arXiv:2305.02377v1 [cs.CR] 3 May 2023

概率调度下群体协议中的隐私

Talley Amir 和 James Aspnes

耶鲁大学，美国康涅狄格州纽黑文市 06511{
talley.amir,james.aspnes}@yale.edu

摘要。群体协议模型 [2] 为设计和分析资源受限的移动代理之间的分布式算法提供了一个理论框架。尽管原始群体协议模型考虑了匿名性的概念，但并未深入研究隐私问题。然而，如果要将这些算法应用于处理敏感数据的场景（例如传感器网络、物联网设备和无人机），群体协议模型就需要具备时间和空间效率的隐私保护技术。在本研究中，我们提出了若干隐私的形式化定义，其范围从仅保证群体输入向量具有合理可否认性，到提供完整的信息论保证，即代理的输入和输出之外的知识不会影响特定输入向量的概率。随后，我们将这些定义应用于现有协议和新协议。我们证明，[9] 中的余数计算协议（已证明其在对抗性调度下满足输出独立隐私）在概率调度下并非信息论安全的隐私。与此相反，我们提出了一种新算法，并证明该算法在概率调度下能够正确且以信息论安全的隐私方式计算余数。

关键词：移动自组网·群体协议·信息论隐私。

1 引言

将理论上的群体协议模型应用于现实世界系统时，会出现各种问题，其中最关键的问题之一便是如何保护隐私。进一步研究群体协议中的隐私问题，旨在使这些算法更好地适应其试图建模的现实世界系统，例如传感器网络、物联网设备系统以及无人机群，而这些系统都需要处理敏感数据。以往关于隐私保护型群体协议的研究仅考虑对抗性调度，这种研究方式对我们无法获知调度器的交互选择作出了过于宽松的假设，而且仅提供了非常薄弱的标准来满足“隐私”的定义。在本研究中，我们进一步细化了这些定义，考虑了任意调度下各种现实的威胁模型和安全问题。

1.1 相关工作

关于自组织网络中隐私计算的研究分散（双关意义上）在多个学术领域之中。我们将文献综述的范围限定为与本文所研究的理论模型最为密切相关的工作。

群体协议中的隐私在 [9]，中被引入群体协议模型，作者在其中定义了一种称为输出独立隐私的隐私概念，并提供了满足该定义的协议，用于计算半线性谓词。输出独立隐私基本上指出，对于任意输入向量以及在某个参与者处产生特定观测序列的执行，都存在另一个不同的输入向量和执行，使该参与者处产生相同的观测序列。该定义的实用性依赖于对抗性调度，这种调度允许交互调度无限多步地延迟参与者对之间的交互。由于采用对抗性调度，只需存在一次执行即可实现合理的可否认性：参与者没有衡量已流逝时间的指标，也无法估算另一参与者参与了多少次交互。因此，无法利用某个参与者的观测状态推断其输入，因为该参与者在经历多次交互的过程中，其状态可能已经偏离了原始状态。然而，如果调度器采用概率性调度，那么就会产生通过推断群体的交互模式而导致数据泄露的问题。

传感器网络 群体协议旨在对传感器网络进行建模，但有大量关于传感器网络的文献与群体协议模型无关。传感器网络领域中的参与者所具备的能力远超群体协议中的假设；特别是，该领域许多保护隐私的算法都涉及加密，而这需要状态空间随群体规模线性增长。

近年来，通过蓝牙进行病毒接触通知已成为一个热门研究领域 [6, 8]，，由于各地广泛施行有关受保护的健康数据的法律，这类研究必须提供可验证的隐私保障。然而， [6, 8] 中的解决方案需要集中化处理，且存储开销很高。 [4, 5] 研究了密切相关的问题——匿名源检测；然而，这些工作需要超常数状态空间，且只解决这一项任务。无线传感器网络领域的其他研究则探讨了隐私数据聚合，这与我们的研究目标最为相似 [7, 13, 16]。同样，这些工作由于采用同态加密来实现解决方案，因此需要大量计算资源和本地内存。若采用替代方法以避免依赖加密，则需要特定的网络拓扑才能成功 [15]，或者只能计算特定函数 [16]。

尽管这并非一份全面的综述，但这些相关研究表明，无线传感器网络中的隐私研究大多受到网络拓扑的限制，或依赖计算开销巨大的加密。因此，我们旨在为群体协议中的数据聚合开发保护隐私的解决方案，同时牢记该模型的资源限制。

1.2 贡献

在这项工作中，我们研究随机调度模型下群体协议的隐私问题。我们展示了在某些建模假设下，现有隐私定义为何会失效；针对这些情形给出新的精确定义，并提出一种适用于均匀随机调度群体协议模型的新协议，该协议满足这些新的隐私定义。在这项工作中，我们将研究范围限定于计算余数谓词。

2 预备知识

群体协议 $\mathcal{P}$是一个元组（Q、δ、Σ、$\mathcal{I}$、O、$\mathcal{O}$），由状态集Q、转移函数 δ、输入集 Σ、输入函数 $\mathcal{I}$、输出集 O和输出函数 $\mathcal{O}$ [2]组成。协议由一个群体运行，该群体由一组 n 参与者 $\{A_j\}_{j=1}^{n}$ 组成， $\{A_j\}_{j=1}^{n}$ 每个参与者都有一些输入 $i_j \in \Sigma$。在协议开始时，每个参与者通过 $\mathcal{I}$ 将其输入转换为 Q 中的一个状态： $\Sigma \to Q$。在早期的群体协议文献中， $\mathcal{I}$ 始终仅被视为确定性函数；然而，在本文中，我们扩展了该模型，允许 $\mathcal{I}$ 进行随机化。转移函数 δ : $Q^2 \to Q^2$ 规定了参与者两两交互时如何更新其状态。作为表示 δ（q_1、q_2） = （q_1'、q_2'）的简写，我们写作 q_1、$q_2 \to q_1'$、q_2' ，其中 δ 是隐含的。该协议旨在根据群体中参与者的初始输入计算某个函数（其输出属于输出集 O）。参与者的输出值是参与者状态的一个函数，由 $\mathcal{O}$ 确定： $Q \to O$。

参与者输入的集合表示为向量 $I \in \Sigma^n$，其中 I 的每个索引都反映群体中特定参与者的输入。沿用 [9], 的术语，我们将 I 称为输入向量。当状态空间的大小为 $O(1)$ 时，协议无法区分处于相同状态或具有相同输入的两个参与者；因此，我们可能希望引用输入向量 I 中输入值的多重集，记为 multiset(I)。将这些输入转换为 Q 的元素后，群体的全局状态称为配置，并表示为向量 $C \in Q^n$，其中向量的第 i 个条目表示第 i 个参与者的状态。滥用记法，我们说 $\mathcal{I}(I) = \langle \mathcal{I}(i_j) \rangle_{j=1}^{n}$ 是将输入函数 $\mathcal{I}$ 应用于 $I = \langle i_j \rangle_{j=1}^{n}$ 中每个参与者输入后得到的配置。

参与者通过彼此交互来更新状态，这些交互以离散间隔进行，称为步。在每一步中，调度器从群体中选取一个有序参与者对 (A_i, A_j)。为区分有序对中的两个参与者，我们将第一个参与者称为发起方，第二个称为响应者。发生交互时，两个被选中的参与者根据转移函数 δ 更新状态；这可能会改变群体中各状态的数量，从而更新配置。设 $\mathcal{C}$ 为配置空间，即状态空间为 Q、包含 n 个参与者的群体所有可能配置的集合。若存在一系列有序参与者对，使得从 C 出发，按照 δ 更新配置后，配置 $D \in \mathcal{C}$ 可以到达 $C \in \mathcal{C}$ ，则称配置 $D \in \mathcal{C}$ 可通过 δ 从 $C \in \mathcal{C}$ 到达

在这些有序参与者对上执行后，得到的配置为 D [2]。若 D 可从 C 到达，则记作 $C \to D$。调度器无限次选择交互对所产生的无限配置序列称为一次执行。若在步骤 τ，对于每一步 $t > \tau$，每个参与者在 t 的状态输出都与其在 τ 的状态输出相同（即每个参与者的输出都收敛到某个值，此后不再改变），则称协议的一次执行在步骤 τ 收敛。终止的一个更强概念是协议稳定，即到达某个配置 C^* 后，从 C^* 可到达的唯一配置所产生的每个参与者的输出，都与 C^* 中的输出相同。为简化记法，如果对每个 $q_j \in C$， $\mathcal{O}(q_j) = \lambda$，我们就说 $\mathcal{O}(C) = \lambda$ （或者说，该配置的输出为 λ）。

群体的目标是在输入向量 I 上计算某个函数 Φ，这意味着群体最终会稳定到一组配置 $\mathcal{D} \subseteq \mathcal{C}$，对于这些配置，所有 $D \in \mathcal{D}$ 均满足 $\mathcal{O}(D) = \Phi(I)$。我们的研究结果与 [9] 的结果相当；后者表明，能够用 Threshold 和 Remainder 表示的半线性谓词，可以在对抗性调度下实现输出独立隐私。我们的研究聚焦于 Remainder，其在群体协议中的定义如下：

定义 1. 给定正整数 k 和 n、非负整数 $r < k$ 以及输入向量 $I \in \mathbb{Z}_k^n$，当且仅当 $\sum_{j=1}^{n} i_j \equiv r \pmod{k}$ 时，$\text{Remainder}(I) = \textsc{True}$。

调度器决定每一步进行交互的代理对。调度器选择代理对的方式可以是对抗性的，也可以是概率性的。对抗性调度器会按自己的意愿选择每一步进行交互的代理对，但须满足公平性条件。最常用的条件称为强全局公平性，它规定：如果某个配置 C 无限次出现，且 $C \to C'$，那么 C' 也必须无限次出现 [2]。这意味着，如果某个配置有可能出现，那么它最终必定会出现，即使对抗性调度器希望无限期地延迟其出现也是如此。在采用对抗性调度的研究中，可以声称协议最终会稳定到正确答案，但无法说明稳定得有多快。随机调度器或概率调度器则根据代理有序对上的某个固定概率分布（通常为均匀分布），选择彼此交互的代理对。尽管群体协议认为交互是按顺序发生的，但其所建模的系统通常由并行参与交互的代理构成。因此，对并行时间的一种自然估算方式是将交互总数除以 n，因为这大致估算了群体中特定代理发起交互的期望次数。请注意，在采用非均匀随机调度的群体协议中，这种时间概念不再一定适用。

我们的工作关键在于区分智能体状态中外部可见的部分与可隐藏的秘密状态。沿用[1], 中的记号，令 S 为内部状态空间， M 为参与者之间可以发送的消息集合。由于每个参与者都有内部和外部两个状态部分，因此总状态空间就是这些集合的笛卡尔积 $Q = S \times M$。这意味着， δ 实际上是每个参与者根据自身状态和其交互伙伴「收到的消息」在本地计算的函数，

其交互伙伴 $\delta\colon\ S \times M \times M \times \{\text{发起方、响应者}\} \to S \times M$。这种新的映射规定，参与者只能使用收到的消息来更新自身状态，并且无法观察其交互伙伴的状态更新。为方便起见，我们使用原有的简写记法$\langle s_0、\ m_0\rangle、\ \langle s_1、\ m_1\rangle \to \langle s_0'、\ m_0'\rangle、\ \langle s_1'、\ m_1'\rangle$来表示参与者的状态变化，其中，$A_b$的状态更新独立于$s_{1-b}$计算。

3 对抗模型

为了评估观察者在群体协议中能够获知多少私人信息，我们必须定义观察者的性质及其能力。在本研究中，我们将代理输入视为私人信息。我们假设观察者采取参与协议交互的代理这一形式，这意味着它只能像其他代理一样观察群体，即通过参与调度器安排的交互来观察群体。然而，我们并不排除观察者可能拥有比普通诚实代理更强的计算能力这一可能性。

我们假设观察者是半诚实的，也就是说，它必须严格遵守协议规则，但可能会尝试从系统 [12] 中推断出额外信息。因此，观察者只能按照转移函数 δ 的规定，通过与其他参与者交互来获取信息。

由于观察者以群体中一个代理的身份出现，我们可以设想多个对手可能渗透到系统中。不过，我们限制每个观察者不得串通，即除了诚实地参与协议交互之外，不得与网络中的其他对抗节点通信。这是因为，否则我们可以设想观察者伪装成群体中的多个代理，占据系统中的任意比例。尽管本文未对此进行研究，但值得探究在任意网络中，观察者最多能模拟多少比例的代理，同时仍能成功隐藏诚实代理的输入。请注意，将观察者限制为既半诚实又不串通，等价于假设群体中只有一个对抗代理，因为从观察者的角度来看，其他所有代理都表现为诚实的。

最后，我们允许区分外部可见的消息和内部隐藏的状态，如 [1] 所示，以便参与者能够隐藏其部分状态，最终实现隐私。在群体模型中研究隐私时，消息与内部状态之间的区分至关重要，因为若没有这种区分，就无法向观察者隐藏信息。

4 输入隐私的定义

本节中，基于我们所指定的对抗模型，考察对抗调度和概率调度下群体协议的隐私定义。

4.1 输出无关隐私

来自 [9] 的隐私保护群体协议采用对抗调度模型，且使用常数大小的状态空间。因此， [9] 仅展示了计算半线性谓词时的隐私。作者在此情形下给出了输入隐私的形式化定义，称为输出独立隐私，定义如下:

> “当且仅当存在一个常数n_0，使得对于任意代理 p、任意大小至少为 n_0且其中 p 具有相同输入的输入 I_1 和 I_2、任意在输入 I_1上的执行 E_1，以及任意 T，都存在一个在输入 I_2上的执行 E_2，使得在 E_1 和 E_2中，p的交互历史截至 T 完全相同，则称“群体协议具有此性质。

本质上，这一定义指出，半诚实进程 p 无法根据其观察序列判断输入向量是 I_1 还是 I_2，因为在对抗性调度器的调度下，任一输入都可能产生相同的观察结果。

输出独立隐私在 [9] 中是一种有效的措施，因为该研究假设调度由对手控制，因此无法推断交互模式。作者利用这一点实现了隐私，其最佳表述是“合理可否认性”——一个代理可能直接观察到另一个代理的输入，但对手控制的调度器具有不可预测性，因此观察者无法确定地断言所观察到的值确实就是该输入。

当调度器采用概率调度时，这一论证便不再成立，因为此时代理可以推断交互模式的概率分布，进而推断与其交互的伙伴代理的输入值的概率分布。鉴于这一发现，我们现在提出新的定义，用于评估带概率调度的人口协议中的隐私。

4.2 概率调度下的隐私定义

考虑一个初始状态为 $q_0^A = (s_0^A, m_0^A)$ 的代理 A。根据其观察到的消息序列，以及 A 在每次交互中扮演的角色（发起方或响应者）， A 可以确定性地计算其后续的每次状态更新。我们将这些消息（由 A观察到）称为 o_1^A,o_2^A,o_3^A, ...，并将 $q_\varepsilon^A = \delta(\rho_\varepsilon^A, s_{\varepsilon-1}^A, m_{\varepsilon-1}^A, o_\varepsilon^A) = (s_\varepsilon^A, m_\varepsilon^A)$ 记为 A（原本处于状态 $q_{\varepsilon-1}^A = (s_{\varepsilon-1}^A, m_{\varepsilon-1}^A)$）在第 ε 次交互中作为 $\rho_\varepsilon^A \in$ {发起方、响应者} 与另一代理交互，且该代理的消息为 o_ε^A 时更新后的状态。采用[12], 中的记法，我们将代理 A 在执行 E 中参与协议 $\mathcal{P}$ 时的视图记为 $\text{view}_A^{\mathcal{P}}(E) = \langle i_A;\ q_0^A; (\rho_1^A, o_1^A), (\rho_2^A, o_2^A)\ ... \rangle$。此视图包含 A的输入、 A的初始状态，以及 A在执行过程中的交互列表，由此可以计算出 A 的每个后续状态。[1]

令 $\text{view}_A^{\mathcal{P}}(C)$ 为一个随机变量，表示代理 A 的视图：从配置 C 开始、由调度器所使用的所有可能随机性产生的全部可实现执行中均匀抽取。

[1] 对于随机化的 δ，我们假设 A 拥有一条固定的随机比特带，用于更新其状态，因此 A 仍然可以根据指定的信息重建其完整视图。

类似地，令 $\mathrm{view}_{\boldsymbol{A}}^{\mathcal{P}}(\boldsymbol{I})$ 为一个随机变量，表示代理 A 的视图：根据 $\mathcal{I}$的随机性所给出的概率分布，从以任意配置 C 开始的所有可能执行中抽取，其中该配置 C 属于 $\mathcal{I}(I)$ 的值域。一般而言，我们约定随机变量以数学粗体显示。

隐私与许多其他安全相关的关键术语一样，有着广泛的技术解释。因此，我们现在针对群体模型提供几种不同的隐私形式化定义。

合理可否认性 或许我们能够定义的最弱隐私形式是合理可否认性，这意味着对手始终会对其对代理输入值的猜测心存疑虑（即使它拥有无限资源）。这并非新概念 [9, 14], ，但在概率群体协议的输入向量隐私语境下，我们将此概念定义如下：

令 $\mathcal{M}_\lambda = \{$多重集(I): $\Phi(I) = \lambda\}$ 为其对应的输入向量求值为 λ 的所有不同输入多重集的集合，[2] 并令 $\mathcal{M}_\lambda^\kappa =\{$多重集$(I)$: 多重集$(I) \in \mathcal{M}_\lambda \wedge \kappa \in$ 多重集$(I)\}$ 为输出 λ 且其中至少包含一个等于 κ 的输入的所有不同输入多重集的集合。

定义 2. 设 $\mathcal{P}$ 是一个作用于 n 个参与者、输入集为 Σ的群体协议，并设 $\mathcal{D}$ 是 Σ^n中输入向量上的任意概率分布。那么，如果对于 Σ^n上的每个分布 $\mathcal{D}$ 、在大小为 n 的群体中执行 $\mathcal{P}$的每个互不串通的半诚实无界参与者 $\mathcal{A}$ ，以及任意视图 $V = \langle i;\ q;\ \{(\rho_\varepsilon^{\mathcal{A}}, o_\varepsilon^{\mathcal{A}})\}\rangle$（其输出 λ由视图 V确定），且满足 $|\mathcal{M}_\lambda^i| > 1$，都存在 I_1 和 I_2 属于 $\mathcal{S}_\lambda$ ，使得

1. multiset(I_1)和multiset(I_2)都是 $\mathcal{M}_\lambda^i$的元素；2. multiset(I_1) $\neq$ multiset(I_2)；3. $\Pr(\mathrm{view}_{\boldsymbol{\mathcal{A}}}^{\mathcal{P}}(\boldsymbol{I_1}) = V) = \Pr(\mathrm{view}_{\boldsymbol{\mathcal{A}}}^{\mathcal{P}}(\boldsymbol{I_2}) = V)$,

其中，最后一个条件中的概率是针对 $\mathcal{D}$、 $\mathcal{I}$的随机性以及调度器的均匀随机性计算的。

通俗地说，定义 2 表明，参与协议的任何代理都不能简单地猜测“最有可能的”输入向量，因为对于每个这样的输入向量，在某些条件下，都存在另一个不同的输入向量，使得该代理的视图以相同的概率相同。此定义不同于输出独立隐私 [9] ，因为它考虑了依赖于从与其他参与者的交互中收集到的分布数据来猜测输入向量的对抗策略。

条件 $|\mathcal{M}_\lambda^i| > 1$ 意味着，弱隐私只能适用于存在合理可否认性的输入多重集。例如，如果 Or 谓词的计算输出为 0，那么只有一种可能的输入多重集能够产生这一结果，因此无法否认输入向量必定是全零向量。

[2] 请注意，处于相同状态的参与者在协议看来无法区分；因此，Φ 必须将输入多重集相同的任何输入向量映射为相同的输出。

信息论输入隐私 一种更强的隐私概念认为，观察者无法根据其观察结果缩小输入向量的可能范围。这引出了我们的下一个定义。

设 $\mathcal{P}$ 是一个输入集为 Σ 的群体协议，并设 $\mathcal{D}$ 是定义在 Σ^n 中输入向量上的概率分布。设 $\boldsymbol{I} \sim \mathcal{D}$ 是表示所选输入向量的随机变量。此外，设 $\boldsymbol{i_A}$ 和 $\boldsymbol{\lambda_A}$ 分别是表示代理 $\mathcal{A}$ 的输入和输出的随机变量，并设 $\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i}, \boldsymbol{\lambda})$ 是一个随机变量，表示代理 $\mathcal{A}$ 参与 $\mathcal{P}$ 的诚实执行时的视图；该执行与代理 $\mathcal{A}$ 的固定输入 i 以及观测到的输出 λ 相一致。

定义 3. 若对于每个不串谋的半诚实无界代理 $\mathcal{A}$，以及每个输入 $i \in \Sigma$、输出 $\lambda \in O$、视图 V、输入向量 $I \in \mathcal{S}_\lambda$ 和定义在 Σ^n 上的概率分布 $\mathcal{D}$，协议 $\mathcal{P}$ 都满足以下条件，则称该协议满足信息论输入隐私：

$$Pr(\boldsymbol{I} = I \mid \mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V) = Pr(\boldsymbol{I} = I \mid \boldsymbol{i_A} = i, \boldsymbol{\lambda_A} = \lambda),$$

其中， V 与输入 i 和输出 λ 相一致。

上述定义的实质是：在已知自身输入和计算输出的条件下，代理在协议执行中的其余视图并不能帮助其更准确地猜测输入向量。

我们提出了一种独立于主要研究结果的隐私替代定义，称为输入不可区分性，详见附录。

直观而言，很容易看出，信息论隐私是本节所讨论定义中最强的（证明见附录）：

定理 1. 如果 $\mathcal{P}$ 是信息论安全的隐私，那么 $\mathcal{P}$ 也满足输出独立隐私、弱隐私和输入不可区分性。

5 对抗性调度下的私有 `Remainder`

作为比较，我们分析 [9], 中展示的余数协议（算法 1）。该协议不区分内部状态空间和消息空间，因此，每个代理的全部状态都对与其交互的伙伴可见。代理状态是元组 (v, f)，其中 v 是代理的数值，而 f 是一个标志位，用于表示代理是否已经确定其输出。该协议通过在单次交互中以单位为增量而非一次性传递完整数值的方式转移数值，从而累积所有代理输入的总和（模 k）。如 (M1) 所示，该协议从一个输入中减去 1，并将其加到另一个输入上，从而保持不变量：在每一步中，群体中所有数值的总和保持不变。由于所有计算均在模 k 下进行，(M1) 可以无限次重复。转移 (M2) 和 (M3) 处理标志位，确保 (M1) 发生的次数无界但有限。

算法 1 满足输出独立隐私的证明关键在于转移 (M1)。当对抗进程 p 与处于状态 (v, f) 的诚实代理 A 交互时， p 无法得知 v 与 A 的原始输入相差多少

算法 1：输出独立的私有余数 [9]	
$(v_1, 1), (v_2, 1) \to (v_1 + 1, 1), (v_2 - 1, 1)$	(M1)
$(*, 1), (*, *) \to (*, 0), (*, *)$	(M2)
$(*, 0), (*, 1) \to (*, 1), (*, 1)$	(M3)
$(v_1, 0), (v_2, 0) \to (v_1 + v_2, 0), (0, 0)$	(M4)
$(v_1, 0), (0, 0) \to (v_1, 0), (\perp_0, 0)$	(M5)
$(\perp_i, *), (*, 1) \to (0, 0), (*, 1)$	(M6)
$(r, 0), (\perp_i, 0) \to (r, 0), (\perp_1, 0)$	(M7)
$(v_1,0), (\perp_i,0) \to (v_1, 0), (\perp_0, 0)$，如果 $v_1 \neq r$	(M8)

每个代理 A_j 的初始状态为 $(i_j,1)$。运算以 k 为模进行，$*$ 是可匹配任意值的通配符。输出值为 $\{\perp_0, \perp_1\}$，分别表示谓词为 False 或 True。当除一个代理外的所有代理的状态均为 $\perp_0$ 或 $\perp_1$ 时，协议收敛。

因为对于 $n \geq 3$，我们可以构造多种执行，使得 A 与 p 交互后具有值 v。例如，我们可以构造这样一种执行：某个代理 B 通过 (M1) 向 A 转移所需数量的单位，使 A 的值变为 v；只要 p 和 B 在 p 与 A 交互之前没有相互交互，p 在这次执行中的视图就相同。

然而，输出独立隐私无法成功延伸至随机调度模型，因为我们再也无法构造任何能够“欺骗”进程 p的执行，这是因为某些此类执行发生的概率极低。例如，参与者 A 和 B 连续交互 v' 次，而在此期间 p 完全不与 B 交互的概率，会随着v'的值增大而变小。这意味着，在执行初期，代理的值偏离其原始输入值的可能性较低。关于该算法不具备隐私性的形式化证明见附录。

6 概率调度下的私有余数

本节将介绍一种新颖的算法，用于在概率调度的群体协议模型中以信息论安全的方式私下计算余数。我们的算法受到环形网络中著名的余数密码学安全多方计算示例的启发。我们将该算法称为 RingRemainder，其工作方式如下：

有 n 个参与者 A_1、...、A_n，排列成一个圆。参与者 A_1 担任领导者，其职责是将一个均匀随机元素 $r \in \mathbb{Z}_k$ 加到其输入上，并将总和（模 k）传递给参与者 A_2。对于其余每个参与者 A_i，在从 A_{i-1}接收到一个值后，A_i 将其自身的输入加到该值上，并将所得总和传递给 A_{i+1}（模 n）。当 A_1 从 A_n接收到一个值后，它减去 r，并将结果广播给所有人。假设参与者的输入为 i_1、...、i_n。

然后， A_1 向 A_2发送 $m_1 = i_1 + r$， A_2 向 A_3发送 $m_2 = i_1 + r + i_2$，依此类推，直到 A_n 向 A_1发送 $m_n = r + \sum_{j=1}^{n} i_j$。因此，广播给所有参与者$m_n - r$ 的值恰好等于 $\sum_{j=1}^{n} i_j$，也就是参与者输入之和模 k。假设参与者诚实且使用安全成对通信，该协议可实现信息论输入隐私（证明见附录）。

现在，我们对这一方案进行调整，使其能够在具有信息论隐私的群体模型中计算余数。

算法概述 我们的协议模拟信息传递，方式与 RingRemainder 完全一致。我们假设协议有一个初始领导者，持有一个特殊令牌，该令牌会在群体中传递。每次参与者收到令牌和随附的某个值时，都会将自己的输入加到该值上，然后将总和连同令牌一起传给另一名参与者。这意味着，令牌当前的持有者掌握着此前持有过令牌的参与者输入的汇总和。当参与者将令牌传给另一名参与者时，会将自己标记为“已访问”，以确保其输入恰好被计入总和一次。令牌访问过所有参与者后，便会连同所有参与者输入的总和一起返回给领导者。为了实现这一功能，我们必须克服两个关键障碍：

首先，我们需要一种机制，以便在两个参与者之间安全地传输消息，使得除发送方和预期接收方之外，其他参与者都无法获知该消息。这项任务并非易事，因为群体协议不允许参与者在交互中传输消息之前验证某个条件；我们假设消息交换和状态更新是瞬时发生的。为此，我们在第 6.1 节提供了一个安全点对点传输子程序。

其次，我们需要一种方法来确定群体中的每个参与者是否都已被令牌访问。当这种情况发生时，我们希望最后一个持有令牌的参与者将令牌传回给领导者，以便领导者能够移除最初添加到聚合值中的随机性，而该聚合值此前一直在参与者之间传递。我们必须尽量防止在所有输入都被纳入聚合值之前将其传回领导者，否则会导致某些参与者被排除在计算之外。为了实现这一点，我们使用来自 [3] 的探测协议，并将在第 6.2 节中对其进行更详细的介绍。

利用这两个子程序，我们在第 6.3 节设计了用于计算余数并实现信息论隐私的主算法。

6.1 安全的点对点传输

为了使我们的算法能够保证输入隐私，任意两个参与者之间中间和的通信都必须保持安全。在此，我们介绍一种新颖的安全点对点传输协议，定义如下：

定义 4。设 M 为消息空间， $\mathcal{D}$ 为 M 上的某个分布，且 I 为 Σ^n 中的任意固定输入向量。一个安全点对点传输例程

算法 2：安全点对点传输群体协议	
$\langle \mu,\ (r, \mathfrak{S})\rangle,\ \langle *,\ (*, \overline{\mathfrak{u}})\rangle \to \langle \mu,\ (r', \mathfrak{S})\rangle,\ \langle *,\ (*, \overline{\mathfrak{u}})\rangle$	(S1)
$\langle \mu, (r, \mathfrak{S})\rangle, \langle *, (*, \mathfrak{u})\rangle \to \langle \bot, (\mu - r, \mathfrak{S}')\rangle, \langle r, (*, \mathfrak{R})\rangle$	(S2)
$\langle \bot, (x, \mathfrak{S}')\rangle, \langle y, (*, \mathfrak{R})\rangle \to \langle \bot, (\bot, \overline{\mathfrak{u}})\rangle, \langle x + y, (*, \mathfrak{S})\rangle$	(S3)

安全点对点传输协议。

是一种协议 $\mathcal{P}$，用于将数据 $m \xleftarrow{\mathcal{D}} M$ 从一个代理发送方传输给另一个接收方，并且存在概率多项式时间算法 W_1、W_2，其中

$$\Pr\left(W_1(\mathbf{view}^{\mathcal{P}}_{\mathit{Sender}}(\boldsymbol{I})) = m\right) = \Pr\left(W_2(\mathbf{view}^{\mathcal{P}}_{\mathit{Receiver}}(\boldsymbol{I})) = m\right) = 1$$

并且对于所有 i： $A_i \notin$ {发送方、接收方} 以及 PPT算法 W'

$$\Pr\left(W'(\mathbf{view}^{\mathcal{P}}_{\boldsymbol{A_i}}(\boldsymbol{I})) = m\right) = \Pr(m \xleftarrow{\mathcal{D}} M)$$

换言之，安全点对点传输例程允许发送方将消息 m 传输给接收方，使得只有发送方和接收方知晓 m，而所有其他参与者都无法在仅知道消息空间上的先验分布的基础上获得任何猜测 m 的优势。

我们的算法 2 满足此定义：每个参与者的状态 $\langle \mu,\ (r, L)\rangle$ 由一个隐藏秘密 μ、一个公开随机数值 r 和标签 L组成。该协议的目标是将一条秘密消息从一个参与者（标记为发送方，标签为$\mathfrak{S}$；群体中最多只有一个这样的参与者）传递给另一个符合特定条件的参与者，其标签为 $\mathfrak{u}$；这样的参与者数量不限（包括零个）。在发送方遇到标签为 $\mathfrak{u}$的参与者之前，它会在每次交互时刷新自己的随机性，以确保传递给接收方的随机性是均匀随机的（S1）。当发送方最终遇到某个标签为 $\mathfrak{u}$的参与者时，它会将该参与者标记为接收方，并传递刚生成的随机数值 r；它还会将自己的令牌更新为 $\mathfrak{S}'$，以记住自己已遇到并标记了一名接收方（S2）。随后，发送方等待再次遇到接收方；此时，它会将一条用前一次交互中发送的随机性掩码的消息交给接收方，并将自己的标签设为 $\overline{\mathfrak{u}}$，表示传输结束（S3）。协议结束时，恰好有一个参与者被选为接收方，并在内部存储着 μ。该协议的状态空间为 $(\mathbb{Z}_k \cup \{\bot\})^2 \times \{\mathfrak{S}, \mathfrak{S}', \mathfrak{R}, \mathfrak{u}, \overline{\mathfrak{u}}\}$；当 k 为常数时，其大小为 $O(1)$。我们在附录中证明算法 2 是一个安全点对点传输例程：

定理 2。算法 2 是一种安全点对点传输例程。

6.2 探测协议

为了将 RingRemainder 适配到群体协议模型，我们需要一种方法来检测聚合是否已涵盖每个参与者，以便将最终总和传回领导者。为此，我们使用探针。

探测协议（或称探针）是一种群体协议，用于检测群体中是否存在满足给定谓词 [3]的参与者。本质上，探针（由领导者启动）会在状态为 0 的参与者群体中发送一个 1 信号。如果 1 信号到达满足该谓词的参与者，该参与者便会启动一个 2 信号，并通过流行传播将其传回领导者。较高编号的流行传播会覆盖较低编号的流行传播，因此，如果群体中有参与者满足 π，领导者最终就会收到 2 信号。探针与同一项研究 [3], 中的相位时钟配合使用，可使领导者在 O 中通过 $O(1)$ 个状态，以 $1-n^{-c}$ 的概率在 $n\log\ n$ 次交互内检测到满足 π 的参与者，其中 $c>0$ 为任意固定常数（见附录）。

我们将协议的“输出”（仅由领导者计算）定义为：状态 0 和 1 对应 0，状态 2 对应 1（也就是说，当且仅当群体中存在某个参与者满足 π 时，领导者的探针才输出 1）。在相位时钟每轮开始时，参与者会将自己的值重置为 0，而领导者会启动一个新的探针。探针状态和相位时钟状态都是消息空间的组成部分，并且这些子程序的转移独立于主协议的转移，因此我们认为这两个“协议”是并行运行的。

6.3 具有信息论隐私的余数

我们在此提出一种新颖的算法，该算法能够计算余数，并在均匀随机调度器的假设下，以高概率在群体协议模型中实现信息论输入隐私。

首先，每个参与者按如下方式将输入函数 $\mathcal{I}$ 应用于自己的输入：

$$\mathcal{I}(i_j,\ell)=\begin{cases}\langle i_j+r^0,(r^j,\mathfrak{S},1,Z=Z_0)\rangle & \ell=1\\ \langle i_j,(r^j,\mathfrak{u},0,Z=Z_0)\rangle & \ell=0\end{cases}$$

其中，r^j 是从 $\mathbb{Z}_k$ 中为 $j\in\{0,1,...,n\}$均匀随机抽取的，而 Z（初始化为 Z_0）是一个探针子程序（包括其关联的相位时钟）。输入函数假定存在一个由 $\ell=1$ 指定的初始领导者。状态 $\langle\mu$、（r、L、ℓ、Z）$\rangle$ 的组成部分分别是 μ（状态中称为秘密的隐藏内部组成部分）、r（掩码）、L（个体标签）、ℓ（领导者位）和 Z（探针）。描述该协议的转移见算法 3。

算法 2 中安全点对点传输协议的转移一般结构用于发送 (R1)、(R2) 和 (R3) 中的中间和。但是，接收方并非只存储收到的消息，而是将该消息与自身输入相加，并在内部存储结果。之后的每个发送方都会在群体中寻找一个输入尚未计入总和的代理（由 $\mathfrak{u}$ 状态表示）。当群体中不再有任何代理拥有 $\mathfrak{u}$ 时，探针便会检测到这一点，并从此开始在领导者处输出 1。

当探针开始输出 1 时，以高概率，每个参与者的标签都会被设为 $\overline{\mathfrak{u}}$，从而提醒领导者将其标签设为 $\mathfrak{u}$。这样一来，只有领导者能够成为下一个接收方。当领导者收到存储在发送方处的最终值后，便可将答案放入外部状态的一个单独区域

算法 3：信息论安全余数

$$\langle *,\ (r, \mathfrak{S}, *, *)\rangle,\ \langle *,\ (*, \overline{\mathfrak{u}}, *, *)\rangle \ \to \langle *,\ (r', \mathfrak{S}, *, *)\rangle,\ \langle *,\ (*, \overline{\mathfrak{u}}, *, *)\rangle \tag{R1}$$

$$\langle u,\ (r, \mathfrak{S}, *, *)\rangle,\ \langle v,\ (*, \mathfrak{u}, *, *)\rangle \to \langle \bot,\ (u - r, \mathfrak{S}', *, *)\rangle,\ \langle v + r,\ (*, \mathfrak{R}, *, *)\rangle \tag{R2}$$

$$\langle *,\ (x, \mathfrak{S}', *, *)\rangle,\ \langle y,\ (*, \mathfrak{R}, *, *)\rangle \to \langle *, (\bot, \overline{\mathfrak{u}}, *, *)\rangle, \langle x + y, (*, \mathfrak{S}, *, *)\rangle \tag{R3}$$

$$\langle \bot, (\bot, \overline{\mathfrak{u}}, 1, 1)\rangle, \langle *, (*, *, *, *)\rangle \ \to \langle \bot, (\bot, \mathfrak{u}, 1, 1)\rangle, \langle *, (*, *, *, *)\rangle \tag{R4}$$

虽然未显示，但每次交互还会通过推进相位时钟以及交互双方代理处的探针，来更新探测子程序。转移（R1-R3）与算法 2 中的转移（S1-S3）类似。

（算法 3 中未显示），以便所有其他参与者都能复制该答案；这以高概率需要额外的 $O(n^2 \log n)$ 步。领导者的隐藏状态中还必须有一个额外组件，用于存储其在初始消息传递中使用的随机性（算法 3 中也未显示）。

算法 3 的正确性陈述如下，并在附录中证明：

定理 3。对于任意固定的 $c > 0$，算法 3 以至少 $1 - n^{-c}$ 的概率，在规模为 n 的群体中用 $\Theta(n^3 \log n)$ 步计算余数。

最后，我们在最终定理中总结算法 3 的隐私保证，并将证明留待附录给出：

定理 4. 当算法 3 正确计算 *Remainder*谓词时，它满足信息论输入隐私。

如果协议因探测子程序中的相位时钟错误而失败，我们实际上并不知道该协议泄露了多少信息，不过我们怀疑泄露量有限。我们将此问题视为超出本文讨论范围，并且只在协议成功时对隐私作出保证。请注意，在异步分布式系统中，不可能以概率 1 实现信息论隐私，因为参与者可能会被无限期排除在协议之外，从而导致提前终止。

结论在本研究中，我们提出了群体协议中的多种新安全定义，例如适用于各种威胁模型和调度假设的多种隐私定义，以及安全点对点通信的形式化定义。我们还开发了用于在该模型中实现安全成对通信，以及对余数谓词进行信息论安全的隐私计算的算法。为了证明我们能够像 [9], 中类似的用于计算 Threshold 和 Or 的算法一样，以高概率实现所有半线性谓词的信息论隐私保护，我们还需要开发相应的算法。我们将这些问题留作未来研究的开放问题。

参考文献

[1]Amir, T., Aspnes, J., Doty, D., Eftekhari, M., Severson, E.：群体协议的消息复杂度。载于：第34届国际分布式计算研讨会（DISC 2020）。莱布尼茨国际信息学会议录（LIPIcs），第179卷，第6:1–6:18页。莱布尼茨信息学中心，Dagstuhl，德国（2020）。https://doi.org/10.4230/LIPIcs.DISC.2020.6[2] Angluin, D., Aspnes, J., Diamadi, Z., Fischer, M.J., Peralta, R.：被动移动有限状态传感器网络中的计算。《ACM分布式计算原理年度研讨会论文集》18，235–253（2006年3月）。https://doi.org/10.1007/s00446-005-0138-3[3] Angluin, D., Aspnes, J., Eisenstat, D.：有领导者的群体协议快速计算。《分布式计算》21，183–199（2006年1月）。https://doi.org/10.1007/s00446-008-0067-z[4] Aspnes, J., Diamadi, Z., Gjøsteen, K., Peralta, R., Yampolskiy, A.：悄然传播警报与子群逃逸问题。载于：密码学进展——ASIACRYPT 2005。第253–272页。施普林格柏林海德堡，柏林，海德堡（2005）[5]Blazy, O., Chevalier, C.：悄然传播警报：来自理论与实践的新见解。载于：第13届国际可用性、可靠性与安全会议论文集。ARES 2018，计算机协会，纽约，美国（2018）。https://doi.org/10.1145/3230833.3230841[6] Canetti, R., Kalai, Y.T., Lysyanskaya, A., Rivest, R., Shamir, A., Shen, E., Trachtenberg, A., Varia, M., Weitzner, D.J.：保护隐私的自动暴露通知。IACR密码学电子印本档案2020，863（2020）[7] Castelluccia, C., Mykletun, E., Tsudik, G.：无线传感器网络中加密数据的高效聚合。载于：第二届移动与泛在系统：网络与服务年度国际会议。第109–117页（2005）[8] Chan, J., Foster, D., Gollakota, S., Horvitz, E., Jaeger, J., Kakade, S., Kohno, T., Langford, J., Larson, J., Sharma, P., Singanamalla, S., Sunshine, J., Tessaro, S.：Pact：用于移动接触者追踪的隐私敏感协议与机制（2020）[9] Delporte-Gallet, C., Fauconnier, H., Guerraoui, R., Ruppert, E.：隐秘的鸟：群体协议中的隐私。载于：分布式系统原理。第329–342页。施普林格柏林海德堡，柏林，海德堡（2007）[10]Goldwasser, S., Micali, S., Rackoff, C.：交互式证明系统的知识复杂度。载于：第十七届ACM计算理论年度研讨会论文集。第291–304页。STOC '85，计算机协会，纽约，美国（1985）。https://doi.org/10.1145/22145.22178

[11] Katz, J., Lindell, Y.：《现代密码学导论》，第二版。Chapman & Hall/CRC，第 2 版（2014）[12]Lindell, Y.：《如何模拟：模拟证明技术教程》，第 277–346 页。施普林格国际出版，瑞士沙姆（2017）。https://doi.org/10.1007/978-3-319-57048-8 6[13] Liu, C.X., Liu, Y., Zhang, Z.J., Cheng, Z.Y.：适用于无线传感器网络的高能效、保护隐私的安全数据聚合。《国际通信系统期刊》26(3)，380–394（2013）。https://doi.org/10.1002/dac.2412[14] Monshizadeh, N., Tabuada, P.：合理可否认性作为一种隐私概念。载于：2019 年 IEEE 第 58 届决策与控制会议（CDC）。第 1710–1715 页（2019）。https://doi.org/10.1109/CDC40024.2019.9030201[15]Setia, P.K., Tillem, G., Erkin, Z.：去中心化网络中的隐私数据聚合。载于：2019 年第 7 届伊斯坦布尔智能电网与城市大会暨展览会（ICSG）。第 76–80 页（2019）。https://doi.org/10.1109/SGCF.2019.8782377[16]Taban, G., Gligor, V.D.：传感器网络中保护隐私且确保完整性的数据聚合。载于：2009 年国际计算科学与工程会议论文集——第 03 卷。第 168–175 页。CSE '09，美国电气与电子工程师学会计算机学会（2009）。https://doi.org/10.1109/CSE.2009.389

A 第4节输入隐私定义的补充材料

A.1 隐私的另一种定义

计算不可区分性 该定义借鉴了经典密码学中关于零知识证明的概念 [10], ，这些概念依赖于所谓的模拟范式。其基本思想是，协议有两个版本：“真实”版本严格按照预期执行；而“理想”版本中，参与者没有输入，而是仅根据协议的输出随机选择输入。如果观察者（例如群体中的某个参与者）无法区分这两种情形，那么它必然无法从其他参与者的输入中获知更多信息，只能获知其通过已有信息所能得到的内容（即自己的输入和计算的输出）；这是因为在后一种情形中，输入完全是随机选择的，同时协议输出保持不变。

宽泛地说，群体模型中的“不可区分性”意味着，参与协议的计算能力受限的代理无法获知其他任何代理的秘密输入的更多信息；这是因为它无法区分协议的“真实”执行与“理想”执行，而它从自身输入和协议输出中获得的信息除外。

为了对此进行形式化定义，我们首先介绍一些其他概念：

定义5。如果对于每个多项式 $p(\cdot)$，都存在一个 N，使得对于所有 $n > N$，均有 $f(n) < p(n)$，则函数 f 是可忽略的 *[11]* 。

换句话说，如果 f 的增长速度比任何多项式函数都慢，我们就称其为可忽略的。

定义 6. 若样本空间 Λ 上的两个分布 $\mathcal{X}$ 和 $\mathcal{Y}$ 在计算上不可区分，则对每个概率多项式时间（PPT）对手 $\mathcal{A}$ ，若其能够预言机访问来自这两个分布之一的多项式（关于 w）个样本实例，就存在一个可忽略函数 ϵ，使得

$$|\Pr(\mathcal{A}^{\mathcal{X}(\cdot)}(w) = 0) - \Pr(\mathcal{A}^{\mathcal{Y}(\cdot)}(w) = 0)| \leq \epsilon(w)$$

也就是说，当只能访问来自其中一个分布的多项式数量的样本时，不存在能够可靠地识别这两个分布中哪一个的 PPT 算法。若 $\mathcal{X}$ 和 $\mathcal{Y}$ 在计算上不可区分，则记作 $\mathcal{X} \equiv \mathcal{Y}$。

回忆一下，执行是一个无限的配置序列。因此，我们将 A 截至某个固定步骤 τ 的视图记为：$\text{view}_A^{\mathcal{P}}(E)^\tau = \langle i_A;\ q_0^A;\ (\rho_1^A, o_1^A),\ (\rho_2^A, o_2^A) \ldots\ldots,\ (\rho_{\alpha_A}^A, o_{\alpha_A}^A) \rangle$，其中 α_A 是 A 从执行开始到执行的某个固定步骤 τ 为止参与的交互次数。

现在，我们在群体模型中定义一种与计算不可区分性相关的隐私概念：

定义 7. 协议 $\mathcal{P}$ 满足输入不可区分性，当且仅当对于每个半诚实且不串通的 PPT 代理 $\mathcal{A}$ 和任意多项式 $\tau(n)$，都存在一个模拟器 S，使得对于每个可能的输出 λ 以及每个 $I \in \mathcal{S}_\lambda$，

$$\{S(n, i_{\mathcal{A}}, \lambda)\} \equiv \{\mathbf{view}_{\mathcal{A}}^{\mathcal{P}}(I)^{\tau(n)} : \Phi(I) = \lambda \wedge i_{\mathcal{A}} \in I\}$$

换言之，仅给定计算输出 λ、一个固定代理 $\mathcal{A}$ 及其输入 $i_{\mathcal{A}}$，以及群体规模 n，就存在一个 PPT 模拟器S，能够计算出某个多项式长度的前缀：该前缀属于 $\tau(n)$ 长度的、代理 $\mathcal{A}$ 的视图，并且在计算上与真实执行中的视图$\mathbf{view}_{\mathcal{A}}^{\mathcal{P}}(I)$不可区分（截至某个固定步骤 $\tau(n)$），前提是由 S 计算出的该视图与输入 n、$i_{\mathcal{A}}$ 和 λ一致。根据模拟范式，此定义意味着，无论执行进行到哪个前缀，$\mathcal{A}$ 从输入向量中获知的信息，都不会超过其从自身输入 $i_{\mathcal{A}}$、公开信息（例如 n 和对已流逝时间的近似值 τ）以及计算输出 λ中所能获知的信息。

请注意，该定义声称，对于每个输入向量，不可区分性这一性质都成立；它并非仅仅是在输入向量空间上取平均后整体成立的性质。

A.2 定义的比较

定理 1. 如果 $\mathcal{P}$ 是信息论安全的隐私，那么 $\mathcal{P}$ 也满足输出独立隐私、弱隐私和输入不可区分性。

证明. 我们分三部分证明这一论断，每部分都通过证明其逆否命题来完成。

i. 如果 $\mathcal{P}$ 不满足输出独立隐私，那么它就不是信息论安全的隐私。由于 $\mathcal{P}$ 不满足输出独立隐私，因此不存在两个输入向量及其对应的执行，使进程 p获得相同的视图。为反证，假设 $\mathcal{P}$ 是弱隐私。那么必定存在输入向量 I_1 和 I_2，使得

$$\Pr(\mathbf{view}_{\mathcal{A}}^{\mathcal{P}}(I_1) = V) = \Pr(\mathbf{view}_{\mathcal{A}}^{\mathcal{P}}(I_2) = V)$$

然而，这将与 $\mathcal{P}$ 不是输出独立隐私这一假设矛盾。因此，$\mathcal{P}$ 必定不是弱隐私；根据 (ii)，它也必定不是信息论安全的隐私。

ii. 如果 $\mathcal{P}$ 不是弱隐私的，那么它就不是信息论安全的隐私。由于 $\mathcal{P}$ 不是弱隐私的，存在某个对手 $\mathcal{A}$，其视图为$V = \langle i;\ q;\ \{(\rho_\varepsilon^{\mathcal{A}}, o_\varepsilon^{\mathcal{A}})\}\rangle$，且输出为 λ，使得 $|\mathcal{M}_\lambda^i| > 1$，但不存在一对输入向量 I_1、$I_2 \in \mathcal{S}_\lambda^2$，使定义 2 中的三个条件全部成立。因此，对于任意一对输入向量 I_1 和 I_2，如果它们属于 $\mathcal{M}_\lambda^i$ 且具有不同的多重集（即它们与对手的输入i 和计算输出 λ一致），我们有：

$$\Pr(\mathbf{view}_{\mathcal{A}}^{\mathcal{P}}(I_1) = V) \neq \Pr(\mathbf{view}_{\mathcal{A}}^{\mathcal{P}}(I_2) = V)$$

因此，我们可以说 нес.

$$\begin{aligned}\Pr(\boldsymbol{I} = I_1 \mid \mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V) &= \frac{\Pr(\boldsymbol{I} = I_1 \wedge \mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V)}{\Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V)} \\ &= \Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{I_1}) = V) / \Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V) \\ &\neq \Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{I_2}) = V) / \Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V) \\ &= \Pr(\boldsymbol{I} = I_2 \mid \mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V)\end{aligned}$$

利用这些信息，我们将证明，对于某个分布 $\mathcal{D}$ on Σ^n ，确实有】【。

$$Pr(\boldsymbol{I} = I \mid \mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V) \neq Pr(\boldsymbol{I} = I \mid \boldsymbol{i}_{\mathcal{A}} = i, \boldsymbol{\lambda}_{\mathcal{A}} = \lambda)$$

对于某个 $I \in \Sigma^n$，这表明 $\mathcal{P}$ 不是信息论安全的隐私。考虑 Σ^n上的均匀分布，这意味着对于所有 $I \in \Sigma^n$，$\mathbf{Pr}(\boldsymbol{I} = I) = 1/|\Sigma|^n$，因此对于 $\mathcal{M}^i_\lambda$中的任意一对输入向量 I_1、I_2，

$$\Pr(\boldsymbol{I} = I_1) = \Pr(\boldsymbol{I} = I_2)$$

由于这些输入来自 $\mathcal{M}^i_\lambda$，我们也可以说，

$$\Pr(\boldsymbol{I} = I_1 \mid \boldsymbol{i}_{\mathcal{A}} = i, \boldsymbol{\lambda}_{\mathcal{A}} = \lambda) = \Pr(\boldsymbol{I} = I_2 \mid \boldsymbol{i}_{\mathcal{A}} = i, \boldsymbol{\lambda}_{\mathcal{A}} = \lambda)$$

然而，因为

$$\Pr(\boldsymbol{I} = I_1 \mid \mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V) \neq \Pr(\boldsymbol{I} = I_2 \mid \mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V)$$

对于某些 $I_1, I_2 \in \mathcal{M}^i_\lambda \subseteq \mathcal{S}_\lambda$，我们必须满足以下条件之一:

$$Pr(\boldsymbol{I} = I_1 \mid \mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V) \neq Pr(\boldsymbol{I} = I_1 \mid \boldsymbol{i}_{\mathcal{A}} = i, \boldsymbol{\lambda}_{\mathcal{A}} = \lambda)$$

or

$$Pr(\boldsymbol{I} = I_2 \mid \mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V) \neq Pr(\boldsymbol{I} = I_2 \mid \boldsymbol{i}_{\mathcal{A}} = i, \boldsymbol{\lambda}_{\mathcal{A}} = \lambda)$$

否则，我们就必须满足

$$\begin{aligned}\Pr(\boldsymbol{I} = I_1 \mid \mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V) &= \Pr(\boldsymbol{I} = I_1 \mid \boldsymbol{i}_{\mathcal{A}} = i, \boldsymbol{\lambda}_{\mathcal{A}} = \lambda) \\ &= \Pr(\boldsymbol{I} = I_2 \mid \boldsymbol{i}_{\mathcal{A}} = i, \boldsymbol{\lambda}_{\mathcal{A}} = \lambda) \\ &= \Pr(\boldsymbol{I} = I_2 \mid \mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V)\end{aligned}$$

这与我们的初始假设—— $\mathcal{P}$ 不是弱隐私——相矛盾。因此，存在某个 $I \in \mathcal{S}_\lambda$ 以及 Σ^n 上的分布，使得

$$\Pr(\boldsymbol{I} = I \mid \mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V) \neq Pr(\boldsymbol{I} = I \mid \boldsymbol{i}_{\mathcal{A}} = i, \boldsymbol{\lambda}_{\mathcal{A}} = \lambda)$$

因此， $\mathcal{P}$ 也不能是信息论安全的隐私。

iii. 如果 $\mathcal{P}$ 无法做到输入不可区分，那么它就不是信息论安全的隐私。

为了使这两个定义具有可比性，我们假设对手始终拥有无限的计算能力。即使如此，我们仍然证明，信息论隐私依然强于无界输入不可区分性（因此，按照计算不可区分性的定义削弱对手能力，也会使这两个定义之间保持相同的关系）。

令 $\mathcal{V} = \Sigma \times Q \times \{\{\text{发起方、响应者}\} \times M\}^{\tau(n)}$（对于某个固定的 τ，其关于 n 是多项式的）表示代理在一次执行 $\mathcal{P}$（其中包含 τ（n）次交互）中所有可能视图的空间。记 $\mathcal{X}$（I）为 $\{S$（n、i、λ）$\}$ 在 $\mathcal{V}$上的分布；并记 $\mathcal{Y}$（I）为 $\{$视图$_{\mathcal{A}}^{\mathcal{P}}$（$\boldsymbol{I}$）$^{\tau(n)}$：$\Phi$（$I$）$= \lambda \wedge i \in I\}$ 在 $\mathcal{V}$上的分布。根据假设，$\mathcal{P}$ 不具备输入不可区分性，因此存在某个输入向量 I^*，使得 $\mathcal{X}$（I^*）$\not\equiv \mathcal{Y}$（I^*），即对于某个 $V \in \mathcal{V}$，有 $\Pr$（$\mathcal{X}$（I^*）$= V$）$\neq \Pr$（$\mathcal{Y}$（I^*）$= V$）。假设该 I^* 在 $\mathcal{A}$ 处的输入为 i，输出为 λ。
因为当 I^* 是真实的输入向量时，$\mathcal{Y}$ 反映了具有输入 i 和输出 λ 的代理 $\mathcal{A}$ 的真实视图分布，我们有

$$\Pr(\mathcal{Y} = V) = \Pr(\mathbf{view}_{\mathcal{A}}^{\mathcal{P}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V \wedge \boldsymbol{I} = I^*)$$

此外，由于 $\mathcal{A}$ 的计算能力不受限制，它可以在所有可能的随机调度下，对每个可能的输入向量模拟该协议（在 $\mathcal{A}$ 处至少进行 $\tau(n)$ 次观测），以计算包含 i 且输出 λ 的输入向量产生视图 V 的精确概率，因此

$$\Pr(\mathcal{X} = V) = \Pr(\mathbf{view}_{\mathcal{A}}^{\mathcal{P}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V \wedge \boldsymbol{i}_{\mathcal{A}} = i, \boldsymbol{\lambda}_{\mathcal{A}} = \lambda)$$

此外，由于 $i \in I^*$ 和 $\Phi(I^*) = \lambda$，我们知道不中返

$$\Pr(\mathbf{view}_{\mathcal{A}}^{\mathcal{P}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V \mid \boldsymbol{I} = I^*) \leq \Pr(\mathbf{view}_{\mathcal{A}}^{\mathcal{P}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V \mid \boldsymbol{i}_{\mathcal{A}} = i, \boldsymbol{\lambda}_{\mathcal{A}} = \lambda)$$

然而，这实际上必须是严格不等式，否则 I^* 就会是 $\mathcal{M}_\lambda^i$ 中唯一的输入向量，这将意味着

$$\Pr(\boldsymbol{i}_{\mathcal{A}} = i, \boldsymbol{\lambda}_{\mathcal{A}} = \lambda \wedge \mathbf{view}_{\mathcal{A}}^{\mathcal{P}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V) = \Pr(\boldsymbol{I} = I^* \wedge \mathbf{view}_{\mathcal{A}}^{\mathcal{P}}(\boldsymbol{i}, \boldsymbol{\lambda}) = V)$$

因此也有 $\Pr(\mathcal{X} = V) = \Pr(\mathcal{Y} = V)$（这与我们最初的假设相矛盾）。

因此，（令 $\varkappa$ 表示事件 $\boldsymbol{i}_{\mathcal{A}} = i$ 和 $\boldsymbol{\lambda}_{\mathcal{A}} = \lambda$）我们有

$$\begin{aligned}
&\Pr(\boldsymbol{I} = I^* \mid \mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(i, \boldsymbol{\lambda}) = V) \\
&= \frac{\Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(i, \boldsymbol{\lambda}) = V \mid \boldsymbol{I} = I^*) \Pr(\boldsymbol{I} = I^*)}{\Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(i, \boldsymbol{\lambda}) = V)} \\
&= \frac{\Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(i, \boldsymbol{\lambda}) = V \mid \boldsymbol{I} = I^*) \Pr(\boldsymbol{I} = I^* \wedge \boldsymbol{i}_{\mathcal{A}} = i, \boldsymbol{\lambda}_{\mathcal{A}} = \lambda)}{\Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(i, \boldsymbol{\lambda}) = V)} \\
&= \frac{\Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(i, \boldsymbol{\lambda}) = V \mid \boldsymbol{I} = I^*) \Pr(\boldsymbol{I} = I^* \wedge \varkappa)}{\Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(i, \boldsymbol{\lambda}) = V)} \\
&= \frac{\Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(i, \boldsymbol{\lambda}) = V \mid \boldsymbol{I} = I^*) \Pr(\boldsymbol{I} = I^* \mid \varkappa) \Pr(\varkappa)}{\Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(i, \boldsymbol{\lambda}) = V)} \\
&< \frac{\Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(i, \boldsymbol{\lambda}) = V \mid \varkappa) \Pr(\boldsymbol{I} = I^* \mid \varkappa) \Pr(\varkappa)}{\Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(i, \boldsymbol{\lambda}) = V)} \\
&= \frac{\Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(i, \boldsymbol{\lambda}) = V \wedge \varkappa) \Pr(\boldsymbol{I} = I^* \mid \varkappa)}{\Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(i, \boldsymbol{\lambda}) = V)} \\
&\leq \frac{\Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(i, \boldsymbol{\lambda}) = V) \Pr(\boldsymbol{I} = I^* \mid \varkappa)}{\Pr(\mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(i, \boldsymbol{\lambda}) = V)} \\
&= \Pr(\boldsymbol{I} = I^* \mid \varkappa) \\
&= \Pr(\boldsymbol{I} = I^* \mid \boldsymbol{i}_{\mathcal{A}} = i, \boldsymbol{\lambda}_{\mathcal{A}} = \lambda)
\end{aligned}$$

因此，我们证明了 $\mathcal{P}$ 也不具备信息论隐私。

B 第 5 节补充材料：对抗性调度下的私有 *Remainder*

定理 5. 在均匀随机调度器下，算法 1 不具备信息论输入隐私。

证明。首先，我们注意到，在均匀随机调度器下，以某代理也与该代理首次交互为条件，该代理可以计算自己与另一代理在其初始状态下交互的概率。固定一个代理 A_j，并考虑它的首次交互。在任意特定的一步中，代理A_j 参与交互的概率为 $2/n$。以第 t 步发生此事件为条件，接下来计算这是否是执行过程中两个代理的首次交互（也就是说，在此前的任何一步中，都没有选择这两个代理进行交互）的概率。在均匀调度器下，该概率等于

$$\left(\frac{\#\text{ pairs not including the two agents}}{\#\text{ total possible pairs}}\right)^{t-1} = \left(\frac{\binom{n-2}{2}}{\binom{n}{2}}\right)^{t-1} = \left(\frac{n^2-5n+6}{n^2-n}\right)^{t-1}$$

对所有可能与 A_j 发生交互的步求和后，我们得到： A_j的首次交互对象也是一个从未交互过的代理（因此仍处于其初始状态）的概率为

$$\sum_{t=1}^{\infty} \frac{2}{n} \left(\frac{n^2 - 5n + 6}{n^2 - n} \right)^{t-1} = \frac{n-1}{2n-3} > \frac{1}{2}$$

因此，以至少 1/2 的概率，一个代理可以看到另一个代理的初始状态。尽管一个代理不一定能检测到这种情况，但如此高的发生概率会改变在给定一个代理的视图的条件下，群体中存在某个已观察到的输入的条件概率：

从人群中均匀抽取的任意特定代理具有输入 σ 的概率为

$$\begin{aligned} p_\sigma = \frac{1}{n} \sum_{j=1}^{n} \Pr(\boldsymbol{i_j} = \sigma) &= \frac{1}{n} \sum_{I \in \Sigma^n} \left(\sum_{j=1}^{n} \Pr(\boldsymbol{i_j} = \sigma \mid \boldsymbol{I} = I) \right) \Pr(\boldsymbol{I} = I) \\ &= \frac{1}{n} \sum_{I \in \Sigma^n} (\text{count of } \sigma \text{ in } I) \Pr(\boldsymbol{I} = I) \end{aligned}$$

鉴于 $\sum_{\sigma \in \Sigma} p_\sigma = 1$，必定存在某个 $\sigma \in \Sigma$ 使得 $p_\sigma \leq 1/|\Sigma|$，并且对于任意非平凡的群体协议 $|\Sigma| \geq 2$。因此，采用上述策略时，代理首次观察到的状态是其交互伙伴实际输入的概率严格大于 1/2；而对于输入集中的某个观察到的输入，该概率并不等于其先验概率。因此，算法 1 不是信息论安全的隐私。

同样可以证明， [9] 中用于求解 Threshold 和 Or 的协议不具备隐私。为简洁起见，我们省略这些证明。

关于输入不可区分性和弱隐私的说明 请注意，算法 1 协议并未为 δ 中的非确定性转移赋予概率。例如，状态为 (0,1) 和 (1,1) 的两个参与者可以执行算法 1 中的转移 (M1) 或转移 (M2)；然而，我们没有可用于作出这一决策的概率分布。因此，要确定该算法是否满足这些输入隐私定义，会更加复杂。

[9] 中提出的隐私概念依赖于一种可能性：对任意特定参与者的状态，(M1) 都可能被应用无限多次，从而不受限制地转换输入向量。如果转移 (M1) 的概率太小，那么参与者的状态值（虽然形式为 (∗,1)）就会被限制在其真实输入值附近的随机游走范围内。相反，即使转移 (M1) 的概率足够大，最初几次交互仍会泄露每个参与者的输入值，因为其交互伙伴以非可忽略的概率能够看到该参与者的全部状态，

如前文在信息论隐私评估中所示。无论哪种情况，代理的视图都会显著缩小输入向量空间，有时甚至将其缩小到仅包含一个输入多重集。

例如，考虑 $k = 2$ 的情况。那么，对于由一次执行产生的 $\mathcal{A}$ 的任何视图，若该执行中有两个除 $\mathcal{A}$ 之外的参与者彼此交互，都可以通过假设交互中参与者的顺序颠倒来混淆。然而，在这类交互仅涉及一个状态为 0 的参与者和另一个状态为 1 的参与者的执行中，这实际上不会产生新的输入多重集（而只是交换两个参与者持有的输入）。事实上，混淆输入多重集的唯一方法是同时改变两个输入相同的参与者的输入，但这必然会改变 $\mathcal{A}$ 的视图。虽然这种情况刻意构造且很罕见，但它仍然使算法 1 无法达到哪怕是弱隐私。这促使我们需要在概率调度群体模型中提出满足这些隐私定义的新协议。

C 概率调度下第 6 节私有 *Remainder* 的补充材料

C.1 RingRemainder

定理 6. RingRemainder 是信息论安全的隐私。

证明。设 $\boldsymbol{I} = \langle \boldsymbol{i_1}, ..., \boldsymbol{i_n} \rangle$ 是表示协议输入向量的随机变量，并设 $\boldsymbol{m_j}$ 是表示从代理 A_j 传递给 A_{j+1} 的 $\mathbb{Z}_k$ 中消息的随机变量。

假设采用安全点对点通信且参与者互不串通，则每个代理 A_j 只能获知自己的输入、收到的消息，以及由领导者广播的最终值（该值恰好等于计算的输出）。因此， A_j 的视图由 $\mathbb{Z}_k$ 中的三个值组成。

我们知道 $\boldsymbol{m_j} = \boldsymbol{r} + \sum_{x=1}^{j} i_x$ （模 k），其中 $\boldsymbol{r}$ 是表示从 $\mathbb{Z}_k$ 中均匀抽取的群元素的随机变量。因此，

$$\Pr\left(\boldsymbol{m_j} = m^* \mid \boldsymbol{I} = I\right) = \Pr\left(\boldsymbol{r} = m^* - \sum_{x=1}^{j} i_x \pmod{k}\right) = 1/k$$

and

$$\Pr(\boldsymbol{m_j} = m^*) = \sum_{I \in \Sigma^n} \Pr(\boldsymbol{m_j} = m^* \mid \boldsymbol{I} = I) \Pr(\boldsymbol{I} = I) = 1/k \sum_{I \in \Sigma^n} \Pr(\boldsymbol{I} = I) = 1/k$$

由此，我们可以得出结论： $\boldsymbol{I}$ 和 $\boldsymbol{m_j}$ 相互独立（这对每个 $j \in \{1, ..., n\}$ 都成立）。

考虑在 A_j处的视图 $V = \langle i^*、\ m^*、\ \lambda^* \rangle$。那么

$$\begin{aligned}\Pr(\boldsymbol{I} = I \mid \mathbf{view}_{\mathcal{A}}^{\mathcal{P}}(\boldsymbol{i_j}, \boldsymbol{\lambda_j}) = V) &= \Pr(\boldsymbol{I} = I \mid \boldsymbol{i_j} = i^*, \boldsymbol{m_{j-1}} = m^*, \boldsymbol{\lambda_j} = \lambda^*) \\ &= \Pr(\boldsymbol{I} = I \mid \boldsymbol{i_j} = i^*, \boldsymbol{\lambda_j} = \lambda^*)\end{aligned}$$

因为 $\boldsymbol{I}$ 和 $\boldsymbol{m_{j-1}}$ 相互独立。因此，我们有

$$\Pr(\boldsymbol{I} = I \mid \boldsymbol{i_j} = i^*, \boldsymbol{\lambda_j} = \lambda^*) = \Pr(\boldsymbol{I} = I \mid \mathbf{view}^{\mathcal{P}}_{\mathcal{A}}(\boldsymbol{i_j}, \boldsymbol{\lambda}) = V)$$

对于所有与输入 i^* 和输出 λ^* 一致的 V（无论m^*的值如何），均需如此。简言之，获知任何参与者的输入（或环上的任何部分和），都归结为猜出由 A_1加入汇总和中的确切随机元素。

C.2 安全点对点传输

定理 2. 算法 2 是一个安全的点对点传输例程。

证明。首先请注意，消息状态的标签分量 L 独立于 μ 更新，因此不会泄露关于 μ的任何信息。因此，我们将重点关注隐藏秘密和公开随机数的状态更新。

有一个指定的发送方持有令牌 $\mathfrak{S}$，以及若干符合条件的接收方参与者。根据 (S1)，发送方外部消息中使用的随机性会在每次交互时刷新，因此，当某个参与者最终在 (S2) 中被标记为指定接收方时，它复制到其内部状态的随机值（以概率 1）只有发送方和接收方知道。令 $\boldsymbol{r} \leftarrow \mathbb{Z}_k$为一个随机变量，表示此次转移中发送方与接收方之间实际传递的随机值。

此时，发送方将其外部消息更改为 $\boldsymbol{s} = \mu - \boldsymbol{r}$（模 k）。回想一下，$\boldsymbol{r}$ 是从 $\mathbb{Z}_k$ 中均匀抽取的，因此对于所有 $x \in \mathbb{Z}_k$，均有 $\mathbf{Pr}(\boldsymbol{r} = x) = 1/k$。因此，对于所有 $x \in \mathbb{Z}_k$，$\mathbf{Pr}(\boldsymbol{s} = \mu - \boldsymbol{r} = x) = \mathbf{Pr}(\boldsymbol{r} = \mu - x) = 1/k$，所以 $\boldsymbol{s}$ 也在 $\mathbb{Z}_k$ 上均匀分布（无论隐藏秘密 μ 是从先验分布 $\mathcal{D}$ 中抽取的）。由于接收方是唯一知道 $\boldsymbol{r}$ 确切值的另一个代理，它会在下一次与发送方交互时获悉 μ。

定义 W_1 为一个算法，该算法在发送方代理作为发起方与标签为 $\mathfrak{u}$ 的另一个代理交互时，选择发送方代理的隐藏秘密；定义 W_2 为一个算法，该算法在接收方代理作为响应者与标签为 $\mathfrak{S}'$ 的另一个代理交互时，选择接收方代理的隐藏秘密。然后，

$$\Pr\left(W_1(\mathbf{view}^{\mathcal{P}}_{\mathsf{Sender}}(\boldsymbol{I})) = \mu\right) = \Pr\left(W_2(\mathbf{view}^{\mathcal{P}}_{\mathsf{Receiver}}(\boldsymbol{I})) = \mu\right) = 1$$

然而，对于群体中的其他每个代理 A_j，每条观测到的外部消息（包括 $\mu - \boldsymbol{r}$）看起来都是均匀随机值，且与抽取隐藏秘密 μ 所依据的分布无关。因此，任何 W' 都必须在除消息空间上的先验分布之外没有其他信息的情况下，猜测 μ 的值：

$$\Pr\left(W'(\mathbf{view}^{\mathcal{P}}_{A_j}(\boldsymbol{I})) = \mu\right) = \Pr(\mu \xleftarrow{\mathcal{D}} \mathbb{Z}_k)$$

因此，只有发送方和选定的接收方能够知道 μ，这使得算法 2 成为一种安全的点对点传输例程。

算法 4：用于流行病驱动探测的群体协议	
$x,\ y \to x,\ \max(x,\ y)$	(P1)
$0,\ y \to 0,\ y$	(P2)
$x,\ y \to x,\ 2 \quad [x > 0]$	(P3)

来自 [3], 的探测协议的转移规则，其中 x 和 y 属于 $\{0, 1, 2\}$。

C.3 探测协议

算法 4 给出了使用 0、1 和 2 三种状态对谓词 π 执行探测的转移规则：当响应者不满足 π 时，参与者通过 (P1) 转移；当响应者满足 π 时，则根据 (P2) 和 (P3) 转移。以下引理给出了算法 4 中探测协议运行时间的界，并量化了其出错概率：

引理 1（见 [3], 引理 6）。对于任意 $c > 0$，存在一个常数 d，使得当 n 足够大时，以至少 $1 - n^{-c}$ 的概率，在探测协议中经过 $dn \ln n$ 次交互后，以下情况之一成立：(a) 没有代理满足该谓词，且每个代理都处于状态 1；或 (b) 某个代理满足该谓词，且每个代理都处于状态 2。

C.4 算法 3 正确性证明

从算法 3 可以看出，计算的有效步骤由发送方通过转移 (R1)、(R2) 和 (R3) 驱动。唯一不涉及发送方的转移是 (R4)：当领导者检测到其他每个代理的标签均为 $\overline{\mathfrak{u}}$ 时，领导者将其标签从 $\overline{\mathfrak{u}}$ 设置为 $\mathfrak{u}$，这意味着每个代理的值都已聚合为由当前发送方持有的单个总和（这种情况只会发生一次）。一旦达到这一时刻，该发送方就应通过 (R2) 和 (R3)，即通过安全点对点传输，将该总和传回领导者。我们旨在证明，算法 3 以高概率在 $\Theta(n^3 \log n)$ 步内计算 Θ 中的余数。首先，我们将证明一系列引理，随后将在定理 3 中利用这些引理证明算法 3 的整体正确性和运行时间。

引理 2. 在执行算法 3 的群体协议的任何配置中，恰好有一个代理的标签为 $\mathfrak{S}$ 或 $\mathfrak{S}'$。

证明。在初始配置中，根据输入函数 $\mathcal{I}$ 的构造，该性质显然成立。在此后的任意一步中，若假设群体中恰好只有一个代理具有 $\mathfrak{S}$ 或 $\mathfrak{S}'$，那么它要么：

– 通过 (R1) 保持对唯一令牌 $\mathfrak{S}$ 的所有权，– 通过 (R2) 将其令牌更新为 $\mathfrak{S}'$（从而消除群体中唯一的 $\mathfrak{S}$，并将其替换为 $\mathfrak{S}'$），或

– 通过 (R3) 将此令牌（最初为 $\mathfrak{S}'$）转移给另一个代理（现在为 $\mathfrak{S}$）。

转换 (R4) 不涉及标签为 $\mathfrak{S}$ 或 $\mathfrak{S}'$ 的参与者。因此，根据归纳法，对于算法 3 的任意执行，该断言在每一步都成立。

引理 3. 对于任意 $c > 0$，从配置 C 开始，其中恰有一个代理的标签为 $\mathfrak{S}$，恰有 t 个参与者的标签为 $\mathfrak{u}$，其余参与者的标签为 $\bar{\mathfrak{u}}$，以至少 $1 - n^{-c}$ 的概率，这些标签在 $\Theta(cn^2 \log n / t)$ 步内保持不变，此时将执行转移 (R2)。

证明. 调度器选中标签为 $\mathfrak{S}$ 的发起方和标签为 $\mathfrak{u}$ 的响应者（从而导致转移 (R2)）的概率为 $\frac{1}{n} \cdot \frac{t}{n-1}$。令 $\boldsymbol{X}$ 为一个随机变量，表示从该初始配置开始，到发生 (R2) 所需的步数。由于调度器的独立均匀随机性，$\boldsymbol{X}$ 服从参数为 $p = \frac{t}{n(n-1)}$ 的几何分布，因此其期望为 $\mathbf{E}[\boldsymbol{X}] = \frac{n(n-1)}{t}$，或者以概率至多 k

$$\begin{aligned}\Pr(\boldsymbol{X} \le k) = \sum_{j=0}^{k} (1-p)^j \, p &= \sum_{j=0}^{k} \left(1 - \frac{t}{n(n-1)}\right)^j \frac{t}{n(n-1)} \\ &= 1 - \left(1 - \frac{t}{n(n-1)}\right)^{k+1}\end{aligned}$$

对于 $k = \frac{n(n-1) \cdot c \log n}{t} - 1$，该概率至少为 $1 - e^{-c \log n}$（这是由以下事实得出的：$(1 - 1/x)^x < 1/e$ 对于 $x \ge 1$）。

从 C 可能发生的唯一其他非空转移，是标签 $\mathfrak{S}$ 与 $\mathfrak{u}$ 之间的转移（从而触发转移（R1）），但这不会对群体中参与者的标签产生任何影响。因此，在（R2）执行一次之前，群体中的标签不会发生变化，而（R2）将在 $\Theta(cn^2 \log n/t)$ 步内以概率 $1 - e^{-c \log n} = 1 - n^{-c}$ 发生，且该概率适用于 $c > 0$的任意选择。

引理 4. 对于任意 $c > 0$，从一个恰有一个代理具有标签 $\mathfrak{S}'$ 且恰有一个代理具有标签 $\mathfrak{R}$的配置开始，我们以概率 $1 - n^{-c}$ 在 $\Theta(cn^2 \log n)$ 步内执行转移 (R3)。

证明。根据引理 2，群体中标签为 $\mathfrak{S}'$ 的代理是唯一具有标签 $\mathfrak{S}$ 或 $\mathfrak{S}'$ 的代理。因此，当带有 $\mathfrak{S}'$ 和 $\mathfrak{R}$ 的参与者相遇时，最终可以执行转移 (R3)。任意一对参与者发生交互的概率为 $1/\binom{n}{2}$。由于概率调度器具有均匀随机性，我们预计平均需要等待 $O(n^2)$ 步才会发生这一事件。此外，对于任意 $c > 0$ 的选择，该事件在 $cn^2 \log n$ 步内发生的概率至少为 $1 - e^{-c \log n} = 1 - n^{-c}$。

由于 (R2) 会产生一个恰好有一个 $\mathfrak{S}'$ 和一个 $\mathfrak{R}$ 的配置，因此我们得到以下推论：

推论1. 对于任意 $c > 0$，在发生 (R2) 的任一步之后，(R3) 都必定会在 $\Theta(cn^2 \log n)$ 步内发生，概率为 $1 - n^{-c}$。

引理 5. 转移 (R2) 和 (R3) 在 Θ 步内，以概率 $1-n^{-c}$ 将发送方和接收方的秘密之和模 k 汇总为接收方的秘密，其中步数为 $cn^2\log n$，且该结论对任意 $c>0$ 均成立。

证明. 根据推论1，(R2) 和 (R3) 实际上是成对执行的（以高概率需要 $\Theta(cn^2\log n)$ 步）。设两个参与者 A_i （作为发起方）和 A_j （作为响应者）之间执行 (R2) 的那一步为 τ_0。这些参与者的秘密分别为 u 和 v，因此我们要证明，在执行 (R2) 和 (R3) 后， A_j 的秘密为 $u+v$ （mod k）。

代理 A_i 的标签为 $\mathfrak{S}$，因此是发送方；执行转移 (R2) 后， A_j 被标记为接收方。转移之后，发送方还会将自己的标签重新标记为 $\mathfrak{S}'$，以表示已选出一名接收方。 A_i 的秘密为 $\perp$，掩码为 $u+r$，而 A_j 的秘密为 $v-r$。

根据引理 4 和推论1，涉及这两个代理的下一次状态更新会执行 (R3)，并且会在 Θ 步内发生，步数为 $cn^2\log n$，其概率为$1-n^{-c}$，且对任意 $c>0$ 均成立。唯一可能发生的其他有效步是 (R4)，因为所有其他转移都涉及某个带有 $\mathfrak{S}$ 标签的代理；根据引理 2，群体中不存在其他 $\mathfrak{S}$ 标签。此外，(R4) 不会改变 A_i 或 A_j 的状态，因为它们的标签不等于 $\overline{\mathfrak{u}}$。

我们知道，当 (R3) 最终执行时，必须由 A_i 作为发起方、 A_j 作为响应者，因为在群体中，只有这两个代理具有相应的标签，并且没有采取其他有效步骤将它们转移给任何其他代理。因此， A_i 仍然将 $u+r$ 作为其掩码，而 A_j 将 $v-r$ 作为其秘密。随后，转移 (R3) 使 A_j 将其秘密更新为 $x+y=(u-r)+(v+r)=u+v$，并使 A_i 保持标签 $\overline{\mathfrak{u}}$（表明其秘密现在为 $\perp$）。

引理 6. 对于任意 $c>0$，从配置 C 开始，其中恰有一个参与者具有标签 $\mathfrak{S}$，恰有 t 个参与者具有标签 $\mathfrak{u}$，其余参与者具有标签 $\overline{\mathfrak{u}}$，我们以至少 $1-n^{-c}$ 的概率，在 $\Theta(cn^2\log n)$ 步内到达配置 C'，其中恰有一个参与者具有标签 $\mathfrak{S}$，恰有 $t-1$ 个参与者具有标签 $\mathfrak{u}$，其余参与者具有标签 $\overline{\mathfrak{u}}$。

证明。根据引理 3，对于任意 $c'>0$ 和配置 C，我们以概率 $1-n^{-c'}$ 在 $\Theta(c'n^2\log n/t)$ 步内执行 (R2)（仅改变若干参与者的随机性值）。当转移 (R2) 被执行时，标签$\mathfrak{S}$ 和 $\mathfrak{u}$ 被 $\mathfrak{S}'$ 和 $\mathfrak{R}$覆盖。根据引理 4，对于任意 $c''>0$，我们以概率$1-n^{-c''}$从此配置出发，在额外的 $\Theta(c''n^2\log n)$ 步内执行 (R3)，此时标签 $\mathfrak{S}'$和 $\mathfrak{R}$ 被$\overline{\mathfrak{u}}$和 $\mathfrak{S}$覆盖。总的来说，在这 $\Theta(c'n^2\log\ n/t+c''n^2\log n)$ 步中，这些变化使标签为 $\mathfrak{u}$ 的参与者数量减少 1（从而使标签为 $\overline{\mathfrak{u}}$的参与者数量增加 1）。对于任意 $c>0$，令 $c'=c''=c+1$。对这些引理中所述每个阶段的失败概率取并集界，可得失败的总概率为 $2n^{-(c+1)}$，对于 $c>0$ 和$n\geq 2$，该概率至多为 n^{-c}。将关于 c 的 c' 和 c'' 这一表达式代回上述关于步数的渐近界，并知晓对于 $t>0$有 $1\leq 1+1/t\leq 2$，我们得到一个

总运行时间为

$$\Theta(c'n^2\log n/t + c''n^2\log n) = \Theta((c+1)(1+1/t)n^2\log n) = \Theta(cn^2\log n)$$

引理 7. 对于任意 $c > 0$，在 $\Theta(cn^3\log n)$ 步内，以概率 $1 - n^{-c}$，算法 3 将所有参与者的个体秘密聚合为一个模 k 的和，并将除一个被标记为发送方且持有该和的参与者外的所有参与者标记为 $\overline{\mathfrak{u}}$。

证明. 在协议开始时，每个参与者 A_j 的状态为 $\langle i_j, (r^j, \mathfrak{u}, 0, 0)\rangle$，而领导者的状态为 $\langle i_j + r^0, (r^j, \mathfrak{S}, 1, 0)\rangle$。这意味着恰好有一个参与者的标签为 $\mathfrak{S}$，而 $n-1$ 个参与者的标签为 $\mathfrak{u}$。

唯一可能执行的转移是先执行 (R2)，再执行 (R3)；根据引理 5，该转移会将领导者的秘密与群体中的某个其他参与者的秘密聚合起来。这还会使领导者被标记为 $\overline{\mathfrak{u}}$，并使所选参与者成为新的发送方（标签为 $\mathfrak{u}$ 的参与者数量减少 1）。

根据引理 6，对于任意 $c' > 0$，以概率 $1-n^{-c'}$，需要 $\Theta(c'n^2\log n)$ 步才能将具有 $\mathfrak{u}$ 的参与者数量减少 1。反复应用引理 6，我们发现需要

$$\Theta\left(\sum_{t=1}^{n-1} c'n^2\log n\right) = \Theta(c'(n-1)n^2\log n) = \Theta(c'n^3\log n)$$

步才能使具有标签 $\mathfrak{u}$ 的参与者数量变为 0。每当具有标签 $\mathfrak{u}$ 的参与者数量减少时，标签被覆盖的参与者的秘密就会聚合到发送方的秘密中，随后该参与者会被标记为 $\overline{\mathfrak{u}}$，以表明其不会再次成为发送方。当没有参与者具有 $\mathfrak{u}$ 时，发送方就拥有参与者秘密的总和。对每次应用引理 6 时的失败概率使用并集界，可得失败的总概率为 $n \cdot n^{-c'}$。对于任意 $c > 0$，令 $c' = c + 1$，即可得到关于 $n \geq 1$的所需结果。

引理 8。转换 (R4) 仅发生一次，即探针确定群体中没有参与者具有标签 $\mathfrak{u}$ 时。对于任意 $c > 0$，以概率 $1 - n^{-c}$，仅当所有参与者都具有 $\overline{\mathfrak{u}}$ 时，探针才会触发。

证明。根据构造，群体中只有一个领导者，且每个代理初始化时都配有一个探针，该探针在实例化时输出 0。探针一旦触发，就无法重置。只有探针触发后才可能发生（R4），因此这种情况只会发生一次，且只会发生在领导者身上。

根据引理 1，当群体中某个代理满足 π 时，对于任意 $c' > 0$，只有以 $n^{-c'}$ 的概率，才会出现并非每个代理都处于状态 2 的情况。如果这样的代理是正在探测标签 $\mathfrak{u}$ 的领导者，那么它会在相位时钟周期结束时错误地输出 0，并导致协议在协议计算中跳过一定数量的代理，这可能造成错误输出。为避免这种情况，我们需要领导者的探针在协议运行期间的每一轮相位时钟中都成功。根据引理 7，对于任意 $c'' > 0$，聚合所有代理输入所需的步数为 $\Theta(c''n^3\log n)$，其概率为 {v

1}。对于 $c'' > 0$，由于相位时钟每轮包含 $O(n \log n)$ 步，因此我们需要探针连续成功 $O(c''n^2)$ 次。根据并集界，在这 $O(c''n^2)$ 次连续试验中探针至少失败一次的概率为 $dc''n^2 \cdot n^{-c'}$，其中 d 是某个常数。因此，对于任意 $c > 0$，令 $c' = c + 3$，当 n 足够大时，即可得到所需结果。

现在，我们已准备好证明以下关于算法 3 正确性的定理：

定理 3. 对于任意固定的 $c > 0$，算法 3 在大小为 n 的种群中，以至少 $1 - n^{-c}$ 的概率在 $\Theta(n^3 \log n)$ 步内计算余数。

证明。根据引理 7，对于任意 $c' > 0$，在 $\Theta(c'n^3 \log n)$ 步内（概率为 $1 - n^{-c'}$），群体会将所有参与者的个体秘密汇总为一个总和，由一个带有单数发送者标签的参与者持有；此时所有其他参与者都标记为 $\overline{\mathfrak{u}}$，而发送方持有秘密

$$r^0 + \sum_{j=1}^{n} i_j \pmod{k}$$

根据引理 1，对于任意 $c'' > 0$，探针在此点之前不会触发，并且在到达此点后的 $O(c''n \log n)$ 步内会以概率 $1 - n^{-c''}$ 成功触发。因此，根据引理 8，对于任意 $c''' > 0$，随后都会执行 (R4)（其期望执行时间为常数，并且以概率 $1 - n^{-c'''}$ 在多对数时间内完成），且仅将领导者的标签更改为 $\mathfrak{u}$。这使得剩余的一个携带 $\mathfrak{S}$ 令牌的代理能够与领导者进行最后一次有效交换。根据引理 5，此次交换会在额外的 $\Theta(c''''n^2 \log n)$ 步内以概率 $1 - n^{-c''''}$ 将汇总和传回领导者，其中该结论对任意 $c'''' > 0$ 均成立。

对这四种误差来源应用并集界，可得失败概率为 $n^{-c'} + n^{-c''} + n^{-c'''} - n^{-c''''}$。对于任意固定的 $c > 0$，令 $c' = c'' = c''' = c'''' = c + 1$，则当 n充分大时，成功概率至少为 $1 - n^{-c}$。

最后，领导者可以从秘密中移除 r^0，从而得到参与者输入之和模 k 的余数。还可以使用单独的机制，通过流行传播将此值分发给群体中的其他参与者。

C.5 算法 3 隐私证明

定理 4. 当算法 3 正确计算 *Remainder*谓词时，它满足信息论输入隐私。

证明。假设该协议正确计算余数谓词（即，在协议收敛后的至少 T 时间内，相位时钟不会发生任何故障），代理的视图由其自身的输入、其初始状态以及一定数量的观测交互组成（从中可以推导出它能够访问的所有其他信息，例如协议输出）。

让我们考虑群体 A_j 中某个代理的视图，直到步骤 $\tau > T$，其中该代理的输入为 (i^*, ℓ^*)，输出为 λ^*，令 $\alpha_j(\tau)$ 表示 A_j 参与的交互次数，直到步骤 τ（因此 $\sum_{j=1}^{n} \alpha_j(\tau) = \tau$）：

$$\mathsf{view}^{\mathcal{P}}_{A_j}((i^*, \ell^*), \lambda^*)^{\tau} = \langle (i^*, \ell^*); q_0^j; \{(\rho_t, (r_t, L_t, \ell_t, Z_t))\}_{t=1}^{\alpha_j(\tau)} \rangle$$

回想一下，$\mathcal{I}$ 是随机化的，因此 q_0^j 是一个随机变量，表示由用于计算 $\mathcal{I}$的随机性所确定的代理 A_j 的实际初始状态。

记 $\mathcal{Z}$ 为探测协议的状态空间，并令

$$\mathcal{V} = \mathbb{Z}_k \times \{0,1\} \times \mathbb{Z}_k \times \{\{\mathsf{Initiator}, \mathsf{Responder}\} \times \mathbb{Z}_k \times \{\mathfrak{S}, \mathfrak{S}', \mathfrak{u}, \overline{\mathfrak{u}}\} \times \{0,1\} \times \mathcal{Z}\}^{\alpha_j(\tau)}$$

表示 A_j 在计算执行至第 τ 步之前所有可能视图的空间。

此外，设 $\theta_t = (\rho_t, (r_t, L_t, \ell_t, Z_t))$ 为表示 A_j 的第 t 次观测的随机变量。在序列 $\{\theta_t\}^{\alpha_j(\tau)}{}_{t=1}$ 中，有四个特殊观测必须出现在任何正确的协议执行中：

$$\begin{aligned}
\theta_a &= (\mathsf{Initiator}, (r_a, \mathfrak{S}, \ell_a, Z_a)), \text{ Sender initiates secure transfer} \\
\theta_b &= (\mathsf{Initiator}, (r_b, \mathfrak{S}', \ell_b, Z_b)), \text{ Sender transmits secret value} \\
\theta_c &= (\mathsf{Responder}, (r_c, \mathfrak{u}, \ell_c, Z_c)), \text{ eligible Receiver is selected} \\
\theta_d &= (\mathsf{Responder}, (r_d, \mathfrak{R}, \ell_d, Z_d)), \text{ secret sent to selected Receiver}
\end{aligned}$$

除领导者外，所有人都按上述顺序观察（而领导者观察到的是：$\theta_c, \theta_d, \theta_a, \theta_b$）。其余观察要么导致发送方刷新其随机性值，要么导致领导者将自身重新标记为 $\overline{\mathfrak{u}}$，要么不会引起状态变化。

我们希望证明，值向量 $\langle q_0^j, \{\theta_t\}^{\alpha_j(\tau)}{}_{t=1} \rangle$ 在给定 A_j 处固定的输入值（i^*, ℓ^*）以及 A_j观察到的输出 λ^* 的条件下，与输入向量 I 条件独立，这将意味着

$$\begin{aligned}
&\Pr(I = I^* \mid \mathsf{view}^{\mathcal{P}}_{A_j}((i^*, \ell^*), \lambda^*)^{\tau} = V) \\
=\ &\Pr(I = I^* \mid i_j = (i^*, \ell^*), \langle q_0^j, \{\theta_t\}_{t=1}^{\alpha_j(\tau)} \rangle = \langle q^*, \{\theta_t^*\}_{t=1}^{\alpha_j(\tau)} \rangle, \lambda_j = \lambda^*) \\
=\ &\Pr(I = I^* \mid i_j = (i^*, \ell^*), \lambda_j = \lambda^*)
\end{aligned}$$

为此，我们首先将注意力限制在一个固定的调度 $s \in S$ 上（即，具体指定在协议的 τ 步中进行交互的代理对），其中

$$S = \{\{(x_t, y_t)\}_{t=1}^{\tau} : (x_t, y_t) \in \mathbb{Z}_n^2 \wedge x_t \neq y_t\}$$

我们假设只考虑协议执行成功的调度 s（也就是说，除去一小部分比例为多项式级别倒数的调度）。

由于 q_0^j 是 (i^*, ℓ^*) 的函数，因此我们可以说，事件 $i_j = (i^*, \ell^*) \wedge q_0^j = q^* = \langle i^*, (r^*, \ell^*, Z_0) \rangle$ 与事件 $i_j = (i^*, \ell^*) \wedge r^j = r^*$ 完全相同，其中 r^j 是一个随机变量，表示 A_j 在计算 $\mathcal{I}$ 时使用的初始随机性。请记住，r^j 在 $\mathbb{Z}_k$ 上均匀分布，且独立于协议中的所有其他变量（包括 I、i_j、λ_j 和 s）。

此外，当调度 s 固定时，被选中进行交互的参与者对的顺序决定了发送方和接收方令牌在参与者之间传递的顺序。这完全确定了 $\boldsymbol{\theta_t}$ 的 $\boldsymbol{L_t}$、$\boldsymbol{\ell_t}$ 和 $\boldsymbol{Z_t}$ 分量。剩下需要考虑的值只有外部随机性值 $\boldsymbol{r_t}$ 和内部秘密状态 $\boldsymbol{\mu_t}$（由于 (R2)，它与 $\boldsymbol{r_t}$ 的值相关）。然而，根据协议成功计算出余数这一假设，我们可以看出，只有当参与者观察到 θ_a，并在之后某个时刻观察到 θ_b（它们分别触发转移 (R2) 和 (R3)）时，每个参与者才会执行一次 (R2)。对于固定的 s，这两次观察之间的交互次数是固定的，因此这两个值之间唯一的相关性，是它们之间隔着正在传递的秘密。对于固定的调度（即参与者值汇总的顺序固定），请记住，该秘密等于 $r^0 + \sum_{x \in X \subseteq \mathbb{Z}_n} i_x$。其他所有观察所具有的 $\boldsymbol{r_t}$，都是从独立于其他所有值的均匀分布中抽取的。因此，事件 $\{\boldsymbol{\theta_t}\}^{\alpha_j(\tau)}{}_{t=1} \wedge s$ 与 $\boldsymbol{r^0} = r^{**} \wedge \{\boldsymbol{r_t} = r_t^*\}^{\alpha_j(\tau)}{}_{t=1} \wedge s$ 相同，其中 $r^{**} = r_t^* - \sum_{x \in X \subseteq \mathbb{Z}_n} i_x$（模 k），且每个 r_t^* 都是 A_j 在每次观察中实际观察到的随机性值。这是因为，在给定固定调度的情况下，各个 $\boldsymbol{\theta_t}$ 的值唯一的变化来源，是参与者在每一步抽取的随机性值；这些值同样是均匀随机的，并且独立于协议中的所有其他变量。

令 $\boldsymbol{S}$ 为一个随机变量，表示调度器选择进行交互的 τ 个参与者的有序对。在给定某个固定调度 s 和在 A_j 处的某个固定视图 V 的条件下，根据协议中均匀抽取的随机值与所有其他变量相互独立的性质，对于任何与 A_j 处的输入 (i^*, ℓ^*) 及输出 λ^* 相一致的 $I \in \Sigma^n$，

$$
\begin{aligned}
&\Pr(\boldsymbol{I} = I \mid \mathsf{view}^{\mathcal{P}}_{\boldsymbol{A_j}}((\boldsymbol{i^*}, \boldsymbol{\ell^*}), \boldsymbol{\lambda^*})^{\boldsymbol{\tau}} = V, \boldsymbol{S} = s) \\
&= \Pr(\boldsymbol{I} = I \mid \boldsymbol{i_j} = (i^*, \ell^*), \langle \boldsymbol{q_0^j}, \{\boldsymbol{\theta_t}\}_{t=1}^{\alpha_j(\tau)} \rangle = \langle q^*, \{\theta_t^*\}_{t=1}^{\alpha_j(\tau)} \rangle, \boldsymbol{\lambda_j} = \lambda^*, \boldsymbol{S} = s) \\
&= \Pr(\boldsymbol{I} = I \mid \boldsymbol{i_j} = (i^*, \ell^*), \boldsymbol{r^j} = r^*, \{\boldsymbol{r_t} = r_t^*\}_{t=1}^{\alpha_j(\tau)}, \boldsymbol{\lambda_j} = \lambda^*, \boldsymbol{S} = s) \\
&= \Pr(\boldsymbol{I} = I \mid \boldsymbol{i_j} = (i^*, \ell^*), \boldsymbol{\lambda_j} = \lambda^*, \boldsymbol{S} = s)
\end{aligned}
$$

因此，在给定 $\boldsymbol{S}$、$\boldsymbol{i_j}$ 和 $\boldsymbol{\lambda_j}$ 的条件下，$\boldsymbol{I}$ 与 $\langle \boldsymbol{q_0^j}, \{\boldsymbol{\theta_t}\}^{\alpha_j(\tau)}{}_{t=1} \rangle$ 条件独立。根据全概率公式，对所有可能的调度 $s \in S$ 求和可得

$$
\begin{aligned}
&\Pr(\boldsymbol{I} = I \mid \mathsf{view}^{\mathcal{P}}_{\boldsymbol{A_j}}((\boldsymbol{i^*}, \boldsymbol{\ell^*}), \boldsymbol{\lambda^*})^{\boldsymbol{\tau}} = V) \\
&= \sum_{s \in S} \Pr(\boldsymbol{I} = I \mid \mathsf{view}^{\mathcal{P}}_{\boldsymbol{A_j}}((\boldsymbol{i^*}, \boldsymbol{\ell^*}), \boldsymbol{\lambda^*})^{\boldsymbol{\tau}} = V, \boldsymbol{S} = s) \Pr(\boldsymbol{S} = s) \\
&= \sum_{s \in S} \Pr(\boldsymbol{I} = I \mid \boldsymbol{i_j} = (i^*, \ell^*), \boldsymbol{\lambda_j} = \lambda^*, \boldsymbol{S} = s) \Pr(\boldsymbol{S} = s) \\
&= \sum_{s \in S} \Pr(\boldsymbol{I} = I \mid \boldsymbol{i_j} = (i^*, \ell^*), \boldsymbol{\lambda_j} = \lambda^*) \Pr(\boldsymbol{S} = s), \text{ by independence of the scheduler} \\
&= \Pr(\boldsymbol{I} = I \mid \boldsymbol{i_j} = (i^*, \ell^*), \boldsymbol{\lambda_j} = \lambda^*) \sum_{s \in S} \Pr(\boldsymbol{S} = s) \\
&= \Pr(\boldsymbol{I} = I \mid \boldsymbol{i_j} = (i^*, \ell^*), \boldsymbol{\lambda_j} = \lambda^*)
\end{aligned}
$$