

翻译问题。

用户:

Now consider a three-party voting scenario. Alice, Bob, and Charlie each have a private vote, $x_1, x_2, x_3 \in \{0, 1\}$ respectively (where 1 is 'Yes' and 0 is 'No'). They want to determine if the 'Yes' vote has a majority (i.e., compute the function that returns 1 if $x_1 + x_2 + x_3 \geq 2$, and 0 otherwise) without revealing their individual votes. Design a secure MPC protocol for this problem. You may need to choose a prime number p (for example, $p = 11$) to use as a modulus for all your calculations. Your algorithm description must include two distinct phases: a **Share Distribution Phase** and a **Computation and Reconstruction Phase**.

翻译问题。

元宝:

现在考虑一个三方投票场景。Alice、Bob 和 Charlie 各自拥有一个私密投票，分别为 $x_1, x_2, x_3 \in \{0, 1\}$ （其中 1 代表“是”，0 代表“否”）。他们希望在不泄露各自选票的情况下，判断“是”的选票是否占多数（即计算一个函数：如果 $x_1 + x_2 + x_3 \geq 2$ 则返回 1，否则返回 0）。请为此问题设计一个安全的多方计算协议。你可以选择一个质数 p （例如 $p=11$ ）作为所有计算的模数。你的算法描述必须包含两个不同的阶段：份额分配阶段和计算与重构阶段。

用户:

什么叫“份额分配阶段”

元宝:

在安全多方计算中，**份额分配阶段**（Sharing Phase 或 Input Sharing Phase）指的是：

- 每个参与者把自己的私密输入（这里就是选票 $x_i \in \{0, 1\}$ ）拆分成若干份（称为“份额” shares）。
- 将这些份额分别发送给其他参与者（或全部参与者），但**每一份份额本身不直接暴露原始数据**。
- 只有把所有份额（或达到一定数量的份额）合在一起，才能还原出原始数据。

这样，在后续的计算阶段，各方只在份额上进行局部计算，不直接接触他人的原始输入，从而保护隐私。

常见的秘密共享方案是 **Shamir 秘密共享**，这里因为我们要计算的是三元多数票，且最终要做加法与比较，用 **加法秘密共享（Additive Secret Sharing）** 更简单。

（注：文档部分内容可能由 AI 生成）