Vers：采用分布式编码的全分布式编码计算系统

Nastaran Abadi Khooshemehr、Mohammad Ali Maddah-Ali

沙里夫理工大学、明尼苏达大学

电子邮箱：abadi.nastaran@ee.sharif.ir，maddah@umn.edu

arXiv:2304.05691v1 [cs.DC] 12 Apr 2023

I. 摘要

编码计算已被证明对分布式计算很有用，并解决了掉队工作节点等挑战。我们注意到，迄今研究的几乎所有编码计算系统都采用一个主节点和若干工作节点的架构。然而，区块链、物联网和联邦学习等新兴技术对编码计算系统提出了新的要求。在这些系统中，数据以分布式方式生成（并可能存储），因此由主节点进行集中式编码/解码既不可行，也难以扩展。本文提出一种全分布式编码计算系统，由 $k \in \mathbb{N}$ 个数据所有者和 $N \in \mathbb{N}$ 个工作节点组成；数据所有者雇用工作节点对其数据进行计算，计算内容由一个次数为 $d \in \mathbb{N}$的目标函数 f 指定。由于没有中央编码器，工作节点会在计算阶段开始前自行编码。该系统面临的挑战是存在对抗性数据所有者：他们并不了解诚实数据所有者的数据，却会向不同工作节点发送不同版本的数据，从而造成差异，损害工作节点的本地编码。对抗性数据所有者至多有 $\beta \in \mathbb{N}$ 个，且每个最多分发 $v \in \mathbb{N}$个不同的数据版本。由于工作节点和诚实数据所有者并不知道对手及其可能存在的串通行为，工作节点除了执行主要计算任务外，还会为收到的数据计算标签，并将标签发送给数据所有者，以帮助其解码。我们提出一种标签函数，使数据所有者能够将工作节点划分为若干集合，其中每个集合内的工作节点此前都从所有数据所有者处收到了相同的数据。随后，我们刻画该全分布式编码计算系统的基本极限，记为 t^*，即其工作结果可用于正确计算诚实数据所有者数据的目标函数的最少工作节点数。我们证明 $t^* = v^\beta d(K-1) + 1$，并给出逆向证明和可达性证明。

II. 引言

编码计算利用编码理论技术来解决分布式计算系统中的基本挑战，例如掉队节点缓解、安全性、隐私保护和通信带宽问题 [1]。编码计算的核心思想是注入数据或计算冗余，以实现上述目标。例如，编码计算可以利用速度更快的工作节点的工作来完成计算任务，而不依赖于任何特定的一组工作节点（掉队节点缓解）。这项研究方向

受到了如此多的关注，其许多方面也已得到探索，参见 [2] 关于编码分布式计算的综述。然而，几乎所有编码计算研究都考虑这样一种设置：一个主节点（也称为参数服务器等其他名称）和若干工作节点，其中主节点对数据进行编码，在工作节点之间分发数据和计算任务，收集工作节点返回的部分结果，并构建最终结果。以下原因促使我们探索有别于集中式单主节点编码计算系统的路径。

- 在一个由单个主节点编码全部初始数据，然后解码所有结果的分布式计算系统中，主节点将成为可扩展性的瓶颈。随着工作节点数量的增加，主节点在计算和通信方面都应变得更强大。

- 在某些应用中，例如联邦学习， [3], 由于数据所有者的隐私保护需求和/或数据量庞大，计算任务所需的数据不会集中在同一地点。因此，单个主节点无法访问全部数据。

- 专门为分布式数据源设计的计算方案更适合数据生产本质上具有分布式特征的应用，例如分片区块链、联邦学习和物联网。

- 在实践中，作为工作节点的服务器集群通常会服务于多个主节点，并为多个主节点并行执行计算任务。这样可以高效利用工作节点的资源，避免资源闲置。

已有一些研究探讨了无主节点编码计算，或由多个主节点参与的编码计算。在 [4], 一种完全去中心化且无主节点的架构中，仅包含 $N \in \mathbb{N}$ 个工作节点。在这种架构下，研究者针对两项特定的计算任务——矩阵乘法和快速傅里叶变换——提出了去中心化编码策略。在两个任意矩阵 A 和 B 的矩阵乘法中，每个工作节点最初都持有 A 和 B 的一部分。随后，每个工作节点对其初始数据进行本地编码，与其他工作节点通信，然后将两个矩阵相乘。算法结束时，每个工作节点都持有计算结果 AB 的一部分。请注意，工作节点执行的编码针对的是各自的初始数据，而非从其他工作节点接收的数据。我们将论证，对来自外部来源的数据进行分布式编码会带来新的挑战。

在 [5], 中，多个主节点各自拥有一个矩阵向量乘法任务，共享一些工作节点，并且这些工作节点具有异构计算能力。问题是将计算任务分配给工作节点，使总体完成时间最短。每个工作节点要么服务于一个主节点，要么服务于多个主节点，并在它们之间分配其计算能力。在后一种情况下，每个工作节点仅分别执行各主节点的计算任务，不涉及编码。在 [6], 中，有 $N \in \mathbb{N}$ 个工作节点，每个工作节点拥有一部分数据，并希望获得所有数据的线性组合。这里没有主节点，工作节点彼此交换消息，以通信高效的方式计算各自的目标线性组合。 [7] 研究了一个由 $N \in \mathbb{N}$ 个节点构成的网络图，每个节点都有一个初始值，其中一部分节点希望计算初始值的某些任意函数。该子集中的每个节点的目标函数可能不同。为了实现这一目标，采用了一种迭代线性策略：节点从其邻居节点接收消息，并获得所接收值与其自身值的线性组合，且不需要任何领导节点。给定节点图具有某种特定的连通性

在这些性质下，节点只需经过有限轮次，即可使用前几轮获得的线性组合计算初始值所对应的目标函数。在 [8], 中研究了一个存在对手的类似问题，其中对手可能错误地计算线性组合，并向其邻居节点发送错误的值。对手不会向不同的邻居节点发送不同的值，并且只能在计算其邻居节点的消息与自身消息的线性组合时偏离。

编码计算可以被设计为在对抗性环境中保证安全性，如 [9]–[12] 所示。在由一个主节点和多个工作节点组成的传统设置中，编码计算方案的安全性意味着该方案能够抵御对抗性工作节点，因为在此类设置中，通常假设主节点是诚实的。对抗性工作节点可以随意偏离协议，例如返回任意结果、向不同的工作节点发送不同的数据、拒绝发送数据等。然而，当存在多个主节点时，合理的假设是其中一些主节点也可能具有对抗性。

本文研究一种由多个主节点和工作节点组成的编码计算系统，其中主节点雇用工作节点为其完成计算任务。我们将主节点称为数据所有者，因为每个数据所有者都拥有一部分数据，并希望借助工作节点获得该数据的多项式目标函数。部分数据所有者具有对抗性，试图在系统中制造差异，从而导致诚实数据所有者的计算任务出错。数据所有者与工作节点之间，以及对抗性数据所有者之间都存在通信链路，因为后者可以彼此串通。然而，与许多编码计算系统一样[1], [2], , 我们的系统中工作节点之间没有通信方式。系统的工作流程如下：数据所有者将数据分发给工作节点，但具有对抗性的数据所有者可能向不同工作节点分发互相矛盾的数据，以破坏计算结果。对抗性数据所有者可以自由合作，决定向工作节点发送什么。收到数据后，每个工作节点对接收到的所有数据所有者的数据进行编码，在编码数据上应用目标函数，然后将结果连同一个用于标记所收数据的小标签一并返回给数据所有者，以便诚实的数据所有者间接识别具有对抗性的数据所有者。标签使诚实的数据所有者能够根据工作节点在第一步中从对手处收到的相同消息，将工作节点划分为不同集合，并帮助诚实的数据所有者解码结果。最后，诚实的数据所有者可以从工作节点返回的结果中提取所需信息。

我们将所描述的系统命名为 Vers，它是 versatile（多用途）的缩写。之所以这样命名，是因为工作节点确实具有多种用途，并执行一系列不同的任务：对接收到的数据进行编码，计算编码数据的目标函数，以及计算标签。我们研究在工作节点中部署拉格朗日编码时 Vers 的基本极限。我们将这一基本极限记为 t^*，它是一个最小数量，使得诚实数据所有者能够从任意一组 t^* 工作节点的结果中正确且可靠地提取所需信息。换言之，对于任何对抗行为，任意一组 t^* 工作节点都应足以让诚实数据所有者正确计算其数据的目标函数。另一方面，为使 t^* 确实为最小值，应存在一种对抗行为和一组 $t^* - 1$ 工作节点，使得它们的结果无法唯一确定诚实数据所有者的目标。值得注意的是，在我们之前的工作 [13], 中，我们研究了这样一种系统的基本极限：一些数据所有者希望将其数据存储在某些存储节点中，以便之后能够正确检索数据。在该系统中，一些数据所有者具有对抗性，可能会向不同的存储节点发送不同的数据。

节点。

我们问题中的对抗行为类型，即对手向其他节点发送不一致的数据，已从不同视角在广泛的分布式系统中得到研究，例如拜占庭协议[14]–[16], 、拜占庭广播 [17]–[19], 、可验证信息分散 [20]–[22],以及分布式密钥生成[23]。尽管这种对抗行为此前已为人所知，但在区块链、联邦学习和物联网等分布式计算系统背景下应对这类行为的研究仍非常有限。本文与那些考虑类似对抗模型的相关工作的区别在于，我们处理的是计算，而非共识。我们的目标是正确计算数据子集的函数，而这并不一定要求在计算之前就对这些数据达成共识。通常，对手造成的不一致性需要以某种方式解决，主要有两种方法：节点可以通过相互通信自行解决彼此之间的不一致性，或者将不一致性的处理推迟到后续步骤，例如解码。第一种方法在许多工作中得到采用，例如 [20],，其中节点相互交换足够的信息，以提取一致的数据，然后在其主要任务中使用这些一致的数据。由于在我们的模型中，工作节点之间不存在通信链路，因此它们无法检测对抗性数据所有者，也无法通过相互交换信息来减轻其影响。因此，仅采用第一种方法是不够的。在本文中，针对我们的特定模型，我们选择将这两种方法结合起来：在解码的最终步骤中处理不一致性，并借助工作节点先前生成并传达给数据所有者的标签。

本文的结构如下。我们在第三节中构建问题，并在第八节中陈述主要结果。然后，在第六节中解释标签函数的概念，并介绍一种良好的标签函数。我们在第七节中定义一些概念并分析系统。最后，我们在第八节中陈述系统的基本极限。

III. 问题表述

本节正式介绍 $\mathsf{Vers}(f, N, K, \beta, v, \mathsf{Enc})$，这是一个参数为 f, N, K, β, v、采用编码算法 Enc 的全分布式线性编码计算系统。考虑一个由 $K \in \mathbb{N}$ 个数据所有者和 $N \in \mathbb{N}$ 个工作节点组成的系统，它们的索引分别为 $1, \ldots, K$ 和 $1, \ldots, N$，其中 $K \leq N$。我们将集合 $\{1, \ldots, m\}, m \in \mathbb{N}$ 记为 $[m]$，并将 $\{0, 1, \ldots, m\}$ 记为 $[m^+]$。数据所有者 $k \in [K]$ 拥有 $X_k \in \mathbb{U}$，并希望获得 $f(X_k)$，其中 $f\colon \mathbb{U} \to \mathbb{V}$ 是一个次数为 $d \in \mathbb{N}$ 的任意多项式函数，称为目标函数； $\mathbb{U}, \mathbb{V}$ 是有限域 $\mathbb{F}$ 上的向量空间。数据所有者以分布式方式雇用工作节点为其执行计算。在数据所有者中，最多有 $\beta \in \mathbb{N}$ 个对抗节点， $\beta < K$。我们将对抗性数据所有者的集合记为 $\mathcal{A} \in [K]$，将诚实数据所有者的集合记为 $\mathcal{H} = [K] \setminus \mathcal{A}$。对抗性数据所有者试图制造差异，以误导诚实数据所有者，使其无法获知 $f(X_k)$ 的正确值， $k \in \mathcal{H}$。对抗性数据所有者放弃使用工作节点进行计算的机会，因此并不在意最终是否得到任何特定的正确计算结果。对抗性数据所有者可以彼此合作，但不知道诚实节点的数据。我们假设所有工作节点都是诚实的。此外，诚实数据所有者和工作节点并不知道哪些数据所有者是对抗性的。

我们从 [12]中汲取灵感来设计 Vers。下文将详细介绍 Vers 的工作流程。该工作流程也展示在算法1中。第一步，数据所有者将数据发送给工作节点。令$X_{k,\ n}$表示数据所有者 $k \in [K]$ 发送给工作节点 $n \in [N]$的消息。诚实数据所有者 $k \in \mathcal{H}$ 向所有工作节点发送X_k，即 $X_{k,\ n} = X_k$，对于所有 $n \in [N]$均成立。我们假设 $X_k, k \in \mathcal{H}$ 是从 $\mathbb{U}$中独立且均匀地随机选取的。对抗性数据所有者 $k \in \mathcal{A}$ 至多生成 $v \in \mathbb{N}$ 条不同的消息，我们称之为版本，即 $X_k^{(1)}, \ldots, X_k^{(v)}$，并将其中一条发送给每个工作节点。换言之，对于 $k \in \mathcal{A}$，有 $\{X_{k,n}, n \in [N]\} = \{X_k^{(1)}, \ldots, X_k^{(v)}\}$。对抗性数据所有者最多可以生成 v 条自选消息，并可将其中任意一条发送给其选择的任意工作节点。对抗性数据所有者还可以串通，以使其整体对抗行为造成尽可能严重的危害。

在第二步中，工作节点 n 从数据所有者处接收到 $X_{1,\ n}, \ldots, X_{K,\ n}$ 后，使用编码算法 Enc 对 K 条接收到的消息进行线性组合，记为 $W_n \in \mathbb{U}$。设 $\gamma_{k,\ n} \in \mathbb{F}, k \in [K]$ 为编码算法 Enc 为工作节点 n 确定的编码系数。因此，

$$W_n = \sum_{k \in [K]} \gamma_{k,n} X_{k,n}, \quad n \in [N]. \tag{1}$$

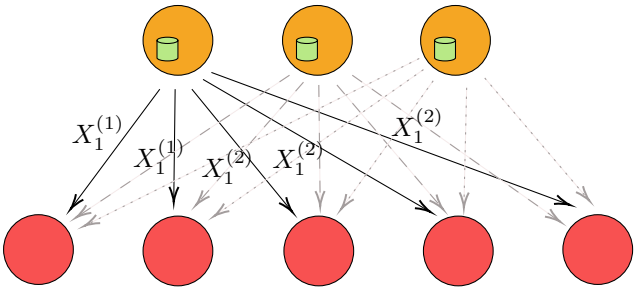
我们强调，Vers 中的编码以去中心化方式在每个工作节点本地进行。这是 Vers 的核心思想，也是它区别于其他采用中央编码器的编码计算系统之处。在对接收到的消息进行本地编码后，在第三步中，工作节点 n 计算 $f(W_n)$，其中 f 是目标函数。

除了计算 W_n 和 $f(W_n)$ 外，工作节点 n 还会在第四步计算标签$_n = J(X_{1,\ n}, \ldots, X_{K,\ n})$，其中 J: $\mathbb{U} \to \mathbb{W}$ 是标签函数， $\mathbb{W}$ 是定义在 $\mathbb{F}$ 上的向量空间。我们将在第六节正式介绍标签函数。直观而言，标签函数会压缩其 K 个输入，并输出这些输入的指纹，因此，两组不同的K 输入所对应的输出几乎总是不同的。

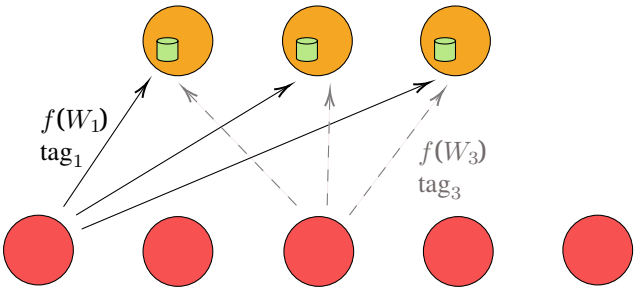
在第五步中，工作节点 n 向所有数据所有者发送标签$_n$ 和 $f(W_n)$。诚实数据所有者可以利用标签识别出在第一步中收到相同消息的工作节点，因此这些工作节点返回的计算结果是一致的。在第六步中，诚实数据所有者 $k \in [K]$ 使用 $\{f(W_n)\ n \in \mathcal{T}\} \cup \{$标签$_n, n \in \mathcal{T}\}$，其中 $\mathcal{T}$是 $[N]$中大小为 $t \in \mathbb{N}$的任意子集，以恢复 $f(X_k)$。图1展示了一个 Vers($f, N = 5, K = 3, \beta =$1$, v = 2$, Enc) 示例。

算法1 Vers 的工作流程

1: 诚实数据所有者 $k \in \mathcal{H}$ 向所有工作节点发送 X_k。对抗性数据所有者 $k \in \mathcal{A}$ 最多生成 $v \in \mathbb{N}$ 不同消息 $X_k^{(1)}, \ldots, X_k^{(v)}$，并将其中一个发送给每个工作节点。
2: 工作节点 $n \in [N]$ 将 $W_n \in \mathbb{U}$ 计算为从数据所有者接收到的 K 条消息的线性组合，使用编码算法 Enc，并根据 (1)。
3: 工作节点 $n \in [N]$ 计算 f({v3)。
4: 工作节点 $n \in [N]$ 计算标签$_n = J(X_{1,n}, \ldots, X_{K,n})$，其中 J: $\mathbb{U} \to \mathbb{W}$ 是一个标签函数，且 $\mathbb{W}$ 是一个向量空间 over $\mathbb{F}$。
5: 工作节点 $n \in [N]$ 向所有数据所有者发送标签$_n$ 和 $f(W_n)$。
6:诚实数据所有者 k 从 $\{f(W_n)\ n \in \mathcal{T}\} \cup \{$标签$_n$恢复 $f(X_k)\ n \in \mathcal{T}\}$，其中 $\mathcal{T}$是任意大小为 $t \in \mathbb{N}$ 的 $[N]$ 子集。



(a) 橙色节点（上排）是数据所有者，红色节点（下排）是工作节点。数据所有者将其数据发送给工作节点。此处，第一个数据所有者是对手，并向工作节点发送两条不同的消息 $X_1^{(1)}$ 和 $X_1^{(2)}$ 。



(b) 工作节点对编码输入应用 f ，计算标签，并返回将结果发送给数据所有者。

图 1：以 Vers($f, N = 5, K = 3, \beta = 1, v = 2$, Enc) 为例（仅显示了数据所有者与工作节点之间的部分消息）。最左侧的数据所有者是恶意的。

备注 1. 请注意，在 Vers 的第四步中，每个工作节点只计算一个标签。因此，Vers 的总通信量为 $O(N)$，即与 N呈线性关系。数据所有者利用这些标签提供的间接信息，将收到对抗性数据所有者发来的相同对抗性数据的工作节点划分为不同集合。另一种可能的设计是让每个工作节点针对从 K 个数据所有者处收到的每份数据分别计算一个标签，共计算 K 个标签。这些标签提供的直接信息会向诚实数据所有者揭示对抗性数据所有者的身份。然而，这种设计会导致 $O(KN)$的通信负载，并在 K 增长时妨碍可扩展性。如第二节所述，Vers 的主要目的之一是弥补单主节点系统在可扩展性方面的瓶颈。因此，在 Vers 中，我们采用通信高效的单标签方法，以便让 K 实现扩展。另一种折中的方案是让每个工作节点针对收到的 K 条消息计算固定数量的标签。这种方法可能会向诚实数据所有者提供更多有关对抗行为的信息。然而，由于该方法较为复杂，本文不讨论这一方案，并将其留待未来研究。

基本极限的正式定义如下。

定义 1（Vers 的基本极限）。Vers(f, N, K, β, v, Enc) 的基本极限记为 $t^*(f, N, K, \beta, v, \text{Enc})$，是指在任何对抗行为下，为使 $f(X_k)$ $k \in \mathcal{H}$ 能够从 $\{f(W_n)$ $n \in \mathcal{T}\}$ 和 $\{$标签$_n$、 $n \in \mathcal{T}\}$ 中正确计算所需的最小 t，其中 $\mathcal{T}$ 是 $[N]$ 中大小为 t 的任意子集。

无论对手如何选择并将其不同的 v 消息 $X_k^{(1)}, \ldots, X_k^{(v)}, k \in \mathcal{A}$ 分发给工作节点，$f(X_k)\ k \in \mathcal{H}$ 都应能够从 $\{f(W_n)\ n \in \mathcal{T}\}$ 和 {标签$_n, n \in \mathcal{T}$}中正确提取。对于大小为 $t^*(f, N, K, \beta, v,$Enc)的任意工作节点子集，这一要求都应成立，以使系统能够抵御掉队节点和节点故障；这类情况在分布式系统中很常见。请注意，我们并未要求能够从 $\{f(W_n)\ n \in \mathcal{T}\}$和 {标签$_n, n \in \mathcal{T}$}中找出所有或任意一个 $f(X_k^{(1)}) \ldots, f(X_k^{(v)})$。这是因为对手放弃了利用工作节点进行计算的机会，以便在系统中造成不一致。事实上，存在一些对抗行为，使得无法找出全部 $f(X_k^{(1)}) \ldots, f(X_k^{(v)})$。例如，考虑所有对抗性数据所有者都将 $X_k^{(1)}$ 发送给工作节点 $1, \ldots, N-1$，并将 $X_k^{(2)}$ 发送给工作节点 N 的情况。在这种情况下，无法从任何一组方程中获得足够的信息来找出 $f(X_k^{(2)})\ k \in \mathcal{A}$。因此，为了避免复杂的情形，我们基于仅需找出 $f(X_k)\ k \in \mathcal{H}$ 这一要求来定义基本极限。在第七节中，我们刻画 Vers($f, N, K, \beta, v,$ 拉格朗日编码) 的基本极限。换言之，我们在 Vers($f, N, K, \beta, v,$ Enc) 中选择拉格朗日编码作为Enc算法，相应地设置(1) 中的 $\gamma_{k,n}$，然后求出 $t^*(f, N, K, \beta, v,$ 拉格朗日编码)。

备注 2。Vers 的挑战部分是最后一步，即数据所有者对工作节点的结果进行解码，因为对抗性数据所有者在工作节点接收到的数据中注入了不一致性。在这项工作中，数据所有者在解码过程中处理不一致性，并使用标签来确定其结果一致的工作节点子集。另一种方法是在第一步就解决不一致性，即使用可靠广播算法，例如 [24], ，来广播每个数据所有者的消息，而不是让数据所有者简单地向工作节点发送消息。这种方法要求工作节点之间进行通信，以帮助它们就从每个数据所有者处接收到的数据达成共识。

IV. Vers 的应用

研究 Vers 的动机在于，我们可以利用它对存在对手的各种新兴计算系统进行建模，例如物联网网络和区块链。在本节中，我们将详细介绍 Vers 的一些应用。

在物联网网络中，由于传感器和设备的资源受限，它们无法处理自身产生的数据。因此，它们需要将计算卸载到外部节点，即 Vers中的工作节点，以计算其数据的复杂目标函数。因此，Vers 可轻松用于对资源受限设备组成的物联网网络进行建模。Vers中的数据所有者相当于物联网网络中的设备。Vers 中的工作节点相当于物联网网络中的外部计算节点。Vers中的目标函数相当于物联网网络中的设备所需的任何处理函数。例如，假设在农业领域，物联网用于智能灌溉。一些设备需要确定所需的适宜水量，该水量由土壤湿度、天气状况和环境温度、作物类型以及可能的其他特征决定。此类函数可以用 Vers 中的目标函数建模。

Vers 中的对抗性数据所有者相当于对抗性物联网设备，包括遭到篡改和感染的设备（例如，参见 [25]）。这些对抗性设备可能会故意向计算节点发送不一致的数据，以误导其他设备的决策过程。例如，假设物联网被用于部署

智能家居，且其中一些设备遭到篡改，并试图影响门上智能门锁的正常功能。由于物联网设备的资源受限，Vers 中对抗性数据所有者可发送的不同消息数量上限同样适用于这些设备。物联网网络的 Vers 模型的基本极限规定了系统要抵御对抗性设备所需的外部计算节点数量。

Vers还可用于对区块链，尤其是分片区块链进行建模。在一个拥有$N \in \mathbb{N}$ 个节点的区块链网络中，节点共同处理交易、生成交易区块、分发区块，并对其进行验证。区块链实际上就是经各方同意的区块链，每当节点就一个新区别达成共识时，该区块就会被追加到链上。在完全复制的区块链系统中，例如比特币，每个节点都会重复其他节点所做的工作。这种去中心化方法能够提供较高的安全性，以抵御对手，因为没有节点依赖于其他节点；但由于复制机制的存在，其吞吐量（每秒确认的交易数量）无法实现扩展。在过去几年中，人们围绕如何使区块链具备可扩展性开展了广泛研究 [26], [27]。其中一条值得关注的研究路线是分片，这一思想源自并行计算。最近，最受欢迎的区块链平台以太坊推出了 Danksharding [28], ，这是一种新颖的分片协议，并计划在不久的将来部署该协议。

在原生分片中，例如， [29], [30], 区块链节点会被划分为称为分片的多个组，例如 $M \in \mathbb{N}$ 个分片。分片并行运行，每个由 $\frac{N}{M}$ 个节点组成的分片都有自己的本地区块链，并生成自己的区块。节点可以随机分配到各个分片，也可以采用其他方法。分片的目的是将吞吐量提高 M 倍。然而，由 $\frac{N}{M}$ 个节点组成的分片，因此整个分片系统，与由 N 个节点组成的系统相比，更容易遭受安全性攻击。例如，与由 N 个节点组成的系统相比，攻击者更容易对由 $\frac{N}{M}$ 个节点组成的分片实施 51% 攻击。为了在保障安全性的同时扩展区块链，编码分片这一概念于 [31] 首次提出，并命名为 PolyShard。

与节点在一个分片中处理各自区块的传统分片不同，在 Ployshard 中，所有分片中的所有节点都处理编码区块。Polyshard 的运作方式如下。在第 $t \in \mathbb{N}$ 轮，每个分片 $m \in [M]$ 都需要验证其新区块 B_m^t。各分片将区块广播给所有节点。每个节点 n 使用拉格朗日编码，从接收到的区块 B_1^t、.....、B_M^t 中计算出一个编码区块 $\tilde{B}_n^t$ 。更具体地说，每个节点 n 构造一个拉格朗日多项式，其系数由 B_1^t、.....、B_M^t 构成，然后在点 α_n 处求值以得到 $\tilde{B}_n^t$。接着，节点对先前计算出的编码区块应用区块验证函数[1] g ，而不是分别对所有 M 个区块应用该函数 M 次。第 t 轮区块的验证需要第 1、2、.....、$t-1$ 轮区块的数据。每个节点 n 都存储每一轮区块编码后的链，并在点 α_n 处求值，而不只是存储其所属分片的区块。因此，在验证第 t 轮的编码区块时，它可以使用其存储的编码本地链，直到第 $t-1$ 轮。节点在网络中广播各自的验证结果。在此步骤中，对手可能会广播任意错误值。节点可以通过解码编码区块的验证结果，获得对 B_1^t、.....、B_M^t 的验证；对手的影响则会转化为解码数据时的错误。 [31] 的主要结论是，该方案的吞吐量和安全性[2] 随 N 线性扩展，而节点的存储需求则大发彩票

[1]区块验证函数由区块链系统的共识算法决定。[2]系统能够抵御的对抗节点数量。

无需扩展。

[31] 中的对抗模型仅限于这样的对手：它们广播错误值，而不是其编码区块的验证结果。即使在这种情况下，对手也会向所有节点发送相同的错误值，因此系统的一致性得以保持。然而，在区块链这类去中心化系统中，对抗节点可以自由偏离协议。在 [32], 中，我们介绍了针对 PolyShard 的差异攻击，其中对手控制部分分片，并且对抗分片向不同节点传播不同的区块。这会导致节点的编码区块出现错误，扰乱验证过程，并破坏 PolyShard 的安全性可扩展性。

Vers 是一个全面的系统，可以对存在对手时的编码分片区块链进行如下建模。Vers 中的数据所有者 $k \in [K]$ ，等价于区块链中分片 k 的代表节点，该代表节点在网络中传播其分片的新区块。请注意，在差异攻击下，分片 k的代表节点可能是对抗节点。Vers 中数据所有者 k 的数据 X_k ，等价于当前轮次中分片 k 的区块 B_k 。Vers 中的工作节点等价于区块链网络中的节点。Vers 中的目标函数 f 等价于区块链中的区块验证函数 g 。在 Vers 中，数据所有者 k 需要 f（X_k）；而在区块链中，分片k 的节点需要 B_k的验证函数结果。Vers 编码分片区块链模型的基本极限规定了系统在对抗分片存在的情况下保持鲁棒性所需的节点数量。

V. 主要结果

以下定理陈述了主要结果。

定理 1. Vers($f, N, K, \beta, v,$ Lagrange) 的基本极限为 $t^* = v^\beta d(K-1) +$ 1.

这一结果意味着 $N \geq v^\beta d(K-1) + 1$ 应当成立。否则，即使没有掉队节点或故障工作节点，且所有工作节点都将其计算结果发送回数据所有者，数据所有者也无法恢复$f(X_k)$ $k \in \mathcal{H}$。

由于基本极限是能够正确恢复$f(X_k)$ $k \in \mathcal{H}$所需的最少工作节点数量，因此定理 1 需要两个证明。首先，在逆向证明中，我们需要证明 t^* 确实是最小值，这意味着应存在一种特定的对抗行为，以及一组 t^*-1 工作节点，其结果无法唯一确定 $f(X_k)$ $k \in \mathcal{H}$ 。但这并不像证明一个方程组是欠定的那么简单。请注意，诚实数据所有者并不知道对抗行为，因此即使借助标签，他们也不知道如何根据工作节点的结果正确地列出方程。因此，诚实数据所有者需要考虑由工作节点的结果构成的每一种可能的方程组。此外，由于使用了拉格朗日编码， $f(X_k)$ $k \in \mathcal{H}$ 并非工作节点方程中的未知数，而是未知数的线性组合。在逆向证明中，我们证明存在一种对工作节点结果的可能解释，该解释会导致 $f(X_k)$ $k \in \mathcal{H}$至少有一个输出不正确。我们在第 VIII-A小节中给出逆向证明。

其次，在可达性证明中，我们需要证明任意一组 t^* 个工作节点都足以正确恢复$f(X_k)$ $k \in \mathcal{H}$。同样，无论存在多少种对工作节点结果的可能解释，这一点都应对任何对抗行为成立。我们在第 VIII-B 小节中给出可达性证明。

注 3. 当所有数据所有者都诚实时，Vers($f, N, K, \beta = 0, v$,Enc) 的基本极限与类似系统的基本极限并无不同；在该类似系统中，中心实体对数据进行编码，并将编码数据发送给工作节点，因为工作节点中的编码数据会完全相同。实际上，Vers面临的挑战源于对抗性数据所有者向系统中注入相互矛盾的数据，以及由此产生并通过工作节点的本地编码传播的错误。 [12] 中研究的编码计算系统包含一个中央编码器，但本质上与 Vers($f, N, K, \beta = 0, v$,Lagrange) 类似，其所谓的恢复阈值等价于 Vers($f, N, K, \beta = 0, v$, Lagrange) 的基本极限。因此，利用 [12], 的结果，我们知道 Vers($f, N, K, \beta = 0, v$, Lagrange) 的基本极限为 $t^*(f, N, K, \beta = 0, v$, Lagrange) $= d(K - 1) + 1$，其中 d 是目标多项式函数 f 的次数。

VI. 标签函数

在算法1的步骤2中，我们说明了工作节点会为其接收到的消息计算标签。本节将介绍标签的概念，并证明存在一个可部署于 Vers(f, N, K, β, v, Enc) 中的标签函数。

回顾一下，每个对抗性数据所有者 $k \in \mathcal{A}$ 都会将 $X_k^{(1)}, \ldots, X_k^{(v)}$ 分发给工作节点。在算法1的步骤5之后，从工作节点 $n \in [N]$ 收到 $f(W_n)$ 的诚实数据所有者并不知道该工作节点接收并用于 W_n 的是哪一条对手的 v 条消息。因此，诚实数据所有者不知道哪些工作节点在计算中使用了相同的一组输入消息。标签的作用是告知诚实数据所有者工作节点所使用的数据。标签使数据所有者能够将 $[N]$ 划分为若干集合，最多 v^β 个集合；每个集合中的工作节点都从第一步中的所有数据所有者处接收了相同的消息。

我们用 $X_\mathcal{H}$ 表示所有 X_k、$k \in \mathcal{H}$ 的串联，也就是诚实数据所有者的数据。另用 $X_{\mathcal{A},\ n}$ 表示所有 $X_{k,\ n}$ 的串联，也就是工作节点 $n \in [N]$ 收到的对抗性数据所有者的消息。工作节点 $n \in [N]$ 对 $X_{1,\ n}$、 ..... 、$X_{K,\ n}$ 应用标签函数 J，但由于对所有 $n \in [N]$ 都有 $\{X_{1,n}, \ldots, X_{K,n}\} = \{X_\mathcal{H}, X_{\mathcal{A},n}\}$，我们采用简化记法，写作 $J(X_\mathcal{H},\ X_{\mathcal{A},\ n})$。我们将标签函数正式定义如下。

定义2（标签函数）。函数 $J:\ \mathbb{U}^K \to \mathbb{U}^l,\ \ l < k$，若满足以下条件，则称为标签函数：对于两个对手，它们分别独立地从 $X_\mathcal{H}$ 中选取两个不同的 $X_{A,\ n_1}$ 和 $X_{A,\ n_2}$，并将它们发送给工作节点 n_1、$n_2 \in [N]$,

$$\Pr\big(J(X_\mathcal{H}, X_{\mathcal{A},n_1}) = J(X_\mathcal{H}, X_{\mathcal{A},n_2})\big) < \epsilon \tag{2}$$

成立，其中 $\epsilon \in (0, 1)$ 是一个可忽略值。

关于此定义，有几点值得注意。

- 式（2）中的概率是针对 $X_\mathcal{H}$ 的，因为诚实数据所有者的数据来自 $\mathbb{U}$ 上的独立同分布均匀分布。

• 如第三节所述，对手并不了解诚实数据所有者的数据。因此，他们无法基于对 $X_{\mathcal{H}}$的了解来选择X_{A,n_1}和X_{A,n_2}，而必须独立于 $X_{\mathcal{H}}$选择X_{A,n_1} 和X_{A,n_2}。

• 之所以要求条件 $l < k$，是因为我们希望标签轻量级且通信高效。如果不施加此类约束，每个工作节点 n 都可以将所有 $X_{k,\,n}$、 $k \in [K]$ 串联起来作为标签，数据所有者也可以很容易地利用这种标签发现工作节点之间的不一致性。然而，这种简单的解决方案完全不具备通信效率。我们需要一种标签函数，对每个工作节点接收到的 K 消息进行压缩，使对手造成的工作节点间差异能够被诚实数据所有者察觉。

假设我们有一个标签函数 J。任何数据所有者都可以比较两个工作节点 n_1、 $n_2 \in [N]$的标签 $\mathsf{tag}_{n_1} = J(X_{\mathcal{H}}, X_{\mathcal{A},\,n_1})$ 和 $\mathsf{tag}_{n_2} = J(X_{\mathcal{H}}, X_{\mathcal{A},\,n_2})$，并根据定义2得出结论：$\mathsf{tag}_{n_1} = \mathsf{tag}_{n_2}$ 意味着 n_1 和 n_2以高概率从所有数据所有者处接收到了相同的数据，即对于所有 $k \in [K]$，$X_{k,\,n_1} = X_{k,\,n_2}$。另一方面，当 $\mathsf{tag}_{n_1} \neq \mathsf{tag}_{n_2}$ 时，数据所有者显然可以确定，对于至少一个 $k \in \mathcal{A}$，$X_{k,\,n_1} \neq X_{k,\,n_2}$（这仅仅是因为 J 是一个函数）。因此，数据所有者可以利用标签将 $\mathcal{T}$划分为若干工作节点集合；集合中的工作节点在算法1的第一步从数据所有者处接收到了相同的数据，并在后续计算中使用了相同的初始数据。

标签函数是通用指纹函数的一种类型。指纹识别用于将任意大的数据映射为小且固定长度的摘要，作为其唯一标识符，适用于许多实际用途，例如避免比较庞大的数据。根据 [21], 一个 ϵ-指纹函数 $fp: \mathcal{K} \times \mathbb{F}^{\delta} \to \mathbb{F}^{\gamma}$ 满足

$$\max_{\substack{d,d' \in \mathbb{F}^{\delta} \\ d \neq d'}} \Pr\left[fp(r,d) = fp(r,d') : r \xleftarrow{R} \mathcal{K}\right] \leq \epsilon, \tag{3}$$

其中，数据是一个长度为 δ 的向量，其元素属于 $\mathbb{F}$，而 $r \in \mathcal{K}$ 是一个随机种子，$\mathcal{K} \subset \mathbb{R}$。一种众所周知的指纹函数是Rabin指纹 [33], 及其派生形式，它使用有限域中的随机多项式来生成指纹。另一类指纹函数是密码学哈希函数。我们知道，如果哈希函数 h 是抗碰撞的

$$\Pr\big((x_0, x_1) \leftarrow A, x_0 \neq x_1 : h(x_0) = h(x_1)\big) \leq \epsilon, \tag{4}$$

其中 A 是一个概率多项式对手（这只是一个粗略定义。正式定义请参阅 [34]）。我们不能将抗碰撞哈希函数用作标签函数，原因有二。

• 仅当 x_0 和 x_1 由概率多项式对手选择时，碰撞概率才较小。然而，标签函数必须能够抵御信息论对手。

• 根据 (2)，标签函数的输入有一个共同部分 $X_{\mathcal{H}}$，但 (4) 中的 x_0 和 x_1没有这样的约束。即使将抗碰撞性定义为 $\Pr\big((x_0, x_1, x_2) \leftarrow A : x_1 \neq x_2, h(x_0||x_1) = h(x_0||x_2)\big) \leq \epsilon$，这样的 h 也不能用作标签，因为对手可以是任意 β 数据所有者，而不一定是最后的 β 数据所有者。注意，(2) 中的 $J(X_{\mathcal{H}}, X_{\mathcal{A},\,n})$ 只是记号上的变通；实际上，工作节点 $n \in \mathbb{N}$会计算 $J(X_{1,\,n}, \ldots, X_{K,\,n})$，其中对抗性消息 β可以分散在任意位置。

在下　　根据定理，我们证明，在所有从 $\mathbb{U}^K$ 到 $\mathbb{U}$的函数中，存在一个标签函数　　n.

定理　　2. 存在一个标签函数 $J^*\colon \mathbb{U}^K \to \mathbb{U}$，当 $|\mathbb{U}|$ 足够大时，该函数满足 (2)。

该定理的证明与香农对离散无记忆信道容量的可达性证明类似，可见附录

注4. 标签有助于数据所有者对工作节点进行分组，但无法消除关于工作节点先前从对抗性数据所有者处收到的数据的所有歧义。例如，假设

$$\begin{aligned}
\mathsf{tag}_1 &= J^*(X_1, X_2^{(1)}, X_3^{(1)}),\\
\mathsf{tag}_2 &= J^*(X_1, X_2^{(1)}, X_3^{(1)}),\\
\mathsf{tag}_3 &= J^*(X_1, X_2^{(2)}, X_3^{(1)}),\\
\mathsf{tag}_4 &= J^*(X_1, X_2^{(2)}, X_3^{(2)}),
\end{aligned}$$

其中 J^* 是一个标签函数。诚实的数据所有者比较这些标签后发现，标签$_1$ = 标签$_2$，标签$_2$ ≠标签$_3$，标签$_2$ ≠标签$_4$，，而标签$_3$ ≠ 标签$_4$。因此，它得出结论：工作节点 1 和 2 以高概率从对手处收到了相同的数据，而工作节点 3 和 4 各自收到了不同的数据。该诚实的数据所有者无法得知工作节点 3 和 4 是否从第二个数据所有者处收到了相同的数据，即 $X_{2,\ 3} = X_{2,\ 4} = X_2^{(2)}$；也无法得知工作节点 2 和 3 是否从第三个数据所有者处收到了相同的数据，即 $X_{3,\ 2} = X_{3,\ 3} = X_3^{(1)}$。

VII. Analysis of Vers$(f, N, K, \beta, v, \text{Lagrange})$

本节将介绍一些工具，并利用它们来分析系统。这些工具有助于我们更好地理解系统，也将在下一节研究 Vers 的基本极限时发挥作用。下文将定义系数向量、单项式向量、特征矩阵和关系矩阵，它们均由与工作节点视角下每种可能的对抗行为相关的部分构成。换言之，本节介绍的概念涵盖了所有对抗行为。在下一节对系统基本极限的逆向证明中，我们将证明，在最坏情况下，工作节点视角下的所有对抗行为都可能出现在系统中，因此必须加以处理。

回顾一下，最多有 β 个对抗性数据所有者，他们可能向不同的工作节点发送不同的消息；而诚实的数据所有者则向所有工作节点发送同一条消息。如下文所定义，工作节点的版本向量指示该工作节点收到的每条对抗性消息的版本。

定义 3（版本向量）。对于工作节点 $n \in [N]$，我们定义一个长度为 K 的版本向量 $\mathbf{v} \in [v^+]^K$，其第 k 个元素（如果 $k \in \mathcal{A}$）位于 $[v]$ 中，并表示工作节点 n 接收到的对抗性数据所有者 $k \in \mathcal{A}$ 的消息版本。此外，如果 $k \in \mathcal{H}$，则 $\mathbf{v}$ 的第 k 个元素为 0，表示所有工作节点都从诚实数据所有者 k 接收到相同的消息。

换句话说，工作节点 n 从对抗性数据所有者 k 接收 $X_k^{(\mathbf{v}[k])}$。请注意，工作节点 n 不知道 $\mathbf{v}$，因为它既不知道对手，也不知道对抗性消息的版本。工作节点 n 接收来自数据所有者的 K 条消息，但对对抗性或诚实数据所有者一无所知，并对这些消息执行一些计算。工作节点 n 的版本向量等价于该工作节点所观察到的对抗行为。由于每个对抗性数据所有者至多有 v 条不同的对抗性消息，因此存在 v^β 个不同的版本向量。下面给出下一个定义。

值得注意的是，在定义3中，我们没有为工作节点 n 的版本向量使用下标 n，因为我们不会按照接收这些版本向量的工作节点来标识版本向量。相反，我们使用版本向量中的下标对其进行编号，正如下面的定义将会说明的那样。

定义 4（版本向量集合）。由 $\mathcal{V} \coloneqq \{\mathbf{v}_1, \mathbf{v}_2、\ldots、\mathbf{v}_{v^\beta}\} \subset [v^+]^K$ 表示的版本向量集合，是所有 v^β 可能的版本向量组成的集合。

根据第三节中描述的系统模型，每个工作节点都会按照式（1）计算所接收消息的线性组合。由于我们研究的是采用拉格朗日编码的 Vers，每个版本向量为$\mathbf{v}$的工作节点 $n \in [N]$ 在对来自数据所有者的所接收数据进行编码时，构造如下拉格朗日多项式。

$$q^{(\mathbf{v})}(z) \coloneqq \sum_{k\in\mathcal{A}} X_k^{(\mathbf{v}[k])} \prod_{j\neq k} \frac{z-\omega_j}{\omega_i-\omega_j} + \sum_{k\in\mathcal{H}} X_k \prod_{j\neq k} \frac{z-\omega_j}{\omega_i-\omega_j}, \tag{5}$$

其中， $\omega_k \in \mathbb{F}、\ k \in [K]$ 是分配给数据所有者的 K 个不同元素。这可以改写为

$$q^{(\mathbf{v})}(z) = \sum_{i=0}^{K-1} L_i(X_s, s\in\mathcal{H}, X_r^{(\mathbf{v}[r])}, r\in\mathcal{A}, f(.))z^i, \tag{6}$$

其中 $L_0, \ldots, L_{K-1}\colon \mathbb{F}^K \to \mathbb{F}$ 是线性映射。然后， $f(q^{(\mathbf{v})}(z))$可以改写为

$$f(q^{(\mathbf{v})}(z)) = \sum_{i=0}^{d(K-1)} u_i(X_s, s\in\mathcal{H}, X_r^{(\mathbf{v}[r])}, r\in\mathcal{A}, f(.))z^i, \tag{7}$$

其中 $u_0, \ldots, u_{d(K-1)}\colon \mathbb{F}^K \to \mathbb{F}$ 是关于 $X_r^{(\mathbf{v}[r])}, r\in\mathcal{A}$ 和 $X_s, s\in\mathcal{H}$的次数为 d 的多项式。工作节点 $n \in [N]$在 α_n处计算 $f(q^{(\mathbf{v})}(z))$，即 $f(W_n) = f(q^{(\mathbf{v})}(\alpha_n))$，其中 $\alpha_n, n \in [N]$ 是从 $\mathbb{F}$中分配给工作节点的 n 个元素。下面，我们定义系数向量和系数子向量，以便将工作节点的计算表示为矩阵乘法。

定义 5（系数向量）。设版本向量集合为 $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_{v^\beta}\}$。Vers$(f, N, K, \beta, v,$ Lagrange$)$ 系统的系数向量定义为

$$\mathbf{U}\big(X_s, s\in\mathcal{H}, X_r^{(i)}, r\in\mathcal{A}, i\in[v], f(.)\big) \coloneqq \begin{bmatrix} \mathbf{u}_{\mathbf{v}_1}(X_r^{(\mathbf{v}_1[r])}, X_s, r\in\mathcal{A}, s\in\mathcal{H}, f) \\ \hdashline \vdots \\ \hdashline \mathbf{u}_{\mathbf{v}_{v^\beta}}(X_r^{(\mathbf{v}_{v^\beta}[r])}, X_s, r\in\mathcal{A}, s\in\mathcal{H}, f) \end{bmatrix}, \tag{8}$$

其中 $\mathbf{u_v}\big(X_s, s\in\mathcal{H}, X_r^{(\mathbf{v}[r])}, r\in\mathcal{A}, f(.)\big)$，$\mathbf{v}\in\mathcal{V}$ 是一个系数子向量，定义为

$$\mathbf{u_v}\big(X_s, s\in\mathcal{H}, X_r^{(\mathbf{v}[r])}, r\in\mathcal{A}, f(.)\big) \coloneqq \begin{bmatrix} u_{d(K-1)}\big(X_s, s\in\mathcal{H}, X_r^{(\mathbf{v}[r])}, r\in\mathcal{A}, f(.)\big) \\ \vdots \\ u_0\big(X_s, s\in\mathcal{H}, X_r^{(\mathbf{v}[r])}, r\in\mathcal{A}, f(.)\big) \end{bmatrix}, \quad \mathbf{v}\in\mathcal{V}. \tag{9}$$

每个系数子向量的长度为 $d(K-1)+1$，因此系数向量 U 的长度为 $v^\beta(d(K-1)+1)$。如前所述，系数向量包含所有 v^β 可能的对抗行为。根据上述定义，显然，对于版本向量为 $\mathbf{v}\in\mathcal{V}$ 的工作节点 $n\in[N]$，

$$f(W_n) = \begin{bmatrix}\alpha^{d(K-1)} & \dots & \alpha & 1\end{bmatrix} \mathbf{u_v}\big(X_s, s\in\mathcal{H}, X_r^{(\mathbf{v}[r])}, r\in\mathcal{A}, f(.)\big). \tag{10}$$

因此，系数向量有助于我们恰当地表示工作节点的计算。这促使我们更深入地研究系数向量。以下示例将进一步阐明上述定义。

例1。考虑以下参数。

K	N	β	v	$\mathcal{A}$	$\mathcal{H}$	$(\omega_1,\omega_2,\omega_3)$	$f(x)$
3	3	1	2	$\{1,2\}$	$\{3\}$	(1, 2, 3)	x^2

由于 $v=2$，版本向量集合为 $\mathcal{V}=\{[1,1,0],[1,2,0],[2,1,0],[2,2,0]\}$。版本向量 $[i_1,i_2,0]\in\mathcal{V}$ 的拉格朗日多项式为

$$\begin{aligned} q^{([i_1,i_2,0])}(z) &= \frac{(z-2)(z-3)}{2}X_1^{(i_1)} + \frac{(z-1)(z-3)}{-1}X_2^{(i_2)} + \frac{(z-1)(z-2)}{2}X_3 = z^2\big(\frac{X_1^{(i_1)}}{2} - X_2^{(i_2)} + \frac{X_3}{2}\big) \\ &\quad + z\big(-\frac{5X_1^{(i_1)}}{2} + 2X_2^{(i_2)} - \frac{3X_3}{2}\big) + \big(3X_1^{(i_1)} - 3X_2^{(i_2)} + X_3\big),\ [i_1,i_2,0]\in\mathcal{V}. \end{aligned}$$

因此，

$$\begin{aligned} f(q^{([i_1,i_2,0])}(z)) &= \Big(\frac{(X_1^{(i_1)})^2}{4} - X_1^{(i_1)}X_2^{(i_2)} + \frac{X_1^{(i_1)}X_3}{2} + (X_2^{(i_2)})^2 - X_2^{(i_2)}X_3 + \frac{X_3^2}{4}\Big)z^4 + \Big(\frac{-5(X_1^{(i_1)})^2}{2} \\ &\quad + 9X_1^{(i_1)}X_2^{(i_2)} - 4X_1^{(i_1)}X_3 - 8(X_2^{(i_2)})^2 + 7X_2^{(i_2)}X_3 - \frac{3X_3^2}{2}\Big)z^3 + \Big(\frac{37(X_1^{(i_1)})^2}{4} - 29X_1^{(i_1)}X_2^{(i_2)} \\ &\quad + \frac{23X_1^{(i_1)}X_3}{2} + 22(X_2^{(i_2)})^2 - 17X_2^{(i_2)}X_3 + \frac{13X_3^2}{4}\Big)z^2 + \Big(-15(X_1^{(i_1)})^2 + 39X_1^{(i_1)}X_2^{(i_2)} \\ &\quad - 14X_1^{(i_1)}X_3 - 24(X_2^{(i_2)})^2 + 17X_2^{(i_2)}X_3 - 3X_3\Big)z + \Big(9(X_1^{(i_1)})^2 - 18X_1^{(i_1)}X_2^{(i_2)} + 6X_1^{(i_1)}X_3 \\ &\quad + 9(X_2^{(i_2)})^2 - 6X_2^{(i_2)}X_3 + X_3^2\Big). \end{aligned} \tag{11}$$

因此,

$$\mathbf{u}_{[i_1,i_2,0]}(X_1^{(i_1)},X_2^{(i_2)},X_3,f)=\begin{bmatrix}\frac{(X_1^{(i_1)})^2}{4}-X_1^{(i_1)}X_2^{(i_2)}+\frac{X_1^{(i_1)}X_3}{2}+(X_2^{(i_2)})^2-X_2^{(i_2)}X_3+\frac{X_3^2}{4}\\ \frac{-5(X_1^{(i_1)})^2}{2}+9X_1^{(i_1)}X_2^{(i_2)}-4X_1^{(i_1)}X_3-8(X_2^{(i_2)})^2+7X_2^{(i_2)}X_3-\frac{3X_3^2}{2}\\ \frac{37(X_1^{(i_1)})^2}{4}-29X_1^{(i_1)}X_2^{(i_2)}+\frac{23X_1^{(i_1)}X_3}{2}+22(X_2^{(i_2)})^2-17X_2^{(i_2)}X_3+\frac{13X_3^2}{4}\\ -15(X_1^{(i_1)})^2+39X_1^{(i_1)}X_2^{(i_2)}-14X_1^{(i_1)}X_3-24(X_2^{(i_2)})^2+17X_2^{(i_2)}X_3-3X_3\\ 9(X_1^{(i_1)})^2-18X_1^{(i_1)}X_2^{(i_2)}+6X_1^{(i_1)}X_3+9(X_2^{(i_2)})^2-6X_2^{(i_2)}X_3+X_3^2\end{bmatrix}, \tag{12}$$

and

$$\mathbf{U}(X_1^{(i_1)},X_2^{(i_2)},i_1,i_2\in[2],X_3,f)=\begin{bmatrix}\mathbf{u}_{[1,1,0]}(X_1^{(1)},X_2^{(1)},X_3,f)\\ \mathbf{u}_{[1,2,0]}(X_1^{(1)},X_2^{(2)},X_3,f)\\ \mathbf{u}_{[2,1,0]}(X_1^{(2)},X_2^{(1)},X_3,f)\\ \mathbf{u}_{[2,2,0]}(X_1^{(2)},X_2^{(2)},X_3,f)\end{bmatrix} \tag{13}$$

备注 5. 在定义 5 中，我们隐含地规定了式 (8) 中系数子向量的特定顺序。系数向量中第一个、也是最上方的子向量是 $\mathbf{u}_{\mathbf{v}_1}$，第二个是 $\mathbf{u}_{\mathbf{v}_2}$，依此类推，直到 $\mathbf{u}_{\mathbf{v}_{v^\beta}}$。在例1中，我们选择了 $\mathbf{v}_1=[1,1,0]$、$\mathbf{v}_2=[1,2,0]$、$\mathbf{v}_3=[2,1,0]$ 和 $\mathbf{v}_4=[2,2,0]$。我们也可以对 $\mathbf{v}_1$,$\mathbf{v}_2$,$\mathbf{v}_3$,$\mathbf{v}_4$, 采用其他赋值，因为不同赋值只会导致同一系数向量的不同排列。因此，系数向量中系数子向量的顺序可以自行选择，但一旦确定便保持不变。

系数向量中的每个元素都是若干单项式的和。例如，在例1的方程 (12) 中，$\mathbf{u}_{[i_1,i_2,0]}(X^{(i_1)}{}_1, X^{(i_2)}{}_2, X_3, f)$ 表明系数向量的元素由若干单项式构成。因此，为了更好地理解系数向量，我们定义一个新向量，其中包含出现在系数向量中的单项式。在定义单项式向量之前，我们需要定义函数的次数集合。单变量函数 $f:\mathbb{U}\to\mathbb{V}$ 的次数集合记为 $\mathcal{D}(f)$，其中包含 f 中单项式的次数。例如，$\mathcal{D}(f)=\{0,1,3\}$ 当 $f(x)=x^3+x+1$ 时。

定义 6（单项式向量）。Vers$(f,N,K,\beta,v,\text{Lagrange})$ 系统的单项式向量包含 X_k、$k\in\mathcal{H}$ 和 $X_k^{(i)}$、$k\in\mathcal{A}$、$i\in[v]$ 的所有 $e\in\mathcal{D}_f$ 次单项式，定义如下:

$$\mathbf{X}\big(X_s,s\in\mathcal{H},X_r^{(i)},r\in\mathcal{A},i\in[v],f(.)\big):=\left[\prod_{k\in\mathcal{A}}(X_k^{(\mathbf{v}[k])})^{i_k}\prod_{k\in\mathcal{H}}(X_k)^{j_k},\ \text{s.t.}\sum_{k\in\mathcal{A}}i_k+\sum_{k\in\mathcal{H}}j_k\in\mathcal{D}(f),\text{and }\mathbf{v}\in\mathcal{V}\right]^\intercal. \tag{14}$$

在下　　一个例子中，我们确定例　　1.

例 2。回顾例 1，$f(x)=x^2$，且 $\mathcal{D}(f)=\{2\}$。因此，单项式向量包含 X_k、$k\in\mathcal{H}$ 和 $X_k^{(i)}$、$k\in\mathcal{A}$、$i\in[v]$ 的所有二次单项式。特别地,

- 对于 $i_1=2$、$i_2=0$、$i_3=0$，我们得到单项式 $(X_{(1)\ 1})_2$、$(X_{(2)\ 1})_2$,
- 对于 $i_1=0,i_2=2,i_3=0$，我们得到单项式 $(X_{(1)\ 2})_2$、$(X_{(2)\ 2})_2$,
- 对于 $i_1=0,i_2=0,i_3=2$，我们得到单项式 $(X_3)_2$,

• 对于 $i_1 = 1,\ i_2 = 1,\ i_3 = 0$，我们得到单项式 $X_{(1)1}\ X_{(1)2},\ \ X_{(2)1}\ X_{(1)2},\ X_{(1)1}\ X_{(2)2},\ \ X_{(2)1}\ X_{(2)2}$，

对于 $i_1 = 1、\ i_2 = 0、\ i_3 = 1$，我们得到单项式 $X_{(1)\,1}\ X_3、\ X_{(2)\,1}\ X_3$，

对于 $i_1 = 0、\ i_2 = 1、\ i_3 = 1$，我们得到单项式 $X_{(1)\,2}\ X_3、\ X_{(2)\,2}\ X_3$。

因此，

$$\mathbf{X}\big(X_1^{(i_1)}, X_2^{(i_2)}, X_3, i_1, i_2 \in [2], f(x) = x^2\big) = \Big[(X_1^{(1)})^2,\ (X_1^{(2)})^2,\ (X_2^{(1)})^2,\ (X_2^{(2)})^2,\ (X_3)^2, X_1^{(1)}X_2^{(1)}, X_1^{(2)}X_2^{(1)},$$
$$X_1^{(1)}X_2^{(2)},\ X_1^{(2)}X_2^{(2)}, X_1^{(1)}X_3,\ X_1^{(2)}X_3, X_2^{(1)}X_3,\ X_2^{(2)}X_3\Big]^{\mathsf{T}}. \tag{15}$$

注释 6。请注意，单项式向量中单项式的顺序可以任意选择，但在后续所有方程中都保持不变。例如，我们也可以将式（15）中的 $\mathbf{X}\big(X_1^{(i_1)}, X^{(i_2)\,2}, X_3, i_1, i_2 \in [2], f(x) = x^2\big)$ 构造为

$$\Big[(X_1^{(2)})^2,\ (X_1^{(1)})^2,\ (X_2^{(2)})^2,\ (X_2^{(1)})^2,\ (X_3)^2, X_1^{(2)}X_2^{(1)},\ X_1^{(1)}X_2^{(1)},\ X_1^{(2)}X_2^{(2)}, X_1^{(1)}X_2^{(2)},$$
$$X_1^{(1)}X_3,\ X_1^{(2)}X_3, X_2^{(1)}X_3,\ X_2^{(2)}X_3\Big]^{\mathsf{T}}.$$

然而，我们选择了式 (15) 中的顺序，并将在示例 2 的后续内容中继续沿用该顺序。

引理 1. 在 Vers$(f, N, K, \beta, v,$ Lagrange$)$ 系统中，单项式向量 $\mathbf{X}\big(X_r^{(i)}, X_s, r \in \mathcal{A}, s \in \mathcal{H}, i \in [v]\big)$ 的长度为

$$\mathsf{Len}(\mathbf{X}) \coloneqq \sum_{e \in \mathcal{D}(f)} \sum_{m=0}^{\min(e,\beta)} \binom{\beta}{m} \binom{K - \beta - 1 + e}{e - m} v^m \tag{16}$$

证明。对于每个 $e \in \mathcal{D}(f)$，我们需要计算不同单项式 $\prod_{k\in\mathcal{A}}(X_k^{(\mathbf{v}[k])})^{i_k} \prod_{k\in\mathcal{H}}(X_k)^{j_k}$ 的数量，其中 $\sum_{k\in\mathcal{A}} i_k + \sum_{k\in\mathcal{H}} j_k = e$ and $\mathbf{v} \in \mathcal{V}$。我们按如下方式计数：

- 我们选择对抗性消息的一个子集 $\mathcal{M} \in \mathcal{A}、\ |\mathcal{M}| = m$ 放入单项式中，因此 $m \le \beta$ 且 $m \le e$，从而 $0 \le m \le \min(e, \beta)$。这使得式 (16) 中的项为 $\binom{\beta}{m}$。
- 单项式中的 m 对抗性消息有 v^m 个不同的版本值，因此得到式 (16) 中的项v^m 。

- 我们需要计算所选 m 对抗性消息和 $K - \beta$ 诚实消息的幂的可能情况数。这就是 $\sum_{k\in\mathcal{M}} i_k + \sum_{k\in\mathcal{H}} j_k = e$的解的数量，其中 $1 \le i_k$对应于 $k \in \mathcal{M}$，而 $0 \le j_k$对应于 $k \in \mathcal{H}$。根据组合学，我们知道此类解的数量为$\binom{K-\beta-1+e}{e-m}$。证明完毕。

□

现在单项式向量已定义，我们可以将系数向量中的每个元素表示为单项式向量中各单项式的线性组合。为此，我们定义特征矩阵。

定义 7（特征矩阵）。Vers($f, N, K, \beta, v,$ Lagrange) 系统的特征矩阵记为 M($\mathcal{H}, \mathcal{A}, v, f$)，用于说明系数向量与单项式向量之间的关系，即。

$$\mathbf{U}\big(X_s, s \in \mathcal{H}, X_r^{(i)}, r \in \mathcal{A}, i \in [v], f(.)\big) = \mathbf{M}\big(\mathcal{H}, \mathcal{A}, v, f\big)\mathbf{X}\big(X_s, s \in \mathcal{H}, X_r^{(i)}, r \in \mathcal{A}, i \in [v], f(.)\big). \tag{17}$$

特征矩阵的维度为 $\big(v^{\beta}(d(K-1)+1)\big) \times$ Len(X)。值得注意的是，与系数向量类似，特征矩阵包含所有的 v^{β} 对抗行为。

例3. 对于例1中介绍的 Vers 系统，我们可以通过使用（12）、（13）以及例2中的（15），轻松构造 M $\big(\mathcal{H}=\{3\}, \mathcal{A}=\{1,2\}, v=2, f(x)=x^2\big)$。所得特征矩阵如（19）所示。

我们可以将特征矩阵分解为

$$\mathbf{M}\big(\mathcal{H}, \mathcal{A}, v, f\big) = \begin{bmatrix} \mathbf{M}_{\mathbf{v}_1}\big(\mathcal{H}, \mathcal{A}, f\big) \\ \vdots \\ \mathbf{M}_{\mathbf{v}_{v^{\beta}}}\big(\mathcal{H}, \mathcal{A}, f\big) \end{bmatrix}, \tag{18}$$

其中 $\{\mathbf{v}_1, \ldots\ldots, \mathbf{v}_{v^{\beta}}\} = \mathcal{V}$。每个子矩阵$\mathbf{M}_{\mathbf{v}}\big(\mathcal{H}, \mathcal{A}, f\big)$、$\mathbf{v} \in \mathcal{V}$的维度为 $\big(d(K-1)+1\big) \times$Len(X)。实际上，子矩阵 $\mathbf{M}_{\mathbf{v}}$ 专用于系数子向量 $\mathbf{u}_{\mathbf{v}}\big(X_s, s \in \mathcal{H}, X_r^{(\mathbf{v}[r])}, r \in \mathcal{A}, f(.)\big)$，并将其元素表示为单项式向量元素的线性组合。因此，正如备注5中所讨论的，我们在系数向量中选择的版本向量顺序决定了特征矩阵中子矩阵的顺序。我们强调，这一顺序是可选的，但一经确定便保持不变。

定义 8.（关系矩阵）在 Vers($f, N, K, \beta, v,$ 拉格朗日）系统中，关系矩阵记为 $\mathbf{P}\big(\mathcal{H}, \mathcal{A}, v, f\big)$，由 $\mathbf{M}\big(\mathcal{H}, \mathcal{A}, v, f\big)$ 的左零空间的一组基构成，即

$$\mathbf{P}\big(\mathcal{H}, \mathcal{A}, v, f\big)\mathbf{M}\big(\mathcal{H}, \mathcal{A}, v, f\big) = \mathbf{0}. \tag{20}$$

关系矩阵的维数为 $\big(v^{\beta}(d(K-1)+1) - \text{rank(M)}\big) \times \big(v^{\beta}(d(K-1)+1)\big)$。根据定义 7，$\mathbf{P}\big(\mathcal{H}, \mathcal{A}, v, f\big)$ $\mathbf{M}\big(\mathcal{H}, \mathcal{A}, v, f\big) = \mathbf{0}$ 等价于

$$\mathbf{P}\big(\mathcal{H}, \mathcal{A}, v, f\big)\mathbf{U}\big(X_s, s \in \mathcal{H}, X_r^{(i)}, r \in \mathcal{A}, i \in [v], f(.)\big) = \mathbf{0}. \tag{21}$$

这就是我们将 $\mathbf{P}\big(\mathcal{H}, \mathcal{A}, v, f\big)$ 称为关系矩阵的原因，因为它揭示了系数向量各元素之间的线性关系。与特征矩阵类似，我们可以将关系矩阵分解为

$$\mathbf{P}\big(\mathcal{H}, \mathcal{A}, v, f\big) = \Big[\ \mathbf{P}_{\mathbf{v}_1}\big(\mathcal{H}, \mathcal{A}, f\big) \ \Big\vdots\ \ldots\ \Big\vdots\ \mathbf{P}_{\mathbf{v}_{v^{\beta}}}\big(\mathcal{H}, \mathcal{A}, f\big)\ \Big], \tag{22}$$

其中 $\{\mathbf{v}_1, \ldots, \mathbf{v}_{v^{\beta}}\} = \mathcal{V}$。每个子矩阵 $\mathbf{P}_{\mathbf{v}}\big(\mathcal{H}, \mathcal{A}, f\big), \mathbf{v} \in \mathcal{V}$ 的维度为 $\big(v^{\beta}(d(K-1)+1) - \text{rank(M)}\big) \times (d(K-1)+1)$。由式 (18) 和 (20)，

$$\mathbf{P}_{\mathbf{v}_1}\mathbf{M}_{\mathbf{v}_1} + \cdots + \mathbf{P}_{\mathbf{v}_{v^{\beta}}}\mathbf{M}_{\mathbf{v}_{v^{\beta}}} = \mathbf{0},$$

$$
\begin{bmatrix}
u_4(X_1^{(1)},X_2^{(1)},X_3)\\ u_3(X_1^{(1)},X_2^{(1)},X_3)\\ u_2(X_1^{(1)},X_2^{(1)},X_3)\\ u_1(X_1^{(1)},X_2^{(1)},X_3)\\ u_0(X_1^{(1)},X_2^{(1)},X_3)\\ \hdashline
u_4(X_1^{(1)},X_2^{(2)},X_3)\\ u_3(X_1^{(1)},X_2^{(2)},X_3)\\ u_2(X_1^{(1)},X_2^{(2)},X_3)\\ u_1(X_1^{(1)},X_2^{(2)},X_3)\\ u_0(X_1^{(2)},X_2^{(2)},X_3)\\ \hdashline
u_4(X_1^{(2)},X_2^{(1)},X_3)\\ u_3(X_1^{(2)},X_2^{(1)},X_3)\\ u_2(X_1^{(2)},X_2^{(1)},X_3)\\ u_1(X_1^{(2)},X_2^{(1)},X_3)\\ u_0(X_1^{(2)},X_2^{(1)},X_3)\\ \hdashline
u_4(X_1^{(2)},X_2^{(2)},X_3)\\ u_3(X_1^{(2)},X_2^{(2)},X_3)\\ u_2(X_1^{(2)},X_2^{(2)},X_3)\\ u_1(X_1^{(2)},X_2^{(2)},X_3)\\ u_0(X_1^{(2)},X_2^{(2)},X_3)
\end{bmatrix}
=
\left[\begin{array}{cccc:cccc:cccc}
\frac{1}{4} & 0 & 1 & 0 & \frac{1}{4} & -1 & 0 & 0 & 0 & \frac{1}{2} & 0 & -1 & 0\\
-\frac{5}{2} & 0 & -8 & 0 & -\frac{3}{2} & 9 & 0 & 0 & 0 & -4 & 0 & 7 & 0\\
\frac{37}{4} & 0 & 22 & 0 & \frac{13}{4} & -29 & 0 & 0 & 0 & \frac{23}{2} & 0 & -17 & 0\\
-15 & 0 & -24 & 0 & -3 & 39 & 0 & 0 & 0 & -14 & 0 & -17 & 0\\
9 & 0 & 9 & 0 & 1 & -18 & 0 & 0 & 0 & 6 & 0 & -6 & 0\\ \hdashline
\frac{1}{4} & 0 & 0 & 1 & \frac{1}{4} & 0 & 0 & -1 & 0 & \frac{1}{2} & 0 & 0 & -1\\
-\frac{5}{2} & 0 & 0 & -8 & -\frac{3}{2} & 0 & 0 & 9 & 0 & -4 & 0 & 0 & 7\\
\frac{37}{4} & 0 & 0 & 22 & \frac{13}{4} & 0 & 0 & -29 & 0 & \frac{23}{2} & 0 & 0 & -17\\
-15 & 0 & 0 & -24 & -3 & 0 & 0 & 39 & 0 & -14 & 0 & 0 & 17\\
9 & 0 & 0 & 9 & 1 & 0 & 0 & -18 & 0 & 6 & 0 & 0 & -6\\ \hdashline
0 & \frac{1}{4} & 1 & 0 & \frac{1}{4} & 0 & -1 & 0 & 0 & 0 & \frac{1}{2} & -1 & 0\\
0 & -\frac{5}{2} & -8 & 0 & -\frac{3}{2} & 0 & 9 & 0 & 0 & 0 & -4 & 7 & 0\\
0 & \frac{37}{4} & 22 & 0 & \frac{13}{4} & 0 & -29 & 0 & 0 & 0 & \frac{23}{2} & -17 & 0\\
0 & -15 & -24 & 0 & -3 & 0 & 39 & 0 & 0 & 0 & -14 & 17 & 0\\
0 & 9 & 9 & 0 & 1 & 0 & -18 & 0 & 0 & 0 & 6 & -6 & 0\\ \hdashline
0 & \frac{1}{4} & 0 & 1 & \frac{1}{4} & 0 & 0 & 0 & -1 & 0 & \frac{1}{2} & 0 & -1\\
0 & -\frac{5}{2} & 0 & -8 & -\frac{3}{2} & 0 & 0 & 0 & 9 & 0 & -4 & 0 & 7\\
0 & \frac{37}{4} & 0 & 22 & \frac{13}{4} & 0 & 0 & 0 & -29 & 0 & \frac{23}{2} & 0 & -17\\
0 & -15 & 0 & -24 & -3 & 0 & 0 & 0 & 39 & 0 & -14 & 0 & 17\\
0 & 9 & 0 & 9 & 1 & 0 & 0 & 0 & -18 & 0 & 6 & 0 & -6
\end{array}\right]
\begin{bmatrix}
(X_1^{(1)})^2\\ (X_1^{(2)})^2\\ \hdashline
(X_2^{(1)})^2\\ (X_2^{(2)})^2\\ \hdashline
(X_3)^2\\ \hdashline
X_1^{(1)}X_2^{(1)}\\ X_1^{(2)}X_2^{(1)}\\ X_1^{(1)}X_2^{(2)}\\ X_1^{(2)}X_2^{(2)}\\ \hdashline
X_1^{(1)}X_3\\ X_1^{(2)}X_3\\ \hdashline
X_2^{(1)}X_3\\ X_2^{(2)}X_3
\end{bmatrix}.
\tag{19}
$$

并且由式 (7),

$$\mathbf{P}_{\mathbf{v}_1}\mathbf{u}_{\mathbf{v}_1}+\cdots+\mathbf{P}_{\mathbf{v}_{v^\beta}}\mathbf{u}_{\mathbf{v}_{v^\beta}}=\mathbf{0}.$$

现在，我们介绍关系矩阵的有效排列和无效排列。

定义 9.（关系矩阵的有效排列）假设

$$\mathbf{P}(\mathcal{H},\mathcal{A},v,f)=\left[\begin{array}{c:c:c}\mathbf{P}_{\mathbf{v}_1}(\mathcal{H},\mathcal{A},f) & \cdots & \mathbf{P}_{\mathbf{v}_{v^\beta}}(\mathcal{H},\mathcal{A},f)\end{array}\right]$$

是 Vers(f,N,K,β,v, Lagrange) 系统的关系矩阵。关系矩阵的一个排列 $\Pi\colon [v^\beta]\to[v^\beta]$ 为 $\mathrm{P}^{\Pi}(\mathcal{H},\mathcal{A},v,f)=\left[\mathrm{P}_{\mathbf{v}_{\Pi(1)}}(\mathcal{H},\mathcal{A},f)\ \vdots\ \ldots\mathrm{P}_{\mathbf{v}_{\Pi(v^\beta)}}(\mathcal{H},\mathcal{A},f)\right]$；若 $\mathrm{P}^{\Pi}\mathrm{M}\neq\mathbf{0}$，则该排列有效；若 $\mathrm{P}^{\Pi}\mathrm{M}=\mathbf{0}$，则该排列无效。

这个定义背后的直觉来自特征矩阵的特殊结构，这一点在 (19) 中也显而易见。这个特殊结构将在下面的示例中变得清晰。

例4。回顾一下，在例（1）中介绍的 Vers 系统里，我们设置了 $\mathbf{v}_1=[1,1,0]$、$\mathbf{v}_2=[1,2,0]$、$\mathbf{v}_3=[2,1,0]$、$\mathbf{v}_4=[2,2,0]$ 和 $\mathrm{P}\big(\mathcal{H}=\{3\},\mathcal{A}=\{1,2\},v=2,f(x)=x^2\big)=[\mathrm{P}_{\mathbf{v}_1}\mathrm{P}_{\mathbf{v}_2}\mathrm{P}_{\mathbf{v}_3}\mathrm{P}_{\mathbf{v}_4}]$。我们展示。

在此　　Vers 系统中，P 有 3 个无效置换，分别为 Π_1、Π_2 和 Π_3，如下所示。

$$\mathbf{P}^{\Pi_1} = [\mathbf{P}_{\mathbf{v}_4}\ \mathbf{P}_{\mathbf{v}_3}\ \mathbf{P}_{\mathbf{v}_2}\ \mathbf{P}_{\mathbf{v}_1}] \tag{23}$$

$$\mathbf{P}^{\Pi_2} = [\mathbf{P}_{\mathbf{v}_3}\ \mathbf{P}_{\mathbf{v}_4}\ \mathbf{P}_{\mathbf{v}_1}\ \mathbf{P}_{\mathbf{v}_2}] \tag{24}$$

$$\mathbf{P}^{\Pi_3} = [\mathbf{P}_{\mathbf{v}_2}\ \mathbf{P}_{\mathbf{v}_1}\ \mathbf{P}_{\mathbf{v}_4}\ \mathbf{P}_{\mathbf{v}_3}] \tag{25}$$

换言之，$\mathrm{P}^{\Pi_1}\mathrm{M} = \mathrm{P}^{\Pi_2}\mathrm{M} = \mathrm{P}^{\Pi_3}\mathrm{M} = \mathbf{0}$，其中 M 由式（19）给出。考虑 Π_1，并证明 $\mathrm{P}^{\Pi_1}\mathrm{M} = \mathbf{0}$。另外两种排列的论证类似。将 M 的列 $i, 1 \le i \le 13$ 记作 $\mathbf{c_i} = [c_{i,1}^\intercal\ c_{i,2}^\intercal\ c_{i,3}^\intercal\ c_{i,4}^\intercal]^\intercal$，其中 $c_{i,\ 1}$,$c_{i,\ 2}$,$c_{i,\ 3}$,$c_{i,\ 4}$ 的长度为 5。我们知道 $\mathrm{P}c_i = \mathrm{P}_{\mathbf{v}_1}c_{i,\ 1} + \mathrm{P}_{\mathbf{v}_2}c_{i,\ 2} + \mathrm{P}_{\mathbf{v}_3}c_{i,\ 3} + \mathrm{P}_{\mathbf{v}_4}c_{i,\ 4} = \mathbf{0}$。下面考虑所有 $1 \le i \le 13$。

- $i = 1, i = 2$

  仔细观察式（19）中 M 的前两列可知，$c_{1,\ 1} = c_{1,\ 2} = c_{2,\ 3} = c_{2,\ 4} \coloneqq c^*_{\{1,2\}}$，且$c_{1,\ 3} = c_{1,\ 4} = c_{2,\ 1} = c_{2,\ 2} = \mathbf{0}$。因此

  $$\mathbf{P}\mathbf{c}_1 = \mathbf{P}_{\mathbf{v}_1}c_{1,1} + \mathbf{P}_{\mathbf{v}_2}c_{1,2} + \mathbf{P}_{\mathbf{v}_3}c_{1,3} + \mathbf{P}_{\mathbf{v}_4}c_{1,4} = \mathbf{0},$$

  等价于

  $$(\mathbf{P}_{\mathbf{v}_1} + \mathbf{P}_{\mathbf{v}_2})c^*_{\{1,2\}} = \mathbf{0}, \tag{26}$$

  and

  $$\mathbf{P}\mathbf{c}_2 = \mathbf{P}_{\mathbf{v}_1}c_{2,1} + \mathbf{P}_{\mathbf{v}_2}c_{2,2} + \mathbf{P}_{\mathbf{v}_3}c_{2,3} + \mathbf{P}_{\mathbf{v}_4}c_{2,4} = \mathbf{0},$$

  等价于

  $$(\mathbf{P}_{\mathbf{v}_3} + \mathbf{P}_{\mathbf{v}_4})c^*_{\{1,2\}} = \mathbf{0}. \tag{27}$$

  因此，对于式 (23) 中经过置换的关系矩阵，

  $$\mathbf{P}^{\Pi_1}\mathbf{c}_1 = \mathbf{P}_{\mathbf{v}_4}c_{1,1} + \mathbf{P}_{\mathbf{v}_3}c_{1,2} + \mathbf{P}_{\mathbf{v}_2}c_{1,3} + \mathbf{P}_{\mathbf{v}_1}c_{1,4} = (\mathbf{P}_{\mathbf{v}_4} + \mathbf{P}_{\mathbf{v}_3})c^*_{\{1,2\}},$$

  且由于 (27)，我们得出 $\mathrm{P}^{\Pi_1}\mathbf{c}_1 = \mathbf{0}$。同理，

  $$\mathbf{P}^{\Pi_1}\mathbf{c}_2 = \mathbf{P}_{\mathbf{v}_4}c_{2,1} + \mathbf{P}_{\mathbf{v}_3}c_{2,2} + \mathbf{P}_{\mathbf{v}_2}c_{2,3} + \mathbf{P}_{\mathbf{v}_1}c_{2,4} = (\mathbf{P}_{\mathbf{v}_2} + \mathbf{P}_{\mathbf{v}_1})c^*_{\{1,2\}},$$

  由于(26)，我们得出 $\mathrm{P}^{\Pi_1}\mathbf{c}_2 = \mathbf{0}$。
- $i = 3, i = 4$

  根据式 (19) 中的 M，$c_{3,\ 1} = c_{3,\ 3} = c_{4,\ 2} = c_{4,\ 4} \coloneqq c^*_{\{3,4\}}$，以及 $c_{3,\ 2} = c_{3,\ 4} = c_{4,\ 1} = c_{4,\ 3} = \mathbf{0}$。因此，

  $$\mathbf{P}\mathbf{c}_3 = \mathbf{P}_{\mathbf{v}_1}c_{3,1} + \mathbf{P}_{\mathbf{v}_2}c_{3,2} + \mathbf{P}_{\mathbf{v}_3}c_{3,3} + \mathbf{P}_{\mathbf{v}_4}c_{3,4} = (\mathbf{P}_{\mathbf{v}_1} + \mathbf{P}_{\mathbf{v}_3})c^*_{\{3,4\}} = \mathbf{0}, \tag{28}$$

and

$$\mathbf{P}\mathbf{c}_4 = \mathbf{P}_{\mathbf{v}_1} c_{4,1} + \mathbf{P}_{\mathbf{v}_2} c_{4,2} + \mathbf{P}_{\mathbf{v}_3} c_{4,3} + \mathbf{P}_{\mathbf{v}_4} c_{4,4} = (\mathbf{P}_{\mathbf{v}_2} + \mathbf{P}_{\mathbf{v}_4}) c^*_{\{3,4\}} = \mathbf{0}. \tag{29}$$

因此,

$$\mathbf{P}^{\Pi_1}\mathbf{c}_3 = \mathbf{P}_{\mathbf{v}_4} c_{3,1} + \mathbf{P}_{\mathbf{v}_3} c_{3,2} + \mathbf{P}_{\mathbf{v}_2} c_{3,3} + \mathbf{P}_{\mathbf{v}_1} c_{3,4} = (\mathbf{P}_{\mathbf{v}_4} + \mathbf{P}_{\mathbf{v}_2}) c^*_{\{3,4\}},$$

并且根据 (29)，我们得出 $\mathbf{P}^{\Pi_1}\mathbf{c}_3 = \mathbf{0}$。类似地,

$$\mathbf{P}^{\Pi_1}\mathbf{c}_4 = \mathbf{P}_{\mathbf{v}_4} c_{4,1} + \mathbf{P}_{\mathbf{v}_3} c_{4,2} + \mathbf{P}_{\mathbf{v}_2} c_{4,3} + \mathbf{P}_{\mathbf{v}_1} c_{4,4} = (\mathbf{P}_{\mathbf{v}_3} + \mathbf{P}_{\mathbf{v}_1}) c^*_{\{3,4\}},$$

并且根据 (28)，我们得出 $\mathbf{P}^{\Pi_1}\mathbf{c}_4 = \mathbf{0}$。

- $i = 5$

  由于根据 (19)，$c_{5,1} = c_{5,2} = c_{5,3} = c_{5,4} \coloneqq c^*_{\{5\}}$ 位于 M 的第五列中,

$$\begin{aligned}\mathbf{P}^{\Pi_1}\mathbf{c}_5 = \mathbf{P}_{\mathbf{v}_4} c_{5,1} + \mathbf{P}_{\mathbf{v}_3} c_{5,2} + \mathbf{P}_{\mathbf{v}_2} c_{5,3} + \mathbf{P}_{\mathbf{v}_1} c_{5,4} &= (\mathbf{P}_{\mathbf{v}_4} + \mathbf{P}_{\mathbf{v}_3} + \mathbf{P}_{\mathbf{v}_2} + \mathbf{P}_{\mathbf{v}_1}) c^*_{\{5\}} \\ &= \mathbf{P}_{\mathbf{v}_1} c_{5,1} + \mathbf{P}_{\mathbf{v}_2} c_{5,2} + \mathbf{P}_{\mathbf{v}_3} c_{5,3} + \mathbf{P}_{\mathbf{v}_4} c_{5,4} = \mathbf{P}\mathbf{c}_5 = \mathbf{0}.\end{aligned}$$

- $i = 6, i = 7, i = 8, i = 9$

  根据 (19) 中的 M，$c_{6,1} = c_{7,3} = c_{8,2} = c_{9,4} \coloneqq c^*_{\{6,7,8,9\}}$。因此,

$$\begin{aligned}\mathbf{P}^{\Pi_1}\mathbf{c}_6 &= \mathbf{P}_{\mathbf{v}_4} c_{6,1} + \mathbf{P}_{\mathbf{v}_3} c_{6,2} + \mathbf{P}_{\mathbf{v}_2} c_{6,3} + \mathbf{P}_{\mathbf{v}_1} c_{6,4} = \mathbf{P}_{\mathbf{v}_4} c^*_{\{6,7,8,9\}} = \mathbf{P}\mathbf{c}_9 = \mathbf{0}, \\ \mathbf{P}^{\Pi_1}\mathbf{c}_7 &= \mathbf{P}_{\mathbf{v}_4} c_{7,1} + \mathbf{P}_{\mathbf{v}_3} c_{7,2} + \mathbf{P}_{\mathbf{v}_2} c_{7,3} + \mathbf{P}_{\mathbf{v}_1} c_{7,4} = \mathbf{P}_{\mathbf{v}_2} c^*_{\{6,7,8,9\}} = \mathbf{P}\mathbf{c}_8 = \mathbf{0}, \\ \mathbf{P}^{\Pi_1}\mathbf{c}_8 &= \mathbf{P}_{\mathbf{v}_4} c_{8,1} + \mathbf{P}_{\mathbf{v}_3} c_{8,2} + \mathbf{P}_{\mathbf{v}_2} c_{8,3} + \mathbf{P}_{\mathbf{v}_1} c_{8,4} = \mathbf{P}_{\mathbf{v}_3} c^*_{\{6,7,8,9\}} = \mathbf{P}\mathbf{c}_7 = \mathbf{0}, \\ \mathbf{P}^{\Pi_1}\mathbf{c}_9 &= \mathbf{P}_{\mathbf{v}_4} c_{9,1} + \mathbf{P}_{\mathbf{v}_3} c_{9,2} + \mathbf{P}_{\mathbf{v}_2} c_{9,3} + \mathbf{P}_{\mathbf{v}_1} c_{9,4} = \mathbf{P}_{\mathbf{v}_1} c^*_{\{6,7,8,9\}} = \mathbf{P}\mathbf{c}_6 = \mathbf{0}.\end{aligned}$$

- $i = 10, i = 11$这种情况与 $i = 1$, $i = 2$ 完全相同。
- $i = 12, i = 13$这种情况与 $i = 3$, $i = 4$ 完全相同。

We 证明了在例1中，(23) 式里的 $\mathbf{P}^{\Pi_1}$ 是 P 的一个无效置换，即 $\mathbf{P}^{\Pi_1}\mathbf{M} = \mathbf{0}$.

注7. 关系矩阵 $\mathbf{P}^{\Pi}$的任意无效置换 $\Pi\colon \{1, \ldots, v^\beta\} \to \{1, \ldots, v^\beta\}$ 都属于关系矩阵 P 的行空间。根据定义，P 构成 M 的左零空间，因此 $\mathbf{P}^{\Pi}\mathbf{M} = \mathbf{0}$ 意味着 $\mathbf{P}^{\Pi}$ 应属于 P 的行空间。换言之，$\mathbf{P}^{\Pi}$ 的任意一行都是 P 各行的线性组合。

引理 2. 在一个 Vers(f, N, K, β, v, 拉格朗日)系统中，在所有 $(v^\beta)!$ 种排列中，关系矩阵有 $(v!)^\beta$ 种无效置换。

我们知道，关系矩阵有 v^β 个子矩阵，因此共有 $(v^\beta)!$ 种可能的排列

关系矩阵，包括有效和无效。此外，我们知道每个子矩阵都关联一个包含来自 $[v]$ 的 β 个元素的版本向量。每组 β 个元素有 $v!$ 种排列，总共有 $(v!)^\beta$ 种排列。该引理指出，只有这 $(v!)^\beta$ 种排列是无效的。该引理的证明见附录 B。

VIII. Fundamental Limit of Vers$(f, N, K, \beta, v, \text{Lagrange})$

在本节中，我们证明定理 1，并给出可达性证明和逆向证明。

A. 逆向证明

在此证明中，我们表明存在一种特定的对抗行为，对于该行为而言，来自 t 工作节点的 $t = v^\beta d(K-1)$ 条消息不足以正确找到 $f(X_k)$ $k \in \mathcal{H}$。换言之，我们表明，$\{f(W_n)$、标签 $_n$、 $n \in \mathcal{T}\}$，其中 $\mathcal{T} \subseteq [N]$，以及 $|\mathcal{T}| = v^\beta d(K-1)$，会使至少一个 $f(X_k)$ $k \in \mathcal{H}$ 存在多个可能的值。

考虑这样一种对抗行为： β 对抗性数据所有者串通，并以 $\mathcal{T}$ 的方式将其消息分发给工作节点，使得对于每个 $\mathbf{v} \in \mathcal{V}$，恰好有 $d(K-1)$ 个工作节点位于 $\mathcal{T}$，它们接收到的对抗性消息的版本符合 $\mathbf{v}$。

对于集合 $\mathcal{S} \subseteq \mathbb{F}$ 和整数 $D \in \mathbb{N}$，令 $\text{Van}_{\mathcal{S}}^{D}$ 为一个 $|\mathcal{S}| \times (D+1)$ 范德蒙德矩阵，它有 $|\mathcal{S}|$ 行，每行由 $\mathcal{S}$ 中一个元素的 0、1、. . .、 D 次幂组成。例如，

$$\text{Van}_{\{\alpha_1,\alpha_2\}}^{3} = \begin{bmatrix} \alpha_1^3 & \alpha_1^2 & \alpha_1 & 1 \\ \alpha_2^3 & \alpha_2^2 & \alpha_2 & 1 \end{bmatrix}.$$

步骤 1.假设诚实数据所有者 $k \in \mathcal{H}$ 从工作节点 $n \in \mathcal{T}$ 接收到 $y_n = f(W_n)$ 和标签$_n$。诚实数据所有者使用来自 $\mathcal{T}$ 中工作节点的标签，组成 v^β 个互不相交的集合 $\mathcal{T}_1, \ldots, \mathcal{T}_{v^\beta} \subseteq \mathcal{T}$，使得每个集合都包含标签相同、但与其他集合中工作节点的标签不同的工作节点。根据我们在此证明中考虑的对抗行为， $N_i := |\mathcal{T}_i| = d(K-1), i \in [v^\beta]$。标签函数的性质确保诚实数据所有者知道，工作节点 $n \in \mathcal{T}_i$ 和 $n' \in \mathcal{T}_j$ 使用的 β 条对抗性消息，至少有一条消息的版本不同，且这些工作节点来自不同集合。请注意，诚实数据所有者并不知道对抗性数据所有者。此外，诚实数据所有者无法知道哪些工作节点此前从对抗性数据所有者处接收过相同的消息，因而在其计算中使用了相同的消息；也无法知道哪些工作节点此前从对抗性数据所有者处接收过不同的消息，因而在其计算中使用了不同的消息。

步骤 2。令 $\mathbf{y}_i = [y_j]_{j \in \mathcal{T}_i}$，且 $\mathbf{Q}_i = \text{Van}_{\{\alpha_n, n \in \mathcal{T}_i\}}^{d(K-1)}$，对于 $i \in [v^\beta]$。存在一个底层映射 $\phi : [v^\beta] \to [v^\beta]$，它表示 $\mathcal{T}_1$，.....， $\mathcal{T}_{v^\beta}$ 中的工作节点此前从对抗性数据所有者处收到的对抗性消息。具体而言， $\phi(i) = j$ 表示 $\mathbf{y}_i = \mathbf{Q}_i \mathbf{u}^{\mathbf{v}_j}$，其中 i、 $j \in [v^\beta]$ 和 $u_{\mathbf{v}_j}$， $\mathbf{v}_j \in \mathcal{V}$ 是式 (9) 中定义的系数子向量。换言之，$\mathbf{y}_i = \mathbf{Q}_i \mathbf{u}^{\mathbf{v}_j}$ 表示工作节点 $n \in \mathcal{T}_i$ 使用的对抗性消息，其版本属于 $\mathbf{v}_j$。不失一般性，假设 ϕ 是恒等排列，即对于所有 $i \in [v^\beta]$， $\mathbf{y}_i = \mathbf{Q}_i \mathbf{u}^{\mathbf{v}_i}$。

诚实数据所有者 k 将系数向量 $\mathbf{U}'$ 视为未知量，并尝试求解方程以求出 $\mathbf{U}'$。然而，诚实数据所有者并不知道 ϕ，因此需要考虑 ϕ的所有不同可能情况，并确保

所有情况都能得到唯一的 $f(X_k)$。假设诚实数据所有者考虑一个有效排列Π：$[v^\beta] \to [v^\beta]$，并假设 $\mathbf{y}_i = \mathbf{Q}_i \mathbf{u}'_{\mathbf{v}_{\Pi(i)}}$。因此，

$$\mathbf{Q}_i(\mathbf{u}_{\mathbf{v}_i} - \mathbf{u}'_{\mathbf{v}_{\Pi(i)}}) = \mathbf{0}, \quad i \in [v^\beta]. \tag{30}$$

步骤 3。回顾一下，对于任意 $\mathbf{v} \in \mathcal{V}$，$f(X_k) = f(q^{(\mathbf{v})}(\omega_k))$，$k \in \mathcal{H}$，因此

$$[f(X_k)]_{k\in\mathcal{H}} = \left(\mathrm{Van}^{d(K-1)+1}_{\{\omega_k, k\in\mathcal{H}\}}\right)\mathbf{u}_{\mathbf{v}_i}, \quad i \in [v^\beta] \tag{31}$$

因此，$\mathrm{Van}^{d(K-1)+1}_{\{\omega_k,k\in\mathcal{H}\}}\mathbf{u}_{\mathbf{v}_i}$ 在 $i \in [v^\beta]$ 下的所有取值都相等。类似地，$\mathrm{Van}^{d(K-1)+1}_{\{\omega_k,k\in\mathcal{H}\}}\mathbf{u}'_{\mathbf{v}_{\Pi(i)}}$ 在 $i \in [v^\beta]$ 下的所有取值也相等。我们将证明，任何有效排列 Π 都会使 $\mathrm{Van}^{d(K-1)+1}_{\{\omega_k,k\in\mathcal{H}\}}\mathbf{u}_{\mathbf{v}_i}$ 与 $\mathrm{Van}^{d(K-1)+1}_{\{\omega_k,k\in\mathcal{H}\}}\mathbf{u}'_{\mathbf{v}_{\Pi(i)}}$ 的取值不同，$i,\ j \in [v^\beta]$。如果存在一个 $i^* \in [v^\beta]$ 使得

$$\mathrm{Van}^{d(K-1)+1}_{\{\omega_k,k\in\mathcal{H}\}}(\mathbf{u}_{\mathbf{v}_{i^*}} - \mathbf{u}'_{\mathbf{v}_{\Pi(i^*)}}) = \mathbf{0}, \tag{32}$$

那么我们可以得出结论：

$$[f(X_k)]_{k\in\mathcal{H}} = \left(\mathrm{Van}^{d(K-1)+1}_{\{\omega_k, k\in\mathcal{H}\}}\right)\mathbf{u}'_{\mathbf{v}_{\Pi(i)}}, \quad i \in [v^\beta], \tag{33}$$

因为

$$[f(X_k)]_{k\in\mathcal{H}} = \left(\mathrm{Van}^{d(K-1)+1}_{\{\omega_k, k\in\mathcal{H}\}}\right)\mathbf{u}_{\mathbf{v}_{i^*}} \tag{34}$$

and

$$\left(\mathrm{Van}^{d(K-1)+1}_{\{\omega_k, k\in\mathcal{H}\}}\right)\mathbf{u}'_{\mathbf{v}_{\Pi(i)}} = \left(\mathrm{Van}^{d(K-1)+1}_{\{\omega_k, k\in\mathcal{H}\}}\right)\mathbf{u}'_{\mathbf{v}_{\Pi(i^*)}}, \quad i \in [v^\beta]. \tag{35}$$

换句话说，如果存在这样的 i^*，那么排列 Π 会为 $f(X_k)$ $k \in \mathcal{H}$ 得到正确答案。通过反证法，我们假设存在这样的 i^*。因此，

$$\begin{aligned} \mathbf{Q}_i(\mathbf{u}_{\mathbf{v}_i} - \mathbf{u}'_{\mathbf{v}_{\Pi(i)}}) &= \mathbf{0}, \quad i \in [v^\beta], \\ \left(\mathrm{Van}^{d(K-1)+1}_{\{\omega_k, k\in\mathcal{H}\}}\right)(\mathbf{u}_{\mathbf{v}_i} - \mathbf{u}'_{\mathbf{v}_{\Pi(i)}}) &= \mathbf{0}, \quad i \in [v^\beta]. \end{aligned} \tag{36}$$

矩阵 $\mathbf{Q}_i$ 仅包含 α_n、$n \in \mathcal{T}_i$ 的幂，$\mathrm{Van}^{d(K-1)+1}_{\{\omega_k,k\in\mathcal{H}\}}$ 包含 ω_k、$k \in \mathcal{H}$ 的幂，并且这些元素彼此不同，且均从 $\mathbb{F}$ 中均匀随机选取。因此，矩阵

$$\left[\begin{array}{c} \mathbf{Q}_i \\ \hline \mathrm{Van}^{d(K-1)+1}_{\{\omega_k,k\in\mathcal{H}\}} \end{array}\right]$$

是 MDS，其维数为 $(d(K-1)+h) \times (d(K-1)+1)$，其秩为 $d(K-1)+1$。$\mathbf{u}_{\mathbf{v}_i} - \mathbf{u}'_{\mathbf{v}_{\Pi(i)}}$的长度为 $d(K-1)+1$，因此，(36) 中任意 $d(K-1)+1$ 个方程都会得出

$$\mathbf{u}_{\mathbf{v}_i} = \mathbf{u}'_{\mathbf{v}_{\Pi(i)}}, \quad i \in [v^\beta]. \tag{37}$$

步骤 4。我们知道 $\mathrm{PU} = 0$ 和 $\mathrm{PU}' = 0$，因此，

$$\mathbf{P}_{\mathbf{v}_1}\mathbf{u}_{\mathbf{v}_1} + \cdots + \mathbf{P}_{\mathbf{v}_{v^\beta}}\mathbf{u}_{\mathbf{v}_{v^\beta}} = \mathbf{0}, \tag{38}$$

$$\mathbf{P}_{\mathbf{v}_{\Pi(1)}}\mathbf{u}'_{\mathbf{v}_{\Pi(1)}} + \cdots + \mathbf{P}_{\mathbf{v}_{\Pi(v^\beta)}}\mathbf{u}'_{\mathbf{v}_{\Pi(v^\beta)}} = \mathbf{0}. \tag{39}$$

我们将（37）代入（39），

$$\mathbf{P}_{\mathbf{v}_{\Pi(1)}}\mathbf{u}_{\mathbf{v}_1} + \cdots + \mathbf{P}_{\mathbf{v}_{\Pi(v^\beta)}}\mathbf{u}_{\mathbf{v}_{v^\beta}} = \mathbf{0}, \tag{40}$$

回顾一下， Π 是一个有效排列，因此根据定义， $[\mathrm{P}^{\mathbf{v}_{\Pi(1)}} \cdots \mathrm{P}_{\mathbf{v}_{\Pi(v^\beta)}}]$ 不属于 P 的行空间。因此，方程（40）对 U 以及数据所有者的所有消息施加了额外约束。然而，无论是诚实数据所有者还是对抗性数据所有者，其消息都可以取任意值，对这些消息施加任何约束都会导致矛盾。至此，逆向证明完成。

B. 可达性证明

在可达性证明中，我们假设诚实数据所有者 $k \in \mathcal{H}$ 从 t^* 个工作节点接收 $t^* = v^\beta d(K-1)+1$ 条消息。诚实数据所有者利用收到的标签，将工作节点划分为 v^β 个互不相交的集合，使每个集合中的工作节点标签相同，而不同集合中的工作节点标签各不相同。由于 $t^* > v^\beta d(K-1)$，这些集合中必有一个至少包含 $d(K-1)$ 个 $+1$ 工作节点，我们将其称为 $\mathcal{T}^*$。存在 $i^* \in [v^\beta]$，使得

$$\left(\mathrm{Van}^{d(K-1)}_{\{\alpha_n, n\in\mathcal{T}^*\}}\right)\mathbf{u}_{\mathbf{v}_{i^*}} = [y_j]_{j\in\mathcal{T}^*}. \tag{41}$$

该方程唯一确定 $\mathrm{u}^{\mathbf{v}_{i^*}}$。随后，诚实数据所有者可以求出 $f(X_k) = \left(\mathrm{Van}^{d(K-1)}_{\omega_k}\right)\mathrm{u}^{\mathbf{v}_{i^*}}$。证明完毕。

定理 1 的一个假设是，工作节点是诚实的，并将 $f(W_n)$、标签$_n$、 $n \in [N]$ 的正确值发送给数据所有者。假设 $a \in \mathbb{N}$、 $a \le N$个工作节点是恶意的，并可能向数据所有者发送错误的 $f(W_n)$ 或标签$_n$。错误的标签值会造成不利影响，因为它们会误导数据所有者进行分区。沿用逆向证明中的记号，假设 $\mathrm{y}_i = \mathrm{Q}_i\mathrm{u}_{\mathbf{v}_i} + \mathrm{e}_i$ 是真实值，而 $\mathrm{y}_i = \mathrm{Q}_i\mathrm{u}'_{\Pi(\mathbf{v}_i)} + \mathrm{e}'_i$ 是诚实数据所有者因不知道真实值而作出的假设，且 e_i、e'_i 是误差向量。因此，在存在对抗性工作节点的情况下，方程（30）不再成立，之前逆向证明中使用的方法也无法在此处使用。不过，根据编码理论，我们知道来自 $t = v^\beta d(K-1) + 1 + 2a$ 个工作节点的结果足以正确解码 $f(X_k)$ $k \in \mathcal{H}$。

附录 A 定理 2 的证明

为了证明标签函数存在，我们将所有从 $\mathbb{U}^K$ 到 $\mathbb{U}$ 的 J 函数视为等概率，并计算任意两个工作节点生成相同标签这一事件的平均概率。考虑两个工作节点 n_1、 $n_2 \in [N]$，它们从对抗性数据所有者处接收到不同的 $X_{\mathcal{A},\ n_1}$ 和 $X_{\mathcal{A},\ n_2}$，并从诚实

数据所有者处接收到 。我们证明，对所有 J 函数以及诚实数据所有者的消息取平均后，标签$_{n_1}$ = 标签$_{n_2}$ 的概率非常小。由此可得，存在一个函数，使得对诚实数据所有者的消息取平均后，标签$_{n_1}$ =标签$_{n_2}$ 的概率非常小，这意味着标签函数存在。

任意函数 J 从 $\mathbb{U}^K$ 到 $\mathbb{U}$，即从一个大空间映射到一个小空间的函数，会将某些不同的输入映射到相同的输出。对于函数 J，以及两条对抗性消息$X_{\mathcal{A},\ n_1}$ 和 $X_{\mathcal{A},\ n_2}$，我们定义函数 g 使得$\mathcal{S}:=g(J,X_{\mathcal{A},\ n_1},X_{\mathcal{A},\ n_2}) \subseteq \mathbb{U}^h$ 是满足以下条件的集合

$$X_{\mathcal{H}} \in \mathcal{S} : J(X_{\mathcal{H}}, X_{\mathcal{A},n_1}) = J(X_{\mathcal{H}}, X_{\mathcal{A},n_2}),$$
$$X_{\mathcal{H}} \in \mathbb{U}^h \setminus \mathcal{S} : J(X_{\mathcal{H}}, X_{\mathcal{A},n_1}) \neq J(X_{\mathcal{H}}, X_{\mathcal{A},n_2}).$$

由于诚实数据所有者的数据是从 $\mathbb{U}^h$ 中均匀且独立选取的，且与$X_{\mathcal{A},n_1}$ 和 $X_{\mathcal{A},n_2}$相互独立，

$$\Pr\big(J(X_{\mathcal{H}}, X_{\mathcal{A},n_1}) = J(X_{\mathcal{H}}, X_{\mathcal{A},n_2})\big) = \frac{|\mathcal{S}|}{\mathbb{U}^h}, \tag{42}$$

其中 $\mathcal{S} = g(J,X_{\mathcal{A},n_1}, X_{\mathcal{A},n_2}) \subseteq \mathbb{U}^h$。由于所有 J函数都是等概率的，即 $\Pr(J) = \frac{1}{|\mathbb{U}|^K}$，

$$\sum_{J:g(J,X_{\mathcal{A},n_1},X_{\mathcal{A},n_2})=\mathcal{S}} \Pr(J) = (1-\frac{1}{|\mathbb{U}|})^{(|\mathbb{U}|^h-|\mathcal{S}|)}(\frac{1}{|\mathbb{U}|})^{|\mathcal{S}|}. \tag{43}$$

利用式 (42) 和式 (43),

$$\begin{aligned}
\Pr(\mathsf{tag}_{n_1} = \mathsf{tag}_{n_2}) &= \sum_{\mathcal{S}} \sum_{J:g(J,X_{\mathcal{A},n_1},X_{\mathcal{A},n_2})=\mathcal{S}} \Pr(J)\Pr\big(J(X_{\mathcal{H}}, X_{\mathcal{A},n_1}) = J(X_{\mathcal{H}}, X_{\mathcal{A},n_2})\big) \\
&= \sum_{\mathcal{S}} (1-\frac{1}{|\mathbb{U}|})^{(|\mathbb{U}|^h-|\mathcal{S}|)}(\frac{1}{|\mathbb{U}|})^{|\mathcal{S}|}\frac{|\mathcal{S}|}{\mathbb{U}^h} \\
&= \sum_{s=1}^{|\mathbb{U}|^h} \binom{|\mathbb{U}|^h}{s} (1-\frac{1}{|\mathbb{U}|})^{(|\mathbb{U}|^h-s)}(\frac{1}{|\mathbb{U}|})^{s}\frac{s}{\mathbb{U}^h} \\
&= \sum_{s=1}^{|\mathbb{U}|^h} \binom{|\mathbb{U}|^h-1}{s-1} (1-\frac{1}{|\mathbb{U}|})^{(|\mathbb{U}|^h-s)}(\frac{1}{|\mathbb{U}|})^{s} = \frac{1}{|\mathbb{U}|}
\end{aligned} \tag{44}$$

如果 $\mathbb{U}$ 足够大，使得$\frac{1}{\epsilon} \leq |\mathbb{U}|$，则 $\Pr$(标签$_{n_1}$ = 标签$_{n_2}$) $\leq \epsilon$，证明完毕。 e.

APPENDIX B
PROOF OF LEMMA 2

在本附录中，我们证明关系矩阵有$(v!)^\beta$ 个无效置换。在第七节中，我们考察了一个示例，并证明了其关系矩阵的一个排列是无效的。理解该示例有助于理解本节中的证明。

我们证明，形式为 $\Pi = \pi_1 \times \pi_2 \times \cdots \times \pi_\beta$ 的关系矩阵的所有排列均为无效排列，即 $\mathbf{P}^{\Pi}\mathbf{M} = \mathbf{0}$，其中 $\pi_i : [v] \to [v], i \in [\beta]$ 是 $[v]$上的一个排列。我们将证明分解为以下步骤。

步骤 1

考虑M 的第 $i \in [L]$ 列，$\mathbf{c}i = [c^{\intercal}{}_{\mathbf{v}1,i} \cdots c^{\intercal}{}_{\mathbf{v}^{v^\beta},i}]^{\intercal}$，其中每个所谓子列的长度为$c_{\mathbf{v}_i}$、$i \in [v^\beta]$，即 $d(K-1)+1$，$\{\mathbf{v}1、..、\mathbf{v}_{v^\beta}\} = \mathcal{V}$，且 L 是单项式向量 X 的长度。

子列 $c_{\mathbf{v},\ i}$，$\mathbf{v} \in \mathcal{V}, i \in [L]$，包含系数子向量 $\mathbf{u_v}$ 中单项式 $\mathbf{X}[i]$ 的系数。如果系数子向量 $\mathbf{u_v}$ 中没有单项式 $\mathbf{X}[i]$，那么 $c_{\mathbf{v},\ i} = \mathbf{0}$。对于两个都含有 $\mathbf{X}[i]$ 的系数子向量 $\mathbf{u_v}$ 和 $\mathbf{u_{v'}}$，$c_{\mathbf{v},\ i} = c_{\mathbf{v}',\ i} := c_i^*$，因为这些系数子向量仅在对抗性消息上有所不同，而常数系数相同。因此，M 的一列中的非零子列相同。这在式 (19) 中例1的 M 矩阵中显而易见。

我们将 M 的两个邻列定义如下。考虑两列 i、$i' \in [L]$，其中单项式 $\mathbf{X}[i]$ 和 $\mathbf{X}[i']$ 中的对抗性消息与诚实消息相同，但其中对抗性消息的版本不同。例如，(19) 中的 $i = 6$、$i' = 7$ 分别对应单项式 $X_1^{(1)}X_2^{(1)}$ 和 $X_1^{(2)}X_2^{(1)}$；它们的 X_1 版本不同，但都由相同的消息 X_1 和 X_2 构成。对于两个邻列，$c_i^* = c_{i'}^*$，即两个邻列中的非零子列相同。例如，(19) 中第 6 列和第 7 列的非零子列相同。

步骤 2

令 $\mathcal{V}_{\mathbf{c}_i}$、$i \in [L]$为系数子向量的版本向量集合，这些系数子向量包含单项式$\mathbf{X}[i]$，换句话说，即与 $\mathbf{c}_i$ 的非零子列相对应的版本向量。例如，在例1中，利用(13)和 (19)，我们得到 $\mathcal{V}_{\mathbf{c}_1} = \{[1,1,0],[1,2,0]\}$、$\mathcal{V}_{\mathbf{c}_2} = \{[2,1,0],[2,2,0]\}$、$\mathcal{V}_{\mathbf{c}_3} = \{[1,1,0],[2,1,0]\}$，依此类推。根据上一步，我们知道$c_{\mathbf{v},i} = c_i^*$ 对于 $\mathbf{v} \in \mathcal{V}_{\mathbf{c}_i}$。对于 M 的每一列 $\mathbf{c}_i$，有两种可能的情况。

- 情形1。单项式 $\mathbf{X}[i]$ 仅由诚实消息组成，例如（X_3）^{2}。在这种情况下，$\mathbf{c}_i$ 的所有子列均相等且非零，即 $c_{\mathbf{v}1,i} = \cdots = c_{\mathbf{v}^{v^\beta},i} := c_i^*$，且 $\mathcal{V}_{\mathbf{c}_i} = \mathcal{V}$。在例 1 中，式（19）中 M 的 $\mathbf{c}_5$ 就属于这种情况，因为

$$c_{\mathbf{v}_1,5} = c_{\mathbf{v}_2,5} = c_{\mathbf{v}_3,5} = c_{\mathbf{v}_4,5} = \begin{bmatrix} \frac{1}{4} \\ -\frac{3}{2} \\ \frac{13}{4} \\ -3 \\ 1 \end{bmatrix} \tag{45}$$

- 情况 2。单项式 $\mathbf{X}[i]$ 包含对抗性消息，例如 $X[i] = (X_{(1)\ 1})^2$。在这种情况下，版本向量 $\mathbf{v} \in \mathcal{V}_{\mathbf{c}_i}$ 对于 $X[i]$ 中的对抗性消息具有相同的版本值。例如，$\mathcal{V}_{\mathbf{c}_1} = \{[1,1,0],[1,2,0]\}$，以及 $\mathcal{V}_{\mathbf{c}_1}$ 中的版本向量表示 X_1 的版本(1)，因为 $X[i] = (X_{(1)\ 1})^2$。再例如，对于 (19) 中的$X[11] = X_{(2)_1}\ X_3$，$\mathcal{V}_{\mathbf{c}_{11}} = \{[2,1,0],[2,2,0]\}$，以及 $\mathcal{V}_{\mathbf{c}_{11}}$ 中的版本向量表示 X_1 的版本 (2)。

步骤 3

我们知道$\mathbf{owPc}_i = \mathbf{0}, i \in [L]$。回忆一下，$\mathbf{P} = [\mathbf{P}_{\mathbf{v}_1} \cdots \mathbf{P}_{\mathbf{v}_{v^\beta}}]$，其中 $\{\mathbf{v}_1, \ldots, \mathbf{v}_{v^\beta}\} = \mathcal{V}$。因此，

$$\mathbf{Pc}_i = \sum_{\mathbf{v} \in \mathcal{V}_{\mathbf{c}_i}} \mathbf{P}_{\mathbf{v}} c_{\mathbf{v},i} = \Big(\sum_{\mathbf{v} \in \mathcal{V}_{\mathbf{c}_i}} \mathbf{P}_{\mathbf{v}} \Big) c_i^* = \mathbf{0} \tag{46}$$

考虑形式为 $\Pi = \pi_1 \times \pi_2 \times \cdots \times \pi_\beta$ 的排列，其中 $\pi_i : [v] \to [v], i \in [\beta]$。假设该排列如下。

- F或某条特定的对抗性消息 $X_k, k \in \mathcal{A}$，以及两个特定的版本值 $m, m' \in [v$],

$$\pi_k[m] = m',$$
$$\pi_k[m'] = m,$$
$$\pi_k[l] = l, l \neq m, m'.$$

- 对于所有 $j \in \mathcal{A}, j \neq k$,

$$\pi_j[l] = l, \quad l \in [v].$$

只需证明该排列无效，因为任何其他排列都由类似这样的排列单元构成。对于每一列 $\mathbf{c}_i$，根据上一步的情况，有两种可能。

- 情况 1. 若 $\mathcal{V}_{\mathbf{c}_i} = \mathcal{V}$，则

$$\mathbf{P}^\Pi \mathbf{c}_i = (\sum_{\mathbf{v} \in \mathcal{V}} \mathbf{P}_{\mathbf{v}}) c_i^* = \mathbf{Pc}_i = \mathbf{0}. \tag{47}$$

- 情况 2. 假设单项式 $\mathbf{X}[i]$ 包含对抗性消息 k的版本 m。存在一个单项式 $\mathbf{X}[i'], i' \in [L]$，它与 $\mathbf{X}[i]$ 的区别仅在于 X_k的版本；在 $\mathbf{X}[i']$中，该版本为 m'。显然，

$$\mathbf{Pc}_{i'} = \sum_{\mathbf{v} \in \mathcal{V}_{\mathbf{c}_{i'}}} \mathbf{P}_{\mathbf{v}} c_{\mathbf{v},i'} = \Big(\sum_{\mathbf{v} \in \mathcal{V}_{\mathbf{c}_{i'}}} \mathbf{P}_{\mathbf{v}} \Big) c_{i'}^* = \mathbf{0}. \tag{48}$$

  容易验证，对于每个 $\mathbf{v}' \in \mathcal{V}_{\mathbf{c}_{i'}}$，都存在一个 $\mathbf{v} \in \mathcal{V}_{\mathbf{c}_i}$，其中 $\mathbf{v}'[k] = m'$ 且 $\mathbf{v}[k] = m$。根据步骤 1 中的定义，我们知道列 i 和 i' 是邻居节点，因此，$c_i^* = c_{i'}^*$。因此，

$$\mathbf{P}^\Pi \mathbf{c}_i = \sum_{\substack{\mathbf{v}' \in \mathcal{V}_{\mathbf{c}_{i'}} \\ \mathbf{v} \in \mathcal{V}_{\mathbf{c}_i}}} \mathbf{P}_{\mathbf{v}'} c_{\mathbf{v},i} = \Big(\sum_{\mathbf{v} \in \mathcal{V}_{\mathbf{c}_{i'}}} \mathbf{P}_{\mathbf{v}} \Big) c_i^* = \Big(\sum_{\mathbf{v} \in \mathcal{V}_{\mathbf{c}_{i'}}} \mathbf{P}_{\mathbf{v}} \Big) c_{i'}^* = \mathbf{0}, \tag{49}$$

  其中，最后一个等式由式 (48) 得出。总之，我们利用 $\mathbf{Pc}_{i'} = \mathbf{0}$证明了 $\mathbf{P}^\Pi \mathbf{c}_i = \mathbf{0}$ 。

参考文献

[1] S. Li 和 S. Avestimehr，“编码计算”，《通信与信息论基础与趋势》，第 17 卷，第 1 期，2020年。[2] J. S. Ng、W. Y. B. Lim、N. C. Luong、Z. Xiong、A. Asheralieva、D. Niyato、C. Leung 和 C. Miao，“编码分布式计算综述：基础、挑战与网络应用”，IEEE《通信调查与教程》，2021年。

[3] Q. Yang、Y.Liu、Y.Cheng、Y. Kang、T.Chen和 H. Yu，《联邦学习》，《人工智能与机器学习综合讲座》，第 13 卷，第 3 期，第 1–207 页，2019年。

[4] H. Jeong, 《面向可靠大规模计算的完全去中心化编码计算》，博士论文，卡内基梅隆大学，2020年。[5] Y. Sun、F. Zhang、J. Zhao、S. Zhou、Z. Niu 和 D. Gündüz, 《具有通信延迟的共享异构工作节点上的编码计算》，arXiv预印本 arXiv:2109.11246，2021年。[6] C. Wang 和 N. Raviv, 《同步系统中的全对全编码》，arXiv预印本 arXiv:2205.05183，2022年。[7] S. Sundaram 和 C. N. Hadjicostis, 《利用线性迭代策略进行分布式函数计算与共识》，IEEE通信领域精选期刊，第26卷，第4期，第650–660页，2008年。[8] ——, 《存在恶意代理时利用线性迭代策略进行分布式函数计算》，IEEE自动控制汇刊，第56卷，第7期，第1495–1508页，2010年。[9] C.-S. Yang 和 A. S. Avestimehr, 《安全布尔计算的编码计算》，IEEE信息论领域精选期刊，第2卷，第1期，第326–337页，2021年。[10] M. Soleymani、R. E. Ali、H. Mahdavifar 和 A. S. Avestimehr, 《列表可译码编码计算：突破对抗容错屏障》，arXiv预印本 arXiv:2101.11653，2021年。[11] L. Chen、H. Wang、Z. Charles 和 D. Papailiopoulos, 《Draco：通过冗余梯度实现拜占庭容错分布式训练》，载于国际机器学习会议。PMLR，2018年，第903–912页。[12] Q. Yu、S. Li、N. Raviv、S. M. M. Kalan、M. Soltanolkotabi 和 S. A. Avestimehr, 《拉格朗日编码计算：容错性、安全性和隐私的最优设计》，载于第22届国际人工智能与统计会议。PMLR，2019年，第1215–1225页。[13] N. A. Khooshemehr 和 M. A. Maddah-Ali, 《分布式线性编码的基本极限》，IEEE信息论汇刊，第67卷，第12期，第7985–7998页，2021年。[14] M. Pease、R. Shostak 和 L. Lamport, 《存在故障时达成一致》，《ACM杂志》（JACM），第27卷，第2期，第228–234页，1980年。[15] L. Lamport、R. Shostak 和 M. Pease, 《拜占庭将军问题》，载于《并发：Leslie Lamport文集》，2019年，第203–226页。[16] R. Canetti 和 T. Rabin, 《具有最优容错性的快速异步拜占庭一致》，载于第二十五届年度ACM计算理论研讨会论文集，1993年，第42–51页。[17] G. Bracha, 《一种异步 [(n-1)/3]容错共识协议》，载于第三届年度ACM分布式计算原理研讨会论文集，1984年，第154–162页。[18] C. Cachin、K. Kursawe、F. Petzold 和 V. Shoup, 《安全高效的异步广播协议》，载于年度国际密码学会议。施普林格，2001年，第524–541页。[19] A. Miller、Y. Xia、K. Croman、E. Shi 和 D. Song, 《拜占庭容错协议中的蜜獾》，载于2016年ACM SIGSAC计算机与通信安全会议论文集，2016年，第31–42页。[20] C. Cachin 和 S. Tessaro, 《异步可验证信息分散》，载于第24届IEEE可靠分布式系统研讨会（SRDS'05）。IEEE，2005年，第191–201页。[21] J. Hendricks、G. R. Ganger 和 M. K. Reiter, 《验证分布式纠删码数据》，载于第二十六届年度ACM分布式计算原理研讨会论文集，2007年，第139–146页。[22] L. Yang、S. J. Park、M. Alizadeh、S. Kannan 和 D. Tse, 《{DispersedLedger}：{可变带宽网络上的高吞吐量} 拜占庭共识》，载于第19届USENIX网络系统设计与实现研讨会（NSDI 22），2022年，第493–512页。[23] I. Abraham、P. Jovanovic、M. Maller、S. Meiklejohn、G. Stern 和 A. Tomescu, 《为异步分布式密钥生成达成共识》，载于2021年ACM分布式计算原理研讨会论文集，2021年，第363–373页。[24] G. Bracha, 《异步拜占庭一致协议》，《信息与计算》，第75卷，第2期，第130–143页，1987年。[25] 《Mirai僵尸网络》，2023年3月。 [在线]。网址：https://www.cloudflare.com/learning/ddos/glossary/mirai-botnet/[26] A. I. Sanka 和 R. C. Cheung, 《区块链可扩展性的系统综述：问题、解决方案、分析与未来研究》，《网络与计算机应用杂志》，第195卷，第103232页，2021年。[27] M. H. Nasir、J. Arshad、M. M. Khan、M. Fatima、K. Salah 和 R. Jayaraman, 《可扩展区块链——系统综述》，《未来一代计算机系统》，第126卷，第136–162页，2022年。[28] 《以太坊分片》，2023年2月。 [在线]。网址：https://ethereum.org/en/upgrades/sharding/[29] E. Kokoris-Kogias、P. Jovanovic、L. Gasser、N. Gailly、E. Syta 和 B. Ford, 《Omniledger：通过分片实现安全、可扩展的去中心化账本》，载于2018年IEEE安全与隐私研讨会（SP）。IEEE，2018年，第583–598页。[30] M. Zamani、M. Movahedi 和 M. Raykova, 《Rapidchain：通过完全分片扩展区块链》，载于2018年ACM SIGSAC计算机与通信安全会议论文集，2018年，第931–948页。

[31] S. Li、M. Yu、C.-S. Yang、A. S. Avestimehr、S. Kannan 和 P. Viswanath，“PolyShard：编码分片同时实现线性扩展的效率和安全性”，IEEE《信息取证与安全学报》，第16卷，第249–261页，2020年。[32] N. A. Khooshemehr 和 M. A. Maddah-Ali，“针对PolyShard区块链的差异攻击”，载于《2021年IEEE信息论国际研讨会（ISIT）》，2021年，第2672–2677页。[33] M. O. Rabin，“通过随机多项式进行指纹识别”，技术报告，1981年。[34] J. Katz 和 Y. Lindell，《现代密码学导论》。CRC出版社，2020年。