一个安全多方计算协议的例子

基于 Shamir 秘密分享的半诚实安全协议

《安全多方计算——可证明安全视角》第三章

2026 年 1 月 18 日

目录

协议概览与假设

本章目标

构建一个**通用**的安全多方计算协议，实现信息论意义下的**完美隐私性**.

协议概览与假设

本章目标

构建一个**通用**的安全多方计算协议，实现信息论意义下的**完美隐私性**.

基本设定：

- **参与方数量**：n 个参与方 $P_1, \ldots, P_n$.
- **安全门限**：$t < n/2$（诚实占多数）.
- **敌手模型**：半诚实 (Semi-honest) / 被动安全 (Passive Security).
  - 敌手严格遵守协议执行步骤，但会收集信息试图推断隐私.
- **通信**：假设存在点对点安全信道.

Shamir 秘密分享 (Shamir Secret Sharing)

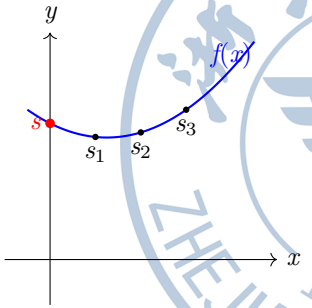
基于拉格朗日插值的阈值方案

分享 (Share):

- 在有限域 $\mathbb{F}$ ($|\mathbb{F}| > n$) 上选定 n 个不同非零点 $\alpha_1, \ldots, \alpha_n$.
- 为分享秘密 s，选择 t 次随机多项式 $f(x)$ 满足 $f(0) = s$.
- 份额 $s_i = f(\alpha_i)$ 发送给 P_i.

重构 (Reconstruct):

- 任意 $t+1$ 个份额可通过**拉格朗日插值**恢复 s.
- 线性重组：$s = f(0) = \sum_{i \in S} \lambda_i s_i$.



秘密分享的线性同态性质

定义符号 $[a; f]_t$ 表示由多项式 f 生成的 a 的份额向量 $(f(\alpha_1), \ldots, f(\alpha_n))$.

引理 (同态计算)

对于任意标量 $c \in \mathbb{F}$ 和份额，以下等式成立：

1. **加法**：$[a; f]_t + [b; g]_t = [a + b; f + g]_t$
2. **标量乘法**：$c \cdot [a; f]_t = [c \cdot a; c \cdot f]_t$
3. **乘法（升维）**：$[a; f]_t * [b; g]_t = [a \cdot b; f \cdot g]_{2t}$

注意：直接乘法会导致多项式阶数变为 $2t$，需要 $2t + 1$ 个份额才能恢复，且不满足 t 隐私性，需特殊处理.

算术电路计算协议

我们将函数表示为由 **加法门**和 **乘法门**组成的算术电路.

算术电路计算协议

我们将函数表示为由 **加法门**和 **乘法门**组成的算术电路.

Phase 1: 输入分享 (Input Sharing)

每个参与方 P_i 将其输入 x_i 生成份额 $[x_i]_t$ 并分发给其他人.

算术电路计算协议

我们将函数表示为由 **加法门**和 **乘法门**组成的算术电路.

Phase 1: 输入分享 (Input Sharing)

每个参与方 P_i 将其输入 x_i 生成份额 $[x_i]_t$ 并分发给其他人.

Phase 2: 电路计算 (Circuit Evaluation)

按拓扑顺序处理每个门:

- **加法门** $(x+y)$: 各方本地计算份额之和.
- **常数乘法门** $(c \cdot x)$: 各方本地乘以常数 c.
- **乘法门** $(x \cdot y)$: 需要交互 (见下页).

算术电路计算协议

我们将函数表示为由 **加法门**和 **乘法门**组成的算术电路.

Phase 1: 输入分享 (Input Sharing)

每个参与方 P_i 将其输入 x_i 生成份额 $[x_i]_t$ 并分发给其他人.

Phase 2: 电路计算 (Circuit Evaluation)

按拓扑顺序处理每个门:

- **加法门** $(x+y)$: 各方本地计算份额之和.
- **常数乘法门** $(c \cdot x)$: 各方本地乘以常数 c.
- **乘法门** $(x \cdot y)$: 需要交互 (见下页).

Phase 3: 输出重构 (Output Reconstruction)

对于输出导线 y, 各方将份额发送给指定的接收方以恢复结果.

核心难点：乘法门 (Multiplication Gate)

目标：给定 $[a; f_a]_t$ 和 $[b; f_b]_t$，计算 $[a \cdot b]_t$.

核心难点：乘法门 (Multiplication Gate)

目标：给定 $[a; f_a]_t$ 和 $[b; f_b]_t$，计算 $[a \cdot b]_t$.

1. **本地乘法**：各方计算 $v_i = f_a(\alpha_i) \cdot f_b(\alpha_i)$.
   - 此时 v_i 是多项式 $h(x) = f_a(x) f_b(x)$ 上的点.
   - $h(0) = ab$，但 $\deg(h) = 2t$（阶数过高）.

核心难点：乘法门 (Multiplication Gate)

目标：给定 $[a; f_a]_t$ 和 $[b; f_b]_t$，计算 $[a \cdot b]_t$.

1. **本地乘法**：各方计算 $v_i = f_a(\alpha_i) \cdot f_b(\alpha_i)$.
   - 此时 v_i 是多项式 $h(x) = f_a(x) f_b(x)$ 上的点.
   - $h(0) = ab$，但 $\deg(h) = 2t$（阶数过高）.
2. **重分享 (Resharing/Degree Reduction)**：
   - 每个参与方 P_i 将自己的 v_i 作为秘密，再次进行 Shamir 分享，得到 $[v_i; g_i]_t$.

核心难点：乘法门 (Multiplication Gate)

目标：给定 $[a; f_a]_t$ 和 $[b; f_b]_t$，计算 $[a \cdot b]_t$.

1. **本地乘法**：各方计算 $v_i = f_a(\alpha_i) \cdot f_b(\alpha_i)$.
   - 此时 v_i 是多项式 $h(x) = f_a(x) f_b(x)$ 上的点.
   - $h(0) = ab$，但 $\deg(h) = 2t$（阶数过高）.
2. **重分享 (Resharing/Degree Reduction)**：
   - 每个参与方 P_i 将自己的 v_i 作为秘密，再次进行 Shamir 分享，得到 $[v_i; g_i]_t$.
3. **线性重组**：利用重组向量 $\boldsymbol{r} = (r_1, \ldots, r_n)$（满足 $h(0) = \sum r_i h(\alpha_i)$）. 各方计算：

$$[ab]_t = \sum_{i=1}^{n} r_i \cdot [v_i; g_i]_t$$

安全性分析

- **正确性 (Correctness)**:
  - 加法和常数乘法由同态性直接保证.
  - 乘法协议利用 $h(0) = \sum r_i h(\alpha_i)$ 确保了结果是 ab，且最终多项式阶数为 t.

安全性分析

- **正确性 (Correctness)**:
  - 加法和常数乘法由同态性直接保证.
  - 乘法协议利用 $h(0) = \sum r_i h(\alpha_i)$ 确保了结果是 ab，且最终多项式阶数为 t.
- **完美隐私性 (Perfect Privacy)**:
  - **证明思路**：基于模拟 (Simulation). 构造模拟器 $\mathcal{S}$，仅通过被攻陷方输入模拟敌手视图.
  - **关键点**:
    1. 输入分享阶段，份额是随机多项式的值（均匀分布）.
    2. 乘法重分享阶段，发送的也是随机多项式的份额.
    3. 输出阶段，虽然能恢复结果，但由于 $t < n/2$，任何 t 个份额能与任意可能结果兼容.

隐私性证明

- 被攻陷方从诚实方接收的信息有两类：
  1. 在输入分享阶段和乘法门中，他们接收诚实方计算的秘密份额集 $[x_i; f_{x_i}]_t$ 和 $[h(\alpha_i); f_i]_t$ 中自己的份额.
  2. 在输出重建阶段中，对于每个被攻陷方的输出 y，他们接收 $[y; f_y]_t$ 中的所有份额.

隐私性证明

- 被攻陷方从诚实方接收的信息有两类：
  1. 在输入分享阶段和乘法门中，他们接收诚实方计算的秘密份额集 $[x_i; f_{x_i}]_t$ 和 $[h(\alpha_i); f_i]_t$ 中自己的份额.
  2. 在输出重建阶段中，对于每个被攻陷方的输出 y，他们接收 $[y; f_y]_t$ 中的所有份额.
- 这些信息不会破坏隐私性，因为：

隐私性证明

- 被攻陷方从诚实方接收的信息有两类：
  1. 在输入分享阶段和乘法门中，他们接收诚实方计算的秘密份额集 $[x_i; f_{x_i}]_t$ 和 $[h(\alpha_i); f_i]_t$ 中自己的份额.
  2. 在输出重建阶段中，对于每个被攻陷方的输出 y，他们接收 $[y; f_y]_t$ 中的所有份额.
- 这些信息不会破坏隐私性，因为：
  - 第一类：阶数最多为 t 的随机多项式的 t 个份额只是均匀随机值.

隐私性证明

- 被攻陷方从诚实方接收的信息有两类：
  1. 在输入分享阶段和乘法门中，他们接收诚实方计算的秘密份额集 $[x_i; f_{x_i}]_t$ 和 $[h(\alpha_i); f_i]_t$ 中自己的份额.
  2. 在输出重建阶段中，对于每个被攻陷方的输出 y，他们接收 $[y; f_y]_t$ 中的所有份额.
- 这些信息不会破坏隐私性，因为：
  - 第一类：阶数最多为 t 的随机多项式的 t 个份额只是均匀随机值.
  - 第二类：给定被攻陷方已知的值和输出 y，被攻陷方自己就能计算出来.

隐私性证明

- 被攻陷方从诚实方接收的信息有两类：
  1. 在输入分享阶段和乘法门中，他们接收诚实方计算的秘密份额集 $[x_i; f_{x_i}]_t$ 和 $[h(\alpha_i); f_i]_t$ 中自己的份额.
  2. 在输出重建阶段中，对于每个被攻陷方的输出 y，他们接收 $[y; f_y]_t$ 中的所有份额.
- 这些信息不会破坏隐私性，因为：
  - 第一类：阶数最多为 t 的随机多项式的 t 个份额只是均匀随机值.
  - 第二类：给定被攻陷方已知的值和输出 y，被攻陷方自己就能计算出来.
- 模拟器可以：
  - 采样均匀随机值来模拟诚实方发送给被攻陷方的份额.
  - 通过被攻陷方已知的值和输出 y 反过来计算诚实方发送给被攻陷方的份额.

计算示例：加法

假设 $t = 1, n = 3$，计算 $c = a + b$. 敌手控制 P_1.

变量	真实值	P_1	P_2	P_3
a	4	5	6	7
b	4	6	8	10
c	8	11	14	17

表: Case 1: $a = 4, b = 4$

计算示例：加法

假设 $t = 1, n = 3$，计算 $c = a + b$. 敌手控制 P_1.

变量	真实值	P_1	P_2	P_3
a	4	5	6	7
b	4	6	8	10
c	8	11	14	17

表: Case 1: $a = 4, b = 4$

模拟一致性：如果真实情况是 $a = 3, b = 5$，但 P_1 看到的份额不变 $(5, 6, 11)$，这在数学上是可能的吗？

- 存在多项式 $a'(x) = 3 + 2x$ 使得 $a'(1) = 5$.
- 存在多项式 $b'(x) = 5 + x$ 使得 $b'(1) = 6$.
- 此时 $c = 8$ 依然成立.
- $\Longrightarrow$ P_1 无法区分 $4 + 4$ 和 $3 + 5$.

计算示例：加法

假设 $t = 1, n = 3$，计算 $c = a + b$. 敌手控制 P_1.

变量	值	P_1	P_2	P_3
a	4	5	6	7
b	4	6	8	10
c	8	11	14	17

表: Case 1: $a = 4, b = 4$

变量	值	P_1	P_2	P_3
a	3	5	7	9
b	5	6	7	8
c	8	11	14	17

表: Case 2: $a = 3, b = 5$

计算示例：乘法

假设 $t = 1, n = 3$，计算 $d = ab$. 敌手控制 P_1.

变量	值	P_1	P_2	P_3
a	2	**3**	4	5
b	4	**3**	2	1
$d = ab$	8	**9**	8	5
d_1	**9**	**7**	**5**	**3**
d_2	8	**8**	8	8
d_3	5	**6**	7	8
$c = 3d_1 - 3d_2 + d_3$	8	**3**	**−2**	**−7**

变量	值	P_1	P_2	P_3
a	1	**3**	5	7
b	8	**3**	-2	-7
$d = ab$	8	**9**	-10	-49
d_1	**9**	**7**	**5**	**3**
d_2	-10	**8**	26	44
d_3	-49	**6**	61	116
$c = 3d_1 - 3d_2 + d_3$	8	**3**	**−2**	**−7**

思考题

1. **门限限制**：为什么必须要求 $t < n/2$？

思考题

1. **门限限制**：为什么必须要求 $t < n/2$？
   - 提示：乘法过程中多项式阶数升至 $2t$，若 $2t \geq n$，则无法通过 n 个点唯一确定多项式或进行错误纠正.

思考题

1. **门限限制**：为什么必须要求 $t < n/2$？
   - 提示：乘法过程中多项式阶数升至 $2t$，若 $2t \geq n$，则无法通过 n 个点唯一确定多项式或进行错误纠正.
2. **域 vs 环**：本协议能迁移到环 $\mathbb{Z}_{2^k}$ 上吗？

思考题

1. **门限限制**：为什么必须要求 $t < n/2$？
   - 提示：乘法过程中多项式阶数升至 $2t$，若 $2t \geq n$，则无法通过 n 个点唯一确定多项式或进行错误纠正.
2. **域 vs 环**：本协议能迁移到环 $\mathbb{Z}_{2^k}$ 上吗？
   - 提示：拉格朗日插值需要求逆元，环上不一定存在逆元.

Q & A