引言

《安全多方计算——可证明安全视角》第一章

2026 年 1 月 26 日

目录

安全多方计算 (MPC) 概览

核心问题

一组互不信任的参与方 $P_1, \ldots, P_n$，各自持有隐私输入 x_i，希望在**不泄露隐私**的前提下协同计算函数 $y = f(x_1, \ldots, x_n)$.

安全多方计算 (MPC) 概览

核心问题

一组互不信任的参与方 $P_1, \ldots, P_n$，各自持有隐私输入 x_i，希望在**不泄露隐私**的前提下协同计算函数 $y = f(x_1, \ldots, x_n)$.

理想世界 (Trusted Third Party, TTP)

- 参与方将 x_i 发送给可信第三方 T.
- T 计算并发布 y.
- 缺点：单点故障，现实中难以找到完全可信的 T.

安全多方计算 (MPC) 概览

核心问题

一组互不信任的参与方 $P_1, \ldots, P_n$，各自持有隐私输入 x_i，希望在**不泄露隐私**的前提下协同计算函数 $y = f(x_1, \ldots, x_n)$.

理想世界 (Trusted Third Party, TTP)

- 参与方将 x_i 发送给可信第三方 T.
- T 计算并发布 y.
- 缺点：单点故障，现实中难以找到完全可信的 T.

现实世界 (Real World)

- 去中心化：没有 TTP.
- **正确性 (Correctness)**：输出 y 正确.
- **隐私性 (Privacy)**：除了 y，不泄露任何 x_i 信息.

典型应用案例

第一价格密封拍卖 (First-price sealed-bid auction)

- **输入**：竞拍者 P_i 的出价 x_i（保密）.
- **函数**：$f(x_1, \ldots, x_n) = (\max(x_i), \text{winner\_id})$.
- **要求**：只公布最高价和赢家，**严禁**泄露失败者的出价.

典型应用案例

第一价格密封拍卖 (First-price sealed-bid auction)

- **输入**：竞拍者 P_i 的出价 x_i（保密）.
- **函数**：$f(x_1, \ldots, x_n) = (\max(x_i), \text{winner\_id})$.
- **要求**：只公布最高价和赢家，**严禁**泄露失败者的出价.

电子投票

- **输入**：$x_i \in \{0, 1\}$ (反对/赞成).
- **函数**：$y = \sum x_i$.
- **要求**：统计总票数，不暴露个人投票情况.

核心工具：秘密分享 (Secret Sharing)

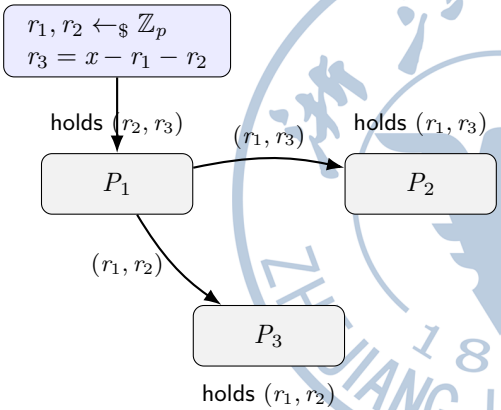
- **目标**：P_1 **想分享秘密** $x \in \mathbb{Z}_p$ **给** P_1, P_2, P_3.

核心工具：秘密分享 (Secret Sharing)

- **目标**：P_1 **想分享秘密** $x \in \mathbb{Z}_p$ **给** P_1, P_2, P_3.
- **方法**：
  1. P_1 **随机选择** $r_1, r_2 \in \mathbb{Z}_p$.
  2. **计算** $r_3 = x - r_1 - r_2 \bmod p$.
  3. **分发份额 (Shares)**：
     - P_1 **持有**：(r_2, r_3)
     - P_2 **持有**：(r_1, r_3)
     - P_3 **持有**：(r_1, r_2)

核心工具：秘密分享 (Secret Sharing)

- **目标**：P_1 想分享秘密 $x \in \mathbb{Z}_p$ 给 P_1, P_2, P_3.
- **方法**：
  1. P_1 随机选择 $r_1, r_2 \in \mathbb{Z}_p$.
  2. 计算 $r_3 = x - r_1 - r_2 \bmod p$.
  3. **分发份额 (Shares)**：
     - P_1 持有：(r_2, r_3)
     - P_2 持有：(r_1, r_3)
     - P_3 持有：(r_1, r_2)
- **性质**：
  - **重构**：$x = r_1 + r_2 + r_3$.
  - **隐私**：任意单个参与方缺失一个分量，无法恢复 x（信息论安全）.



安全加法协议 (Secure Addition)

目标：计算 $y = \sum x_i$.

安全加法协议 (Secure Addition)

目标：计算 $y = \sum x_i$.

安全加法协议

假设参与方 P_1, P_2, P_3 已将输入 x_i 进行了秘密分享.

whiteboard

安全加法协议 (Secure Addition)

目标：计算 $y = \sum x_i$.

安全加法协议

假设参与方 P_1, P_2, P_3 已将输入 x_i 进行了秘密分享.

1. **本地求和**：每个参与方 P_j 将其持有的关于 x_1, x_2, x_3 的对应份额相加.
   - 例如 P_1 持有 $(r_{i,2}, r_{i,3})$ for $i = 1, 2, 3$.
   - P_1 计算 $s_2 = \sum_i r_{i,2}$ 和 $s_3 = \sum_i r_{i,3}$.

安全加法协议 (Secure Addition)

目标：计算 $y = \sum x_i$.

安全加法协议

假设参与方 P_1, P_2, P_3 已将输入 x_i 进行了秘密分享.

1. **本地求和**：每个参与方 P_j 将其持有的关于 x_1, x_2, x_3 的对应份额相加.
   - 例如 P_1 持有 $(r_{i,2}, r_{i,3})$ for $i = 1, 2, 3$.
   - P_1 计算 $s_2 = \sum_i r_{i,2}$ 和 $s_3 = \sum_i r_{i,3}$.
2. **公布**：
   - P_1 公布 s_2, s_3；P_2 公布 s_1, s_3；P_3 公布 s_1, s_2.

安全加法协议 (Secure Addition)

目标：计算 $y = \sum x_i$.

安全加法协议

假设参与方 P_1, P_2, P_3 已将输入 x_i 进行了秘密分享.

1. **本地求和**：每个参与方 P_j 将其持有的关于 x_1, x_2, x_3 的对应份额相加.
   - 例如 P_1 持有 $(r_{i,2}, r_{i,3})$ for $i = 1, 2, 3$.
   - P_1 计算 $s_2 = \sum_i r_{i,2}$ 和 $s_3 = \sum_i r_{i,3}$.
2. **公布**：
   - P_1 公布 s_2, s_3；P_2 公布 s_1, s_3；P_3 公布 s_1, s_2.
3. **重构**：
   - 各方计算 $y = s_1 + s_2 + s_3 \bmod p$.

安全性证明思路：基于模拟 *(Simulation)*. 给定输出 y 和输入 x_i，可以模拟出所有中间视图.

安全乘法协议 (Secure Multiplication)

目标：计算 $y = a \cdot b \bmod p$.

安全乘法协议 (Secure Multiplication)

目标：**计算** $y = a \cdot b \bmod p$.
已知 $a = \sum a_i, b = \sum b_i$. **展开公式**：

$$a \cdot b = (a_1 + a_2 + a_3)(b_1 + b_2 + b_3) = \sum_{i=1}^{3} \sum_{j=1}^{3} a_i b_j$$

安全乘法协议 (Secure Multiplication)

目标：计算 $y = a \cdot b \bmod p$.
已知 $a = \sum a_i, b = \sum b_i$. **展开公式**：

$$a \cdot b = (a_1 + a_2 + a_3)(b_1 + b_2 + b_3) = \sum_{i=1}^{3} \sum_{j=1}^{3} a_i b_j$$

安全乘法协议

1. **本地计算项**：利用复制秘密分享的特性，每一项 $a_i b_j$ 都可以由某个参与方在本地计算.
   - P_1 (持有下标 2,3)：计算 $u_1 = a_2 b_2 + a_2 b_3 + a_3 b_2$.
   - P_2 (持有下标 1,3)：计算 $u_2 = a_3 b_3 + a_1 b_3 + a_3 b_1$.
   - P_3 (持有下标 1,2)：计算 $u_3 = a_1 b_1 + a_1 b_2 + a_2 b_1$.

安全乘法协议 (Secure Multiplication)

目标：计算 $y = a \cdot b \bmod p$.
已知 $a = \sum a_i, b = \sum b_i$. **展开公式**：

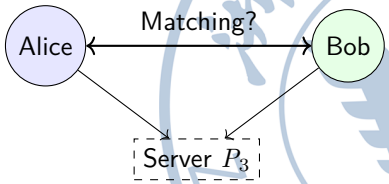
$$a \cdot b = (a_1 + a_2 + a_3)(b_1 + b_2 + b_3) = \sum_{i=1}^{3} \sum_{j=1}^{3} a_i b_j$$

安全乘法协议

1. **本地计算项**：利用复制秘密分享的特性，每一项 $a_i b_j$ 都可以由某个参与方在本地计算.
   - P_1 (持有下标 2,3)：计算 $u_1 = a_2 b_2 + a_2 b_3 + a_3 b_2$.
   - P_2 (持有下标 1,3)：计算 $u_2 = a_3 b_3 + a_1 b_3 + a_3 b_1$.
   - P_3 (持有下标 1,2)：计算 $u_3 = a_1 b_1 + a_1 b_2 + a_2 b_1$.
2. **安全加法**：调用安全加法协议计算 $y = u_1 + u_2 + u_3$.

应用：隐私配对

- **场景**：Alice (a) 和 Bob (b) 互选.
- **输入**：$a, b \in \{0, 1\}$ (1= 感兴趣, 0= 不感兴趣).
- **计算**：$y = a \cdot b$.
  - $y = 1 \implies$ 配对成功.
  - $y = 0 \implies$ 至少一方不感兴趣（但不暴露是谁）.
- **角色**：引入服务器作为 P_3 协助计算（但不获取隐私）.



如果参与方作恶怎么办？(Malicious Model)

1. 输入替换 (Input Substitution)

- **行为**：Alice 总是输入 $a = 1$ 来试探 Bob.
- **结论**：这是 **无法通过协议防止的**. 只能通过博弈论或外部机制解决.

如果参与方作恶怎么办？(Malicious Model)

1. 输入替换 (Input Substitution)

- **行为**：Alice 总是输入 $a = 1$ 来试探 Bob.
- **结论**：这是 **无法通过协议防止的**. 只能通过博弈论或外部机制解决.

2. 违背协议 (Protocol Deviation)

- **行为**：安全加法协议中，发送错误的份额，或者计算错误的结果.
- **解决方案**：**一致性检查 (Consistency Check)**.
  - 因为是复制秘密分享，每份数据都有两人持有.
  - P_2 和 P_3 可以互相验证来自 P_1 的数据是否一致.
  - 这种增强后的模型称为 **恶意安全 (Malicious Security)**.

通用解决方案

通用性:

- $\mathbb{Z}_p$ 上的加法 $+$ 乘法 $=$ **完备集**.
- 可以用上一步计算的秘密份额安全地计算下一步.
- 任何函数都可以表示为加法和乘法电路 $\Longrightarrow$ 我们可以安全计算任何函数!

通用解决方案

通用性:

- $\mathbb{Z}_p$ 上的加法 $+$ 乘法 $=$ **完备集**.
- 可以用上一步计算的秘密份额安全地计算下一步.
- 任何函数都可以表示为加法和乘法电路 $\Longrightarrow$ 我们可以安全计算任何函数!

局限性:

- 安全乘法协议不能检测违背协议的行为;
- 只考虑了三个参与方的情况;
- 假设不会有两个参与方串通;
- 假设参与方可以安全地通信;
- ……

全书概览

本书后续章节安排:

- **Ch 2**: 密码学基础.
- **Ch 4**: MPC 形式化定义与模型 (UC 框架).
- **Ch 5**: 基础原语：茫然传输 (Oblivious Transfer, OT).
- **Ch 3, 6-8**: 半诚实安全协议 (线性秘密分享, Beaver Triple, 混淆电路).
- **Ch 9**: 恶意安全协议 (GMW 编译器, LP11, 恶意安全 BGW, BDOZ, SPDZ).

思考题

1. **隐藏函数**：是否可能让参与方在不知道函数 f 的情况下完成计算？(Universal Circuit / FHE?)
2. **随机化函数**：如何安全地计算概率函数（如共同抛硬币）？
3. **其他属性**：除了隐私性和正确性，特定场景下还需要什么特性？(如公平性 Fairness, 可审计性 Auditability).

Q & A