\mathcal{S}$ 以真实输入 x_i 模拟 P_i 执行输入分享.
  - 若 $P_j \notin C$，设定 $x_j = 0$，模拟 P_j 执行输入分享，并将份额发送给被攻陷方.
- **加法门/常数乘法门**：$\mathcal{S}$ 仅模拟被攻陷方遵循协议执行.
- **乘法门**：模拟被攻陷方执行；对每个诚实方模拟其向被攻陷方分发 $[0]_t$.
- **输出重建**：当需要重建 P_i 的输出时，$\mathcal{S}$ 向 $\mathcal{F}_{\mathsf{sfe}}$ 发送 $(\textsc{Output}, \mathsf{sid}, P_i)$.
  - 若 $P_i \in C$，得到 y_i，构造 t 次多项式 f_{y_i} 满足 $f_{y_i}(0) = y_i$ 且与已知的 $|C|$ 个份额一致:
    - 若 $|C| = t$，用拉格朗日插值构造 f_{y_i}；

模拟器 $\mathcal{S}$

- **输入分享阶段**：
  - 若收到 $(\textsc{Input}, \mathsf{sid}, P_i)$ 且 $P_i \in C$，$\mathcal{S}$ 以真实输入 x_i 模拟 P_i 执行输入分享.
  - 若 $P_j \notin C$，设定 $x_j = 0$，模拟 P_j 执行输入分享，并将份额发送给被攻陷方.
- **加法门/常数乘法门**：$\mathcal{S}$ 仅模拟被攻陷方遵循协议执行.
- **乘法门**：模拟被攻陷方执行；对每个诚实方模拟其向被攻陷方分发 $[0]_t$.
- **输出重建**：当需要重建 P_i 的输出时，$\mathcal{S}$ 向 $\mathcal{F}_{\mathsf{sfe}}$ 发送 $(\textsc{Output}, \mathsf{sid}, P_i)$.
  - 若 $P_i \in C$，得到 y_i，构造 t 次多项式 f_{y_i} 满足 $f_{y_i}(0) = y_i$ 且与已知的 $|C|$ 个份额一致：
    - 若 $|C| = t$，用拉格朗日插值构造 f_{y_i}；
    - 若 $|C| < t$，先均匀随机选取 $t - |C|$ 个诚实方份额再插值.

模拟器 $\mathcal{S}$

- **输入分享阶段**：
  - 若收到 $(\textsc{Input}, \mathsf{sid}, P_i)$ 且 $P_i \in C$，$\mathcal{S}$ 以真实输入 x_i 模拟 P_i 执行输入分享.
  - 若 $P_j \notin C$，设定 $x_j = 0$，模拟 P_j 执行输入分享，并将份额发送给被攻陷方.
- **加法门/常数乘法门**：$\mathcal{S}$ 仅模拟被攻陷方遵循协议执行.
- **乘法门**：模拟被攻陷方执行；对每个诚实方模拟其向被攻陷方分发 $[0]_t$.
- **输出重建**：当需要重建 P_i 的输出时，$\mathcal{S}$ 向 $\mathcal{F}_{\mathsf{sfe}}$ 发送 $(\textsc{Output}, \mathsf{sid}, P_i)$.
  - 若 $P_i \in C$，得到 y_i，构造 t 次多项式 f_{y_i} 满足 $f_{y_i}(0) = y_i$ 且与已知的 $|C|$ 个份额一致:
    - 若 $|C| = t$，用拉格朗日插值构造 f_{y_i};
    - 若 $|C| < t$，先均匀随机选取 $t - |C|$ 个诚实方份额再插值.
  - 对于 $P_j \notin C$，发送 $f_{y_i}(\alpha_j)$ 作为 P_j 的份额.

引理 1：Strip

定义：对被攻陷方 $P_i \in C$ 的视图 view_i，$\text{Strip}(\text{view}_i)$ 移除输出重建阶段来自诚实方的份额，仅保留输入 x_i、随机性、输入分享/乘法阶段收到的份额，以及输出重建阶段来自被攻陷方的份额.

引理 1

令输入向量 $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}$ 满足对所有 $P_i \in C$ 有 $x_i^{(0)} = x_i^{(1)}$. 则

$$\{\text{Strip}(\text{view}_i^{(0)})\}_{P_i \in C} \overset{\text{perf}}{=} \{\text{Strip}(\text{view}_i^{(1)})\}_{P_i \in C}.$$

证明要点：

- 输入分享与乘法门中，被攻陷方看到的不超过 t 个份额均为均匀随机值；
- 加法门和常数乘法门无需交互；
- Strip 移除了输出重建阶段来自诚实方的份额；

引理 2：Dress

定义：给定 $\boldsymbol{y}_C = \{y_i\}_{P_i \in C}$，$\mathsf{Dress}_{\boldsymbol{y}_C}$ 接受 $\{\mathsf{Strip}(\mathsf{view}_i)\}_{P_i \in C}$，并按照模拟器的方式为每个 $P_i \in C$ 补全诚实方在输出重建阶段的份额：构造 t 次多项式 f_{y_i}，满足 $f_{y_i}(0) = y_i$ 且与已知份额一致.

引理 2

对任意输入向量 $\boldsymbol{x}$ 和输出 $\boldsymbol{y} = f(\boldsymbol{x})$，

$$\mathsf{Dress}_{\boldsymbol{y}_C}\big(\{\mathsf{Strip}(\mathsf{view}_i)\}_{P_i \in C}\big) \overset{\mathsf{perf}}{=} \{\mathsf{view}_i\}_{P_i \in C}.$$

证明要点：

- 当 $|C| < t$ 时补充的随机份额与真实分布一致；
- 拉格朗日插值唯一确定剩余的诚实方份额.

不可区分性

令 $\boldsymbol{x}^{(0)}$ 满足对 $P_i \in C$ 有 $x_i^{(0)} = x_i$，对 $P_j \notin C$ 有 $x_j^{(0)} = 0$. 理想世界视图是

$$\mathsf{Dress}_{\boldsymbol{y}_C}(\{\mathsf{Strip}(\mathsf{view}_i^{(0)})\}_{P_i \in C})$$

不可区分性

令 $\boldsymbol{x}^{(0)}$ 满足对 $P_i \in C$ 有 $x_i^{(0)} = x_i$，对 $P_j \notin C$ 有 $x_j^{(0)} = 0$. 理想世界视图是

$$\mathsf{Dress}_{\boldsymbol{y}_C}(\{\mathsf{Strip}(\mathsf{view}_i^{(0)})\}_{P_i \in C})$$

令真实世界的视图为 $\{\mathsf{view}_i\}_{P_i \in C}$，根据引理 1 可知

$$\{\mathsf{Strip}(\mathsf{view}_i^{(0)})\}_{P_i \in C} \overset{\mathsf{perf}}{=} \{\mathsf{Strip}(\mathsf{view}_i)\}_{P_i \in C}$$

不可区分性

令 $\boldsymbol{x}^{(0)}$ 满足对 $P_i \in C$ 有 $x_i^{(0)} = x_i$，对 $P_j \notin C$ 有 $x_j^{(0)} = 0$. 理想世界视图是

$$\mathsf{Dress}_{\boldsymbol{y}_C}(\{\mathsf{Strip}(\mathsf{view}_i^{(0)})\}_{P_i \in C})$$

令真实世界的视图为 $\{\mathsf{view}_i\}_{P_i \in C}$，根据引理 1 可知

$$\{\mathsf{Strip}(\mathsf{view}_i^{(0)})\}_{P_i \in C} \overset{\mathsf{perf}}{=} \{\mathsf{Strip}(\mathsf{view}_i)\}_{P_i \in C}$$

因为算法不能增加两个随机变量的统计距离，所以

$$\mathsf{Dress}_{\boldsymbol{y}_C}(\{\mathsf{Strip}(\mathsf{view}_i^{(0)})\}_{P_i \in C}) \overset{\mathsf{perf}}{=} \mathsf{Dress}_{\boldsymbol{y}_C}(\{\mathsf{Strip}(\mathsf{view}_i)\}_{P_i \in C})$$

不可区分性

令 $\boldsymbol{x}^{(0)}$ 满足对 $P_i \in C$ 有 $x_i^{(0)} = x_i$，对 $P_j \notin C$ 有 $x_j^{(0)} = 0$. 理想世界视图是

$$\mathsf{Dress}_{\boldsymbol{y}_C}(\{\mathsf{Strip}(\mathsf{view}_i^{(0)})\}_{P_i \in C})$$

令真实世界的视图为 $\{\mathsf{view}_i\}_{P_i \in C}$，根据引理 1 可知

$$\{\mathsf{Strip}(\mathsf{view}_i^{(0)})\}_{P_i \in C} \overset{\mathsf{perf}}{=} \{\mathsf{Strip}(\mathsf{view}_i)\}_{P_i \in C}$$

因为算法不能增加两个随机变量的统计距离，所以

$$\mathsf{Dress}_{\boldsymbol{y}_C}(\{\mathsf{Strip}(\mathsf{view}_i^{(0)})\}_{P_i \in C}) \overset{\mathsf{perf}}{=} \mathsf{Dress}_{\boldsymbol{y}_C}(\{\mathsf{Strip}(\mathsf{view}_i)\}_{P_i \in C})$$

再由引理 2，

$$\mathsf{Dress}_{\boldsymbol{y}_C}(\{\mathsf{Strip}(\mathsf{view}_i)\}_{P_i \in C}) \overset{\mathsf{perf}}{=} \{\mathsf{view}_i\}_{P_i \in C}.$$

因此真实世界与理想世界视图完美一致，定理得证.

不可区分性

令 $\boldsymbol{x}^{(0)}$ 满足对 $P_i \in C$ 有 $x_i^{(0)} = x_i$，对 $P_j \notin C$ 有 $x_j^{(0)} = 0$. 理想世界视图是

$$\mathsf{Dress}_{\boldsymbol{y}_C}(\{\mathsf{Strip}(\mathsf{view}_i^{(0)})\}_{P_i \in C})$$

令真实世界的视图为 $\{\mathsf{view}_i\}_{P_i \in C}$，根据引理 1 可知

$$\{\mathsf{Strip}(\mathsf{view}_i^{(0)})\}_{P_i \in C} \overset{\mathsf{perf}}{=} \{\mathsf{Strip}(\mathsf{view}_i)\}_{P_i \in C}$$

因为算法不能增加两个随机变量的统计距离，所以

$$\mathsf{Dress}_{\boldsymbol{y}_C}(\{\mathsf{Strip}(\mathsf{view}_i^{(0)})\}_{P_i \in C}) \overset{\mathsf{perf}}{=} \mathsf{Dress}_{\boldsymbol{y}_C}(\{\mathsf{Strip}(\mathsf{view}_i)\}_{P_i \in C})$$

再由引理 2，

$$\mathsf{Dress}_{\boldsymbol{y}_C}(\{\mathsf{Strip}(\mathsf{view}_i)\}_{P_i \in C}) \overset{\mathsf{perf}}{=} \{\mathsf{view}_i\}_{P_i \in C}.$$

因此真实世界与理想世界视图完美一致，定理得证.

注：若输出门之前没有乘法门，敌手可检验 $f_a + f_b \overset{?}{=} f_{a+b}$ 区分真实世界和理想世界.
加入乘法门可引入新随机性以简化证明.

GMW 协议概述 (Goldreich, Micali, Wigderson '87)

基本设定

- **基础**: 加法秘密分享 (XOR Sharing).

$$x = x_1 \oplus x_2 \oplus \cdots \oplus x_n$$

- **安全门限**: $t < n$ (允许 $n-1$ 个恶意方).
- **模型**: 布尔电路 (XOR, AND, NOT) 或算术电路.
- **核心工具**: 茫然传输 (OT).

布尔电路 GMW

假设参与方持有 a, b 的份额 a_i, b_i.

布尔电路 GMW

假设参与方持有 a, b 的份额 a_i, b_i.

- **XOR 门 (本地)**: $c_i = a_i \oplus b_i$.

布尔电路 GMW

假设参与方持有 a, b 的份额 a_i, b_i.

- **XOR 门 (本地)**: $c_i = a_i \oplus b_i$.
- **NOT 门 (本地)**: P_1 计算 $c_1 = a_1 \oplus 1$，其他人不变.

布尔电路 GMW

假设参与方持有 a, b 的份额 a_i, b_i.

- **XOR 门 (本地)**: $c_i = a_i \oplus b_i$.
- **NOT 门 (本地)**: P_1 计算 $c_1 = a_1 \oplus 1$，其他人不变.
- **AND 门 (交互)**: $c = (\bigoplus a_i) \wedge (\bigoplus b_j)$

$$c = \left(\bigoplus_i a_i b_i \right) \oplus \left(\bigoplus_{i<j} (a_i b_j \oplus a_j b_i) \right)$$

布尔电路 GMW

假设参与方持有 a, b 的份额 a_i, b_i.

- **XOR 门 (本地)**: $c_i = a_i \oplus b_i$.
- **NOT 门 (本地)**: P_1 计算 $c_1 = a_1 \oplus 1$，其他人不变.
- **AND 门 (交互)**: $c = (\bigoplus a_i) \wedge (\bigoplus b_j)$

$$c = \left(\bigoplus_i a_i b_i\right) \oplus \left(\bigoplus_{i<j} (a_i b_j \oplus a_j b_i)\right)$$

  - $a_i b_i$: 本地计算.

布尔电路 GMW

假设参与方持有 a, b 的份额 a_i, b_i.

- **XOR 门 (本地)**: $c_i = a_i \oplus b_i$.
- **NOT 门 (本地)**: P_1 计算 $c_1 = a_1 \oplus 1$，其他人不变.
- **AND 门 (交互)**: $c = (\bigoplus a_i) \wedge (\bigoplus b_j)$

$$c = \left(\bigoplus_i a_i b_i \right) \oplus \left(\bigoplus_{i<j} (a_i b_j \oplus a_j b_i) \right)$$

  - $a_i b_i$: 本地计算.
  - $a_i b_j$: 需要 P_i 和 P_j 交互. 利用 **1-out-of-4 OT** 计算交叉项.

GMW 中的 AND 门协议 (利用 OT)

目标：P_i (持有 a_i, b_i) 和 P_j (持有 a_j, b_j) 计算 $a_i b_j \oplus a_j b_i$ 的份额.

协议步骤

GMW 中的 AND 门协议 (利用 OT)

目标：P_i (持有 a_i, b_i) 和 P_j (持有 a_j, b_j) 计算 $a_i b_j \oplus a_j b_i$ 的份额.

协议步骤

1. P_i 构造 OT 表 (对应 P_j 的 4 种可能输入):

$$T_{u,v} = (a_i \cdot v) \oplus (u \cdot b_i) \oplus r \quad (u, v \in \{0, 1\})$$

GMW 中的 AND 门协议 (利用 OT)

目标：P_i (持有 a_i, b_i) 和 P_j (持有 a_j, b_j) 计算 $a_i b_j \oplus a_j b_i$ 的份额.

协议步骤

1. P_i 构造 OT 表 (对应 P_j 的 4 种可能输入):

$$T_{u,v} = (a_i \cdot v) \oplus (u \cdot b_i) \oplus r \quad (u, v \in \{0, 1\})$$

2. P_i 和 P_j 执行 1-out-of-4 OT:
   - P_i 作为发送方，输入表 T.
   - P_j 作为接收方，输入选择 (a_j, b_j).

GMW 中的 AND 门协议 (利用 OT)

目标：P_i (持有 a_i, b_i) 和 P_j (持有 a_j, b_j) 计算 $a_i b_j \oplus a_j b_i$ 的份额.

协议步骤

1. P_i 构造 OT 表 (对应 P_j 的 4 种可能输入):

$$T_{u,v} = (a_i \cdot v) \oplus (u \cdot b_i) \oplus r \quad (u, v \in \{0, 1\})$$

2. P_i 和 P_j 执行 1-out-of-4 OT:
   - P_i 作为发送方，输入表 T.
   - P_j 作为接收方，输入选择 (a_j, b_j).
3. 结果分配:
   - P_i 获得份额 r.
   - P_j 获得 OT 输出 $z = (a_i b_j \oplus a_j b_i) \oplus r$.

布尔电路 GMW 协议的安全性

理想功能 $\mathcal{F}_{\mathsf{OT}_4^1}$

输入：发送方 P_s 的输入为四条消息 $(x_{0,0}, x_{0,1}, x_{1,0}, x_{1,1})$；接收方 P_r 的输入为选择比特 (b_0, b_1).

输出：接收方 P_r 得到 x_{b_0,b_1}，发送方无输出.

布尔电路 GMW 协议的安全性

理想功能 $\mathcal{F}_{\mathsf{OT}_4^1}$

输入：发送方 P_s 的输入为四条消息 $(x_{0,0}, x_{0,1}, x_{1,0}, x_{1,1})$；接收方 P_r 的输入为选择比特 (b_0, b_1).

输出：接收方 P_r 得到 x_{b_0,b_1}，发送方无输出.

定理 (GMW 布尔电路)

假设被攻陷方数量不超过 $n-1$. 协议 $\Pi_{\mathsf{GMW}}^{\mathsf{bool}}$ 在 $\mathcal{F}_{\mathsf{OT}_4^1}$-混合模型下对静态半诚实敌手 *UC*-安全实现了理想功能 $\mathcal{F}_{\mathsf{sfe}}$.

由于 OT 只能计算安全，本证明采用计算不可区分性；环境 $\mathcal{Z}$ 为 PPT.

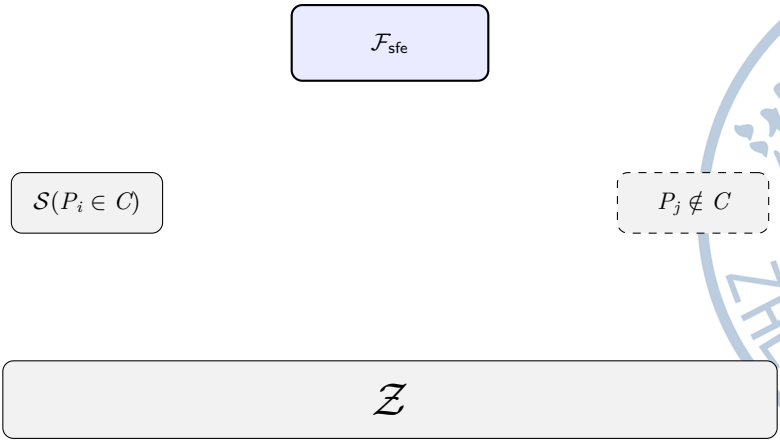
证明目标

- 构造模拟器 $\mathcal{S}$，使得对任意 PPT 环境 $\mathcal{Z}$，

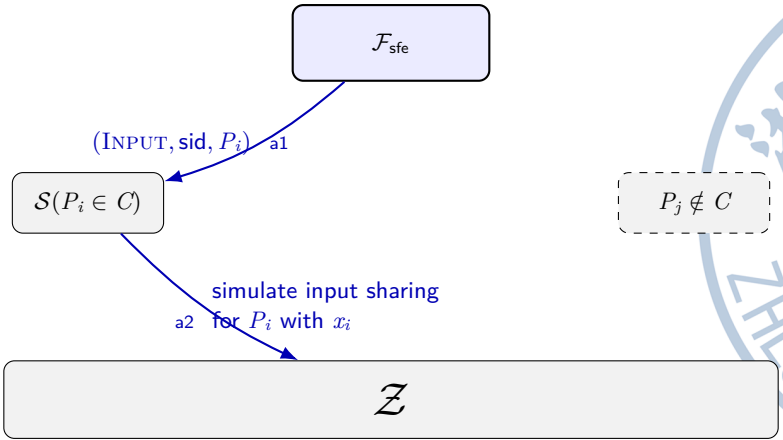
$$\mathrm{EXEC}^{\mathcal{F}_{\mathrm{OT}_4^1}}_{\Pi^{\mathsf{bool}}_{\mathsf{GMW}},\mathcal{A},\mathcal{Z}} \approx \mathrm{EXEC}_{\mathcal{F}_{\mathsf{sfe}},\mathcal{S},\mathcal{Z}}.$$

- $\mathcal{A}$ 为平凡敌手；$\mathcal{S}$ 内部运行 $\mathcal{A}$ 并转发其与 $\mathcal{Z}$ 的交互.
- 令 $C \subset \{P_1, \ldots, P_n\}$ 为被攻陷方集合，$|C| \leq n-1$.

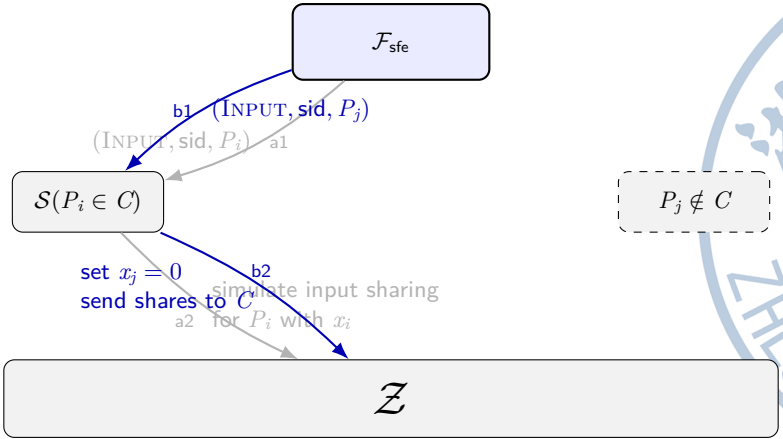
模拟器 $\mathcal{S}$



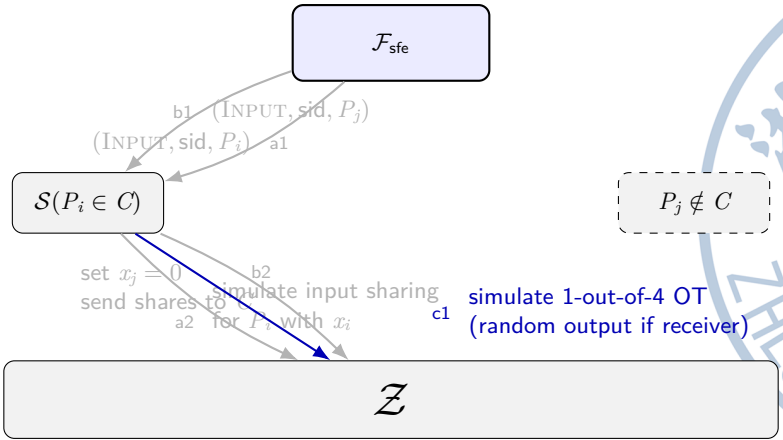
模拟器 $\mathcal{S}$



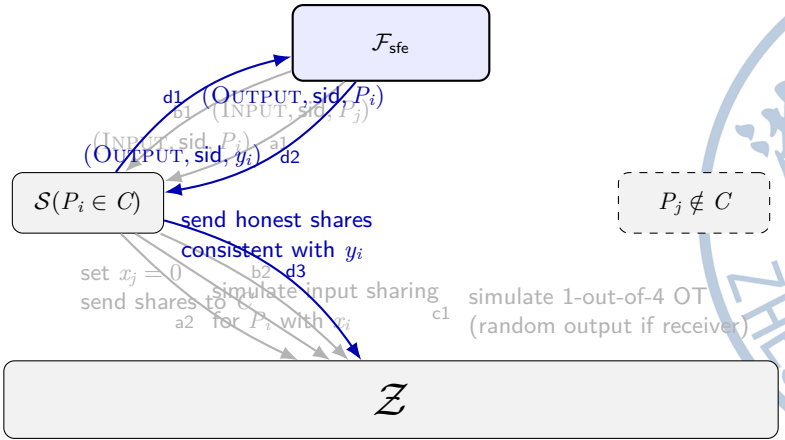
模拟器 $\mathcal{S}$



模拟器 $\mathcal{S}$



模拟器 $\mathcal{S}$



不可区分性

不可区分性

- **正确性**：协议逐门计算保持每条导线为正确的加法秘密分享，故两个世界诚实方输出相同.

不可区分性

- **正确性**：协议逐门计算保持每条导线为正确的加法秘密分享，故两个世界诚实方输出相同.
- **输入分享**：对任意 $|C| \leq n-1$，诚实方份额对被攻陷方均匀随机.

不可区分性

- **正确性**：协议逐门计算保持每条导线为正确的加法秘密分享，故两个世界诚实方输出相同.
- **输入分享**：对任意 $|C| \leq n-1$，诚实方份额对被攻陷方均匀随机.
- **AND 门 OT**：被攻陷方接收的 OT 输出与随机比特不可区分.

不可区分性

- **正确性**：协议逐门计算保持每条导线为正确的加法秘密分享，故两个世界诚实方输出相同.
- **输入分享**：对任意 $|C| \leq n-1$，诚实方份额对被攻陷方均匀随机.
- **AND 门 OT**：被攻陷方接收的 OT 输出与随机比特不可区分.
- **输出阶段**：模拟器补充的诚实方份额与真实分布一致.

不可区分性

- **正确性**：协议逐门计算保持每条导线为正确的加法秘密分享，故两个世界诚实方输出相同.
- **输入分享**：对任意 $|C| \leq n-1$，诚实方份额对被攻陷方均匀随机.
- **AND 门 OT**：被攻陷方接收的 OT 输出与随机比特不可区分.
- **输出阶段**：模拟器补充的诚实方份额与真实分布一致.

因此真实世界与理想世界的视图计算不可区分，定理成立.

算术电路 GMW 协议

扩展到有限域 $\mathbb{F}$:

- **加法/常数乘法门**: 依然本地进行.
- **乘法门**: 需要计算 $(\sum a_i)(\sum b_j)$: 将 OT 替换为 **OLE (Oblivious Linear Evaluation)**

算术电路 GMW 协议

扩展到有限域 $\mathbb{F}$:

- **加法/常数乘法门**: 依然本地进行.
- **乘法门**: 需要计算 $(\sum a_i)(\sum b_j)$: 将 OT 替换为 **OLE (Oblivious Linear Evaluation)**

OLE 的理想功能 $\mathcal{F}_{\text{OLE}}$

输入: P_s 的输入是 $\alpha, \beta \in \mathbb{F}$. P_r 的输入是 $x \in \mathbb{F}$.
输出: P_r 输出 $\gamma = \alpha x + \beta$. P_s 没有输出.

算术电路 GMW 协议

扩展到有限域 $\mathbb{F}$:

- **加法/常数乘法门**: 依然本地进行.
- **乘法门**: 需要计算 $(\sum a_i)(\sum b_j)$: 将 OT 替换为 **OLE (Oblivious Linear Evaluation)**

OLE 的理想功能 $\mathcal{F}_{\text{OLE}}$

输入: P_s 的输入是 $\alpha, \beta \in \mathbb{F}$. P_r 的输入是 $x \in \mathbb{F}$.
输出: P_r 输出 $\gamma = \alpha x + \beta$. P_s 没有输出.

OLE 的实现:

1. 使用通用 MPC 协议;
2. 使用（加法）同态加密;
3. ……

输入分享阶段

参与方 $P_1, \ldots, P_n$ 计算函数

$$f\colon \mathbb{F}^{M_{\mathsf{in}}^1} \times \cdots \times \mathbb{F}^{M_{\mathsf{in}}^n} \to \mathbb{F}^{M_{\mathsf{out}}^1} \times \cdots \times \mathbb{F}^{M_{\mathsf{out}}^n}.$$

输入分享阶段

参与方 $P_1, \ldots, P_n$ 计算函数

$$f\colon \mathbb{F}^{M_{\text{in}}^1} \times \cdots \times \mathbb{F}^{M_{\text{in}}^n} \to \mathbb{F}^{M_{\text{out}}^1} \times \cdots \times \mathbb{F}^{M_{\text{out}}^n}.$$

对于 P_i 的每个输入 $x_{i,k} \in \mathbb{F}$:

- 选择随机 $r_{i,k}^1, \ldots, r_{i,k}^n \in \mathbb{F}$，满足

$$\sum_{\ell=1}^{n} r_{i,k}^{\ell} = x_{i,k}.$$

- 将 $r_{i,k}^{\ell}$ 发送给 P_ℓ 作为秘密份额.

这是 $(n-1, n)$-门限加法秘密分享.

计算阶段

设输入导线为 w_i, w_j，输出导线为 w_k，P_ℓ 持有 s_i^ℓ, s_j^ℓ.

- **加法门**：P_ℓ 设置 $s_k^\ell = s_i^\ell + s_j^\ell$.
- **常数乘法门 (α)**：P_ℓ 设置 $s_k^\ell = \alpha \cdot s_i^\ell$.
- **常数加法门 (α)**：P_1 设置 $s_k^1 = s_i^1 + \alpha$，其余 P_ℓ 设置 $s_k^\ell = s_i^\ell$.
- **乘法门**：调用 OLE 理想功能计算交叉项.

计算阶段：乘法门

$$w_k = w_i w_j = \sum_{\ell=1}^{n} s_i^{\ell} s_j^{\ell} + \sum_{\ell_1 \neq \ell_2} s_i^{\ell_1} s_j^{\ell_2}.$$

- **本地项**：P_ℓ **计算** $s_i^{\ell} s_j^{\ell}$.

计算阶段：乘法门

$$w_k = w_i w_j = \sum_{\ell=1}^{n} s_i^{\ell} s_j^{\ell} + \sum_{\ell_1 \neq \ell_2} s_i^{\ell_1} s_j^{\ell_2}.$$

- **本地项**：P_ℓ **计算** $s_i^{\ell} s_j^{\ell}$.
- **交叉项**：**对每对** (ℓ_1, ℓ_2)，P_{ℓ_1} **随机取** $r_{\ell_1,\ell_2} \leftarrow_{\$} \mathbb{F}$，**与** P_{ℓ_2} **执行 OLE**：

$$(\alpha, \beta) = (s_i^{\ell_1}, -r_{\ell_1,\ell_2}), \quad x = s_j^{\ell_2}.$$

  接收方得到 $\gamma_{\ell_1,\ell_2} = s_i^{\ell_1} s_j^{\ell_2} - r_{\ell_1,\ell_2}$.
- **输出份额**：P_ℓ **设**

$$s_k^{\ell} = s_i^{\ell} s_j^{\ell} + \sum_{\ell' \neq \ell} (r_{\ell,\ell'} + \gamma_{\ell',\ell}).$$

输出重建阶段

对于每个输出 $y_{i,k}$:

- 其他参与方将份额 $s_{i,k}^{\ell}$ $(\ell \neq i)$ 发送给 P_i.
- P_i 计算

$$y_{i,k} = s_{i,k}^{1} + \cdots + s_{i,k}^{n}.$$

对所有输出均重复上述过程.

本章总结

特性	BGW 协议	GMW 协议
基础分享	Shamir (多项式)	加法 (XOR/Add)
安全门限	$t < n/2$	$t < n$
计算模型	算术电路	布尔/算术电路
核心开销	乘法需通信 (降维)	AND/Mult 需 OT/OLE
安全性	信息论安全	计算安全 (依赖 OT)

思考题

思考题

1. 为什么安全多方计算的函数 f 要以算术电路或者布尔电路的方式来表达？安全多方计算能否支持一般性的随机存取机器（Random Access Machine, RAM）模型计算任务，例如 C 语言代码？如何安全地处理 RAM 模型中的条件分支和跳转指令？

思考题

1. 为什么安全多方计算的函数 f 要以算术电路或者布尔电路的方式来表达？安全多方计算能否支持一般性的随机存取机器（Random Access Machine, RAM）模型计算任务，例如 C 语言代码？如何安全地处理 RAM 模型中的条件分支和跳转指令？
2. 本章节所介绍的 BGW 协议和 GMW 协议都假设参与方之间存在点对点安全信道，在现实中我们应该如何实现这些安全信道，相对应的，会引入哪些安全假设？

思考题

1. 为什么安全多方计算的函数 f 要以算术电路或者布尔电路的方式来表达？安全多方计算能否支持一般性的随机存取机器（Random Access Machine, RAM）模型计算任务，例如 C 语言代码？如何安全地处理 RAM 模型中的条件分支和跳转指令？
2. 本章节所介绍的 BGW 协议和 GMW 协议都假设参与方之间存在点对点安全信道，在现实中我们应该如何实现这些安全信道，相对应的，会引入哪些安全假设？
3. GMW 协议中与门是通过使用 OT 协议来实现的，如果使用其他密码学原语（例如，同态加密）替代 OT，协议会如何变化？请思考这种替代对安全性和效率的影响.

Q & A