Chainlink 2.0：去中心化预言机网络的未来发展计划

Lorenz Breidenbach[1] Christian Cachin[2] Benedict Chan[1]
Alex Coventry[1] Steve Ellis[1] Ari Juels[3] Farinaz Koushanfar[4]
Andrew Miller[5] Brendan Magauran[1] Daniel Moroz[6]
Sergey Nazarov[1] Alexandru Topliceanu[1] Florian Tramèr[7]
Fan Zhang[8]

2021 年 4 月 15 日
v1.0

[1]Chainlink Labs

[2]作者是伯尔尼大学教职员工。他同时也是 Chainlink Labs 的一名顾问，并参与撰写了本白皮书。

[3]作者是 Cornell Tech 的教职员工。他同时也是 Chainlink Labs 的首席科学家，并参与撰写了本白皮书。

[4]作者是加利福尼亚大学圣迭戈分校的教职员工。她同时也是 Chainlink Labs 的一名顾问，并参与撰写了本白皮书。

[5]作者是伯伊利诺伊大学厄巴纳-香槟分校的教职员工。他同时也是 Chainlink 顾问，并参与撰写了本白皮书。

[6]作者是哈佛大学的在读博士生，他同时也是 Chainlink Labs 的一名研究员，并参与撰写了本白皮书。

[7]作者是哈佛大学的博士生，他同时也是 Chainlink Labs 的一名顾问，并参与撰写了本白皮书。

[8]作者将在 2021 年秋季成为杜克大学的员工，他同时也是 Chainlink Labs 的研究员，并参与撰写了本白皮书。

摘要

本白皮书在 Chainlink 第一版白皮书的基础上，突破最初提出的概念，详细勾勒了 Chainlink 未来的发展愿景。我们预测预言机网络将在未来发挥越来越多元化的价值，为智能合约快速可靠地接入任何链下数据源和链下计算资源，并同时保障隐私，补充并增强已有和新建区块链的性能。

我们将这个计划的根基称为“去中心化的预言机网络”，下文简称 DON。DON 是由一组 Chainlink 节点负责维护的网络，节点可灵活选择并部署任何预言机功能。因此，DON 是一个强大的抽象层，为智能合约接入丰富的链下资源，同时 DON 本身也可以展开高效的去中心化链下计算。

Chainlink 计划将 DON 作为跳板，在以下七个关键领域实现跃迁：

- 混合型智能合约：提供强大的通用框架，将链上和链下计算资源通过安全的方式组合成“混合型智能合约”，以提升现有智能合约的性能。
- 降低复杂性：为开发者和用户提供简单的功能，开发者无须研究复杂的底层协议和系统边界。
- 扩容：降低预言机服务延迟并提升吞吐量，满足高性能去中心化系统的需求。
- 隐私性：打造下一代系统，既保留区块链本身的透明性，又能保护敏感数据隐私。
- 为交易公平排序：为终端用户提供公平的交易排序服务，防止矿工抢跑或其他机器人或矿工发起攻击。
- 信任最小化：通过去中心化、稳健的区块链、加密技术和加密经济保障等手段为智能合约和其他接入预言机的系统创建非常值得信任的支持层。
- 基于加密经济激励的安全性：机制经过严谨的设计和稳健的部署，可以为 DON 建立强大的经济激励机制，约束节点诚实守信，甚至在面对猛烈攻击时也毫不动摇。

本白皮书介绍了 Chainlink 在以上每个领域取得的初步以及持续的创新，详细阐述了 Chainlink 网络在未来将如何不断建设能力。

目录

1 引言

如今，区块链预言机通常被看作是将链下数据传输到区块链的去中心化服务。从向区块链传输数据，到计算、储存并双向传输数据，其实只有一步之遥。这个理念拓宽了预言机的功能范围，同样地，由于智能合约的服务要求越来越高，以及接入的预言机网络在技术上越来越多元化，因此预言机的功能必须不断拓展。简而言之，预言机需要发展成双向传输数据的通用接口，连接链上和链下系统，并展开计算。对于区块链生态来说，预言机的价值是提升智能合约的性能、功能以及互操作性，为各个行业建立新的信任模型并实现透明性。混合型智能合约的广泛应用将推动这一转型。混合型智能合约既保留了区块链的特质，又加入了预言机网络等链下系统的独特功能，因此相比单独的链上系统来说拥有更广阔的应用场景和更强大的功能。

本白皮书详细阐述了 Chainlink2.0 的愿景，即在原版白皮书 [98] 概念的基础上进一步扩展。我们预测预言机网络将在未来发挥越来越多元化的价值，为智能合约快速可靠地接入任何链下数据源和链下计算资源，并同时保障隐私，补充并增强已有和新建区块链的性能。我们相信预言机网络将进一步发展，为区块链生态以外的系统输出具有区块链级别安全性的数据。

如今，Chainlink 预言机由各个节点运营商运行，并组合成预言机网络，通过“预言机报告”的形式将数据传输至智能合约。我们可以把这些预言机节点看作是典型区块链共识机制中的“节点委员会” [72]。但不同的是，预言机节点的目标不是提供独立的功能，而是服务于现有区块链。Chainlink 通过可验证随机函数（VRF）和链下报告（OCR）功能，已经逐渐演变成了一个通用框架和基础设施，为智能合约提供计算资源，实现高级功能。

我们将这个计划的根基称为“去中心化的预言机网络”，下文简称 DON。自从原版白皮书中引入了“预言机网络”的概念 [98] 后，预言机的功能不断丰富，应用场景也不断拓宽。本白皮书中根据 Chainlink 生态的愿景为这个术语做了全新定义。白皮书中认为，DON 是由一组 Chainlink 节点负责维护的网络，节点基于共识协议，可灵活选择并部署任何预言机功能。因此，DON 是区块链抽象层，为智能合约和其他系统接入链下资源。此外，它还能接入高效的去中心化链下计算资源。

总而言之，DON 可以为主链上的操作提供支持，其目标是实现安全灵活的混合型智能合约，结合链上和链下计算，并连接链下资源。

这里需要强调的是，虽然 DON 中采用节点委员会制度，但 Chainlink 本身仍然是无须许可的。DON 为无须许可的框架提供了基础，节点可以灵活组合，定制化预言机网络功能，网络可以设置成联盟制，也可以是无须许可的。

在 DON 的支持下，我们计划聚焦 Chainlink 2.0 在以下七大关键领域的发展，即：混合型智能合约、简化开发和使用体验、扩容、隐私性、为交易公平排序、信任最小化以及加密经济安全。在本论文的引言中，1.1 章节介绍了去中心化预言机网络的概览，1.2 章节绍了七大关键创新领域，1.3 章节详细介绍了论文剩余部分的提纲。

1.1 去中心化预言机网络

去中心化预言机网络的设计初衷是为目标区块链或主链上的智能合约接入链上无法实现的功能，以增强并扩展其性能。主要实现方式是提供计算机系统中的三个基础资源，即：网络连接、储存和计算。DON 旨在将智能合约接入这些资源，并同时保障隐私性、数据完整性、可用性[1]以及问责性。

DON 由预言机节点委员会组成，节点互相合作，完成具体的任务，或者建立长期合作关系，持续为客户端提供服务。DON 本身可以兼容任何区块链，并承诺为应用开发者提供强大且灵活的工具，为任何主链上的智能合约提供链下资源。

DON 包含两种功能，即：可执行程序以及外部适配器。可执行程序是在 DON 中以去中心化的方式持续运行的程序。它们不直接储存主链上的资产，但是却有重要的价值，因为它们拥有强大的性能，而且可以开展隐私计算。可执行程序在 DON 中自动运行，并执行确定性的操作。与此同时，适配器将 DON 接入链下资源，并且可以通过可执行程序调用。DON 中的适配器泛指目前 Chainlink 网络中的外部适配器。现有的适配器通常只从数据源获取数据，但实际上适配器可以实现双向操作。DON 中的适配器还可以利用 DON 节点进行联合计算，实现更强大的功能，比如为可执行程序的隐私计算报告加密。

图 1 展示了 DON 的基本运行原理，可以通过这张图了解 DON 如何向区块链传输报告，并实现现有的预言机功能。然而，DON 的功能远远超越了现有的 Chainlink 网络。比如在图 1 的框架中，可执行程序可以在 DON 中记录获取的价格数据，并用数据进行计算（如：为报告生成尾随平均值）。

DON 的最大优势是可以快速启动新的区块链服务。DON 可以为现有区块链网络提供载体，为其快速建立服务和应用，无需再创建专用的网络。第 4 章中详细阐述了几个用例。

第 3 章对 DON 做了深入解析，详细阐述面向开发者和用户接口，并以此剖析 DON 的功能。

[1]信息安全的“CIA 三要素” [123, p. 26, §2.3.5]

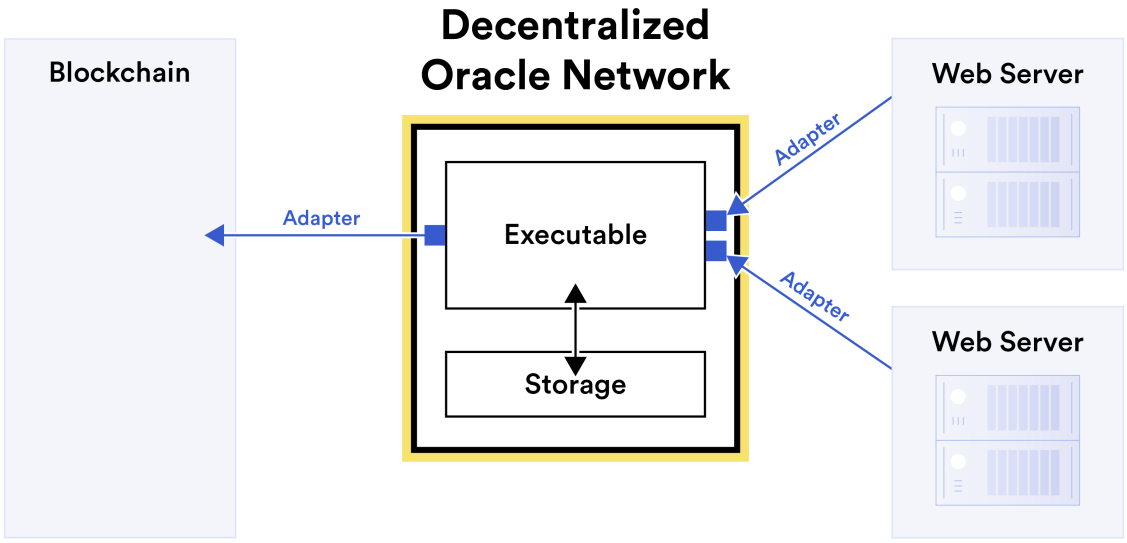

图 1: 示例中展示去中心化预言机网络如何实现基本的预言机功能，即向链上合约传输链下数据。可执行程序通过适配器获取链下数据，并对数据开展计算，将计算结果通过另一个适配器传输至目标区块链。（适配器由 DON 中的代码启动，即图中的蓝色方块；示例中的箭头展示了数据传输方向。）可执行程序还可以在 DON 本地读写数据，维护状态并与其他可执行程序进行通讯。DON 拥有灵活的数据传输、计算和储存功能，可以实现一系列创新的应用。

1.2 七大关键设计目标

在这里我们先简要探讨一下上文中提到的七大关键发展领域，即：

混合型智能合约： Chainlink 愿景的核心理念就是在智能合约中安全地集成链上和链下组件。我们将实现这一愿景的智能合约成为“混合型智能合约”或“混合型合约”。[2]

区块链在目前和未来都会对去中心化服务生态起到两个关键作用，即：加密货币所有权储存在区块链上，且去中心化服务架构在区块链上。因此，智能合约必须放在链上执行，但是这严重限制了它们的链上功能。将合约代码完全放在链上，速度慢、成本高、而且功能少。链上无法实现许多只有链下数据可以实现的功能，比如各种形

[2]之前也提出过在合约中结合链上链下模块的想法，但实现方式都存在一定限制，比如 layer-2 系统，基于可信执行环境（TEE）的区块链 [80] 等。我们的目标是打造更为通用的解决方案，让智能合约既能连通链下数据，又能获得其他关键的预言机服务

式的保密计算、生成（伪）随机数，防止矿工/验证者操纵。

智能合约要想充分实现潜力，就必须同时拥有两个模块，即：链上模块（我们通常称为 SC）和链下模块（DON 上运行的可执行程序，通常称为 exec）。最终目标是安全地集成链上功能和 DON 提供的链下服务。这两个模块共同组成了混合型合约。图 2 展示了这个概念。如今，Chainlink 喂价和 VRF 等服务[3]已经实现了前所未有的智能合约应用，其中包括 DeFi、公平的 NFT 以及去中心化保险。而这是实现通用框架的第一步。依据本白皮书中的愿景，Chainlink 服将不断丰富，其功能也越来越强大，这将推动所有区块链上的智能合约系统向前发展。

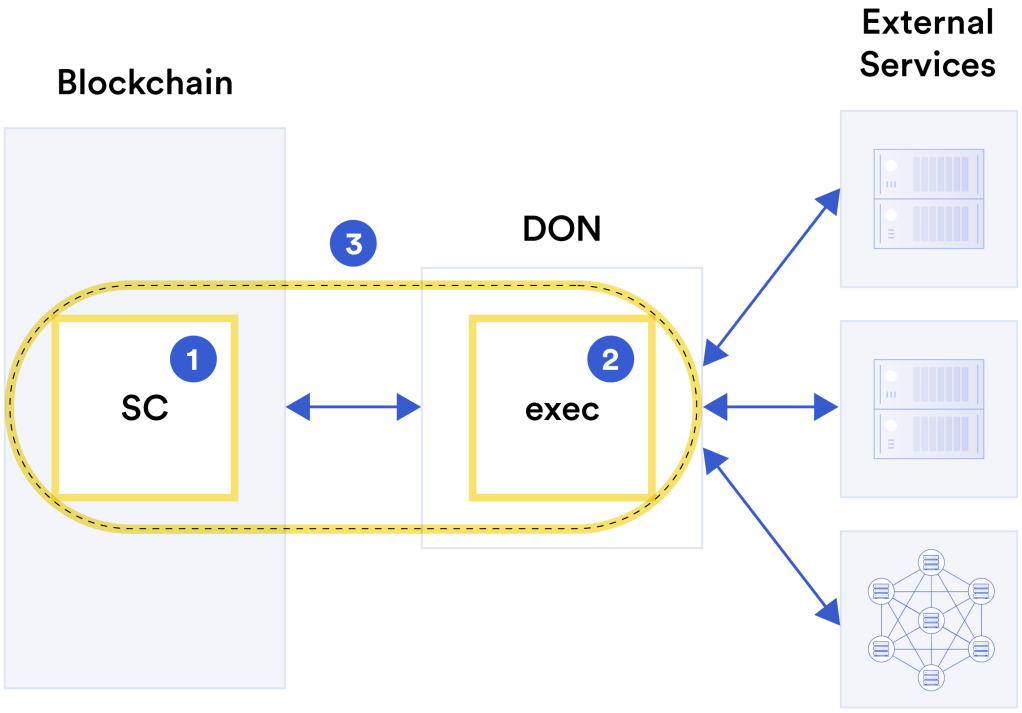

图 2: 智能合约的链上和链下模块。混合型智能合约 ③ 由两部分组成，即：链上模块 SC ①，这个模块在链上运行；链下模块 exec ②，这个模块在 DON 中执行。DON 在这两个模块之间搭建沟通的桥梁，并且将混合型智能合约连通 web 服务、其他区块链以及去中心化存储等各类链下资源。

其余的六大关键领域都是围绕混合型智能合约这一最关键的领域展开的。其中包括简化混合型合约的开发和使用，打造更多的链下服务，提升混合型合约的性能。

[3]Chainlink 服务包含众多去中心化服务和功能。这些服务由生态中各个节点运营商提供，这些节点运营商可以灵活组成预言机网络。

信任最小化则是为了提升混合型合约的安全性。论文中虽然没有处处提到“混合型智能合约”这个概念，但是只要是将 MAINCHAIN （主链）代码逻辑与 DON 结合的智能合约都可以被视作是混合型智能合约。

简化开发和使用体验： DON 在设计上简化了其背后复杂的流程和机制，让开发者和用户可以轻松使用去中心化系统强大灵活的服务。目前 Chainlink 已经有了这个功能。比如，开发者在使用链上喂价时无需研究协议具体底层规则，如 OCR 如何在去中心化节点之间达成共识。DON 进一步拓展了 Chainlink 的服务，为开发者提供了一个抽象层，并简化了高级服务的接口。

第 4 章展示了几个这方面的用例。比如设想企业可以将 DON 作为安全的中间件，将遗留系统接入区块链。（参见第 4.2 章节。）DON 可以简化通用区块链机制（比如费用、区块重组等）。另外，还可以简化某些区块链的功能，企业无须专门研究需要接入的区块链系统或甚至无须具备去中心化系统开发经验，就可以轻松将企业系统接入不断扩张的区块链生态。

我们的终极目标是通过 Chainlink 不断简化开发和使用体验，最终实现“去中心化的元层”。这个元层将为所有 DApp 的开发者和用户模糊链上链下的边界，无缝开发并使用去中心化服务。

为了简化开发流程，开发者可以在元层中以统一的机器模型具体定义 DApp 功能。然后使用去中心化的元层编译器自动将 DApp 实例化成一组具有互操作性的去中心化功能，覆盖区块链、DON 以及链下服务。（链下服务包括企业系统，可以帮助应用通过元层接入企业遗留系统。企业系统可以是其中一个链下服务，因此元层可以为应用接入企业遗留系统。）这个编译过程类似开发者为了充分利用异步硬件架构的优势而使用现代编译器和 SDK。架构中可能有通用 CPU 以及一些专业硬件，比如 GPU、机器学习加速器或者可信环境等。图 3 具体展示了这一概念。

混合型智能合约是我们实现“元合约”（meta contract）的第一步。元合约是在去中心化元层上开发的应用，包含链上逻辑（即智能合约），并能接入链下计算资源、其他区块链和链下服务。然而，由于需要实现编程语言和编译器、建立新的安全模型、并且在技术和概念上协调不同的技术，要充分实现这个去中心化的元层仍需要较长时间。不过这仍是一个理想的模式，希望各位读者在看本白皮书的时候能够在脑中构建这一概念。虽然在此我们对这个概念不做赘述，但这是我们在未来工作中重点关注的内容。

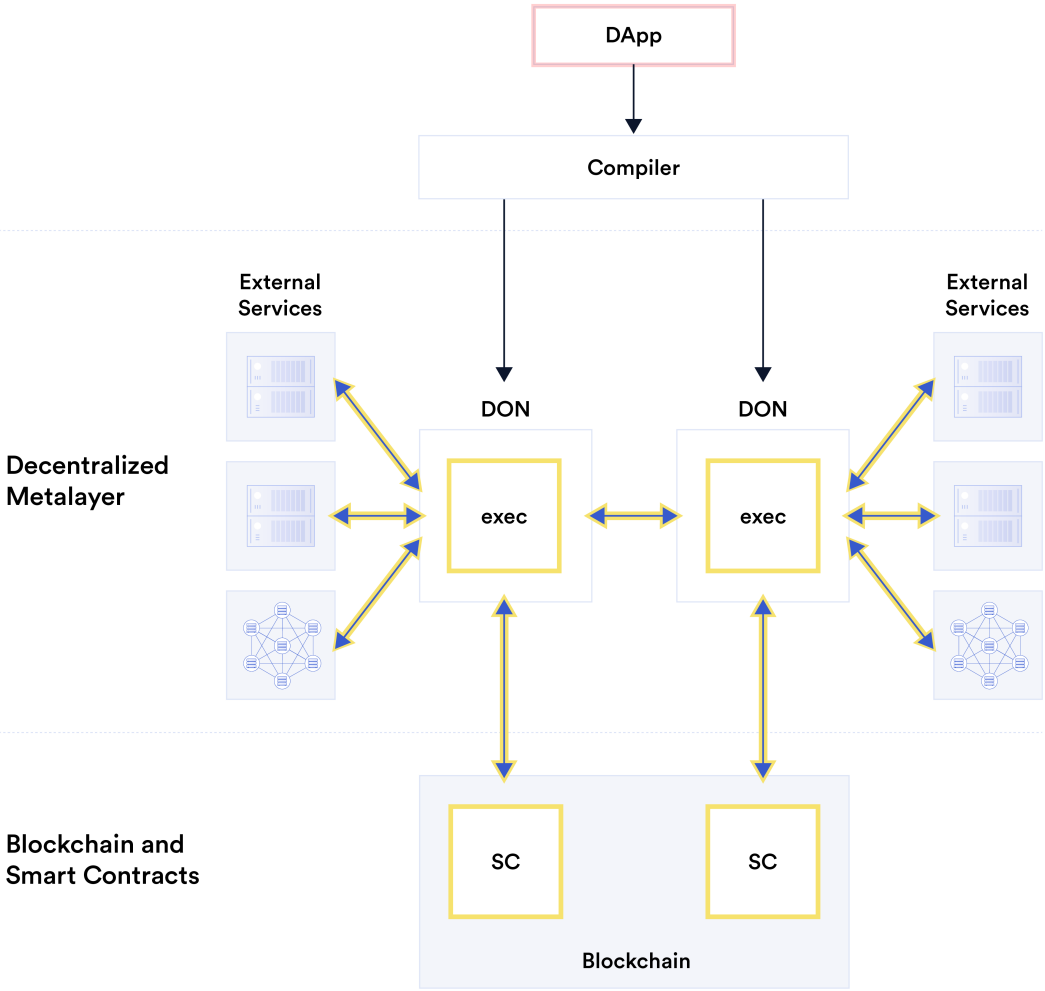

图 3: 展示了去中心化元层的理想实现效果。为了简化开发流程，开发者可以用统一机器模型以虚拟的方式定义 DApp 功能（粉色区域）。去中心化的元层编译器自动生成对应的交互功能，即：智能合约（SC ）、DON 的逻辑（exec ）、以及接入链下服务的适配器等（黄色区域）。

扩容： 我们致力于升级 Chainlink 网络功能的主要目的是让 Chainlink 能够满足区块链生态日益增长的扩容需求。

由于区块链网络拥堵目前已成为公链中不可回避的一个问题 [86]，因此催生出了许多性能更强大的新型区块链 [103, 120, 202]，以及各种 layer-2 扩容方案，如 [5, 12, 121, 141, 168, 185, 186]。预言机服务必须在延迟性和吞吐量方面满足这些系统的需求，并同时将合约运行者和普通用户的链上费用降至最低（注：gas 费用）。DON 可以进一步提升 Chainlink 的功能和性能，满足 web 系统的要求。

DON 在为区块链提供服务时采用了基于委员会的共识机制（或无须许可的共识机制），因此性能和速度得到了大幅提升。之后会有许多不同配置的 DON 并行，各个 DApp 和用户可以权衡不同 DON 的底层共识机制，根据自身应用需求做出选择。

DON 实际上可以被看作是 layer-2 技术。除了各种服务之外，DON 还将支持交易执行框架（TEF），该框架可以高效地将 DON 集成至其他高性能的 layer-2 系统（如：rollups），这些系统可以将链下交易打包，以提升系统性能。第六章将详细介绍 TEF。

图 4 展示了 DON 如何将交易和预言机报告放在链下而不是链上处理，以扩展区块链（智能合约）。将计算从链上转移到链下可以降低交易延迟和交易成本，并同时提升交易吞吐量。

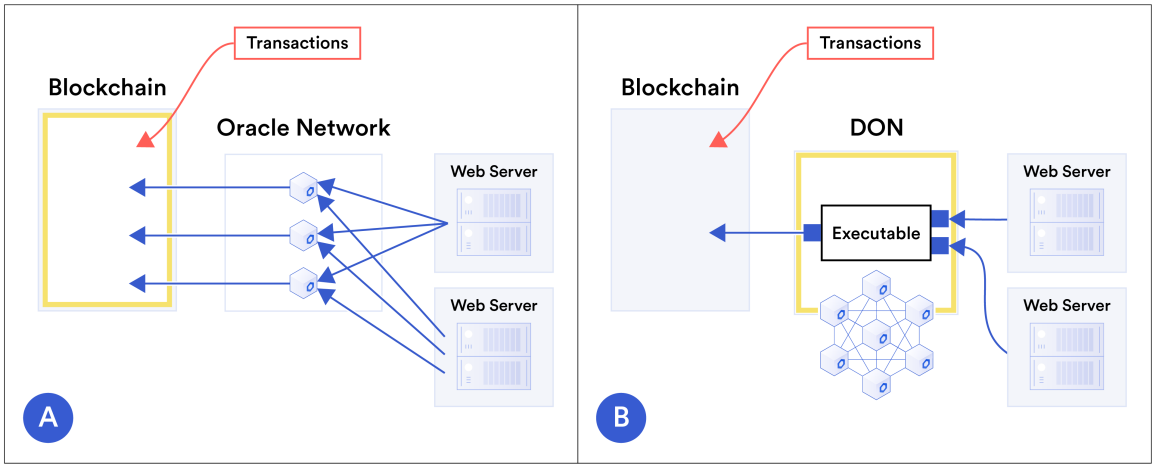

图 4: 图中展示了去中心化预言机网络如何提升区块链智能合约的可扩展性。图 A 展示了传统的预言机基础架构。交易和预言机报告都直接被发送至链上。因此，黄色部分的区块链是处理交易的主要地点。图 B 展示了如何使用 DON 为区块链智能合约提供支持。DON 可执行程序处理交易以及来自链下系统的数据，并将结果（如：打包的交易或交易产生的状态变更）发送至链上。因此，黄色部分的 DON 是处理交易的主要地点。

隐私性： 区块链为智能合约应用提供了前所未有的透明性，但是透明性和隐私性之间存在天然的冲突。比如，如今用户的去中心化交易被记录在链上，因此比较容易监控交易行为，但同时也公开了用户的交易内容。同样地，传输至智能合约的数据也保留在链上。这使得数据方便被审查，但也使数据提供商不敢向智能合约传输敏感或内部数据。

我们认为，预言机网络将极大推动下一代系统的发展，既保留区块链的透明性，又保障隐私性。本白皮书将向各位展示实现这一目标的三种主要方式：

- *保护隐私的适配器：* Chainlink 计划在其网络中部署两种技术，即 DECO [233] 和 Town Crier [232]。这两项技术可以为节点传输链下系统的数据，并同时保护用户和数据隐私。它们将对 DON 适配器的设计起到关键作用。（参见 3.6.2 章节了解技术细节）

- *保密计算：* DON 可以向区块链隐藏计算内容。采用安全多方计算或可信执行环境也可以增强隐私性，DON 节点在计算数据过程中自己都无法查看数据内容。

- *兼容具有隐私功能的 layer-2 系统：* TEF 在设计上可兼容各类 layer-2 系统，其中许多系统采用零知识证明，以各种方式保障交易隐私。

第 3 章会讨论具体方法（第 6 章、附件 B.1 和 B.2 中也会涵盖相关内容）。

图 5 展示了敏感数据如何从链下数据源通过保障隐私的适配器和 DON 保密计算传输至链上智能合约。

我们认为这个处理隐私数据的强大工具将催生出一系列创新的应用。其中包括隐私去中心化（以及中心化）金融、去中心化身份、链上信用贷款以及更加高效好用的 KYC 和身份认证协议。第四章会具体展开讨论。

为交易公平排序： 如今区块链设计存在一个“公开的秘密”，那就是区块链在一段很短暂的时间内其实是中心化的。矿工和验证者可以按照自己的心意为交易排序。网络中的用户也可以通过支付交易费来左右交易排序（如：以太坊中的 gas 费），并在一定程度上利用网络连接速度占得先机。这类操纵可以是交易抢跑的形式。矿工会观察用户交易，并在同一区块中将自己的交易插在前面，以谋取私利。他们这样做是利用了自己可以提前看到交易池的优势来榨取用户的利益。比如，可以开发一个机器人，将买入交易插在某个用户交易的前面。这样就可以在用户的买单拉高资产价格前先买入，以此获利。

机器人抢跑会损害普通用户的利益，这点类似华尔街的高频交易，而这种情况已经非常普遍并且有详细记录 [90]。同样地，还存在尾随攻击（back-running）[159] 和自动复制交易（automated transaction mimicking）[194] 等情况。近期甚至有人提出要将矿工操纵交易的行为体制化 [110]。

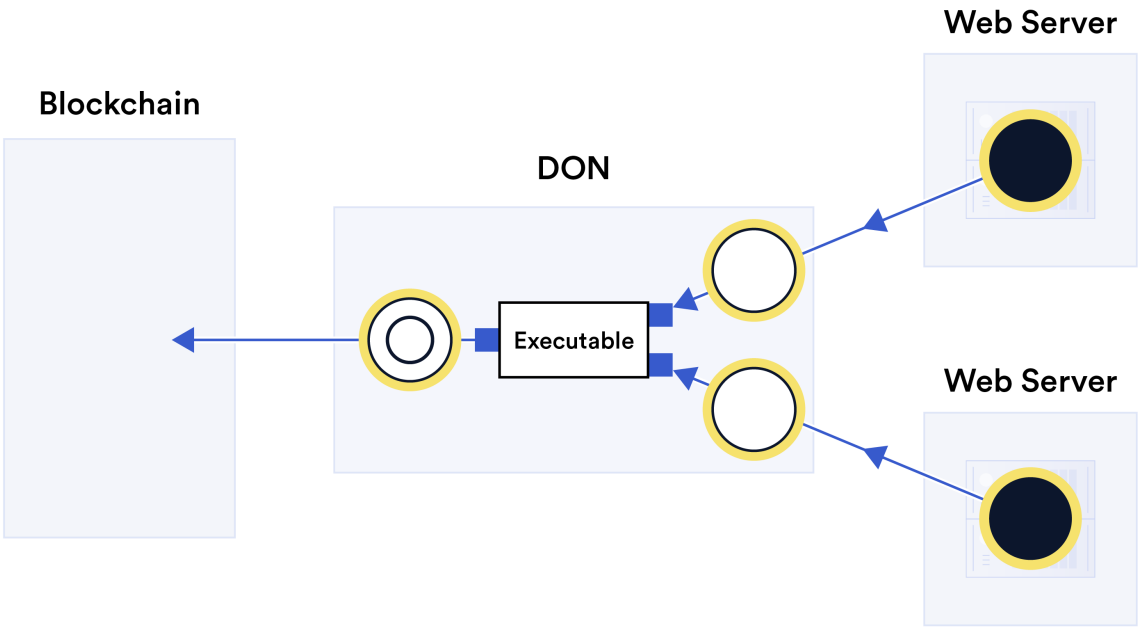

图 5: 在 DON 中对隐私数据进行保密操作（黄色部分）。使用保护隐私的适配器（蓝色双箭头）将 web 服务器中的隐私数据（黑色圆圈）传输至 DON。DON 收到适配器传输的派生数据（空心圆圈），派生数据（derived data）指对敏感数据应用函数运算或加密共享后的结果。DON 的可执行程序可以对派生数据进行保密计算，生成报告（双圆圈），然后将报告通过适配器发送到链上。

Rollup 等 layer-2 技术并没有解决这个问题，而只是重新将交易排序中心化，将交易排序的职责交给了创建 Rollup 的实体。

我们的目标是在 Chainlink 网络中推出名为“公允排序服务”（FSS）的服务 [137]。FSS 保障了智能合约设计者可以为交易公平地排序，并避免交易抢跑和尾随攻击等各类对用户交易以及其他类型的交易攻击。FSS 可以为 DON 严格基于时间顺序公平地为交易排序，[144] 中会详细阐述这个概念。除此之外，FSS 还可以为用户降低网络费用（如：gas 费）。

简而言之，在 FSS 中，交易需要通过 DON，而不是直接传输至目标智能合约。DON 会先为交易排序，然后再将交易上传至智能合约。

图 6 将标准挖矿流程与 FSS 做比较。在标准挖矿流程中，交易排序是中心化的，权利在矿工手中，因此可能被操纵，比如将后到的交易排在先来的交易之前。相反，FSS 则是用 DON 节点以去中心化的方式排序交易。假设一组节点是诚实的，FSS 可以严格按照时间顺序为交易排序，降低矿工和其他实体操纵排序的可能性。另外，由

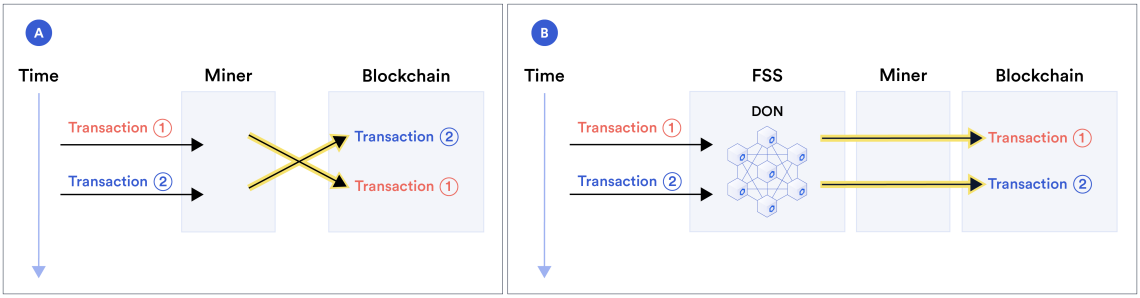

图 6: FSS 功能示例。图 A 展示了矿工如何利用其中心化的权利为交易排序，将后到达交易池的交易排在先到达的交易前面。相反，图 B 展示了 DON 如何通过去中心化的节点排序交易。如果一组诚实的节点先收到交易 ① 再收到交易 ②，那么 FSS 就会将 ① 排到 ② 上链，防止矿工通过附加序列号码的方式重新排列交易。

于用户不需要通过 gas 费竞价来左右交易排序，因此大家都可以相应降低 gas 成本。另外，DON 的交易还可以批量执行，进一步降低 gas 费。

信任最小化： DON 在设计时的主要目的是为智能合约和其他接入预言机的系统创建一个高度可信的支持层，提供去中心化架构、加密工具，并建立加密经济激励机制。DON 本身是去中心化的，用户可以任意选择 DON 接入区块链，或选择自己信任的节点组成 DON。

然而，对于一些应用——特别是智能合约来说，Chainlink 用户希望建立一种特殊的信任模型，使 DON 所接入的区块链的可信程度甚至超过 DON 本身。我们针对这类用户已经推出或正计划推出一系列 Chainlink 网络机制，提升 DON 对区块链的安全保障，并同时保护来自 web 服务器的数据不受到操控。

第七章会详细阐述这些机制。机制可以划分为以下五大类：

- 数据源认证：为数据提供商提供工具，对数据进行数字签名，以增强数据源和链上合约之间的监管链（chain of custody）。
- DON 少数派报告：少数 DON 节点发现多数节点的恶意操作并上报。
- 安全护栏：主链上的合约逻辑监测异常情况，并有权暂停或停止合约执行（或采取其他补救措施）。
- 信任最小化治理：采用降速更新机制辅助社区监督，并采取去中心化的紧急干预机制快速响应系统故障。：使用公钥基础设施（PKI）识别 Chainlink 网络中的节点身份。

图 7 是信任最小化目标的概念图。

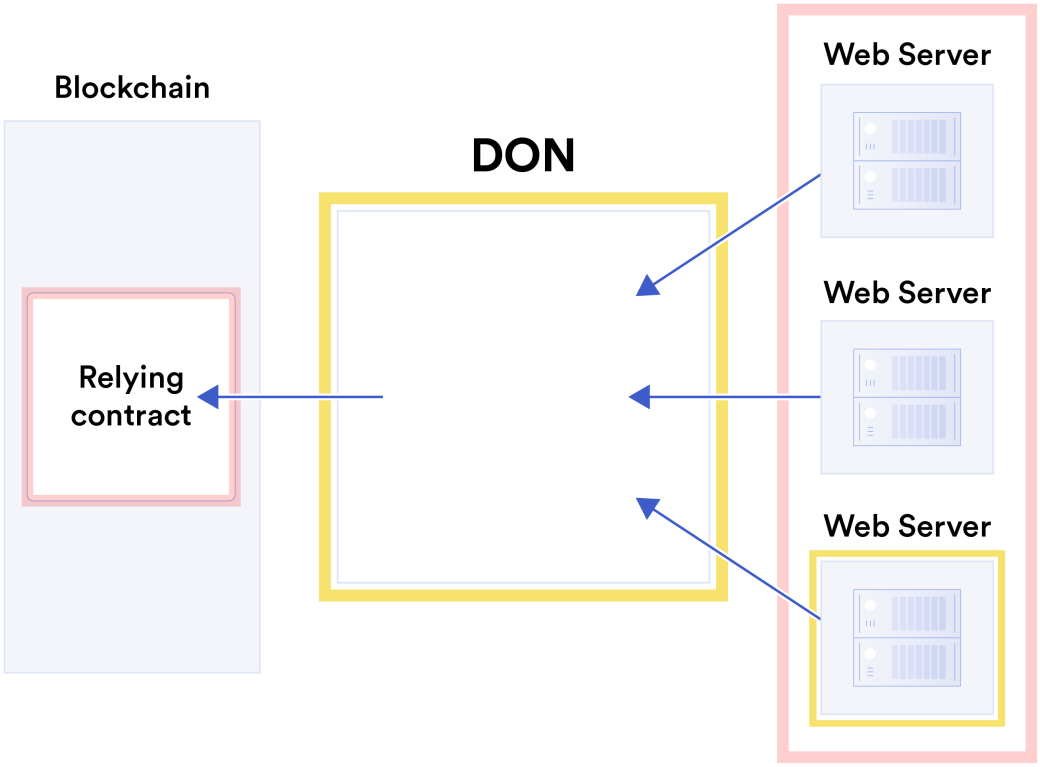

图 7: Chainlink 信任最小化目标的概念图，目的是将用户对 DON 以及 web 服务器等数据源的信任最小化。图中黄色部分表示信任最小化的发生地点，即：DON、单个或少数 web 服务器。粉色部分表示预设可信度非常高的系统组件，即：区块链上的智能合约以及多数 web 服务器，即：web 服务器集合。

基于加密经济激励的安全性： 将报告生成去中心化，分散到各个预言机节点，即使节点受到操控，也可以保障安全性。然而，同样重要的是要确保节点有足够的经济激励保持诚实守信。权益质押（注：节点质押 LINK 保证金，如果出现违规行为则保证金被没收）将在 Chainlink 网络中起到关键作用。许多区块链都采用了这一重要的激励机制，其中包括 [81, 103, 120, 203]。

然而，Chainlink 中的质押机制与普通区块链大相径庭。区块链中的质押机制是为了避免攻击者对共识发起攻击。而 Chainlink 的目标则不同，它的质押机制是为了保障及时传输正确的预言机报告。预言机网络的质押机制如果设计得当，应有能力使攻击者即便买通了预言机也无利可图，就算攻击目标是高值智能合约也得不偿失。

本白皮书阐述了 Chainlink 质押机制在以下三个关键领域的创新：

1. 包含了现存方法忽视的攻击的强大对抗模型。其中一个例子就是所谓的“预期性贿赂”（prospective bribery）。这种类型的贿赂提前设定了付款条件，比如：提前保证向质押机制随机选中的节点提供贿赂，这些节点通常被选中完成触发报告仲裁等具体任务。

2. 超线性质押影响，指如果攻击者要有效攻击预言机网络，资金量必须大于所有预言机节点质押保证金的总和。作为 n 的函数，$\$B(n) \gg \$dn$。假设网络中有 n 个预言机节点，每个节点都质押 $\$d$ 保证金，随着 n 的增长，$\$B(n)$ 逐渐大于 $\$dn$。图 8 中具体说明。

3. 隐性激励框架（IIF）是我们设计的一个激励模式，在显性质押金额之上再添加了一层实证维度可衡量的激励，比如节点的未来收入机会。IIF 拓宽了质押的概念，超出了节点质押保证金的范畴。

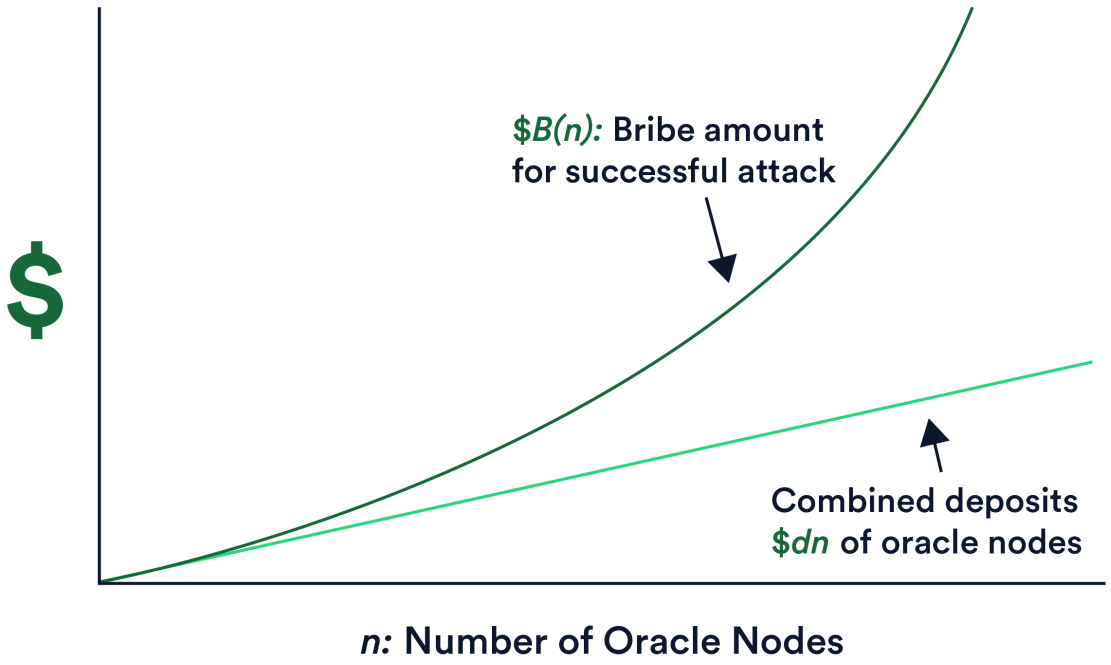

图 8: Chainlink 质押机制中的超线性扩展。攻击者贿赂的金额 $\$B(n)$ 比所有预言机节点保证金总和 $\$dn$ 增长更快

IIF 和超线性质押共同为预言机网络创建了经济安全良性循环。更多新用户进入系统，将提升 Chainlink 节点运营商的潜在收益，这也会为当前以及未来用户降低经

济安全的边际成本。在弹性需求机制下，边际成本降低会激励更多用户进入网络，因此建立良性循环，驱动网络持续运转。

注：本白皮书中虽然描绘了 Chainlink 未来发展愿景的关键要素，但并未详细阐述，也缺少技术细节。之后我们将发布具体的技术文档。另外还需指出的是，本白皮书中提到的许多内容（如：扩容、隐私技术以及公允排序服务等）都将在高级版 DON 成为 Chainlink 基础功能之前就先部署在初级版本中。

1.3 论文提纲

第 2 章解释了 Chainlink 的安全模型及其符号；第 3 章简要描述了去中心化预言机网络的 API。第 4 章列举了多个将 DON 作为应用部署平台的用例。读者到此就可以了解白皮书中的大部分重点概念。

剩下的内容则是对这些概念的具体阐释。第 5 章详细解释了公允排序服务 (FSS)；第 6 章深入探讨了交易执行框架（TEF）。第 7 章详述了实现信任最小化的方法。第 8 章讨论了一些部署 DON 的关键要求，即渐进式功能发布、动态 DON 成员制以及可问责性。最后第 9 章概括了 Chainlink 的激励机制设计思路。第 10 章为总结。

为了帮助对白皮书中概念不熟悉的读者理解，附件 A 为术语库。附件 B 进一步解释了 DON 接口和功能的细节，附件 C 展示了一些适配器案例。附件 D 具体阐述了信任最小化数据源认证的加密学概念“函数签名”，并引入了一个新的衍生概念，叫“离散函数签名”。附件 F 中讨论了 DON 选择委员会的一些考量因素。

2 安全模型和目标

去中心化预言机网络是一个独特的分布式系统，最初预期将由基于委员会的共识协议建立（注：但这并不是唯一的模式），并由一组预言机节点运行。DON 的主要作用是为主链上智能合约提供预言机报告以及其他服务，以增强智能合约的功能。但与此同时，DON 还可以为其他非区块链系统提供同样的服务，因此不一定需要接入区块链。

因此，这个模型及其特质与 DON 中具体使用什么应用基本上没有关联。

2.1 目前的架构模式

这里需要指出的是，Chainlink 并非单体服务，而是无须许可的框架，在其中可以发布各种独特且独立的预言机节点网络 [77]。Chainlink 网络中有许多不同的节点网络，每个都有自己独特的机制。这些网络分别提供不同类型的服务，比如喂价、储备金证明以及可验证随机数等。另外，每个网络的去中心化水平、资金规模以及服务水平参数（如：数据更新频率和准确性）都有所不同。

Chainlink 无须许可的模式可以推动生态扩张，服务提供商可以专注于提供他们最擅长的服务。相比起要求所有节点和网络都提供全套服务，这种模式可以为用户降低成本并提升服务质量。而前者很可能最终导致整个系统充斥着各种无人问津的服务。

随着 Chainlink 逐渐过渡到 2.0 版，并建立在 DON 的基础上，我们将持续开发无需许可的开放式框架，并坚持为用户提供各类服务，最好地满足具体应用需求。

2.2 共识假设

我们使用“去中心化预言机网络”一词涵盖我们所描述的预言机系统的全部功能，既包含预言机节点维护的数据结构，又包含上面的核心 API。

我们使用 ledger（账本）一词表示底层数据结构，符号是 L。架构由 DON 维护，并支持 DON 提供的具体服务。在这里要强调的是，Chainlink 的 DON 框架不把 L 看成是类似区块链的独立系统：其目的是支持区块链和其他系统。当然，区块链是实现可信账本的一种方式，但除此之外还有其他方式。我们期望 DON 在许多情况下能采用拜占庭容错机制（BFT）实现底层账本，这个机制远远早于比特币等区块链出现 [173]。为了方便起见，我们在白皮书中通篇采用 BFT 相关的标记和属性。但同时也要强调 DON 可以通过无须许可的共识协议实现。

理论上来说，账本 L 是一个公告板，数据在上面按照线性排列。我们认为账本拥有几个区块链专有的关键属性 [115]。账本的属性包括：

- 只能添加（Append-only）：数据一旦添加，就不能移除或修改。
- 公开：任何人都可以读取账本内容，并且任何用户在任何时间读取的内容都保持一致。[4]

[4]如果一个没有最终确定性的区块链实现了账本，会通过忽略不够深的区块或者“减枝”来消除不一致性 [115]

- 可用：账本可以由授权用户写入，并由任何人及时读取。

当账本由委员会实现时，DON 还具有额外的属性。比如，只有某些用户有权限写入账本，或者只有某些应用可以读取账本。也就是说账本无须像上文定义中一样向所有人开放。同样地，账本还可以设置成允许数据更改或编辑。不过，本白皮书不具体探讨这些情况。

DON 的模块化设计可以兼容各种现代 BFT 协议，比如 Hotstuff [230]。具体怎么选择要取决于预言机节点之间的信任模型和网络特点。理论上，DON 还可以接入无须许可的高性能区块链，使账本支持扩展性同样高的 layer-2 或区块链系统。同样，也可以采用混合模式：原则上，DON 可以由现有区块链上的验证者作为节点，比如 POS 机制中，选出委员会执行交易 [8, 81, 120, 146, 203]。这种运行模式要求节点采取双模式运行，既是区块链节点，也是 DON 节点。（参见 8.2 章节，了解如何通过技术保障委员会变更的连续性；参见附件 F 了解随机选择委员会的注意事项。）

实际上在现代 BFT 算法中，节点会对账本中的数据进行数字签名。为了方便起见，我们假设$\mathcal{L}$ 有公钥 $\mathsf{pk}_{\mathcal{L}}$，其内容由相应的私钥签名。即使在用门限签名来对数据签名时，这个通用符号也适用。[5] 门限签名很方便，因为可以为 DON 创建永久身份，即使运行 DON 的节点出现变更，DON 身份也不会变。（参见附件 B.1.3。）因此，我们假设 $\mathsf{sk}_{\mathcal{L}}$ 基于 (k, n) 的门限机制加密共享，安全参数是 k，如：$k = 2f + 1$ 并且 $n = 3f + 1$。其中 f 表示潜在问题节点的数量。（通过这种方式选择 k，可以确保问题节点无法得知 $\mathsf{sk}_{\mathcal{L}}$，也无法发起 DoS 攻击。）

$\mathcal{L}$ 上的消息用 $M = (m, z)$ 表示，其中 m 是字符串，z 是独特的顺序索引编号。在适用的情况下，我们将消息写成 $m = \langle \texttt{MessageType} : \text{payload} \rangle$。消息类型 `MessageType` 是“语法糖”，表明某条消息的函数。

2.3 符号注释

我们用 $\mathcal{O} = \{\mathcal{O}_i\}_{i=1}^{n}$ 来表示账本中运行的一组 n 个预言机节点。这一组节点通常被称为“委员会”。为了简便，我们假设实现 DON 功能（即在）的这组预言机与维护 L 的预言机是同一组，但实际上它们也有可能是不同的。我们用公钥 pk_i 表示参与者 $\mathcal{O}_i$，并用 sk_i 表示相应的私钥。

[5] 实际操作中，LibraBFT 等代码库 [204]（注：LibraBFT 是 Hotstuff 的衍生物）目前并没有采用门限签名，而是采用了多重签名，为了简化工程设计而牺牲了通讯简洁性。预言机节点只要额外支付一笔费用，就可以向写入 $\mathcal{L}$ 的消息中添加门限签名，即使 $\mathcal{L}$ 的共识协议未采用门限签名也可以这么做。

大部分 BFT 算法都要求至少 $n = 3f + 1$ 个节点，其中 f 代表潜在问题节点数量；剩余的节点都是诚实的，也就是说它们会严格按照协议执行。如果委员会 $\mathcal{O}$ 满足这个要求，即：诚实节点超过 2/3，我们就认为这个委员会是诚实的。除非另作说明，否则我们都认为 $\mathcal{O}$ 是诚实的。我们根据不同的情景，将 $\mathsf{pk}_{\mathcal{O}}$ / $\mathsf{sk}_{\mathcal{O}}$ 和 $\mathsf{pk}_{\mathcal{L}}$ / $\mathsf{sk}_{\mathcal{L}}$ 混合使用。

我们用 $\sigma = \mathsf{Sig}_{\mathsf{pk}}[m]$ 表示 pk 相关消息 m 的签名，即：使用相应的私钥 sk。用 $\mathsf{verify}(\mathsf{pk}, \sigma, m) \to \{\mathtt{false}, \mathtt{true}\}$ 表示相应的签名验证算法。（注：论文通篇都隐约提到了密钥生成。）

我们用符号 S 表示数据源，用 $\mathcal{S}$ 表示具体情境下 n_S 个数据源集合。我们用 MAINCHAIN 表示 DON 接入的智能合约区块链。我们用依赖合约（relying contract）表示 MAINCHAIN 上任何与 DON 交互的智能合约，并用符号 SC 表示此类智能合约。

我们通常假设一个 DON 只接入一个 MAINCHAIN，但实际上 DON 也可以接入多个 MAINCHAIN，第 4 章中会讨论相关用例。DON 通常会接入。（如上所述，DON 也可以支持非区块链服务。）

2.4 信任模型注释

如上所述，DON 可以采用委员会制 (committee-based) 的共识协议，我们预期这将是未来主流的协议类型。对于委员会制更安全还是无需许可制更安全，人们总是争论不休，两方各执一词。

这里必须要认识到一点，委员会制和无须许可制在安全性上无法直接对比。对 PoW 或 PoS 区块链发起 51% 攻击，攻击者需要在一段时间内匿名获得大多数资源，比如在 PoW 系统中租用哈希算力。实际上此类攻击已经成功攻陷了几个区块链 [199, 34]。相比之下，要攻陷基于委员会的系统则需要拿下网络中通常至少 1/3 的节点。这些节点可能是公开身份的，且资源丰富，可信度高。

另一方面，委员会制的系统（以及委员会下面的“混合型”无须许可的系统）可以比完全无须许可的系统实现更丰富的功能。比如可以保护签名或密钥等隐私数据，这是 Chainlink 设计中的一个可能实现的功能。

我们在此要强调一点，DON 在理论上既可以采用委员会制也可以采用无须许可的共识协议，DON 部署者可以灵活选择任意一种模式。

增强信任模型： 目前 Chainlink 的关键特征是让用户根据节点去中心化的历史性能记录来选择节点，3.6.4 章节会详细讨论这一点。第 9 章的质押机制和隐性质押框架

共同搭建了一个覆盖广泛且严谨的机制设计框架，可以帮助用户更好地衡量 DON 的安全水平。这个框架还能使 DON 对参与节点提出各种安全要求，并保障高度的信任水平。

另外，还可以在 DON 中使用白皮书中提到的各类工具，实现特殊的信任模型要求，比如监管合规等。举个例子，节点可以采用 4.3 章节中提到的技术，提供节点运营商相关资质的证明，如：运行地区，以保障符合 GDPR（一般数据保护法）第三条“适用范围”的相关规定 [105]。去中心化的系统中很难符合此类监管规定 [45]。

另外，第 7 章会讨论 Chainlink 计划如何通过在主链上建立信任最小化机制从而增强 DON 的稳健性。

3 去中心化预言机网络的接口和功能

这一部分将阐述 DON 可以实现的简单但强大的接口，以简要概述 DON 的功能。

DON 中的应用由可执行程序（executable）和适配器（adapter）组成。可执行程序的核心逻辑是一个确定性程序，类似一个智能合约。可执行程序还伴随许多启动器（initiator），这类程序在预先设定的事件发生时，将调用可执行程序代码逻辑的入口点（entry point）。比如在特定时间（比如 cron job），当价格超过阈值时，触发执行。这跟 Keepers 的原理很像（参见 3.6.3 章节）。适配器可以接入链下资源，由启动器或可执行程序的核心逻辑调用。启动器和适配器的行为基于链下资源，所以并非确定性的。

我们可以用计算机系统中的三种典型资源，即：网络连接、计算和储存，来描述开发接口以及可执行程序和适配器的功能。下文简要描述了这三种资源，具体内容参见附件 B。

3.1 网络连接

DON 中的可执行程序通过适配器作为接口，与 DON 以外的系统双向发送和接收数据。这里适配器的概念涵盖了 Chainlink 目前已经推出的适配器 [20]。适配器可以是双向的，可以在 DON 和 web 服务器之间双向传输数据。它还可以采用分布式协议和安全的多方计算等加密功能。

图 9 展示了可以创建适配器的外部资源示例。其中包括：

- 区块链：适配器可以定义如何向区块链传输交易，以及如何读取区块链、单笔交易或其他状态。适配器还可以专门接入区块链的交易池。（参见第 3.5 章节。）

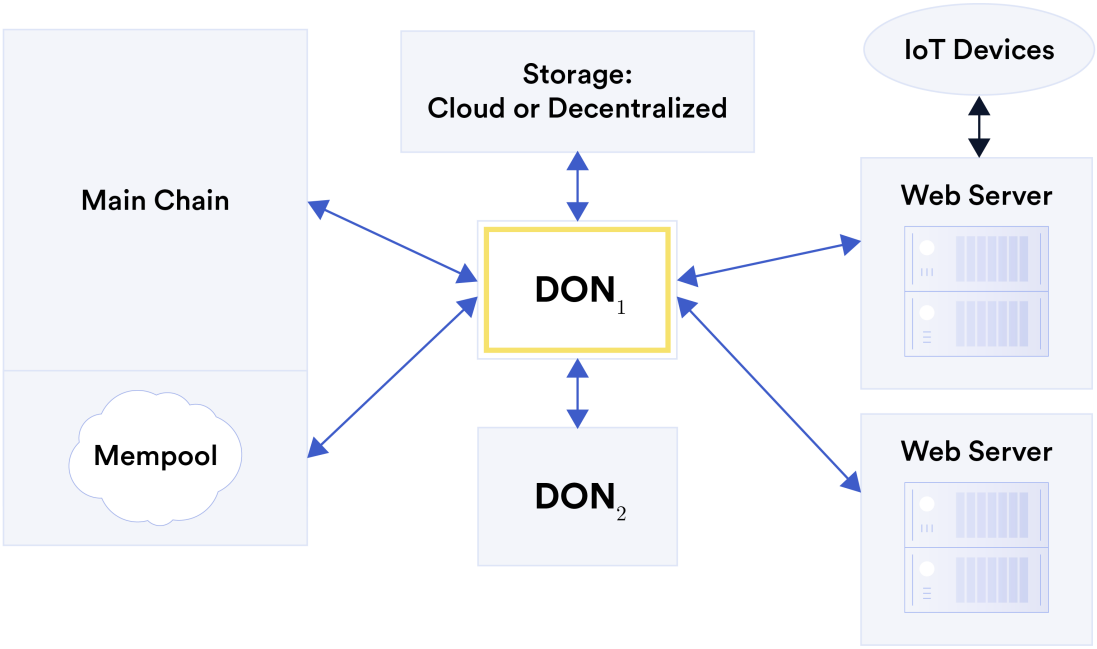

图 9: 适配器将 DON （用 DON_1 表示）与一系列资源连通，其中包括其他 DON （用 DON_1 表示）、区块链（MAINCHAIN）及其交易池、外部存储系统、web 服务器以及物联网设备（通过 web 服务器连接）。

- *Web 服务器*：适配器可以定义 API 接口，通过 API 从 web 服务器接收数据，包括原本无法接入 DON 的遗留系统。这种适配器本身也包含 API 接口，可以将数据传输至服务器。DON 接入的 web 服务器还可以作为网关接入物联网设备等其他资源。
- *外部存储*：适配器可以定义读写 DON 以外的存储服务的方式，比如去中心化的文件系统 [40, 187] 或云存储平台。
- *其他 DON:* 适配器可以在 DON 之间相互传输数据。

预期 DON 在部署初期会通过一套适配器接入常用的链下数据源，之后 DON 节点将发布针对具体 DON 的适配器。目前智能合约开发者正在不断开发出新的适配器，我们相信这些开发者将使用这一新功能开发出更强大的适配器。

最终，用户将能够以无须许可的方式创建新的适配器。

适配器必须保障 DON 能够持续稳定地接入链下资源。比如，云存储平台会要求维护云服务账户。另外，DON 可以代表用户或可执行程序，以去中心化的方式管理

私钥（如 [160]）。因此，DON 能够操控加密货币等资源，进行各种操作，如：向目标区块链发送交易。

参见附件 B.1 了解 DON 适配器的具体细节，并参见附件 C 了解部分适配器用例。

3.2 计算

可执行程序是 DON 的基本代码单元。一个可执行程序可表示成 $\mathsf{exec} = (\mathsf{logic}, \mathsf{init})$。其中，$\mathsf{logic}$ 是确定性程序，有一系列指定的入口点 $(\mathsf{logic}_1, \mathsf{logic}_2, \ldots, \mathsf{logic}_\ell)$，$\mathsf{init}$ 指一组对应的启动器 $(\mathsf{init}_1, \mathsf{init}_2, \ldots, \mathsf{init}_e)$。为了充分保证 DON 的可审查性，可执行程序的 logic 将底层账本 $\mathcal{L}$ 用于所有数据输入和输出。因此，适配器输入到可执行程序的任何数据都必须先储存在。

启动器： 目前，Chainlink 的启动器可以基于事件在 Chainlink 节点上执行任务 [21]。DON 中的启动器功能基本上与目前 Chainlink 的启动器差不多。只不过，DON 的启动器专门与一个可执行程序挂钩。启动器可以基于链下事件或状态、当前事件或 DON 状态执行。由于启动器需要基于外部事件执行，因此不具有确定性（当然适配器也是如此）。启动器可以在单个 DON 节点中执行，因此无需接入适配器。（参见下方案例 1 ）

启动器是区分可执行程序和智能合约的关键功能。由于可以用启动器来启动可执行程序运行，因此可执行程序可以自动运行，此外混合型智能合约也可以嵌入可执行程序。Chainlink Keepers 就是这样一种启动器，可以提供自动交易服务，基于预言机报告触发智能合约执行（如清算抵押率不足的贷款以及执行限价单交易）。

为了方便起见，DON 的启动器还可以被视作是对可执行程序服务协议的具体阐述，定义了 DON 调用可执行程序的条件。

下方示例解释了启动器在可执行程序中的运行机制：

Example 1 (基于价格偏差更新喂价). *每当一对资产（如 ETH-USD）之间的汇率变化较大时（如 1% 的变动），智能合约 SC 可能需要新的喂价（参考 3.6.3）。Chainlink 目前已发布了对价格波动敏感的喂价，但我们可以探讨一下如何通过 DON 的可执行程序 $\mathsf{exec}_{\mathsf{feed}}$ 实现这个功能。*

可执行程序 $\mathsf{exec}_{\mathsf{feed}}$ 在 $\mathcal{L}$ 上维护最新的以太币/美元价格 r，使用 $\langle \texttt{NewPrice} : j, r \rangle$ 格式的数据，其中 j 表示每次价格更新时递增的索引。

启动器 init_1 使每个节点 $\mathcal{O}_i$ 监控当前的 *ETH-USD* 价格与最近储存的索引为 j 的价格 r 是否存在至少 *1%* 的偏差。一旦监测到这样的价格偏差，$\mathcal{O}_i$ 会将新的当前价格写入$\mathcal{L}$，使用 $\langle \texttt{PriceView} : i, j+1, r_i \rangle$ 的数据格式。

当至少 k 个由不同节点创建的索引为 $j+1$ 的新价格数据的 PriceView 条目在账本 $\mathcal{L}$ 上累积时，将触发第二个启动器 init_2。接着，init_2 会调用 logic_2 入口点，计算前 k 个有效更新价格的中位数 ρ，并在 $\mathcal{L}$ 上更新一个值 $\langle \texttt{NewPrice} : j+1, \rho \rangle$。（实际操作中，节点可以轮流向 $\mathcal{L}$ 更新数据。）

第三个启动器 init_3 监控$\mathcal{L}$ 上的 NewPrice。一旦出现 $\langle \texttt{NewPrice} : j, r \rangle$ 的新报告，则调用 logic_3 入口点，通过适配器将 (j, r) 发送至 SC。

正如上文所述，可执行程序的功能与智能合约类似。然而，可执行程序比智能合约性能更强，除此之外，还存在两个关键差异：

1. 隐私性：可执行程序可进行保密计算，即保密程序可以处理明文数据输入，或公开程序可以处理保密数据输入，或二者结合。在简单的模式下，DON 节点可以访问保密数据，节点将隐藏中间结果，只向 MAINCHAIN 披露经过保密处理的数值。另外，还可以将敏感数据设置为 DON 自己也无法查看：DON 可以支持多方通讯（如 [42, 157]）和可信执行环境（TEE）（如 [84, 133, 152, 228]）。[6]

2. 辅助作用：可执行程序的作用是辅助而非替代链上智能合约。可执行程序存在一些智能合约所没有的限制：

   (a) 信任模型：可执行程序基于 DON 定义的信任模型运行：可执行程序要正确执行，前提是 $\mathcal{O}$ 的行为是诚实的。（而主链则可以创建一些 “安全护栏”，防止 DON 的不当操作造成的影响。7.3 章节将详细讨论。）

   (b) 资产权限： DON 可以操控区块链上的账户，因此可以通过适配器控制账户中的资产。但是 DON 无权表示主链上创建的资产（如以太币或其他 ERC20 通证），资产所有权记录在资产所在的区块链上。

   (c) 生命周期： DON 可以为了某一目的临时组建，具体规则依据 DON 和依赖合约创建者之间签署的链上服务协议而定。相比之下，区块链是永久性的档案系统。

参见附件 B.2，了解关于 DON 计算的具体内容。

[6]除此之外，还可以设置可执行程序向 DON 节点保密，不过目前只能通过 TEE 应用于非平凡（non-trivial）可执行程序。

3.3 存储

DON 实行委员会制度，可以持续在$\mathcal{L}$ 上储存一定量的数据，并且储存成本远远低于无须许可的区块链。另外，DON 还可以通过适配器接入链下去中心化系统储存数据，如 Filecoin [85]，并且将这些系统接入链上智能合约。这个方案尤其适用于储存大量数据的情况，解决区块链现存的“臃肿”问题。

因此，DON 在本地或外部系统中储存数据，并在其服务中使用这些数据。除此之外，DON 还可以对数据进行加密计算，如：（1）在 DON 节点中加密共享，或用密钥加密，密钥由 DON 节点管理，可进行安全的多方计算或部分或完全同态加密（homomorphic encryption）；（2）使用可信执行环境保护隐私。

我们预期 DON 会采用与智能合约系统相同的的内存管理模式。可执行程序只能写入自己的内存，但是可以读取其他可执行程序的内存。

参见附件 B.3，了解关于 DON 储存的具体内容。

3.4 交易执行框架（TEF）

DON 的目的是支持 MAINCHAIN （或多条主链）上的智能合约。第 6 章将详细讨论交易执行框架（TEF），这是一个通用技术方案，可以连接 MAINCHAIN 和 DON，高效执行合约 SC 。TEF 可以同时兼容公允排序服务（FSS）和 layer-2 技术。实际上，TEF 将有可能成为实现 FSS 功能的主要平台（因此，我们在这章不会进一步讨论 FSS）。

简而言之，TEF 将原本为 MAINCHAIN 开发的智能合约 SC 重构成混合型智能合约。这种重构产生了混合型智能合约的两个互操作的模块，即：其一为 SC_a，为了表述清晰，我们将其称为 anchor contract（主链合约）；其二为 DON 的可执行程序 exec_s。合约 SC_a 负责托管用户资产，执行权威的状态变更，并且建立“安全护栏”（参见 7.3 章节），以防止 DON 失效。可执行程序 exec_s 负责排序交易，并提供相关的预言机数据。可执行程序通过多种方式为 SC_a 打包交易，其中包括基于有效性证明的 rollup 或 optimistic rollup，以及 DON 执行保密操作等。

我们将开发工具，帮助开发者更好地将高级语言撰写的合约 SC 划分成 MAINCHAIN 和 DON 逻辑，即：SC_a 和 exec_s，并安全高效地将这两个模块组合在一起。

使用 TEF，为高性能交易机制集成高性能预言机，这是预言机实现扩容的关键要素。

3.5 交易池服务

我们希望在 DON 中部署一个重要的应用层功能，以实现 FSS 和 TEF，那就是交易池服务（Mempool Service，简称 MS ）。MS 可以看作是性能一流的适配器，可以兼容遗留系统处理交易。

MS 从链上交易池获取需要传输至 MAINCHAIN 智能合约 SC 的交易。接着，将这些交易传输至 DON 中的可执行程序，按具体要求处理交易。

DON 可以为 MS 传输的交易排序，然后直接传输至 SC ，或者传输至另一个调用 SC 的合约。比如说，DON 可以通过 MS 获取交易，也可以利用 MS 数据为发送至 MAINCHAIN 的交易设置 gas 价格。

由于 MS 负责监控交易池，因此可以从直接与 SC 交互的用户中获取交易。这样一来，用户就可以继续使用现有软件生成交易（注：这些软件并不知道 MS 或者配置了 MS 合约的存在）。在这种情况下，必须更改 SC ，忽略原来的交易，只接受 MS 处理过的交易，以避免重复处理。

MS 可以与 FSS 和 TEF 结合，连接链上合约 SC 。

3.6 跳板：现有的 Chainlink 功能

3.6.1 链下报告（OCR）

链下报告（*OCR*）[60] 是 Chainlink 发布的机制，用于聚合预言机报告并传输至依赖合约 SC。OCR 近期部署到了 Chainlink 喂价网络，是 Chainlink 充分实现 DON 的第一步。

OCR 的核心是 BFT 协议，可以在部分同步网络（partially synchronous network）中运行。OCR 保障了报告的及时性和准确性，条件是任意错误节点数量 $f < n/3$。这个机制实现了拜占庭式的可靠广播，但并非完全意义上的 BFT 共识协议。如果账本中所有观点都相同，则节点无需维护消息日志。协议领导者可以在不影响安全的情况下含糊其辞。

OCR 目前只支持一种消息类型，即：聚合（至少 $2f+1$ 个）参与节点的数值并取中位数。OCR 为输入至 SC 的报告提供了关键保障，称作“证明报告”（attested reports）：证明报告中的中位数居于两个诚实节点报告的数值中间，这是 OCR 的关键条件。领导者可能对证明报告中的中位数有一定影响力，但必须遵守这一条件。OCR 还可以扩展至其他的聚合方式。

目前 Chainlink 网络无需完全实现 OCR 共识协议就可以达成在活跃度和准确性方面的目标，但是如果要实现传统 BFT 协议无法实现的功能，就必须要依靠 OCR，

比如：

1. 全有或全无的（*all-or-nothing*）链下报告广播： OCR 可以快速生成证明报告，向全部诚实节点广播，或不向任何节点广播。这个机制可以保障公平性，确保诚实节点有机会参与传输证明报告。

2. 可靠的传输：即使在存在恶意或错误节点的情况下，OCR 也能保障所有 OCR 报告和消息都在规定时间内传输至 SC。这个机制保障了网络活跃度。

3. 基于合约的信任最小化： SC 可以过滤可能存在错误的 OCR 报告。比如，如果报告的数值严重偏离接收到的其他数值，则将其剔除。这在更大程度上保障了数据准确性。

这三种机制都对 DON 起到了相应作用。全有或全无的链下（DON）广播为可靠的传输提供了关键的加密经济保障，这也是不可或缺的适配器功能。SC 的信任最小化机制提供了一个 “防安全护栏”，7.3 章节中将详细讨论。

OCR 为 Chainlink 预言机网络中部署并完善 BFT 协议奠定了基础，因此，如上文所述，是充分实现 DON 的第一步。

3.6.2 DECO 和 Town Crier

DECO [233] 和 Town Crier [232] 是一对互相关联的技术，目前正在 Chainlink 网络开发中。

如今多数 web 服务器都可以使用一种叫 TLS 的协议 [94] 将用户接入安全的通道。（HTTPS 是在 HTTP 基础上添加 TLS，即：URL 前缀中的 “https” 表示用 TLS 保障安全。）然而，大部分 TLS 服务器都存在一个明显的瓶颈，那就是它们无法在数据上添加数字签名。因此，用户或证明者无法向第三方或验证者（比如预言机或智能合约）展示他收到的数据，并保障数据来源的真实性。

即使服务器可以在数据上添加数字签名，也会面临隐私问题。证明者可能会想要先编辑或修改敏感数据，再将其展示给验证者。然而，数字签名的作用恰好是证明修改过的数据是无效的。因此，证明者无法出于隐私保护目的对数据做修改。（7.1 章节会详细讨论。）

DECO 和 Town Crier 可以允许证明者从 web 服务器获取数据，并向验证者展示数据，同时保障数据完整性和隐私性。这两个系统可以保障证明者向验证者展示的数据源自目标服务器，以此保护数据完整性；并在此基础上允许证明者编辑或修改数据，以此保护数据隐私。

这两个系统都有一个关键功能，那就是无需对目标 web 服务器做任何修改，系统可以与任何现有的 TLS 服务器交互。实际上，它们对服务器来说是透明的，即：对服务器来说，证明者只是创建了一次普通的连接。

两个系统的目标一致，但信任模型和实现方式有所不同。接下来我们来简单解释一下。

DECO 在底层采用加密协议，以保护数据完整性和隐私。使用 DECO 与目标服务器创建会话时，证明者还会通过协议与验证者交互。这个协议能够使证明者向验证者证明其在当前会话中从服务器收到了一条数据 D。证明者还可以向验证者提供零知识证明，证明 D 的某些属性，而不是直接披露 D 的内容。

举个例子，某一用户或某一节点可以与 web 服务器建立私人会话，从中将数据 D 导出到 DON 中的所有节点。然后，整个 DON 可以验证 D 的真实性（或通过零知识证明验证 D 的一个属性是否真实）。后面我们还会举其他例子，除此之外，这个功能还能使 DON 更完整地访问数据源。即使只有一个节点可以直接访问数据源（可能是因为数据提供商只对一个节点开放），整个 DON 也可以验证节点报告的准确性。

Town Crier 采用英特尔 SGX 等可信执行环境（TEE）。简而言之，TEE 的功能类似黑盒，在其中可以防篡改且保密地执行应用。理论上来说，即使是 TEE 所运行的宿主管理者也无法私下更改 TEE 中的应用或查看应用状态，因为其中可能包含隐私数据。

Town Crier 可以实现 DECO 所有具备以及不具备的功能。DECO 约束证明者只能与一个验证者交互，而 Town Crier 中的证明者则可以对来自目标服务器的数据 D 生成一份可验证的公开证明，任何人，甚至是智能合约都可以直接验证。Town Crier 还可以安全地使用这些隐私数据（如用户密码）。

Town Crier 的最大瓶颈在于它必须依赖可信执行环境。目前已发布的 TEE 存在一系列严重的安全漏洞，不过这项技术还在发展初期，之后肯定会不断成熟。参见 B.2.1 和 B.2.2 章节，了解关于 TEE 的更多内容。

4.3、4.5 和 9.4.3 章节以及附件 C.1 中描述了几个 DECO 和 Town Crier 的案例。

3.6.3 目前已发布的链上 Chainlink 服务

Chainlink 预言机网络在众多区块链和去中心化系统中已发布了许多关键服务。本白皮书中提到的技术发展将提升这些服务的功能和范围。目前已推出了如下三种服务：

数据： 如今，大多数智能合约接入 Chainlink 都是为了获取链下数据。Chainlink 接入权威链下数据源，传输关键数据的当前数值。比如，Chainlink 喂价接入交易所或数据聚合商，报告加密货币、大宗商品、外汇、指数和证券等各类资产的价格。Chainlink 喂价目前已为 Aave [147] 以及 Synthetix [207] 等 DeFi 系统保障了数十亿美元的链上资产。除此之外，Chainlink 还可以传输天气数据至参数型农作物保险 [75]、选举数据 [93] 以及其他各类数据。

一旦部署了 DON 以及白皮书中提到的其他技术，将在许多方面提升 Chainlink 网络的数据传输功能：

- 扩展性： OCR 以及之后将部署的 DON 将在众多区块链上极大提升 Chainlink 服务的可扩展性。比如，DON 将使 Chainlink 节点传输的数据量从几百提升至几千或甚至更多。扩容后，Chainlink 生态将可以为智能合约提供任何所需的数据，满足现在和以后的各种需求。

- 提高安全性： DON 可以储存中间过程报告，因此将保存节点服务记录，准确监控并衡量节点表现和数据准确性，为 Chainlink 节点声誉系统提供强大经验支撑。FSS 和 TEF 将以灵活的方式结合喂价和交易数据，避免矿工抢跑等攻击。（显性）质押机制将增强现有的加密经济机制，保护数据安全。

- 数据敏捷性： DON 可兼容任何区块链系统（或更宽泛地说，可以兼容任何用户系统），因此可以将数据传输至任何系统中。DON 可以同时将数据传输至不同区块链，无需针对每条区块链建立预言机网络。DON 还可以快速将现有数据发送至新的区块链，并向已经接入的区块链发送新的数据。

- 隐私性： DON 可以开展通用型计算，可以在链下计算敏感数据，避免隐私在链上曝光。除此之外还可以使用 DECO 或 Town Crier，在更大程度上保护隐私，基于隐私数据生成报告，即使连 DON 节点都无法查看数据。参见 4.5 和 4.3 章节了解详细用例。

可验证随机函数（VRF）： 某些 DApp 需要接入可验证的准确随机数，保障应用的公平性。非同质化通证（NFT）就是其中一个案例。Aavegotchi [23] 和 Axie Infinity [35] 接入 Chainlink VRF，为 NFT 赋予稀缺性；Ether Cards [102] 接入 Chainlink VRF，通过抽奖形发放 NFT；各类游戏 DApp 需要随机的游戏结果；PoolTogether [89] 的无损储蓄游戏以及各类非常规金融工具需要随机选出中奖者。其他区块链和非区块链应用也需要安全的随机数来源，比如选举去中心化系统的委员会以及彩票抽奖等。

区块哈希值可以作为随机数，但它们容易被矿工操纵（在某种程度上甚至还会被向区块发送交易的用户操纵）。Chainlink VRF [78] 相比之下是更加安全的方案。预言机拥有一对私钥和公钥 $(\mathsf{sk}, \mathsf{pk})$，私钥在链下保存，公钥 pk 是公开的。为了生成随机数，预言机会对链上合约提供的不可预测的种子 x（注：种子可以是区块哈希值或者 DApp 具体的参数）使用 sk，调用函数 F，导出结果 $y = F_{\mathsf{sk}}(x)$，并生成一份证明。（参见 [179] 了解 Chainlink VRF。）VRF 之所以可以验证，是因为可以使用 pk 对证明以及 y 进行验证。因此，攻击者如果无法预测到 x 或获得 sk，就不可能预测到 y，也无法进行操纵。

Chainlink VRF 可以看作是一种在链下托管私钥的应用。更笼统地来说，DON 可以为应用和用户安全、去中心化地储存密钥，并将这个功能与通用计算相结合。这将催生出一系列应用，本白皮书之后会具体解释，其中包括为储备金证明（参见 4.1 章节）和用户的去中心化身份认证（以及其他数字资产）（参见 4.3 章节）管理密钥。

Keepers: 开发者可以使用 Chainlink Keepers [87] 编写代码，去中心化地执行链下任务，通常是触发链上智能合约执行。在 Keepers 出现前，开发者往往需要手动运行这些链下逻辑，这会导致中心化的单点故障（以及大量重复开发）。Keepers 可以为开发者提供简单易用的框架，将这些操作去中心化，缩短开发周期，并提升网络的活跃度和以及其他安全性能。Keepers 可以触发各种类型的任务，比如基于价格清算贷款或执行金融交易以及基于时间启动空投或付款等。

在 DON 框架中，Keepers 可以被看做是其中一种启动器。启动器可以连接适配器，因此可以接入链上链下系统中模块化的代码库，可以快速开发出安全复杂的功能。启动器可以启动可执行程序中的运算，可执行程序可实现 DON 的所有功能，可以为链上和链下应用提供各种白皮书中提到的去中心化服务。

3.6.4 节点声誉 / 历史记录

目前 Chainlink 生态在链上记录节点的服务历史。这个功能催生出了一系列围绕节点声誉展开的资源，可以分析并过滤节点运营商和数据源的性能数据，并开发可视化工具。用户可以参考这些数据，理性选择节点，并监控现有节点网络的运行情况。DON 也设立了同样的功能，帮助用户做出选择。

比如，节点运营商可以将其预言机服务挂在 market.link 等无须许可的市场中，并通过 Keybase [4] 等服务验证其链下身份，这类服务可以将 Chainlink 预言机与运营商的域名和社交媒体账号关联。另外，用户还可以使用 market.link 和 reputation.link 等预言机性能分析工具查看预言机节点的历史服务记录，其中包括节点平均响应延

迟、上报数据与最终传输到链上的共识结果之间的偏差值、产生的收入以及完成的任务等。用户还可以使用这些分析工具追踪各个预言机网络对其他用户的服务记录，这间接证明了这些节点的声誉。最终，会形成一张“信任网络”，被高价值的去中心化应用选择的节点将获得更多人的信任，其他用户可以看到节点服务的对象，并基于此做出判断。

OCR 是实现 DON 的第一步，而 DON 的建立将使大部分交易处理和合约操作转移到链下。在 DON 中可以建立去中心化的模式，储存节点服务数据。实际上，由于 DON 拥有强大的性能和数据存储能力，因此可以非常精细地储存节点服务数据，并对其展开去中心化计算，生成可信的结果，传输至声誉系统中，并在 MAINCHAIN 上查看。

理论上，如果 DON 中大多数节点被收买，DON 有可能输出错误的结果，但是整个 DON 向链上传输数据的记录都可以在 MAINCHAIN 上查看，因此是不能造假的。除此之外，我们还计划建立有效机制，激励 DON 中节点发现并举报问题节点。比如，DON 会聚合最先上报的预言机数据，并传输到链上，这样可以激励节点对错误报告提出挑战：错误数据被放到集合中，意味着本来应该被放到集合中的正确数据被排除在外，因此受到不应该的惩罚。DON 反复报告失败也会导致诚实节点离开。

真实的服务记录以去中心化的方式被储存，用户就可以有效甄别高性能的节点，而节点运营商也可以树立声誉，这些是 Chainlink 生态最与众不同的功能。第 9 章将详细阐述这些功能为何对 DON 实现稳健的经济安全至关重要。

4 去中心化预言机网络的去中心化服务

本章将列举五个 DON 的用例，并具体阐述实现这些用例的混合型智能合约，以展示 DON 的丰富功能及其实现的各种服务。(1) 储备金证明，即一种跨链服务；(2) 接入企业遗留系统，即创建基于中间件的抽象层，辅助开发区块链应用，开发者几乎无需掌握具体区块链的代码或专长；(3) 去中心化身份认证，用户可以获得并保管自己的身份文件和证明；(4) 优先通道，可以及时将关键交易（如预言机报告）发送至链上；(5) 保障隐私的 DeFi，这种金融智能合约可以向参与各方隐藏隐私数据。在这里，我们用 SC 代表混合型智能合约的链上部分，并用可执行文件 exec 符号代表 DON 的部分。

4.1 储备金证明

许多应用都需要在区块链之间传递状态。其中一类常见的应用就是“包装”(wrapping) 加密货币。WBTC [15] 就是这样一种包装虚拟货币，在去中心化金融领域越来越常见。它的工作原理是这样的：包装起来的储备资产被存放在原始区块链 $\mathsf{MAINCHAIN}^{(1)}$ 上，并在另一个目标区块链 $\mathsf{MAINCHAIN}^{(2)}$ 上创建对应的通证。比如，WBTC 是以太坊区块链上的 ERC20 通证，与比特币区块链上的比特币挂钩。

由于 $\mathsf{MAINCHAIN}^{(2)}$ 上的合约无法直接查看 $\mathsf{MAINCHAIN}^{(1)}$ 的内容，因此必须直接或间接接入预言机，获取智能合约中的包装资产余额报告，并生成我们常说的 “储备金证明”。比如在 WBTC [15] 的用例中，托管方 BitGo 持有比特币，并发行 WBTC，同时接入 Chainlink 网络生成储备金证明 [76]。

DON 本身也可以提供储备金证明。只不过 DON 在这个方向往前走得更远。DON 可以管理保密数据，通过接入适配器，可以在任何区块链上展开交易。因此，DON 可以作为包装资产其中一个托管方，或甚至是唯一的去中心化托管方。DON 可以提升现有储备金证明服务的安全性。

比如，假设 $\mathsf{MAINCHAIN}^{(1)}$ 是比特币、$\mathsf{MAINCHAIN}^{(2)}$ 是以太坊。智能合约 SC 在 $\mathsf{MAINCHAIN}^{(2)}$ 上发行 WBTC 通证。DON 控制了比特币地址 $\mathsf{addr}^{(1)}_{\mathsf{DON}}$。用户 $\mathcal{U}$ 要创建 WBTC，可以从 $\mathsf{addr}^{(1)}_{\mathcal{U}}$ 发送 X 个比特币以及 $\mathsf{MAINCHAIN}^{(1)}$ 地址 $\mathsf{addr}^{(2)}_{\mathcal{U}}$. 到 $\mathsf{addr}^{(1)}_{\mathsf{DON}}$。DON 通过连接 $\mathsf{MAINCHAIN}^{(1)}$ 的适配器监控 $\mathsf{addr}^{(1)}_{\mathsf{DON}}$。查看 $\mathcal{U}$ 的余额并获得足够多的保障后，DON 会通过适配器向 $\mathsf{MAINCHAIN}^{(2)}$ 的 SC 发送一条消息。这条消息会通知 SC 为 $\mathsf{addr}^{(2)}_{\mathcal{U}}$ 铸造 X 个通证。

如果 $\mathcal{U}$ 要释放 X 个通证，则反向操作。而在 $\mathsf{MAINCHAIN}^{(1)}$ 上，$\mathsf{addr}^{(1)}_{\mathsf{DON}}$ 会发送 X 个比特币到 $\mathsf{addr}^{(1)}_{\mathcal{U}}$（或者根据用户请求发送到其他地址）。这些协议还可以经过调试兼容交易平台，而不是直接与用户交互。

4.2 接入企业 / 遗留系统

DON 可以连通不同区块链，比如上文提到的储备金证明用例，但其另一个作用是双向连通区块链和遗留系统 [175] 或央行数字货币等类似区块链的系统 [30]。

企业将现有系统流程接入去中心化系统时往往面临一系列挑战，如：

- 区块链行业发展速度快：区块链系统瞬息万变。新的区块链层出不穷，而且应用规模不断扩大，吸引了越来越多企业，但它们本身的基础架构却无法与区块链兼容。总的来说，区块链独有的特质使得企业很难跟上整个生态的发展。

- *接入每条区块链需要专门的开发资源*：对许多公司来说，招聘或内部培养前沿区块链专业团队非常困难，特别是因为区块链行业发展非常迅速。
- *私钥管理*：管理区块链或加密货币私钥所需的运营专长不同于传统网络安全领域的能力，而且许多企业都不具备这个能力。
- *隐私性*：企业都不希望将其内部隐私数据在链上公开。

为了解决上述前三个挑战，开发者可以使用 DON 作为安全的中间件，帮助企业系统读写区块链上的内容。DON 可以为开发者和用户简化技术问题，比如 gas 费波动和区块链重组等。DON 可以简化区块链与企业系统的交互方式，并在很大程度上简化企业区块链应用的开发流程，消除企业招聘或培养区块链专业开发团队的负担。

有了 DON，企业开发者就可以专注于开发智能合约应用，并兼容大部分区块链。这样一来，DON 能接入多少区块链，就意味着企业用户可以轻松访问多少区块链。开发者可以将应用从一个区块链导入另一个区块链，并尽量不对内部开发的应用做调整。

开发者还可以部署本白皮书中介绍的工具，解决 DON 应用的隐私问题。其中包括 3.6.2 章节中提到的 DECO 和 Town Crier、7.1.2 章节中提到的保障隐私 API 以及本章接下来会提到的针对具体应用的各种方式。这些 DON 可以为企业系统状态提供可靠的链上证明，同时向区块链隐藏企业敏感数据。

4.3 去中心化身份

*去中心化身份*泛指用户获取并保管自己的身份认证信息，而不是将其托管在第三方平台。去中心化身份认证可以证明持有者的属性和断言，通常称为“claims”（主张）。身份认证通常包含某一实体的电子签名，我们通常称为“issuer”（发行方），issuer 有权将 claim 与用户关联。在最常见的机制中，claim 与一个去中心化身份认证（DID）关联，即向某一用户提供通用身份认证。身份认证与一个公钥绑定，而相对的私钥由用户保管。用户可以用私钥来证明其持有 claim。

虽然去中心化身份认证非常具有前瞻性，但目前已实现或提出的方案（如 [14, 92, 129, 215]）存在三个严重的瓶颈：

- *无法与遗留系统兼容*：现有的去中心化身份认证系统需要创建一个有权限发行 DID 证明的 issuer 社区。由于现存 web 服务通常没有数字签名功能，因此必须专门建立 issuer 系统。但是如果不建立去中心化身份认证生态，又没有动力去

这么做，因此这是一个先有鸡还是先有蛋的问题。换句话说，现在我们还不知道如何启动 issuer 生态。

- *私钥管理不具有操作性*：去中心化身份认证系统要求用户自行保管私钥，介于加密货币的经验来看，这个方案并不可行。据估计有大约 400 万个比特币由于私钥丢失而再也无法找回 [193]，许多用户将加密资产存放在交易平台上 [192]，这也降低了去中心化水平。

- *无法在保障隐私的基础上实现抗女巫攻击*：应用中的基本安全要求包括投票以及在通证销售过程中公平分配通证，而要满足这些要求，用户必须只能主张一个身份。目前的去中心化身份认证方案都要求用户披露其现实中的身份，以抵抗女巫攻击，因此可能无法保障隐私。

然而，我们可以使用一组节点在 DON 中开展分布式计算，并使用 DECO 或 Town Crier 等工具来解决这一问题，CanDID 系统正是采用了这个方案 [160]。

DECO 或 Town Crier 可以无需任何修改，就可以将现有 web 服务变成保障隐私的身份认证发行方（issuer）。这样一来,DON 可以将相关数据导出至身份认证中，并同时隐藏敏感数据。

另外，DON 允许用户通过加密共享的方式储存私钥，因此可以随时找回密钥，解决了密钥管理的问题。用户可以使用 Town Crier 或 DECO 向节点提供证明，以找回密钥。可以用预设的 web 提供商（如 Tweeter、Google 和 Facebook）登录账户。相对 OAUTH 来说，使用 Town Crier 或 DECO 的优势是可以保护用户隐私。用户可以使用这两种工具向 DON 隐藏 web 服务提供商信息，因为通常可以基于此推断出现实世界中的身份。

最后，DON 可以在用户注册后将其独一无二的现实世界身份证明（如社保账号）转换成链上身份证明，并同时保障隐私，以抵御女巫攻击 [160]。系统无需获取隐私数据也可以检测到重复注册，社保账号等隐私数据不用向 DON 节点披露。[7] DON 可以在 Hyperledger Indy [129] 等公链或联盟链上代表链下去中心化身份认证系统提供任何一种上述服务。

应用案例：KYC： 去中心化身份认证可以降低区块链上金融应用的门槛，并同时增强用户隐私保护。它可以应对两个挑战，即：认证和合规监管，遵守 AML/KYC 相关规定。许多国家的 AML（注：反洗钱）法规要求金融机构等企业创建并验证交

[7] 实现这一转变的关键是“分布式伪随机函数”（PRF）。

易中相关个人和公司的身份。KYC 是金融机构 AML 政策中的一部分，通常指监控用户行为和资金流向等。

KYC 通常指以某种方式表示用户身份认证（如：填写网页表格或者用户在视频通话中手持身份证明文件等）。安全地创建并表示去中心化身份认证，理论上拥有以下几个好处：（1）为用户和金融机构提高 KYC 流程效率，因为一旦获取了身份认证，就可以无缝传输到任何金融机构。（1）减少因身份盗窃而发生的欺诈行为（如盗用个人身份信息或在视频验证中造假）。（2）由于用户自己保管个人数据，因此也降低了个人信息盗窃风险。

目前金融机构由于未能遵守 AML 法规而被罚款几十亿美元，许多金融机构每年会花几百万美元用于保障 KYC，因此安全的去中心化身份认证将为金融机构及其用户大幅缩减成本 [195]。虽然传统金融行业采用创新合规工具的脚步很慢，但 DeFi 领域却正在快速拥抱这项新技术 [43]。

应用案例：低抵押贷款： 如今大多数 DeFi 借贷产品都只支持抵押率充足的贷款。贷款人抵押的加密资产价值必须超出贷款价值。近期，DeFi 社区出现了一种新的贷款模式，称为“低抵押贷款”（under-collateralized loans）。这类贷款的抵押资产价值小于贷款价值。低抵押贷款类似传统金融机构发行的贷款。这类贷款基于贷款人的信用记录发放，而不是通过抵押资产保证还款。

低抵押贷款是 DeFi 借贷市场的最新发展方向，与传统金融机构采用相同的机制，如法律合约 [91]。其中的关键环节是获取用户信用记录，并安全可靠地传输至 DeFi 系统，用户信用记录是传统借贷产品的关键要素。

接入 DON 的去中心化身份认证系统可以为有意向的贷款人生成高度可信的身份证明，证明其信用记录，并同时保护隐私数据。具体而言，贷款人可以用权威在线数据源的记录生成身份证明，只披露 DON 验证的数据，无需公开其他隐私数据。比如，贷款人可以从一组征信机构获取信用分，并证明信用分超过某一阈值（如：750），同时无需披露具体的信用分或其他数据。另外，用户还可以选择匿名生成这些证明。也就是说用户姓名也可以作为隐私数据被隐藏，无需向预言机节点或去中心化认证披露。证明可以在链上或链下使用，视应用具体需求而定。

总而言之，贷款人可以向借款人提供关于其信用记录的关键信息，同时隐藏不必要的敏感数据。

贷款人还可以提供贷款所需的其他证明，并同时保障隐私。比如下一个用例中提到的贷款人（链下）资产证明。

应用案例：资格证明： 许多司法管辖区都限制了非公开证券的投资者类别。比如，美国证券交易委员会 D 条例规定个人必须净值超过 100 万美元、满足某些最低收入要求、或拥有某些专业资质，才能有资格投资此类证券 [208, 209]。目前的证明流程非常繁琐且低效，通常需要会计出示一份证明文件或相关证据。

用户可以使用去中心化的身份认证系统从在线金融服务账户生成认证，证明符合相关条例规定，并提高 KYC 流程的效率和隐私保护。另外，DECO 和 Town Crier 等隐私保护功能还可以增强这些认证的安全性，无需直接披露用户资产的具体细节。比如，用户可以生成认证，证明她个人净值超过 100 万美元，除此之外不许透露任何其他的资产状况。

4.4 优先通道

优先通道是一个非常实用的新功能，可以使用 DON 轻松实现。这个功能的目的是在网络拥堵期间向 MAINCHAIN 选择性地传输优先级别较高的交易。优先通道可以看作是一种关于区块空间的期货合约，因此也是一种加密大宗商品（cryptocommodity）。这个词是芝加哥项目首先提出的 [61, 136]。

优先通道功能专门针对矿工而非普通用户，旨在实现预言机和治理等基础架构服务。实际上，网络中算力不足 100% 时，优先通道只能保证大致的传输时间，因此可以防止矿工抢跑等对时间要求比较精确的操作。

优先通道是一名矿工或一组矿工（或矿池）$\mathcal{M}$ 与用户 $\mathcal{U}$ 之间签署的协议，矿工或矿池提供通道，用户支付费用。$\mathcal{M}$ 同意当 $\mathcal{U}$ 向交易池提交交易 τ 时（注：gas 价格为低于约定上限的任意价格），会将交易在之后 D 个区块内发送到链上。[8]图 10 具体描述了这个机制。

优先通道合约描述： 混合型智能合约可以通过下述方式实现优先通道。我们用 SC 表示 MAINCHAIN 上的逻辑，用 exec 表示 DON 的逻辑。

$\mathcal{M}$ 收到保证金$\$d$，并从 $\mathcal{U}$ 收到一笔预付款$\$p$。DON 可执行程序 exec 监控交易池，触发用户发送的交易。如果 $\mathcal{U}$ 发送的交易在规定时间内被 $\mathcal{M}$ 挖出，则发送消息 `success`，如果服务失败则发送消息 `failure` 。

如果收到 *success*，SC 会将预付款 $\$p$ 发送给 $\mathcal{M}$ ，如果收到 `failure` ，则会将包括 $\$d$ 在内的剩余所有资金发送给$\mathcal{U}$ 。成功完成后，保证金 $\$d$ 将退回给 $\mathcal{M}$ 。

[8] D 必须要足够大，确保 $\mathcal{M}$ 可以大概率地遵守。比如，如果 $\mathcal{M}$ 控制网络中 20% 的算力，可能会选择 $D = 100$，确保失败概率 $\approx 2 \times 10^{-10}$，即不到 10 亿分之一。

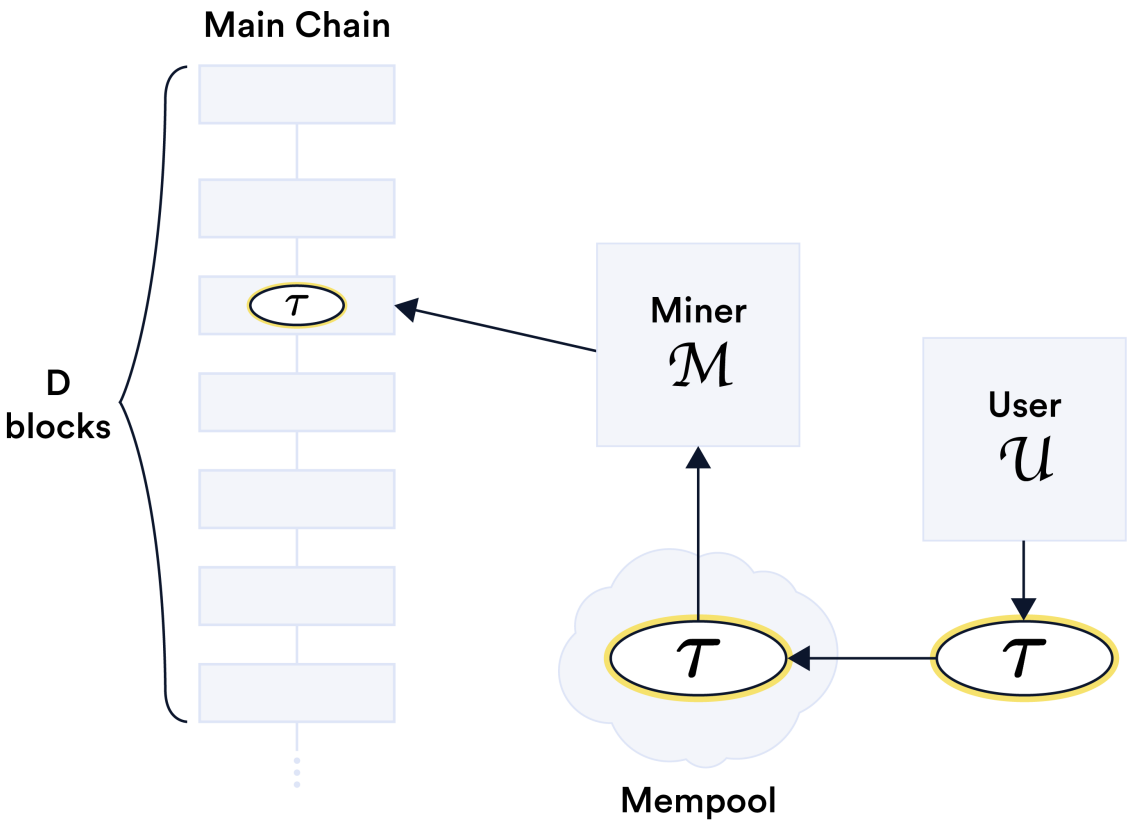

图 10: 优先通道指矿工 $\mathcal{M}$（或一组矿工 $\mathcal{M}$）向用户 $\mathcal{U}$ 保证其交易 τ 将在 D 个区块内发送至链上。智能合约 SC 可以利用 DON 监控优先通道服务的执行情况。

矿工 $\mathcal{M}$ 可以同时向多个用户提供优先通道，并且按照约定的消息数量为 $\mathcal{U}$ 开通优先通道。

4.5 保障隐私的 DeFi / Mixicles

如今，DeFi 应用 [1] 几乎无法为用户保障任何隐私：所有交易都可以在链上查看。各种零知识证明方案（如 [149, 216]）可以保护交易隐私，而且交易执行框架也可以兼容。但是这些方案并不全面，而且无法隐藏交易所用的资产。

DON 最后将实现一系列计算工具，以各种方式弥合这个缺口，弥补其他系统在隐私保障方面的不足。比如，Chainlink Labs 的研究者提出了隐私保护 DeFi 工具 Mixicles [135]，可以隐藏金融工具中的资产类别，并且可以完美集成至 DON 框架。

用最简单的话概括 Mixicles，就是它可以实现简单的二元期权（binary option）。二元期权是一类金融工具，其中两名用户（文中 [135] 统称“参与者”）针对某一事件

的两种可能的结果进行对赌，如：资产在某一时间点是否超出目标价格。具体请参照下述案例。

Example 2. *Alice 和 Bob 签署二元期权合约，对赌 Carol's Bubble Token* (CBT) *资产的价值。Alice 赌 CBT 在时间 $T = 2025$ 年 6 月 21 日中午时市场价格将超过 250 美元；Bob 是她的对家。每名参与者在具体截止时间前质押 100 个以太币。最后对赌的赢家收到 200 个以太币（即净赚 100 个以太币）。*

由于已经建立了 Chainlink 预言机网络 $\mathcal{O}$ ，因此可以轻松实现案例 2 协议的智能合约 SC 。两名参与者每人在 SC 中质押 100 个以太币。在 T 后的一段时间内，请求 q 被发送至 $\mathcal{O}$ ，请求 CBT 在时间点 T 的价格 r。$\mathcal{O}$ 将价格报告 r 发送至 SC 。如果 $r \geq 250$，SC 将钱打给 Alice，否则将打给 Bob。然而，这个方法会在链上披露 r，观察者很容易推断出二元期权的底层资产。

如果使用 Mixicles，可以将 SC 的结果抽象理解成是一个开关（Switch），传输计算出的布尔值 switch(r)。在这个案例中，如果 $r \geq 250$，则 switch$(r) = 0$，即意味着 Alice 赢了。否则，switch$(r) = 1$，即意味着 Bob 赢了。

DON 可以将 Mixicle 变成混合型智能合约，在运行可执行程序 exec，在链下计算 switch(r) ，并发送至链上 SC 。图 11 展示了这个架构。

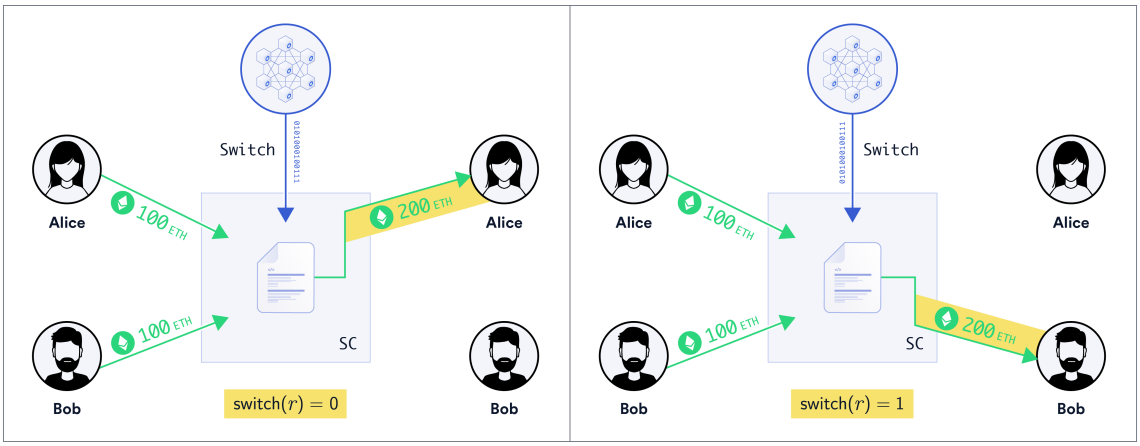

图 11: 案例 2 中的基础 Mixicle 架构。为了向区块链隐藏 r 数值，并保护二元期权底层资产的隐私，预言机通过 Switch 只向链上合约 $switch(r)$

附件中具体阐述了适配器 ConfSwitch，可以轻松在 DON 中实现这个目标。ConfSwitch 背后的基本逻辑非常简单。ConfSwitch 不报告 r 值，而是只报告 switch(r) 返回的布尔值。SC 的逻辑可以设置成只基于 switch(r) 正确付款，而 switch(r) 本身不

会披露关于底层资产的任何信息，即案例中的 CBT 。除此之外，适配器 ConfSwitch 在账本中放置一段 (q, r) 的秘文，用 $\mathsf{pk}_{\mathsf{aud}}$ 加密（注：审核者的公钥），可以保护审计追踪的隐私。

我们这里简要描述的基础版 Mixicle 只隐藏了资产和对赌内容。完整版的 Mixicle [135] 可以在两个方面提供隐私保护。它可以向观察者隐藏：（1）参与者对赌的内容（即 q 和 r）；以及（2）谁赢得了对赌。

由于 Mixicles 在 MAINCHAIN 上执行，其中任意一名参与者需要将 switch(r) 从 DON 发送至 MAINCHAIN ，或者可以创建一个可执行程序 exec，由 ConfSwitch 输出的结果触发，并能调用另一个适配器，将 switch(r) 发送至 MAINCHAIN 。

第三种更低调的保密功能也值得我们注意。在基础版 ConfSwitch 中，，因此可以得知我们案例中的资产 CBT 以及二元期权的性质。然而，如附件 C.3 所说，还可以使用 DECO 或 Town Crier 向 $\mathcal{O}$ 隐藏这些信息。这样一来，SC 的观察者一样都无法查看到任何保密信息。

若想了解关于 Mixicles 的更多详情，请参考 [135] 的内容。

5 公允排序服务

我们期待 DON 将利用其网络、计算和存储功能实现一个关键服务，那就是公允排序服务（FSS）。虽然大家可能会觉得 FSS 只是 DON 框架中的一个应用，但我们在这里要重点提到这个服务，因为我们认为它将有可能应用在各个区块链上，并且成为 Chainlink 网络的重要服务。

如今许多 DeFi 应用在公链上执行时，都会在链上公开数据，用户可以利用这些数据谋取私利，这有点类似传统市场中泄露和操纵内部信息 [64, 155]。而 FSS 可以为 DeFi 生态保障公平性。开发者可以利用 FSS 保障 DeFi 智能合约不会因为信息泄露而受到操纵。介于下文中的一些问题，FSS 对 layer-2 服务尤其重要，而且适用于第 6 章讨论的服务框架。

挑战： 在现存公链系统中，交易完全由矿工排序。在联盟链中，验证节点拥有同样的权利。这在本该去中心化的系统中导致了短暂的中心化问题，而这个问题并没有引起大家重视。矿工可以（暂时）操纵交易 [170] 或重新排序，以谋取私利，这个概念被称为矿工可提取价值（MEV） [90]。矿工可提取价值这个词具有一定欺骗性：因为它不仅指矿工可以获取的价值，普通用户也可以获取一部分 MEV。然而，由于矿工比普通用户拥有更大的权利，所以 MEV 代表任何实体通过将交易重新排序而榨取的

最大价值。即使矿工完全基于 gas 费为交易排序，用户也可以通过操纵 gas 价格将自己的交易排在其他交易前面。

Daian 等人 [90] 记录并量化了机器人（非矿工）利用 gas 价格损害 DeFi 用户价值的方式，以及 MEV 如何威胁到区块链底层共识的稳定。除此之外，操纵交易排序的例子层出不穷，如 [50, 154]。

Rollup 等创新交易处理方案可以很好地解决高吞吐量区块链扩容问题。然而，这些方案不仅不能解决 MEV 问题，还将问题推给了 rollup 创建者。智能合约运营商或者为（zk）rollup 提供证明的用户都有权改变交易排序。换句话说，rollup 方案将 MEV 问题变成了 REV 问题，即：rollup 可提取价值

MEV 影响了提交至交易池但暂时还未提交至链上的交易。在网络中可以查看这些交易的相关数据。因此，矿工、验证者和网络中的普通参与者都可以基于这些数据创建交易。另外，矿工和验证者还可以改变自己提交的交易排序，以谋取私利。

自上世纪 90 年代，就有文献提到共识协议中参与方利用自己的权利操纵交易顺序 [71, 189]，但到目前为止还没有令人满意的解决方案 [97]。主要原因是一直以来提出的方案都无法直接集成至公链，因为这些方案都需要将交易内容隐藏起来，直到排序完成才公布。

公允排序服务概览： DON 将为去中心化交易排序提供工具，并按照链上合约创建者制定的规则执行。目的是保障排序的公平性，其中任何一方都无法操纵交易排序以谋取私利。这些工具共同组成了公允排序服务（FSS）。

FSS 包含三大要素。第一是交易监控。在 FSS 中，$\mathcal{O}$ 中的预言机节点既监控 MAINCHAIN 上的交易池，也允许从链下通过特殊通道提交交易。第二是交易排序。$\mathcal{O}$ 中的节点根据链上合约制定的规则为交易排序。第三是交易发布。交易排序后，$\mathcal{O}$ 中的节点共同将交易发送至链上。

FSS 的潜在价值包括：

- 排序公平性：FSS 为开发者提供工具，保障传输至某一链上合约的交易经过公平排序，不会向有特权或精通技术的用户倾斜。排序规则可以定制化。
- 减少或消除信息泄露：FSS 保障了网络参与者无法利用交易池中的信息谋取私利，可以缓解或杜绝抢跑等现象。避免交易信息泄露，保障了交易无法插队，先提交的交易先上链。
- 降低交易成本：有了 FSS，参与者无须通过提高 gas 费将交易排在前面，因此可以大幅降低交易成本。

- 优先排序： FSS 可以自动优先排序某些关键交易。比如，为了避免对预言机发起抢跑攻击 [79]，FSS 可以回溯将预言机报告插入一组交易中。

FSS 的总体目标是使 DeFi 创建者建立完全公平的金融系统，即系统不会因为速度、内部消息或技术能力而偏向任何用户（或矿工）。虽然我们很难清晰定义什么是公平，而且在任何意义上都不存在绝对的公平，但 FSS 希望能够为开发者提供强大的工具，帮助他们实现 DeFi 应用最初的设计目标。

我们注意到，虽然 FSS 的主要目标是为 DON 服务的 MAINCHAIN 提供公允排序服务，但 FSS 所保障的公平性也可以应用于 DON 节点之间运行的（去中心化）协议。因此，FSS 可以被视作是一组 DON 节点提供的服务，不仅能够公平地排序用户发送至 MAINCHAIN 的交易，还能够排序其他 DON 节点之间共享的交易（消息）。本章将主要讨论 MAINCHAIN 上的交易排序。

章节大纲： 5.1 章节会讨论推动 FSS 设计的两个高阶应用场景，即：防止抢跑预言机报告以及防止抢跑用户交易。之后，5.2 章节将讨论关于 FSS 设计的更多细节。5.3 章节将讨论公允排序的一些案例以及实现方式。最后，5.4 和 5.5 章节会讨论网络中对这些规则存在的威胁，以及应对网络洪泛攻击和女巫攻击的方案。

5.1 抢跑问题

为了解释 FSS 的目标和架构，我们需要先讨论两类主要的抢跑攻击以及现有解决方案存在的瓶颈。抢跑是一种交易排序攻击，除此之外 FSS 还可以解决许多其他的相关攻击，比如尾随攻击（back-running）和三明治攻击（sandwiching）（注：抢跑和尾随攻击组合在一起） [236] ，本白皮书不一一赘述。

5.1.1 预言机抢跑攻击

预言机的传统职责是将链下数据传输至链上应用，因此自然会受到抢跑攻击。

通常，人们会使用预言机将各类喂价传输至链上交易平台，比如每隔一段时间（如每个小时），预言机会搜集各类资产的价格数据，并发送至交易合约。这些价格数据交易中存在明显的套利机会：比如，如果最新的预言机报告中某一资产的价格明显上涨，那么攻击者就可以将自己的交易插在预言机报告之前，买入资产，并在预言机报告处理完毕后立刻卖出资产。

减速带和价格回溯（retroactive pricing）： 预言机抢跑攻击最直观的解决方案就是给预言机报告特殊的优先权限，排在其他交易前面。比如，预言机报告 gas 费较高，以激励矿工优先处理。但如果套利空间很大，这样做也无法杜绝抢跑，更无法阻止矿工自己去套利。

因此，一些交易平台设置了更大的“减速带”，比如将用户交易排在一些区块之后再处理，或者在预言机报告到达时回溯调整价格。这些解决方案存在一个问题，那就是它们增加了交易平台运行的复杂度，提高了储存要求和交易成本，并影响了用户体验，因为交易平台需要花很长时间确认交易。

捎带（piggybacking）： 在讨论 FSS 之前，我们先来说一下捎带这个方案。这是解决预言机抢跑攻击简单有效的的方案。然而，这个方案无法解决其他类型的抢跑问题。

简而言之，预言机不会定期将报告发送至链上，而是对报告签名，用户在交易链上资产时将签名的预言机报告附加在交易中。交易平台只需查看报告的有效性，并从中提取相关喂价。

这个简单的方案相比“减速带”机制存在许多优势：（1）交易平台合约无需保存喂价状态，因此可以降低交易成本；（2）预言机报告按需发送至链上，预言机可以提高更新频率（如：提高至每分钟一次），因此最大限度降低抢跑预言机报告带来的套利机会。[9]（3）交易可以立即验证，因为交易价格永远都是最新的。

然而，这个方案并非完美无缺。首先，交易所用户承担了获取最新预言机报告并附在交易中的责任。其次，虽然捎带方案在最大程度上减少了套利行为，但却不得不以牺牲链上合约的活跃度作为代价。事实上，如果预言机报告的时效性一直持续到 n 号区块，那么发送至 $n+1$ 号区块的交易就需要一份新的报告。由于预言机向用户传输报告存在延迟，因此 $n+1$ 号区块的新报告需要在 $n+1$ 号区块挖出之前就发布。假设在 $n-k$ 号区块中发布，那么在 k 个区块期间也会产生短暂的套利机会。

现在我们来说说 FSS 如何解决上述这些问题。

FSS 优先排序预言机报告： FSS 在上文提到的捎带方案基础上进一步拓展，以解决抢跑问题。与此同时，将预言机报告的处理工作交给了去中心化的预言机网络。

预言机节点收集本来发送至链上交易平台的交易，对当前喂价达成共识，并将喂价和收集的交易共同发送至链上合约。理论上，这个方案类似“数据增强型交易批量

[9]只有资产价格的差值大于交易资产的无关费用（如矿工和交易平台的手续费），才会存在套利空间。

操作”（data-augmented transaction batching），预言机保障在交易中永远添加最新的喂价。

FSS 方案在执行时可以几乎不对交易平台用户产生任何影响，也几乎无需对合约逻辑做任何修改，5.2 章节会详细论述。确保最新的预言机报告永远优先于用户交易，这是 FSS 可以采用的其中一种排序规则。5.3 章节会具体阐述 FSS 的排序规则。

5.1.2 用户抢跑攻击

接下来我们谈谈一般应用中的抢跑问题，以及为何上述提到的防御机制无法奏效。请看这样一个例子：攻击者看到某个用户向 P2P 网络发送交易 tx_1，并插入自己的交易 tx_2，让 tx_2 排在在 tx_1 之前（可以通过支付更高的交易费实现）。这类抢跑通常是由机器人执行的，利用了 DeFi 系统中的套利机会 [90] ，对各类去中心化应用的用户产生了不利影响 [101]。在区块链上执行公平的交易排序机制可以解决这个问题。

实际上，有时候甚至不需要看到 tx_1 的内容，而是只要知道 tx_1 存在就足以让攻击者将自己的交易 tx_2 插在 tx_1 前面，并损害用户的利益。比如，攻击者得知用户可能会在固定时间交易某种资产。要防范此类攻击，也需要避免元数据被泄露 [62]。目前有一些应对这个问题的方案，但它们会造成延迟并影响可用性。

从网络排序过渡到用 FSS 完成排序： 由于系统无法保障交易处理顺序与外部事件和信息流的顺序保持一致，因此就会出现抢跑问题。这其实是区块链应用（如交易平台）功能无法充分实现所导致的。理论上，应该确保交易发送到区块链上的顺序与其创建和发送到 P2P 网络的顺序一致。但是由于区块链网络是分布式的，因为无法实现这个机制。FSS 可以为用户建立一个安全机制，公平地为交易排序，弥补分布式区块链网络在这方面的不足。

5.2 公允排序服务详解

图 12 展示了 FSS 的大致架构。为了保障公平性，FSS 必须在交易进入 MAINCHAIN 过程中进行干预。因此可能需要对客户端或 MAINCHAIN 上的智能合约做一些调整，或两者同时做调整。FSS 处理交易可以大致分成如下三个阶段：（1）交易监控；（2）交易排序；以及（3）交易发布。视具体的交易排序方法而定，可能还需要其他协议步骤，下一章会具体描述。

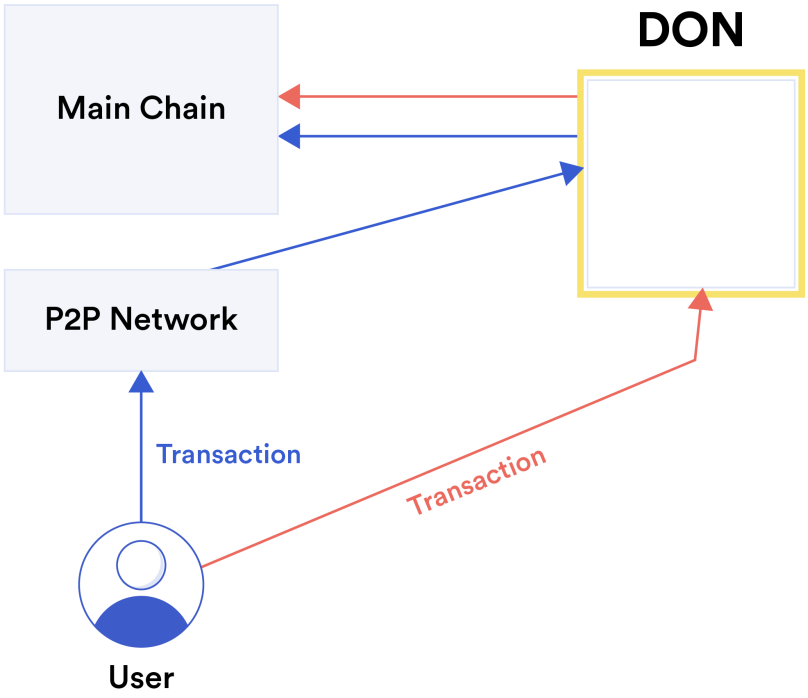

图 12: 公允排序的交易池存在两种不同的交易路径，即：直接发送和通过交易池发送。

5.2.1 交易处理

交易监控： 我们设计了两种方式，让 FSS 监控用户发送至某一智能合约的交易，即：直接发送和通过交易池发送。

- *直接发送*：直接发送是最简单的方式，但是需要更改用户客户端，让交易直接发送至去中心化的预言机网络节点，而不是发送至 MAINCHAIN 上的节点。DON 会收集用户发送至某一智能合约 SC 的交易，并基于特定的排序规则进行排序。然后，DON 将排序过的交易发送至链上智能合约。某些排序机制必须采用直接发送的方式，因为用户将交易发送至 FSS 之前必须对其进行加密保护。
- *通过交易池发送*：如果要将 FSS 集成至原有客户端，DON 可以采用交易池服务（Mempool Services，简称 MS）来监控链上交易池并收集交易。

许多智能合约可能更偏好直接发送方式，我们认为这也是更加实际的一种方案。

接下来我们简单讨论一下现有 DApp 如何做最小的改动，实现这个方案，并同时保障用户体验。我们决定用以太坊和 MetaMask 举例 [6]，因为他们是目前最主流

的方案。但例子中提到的技术也适用于其他区块链和钱包。以太坊近期的改进提案“EIP-3085：钱包添加以太坊链 RPC 的方法” [100] 可以轻松从 MetaMask 和其他浏览器钱包覆盖定制化的以太坊区块链（使用不同的 CHAIN ID，而不是 MAINCHAIN ID，防止重放攻击），实现这一提案后，DApp 只需在前端调用一个方法，就可以直接接入任何 API 与以太坊兼容的 DON。与此同时，“EIP-712：以太坊类型结构化数据散列和签名” [49] 提案则需要更多操作，但目前这个替代方案已广泛部署，DApp 用户可以使用 MetaMask 对指定 DON 交易的结构化数据签名。DApp 可以将签过名的结构化数据发送至 DON。

最后，我们发现也可以采用混合式方案。比如，原有客户端可以继续将交易发送至链上交易池，但预言机报告等关键交易池则直接发送至 DON 节点（提供喂价等预言机报告的节点可能与提供 FSS 的节点是重合或者完全一样的）。

交易排序： FSS 的主要目的是确保基于约定的规则排序用户交易。具体规则可以根据应用需求及其希望避免的不公平排序而定。

由于 DON 的 FSS 可以处理数据并维护本地状态，因此可以基于预言机上的数据制定任意排序规则。

5.3 章节将详细介绍排序规则和实现方式。

交易发布： DON 收集并排序用户交易后（用户直接发送或通过交易池发送），DON 将这些交易发送至链上。因此，DON 与主链的交互仍依赖于链上矿工对交易的排序方式（可能是不公平的）。为了充分实现去中心化交易排序的价值，链上目标智能合约 SC 必须要将 DON 视作“一等公民”。我们在这里提出两种方式：

- *合约只接受 DON 的交易*：最简单的方法是让链上智能合约 SC 只接受 DON 处理过的交易。这可以保障智能合约按照 DON 的顺序处理交易，但同时限制了智能合约只能采用委员会制度运行（即：DON 委员会有权利决定交易顺序）。
- *合约接受来自两个渠道的交易*：更理想的方案是让链上智能合约 SC 接受 DON 和传统用户两个渠道的交易，[10] 但对直接来自客户端的交易设置“减速带”。比如，来自 DON 的交易会立即处理，而普通交易则需要等待一段时间。另外，还可以对普通交易应用其他标准化机制，以避免抢跑，比如先提交后揭示（commit-

[10] 如果 DON 监控交易池中的交易，那么原有交易必须与 DON 交易区分开，以防出现重复。比如可以在交易中嵌入特殊标签，或标注具体的 gas 价格，如：DON 交易的 gas 价格低于某一数值。

reveal）或 VDF [53] 。这样做可以保障来自 DON 的交易按照约定的顺序上链，同时不给 DON 过多的权利审查交易。

FSS 为交易排序需要在链下聚合交易，因此这个方案结合了一些聚合技术，目的是降低链上成本。比如，DON 在收集和排序交易后，会将这些交易打包（如 rollup）发送至链上，因此降低了交易成本。

执行交易排序： 无论是采用上述哪种方案，链上智能合约 SC 和 DON 都必须经过共同设计，保障按照 DON 的交易排序执行。在这里，我们设计了几种不同的方案：

- 序列号： DON 可以对每笔交易添加一个序列号，并将这些交易发送至链上交易池。链上智能合约 SC 会无视不符合顺序的交易。我们注意到，区块链上的矿工也可以无视 DON 的交易排序，因此导致交易失败。维持（高成本的）状态，让 SC 执行正确的交易排序，这有点类似 TCP 协议在收到缺失的数据包之前会一直无视顺序混乱的数据包。

- 交易 *nonce* 值：对于许多区块链——特别是以太坊——来说，上文中的序列号方案可以利用内置交易 nonce 值保障链上智能合约 SC 按照顺序处理交易。DON 节点向通过统一的账号向链上发送交易，这个账号的密钥在 DON 节点之间共享。账号的交易 nonce 值保障交易按照正确的顺序被挖出并处理。

- 聚合交易： DON 可以在一个 rollup（或与 rollup 相似的打包）中聚合多笔交易。链上智能合约要能够处理此类聚合交易

- 使用链上代理的聚合交易：DON 将交易打包成一个 “元交易” 发送至链上，但需要一个定制化的代理智能合约解开交易并传输至链上智能合约 SC.。这个技术可以用来保障与遗留应用兼容。元交易功能类似 rollup，但不同点在于元交易中的交易未压缩过，并一次发送至链上。

最后一个方案的优势是可以无缝支持通过代理链上合约发送的用户交易，交易先经过链上代理再传输至 DON 所对应的链上智能合约 SC。比如，一名用户将交易发送至某个钱包合约，然后再发送一笔内部交易至 SC。在交易中添加序列号会比较麻烦，除非用户钱包合约专门设置成将序列号和每笔内部交易都发送至 SC。同样地，这些内部交易无法轻松聚合成原交易，并直接发送至 SC。下文中我们会具体讨论此类代理交易的问题。

5.2.2 原子化交易

目前为止，我们的讨论都建立在一个假设之上，那就是与单一链上智能合约展开交易（如：用户向一个交易平台发送买入交易。）然而，在以太坊等系统中，一笔交易可能包含多个内部交易，如：一个智能合约调用另一个合约的函数。下面，我们会讨论为“多合约”交易排序的两种高阶方案，并同时保障交易的原子性（即：交易要么全部按照正确的顺序执行，要么全部不执行）。.

强原子性： 最简单的解决方案就是将上文中的 FSS 直接应用于“多合约”交易。用户将交易发送至网络，FSS 监控、排序并在链上发布交易。

这个方案在技术上比较简单，但是存在一个潜在问题：如果用户交易与两个合约 SC_1 和 SC_2 交互，且这两个合约都想接入公允排序服务，那么这两个合约必须采用一致的排序规则。也就是说，如果交易 tx_1 和交易 tx_2 要与 SC_1 和 SC_2 交互，且 SC_1 将 tx_1 排在 tx_2 前面，那么 SC_2 就绝对不能将 tx_2 排在 tx_1 的前面。

在大多数情况下，不同合约的排序规则都是一致的。比如，SC_1 和 SC_2 都基于交易到达交易池的大致时间进行排序，SC_1 可能还希望总是将某些预言机报告排在前面。由于 SC_1 的预言机报告交易不会与 SC_2 交互，因此不影响规则的一致性。

弱原子性： 完整的 FSS 可以应用于单笔内部交易。

比如 $\mathsf{tx} = \{\tilde{\mathsf{tx}}_{\mathrm{pre}}, \tilde{\mathsf{tx}}_{\mathsf{SC}}, \tilde{\mathsf{tx}}_{\mathrm{post}}\}$ 形式的交易，其中包含初始交易 $\tilde{\mathsf{tx}}_{\mathrm{pre}}$，$\tilde{\mathsf{tx}}_{\mathrm{pre}}$ 调用内部交易 $\tilde{\mathsf{tx}}_{\mathsf{SC}}$ 发送至 SC，然后发布内部交易 t˜xpost。SC 的排序规则规定了内部交易 t˜xSC 相对其他发送至 SC 交易的排序方式，但没有规定 $\tilde{\mathsf{tx}}_{\mathrm{pre}}$ 和 $\tilde{\mathsf{tx}}_{\mathrm{post}}$ 的排序方式。

由于以太坊等系统独特的交易处理机制，针对具体内部交易开发排序机制并非直截了当的工作。可以使用特殊设计的合约 SC 按照以下方式实现：

1. 交易 tx 发送至网络并被挖出（不经过 FSS 排序）。初始交易 $\tilde{\mathsf{tx}}_{\mathrm{pre}}$ 执行，并调用 $\tilde{\mathsf{tx}}_{\mathsf{SC}}$。
2. $\tilde{\mathsf{tx}}_{\mathsf{SC}}$ 并返回。
3. FSS 监控发送至 SC 的内部交易，为交易排序并重新发布至 SC（即将交易 $\tilde{\mathsf{tx}}_{\mathsf{SC}}$ 直接发送至 SC）。
4. SC 处理来自 FSS 的交易 $\tilde{\mathsf{tx}}_{\mathsf{SC}}$，并基于此发布内部交易 $\tilde{\mathsf{tx}}_{\mathrm{post}}$ 。

这个方法中，交易并非完全原子化的执行（原本的交易 tx 被分解成多笔链上交易），但实现了内部交易排序。

这个方案存在一些设计上的限制。比如，$\tilde{\mathsf{tx}}_{\text{pre}}$ 无法假设 $\tilde{\mathsf{tx}}_{\mathsf{SC}}$ 和 $\tilde{\mathsf{tx}}_{\text{post}}$ 会被执行。另外，即使交易 $\tilde{\mathsf{tx}}_{\mathsf{SC}}$ 和 $\tilde{\mathsf{tx}}_{\text{post}}$ 来自 FSS，SC 也必须能够代表用户执行交易。因此，粗放式的“强原子性”解决方案更具有实操性。

为了遵守多个交易及其各自内部交易复杂的依赖关系，FSS 交易排序机制必须建立精密复杂的功能，类似关系型数据库的事务管理器。

5.3 公允交易排序

这里，我们要讨论两个公允交易排序的概念，以及 FSS 针对每个概念的实现方式，即：基于 FSS 排序规则的排序公平性，以及安全的因果一致性（secure causality preservation），这需要在 FSS 中实现加密技术。

排序公平性： 排序公平性指在共识协议中实现暂时的公平性（temporal fairness），此概念首次被 Kelkar 等人正式提出 [144]。

希望建立一种自然的规则，基于交易到达 DON（或 P2P 网络）的时间为交易排序。然而，在去中心化系统中，不同节点看到交易达到的顺序可能是不一样的。为所有交易建立排序是 MAINCHAIN 底层共识协议解决的关键问题。因此，Kelkar 等人 [144] 引入了一个弱化版的概念，可以通过去中心化的预言机网络实现“区块排序公平性”。DON 在一段时间间隔内收到的交易被打包成一个“区块”，并将区块内所有交易以同样的高度一起发送至 MAINCHAIN。交易被排在一起，并行执行，不存在任何冲突。简而言之，交易公平性的原则规定如果大部分节点先看到交易 τ_1 再看到交易 τ_2，则 τ_1 将被排在 τ_2 前面，或放在同一个区块。这种粗放式的交易排序机制大幅降低了抢跑或其他排序相关攻击的出现频率。

Kelkar 等人提出了名为 Aequitas [144] 的一组协议，覆盖了各种不同的部署模式，其中包括同步、部分同步以及异步网络设置。相比基础 BFT 共识来说，Aequitas 协议的通讯成本极高，因此没有太大的实用价值。然而，我们相信可以开发出更加实用的 Aequitas 协议，在 FSS 和其他应用中为交易排序。目前已经提出了一些在形式和属性上都较弱的方案，如 [36, 151, 235]，但这些方案都具有较强的实用性，而且都可以兼容 FSS。

另外值得一提的是，“公平性”这个词出现在其他区块链相关文献中的意思是不一样的，多指矿工所获得的机会与其投入的资源成正比 [106, 180] ，或验证者拥有平等的机会 [153]。

安全的因果一致性： 分布式平台最常使用加密技术来防止抢跑等违规行为。各类加密技术有一个共同点，那就是隐藏交易数据，等到共识层完成排序后再披露交易数据。这保存了区块链上交易之间的因果顺序。早在区块链出现之前，相关的安全概念和加密协议就已经被提出 [71, 189]。

“输入因果性”（input causality）[189] 以及“安全的因果一致性”（secure causality preservation）[71, 97] 规定在交易顺序确定之前不得公布交易数据。交易数据受到加密保护，攻击者在那之前都无权得知。

有四种加密技术可以保障因果一致性：

- *先提交后揭示协议 [29, 142, 145]*：只广播交易相关的加密文字（commitment），不公布交易内容。等到除隐藏交易以外的其他所有交易都被排序后（早期区块链系统是 MAINCHAIN 自己排序，这里是 FSS 排序），提交者必须在约定的时间段内披露交易数据。然后，网络会验证内容是否与之前的加密文字相匹配。这个方法在区块链发明之前就存在了。

  虽然这个方法非常简单，但是却存在很多问题，很难实现，原因有两个。首先，由于加密文字只存在于排序协议层面，交易内容无法在共识中得到验证。因此还需要再往返一次客户端。不过更严重的情况是在提交之后拒绝披露数据，这可能会导致 DoS 攻击。另外，很难以统一且分布式的方式判断披露的数据是否有效，因为所有参与者必须就是否在约定时间内披露数据达成一致意见。

- *具有恢复延迟（delayed recovery）功能的先提交后揭示协议 [145]*：先提交后揭示方案存在一个问题，那就是客户端可以试探性地提交一笔交易，只有当有利可图的时候才披露交易内容。近期，这个方案有了改进，可以有效抵御这种风险。TEX 协议 [145] 采用了精妙的方式解决这一问题。其中加密交易中包含一个密钥，可以通过计算可验证延迟函数（VDF）获得密钥 [53, 220]。如果客户端无法及时解密交易，系统中的其他人可以通过解一道中等难度的加密算法题来为他解密交易。

- *门限加密 [71, 189]*：这个方法需要 DON 开展门限加密计算。假设 FSS 拥有一个加密公钥 $\mathsf{pk}_{\mathcal{O}}$，预言机节点之间共享对应的私钥。客户端可以用 $\mathsf{pk}_{\mathcal{O}}$ 加密交易，然后发送至 FSS。FSS 在 DON 中为交易排序，并解密，最后将交易以固定顺序发送至 MAINCHAIN。因此，通过将交易加密，我们可以保障交易排序不基于交易内容，但如有需要也可以查看交易内容。

  这个方案最早由 Reiter 和 Birman [189] 提出，之后由 Cachin 等人进一步完

善 [71]，并与许可制共识协议结合。近期，研究者尝试将门限加密作为共识层机制，实现通用消息 [33, 97] 以及使用共享数据开展通用计算 [41]。

与先提交后揭示协议相比，门限加密能够防止简单的 DoS 攻击（不过还是要注意解密的计算成本）。门限加密技术能够让 DON 自动运行，保持原有速度，无需等待客户端的进一步操作。交易一被解密就可以立即得到验证。另外，客户端还使用同一把密钥为 DON 的所有交易加密，通讯模式也与其他交易一模一样。不过，安全管理门限密钥以及应对 $\mathcal{O}$ 中节点变更可能会带来一些挑战。

- *提交的加密共享 [97]*：客户端不用 DON 的密钥为交易数据加密，而是在 $\mathcal{O}$ 中的节点间加密共享交易数据。建立安全的混合型加密共享机制，先用随机密钥的对称加密算法对交易进行加密。共享相应的对称密钥，秘文发送至 DON。客户端必须在另一个加密消息中向 $\mathcal{O}$ 中每个节点发送一部分密钥。剩下的协议步骤与门限加密相同，唯一不同的是先把每笔交易的密钥重构出来，再使用对称算法解密。

  这个方法不需要设置或管理与 DON 关联的公钥加密系统。然而，客户端必须能看到 O 中的节点，并通过秘文与每个节点通讯，这增加了客户端的负担。

虽然加密技术可以充分保护交易信息不被泄露，但却无法隐藏元数据。比如，攻击者仍可以利用发送者的 IP 地址或以太坊地址发起抢跑等攻击。需要在网络层（如 [52, 95, 107]）或交易层（如 [13, 65]）部署各类隐私保护技术，以解决这一问题。

可以通过在同一个 DON 中复用多个智能合约，将某些元数据（比如交易发送到的合约）部分隐藏起来。通过加密技术隐藏交易内容本身无法防止 DON 节点与发送者共谋操纵交易排序。

加密协议保障了安全的因果性，可以增强排序公平性，我们希望尽可能地将两种方式有效结合。如果攻击者无法从元数据获得任何攻击优势，那么安全的因果一致性协议也可以采取简单的排序机制。比如，预言机节点一收到交易就可以立刻写入 $\mathcal{L}$，消除重复劳动。可以基于交易到达$\mathcal{L}$ 的顺序排序，并进行解密。

我们还计划使用 TEE 来实现公平排序，比如 Tesseract [44] 可以视作是一种因果排序机制，由于有 TEE 的加持，可以直接处理交易并同时保护隐私。

5.4 网络层注意事项

到目前为止，FSS 主要关注的是确保最终交易顺序符合网络中观察到的顺序。之后，我们将讨论网络层本身存在的公平性问题。

传统在线交易平台的高频交易者会投入大量资源获得超高的网速 [64]，同样地，加密货币交易平台上的交易者也会采取类似的行为 [90]。网速快意味着可以观察到其他交易者的交易，并且能更快提交自己的交易。Flash Boys [155] 一书中提到了一种很常见的应对措施，就是设置“减速带”，这个概念首先由 IEX 交易所 [128] 提出，之后在其他交易所逐渐实施 [178]（结果好坏不一 [19]）这个机制会降低交易者访问市场的速度（IEX 上延迟是 350 微秒），目的是消除速度优势。 [128] 等实践经验证明它确实可以为普通投资者降低交易成本。在这里可以用 FSS 实现非对称减速带，即只延迟进入平台的交易。

Budish、Cramton 和 Shim [64] 表示，在连续时间的市场中是无法完全杜绝利用速度优势获利的行为，并且提议建立批量拍卖制市场（batch-auction-based markets）。但是这个方法并没有在交易平台中普及起来。

传统的交易平台是中心化的，通常通过单一网络连接获取交易。而在去中心化的系统，可以从多个角度观察交易传播（transaction propagation）。也就是说，可以在 P2P 网络中观察网络洪泛攻击等行为。我们希望尝试通过网络层的方式实现 FSS，帮助开发者制定规则，防止网络攻击行为。

5.5 实体级公平政策

排序公平性和安全因果性的目的是基于交易创建和发送至网络的时间进行排序。这个公平概念存在一个问题，那就是无法抵御洪泛攻击。比如攻击者对系统发起大量交易的洪泛攻击，这个方法通常用来对通证销售进行有效的交易阻击 [159] ，制造拥堵，导致抵押债仓清算 [48]。换句话说，排序公平性保障的是交易公平性，而不是参与者公平性。

节点委员会（比如 DON）可以采用 DECO 或 Town Crier 等工具，以各种方式抵御女巫攻击，并同时保障隐私，比如 CanDID 系统 [160]。用户可以注册身份并证明其独特身份，同时无需披露身份信息。身份证明可以抵御女巫攻击，可以降低洪泛攻击发生的概率，并丰富交易排序规则。比如，通证销售中每名注册用户只允许进行一次交易，交易需要提供国民身份证明，比如社保卡号。这个方式虽然并非万无一失，但却可以在很大程度上防止交易洪泛攻击。

6 DON 交易执行框架（DON-TEF）

DON 将通过去中心化预言机网络交易执行框架（DON-TEF，或简称 TEF）为 layer-2 解决方案提供预言机以及去中心化的资源。

如今，区块链延迟问题限制了 DeFi 合约的更新频率，比如以太坊平均每 10-15 秒产生一个区块 [104]。除此之外，将大量数据发送至链上需要耗费巨大成本，而且交易吞吐量较低，因此催生出了各种扩容方案，比如分片 [148, 158, 231] 和 layer-2 执行 [5, 12, 121, 141, 168, 185, 186]。即使是交易速度提高了许多的区块链（如 [120],）也提出了链下计算等扩容方案 [167]。TEF 的作用是为所有 layer-1/MAINCHAIN 提供 layer-2 资源。

DON 采用 TEF，可以更快更新 MAINCHAIN 合约，并同时保留主链关键的可信度。TEF 可以支持任意 layer-2 执行技术和范式，包括 rollup[11]、optimistic rollups 以及 Validium 等。另外还有供 DON 节点执行的门限签名模式。

TEF 与 FSS 互为补充。换句话说，TEF 中的任何应用都可以使用 FSS。

6.1 TEF 概览

TEF 为高性能混合型智能合约 SC 的构建和执行提供了设计框架。

TEF 基于混合型智能合约的核心原则，将 SC 分成两个部分：（1）在 TEF 中称为 MAINCHAIN 上的主合约 SC_a；（2）DON 逻辑 exec_t ，也称为 TEF 可执行程序。我们在此用 SC 表示 SC_a 和 exec_t 共同实现的逻辑合约。（正如上文所述，我们希望开发编译工具，将合约 SC 自动编译成这些功能）

TEF 可执行程序 exec_t 是一个引擎，处理 SC 中的用户交易。它在 DON 中运行，因此可以高效执行。TEF 可执行程序拥有以下功能：

- 获取交易：exec_t 收到或获取用户交易。可以直接获取，即直接向 DON 提交交易，也可以使用交易池服务（MS）从 MAINCHAIN 交易池获取交易。
- 快速执行交易：exec_t 处理 SC 中的资产交易。并且在本地，即在 DON 中执行
- 快速且低成本地访问预言机/适配器： exec_t 可以在链下访问预言机报告以及其他适配器数据，因此可以获得相比 MAINCHAIN 更快速、低成本且准确的资产喂价。另外，在链下接入预言机，无需在链上付出高昂成本储存数据，因此可以降低预言机的运行成本以及系统使用成本。

[11]通常被成为“zk-rollup”，但是这个词不恰当，因为不一定需要零知识证明。

- 同步：exec_t 会定时从 DON 同步到 MAINCHAIN，更新 SC_a。

主合约是 SC 在 MAINCHAIN 的前端应用。它作为 SC 可信度更高的组件，拥有以下几个作用：

- 资产托管：用户在 SC_a 中存取资金。
- 同步验证： SC_a 可以在 exec_t 同步时验证状态更新是否正确，如：在 rollup 中附加 SNARKs
- 安全护栏： SC_a 还可以防止 exec_t 被攻击或出现故障。（第 7 章会详细讨论）

这里要指出的是，用户资金还是在链上托管，也就是说 DON 本身不托管用户资金。用户只用 DON 来获取准确的预言机报告并且及时与 MAINCHAIN 同步，具体视同步方案而定（参见下方）。因此，信任模型很接近订单簿制度的去中心化交易平台（如 [2]），目前这类交易平台通常在链下撮合交易，并在链上进行清算和结算。

如果是用支付系统做类比的话，那么可以将 exec_t 理解成负责清算的 SC，而 SC_a 负责结算。图 13 为 TEF 示意图。

TEF 优势： TEF 拥有三个关键优势：

- 高性能：DON 在处理交易和预言机报告方面的能力远远高于 MAINCHAIN。另外，exec_t 可以更快速地处理交易，并更及时地响应预言机报告。
- 交易费更低：相比交易处理，同步流程对时间的要求没有那么高，而且 DON 可以将交易批量发送到 MAINCHAIN 。因此每笔交易的链上费用（如 gas 费）将大幅降低。
- 隐私性：DON 可以大幅提升 SC 的隐私。

TEF 的瓶颈： TEF 的瓶颈在于它无法立即提取资金，因为只能在 MAINCHAIN 上提取资金：用户对 SC_a 发送提取资金请求后，需要等待 exec_t 执行一次状态更新。6.2 章将讨论一些补救措施。

TEF 的另一个瓶颈是无法在链上 DeFi 合约中实现原子化交易，也就是说它无法在一笔交易中将资产发送至多个 DeFi 合约。然而，如果 DeFi 合约在同一个 DON 中运行，TEF 就可以实现原子化交易。6.2 章节中会讨论一些解决方案。

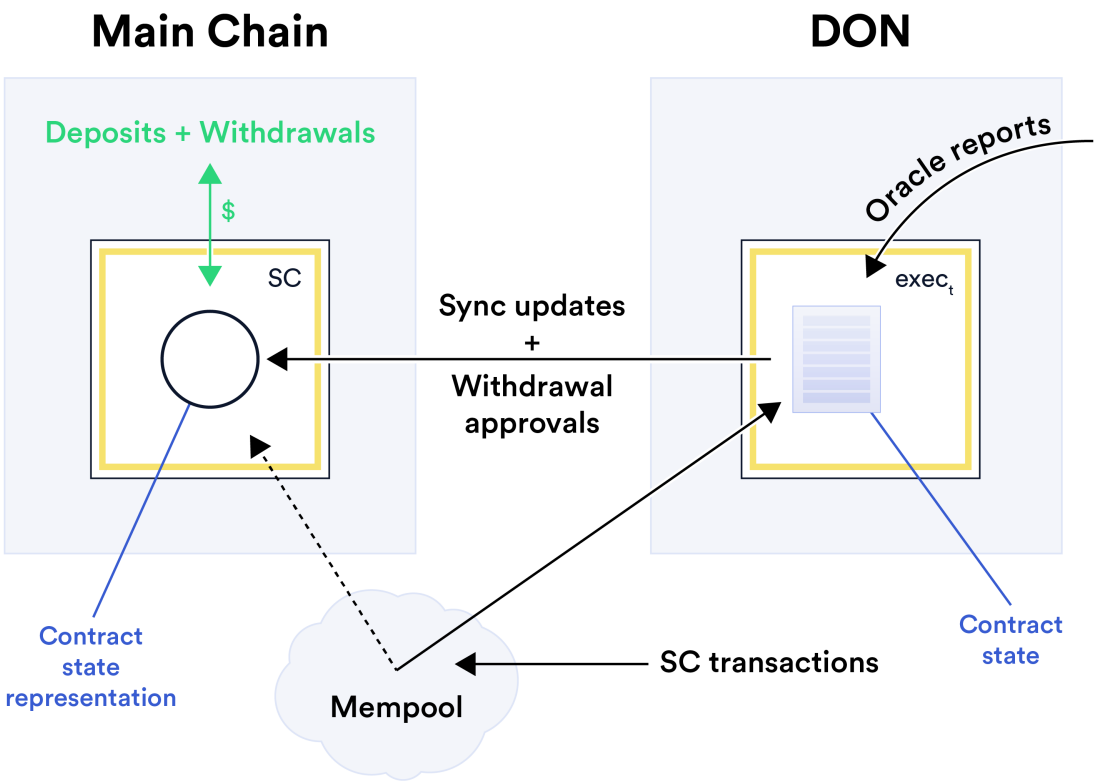

图 13: TEF 示意图。在这个例子中，交易通过 MS（Mempool Service）从 MAINCHAIN 交易池输入至 DON。

6.2 交易路由

用户可以直接将交易发送到 DON，也可以先发送到 MAINCHAIN 上的交易池再通过 FSS 发送至 DON。有四种交易类型，每种类型都需要不同的处理方式：

合约内交易： 由于绕开了复杂的 gas 费，因此 TEF 可以让 SC 在处理交易时获得比 layer-1 合约更高的灵活性。比如，以太坊上交易池中的交易只要出更高的 gas 费就可以插队，而 SC 的交易只要一到达交易池就会严格按照顺序执行。因此，SC 无需等待交易在区块中被确定，从而大幅降低了延迟。

代理： 用户希望通过钱包合约或 MAINCHAIN 上的其他合约向 SC 发送交易 τ。DON 可以模拟 MAINCHAIN 上的 τ 执行，确定是否将交易传输至 SC。如果是，则将 τ 与 SC 的其他交易进行排序。DON 有几种方式可以甄别此类交易：（1）DON 可以模拟交易池中的所有交易（这个方法成本较高）；（2）某些合约或合约类型（如钱包）可

以放在 DON 的监控名单上；（3）用户可以为 DON 标注交易。

如果一笔交易要与两个合约 SC_1 和 SC_2 同时交互，而两个合约都使用公允排序服务并且排序规则存在冲突，那么问题就更加复杂了。比如 DON 可能会以两个合约都兼容的最近时间排序 τ。

存入资金： 交易将 MAINCHAIN 资产存入 SC，需要先在区块中确认，SC 才能认定它是有效的。而如果 exec_t 监测到一笔交易将资产（如以太币）发送至 SC_a，可以立刻确认存款。比如，可以用 DON 的当前预言机报告价格计算。

提取资金： 如上所述，TEF 的瓶颈是无法立即提取资金。在 rollup 模式中，提取资金请求必须与其他交易放在一起排序，以保障安全性。然而，有方法可以解决这个问题。

如果 DON 可以快速快速计算出 rollup 中直到资金提取交易的有效性证明，那么 exec_t 观察到交易池中的用户交易 τ，就可以以更高的 gas 费发送状态更新交易 τ'，这是一种有益的抢跑。假设 τ 没有在 τ' 到达交易池前被挖出，那么 τ' 会排在 τ 前面，τ 会影响提取资金请求被通过。

TEF 使用 DON 计算状态更新（参见下文的门限签名），DON 可以根据 SC 执行时的状态判断 τ 是否应该通过。然后，DON 可以发送交易 τ' 通过提取资金请求 τ，不影响完整的状态更新。

如果这个方法不可行，或不成功，DON 可以发起交易 τ' 按照交易 τ 将资金发送给用户，这样用户就不必发起其他交易。

6.3 同步

TEF 可执行程序 exec_t 会定期将 DON 的更新推送至 MAINCHAIN，更新 SC_a 状态，我们将这个流程称为同步。可以将同步理解成将 layer-2 交易传输至 layer-1，TEF 可以利用任何技术实现这个功能，包括 rollup[5, 12, 16, 69]、optimistic rollups [5, 12, 16, 69] 以及 Validium [200] 或基本的门限签名技术（如 threshold BLS、Schnorr 或 ECDSA [24, 54, 116, 201]。理论上，可信执行环境也可以验证状态改变，这为 rollup 提供了更好的替代方案，但这个方案需要依赖可信硬件才能实现。（参见 [80]。）

下面我们将这些同步方案与 TEF 的三个关键特质进行了比较：

- *数据可用性*：SC 状态储存在哪里？TEF 至少有三种方案：储存在 MAINCHAIN、DON 或 IPFS 等第三方存储服务提供商。这三种方案在安全保障、可用性以

及性能方面都各有不同。简而言之，将状态储存在 MAINCHAIN 上可以实现链上审核，并无须依赖任何一方查看状态；另一方面，将状态存在链下可以降低存储成本并提升吞吐量，但必须相信存储服务提供方（DON 或第三方）可以保障数据可用性。当然，也可以灵活结合这几种方案。表 1 展示了要求的可用性形式。

- 准确性保障： SC_a 如何保障 exec_t 推送的更新是准确的？这影响了 exec_t 和 SC_a 的运算负载，以及同步延迟（见下文）。
- 延迟：同步延迟有三个原因：（1）exec_t 生成同步交易 τ_{sync} 需要时间；（2）τ_{sync} 在 MAINCHAIN 上确认需要时间；（3）τ_{sync} 对 SC_a 生效需要时间。在 TEF 中，延迟对于资金提取尤其重要（但是对合约内交易来说没那么重要），因为资金提取必须要同步一次状态（至少是部分同步）。

同步方案	**数据可用性**	**状态准确性**	**延迟**
Rollup [5, 12, 16, 69]	链上	有效性证明	生成有效性证明所需的时间（如，当前系统是分钟级）
Validium [200]	链下	有效性证明	同上
Optimistic rollup [10, 11, 141]	链上	欺诈证明	挑战期长度（几天或几周）
门限签名 [24, 54, 116, 201]	灵活	DON 门限签名	即时
可信执行环境 [80]	灵活	基于可信硬件生成证明	即时

表 1: TEF 的各种同步方案及其特性

表 1 总结了 TEF 五种主要的同步方案。（这里需要指出，我们不将这些技术作为单独的 layer-2 扩容方案进行比较。关于这个内容请参考 [121]。）

接下来我们来详细讨论每种同步方案。

Rollup： Rollup [69] 是一种协议，在链下批量计算交易状态变更。然后将变更的状态发送至 MAINCHAIN 。

在 rollup 中，主智能合约 SC_a 需用 R_{state}（如默克尔树的根哈希值）表示实际状态的压缩值，并储存 Rstate。要实现同步，exec_t 需向 SC_a 发送 $\tau_{\text{sync}} = (T, R'_{\text{state}})$，其中 T 是自上次同步以来处理的一组交易，R'_{state} 代表新计算出来的状态的压缩值，将 T 中交易应用于之前的状态 R_{state}。

有两种比较主流的 rollup 方式，其中 SC_a 验证 τ_{sync} 状态更新的方法有所不同。首先是 **(zk-)rollup**，为 $R_{\text{state}} \to R'_{\text{state}}$ 的状态变更附加一个简洁的证明，有时又叫“有效性证明”（validity proof）。具体实现方法为：exec_t 计算并提交有效性证明（如 zk-SNARK 证明），并在 τ_{sync} 中提交，主合约只有在验证这份证明后才会接受状态更新。

Optimistic rollup 不需要附加证明，但设置了质押和挑战机制，以验证状态变更的有效性。具体实现方式是：SC_a 总是先假设 τ_{sync} 是正确的（因此称为“乐观的”rollup），但 τ_{sync} 需要通过挑战期才会生效，在挑战期任何 MAINCHAIN 上的观察者都可以发现错误状态更新并通知 SC_a 采取必要行动（如：回滚状态或对 exec_t 罚款）。

两种 rollup 方式都可以实现链上数据可用性，因为交易发布在链上，并可以构建出全部状态。zk-rollup 的延迟主要是因为需要花时间生成有效性证明，通常会延迟数分钟时间 [16]，之后可能会得到完善。而 Optimistic rollup 延迟则更严重（通常延迟几天或几周），因为挑战期需要足够长，才能甄别并防止欺诈行为。确认时间长带来的影响通常不能一眼看出，有时候不同机制的影响不一样，因此无法进行全面分析。比如，某些机制认为支付交易是状态更新确认前“无需信任的最终结果” [109]，因为用户对 rollup 的验证速度比 MAINCHAIN 要快很多。

Validium: Validium 是一类 (zk-)rollup，只在链下储存数据，不会将所有数据都发送至 MAINCHAIN 。具体而言，exec_t 只向 SC_a 发送新状态和证明，而不发送交易。通过 Validium 实现同步，完整状态只储存在 exec_t 和执行 exec_t 的 DON 上，并且交易也只在这上面执行。与 zk-rollup 相同的是，同步延迟主要是因为生成有效性证明需要一定时间。然而与 zk-rollup 不同的是，Validium 同步可以降低存储成本并提升吞吐量。

DON 门限签名： 假设超过门限阈值的 DON 节点是诚实的，一个简单快捷的同步方案就是让 DON 节点共同对新状态进行签名。这个方法可以同时在链上和链下实现数据可用性。这里要指出，如果用户信任 DON 的预言机报告，就完全可以信

任 DON 的状态更新，因为已经建立了门限信任模型。门限签名另一个好处就是延迟较低。可以支持 EIP-2938 [70] 中提出的新的交易签名格式，并且账户抽象（account abstraction）可以极大提升门限签名的可行性，因为无需采用 ECDSA 门限及其复杂的协议（如 [116, 117, 118]），而是采用更简单的 Schnorr [201] 或 BLS [55] 门限签名。

可信执行环境（TEE）： TEE 是隔离的执行环境（通常通过硬件实现），目标是为内部运行的程序提供强大的安全保障。一些 TEE（如英特尔 SGX [84]）可以提供证明（attestation），证明某一程序基于具体的数据计算出了准确的结果[12]。TEF 结合 TEE，可以用 TEE 证明替代 (zk-)rollup 或 Validium 的证明，使用 [80] 的技术。

与 rollup 和 Validium 的零知识证明相比，TEE 性能更高。与门限签名相比，TEE 降低了生成 ECDSA 门限签名的难度，因为理论上只需要一个 TEE。然而，采用 TEE 需要依赖硬件的安全性。TEE 只有与门限签名结合使用，才能有效抵御 TEE 的安全漏洞，不过这层保护机制又重新引入了 ECDSA 门限签名生成的复杂性。

更高的灵活性： 这些同步方案将不断完善，提高灵活性，具体方式如下。

- *灵活触发同步*：TEF 应用可以设置同步触发条件。比如，可以基于批次同步，如：每 N 个交易同步一次；或基于时间同步，如：每 10 个区块同步一次；或基于事件同步，如：每当资产价格波动幅度超过阈值就同步一次。
- *部分同步*：exec_t 可以快速同步少量状态，只有定期会完全同步。在某些情况下在是更好的方案（如：rollup 采用部分同步可以降低延迟）。比如，exec_t 可以在 SC_a 中更新用户余额，通过提取资金请求，无需再更新 MAINCHAIN 状态。

6.4 区块重组

网络不稳定或甚至 51% 攻击都会导致区块链重组，而这会对 MAINCHAIN 带来巨大的安全威胁。在现实中，攻击者会利用区块重组发起双花攻击 [34]。此类攻击虽然难度较大，但仍然具有一定可行性 [88]。

由于 DON 在链下独立运行，因此可以作为旁观者，为区块链提供一些保护，抵御区块重组相关的攻击。

比如，DON 可以向 MAINCHAIN 上的 SC 报告区块链分叉，最小长度为 τ。DON 可以提供额外的证明——PoW 或 PoS 都可以，以证明区块链分叉的存在。SC 将采

[12]附件 B.2.1 中有详细内容，但并不需要理解。

取相应的防御措施，比如在一段时间内暂停交易执行（如允许交易所将双花资产放入黑名单）。这里要指出，虽然发起 51% 攻击的攻击者可以审查来自 DON 的报告，但 SC 也可以采取应对措施，那就是在处理交易时要求 DON 定期提供报告，或验证高值交易时要求提供最新报告。

虽然 DON 可以出于各种目的提供区块链分叉报警服务，但我们计划将这个服务放在 TEF 中。

7 信任最小化

Chainlink 网络是一个去中心化系统，其中有许多不同的参与者，因此可以在网络活跃度（即可用性）和安全性（即报告准确性）方面提供强大的安全保障。然而大多数去中心化系统中节点的去中心化程度都各不相同。即使在规模庞大的系统中，矿工 [32] 和中介 [51] 的去中心化水平都比较有限。

而去中心化最核心的目的是实现信任最小化：我们希望降低 Chainlink 网络中出现系统性攻击或故障的风险，其中包括 DON 发起的恶意攻击。我们采取了最小特权原则（Principle of Least Privilege）[196]。系统中参与者的特权得到严格限定，只需让他们能够成功完成任务即可。

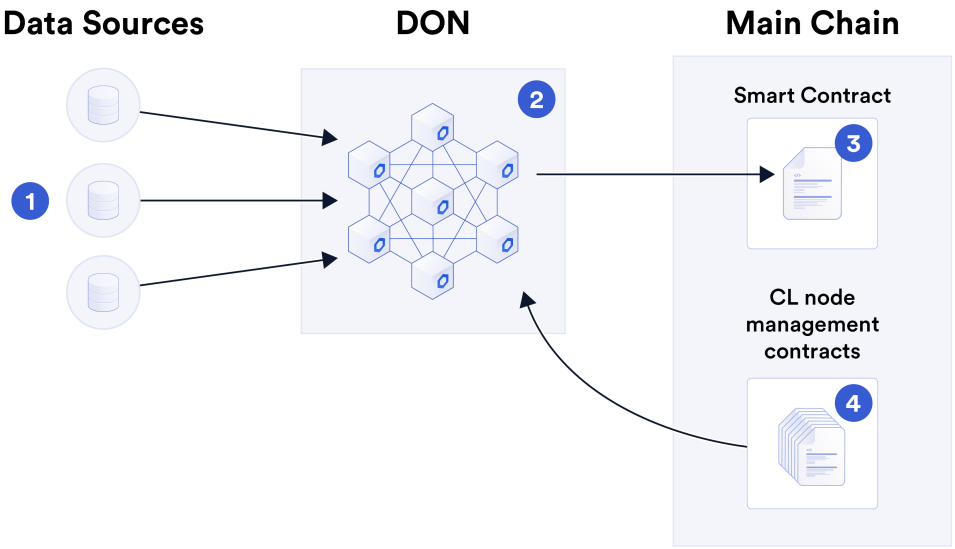

图 14: 本章讨论的信任最小化所在位置。① 数据源向 ② DON 提供数据，然后 DON 将数据的计算结果传输至相应 ③ 智能合约。另外，DON 或预言机网络中还包含 ④ MAINCHAIN 上的节点管理智能合约，向节点付款、提供安全护栏等。

我们列举了几个推动 Chainlink 进一步实现信任最小化的方案。我们按照其在系统中的位置分类，请参照图 14。接下来我们会逐一阐述。

7.1 数据源认证

目前预言机的运营模式存在一个限制，即几乎没有数据源会对其数据进行加密签名，一大部分原因是 TLS 无法实现数据源签名。TLS 在握手协议中采用了数字签名，在服务器和客户端之间建立了共享密钥。因此，HTTPS 服务器拥有公钥认证，理论上可以对数据进行签名，但通常它们不会这么做。由于上述原因，如今的去中心化预言机网络的安全性完全依赖于预言机节点是否能诚实地将数据从数据源传输至智能合约。

Chainlink 实现信任最小化的长期方案是开发数据签名工具和标准，实现数据源认证。数据签名可以在端到端保障数据完整性。理论上，如果合约收到直接由数据源签名的数据 D，那么预言机网络就无法篡改 D。目前已经推出了许多数据源签名方案，比如 OpenID Connect，这个方案主要实现的是用户认证功能 [9]；TLS-N，这是一个学术项目，通过修改 TLS 认证扩展 TLS 功能 [190] 以及 TLS Evidence Extensions [63].。OpenID Connect 目前已经实现了部分应用，但 TLS Evidence Extensions 和 TLS-N 还没有得到应用。

另一个具有潜力的数据源认证方案是用数据提供商自己经过签名的 HTTP 交换协议（SXG） [229]，将数据缓存在内容传输网络中，作为 Accelerated Mobile Pages（AMP）协议的一部分 [224]。Chrome 移动浏览器展示的内容是经过 AMP 协议缓存的 SXG 签名数据。对于用户来说，数据就相当于直接来自数据源，而非 Chrome 的缓存服务器。这对品牌来说是一个很大的激励，再加上使用 Cloud-Flare 的 Real URL [83] 以及 Google 的 amppackager [124] 可以相对比较容易实现这个功能，因此缓存新闻内容将广泛应用 SXG，这将使 Chainlink 预言机节点实现以防篡改的方式基于 SXG 中具有新闻价值的事件触发链上合约。虽然新闻内容提供商的 AMP 缓存 SXG 可能对交易数据等快节奏应用没有太大价值，但是可以为一些基于链下事件的定制化智能合约提供安全的数据源，比如极端天气和选举结果等。

我们相信一旦有了简单的部署方式、成熟的工具以及足够大的灵活性，数据源签名将实现飞速发展。数据提供商将 Chainlink 作为经过认证的 API 前端，这应该是一个很有潜力的方案。我们希望让节点实现这一功能，节点既可以只负责数据源签名，也可以兼具完整的预言机功能。这个功能我们称之为 “数据来源认证”（authenticated data origination，简称 ADO）。Chainlink 节点建立 ADO 功能后，数据源将获得

Chainlink 社区开发的工具和功能，为现有的链下 API 添加数字签名功能。如果数据源想要自己运行节点，还可以多增加一条收入渠道，与 Kraken [28] 和 Kaiko [140] 等数据提供商以同样的方式出售数据，通过运行 Chainlink 节点将 API 数据销售到链上。

7.1.1 数据源认证的瓶颈

数据源签名虽然可以增强数据认证，但本身不足以实现预言机网络在安全和运行方面的所有目标。

首先，数据 D 仍然需要准确及时地从数据源传输到智能合约。也就是说，即使用预先写入智能合约的密钥对所有数据进行了签名，仍需要 DON 可靠地将数据从数据源传输至智能合约。

另外，许多情况下，合约还需要访问基于原始数据展开的各种经过验证的函数运算结果，原因如下：

- 隐私性：数据源 API 可能会提供敏感数据或专有数据，在上链前需要经过修改或隐私处理。然而，对签名过的数据进行的任何修改都会使签名无效。换句话说，在数据源实现 ADO 与数据隐私处理无法兼得。案例 3 阐述了如何通过增强 ADO 功能解决这一矛盾。

- 数据源错误：数据错误或失效都会影响数据源质量，而这两个问题都无法通过数字签名解决。Chainlink 自发布以来 [98] 就建立了应对这类错误的机制，即：冗余备份。预言机网络的报告通常聚合了来自多个数据源的数据。

下文将讨论 ADO 希望建立的机制，以增强数据源的隐私性，并安全地从多个数据源聚合数据。

7.1.2 隐私性

数据源无法预测到用户需要哪些 API，而且用户为了保障隐私性可能需要预先处理过的数据。下面的例子描述了这个问题。

Example 3. *Alice 希望获得一份去中心化身份认证（DID），证明她超过 18 岁（因为超过 18 岁才能贷款）。因此，她需要从 DID 发行方那里获得一份年龄证明。*

Alice 希望能从美国车辆管理机构（DMV）网站获得数据。DMV 保存了她的生日记录，并以下方格式发送了一份经过电子签名的证明 A：

$$A = \{\texttt{Name: Alice}, \texttt{DoB: 02/16/1999}\}.$$

在这个案例中，Alice 用证明 A 就可以向 DID 发行方证明她超过 18 岁。但是这个过程中并不需要透露隐私信息，即：Alice 具体的出生年月日。在理想情景中，Alice 希望从 DMV 获得一份附上签名的声明 A' 证明 “Alice 超过 18 岁。” 换句话说，她要计算关于她生日的函数 G，其中，如果 $\mathsf{CurrentDate} - X \geq 18$，则 $A' = G(X) = \texttt{True}$，否则 $G(X) = \texttt{False}$。

在此基础上进一步拓展，Alice 可以向数据源请求一份以下格式的签名证明 A'：

$$A' = \{\texttt{Name: Alice}, \texttt{Func:}\mathsf{G(X)}, \texttt{Result: True}\},$$

其中 G(X) 是函数 G 的表达式，输入参数 X。用户可以传入 G(X)，获得相应的证明 A'。

这里要注意，数据源的证明 A' 必须包含 G(X) 表达式，确保 A' 得到正确阐释。在上述案例中，G(X) 定义了 A' 中的布尔值，因此 True 表示证明中的主体年龄超过 18 岁。

我们采用了灵活的查询方式，用户可以定义一个 G(X) 函数作为查询条件。为了实现案例 3 中的应用场景并支持直接来自智能合约的查询，我们希望能实现包含简单函数 G 的函数查询，以实现 ADO 功能。

7.1.3 聚合数据源数据

为了降低链上成本，合约通常会接收来自多个数据源的聚合数据，下方案例将具体阐释。

Example 4 (价格数据取中位数)**.** 预言机网络要传输喂价（注：喂价指一种资产与另一种资产的兑换汇率，比如以太币对美元），通常需要从多个交易平台等数据源获得当前价格数据。网络通常会向链上智能合约 SC 发送这些价格数据的中位数。

在有数据签名功能的系统中，正常运行的预言机网络从数据源 $\mathcal{S} = \{S_1, \ldots, S_{n_S}\}$ 中获取一系列数值 $V = \{v_1, v_2, \ldots, v_{n_S}\}$，数据来自 n_S 个数据源，并附有数据源的签名 $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_{n_S}\}$。验证完签名后，将喂价 $v = \mathsf{median}(V)$ 发送至 SC。

可惜目前预言机没有简单的方法将案例 4 中的中位数 v 以及简单的证明 σ^* 一起发送至 SC。

另一种方案是将所有 n_S 个数据源的公钥写入 SC。然后，预言机网络将 (V, Σ) 发送至链上，让 SC 计算 V 的中位数。然而，这个方案会生成 $O(n_S)$ 大小的证明 σ^*，在简洁性上大打折扣。另外，这个方案还会为 SC 增加 gas 成本，因为 SC 需要验证 Σ 中的所有签名。

相反，用 SNARK 可以生成简洁的证明，覆盖所有经过验证的价格数据。这个方案也许是可行的，但证明者需要耗费高额计算成本以及链上 gas 费。另外一种方案是采用 Town Crier，但是这需要可信执行环境，因此并不适用于所有信任模型。

在思考数据签名问题时一个有用的概念叫作“函数签名”（functional signatures [59, 132]），这是一种密码学工具。简而言之，签名者可以使用这个工具委派他人签名，受委派者只能对函数 F 范围内的消息进行签名，函数由签名者定义。附件 D 中展示了这个工具如何限定 DON 传输并由数据源签名的数值范围。我们还引入了一个新概念，叫“离散函数签名”，对准确性的要求比较宽松，但却比 SNARK 等方案的性能高许多。

数据聚合商从多个数据源聚合数据时也面临同样的数据签名问题，比如 CoinCap、CoinMarketCap、CoinGecko 以及 CryptoCompare 等。这些数据聚合商从多个交易所聚合数据，基于交易量进行加权计算，有些时候将算法公开，有些时候则隐藏。数据聚合商与节点一样，都面临数据源认证的问题。

7.1.4 数据源数据处理

高级的智能合约可能会接入定制化的聚合数据而非数据源，比如各类资产的近期价格波幅或特定事件的新闻报道和照片。

由于 DON 的计算和带宽成本相对较低，可以用较低成本处理这些数据，甚至是需要大量数据输入的机器学习模型也是如此，因为只需控制最后传输到链上的数据量即可。

如果开展运算密集型任务，DON 节点可能对复杂的数据输入持不同观点，那么就需要在 DON 节点之间展开更多轮通讯，在计算结果之前对数据输入达成共识。只要最终结果完全取决于数据输入，并且所有节点对数据输入达成了共识，只需要分别进行计算并使用部分签名向其他节点广播或发送至数据聚合商即可。

7.2 DON DON 信任最小化

我们设计了两种方案为 DON 中的组件实现信任最小化：客户端故障转移和少数派报告。

7.2.1 客户端故障转移

密码学和分布式系统相关文献中描述的攻击模式通常认定攻击者有能力买通一组节点，许多 BFT 协议中攻击者可以买通不到三分之一的节点。然而，如果所有节点都运行同样的软件，理论上攻击者可以在差不多同一时间危害所有节点，以发起攻击。这个现象通常被称为“软件单一性”（software monoculture [47]）。

人们提出了许多软件以及软件配置多样化方案，如 [47, 113]。然而，正如 [47], 所述，多版本软件是一个复杂的问题，需要谨慎考虑。比如，使用多个软件相比单一软件的安全性更低。因为多元化增加了系统的攻击面，因此这个方案带来的潜在攻击向量大于其安全价值。

我们认为软件在实现多样化的同时必须保障稳健的客户端故障转移，即：节点客户端在遭遇灾难性事件时可以迅速切换成其他版本的客户端。备用客户端不会增加攻击向量，因为它们不是主客户端。然而，备用客户端提供了二层防御机制。我们希望在 DON 中建立客户端故障转移机制，以降低对某一版本客户端的依赖。

Chainlink 已经打造了稳健的备用客户端系统，继续维护之前成熟的客户端版本。比如，目前采用链下报告（OCR）作为主要客户端的 Chainlink 节点在有需要时可以切换回之前的 FluxMonitor 系统。FluxMonitor 被使用了很长时间，并且经过了安全审计和实战考验。它与 OCR 拥有同样的功能，唯一的区别就是它成本更高，不过只有在需要的时候才会发生成本。

7.2.2 少数派报告

如果有足够多的少数派节点 $\mathcal{O}_{minority}$（注：观察到网络中多数节点造假行为的少数诚实节点），那么这些节点就可以生成一份少数派报告。少数派报告是平行报告或标记，将由 $\mathcal{O}_{minority}$ 传输至链上 SC。SC 会根据具体的规则利用这份报告。比如，如果合约中安全性比活跃度更重要，那么少数派报告可能会导致合约从另一个 DON 请求补充报告，或触发熔断（参见下一章）。

即使在大多数节点都诚实的情况下，少数派报告也可以发挥重要价值。因为包括使用了函数签名在内的所有报告聚合机制都必须以门限签名的方式运行，这样做是为了防止预言机或数据受到操纵。换句话说，必须可以基于 $k_S < n_S$ 个预言机的输入生成有效报告，k_S 为门限值。这意味着被收买的 DON 有可能在 n_S 个节点中任意取前 k_S 个数值，从而操纵最终结果。在这种情况下，即使所有数据源都是诚实的，也无法保证结果正确。

比如，假设系统中 $n_S = 10$，$k_S = 7$ ，系统使用函数签名验证以太币对美元喂价

V 的中位数。假设五个数据源报告的喂价为 500 美元，另外五个报告了 1000 美元。那么在最低的 7 个喂价中取中位数，则输出 $v = 500$ 美元，如果在最高的七个喂价中取中位数，则输出 $v = 1000$ 美元。

随着 DON 协议不断完善，所有节点都能看到有哪些数据存在，以及报告中采用了哪些数据。这样一来，节点就可以发现并标记明显的偏向，认同某组预言机报告，并生成少数派报告。

7.3 安全护栏

在我们的信任模型中，DON 认为 MAINCHAIN 是比 DON 更安全且权限更高的系统。（这种信任模型并不总是成立，不过可以随时调整机制，反过来认为 DON 是安全性更高的平台。）

因此，要实现信任最小化，就自然要在智能合约中建立监控和失效安全机制，可以设置在 DON 的 MAINCHAIN 前端，也可以直接设置在链上 SC 中。我们将此类机制称为“安全护栏”（guard rails），下列是部分最重要的机制：

- 熔断机制：SC 可以根据状态更新本身（如连续一组预言机报告都出现大幅偏差）或其他数据输入暂停或停止状态更新。比如，如果预言机报告在一段时间内出现严重偏差，则触发熔断机制。另外，少数派报告也可以触发熔断机制。因此，熔断机制可以防止 DON 输出严重不实的报告。

  熔断机制可以提供一定时间窗口，考虑或采取额外干预措施。其中一种干预措施就是“逃生出口”。

- 逃生出口：当一组托管方、社区内的通证持有者或其他受委托方发现攻击事件，智能合约可以启动名为“逃生出口” [162] 的应急方案。逃生出口启动后 SC 会关闭，或终止等待执行以及之后可能出现的交易。比如 SC 可能将托管资金还给用户 [17]，可能终止合约条款 [17] 或取消待执行或以后的交易 [172]。逃生出口不仅可以部署在接入了 DON 的合约中，还可以部署在任何类型的合约中，但是它的价值在于可以抵御对 DON 发起的攻击。

- 故障转移：如果系统中 SC 依赖 DON 获得关键服务，那么 SC 就可以提供故障转移机制，保障在 DON 出现故障或被操纵的情况下服务正常运转。比如在 TEF（第 6 章）中，主合约 SC_a 可以提供双接口，提取资金等关键交易以及其他普通交易支持链上和链下两个执行接口，并设置适当延迟以避免 DON 交易

抢跑。在数据源签名数据的情景中，如果 DON 无法正常运行，用户也可以将报告直接传输至 SC_a。

针对各种类型的 optimistic rollup 提出的欺诈证明（6.3 章节）与上述机制比较类似且互为补充。这些机制也为链下模块提供了监控和保护。

7.4 信任最小化治理

Chainlink 网络与所有去中心化系统一样，都需要建立治理机制，视情况调节参数、应对突发情况并制定发展路线。

其中一些机制目前放在 MAINCHAIN 上，之后即使是部署了 DON 也将保持下去。其中一个例子就是预言机节点提供商（DON 节点）的支付机制。DON 在 MAINCHAIN 上的前端合约增设安全护栏等其他机制，之后会定期修改。

我们预期将出现两种治理机制：渐进式和紧急式。

渐进式治理模式： Chainlink 生态中许多更新的紧迫性并不高，比如性能提升、功能改善或者（非紧急的）安全升级等。随着 Chainlink 治理社区的规模不断增大，我们预期大多数此类变更都会由具体受其影响的 DON 推进。在中期，我们将采用当前最小特权原则（temporal least privilege ）实现渐进式治理，最终这将成为一个平行机制。简而言之，这个机制的主要目的是渐进式地推动变革，让社区有充足的时间做调整。比如，迁移到新的 MAINCHAIN 合约可能存在阻碍，因此新合约必须在启动前至少 30 天部署。

紧急式治理： MAINCHAIN 合约出现漏洞或者其他影响活跃度和安全的问题时都需要立即干预，以避免灾难性后果。我们希望建立多重签名干预机制，签名者必须分散在组织各个角落，以避免共谋。要保障签名者随时在线并且能够及时访问指挥链授权紧急变更，必须要谨慎规划运行流程并定期审查。这与测试其他网络安全事件响应能力 [134] 一样，都需要应对警戒下降等常见问题 [222]。

DON 可能会实现极高的异构性，因此其治理模式也与许多去中心化系统不同。每个 DON 都有独特的数据源、可执行程序和服务水平协议（如运行时间和用户）。Chainlink 网络的治理机制必须足够灵活，才能满足这些不同的运行目标和参数。我们正在积极探索不同的设计理念，并计划在未来发布更多相关的研究报告。

7.5 公钥基础架构

随着去中心化程度不断上升，将需要为 DON 节点等网络参与者建立稳健的身份识别系统。具体而言，Chainlink 需要建立强大的公钥基础架构（PKI）。PKI 系统能够将密钥与身份信息绑定。如，PKI 为互联网安全连接系统（TLS）奠定了基础：你每次通过 HTTPS 地址（比如 https://www.chainlinklabs.com）登录网站时，浏览器地址栏都会出现一把锁的图标，这个图标意味着网域所有者的公钥已经与其身份绑定，具体方式是通过叫作“证书”（certificate）的数字签名绑定。建立分等级的证书管理机构系统可以保障证书是由真实的网域所有者发布的，常见浏览器都嵌入了最高级别的证书管理机构。

我们希望 Chainlink 采用去中心化的域名服务为 PKI 奠定基础，初步将采用以太坊域名服务（ENS）[22]。ENS 正如名字所暗示的那样，是效仿了 DNS，DNS 将人类可读的域名映射到互联网 IP 地址。只不过 ENS 是将人类可读的以太坊域名映射到区块链地址。由于 ENS 是在以太坊区块链运行，密钥无法被操纵，因此理论上篡改命名空间与篡改底层智能合约或区块链同样困难。（相比较而言，DNS 一直都存在安全漏洞，会出现欺骗（spoofing）和劫持（hijacking）等各类攻击。）

我们在以太坊主网上注册了 data.eth 的 ENS 域名，并计划将 data.eth 作为根域名，为 Chianlink 网络中的所有预言机服务以及其他实体命名。

ENS 域名是多层结构，每个域名下面都包含子域名。ENS 子域名具有同样高的可信度。

data.eth 的主要作用是在链上提供数据目录服务。一般来说，预言机开发者和用户使用链下数据源（如 docs.chain.link 或 data.chain.link 等网站或推特等社交网络）来发布并获取预言机数据地址（如以太币对美元的喂价）。现在有了 data.eth 这样可信的根域名，就可以直接将 eth-usd.data.eth 映射到以太币对美元喂价聚合商的链上智能合约地址。这样一来，任何人都可以安全查看某一喂价对的链上真实数据源（如以太币对美元喂价）。因此，使用 ENS 可以实现两个链下数据源不具备的优势：

- *安全性更高*：所有对域名的更新都以加密方式安全地记录在区块链上，无法被篡改，而相比之下，网站上的文字地址是不具有这样的安全属性的。
- *自动上传至链上*：数据所在智能合约底层地址如果出现任何更新，都会发出通知，直接上传至链上智能合约，并且可以自动将合约更新至新的地址。[13]

[13]链上智能合约可以灵活设置延迟时间，让合约管理员手动检查和干预。

然而，ENS 这样的命名空间不会自动验证域名所有者的合法身份。比如，如果命名空间包含

$$\langle \text{“Acme Oracle Node Co.”}, \texttt{addr} \rangle,$$

那么用户就可以确信 addr 属于域名认领者 Acme Oracle Node Co.。然而，如果命名空间管理不建立其他机制，那么用户就无法得知域名所有者的合法身份是否真的是 Acme Oracle Node Co. 。

我们提出了一个方案来验证域名所有者在现实世界的合法身份，其中包含几个关键要素。目前，Chainlink Labs 在 Chainlink 网络中担任证书管理机构。Chainlink Labs 的这个身份会持续下去，不过我们的 PKI 将在以下两个方面实现进一步去中心化：

- *信任网络模式*：多层级 PKI 的去中心化版本通常称为“信任网络”[14] 。自从上世纪九十年代以来共提出了十四种方案，如 [98]。另外，许多研究人员也发现区块链可以实现这个模式，在全局账本中记录证书 [226]。我们正在探索这个模式的各个方案，以更加去中心化的方式验证 Chainlink 网络中实体的链下身份。

- *与验证数据关联*：如今，大量预言机性能数据都放在链上，因此与节点地址相关联。这些数据证明了节点参与网络的记录，可以丰富 PKI 的身份信息。另外，可以用 DECO 和 Town Crier 开发去中心化身份认证工具 [160]，让节点可以从链下获取身份认证。举个例子，节点运营商可以将证书附在 PKI 身份信息中，证明拥有邓白氏风险预警评分（Dun and Bradstreet rating）。这些验证方式将与权益质押互补，为网络保障安全。预言机节点在真实世界中的声誉也可以扩展至权益质押系统中。（参见 4.3 章节和 9.6.3 章节）

Chainlink PKI 的最后一个要求就是安全地启动系统，即安全地为 Chainlink 网络创建根域名，目前根域名是 data.eth（这与在浏览器中内置顶级域名相似）。换句话说，Chainlink 用户如何判断 data.eth 到底是不是与 Chainlink 项目关联的顶级域名呢？Chainlink 网络采用了多种策略解决这一问题：

- 在 Chain.link 的域名记录中添加一个 TXT 记录 [223]，明确说明 data.eth 是 Chainlink 生态的根域名。（因此 Chainlink 也利用 PKI 让互联网域名验证其 ENS 根域名。）

[14] 由 Phil Zimmermann 为 PGP 提出的一个概念 [237]。

- 从 Chainlink 官网（如 https://docs.chain.link）连接至 data.eth（这是另一种对互联网域名使用 PKI 的方式）
- 在各类文档以及本白皮书中说明 data.eth 是 Chainlink 项目的顶级域名。
- 在 Chainlink 社交媒体平台上公布 data.eth 为顶级域名（如推特和 Chainlink 官方博客）[18]。
- 在 data.eth 注册地址下存放大量 LINK 通证。

8 DON 部署注意事项

虽然 Chainlink 部署并不是我们的核心计划，但实现 DON 的过程中存在几个重要的技术问题，需要单拎出来讨论一下

8.1 推出方案

本白皮书阐述了 Chainlink 的宏大愿景，实现所有这些高级功能需要一路上应对许多挑战。白皮书中提到了其中一些挑战，但在具体实践过程中肯定会出现新的未预料到的挑战。

我们计划在很长一段时间内逐步实现这一愿景。我们预期 DON 在发布初期将只支持由 Chainlink 社区内部团队合作开发的预编译组件。这样做的目的是为之后的全面应用奠定基础，最终 DON 将对所有用户开放，可以任意发布可执行程序。

我们采取这样谨慎的态度，一部分是因为智能合约存在复杂、不可预期且危险的风险，最近的闪电贷攻击就是最好的例子 [127, 188]。同样地，将智能合约、适配器和可执行程序组合在一起需要极其小心谨慎。

在 DON 部署初期，计划只支持预编译的的可执行程序和适配器模板。这将让我们有机会研究这些功能组合在一起的安全性 [46, 169]。另外，这样做还能简化定价：DON 节点可以为每个功能定价，而不是采用通用计价方式，[156] 案例中采用的就是这种计价方式。我们预期 Chainlink 社区会积极开发新的模板，将各种适配器和可执行程序组合在一起，为成百上千个 DON 开发出实用的去中心化服务。

另外，这各方式还可以避免状态膨胀，即 DON 节点需要在工作内存中预留一部分空闲状态。目前公链已经出现了这个问题，因此催生出了“无状态客户端”（参见 [205] 案例）等各种解决方案。在高吞吐量的系统中，这个问题甚至更加严重，因此 DON 可以只部署状态规模优化过的可执行程序。

随着 DON 不断发展成熟，并建立稳健的安全护栏（第 7 章）、加密经济和建立在声誉之上的安全机制（第 9 章）等各种功能，将为 DON 用户带来更大的安全保障。另外我们还将开发一套框架和工具，让社区中更多开发者都能发布并使用 DON。这些工具将使众多节点运营商走到一起，组成预言机网络，并以无须许可或自助的方式发布 DON，也就是说节点可以根据自己的意愿进行发布。

8.2 动态 DON 会员制

DON 中的节点可能会随着时间推移而发生改变。在动态会员制下，密钥 $\mathsf{sk}_{\mathcal{L}}$ 有两种管理方式。

第一种是在会员出现变更时更新节点手上的 $\mathsf{sk}_{\mathcal{L}}$，同时保留 $\mathsf{pk}_{\mathcal{L}}$ 不变。这种方式（ [41, 161, 197] 具体讨论）的优点是不需要更新 $\mathsf{pk}_{\mathcal{L}}$ 。

经典的密钥重新共享（share resharing）技术 [122] 可以简单高效地实现密钥更新。密钥可以在一组节点 O(1) 与另一组也许是互相连接的节点 O(2) 之间互相传输。其中，每个节点 $\mathcal{O}^{(1)}$ 都通过 $(k^{(2)}, n^{(2)})$ 对 $\mathcal{O}^{(2)}$ 的所有节点进行密钥共享，其中 $n^{(2)} = |\mathcal{O}^{(2)}|$，（也许是新的）门限设为 $k^{(2)}$。有多种可验证的密钥共享机制 [108] 可以抵御攻击者对节点以及底层协议发起攻击。 [161] 中提到的技术就可以实现这一目的，并同时简化消息传输流程，防止密码学难度假设出现漏洞。

第二种方法是更新账本密钥 $\mathsf{pk}_{\mathcal{L}}$。这个方法的优势是可以实现前向保密性：$\mathsf{pk}_{\mathcal{L}}$ 旧密钥共享失效不会影响当前密钥。然而更新 $\mathsf{pk}_{\mathcal{L}}$ 存在两个问题：（1）密钥更新过程中需要将 pkL 下加密的数据需重新加密；（2）更新的密钥需要传输至对应各方。

我们希望能够同时探索两种方案，并且将其有机组合在一起。

8.3 DON 可问责性

DON 与目前的 Chainlink 预言机网络一样，也将建立问责制，即：记录、监控并约束节点的行为。DON 的数据存储能力将大幅超越现存的许多公链，特别是因为 DON 可以接入外部去中心化存储平台。因此，DON 中可以非常具体地记录节点历史服务数据，并建立更精细化的问责机制。比如，在链下计算资产价格时输入的数据可能在最终结果上链之后就会删除。而在 DON 中，这些中间过程数据都会被保留。因此，DON 可以针对每个节点建立精细化的补救或惩罚机制。

另外，7.3 章节还详细讨论了建立安全护栏的方案，应对系统性失效对合约的具体影响。然而，还需要注意到一点，那就是 DON 自己也要建立失效安全机制，即抵御系统性和灾难性的 DON 故障，尤其是分叉和服务水平协议失效。

分叉： 如果 DON 存在足够多的问题节点，那么就会分叉，在$\mathcal{L}$ 中产生两条不同的区块链。然而，由于 DON 会对$\mathcal{L}$ 的内容附加电子签名，因此可以利用主链 MAINCHAIN 防止或惩罚分叉行为。

DON 可以定期从 MAINCHAIN 的审计合约查看状态。如果未来状态偏离了查看到的状态，用户/审核者可以向审计合约出示证明。证明可以生成报警，或没收 DON 节点合约中的保障金作为惩罚。后面一种方式介绍的激励机制与针对具体预言机的机制类似，并且可以利用第 9 章提到的技术。

增强服务水平协议： 虽然 DON 并不一定会永远运行下去，但也必须遵守与用户签订的服务水平协议（SLA）。主链上可以执行基本的 SLA 协议。比如，DON 节点可以在规定时间内负责维护 DON，或终止服务前提前发出通知（如：三个月前发出通知）。MAINCHAIN 上的合约可以执行基本的加密经济 SLA 协议。

比如，如果在规定时间内没有更新报告，则没收 DON 的保证金。用户可以质押保证金，质疑 DON 是否正确反应了有效区块的顺序（与 [141] 中案例类似）。当然，区块生成与交易处理不一样，但是 SLA 协议也可以用来约束交易执行。比如，兼容旧客户端版本的 FSS 可以从交易池获取交易（参见 5.2 章节），交易最终被挖出并传输至链上。用户可以向 SLA 合约证明 DON 没有在规定时间内将挖出的交易发送至链上合约。[15]

还可以证明更具体的违反 SLA 规定的行为，并予以处罚，其中包括可执行程序计算错误（如通过 6.3 章节中提到的机制证明正确的链下状态交易）或未能通过 DON 中的启动器运行可执行程序，或未能及时将 DON 中的数据传递至 MAINCHAIN ，等等。

9 经济制度和加密经济激励

Chainlink 网络建立安全的去中心化信任模型，就必须保障节点都能诚实守信。也就是说节点必须大部分时间都严格遵守 DON 协议。本章将讨论如何通过经济激励机制（也称“加密经济激励机制”）实现这一目标。这些激励机制分为两类：显性机制和隐性机制，分别通过权益质押以及未来收益机会（future fee opportunity, 简称 FFO）实现。

[15]由于用户可以替换交易池中的交易，因此需要小心区分被挖出的交易和 DON 提交的交易。

权益质押： Chainlink 中的权益质押与其他区块链系统一样，都包含网络参与者（即预言机节点），参与者质押一定数量的 LINK 通证。这些资金也被称为权益质押，或在显性激励机制中被称为显性质押。如果节点失效或违规，质押的保证金金将被没收，在区块链中这通常称为“slashing”。

然而，Chainlink 预言机节点的权益质押与公链上的验证者存在根本差异。验证者可能出现分歧或操纵交易排序。然而，公链的底层共识协议采用了不容置疑的区块验证规则和密码学机制，以防止验证者生成无效区块。相比之下，无法通过代码逻辑防止预言机网络生成无效预言机报告。原因在于这两种系统存在一个关键差异，即：区块链上的交易验证本质上是保障链上一致性，而预言机报告则是验证链下数据。

我们为 Chainlink 网络建立了初步的权益质押机制，预言机节点之间采用交互式的协议接入链下数据。这个机制中建立了经济奖惩机制鼓励诚实的预言机行为。这个经济机制的目的是避免攻击者通过经济贿赂收买节点。（攻击者范围非常广泛，还包括节点共谋。）

Chainlink 质押机制有一些非常强大和创新的功能。[16]其中一个主要功能就是超线性权益质押影响（具体而言，影响呈二次方）。攻击者的攻击成本必须远远超出节点质押的保证金，才能够有效攻击。相比其他类似机制，我们的质押机制能够更好地抵御攻击，防止攻击者通过贿赂收买节点。另外，我们还要讨论 DECO 等 Chainlink 工具如何提升我们的质押机制，对问题节点进行正确裁决。

未来费用收入机会（FFO）： 无论是采取 PoW 还是 PoS 的各种公链如今都主要依靠我们所说的“隐性激励”。隐性激励是一种经济激励，指诚实节点得到的激励不是显性的经济奖励，而是平台参与。比如，比特币矿工社区为了维持市场对比特币的信心、比特币的价格以及他们对挖矿基础设施的资本投入，不会轻易发起 51% 攻击 [150]。

Chainlink 网络也建立了类似的隐性激励机制，我们称之为未来收益机会（FFO）。服务质量高、声誉好的节点才能吸引到更多用户。预言机节点如果传表现不佳，有可能葬送未来的收益机会，因此节点获得的惩罚是损失未来参与网络赚取收益的机会成本，与显性质押相比，FFO 可以看做是一种隐性质押，网络中的节点共同努力维持平台声誉和服务质量，从而获取收益。这个激励机制虽然是 Chainlink 网络协议的底层逻辑，但并不是一个成文的机制。上文中提到的维持比特币挖矿价值也可以视作是一种隐性质押。

[16]这里阐述的质押机制目前只有一个目的，那就是保证预言机网络输出正确的报告。我们期望之后可以扩展其功能，以保障许多其他 DON 功能的正常执行。

这里要强调的是，目前 Chainlink 网络中已经建立了 FFO。Chainlink 在之后的工作中将以实证为基础，严谨地评估隐性质押框架（IIF）中 FFO 等隐性激励机制的作用和价值。为了定量估算节点的未来收益机会，IIF 将持续参考 Chainlink 网络中各种服务和支付数据。估算结果将为 IIF 质押系统建立参数，比当前的启发式或静态模式更准确地反映节点激励。

综上所述，Chainlink 网络中对节点行为的两种主要的经济激励机制为：

- 保证金质押 } 显性激励机制
- 未来收益机会（*FFO*） } 隐性激励机制

这两种激励机制形成互补关系。预言机节点可以参与 Chainlink 质押协议，持续获得用户收入，共同努力维护平台声誉和服务质量，并从中受益。因此，这两种激励机制可以增强预言机网络的加密经济安全。另外，这两种激励机制还可以相辅相成，或结合运用。比如，一个新的预言机节点运营商没有任何过往服务历史和收入源，可以质押高额的 LINK 作为保证金，以吸引用户。相反，有一定声誉的预言机节点运营商，在 Chainlink 网络中提供了很长时间的优质服务，就可以向用户收取高额费用。对这类节点来说，FFO 就是一个重要的隐性激励机制。

总的来说，我们希望用有限的预言机网络资源创造出最好的经济激励机制，约束理性节点（即希望收益最大化的节点）的行为。换句话说，我们的目标是最大程度上提高攻击者的攻击成本。我们希望采用基于算法定义的经济安全机制打造质押协议并结合 IIF，尽可能准确地衡量 Chainlink 的激励。链上合约创建者就可以明确地判断出预言机网络是否满足他们的加密经济安全需求。

经济安全良性循环： 本章讨论的权益质押和 FFO 激励机制的影响不仅限于为 DON 提供安全保障，还能够形成 “经济安全良性循环”。随着 DON 的安全性不断升高，超线性质押影响（以及其他规模经济）的运行成本会进一步降低。成本降低会吸引更多用户进入 DON，并使网络整体用户费用上升。费用上升会不断激励网络发展，并反过来推动这个良性循环。

我们认为经济安全的良性循环只是一种规模经济和网络效应，本章还会讨论其他的类型。

章节大纲：权益质押在技术上和概念上都面临重大挑战，我们对此设计了一个拥有创新功能的机制。本章节将重点关注权益质押。

9.1 章节介绍了权益质押机制概览，9.2 至 9.5 章节详细进行了阐述。9.6 章节介绍了 IFF。9.7 章节概述了 Chainlink 网络的激励机制。

9.8 章节讨论了我们提出的权益质押机制能够为预言机网络带来的经济安全良性循环。最后，9.9 章节简要讨论了推动 Chainlink 网络向前发展的其他潜在影响。

9.1 权益质押概览

本白皮书中介绍的权益质押机制是指预言机节点之间的交互式协议，用以解决预言机报告中外部数据出现分歧的情况。权益质押的目的是确保理性预言机节点诚实守信。因此，我们在此构建一个对权益质押协议发起攻击的攻击者，称为“贿赂者”。攻击方式是用经济手段买通预言机节点。攻击者可能答应与被买通的节点分享成功篡改预言机报告后所获得的经济利益。

我们的权益质押机制可以同时实现两个目标：

1. 抵御强大攻击：其他预言机模式并没有覆盖现实中所有类型的攻击，而就我们所知，本白皮书里介绍的机制是第一个直面各种预言机攻击模式并提出解决方案的。我们假设的前提是，节点和攻击者都是经济理性的（而非诚实的）。我们假设存在一个权威事实来源，由于费用太高而无法在平常使用，但一旦出现分歧时可以拿出来用（下文中会进一步讨论）。

2. 实现超线性质押影响：我们的目标是确保由理性参与者组成的预言机网络能够报告真实的链下数据，即使攻击者的攻击成本与预言机网络质押的保证金总和呈超线性关系，也能有效抵御攻击。目前的权益质押系统中，如果 n 个节点中每个节点都质押一笔保证金 $\$d$，攻击者可以通过贿赂收买节点造假，并给每个节点一笔金额大于 $\$d$ 的贿赂，那么攻击者的总成本就大约是 $\$d$。这个攻击门槛已经很高了，因为攻击者持有的流动性资金必须与网络中质押的保证金总和一样多。而我们则要在这个基础上更进一步。我们希望设计出一个质押机制，即使攻击者在拥有 $\$d$ 攻击成本的情况下也无法有效攻击。

   虽然如下文所述，实际上的影响可能没有那么高，但我们初步的设计将要求攻击者必须拥有远超 $\$dn^2/2$ 的攻击成本，即节点数量 n 的二次方。因此，即使是规模较小的节点网络也是几乎无法被有效攻击的。

实现这两个目标需要以创新的方式结合激励机制设计和密码学。

关键理念： 我们的权益质押机制基于“看门狗优先级”概念。

Chainlink 预言机网络生成报告（如资产喂价）并发送至链上智能合约，报告聚合了多个节点提供的数据（如取中位数）。通常来说，服务水平协议（SLA）会具体规定可以接受的数据偏差区间，比如节点报告可以偏离聚合报告多远，或者聚合数据距离真实数值的偏差值最远可以是多少。

Chainlink 权益质押机制规定，在每一轮报告中每个节点都担任监督职责（即 watchdog），如果节点认为聚合报告有误，则发出报警。每一轮报告中，每个节点都被分配一个优先顺序，这个优先顺序决定了节点报警的处理顺序。这个机制的重点是合并奖励，也就是说最高优先级的报警节点会获得全部没收的问题节点保证金作为奖励。

Chainlink 权益质押机制在设计上分为两层，即第一层“默认层”和第二层“后备层”。第一层是预言机网络本身，其中包含 n 个节点。（为了简化描述，我们假设 n 是奇数。）如果多数节点都报告错误数据，则第一层的监督者会有强大的激励上报错误。一旦错误被上报，第一层就会将问题升级至第二层，第二层系统的特点是成本高且在最大程度上实现可靠性，用户可以在网络 SLA 协议中定制化。比如，二层节点可以全都是声誉非常好的优质节点，或者二层节点数量可以远超一层节点。另外，正如 9.4.3 章节所述，DECO 或 Town Crier 可以有效保障二层实现高效全面的裁决。为了简化描述，我们假设二层系统可以获取正确的数据。

虽然也可以直接用二层来生成所有预言机报告，但我们这个机制的优势在于，二层节点的存在可以一直保障安全性，而平时只需要支付一层节点的运行成本，二层节点只有使用时才产生费用。

看门狗优先级通过以下方式实现超线性质押影响：如果一层预言机网络生成错误结果，并且多个监督者（watchdog）都上报了错误，那么优先级最高的监督者会得到问题节点质押的全部保证金（由于问题节点超过半数，因此保证金至少是 $\$dn/2$。因此，一个监督者获得了全部的奖励，攻击者如果要确保监督者不上报错误，至少要付这个金额才能收买它。由于我们的机制保障了每个节点都有机会担任监督职责，如果优先顺序最高的节点被收买，那么就会顺延到下一个节点担任监督者。因此攻击者必须给每个节点至少 $\$dn/2$，才能避免整个网络中所有节点都不上报。网络中总共有 n 个节点，因此攻击者要成功高攻击，必须花费 $\$dn^2/2$，网络攻击成本与节点数量呈二次方关系。

9.2 背景介绍

我们的质押机制参考了博弈论和机制设计（MD）方面的研究成果（文献参考请看 [176]）。博弈论以数学的方式研究了战略互动（strategic interaction）。在本白皮书中，博弈指的就是一种这样的互动模式。在现实世界中，它规定了博弈中“参与者”可以采取的一系列行动。博弈还包含每个参与者可以获得的“报酬”，即根据各个参与者选择的行动给予奖励。也许博弈论中最知名的研究案例就是囚徒困境 [177]。博弈论研究者的目的是理解一场博弈中的平衡点（equilibrium）。每名参与者都采取一种策略，所有策略达成了一个平衡点，参与者按照策略执行可以获得最大的利益，如果偏离都会损害其自身利益。

而机制设计是一种设计激励机制的学科，让互动达到平衡点时能够产生最佳的效益。机制设计与博弈论可以看作是一体两面：博弈论的一个经典问题是：“以目前的激励机制和模型来看，平衡点在哪里？”而机制设计中常问的问题则是：“什么样的激励机制可以使博弈达到最佳平衡点?”机制设计者的目的通常是创建“激励相容”机制，也就是说机制中的参与者（如拍卖会或其他信息征询系统 [227]）会受到激励汇报真实情况（如他们对某一物品的真实估值）。维克里拍卖也许是最知名的激励相容机制，其中参与者提交针对某一物品的秘密报价，出价最高者获得物品所有权，但只需支付第二高的价格。[213] 加密经济属于机制设计领域，利用加密技术在去中心化系统中创建最佳平衡。

贿赂和共谋行为对机制设计带来了巨大挑战。几乎所有机制都无法抵御共谋的打击，共谋的定义是机制中的参与者私下签署协议串通起来 [125, 130]。而贿赂则指外部方在博弈中引入新的激励，这个问题甚至比共谋更为严重；共谋可以被看作是博弈参与者之间的一种特殊的贿赂方式。

区块链系统本质上就是设计金钱激励的博弈（注：金钱指加密货币）。其中一个简单的例子就是 PoW 挖矿：矿工有权利选择用什么样的算力挖出区块。挖矿最终的奖励是挖矿工补贴（正向激励）减去挖矿产生的电力和设备成本（负面激励），这取决于其他活跃的矿工数 [106, 171] 以及交易费。另一个例子是 SchellingCoin [68] 等众筹型预言机（crowdsourced oracles）：预言机可以选择发送哪些报告，而报酬是由预言机机制规规定，如：根据预言机报告离所有报告中位数值的偏差来支付报酬 [26, 68, 119, 184]。区块链游戏也充满了共谋和贿赂的机会；实际上，智能合约甚至可以帮助发起类似攻击 [96, 164]。对众筹型预言机发起的攻击中最知名的就是 P+Epsilon 攻击 [67]。这类攻击发生在与 SchellingCoin 类似的机制中，其中参与者提交布尔值报告（即 true 或 false），如果与大部分参与者的提交结果一样，则获得奖励

p 。在 p+epsilon 攻击中，攻击者承诺：如果用户提交 false 则获得 $\$p+\epsilon$，且必须的前提条件是大部分人都提交 true。结果就出现了一个平衡点，无论其他人怎么做，所有参与者都会受到激励提交 false；因此，贿赂者可以诱导节点提交 false 而实际上不用付一分钱。

然而，对于预言机，特别是非众筹型预言机的其他攻击类型的研究却仅限于非常弱的攻击模式。比如在 PoW 模式中，研究者研究了基于结果的贿赂模式，只有目标消息被成功屏蔽且不出现在区块中，才支付贿赂 [96, 164]。然而，在预言机领域，除了 p+epsilon 攻击外，受到关注的攻击模式非常有限，比如贿赂者根据参与者行动而非结果支付贿赂金额。

下个章节大致描述了即使在强大攻击下也能保持激励平衡的信息征询机制。

9.3 建模假设

本章节探讨了我们如何为系统中参与者的行为和能力建模，特别是第一层预言机节点、第二层（裁决层）节点以及攻击者。

9.3.1 第一层激励模型：理性参与者

多数区块链系统都出于安全考量会假定一定数量的诚实节点。节点如果会牺牲自己的经济利益去遵守协议，那么就被定义为诚实节点。PoW 系统通常要求超过半数的算力是诚实的，而 PoS 系统则要求 2/3 或以上的参与者是诚实的。甚至是 Arbitrum 这样的 layer-2 系统 [141] 都要求至少有一个参与者是诚实的。

我们在为权益质押机制建模时，提出了一个更弱的假设。（这里要澄清一点，更弱的假设意味着具有更强的安全属性，因此是更加优越的。）我们假设攻击者控制了部分（小于半数）的一层预言机节点。我们不假设剩余的节点是诚实节点，而是假设他们是以利益最大化为出发点的理性参与者。这些节点的行为完全取决于经济利益，并会做出对他们产生最大经济利益的选择。比如，如果节点获得的贿赂金额高于诚实传输报告获得的奖励，那么节点将毫不犹豫被收买。

关于攻击节点的注释： 与去中心化系统常见的信任模型一样，我们假设所有节点都是理性的，即他们的目的是最大化净收益，而不是被攻击者控制。然而，我们的主张——特别是超线性或二次方质押影响——是渐近式的，条件是被攻击者控制的节点数量最多为 $(1/2-c)n$，c 为正常数。

9.3.2 第二层仲裁模型：假设是正确的

Chainlink 权益质押机制有一个关键功能，有助于实现针对理性节点的安全性，那就是第二层裁决系统。

在我们提出的权益质押机制中，任何预言机如果对结果有质疑，都可以发出报警。报警会激活高度可信的第二层系统，二层节点会报告正确结果。因此，这个方案的关键点就是正确的裁决，即第二层系统报告正确结果。

Chainlink 权益质押模型假设第二层系统是无法被收买而且可靠性极高的权威事实来源。此类系统很可能又慢成本又高，因此不适合在平常使用。然而，如果价值达到平衡，即第一层系统可以正常运作，那么就不会激活第二层系统。而第二层系统的存在就是一个可靠的“大后方”，保障整个机制的安全。

这个高成本且可靠的裁决层与大多数司法系统中的上诉流程很像。而且这个机制在预言机系统中已经十分常见，如 [119, 184]。9.4.3 章节简要讨论了第二层系统如何实现。

Chainlink 权益质押系统假设第二层系统有能力正确地做出裁决，因此对预言机节点来说具有强大的威慑力。第二层系统最终判定预言机报告为错误，则预言机节点的部分或全部保证金都将被协议没收。因此，预言机节点会因为害怕被罚款而约束自己行为。这个方法类似 optimistic rollup 采用的机制，如 [141, 10]。

9.3.3 攻击模型

Chainlink 权益质押机制可以保证获取到的信息是真实的，并同时能够抵御各类攻击。我们在之前的研究基础上，对显性攻击模型以及 p+epsilon 等子攻击模型进行了扩展。我们希望 Chainlink 权益质押机制可以抵挡所有类型的攻击，并且安全性经过正式考验。

我们假设攻击者拥有一笔固定预算，用 $\$B$ 表示。攻击者可以与网络中每个节点一对一秘密交流，并私下保证给节点一笔钱，让他们对结果作假，比如对预言机报告、节点发送给机制的公开消息（如报警）、其他预言机的报告以及机制最终生成的结果造假。

没有一个机制可以完美抵御拥有无限能力的攻击者。因此我们在模型中认为某些行为是不具可行性或不切实际的。我们假设攻击者无法突破标准加密机制，并且如上文所述，有一笔固定的大额预算 $\$B$。我们进一步假设攻击者不能操控预言机网络中的通讯，特别是无法大幅延迟一层和二层节点之间的通讯。（攻击者是否能观察到通讯内容取决于具体的机制，下文中会具体解释。）

然而，正如上文所述，我们可以非正式地假设攻击者具有以下能力：（1）可以买通节点操纵一部分节点（$1/2 - c$ 个，其中 c 为常数）即可以完全控制这些节点；（2）可以贿赂任何节点，并基于攻击者制定的具体条件付款（如上文所述）。

虽然本白皮书没有针对所有类型的攻击者正式建立模型或完整的分类，但以下是我们的模型包含的部分贿赂类型。为了简化叙述，我们假设预言机提交正确数值为 true 的布尔值报告，最后对这些报告进行聚合计算得出结果，并传输至链上智能合约。贿赂者的目的是让最终得出的结果为错误，即 false。

- *无条件支付贿赂金*：贿赂者向任何报告 false 的预言机支付 $\$b$。
- *按照概率支付贿赂金*：贿赂者基于概率 q 向任何报告 false 的预言机支付 $\$b$。
- *只有当最终结果为 false 时支付贿赂金*：贿赂者只有在最后结果为 false 时才支付给报告 false 的预言机 $\$b$。
- *只有系统中无人报警时才支付贿赂金*：贿赂者只有在系统中没有人提出报警时才向任何报告 false 的预言机支付 $\$b$ 。
- *p+epsilon 贿赂*：只有当超过半数预言机不报告 false 时，贿赂者才会向任何报告 false 的预言机支付 $\$b$。
- *预期性贿赂*：贿赂者预先向被选中担任随机职责并报告 false 的预言机支付贿赂金 $\$b$。在我们提出的权益质押协议中，所有节点都可以担任监督者（watchdog），并且监督者的顺序是随机选定的，因此可以避免预以及期性贿赂。然而，许多 PoW、PoS 和联盟链系统都可能出现预期性攻击，这充分说明了 Chainlink 攻击模型和权益质押协议必须能够抵御这类型的攻击。更多细节请参考附件 E。

9.3.4 加密经济安全水平需要多高？

理性的攻击者只会在收益大于成本的情况下才会发起攻击。因此在我们的攻击模型以及权益质押系统中，$\$B$ 可以被用来衡量攻击者通过操控预言机网络输出错误报告可以从链上智能合约获取的利润。要判断预言机网络的加密经济安全是否足够高，用户可以从这个角度来评估网络。

对现实场景中的有效攻击者来说，我们预期$\$B$ 应该远远小于链上智能合约中的总资产。在大多数情境中，攻击者不可能成功偷走合约中的所有资产。

9.4 权益质押机制：草图

本章节将分享我们目前正在开发的权益质押机制的大致概念和架构。为了简化描述，我们在此描述了一个简化和慢速版（多轮）的协议。不过需要指出的是，这个机制可行性非常高。由于已经建立了一层经济安全保障，即节点的奖惩机制，许多用户可以“乐观”地接受预言机报告。也就是说，这些用户在二层节点出面裁决之前就可能接受了预言机报告。

不愿意乐观接受报告的用户可以等到协议执行完毕，也就是等到第二层裁决完所有上报的分歧后。然而，这将大幅延长报告的确认时间。因此，我们简要描述了如何针对 9.5 章节提到的更复杂的情景完善机制，提升速度（变成单轮）。

这里要提醒各位，我们质押机制中的一层节点为预言机网络本身。

正如上文所述，Chainlink 权益质押机制的主要架构为，每一轮中每个节点都按照一定顺序担任监督者。因此当系统中产生错误结果 $\tilde{r}$ 时，每个节点都有机会上报错误。一层节点的报警会触发二层出面裁决，我们在此假设二层节点会生成正确的报告。提交错误报告的节点会被惩罚，他们的保证金会被没收，并奖励给监督节点。这个架构在预言机系统中非常常见，如 [119, 184]。

而我们的设计中关键的创新点在于，每个节点是按照顺序担任监督者的。也就是说，监督节点按照某一优先顺序轮流报警。我们之前说过，如果节点获得了最高的优先权，并上报了问题，则会收到每个问题节点的保证金$\$d$作为奖励，因此奖金总额超过 $\$dn/2 = \$d \times n/2$，因为最终结果出错意味着问题节点超过半数。因此，攻击者必须对任意节点支付至少这么多钱才能收买它。因此，如果要收买超过半数的节点，攻击者必须向大多数节点支付一大笔贿赂金，金额大于 $\$dn^2/2$。

图 15 展示了监督节点上报问题的流程。

9.4.1 机制详解

我们接下来将详细阐述的防贿赂系统是一个简化版的两层结构。其中大部分篇幅会用来描述第一层网络及其激励机制和向第二层上报的流程。

假设 Chainlink 网络由 n 个预言机节点组成，节点负责定时（如每分钟一次）报告一个布尔值（如比特币的市值是否超过以太币）。权益质押机制下，节点必须质押两份保证金：预言机节点保证金 $\$d$，如果与多数节点出现分歧则被没收；监督节点保证金 $\$d_w$ ，如果上报的问题最终判定为虚假，则被没收。假设节点无法复制其他节点提交的内容，如通过 5.3 章节中提到的先提交后揭示机制实现。每一轮中，节点先提交报告，一旦所有节点都提交（或时间到），节点再披露报告的内容。

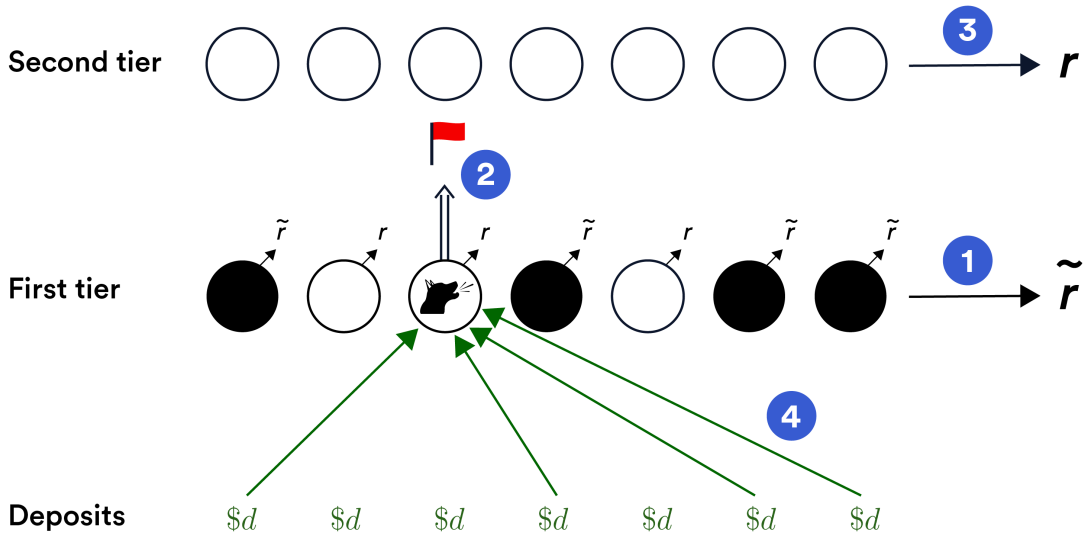

图 15: 权益质押机制和分歧上报示意图。在这个例子中，①超过半数节点被收买，并提交错误报告 $\tilde{r}$，而不是正确报告 r。监督节点 ②向第二层的委员会发出报警，然后二层委员会 ③裁决结果，并提交正确的报告 r。最后，被操控的节点会损失保证金，每个节点向监督节点 ④付 $\$d$。

每个节点会分配到 1 到 n 之间的随机数，以决定监督顺序，数字 1 的优先级最高。最终奖励会全部发给一个监督节点。所有报告都公布后，监督节点就会按顺序报警。n 个节点按顺序轮流担任监督节点，拿到数字 i 的节点有机会在第 i 轮中报警。

我们来设想一下节点揭示报告内容后可能出现的结果。我们在这里再次假设报告的数据是布尔值，并假设正确的值用 true，错误的值用 false 表示。另外，假设一层节点取半数以上节点报告的值作为最终报告 r。

那么就会产生三种可能的结果：

- 所有节点达成共识：最好的情景是，全部节点达成共识：所有节点都可用，而且都及时报告了同样的值 r（用 true 或 false 表示）。这种情况下，网络只需将 r 传输至链上智能合约，并把 $\$p$ 发给每个节点作为每轮固定的报酬，这个金额比 $\$d$ 小很多。

- 部分达成共识：有可能部分节点下线了，或者对正确的结果存在分歧，但是大多数节点都报告了 true，只有少数节点报告 false。这种情况也很简单明了。网络取大多数节点报告的值（true），并生成正确的报告 r。所有报告 r 的节点都获得报酬 $\$p$，而报告错误数据的节点被没收一部分保证金，比如没收 $\$10p$。

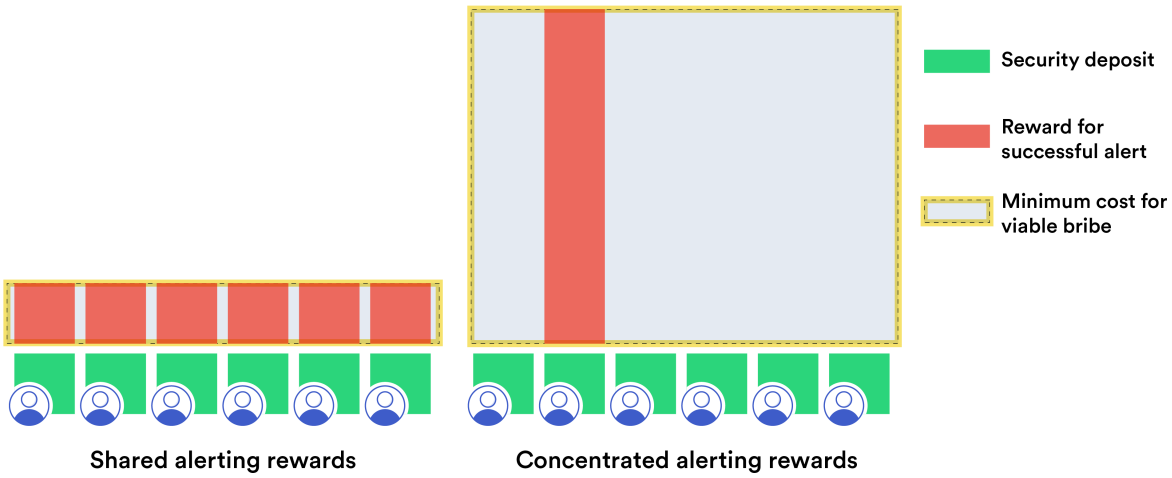

图 16: 通过集中奖励一个节点来提高攻击者的贿赂成本。攻击者必须向网络中每个节点支付高于奖金金额的贿赂（红色部分所示）。如果几个节点分摊奖金，那么攻击者要支付的贿赂金额也会相对较少。集中奖励一个节点可以提高单个节点可能获得的奖金金额（所有红色部分）。因此，集中奖励机制下，攻击者的攻击成本（灰色部分）要远高于分摊奖励机制。

- 监督节点报警：如果监督者认为最终生成的报告有误，可以公开触发报警，上报二层网络。可能会产生两种结果：
  - 有效上报：如果第二层网络确认一层网络的报告有误，那么监督节点将获得所有被没收的保证金作为奖励，即超过 $\$dn/2$。
  - 无效报警：如果第二层认为第一层预言机节点的报告为真实的，则报警的节点被没收 $\$d_w$ 保证金。

  如果一层预言机报告被接受，则监督节点的报警不会对链上合约执行产生任何影响。即使合约需要等待二层节点委员会的仲裁结果才能执行，监督节点报警也只会延迟合约执行，而不会冻结合约执行。在仲裁阶段合约还可以指定一个失效转移（failover）DON。

9.4.2 二次方质押机制影响

由于每个节点都有机会成为监督节点，而且监督节点是按照随机顺序轮流当，并且最后全部奖金都被一个监督节点拿走，因此可以对各类攻击者造成二次方质押影

响（9.3.3 章节详述）。也就是说，我们假设网络中有 n 个节点，每个节点质押 $\$d$ 保证金，任何上述类型的攻击者必须花费超过 $\$dn^2/2$ 的预算才能攻击成功。

简而言之，攻击者必须至少买通 $(n+1)/2$ 个节点，因为他必须买通 n 个节点中超过半数的节点（假设 n 为奇数）。因此，监督节点获得的奖金为 $\$d(n+1)/2$。攻击者必须向每个节点付这么多钱才能确保没有节点上报。如果攻击者的预算最多为$\$d(n^2+n)/2$，则博弈中的完美平衡点（即：博弈中任何时刻的平衡点）就是攻击者不贿赂任何节点，每个节点都诚实地报告真实数据。

上文解释过攻击者的攻击成本需要远高于节点保证金总和才能攻击成功。图 16 展示了集中奖励机制的影响。正如我们所见，如果本来给监督节点的奖金（即所有报告 false 结果的节点保证金总和）在所有节点中平分，那么每个节点分到的奖励都会相对很少，为 $\$d$。攻击者知道节点得到的奖励不会超过 $\$d$ ，因此就可以向每个节点支付比 $\$d+\epsilon$ 高一点的贿赂，前提条件是结果为 false。

图 16 中的机制反向操作，将所有奖金分给一个监督节点，而不是平分给所有节点，因为平分的效果会弱很多。

参数示例： 假设（一层）节点网络中有 $n=100$ 个节点，每个节点质押 $\$d$=20000 美元。这一层网络总共质押了 200 万美元，但攻击者需要花 1 亿美元 =$\$dn^2/2$ 以上的预算才能攻击成功。增加预言机数量比增加保证金 $\$d$ 效果更好，而且会产生巨大的效应：$n=300$ 个节点组成的网络，每个节点质押 $\$d$=20000 美元，攻击者成本必须高于 9 亿美元。

这里要指出，许多情况下，质押机制保护的智能合约价值要远高于攻击成本。这是因为智能合约的攻击者在许多情况下无法榨取合约的全部价值。比如，价值 10 亿美元的 Chainlink 智能合约，只需要防范攻击成本为 1 亿美元的攻击者，因为攻击者只能榨取合约 10% 的价值。

注： 网络价值可以呈二次方增长的概念来自著名的梅特卡夫定律 [166, 234]，其中表明网络的价值与网络中连接数量呈二次方关系。然而，梅特卡夫定律指的是网络中成对的连接数量，这与我们激励机制的二次方质押影响稍有不同。

9.4.3 第二层网络实现方式

第二层网络要保障可靠性需具有以下关键属性：1）第二层裁决应该在预言机网络中极少发生，其运行成本远远超过第一层；（2）假设一层报告被接受，或合约执行需要等待仲裁，第二层无需实时执行。要实现这些属性，第二层需要进行一系列配置，

以满足 DON 的具体需求。比如，二层节点可以由 DON（即一层节点）从 Chainlink 网络服务时间最长且最可靠的节点中选出。二层节点不仅拥有丰富的运营经验，而且还有大量隐性激励（即未来收入机会），因此会受到激励去维持 Chainlink 网络的可靠性。另外，节点的历史服务记录可公开查询。这里需要注意的是，二层节点不需要参与第一层网络，并且可以同时裁决多个第一层网络的分歧。

DON 中的节点可以事先指派 n' 个这样的节点组成二层委员会。另外，DON 节点会发布参数 $k' \leq n'$ 。确定惩罚一层节点的二层节点投票数量。二层节点会针对每个一层节点提交的结果进行投票。如果一层节点收到 k' 个投票确认其造假，其保证金会被没收，并给到监督节点。

由于真正需要裁决并且延长执行时间的概率非常低，因此相比一层节点，二层节点可以：

1. 获得高昂的裁决收入。
2. 接入其他数据源，其中甚至包括一层节点没有接入的数据源。
3. 依赖人工或专家监督干预，如甄别数据源数据错误，并辨别是节点本身造假还是不小心传输了错误数据。

这里要强调，上文中描述的二层节点筛选方案和裁决机制仅为冰山一角，除此之外还有许多其他的实现方式。我们的激励机制可以非常灵活地实现二层节点。因此，每个 DON 可以分别为二层节点设置规则，满足节点和用户的特殊需求。

DECO 和 Town Crier 辅助裁决流程： 在我们的机制中，二层必须要有能力区分一层中故意提交错误报告的问题节点以及不小心传输了错误数据的节点。只有这样，二层节点才能够正确裁决，没收问题节点的保证金，而这也是我们机制的目的。二层节点可以利用 DECO 和 Town Crier 这两个强大的工具来可靠地进行区分。

在某些情况下，二层节点可以直接询问一层节点接入的数据源，或使用 7.1 章节提到的 ADO 查看数据源数据是否有误。然而，有时候二层节点可能无法直接访问一层节点的数据源。这种情况下，就无法做出正确裁决，或者需要依靠主观判断。之前的预言机纠纷机制通过不断升级的多轮投票来解决这个问题，但是这个方案效率低下。

而 Chainlink 提出采用 DECO 或 Town Crier 来帮助一层节点向二层节点证明诚实的行为。（参见 3.6.2 章节了解 DECO 和 Town Crier）具体而言，如果二层节点认为一层节点生成了错误报告 $\tilde{r}$，则一层节点可以使用 DECO 或 Town Crier 生成防篡

改的证明，提交至二层节点，证明自己从 DON 认证的 TLS 数据源正确传输了 $\tilde{r}$。最关键的是，二层节点不用直接访问数据源。[17]因此，Chainlink 可以对任何数据源展开正确裁决。

9.4.4 错误报告保险

我们的权益质押机制可以有效防范贿赂攻击，其底层逻辑是将没收的保证金奖励给举报者。如果没有经济激励，节点就没有直接动力拒绝贿赂。然而，这样会导致受到错误报告牵连的用户没有任何经济补偿，比如报告中错误喂价输入智能合约导致用户资金损失。

理论上，如果报告在二层节点裁决后才被智能合约接受，那么就不会存在问题。然而，正如上文所述，为了优化性能，智能合约会默认生成的报告是正确的，也就是说，它们在二层裁决前就会接受预言机报告。确实，由于我们的模式中假设理性攻击者的攻击预算不会超过权益质押影响，因此就算默认报告是正确的也不会有任何安全风险。

如果用户担心在极端情况下攻击者的预算大到超过我们的权益质押影响，那么可以再添加一层经济安全机制，即错误报告保险。我们知道有几家保险公司计划在近期为接入 Chainlink 的协议推出智能合约保险产品，其中包括通过 DAO 等创新机制，如 [7]。Chainlink 节点的服务激励以及其他关于节点的数据（如质押金额）可以帮助保险公司非常精确地评估风险，并推出价格优惠且有盈利空间的保险产品。

可以使用智能合约高效可信地实现基础版的错误报告保险。举个简单的例子，如果二层节点发现一层节点报告有误，参数型保险 $\mathsf{SC}_{\mathsf{ins}}$ 就会自动向受保人进行赔偿。

如果链上智能合约 SC 的创建者 $\mathcal{U}$ 想要购买保险，可以向去中心化保险公司提交请求，购买金额为$\$M$的保险。请求通过后，保险公司可以在 $\mathsf{SC}_{\mathsf{ins}}$ 中设置连续保费（如每月）为$\$P$。只要 $\mathcal{U}$ 一直缴纳保费，她的保险合约就一直有效。

如果 SC 中预言机报告出现错误，将发送一组对比的 SC 预言机报告 (r_1, r_2)，r_1 由一层节点签名，r_2 是修改过的报告，由二层节点签名。如果 $\mathcal{U}$ 将这组报告 (r_1, r_2) 发送至 $\mathsf{SC}_{\mathsf{ins}}$，并且保险正常续费，则合约自动向她支付 $\$M$。

[17]一层节点采用 Town Crier，还可以在本地生成证明，证明报告的准确性，并按需将证明提交给二层节点。

9.5 一轮方案

上文中提到的协议要求二层委员会须等监督节点在 n 轮内提出报警。即使在一层节点正常运行的乐观情景下，这个规则也同样适用。如果用户不愿意在裁决前直接接受报告，那么等待时间过长将阻碍智能合约应用正常执行。

因此，我们目前也在探索其他只需要一轮的协议。协议中，所有预言机节点都提交秘密字节，表明自己是否要提出报警。二层委员会按照优先次序查看这些结果。以下是这类机制的草图：

1. 监督节点提交消息：每个节点 $\mathcal{O}_i$ 都在二层节点中加密共享监督节点消息 $w_i \in \{\texttt{no\_alert}, \texttt{alert}\}$。

2. 匿名举报：任何预言机节点都可以在同一轮中向二层委员会提交匿名举报 α。α 提示有节点针对当前报告提出报警。

3. 检查监督节点提交的消息：二层委员会按照优先顺序揭示预言机节点提交的单字节消息。

节点如果不提交报警，必须发送 `no_alert`，否则就会揭示所有节点发送的消息。监督节点发送的 `no_alert` 消息排在最高优先级的报警监督节点之前揭示。

这里要指出的是，揭示的内容与多轮协议中的一致。奖金分配机制也与 n 轮协议一致，即第一个报警的监督者获得造假节点被没收的所有保证金。

采用匿名举报机制可以让二层委员会在没有报警的情况下不采取任何行动，这样可以简化沟通。这里还要指出，任何提出报警的监督节点都受到经济激励发出匿名举报：如果没有节点匿名举报，则没人能得到奖金。

为了保障攻击者无法通过网络数据识别匿名举报 α 的发送者 $\mathcal{O}_i$，可以通过匿名通道进行举报，如通过 Tor 或云服务提供商代理。为了验证举报源自于 $\mathcal{O}$，$\mathcal{O}_i$ 可以用环签名来对 α 签名 [39, 191]。另外，为了避免恶意预言机节点对二层委员会发起 DoS 攻击，α 可以是具有可撤销匿名性的匿名证书 [73]。

这个协议虽然具有一定可行性，但在技术上却比较笨重（我们目前正探索解决方案）。比如，一层节点必须直接与二层节点沟通，因此需要维护目录。另外，匿名通道和环签名也增加了技术难度。最后，还需要建立特殊的信任机制，下方注释简要讨论了这点。因此，我们目前还在探索更加简单的机制，同样能实现超线性质押影响，只不过也许影响小于 n 的二次方，但行贿者至少需要投入渐近 $\$n \log n$ 的资源。我们目前正在探索的方案包括随机选择一个节点子集担任监督节点，在这个情景中，预期性贿赂的威力就会非常大。

注释： 要保障单轮质押机制的安全性，需要在一层节点和二层节点之间建立畅通无阻的通道。在投票等机制中这是最基本的要求 [82, 138]，也是合理的要求。

然而，节点 $\mathcal{O}_i$ 如果想与贿赂者合作，可以通过密钥共享，向贿赂者证明它加密了某个值。比如，如果 $\mathcal{O}_i$ 不知道贿赂者控制着哪些节点，就可以向所有委员会成员提交值为 0 的密钥共享。然后贿赂者可以通过概率来验证 $\mathcal{O}_i$ 的合规性。为了在单轮协议中避免这个问题，我们要求 $\mathcal{O}_i$ 必须至少知道二层中一个诚实节点的身份。

在交互式协议中，每个二层节点都对密钥共享添加一个随机因素，贿赂者最多只能让 $\mathcal{O}_i$ 随机选择监督者的消息。

9.6 隐性激励框架（IIF）

Chainlink 网络采用 FFO 作为一种隐性激励机制，激励正确的节点行为。它的功能类似显性质押激励，显性质押激励指通过质押保证金保障网络的经济完全。换句话说，FFO 应该算作是网络中节点质押保证金 $\$d$ 的一部分。那么问题是：我们如何衡量 Chainlink 网络中的 FFO 以及其他类型的隐性激励呢？我们计划为隐性激励框架开发一套规则和技术。区块链系统在许多方面实现了前所未有的透明性，节点的服务记录被完整真实地记录在了区块链上，为 IIF 奠定了坚实基础。下文简要描述了 IIF 的关键要素和概念。

IIF 中包含的一系列因素对评估隐性激励来说非常重要，除此之外还通过各种机制将相关数据可靠地发送至分析算法中。不同 Chainlink 用户使用 IIF 的方式可能不同，比如对不同因素设置不同的权重。我们期待看到社区中开发出各种分析服务，让用户可以根据他们个人的风险评估偏好来应用 IIF。我们的目标是保障这些服务能够可靠及时地接入所需数据，下文会具体讨论（章节）。

9.6.1 未来费用收入机会（FFO）

节点参与 Chainlink 生态的目的是通过提供各种服务来赚取费用收入。服务内容包括普通的数据传输以及去中心化身份、公允排序和保密 DeFi 等高级服务。Chainlink 网络中的费用可以负担节点运营商的成本，成本包括运行服务器、获得所需的数据授权、以及在全球各地组建团队，以确保服务不间断。FFO 代表节点在未来赚得或损失的服务收入减去成本的净收入。FFO 是一种权益质押形式，能够保障网络安全。

其中一个实用的功能是使用链上数据（以及链下数据作为补充）构建出可信度非常高的节点历史记录，并通过实证方式透明地计算出 FFO。

FFO 的一个最简单的实现方式是计算节点在一段时间内的平均净收入（即：总

收入减去运营成本)。FFO 还可以计算成未来累积净收入的净现值 [114] ，换句话说，就是所有未来收益的贴现值。

然而，如图 17 所示，节点收入可能波幅较大。

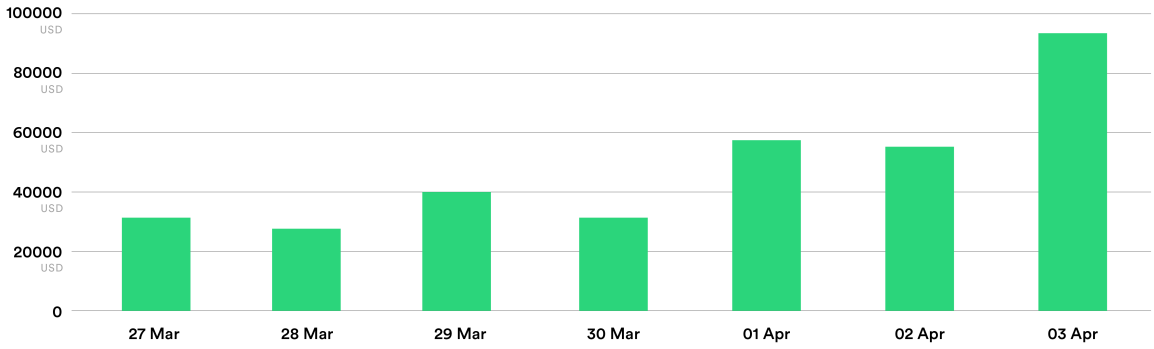

图 17: 2021 年 3 月某一周内，Chainlink 节点针对某一喂价对（如以太币对美元）的收入。

更重要的是，节点收入分布可能随时间而出现变化。因此，我们在估算 FFO 时还计划引入其他考虑因素：

- *节点历史服务水平*：节点运营商的历史服务水平包括报告的准确性和及时性以及节点在线时间，为用户评估节点可靠性提供了客观依据。节点服务记录将作为用户选择预言机节点（或者选择 DON）时的重要考虑依据。节点历史服务水平越高，未来收入就可能越高。[18]

- *数据访问权*：预言机可以接入开放 API 获得许多类型的数据，但是某些类型的数据或某些优质的数据源只能通过订阅或签协议的方式获取。拥有特殊权限访问某些数据源将为节点带来稳定的收入。

- *DON 参与权*：社区中的节点会聚在一起组成 DON，并提供服务。我们预期许多 DON 会自行选择参与的节点，节点被声誉高的 DON 选中，可以提升其市场地位并为其带来持续的收入。

[18]我们想要研究的一个重要问题是如何甄别伪造的服务记录。这有点类似网站的虚假评论，不同在于预言机网络比较容易解决这个问题，因为商品（即预言机报告）的订购和交付数据都储存在不可篡改的区块链账本中，相比之下电商的实物商品信息更难追踪。换句话说，即使客户真实身份无法得到验证，预言机网络的服务水平也可以验证。

- 跨平台产生影响：一些节点运营商可能也活跃于其他领域，并且建立了一定声誉，比如他们可能也是 PoS 区块链上的验证节点或非区块链行业中的数据提供商。他们在其他系统中的历史服务水平也可以作为评估的参考依据（前提条件是这些数据是准确可信的）。同样地，节点在 Chainlink 网络中的造假行为也会影响在其他系统中的声誉和收入，因此 FFO 是可以跨越不同平台产生影响的。

9.6.2 投机性 FFO

节点运营商参与 Chainlink 网络不仅为了赚取服务费，还为了建立品牌和声誉，以便未来在其他网络中获得更多的任务机会。换句话说，预言机节点在 Chainlink 网络的投资可以成为他们未来在其他 DeFi、智能合约应用以及非区块链应用的预言机网络中得到认可的敲门砖。如今，节点运营商在 Chainlink 网络中赚取服务费的同时生成关于声誉、服务历史和运营经验的数据。这些数据将成为他们未来在其他网络中获得任务的敲门砖（当然前提是他们是诚实的）。因此，Chainlink 生态中的现有节点比新节点拥有更大优势，未来 Chainlink 发布更多服务后，可以赚取更多收入。另外，本来就建立了声誉的新进节点运营商和科技公司也具有一定优势。比如，传统科技公司 T-Systems（德国电信子公司）和大型中心化交易平台 Kraken 在刚进入 Chainlink 生态时就建立了一定声誉 [28, 143]。

预言机节点未来的业务机会是一种投机性 FFO，因此是 Chainlink 网络中的一种质押类型。

9.6.3 链下声誉

IIF 可以适用于严格匿名的节点网络中，即不披露节点的真实身份。

然而，用户选择节点时还有一个重要的考量因素，那就是节点在链下的声誉。链下声誉是指与节点链下真实身份挂钩的声誉。将节点的链上声誉与链下真实身份挂钩，可以视为是一种隐性激励机制。我们从 IIF（即加密经济）的视角讨论声誉，将 IIF 作为一种实现跨平台活动的方式，可以作为计算 FFO 时的参考因素。

将节点链下声誉算入 FFO 的好处是，这不仅会影响节点运营商的当前业务，还会影响其未来业务收入。比如，如果节点作恶导致其声名狼藉，那么这也会影响节点未来所有参与的事业。换句话说，链下声誉覆盖的 FFO 范围比匿名记录要大得多，因为某个人或企业如果在链上声誉很差，并且链上声誉关联了链下真实身份，那么就很难摆脱。

Chainlink 可兼容去中心化身份认证技术（章节），可以将 IIF 中的链上声誉关联

链下真实身份。这些技术可以验证并保障节点运营商的链下真实身份。[19]

9.6.4 开放式 IIF 分析

正如上文所述，IIF 的目的是为隐性激励分析机制提供可靠的开源数据和工具。最终目标是让社区中的服务提供商能够针对 Chainlink 用户具体的风险评估需求开发专门的分析工具。

大量关于节点收入和服务水平的数据都储存在不可篡改的可信区块链账本中。然而，我们希望能提供最全面的数据，包括关于链下服务的数据（如：链下报告 OCR 或 DON 的活动）。这类数据的规模是非常庞大的。完整储存这些数据的最好方式就是利用 DON 以及 3.3 章节提到的技术。

一些激励机制可以直接量化，比如权益质押和基础 FFO。而投机性 FFO 和声誉等其他类型的激励机制就比较难客观量化了，但是我们认为其中的支撑性数据，比如 Chainlink 生态的历史增速、社交媒体上的声誉指数等，可以在比较难量化的维度支撑 IIF 分析模型。

我们可以专门建立 DON 来监督、验证并储存节点链下服务的数据以及身份验证信息等其他 IIF 中所需的数据。这些 DON 可以为 Chainlink 社区中任何分析服务提供商提供统一且可信的 IIF 数据。他们还会提供客观权威的记录，社区可以对分析结果进行独立验证。

9.7 综述：节点运营商激励

综合上文中关于节点运营商显性和隐性激励机制的讨论，可以全面地评估节点运营商参与 Chainlink 网络并且盈利的方式。

我们可以将某一 Chainlink 节点运营商质押的资产总额$\$S$用以下公式来抽象地表示：

$$\$S \approx \$D + \$F + \$FS + \$R,$$

其中：

- $\$D$是节点运营商在所有网络中通过显性质押机制质押的保证金总额；
- $\$F$ 是节点运营商在所有网络中 FFO 总额的净现值；

[19]去中心化身份认证也可以使用经过验证的补充信息增强匿名系统。比如，节点运营商理论上可以使用去中心化身份认证证明它是一家财富 500 强公司，除此之外无需披露其他信息。

- $\$FS$ 是节点运营商投机性 FFO 的净现值；
- $\$R$是节点运营商在 Chainlink 生态以外可能受到其不诚实行为影响的声誉价值。

这个公式虽然非常抽象，但我们希望可以通过它来说明 Chainlink 为节点设置了多个经济激励，鼓励节点诚实守信的行为。目前，除了$\$D$以外的所有要素都已经在 Chainlink 网络中实现。

9.8 经济安全良性循环

在 IIF 中结合超线性权益质押影响与未来收入机会（FFO），可以在预言机网络中实现所谓的“经济安全良性循环”。这可以视作是一种规模经济。随着某一网络中保障的价值不断升高，网络中平均每名用户质押的权益就会下降。因此，用户加入现有网络并实现网络经济安全所付出的成本要低于创建新网络的成本。值得一提的是，网络中每新增一名新用户，所有用户分摊的服务成本就会相应降低。

如果网络中费用收入总额增长，会吸引更多质押保证金涌入，并实现更强大的安全保障。具体而言，如果系统中单一节点可以质押的保证金存在上限，那么当新用户进入系统时，节点的 FFO 就会升高，并吸引更多节点加入。由于我们在激励机制中建立了超线性权益质押，因此系统的加密经济安全水平将如章节所述，以 n 的二次方增长。因此，加密经济安全平均成本，即所需的质押保证金总额，将下降。用户使用网络的成本也将下降。假设对预言机服务的需求是弹性的（如 [31] 中的简单示例），那么需求将上升，并产生新的费用和 FFO。

下方案例中将说明这个观点。

Example 5. *Chainlink 激励机制能为预言机网络实现 $\$dn^2$ 的加密经济安全，其中节点质押保证金 $\$dn$。那么 1 美元的保证金贡献了 n 的加密经济安全。也就是说，1 美元的经济安全所需的质押保证金为 $1/n$。*

假设网络中加密经济激励完全由 FFO 构成，每个节点的 FFO 上限为 $\$d \leq \$10K$。假设网络中有 $n = 3$ 个节点。每实现 1 美元的加密经济安全所需的平均成本约为 $0.33。假设网络中的总 FFO 超过 3 万美元（如达到 31000 美元）考虑到每个节点的 FFO 存在上限，因此网络的节点（至少）增长至 $n = 4$。现在，实现 1 美元加密经济安全的平均成本降至 $0.25.。

图 18 展示了预言机网络加密经济安全良性循环的全景图。

这里要强调的是，经济安全良性循环的源头是将用户费用放在一个池子里产生效果。用户费用聚在一起形成的 FFO 有利于网络规模扩张，因此可以实现更大的安全保障。我们还发现经济安全良性循环可以为 DON 实现经济可持续性。DON 如果可以满足用户需求，就会不断增长，最后收入会超过预言机节点的运营成本。

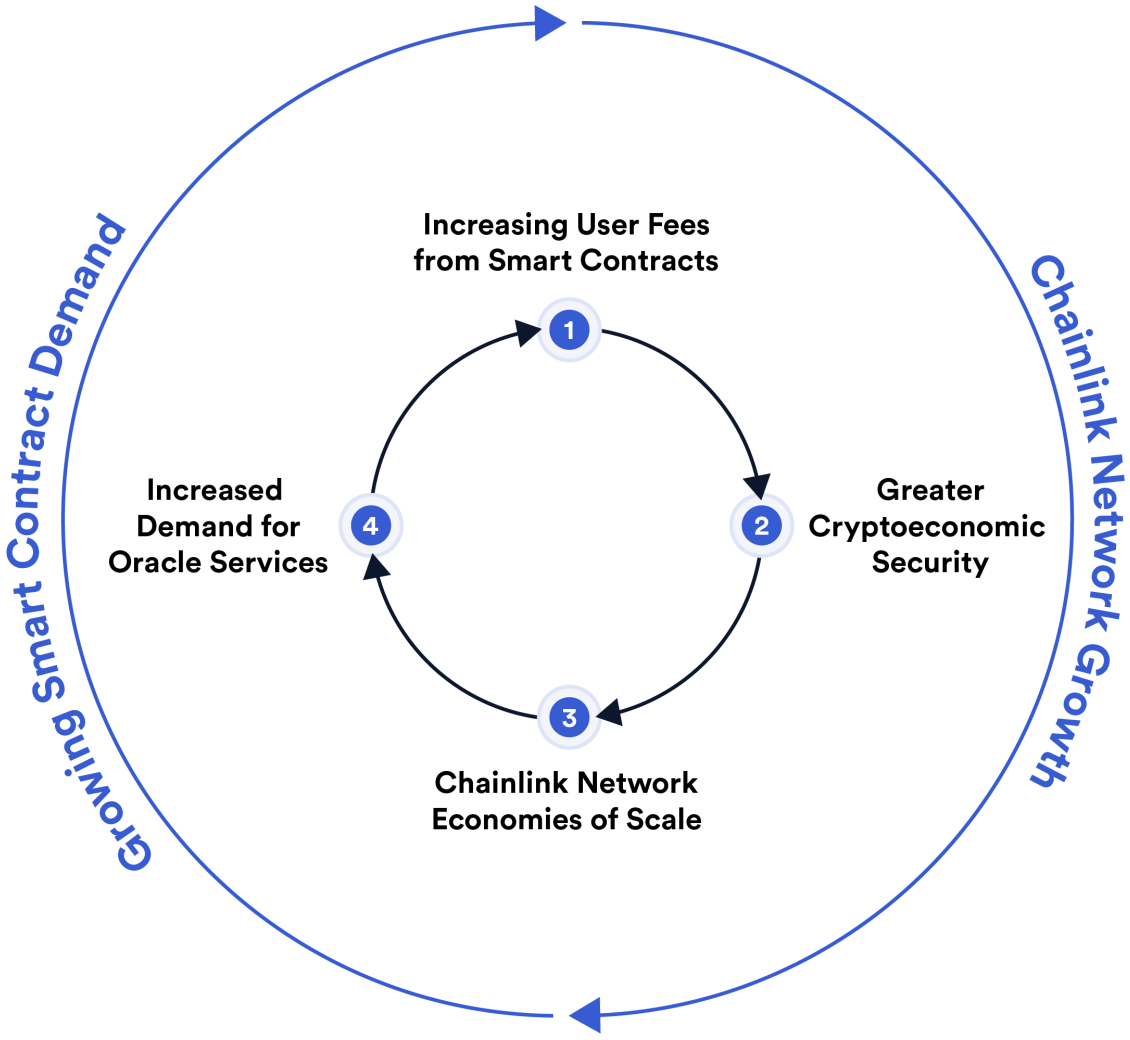

图 18: Chainlink 权益质押机制的良性循环。预言机网络 ① 的用户费用收入增长，导致其规模增长，并因此实现更强大的经济安全保障 ② 。超线性增长可以为 Chainlink 网络实现规模经济 ③ 。具体来说，这意味着加密经济安全的平均成本有所下降，即同样的用户费用或其他类型的权益质押可以实现更高的加密经济安全保障。成本降低最终会使用户受益，并因此吸引到更多预言机用户 ④ 。

9.9 推动网络增长的其他要素

随着 Chainlink 生态不断扩张，将加速吸引更多新用户进入生态，并快速成为区块链经济中重要的基础设施。预言机网络的价值呈超线性增长，也就是说它的增速会超过网络自身的规模扩张。价值增长来既自于规模经济——即随着服务量增长，每名用户的成本效益也升高；又来自于网络效应——即由于更多用户使用 DON 导致网络用量提高。

随着智能合约锁仓量越来越高，并且不断有新的智能合约冒出来，DON 的用户费用收入将不断增长。不断扩大的用户费用池将推动创建更多去中心化服务，并实现良性循环。这个良性循环将解决混合型智能合约生态中关键的“先有鸡还是先有蛋”的问题：创新型智能合约通常需要接入的去中心化服务往往还不存在（比如新的 DeFi 市场通常需要接入新的数据），但与此同时，这些智能合约必须有足够多的经济收入才能存活下去。将 DON 中所有智能合约的用户费放在一个池子里，可以催生出新的去中心化服务，并且不断开发出各种新的混合型智能合约。

总而言之，我们认为 Chainlink 权益质押机制将形成良性循环，增强网络安全，并激活链上去中心化的服务经济。

10 总结

本白皮书为 Chainlink 的未来发展制定了愿景。其核心是极大拓宽预言机网络的服务范围，超越单纯的数据传输功能。Chainlink 将 DON 作为实现去中心化服务的基础，将致力于开发高性能且保障隐私的预言机功能。Chainlink 预言机网络将结合权益质押、安全护栏以及服务水平执行等精心设计的加密经济安全机制，来实现信任最小化。DON 还将为 layer-2 系统实现灵活公允的交易排序服务，并且为经交易池上链的交易降低 gas 费。综上所述，这些能力都将助力 Chainlink 打造出安全且功能丰富的混合型智能合约。

DON 具有非常高的灵活性，将增强现有的 Chainlink 服务，并催生出更多新的智能合约功能和应用。其中包括无缝接入一系列链下系统，基于现有数据创建去中心化身份，建立优先通道，及时传输对基础设施至关重要的交易，以及打造保护隐私的 DeFi 金融工具。

这个愿景非常宏大。短期内，我们希望使混合型智能合约在功能上超越现有智能合约；而长期，我们的目标是打造去中心化的元层。我们对社区开发的各种工具和技术——包括各种共识算法以及零知识证明系统——秉持开放的态度。同样地，我们会

先将本白皮书中提出的想法在 Chainlink 社区中实现，以满足社区用户的需求。我们期待 Chainlink 开启下一个发展阶段，为智能合约实现通用连接性，并打造去中心化的技术，为建立下一代金融和法律体系奠定坚实基础。

致谢

感谢 Julian Alterini 和 Shawn Lee 为白皮书中提供精确的引用数字。

参考文献

[1] DeFi pulse. https://defipulse.com. [Online; accessed 30 Mar. 2021].

[2] dYdX. dydx.exchange. [Online; accessed 30 Mar. 2021].

[3] Ethereum WebAssembly: Metering. https://ewasm.readthedocs.io/en/mkdocs/metering. [Online; accessed 30 Mar. 2021].

[4] Keybase: End-to-end encryption for things that matter. keybase.io. [Online; accessed 30 Mar. 2021].

[5] Loopring: zkRollup exchange and payment protocol. https://loopring.org. [Online; accessed 30 Mar. 2021].

[6] MetaMask: A crypto wallet and gateway to blockchain apps. https://metamask.io. [Online; accessed 30 Mar. 2021].

[7] Nexus Mutual: a people-powered alternative to insurance. nexus.io. [Online; accessed 30 Mar. 2021].

[8] Oasis labs. https://oasislabs.com. [Online; accessed 30 Mar. 2021].

[9] OpenID connect authentication. https://openid.net/connect. [Online; accessed 30 Mar. 2021].

[10] Optimism. https://optimism.io. [Online; accessed 30 Mar. 2021].

[11] Optimistic Rollups - EthHub. https://docs.ethhub.io/ethereum-roadmap/layer-2-scaling/optimistic_rollups. [Online; accessed 30 Mar. 2021].

[12] Starkware: bringing scalability and privacy to a blockchain near you. starkware.co. [Online; accessed 30 Mar. 2021].

[13] Tornado Cash. https://tornado.cash. [Online; accessed 30 Mar. 2021].

[14] uPort: Open identity system for the decentralized web. https://www.uport.me. [Online; accessed 30 Mar. 2021].

[15] Wrapped Bitcoin: WBTC. wbtc.network. [Online; accessed 30 Mar. 2021].

[16] zkSync: Secure, scalable crypto payments. https://zksync.io. [Online; accessed 30 Mar. 2021].

[17] Introduction to emergency shutdown in multi-collateral Dai. *MakerDAO blog*. https://blog.makerdao.com/introduction-to-emergency-shutdown-in-multi-collateral-dai, 12 August 2019.

[18] Chainlink announcement: Introducing the Chainlink on-chain data directory: Data.eth. *Chainlink Blog*. https://blog.chain.link/introducing-the-chainlink-on-chain-data-directory, 28 Dec. 2020.

[19] Speed bump on former AMEX exchange made spreads worse, NYSE say. S&P Global Market Intelligence, https://www.spglobal.com/marketintelligence/en/news-insights/trending/O1_i8QswCPPwWV9HbS-5dw2, 4 Nov. 2019.

[20] Chainlink developers documentation: Adapters. https://docs.chain.link/docs/adapters, [Online; accessed 30 Mar. 2021].

[21] Chainlink developers documentation: Initiators. https://docs.chain.link/docs/initiators, [Online; accessed 30 Mar. 2021].

[22] Ethereum name service: Decentralized naming for wallets, websites, & more. https://ens.domains, [Online; accessed 30 Mar. 2021].

[23] Aavegotchi Wiki contributors. Aavegotchi: Introduction — Aavegotchi Wiki, 2021. [Online; accessed 30 Mar. 2021].

[24] Abe, M., and Fehr, S. Adaptively secure Feldman VSS and applications to universally-composable threshold cryptography. In *Advances in Cryptology (CRYPTO)* (2004), pp. 317–334.

[25] Adler, J., Berryhill, R., Veneris, A., Poulos, Z., Veira, N., and Kastania, A. Astraea: A decentralized blockchain oracle. In *IEEE iThings / GreenCom / CPSCom / SmartData* (2018), pp. 1145–1152.

[26] Adler, J., Berryhill, R., Veneris, A. G., Poulos, Z., Veira, N., and Kastania, A. Astraea: A decentralized blockchain oracle. In *2018 IEEE International Conference on Internet of Things (iThings)* (2018).

[27] Ahmad, A., Joe, B., Xiao, Y., Zhang, Y., Shin, I., and Lee, B. Obfuscuro: A commodity obfuscation engine on Intel SGX. In *Networks and Distributed Security Systems (NDSS)* (2019).

[28] Akhtar, T. Kraken Exchange brings its spot price data to DeFi via new Chainlink node. *Coindesk* (1 Feb. 2021). https://www.coindesk.com/kraken-exchange-brings-its-spot-price-data-to-defi-via-new-chainlink-node.

[29] Ali, M., Nelson, J., Shea, R., and Freedman, M. J. Blockstack: A global naming and storage system secured by blockchains. In *USENIX Annual Technical Conference (ATC)* (2016).

[30] Allen, S., Čapkun, S., Eyal, I., Fanti, G., Ford, B., Grimmelmann, J., Juels, A., Kostiainen, K., Meiklejohn, S., Miller, A., et al. Design choices for central bank digital currency: Policy and technical considerations. *NBER Working Paper Series*, Working paper 27634 (Aug. 2020).

[31] Anderson, P. L., McLellan, R. D., Overton, J. P., and Wolfram, G. L. Price elasticity of demand. *McKinac Center for Public Policy 13* (1997), 2010.

[32] Arnosti, N., and Weinberg, S. M. Bitcoin: A natural oligopoly. *arXiv preprint arXiv:1811.08572* (2018).

[33] Asayag, A., Cohen, G., Grayevsky, I., Leshkowitz, M., Rottenstreich, O., Tamari, R., and Yakira, D. Helix: a scalable and fair consensus algorithm. Tech. rep., Technical report, Orbs Research, 2018.

[34] Attah, E. Five most prolific 51% attacks in crypto: Verge, Ethereum Classic, Bitcoin Gold, Feathercoin, Vertcoin. *CryptoSlate* (24 April 2019).

[35] Axie Infinity. Axie Infinity integrates Chainlink oracles! *Axie Infinity blog.* https://axieinfinity.medium.com/axie-infinity-integrates-chainlink-oracles-aa93d3d0983e, 16 Nov. 2020.

[36] Baird, L., Luykx, A., and Madsen, P. Hedera technical insights: Fair timestamping and fair ordering of transactions. *Hedera Blog.* https://hedera.com/blog/fair-timestamping-and-fair-ordering-of-transactions, 12 Apr. 2020.

[37] Baum, C., Orsini, E., Scholl, P., and Soria-Vazquez, E. Efficient constant-round MPC with identifiable abort and public verifiability. In *Advances in Cryptology (CRYPTO)* (2020), pp. 562–592.

[38] Ben-Sasson, E., Bentov, I., Horesh, Y., and Riabzev, M. Scalable zero knowledge with no trusted setup. In *Advances in Cryptology (CRYPTO)* (2019), pp. 701–732.

[39] Bender, A., Katz, J., and Morselli, R. Ring signatures: Stronger definitions, and constructions without random oracles. *Journal of Cryptology 22*, 1 (2009), 114–138.

[40] Benet, J. IPFS-content addressed, versioned, P2P file system. *arXiv preprint arXiv:1407.3561* (2014).

[41] Benhamouda, F., Gentry, C., Gorbunov, S., Halevi, S., Krawczyk, H., Lin, C., Rabin, T., and Reyzin, L. Can a blockchain keep a secret? *IACR Cryptol. ePrint Arch. 2020* (2020), 464.

[42] Benhamouda, F., Halevi, S., and Halevi, T. Supporting private data on Hyperledger Fabric with secure multiparty computation. *IBM Journal of Research and Development 63*, 2/3 (2019), 3–1.

[43] Benson, J. ConsenSys wades into compliance for Ethereum tokens. https://decrypt.co/31641/consensys-wades-into-compliance-for-ethereum-tokens, 8 June 2020. [Online; accessed 30 Mar. 2021].

[44] Bentov, I., Ji, Y., Zhang, F., Breidenbach, L., Daian, P., and Juels, A. Tesseract: Real-time cryptocurrency exchange using trusted hardware. In *ACM Conference on Computer and Communications Security (ACM CCS)* (2019), pp. 1521–1538.

[45] Berberich, M., and Steiner, M. Blockchain technology and the GDPR-how to reconcile privacy and distributed ledgers. *Eur. Data Prot. L. Rev. 2* (2016), 422.

[46] Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Gollamudi, A., Gonthier, G., Kobeissi, N., Kulatova, N., Rastogi, A., Sibut-Pinote, T., Swamy, N., et al. Formal verification of smart contracts: Short paper. In *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security* (2016), pp. 91–96.

[47] Birman, K. P., and Schneider, F. B. The monoculture risk put into context. *IEEE Security & Privacy 7*, 1 (2009), 14–17.

[48] Blocknative. Evidence of mempool manipulation on Black Thursday: Hammerbots, mempool compression, and spontaneous stuck transactions. *Blocknative Blog*. https://blog.blocknative.com/blog/mempool-forensics, 22 July 2020.

[49] Bloemenm, R., Logvinov, L., and Evans, J. Ethereum Improvement Proposal (EIP) 712: Ethereum typed structured data hashing and signing. https://eips.ethereum.org/EIPS/eip-712, 12 Sept. 2017.

[50] Bogatyy, I. Implementing Ethereum trading front-runs on the Bancor exchange in Python. *Hackernoon*. https://hackernoon.com/front-running-bancor-in-150-lines-of-python-with-ethereum-api-d5e2bfd0d798, 17 Aug. 2017.

[51] Böhme, R., Christin, N., Edelman, B., and Moore, T. Bitcoin: Economics, technology, and governance. *Journal of Economic Perspectives 29*, 2 (2015), 213–38.

[52] Bojja Venkatakrishnan, S., Fanti, G., and Viswanath, P. Dandelion: Redesigning the Bitcoin network for anonymity. *Proceedings of the ACM on Measurement and Analysis of Computing Systems 1*, 1 (2017), 1–34.

[53] Boneh, D., Bonneau, J., Bünz, B., and Fisch, B. Verifiable delay functions. In *Advances in Cryptology (CRYPTO)* (2018), pp. 757–788.

[54] Boneh, D., Lynn, B., and Shacham, H. Short signatures from the Weil pairing. In *Advances in Cryptology (ASIACRYPT)* (2001), pp. 514–532.

[55] Boneh, D., Lynn, B., and Shacham, H. Short signatures from the weil pairing. In *ASIACRYPT* (2001), vol. 2248 of *Lecture Notes in Computer Science*, Springer, pp. 514–532.

[56] Bootle, J., Cerulli, A., Chaidos, P., Groth, J., and Petit, C. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In *Advances in Cryptology (EUROCRYPT)* (2016), pp. 327–357.

[57] Bowers, K. D., Juels, A., and Oprea, A. HAIL: A high-availability and integrity layer for cloud storage. In *ACM Conference on Computer and Communications Security (ACM CCS)* (2009), pp. 187–198.

[58] Bowman, M., Miele, A., Steiner, M., and Vavala, B. Private data objects: an overview. *arXiv preprint arXiv:1807.05686* (2018).

[59] Boyle, E., Goldwasser, S., and Ivan, I. Functional signatures and pseudorandom functions. In *Public Key Cryptography (PKC)* (2014), pp. 501–519.

[60] Breidenbach, L., Cachin, C., Coventry, A., Juels, A., and Miller, A. Chainlink off-chain reporting protocol. https://chain.link/ocrpaper, 2021. [Online; accessed 30 Mar. 2021].

[61] Breidenbach, L., Daian, P., and Tramèr, F. GasToken. gastoken.io. [Online; accessed 30 Mar. 2021].

[62] Breidenbach, L., Daian, P., Tramèr, F., and Juels, A. Enter the Hydra: Towards principled bug bounties and exploit-resistant smart contracts. In *USENIX Security Symposium (USENIX Security)* (2018), pp. 1335–1352.

[63] Brown, M., and Housley, R. Transport layer security (TLS) evidence extensions, Nov. 2006. Working Draft, IETF Secretariat, Internet-Draft drafthousley-evidence-extns-01.

[64] Budish, E., Cramton, P., and Shim, J. The high-frequency trading arms race: Frequent batch auctions as a market design response. *The Quarterly Journal of Economics 130*, 4 (2015), 1547–1621.

[65] Bünz, B., Agrawal, S., Zamani, M., and Boneh, D. Zether: Towards privacy in a smart contract world. In *Financial Cryptography and Data Security (FC)* (2020), pp. 423–443.

[66] Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., and Maxwell, G. Bulletproofs: Short proofs for confidential transactions and more. In *IEEE Symposium on Security and Privacy (SP)* (2018), IEEE, pp. 315–334.

[67] Buterin, V. The p + epsilon attack. *Ethereum Blog*. https://blog.ethereum.org/2015/01/28/p-epsilon-attack, 28 Jan. 2015.

[68] Buterin, V. SchellingCoin: A Minimal-Trust Universal Data Feed. *Ethereum Blog*. https://blog.ethereum.org/2014/03/28/schellingcoin-a-minimal-trust-universal-data-feed, 28 Mar. 2014.

[69] Buterin, V. On-chain scaling to potentially ∼500 tx/sec through mass tx validation. *Ethereum Blog*. https://ethresear.ch/t/

on-chain-scaling-to-potentially-500-tx-sec-through-mass-tx-validation/3477, 3 Sept. 2018.

[70] Buterin, V., Dietrichs, A., Garnett, M., Villanueva, W., and Wilson, S. Ethereum Improvement Proposal (EIP) 2938: Account abstraction. https://eips.ethereum.org/EIPS/eip-2938, 4 Sept. 2020.

[71] Cachin, C., Kursawe, K., Petzold, F., and Shoup, V. Secure and efficient asynchronous broadcast protocols (extended abstract). In *Advances in Cryptology: CRYPTO 2001* (2001), J. Kilian, Ed., vol. 2139, Springer, pp. 524–541.

[72] Cachin, C., and Vukolić, M. Blockchain consensus protocols in the wild. *arXiv preprint arXiv:1707.01873* (2017).

[73] Camenisch, J., and Lysyanskaya, A. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *International conference on the theory and applications of cryptographic techniques* (2001), pp. 93–118.

[74] Cecchetti, E., Fisch, B., Miers, I., and Juels, A. PIEs: Public incompressible encodings for decentralized storage. In *ACM Conference on Computer and Communications Security (ACM CCS)* (2019), pp. 1351–1367.

[75] Chainlink. How to build a parametric insurance smart contract. *Chainlink Blog*. https://blog.chain.link/parametric-insurance-smart-contract, 15 Dec. 2020.

[76] Chainlink. Chainlink Proof of Reserve: Bringing transparency to DeFi collateral. *Chainlink Blog*. https://blog.chain.link/chainlink-proof-of-reserve-bringing-transparency-to-defi-collateral, 30 Nov. 2020.

[77] Chainlink. How Chainlink supports any off-chain data resource and computation. *Chainlink Blog*. https://blog.chain.link/how-chainlink-supports-any-off-chain-data-resource-and-computation, 8 Mar. 2021.

[78] Chainlink. Introduction to Chainlink VRF. *Chainlink Developers Documentation*. https://docs.chain.link/docs/chainlink-vrf, [Online; accessed 30 Mar. 2021].

[79] Chan, J., Warwick, K., and Ennis, C. Synthetix improvement proposal (SIP) 6: Frontrunning protection. https://sips.synthetix.io/sips/sip-6, 27 June 2019.

[80] Cheng, R., Zhang, F., Kos, J., He, W., Hynes, N., Johnson, N., Juels, A., Miller, A., and Song, D. Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts. In *IEEE European Symposium on Security and Privacy (EuroS&P)* (2019), IEEE, pp. 185–200.

[81] Chjango Unchained. Tendermint explained —bringing BFT-based PoS to the public blockchain domain. *Cosmos Blog*. https://blog.cosmos.network/

tendermint-explained-bringing-bft-based-pos-to-the-public-blockchain-domain-f22e274a0fdb, 10 May 2018.

[82] Clarkson, M. R., Chong, S., and Myers, A. C. Civitas: Toward a secure voting system. In *IEEE Symposium on Security and Privacy (SP)* (2008), IEEE, pp. 354–368.

[83] Cloudflare. Understanding AMP real URL. https://support.cloudflare.com/hc/en-us/articles/360029367652-Understanding-Amp-Real-URL, [Online; accessed 30 Mar. 2021].

[84] Costan, V., and Devadas, S. Intel SGX explained. *IACR Cryptol. ePrint Arch. 2016*, 86 (2016), 1–118.

[85] Crawley, J. Chainlink integration connects Filecoin to smart contract-enabled blockchains. *Coindesk* (24 Mar. 2021). https://www.coindesk.com/filecoin-chainlink-integration-smart-contract-enabled-blockchains.

[86] Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Sirer, E. G., et al. On scaling decentralized blockchains. In *Financial Cryptography and Data Security (FC)* (2016), pp. 106–125.

[87] Cronje, A. Scaling Keep3r with Chainlink. *Medium Blog Post*. https://andrecronje.medium.com/scaling-keep3r-with-chainlink-2832bbc76506, 2 Dec. 2020.

[88] Crypto51. PoW 51% attack cost. https://www.crypto51.app. [Online; accessed 30 Mar. 2021].

[89] Cusack, L. Improving PoolTogether with Chainlink VRF. *PoolTogether Blog*. https://medium.com/pooltogether/improving-pooltogether-with-chainlink-vrf-dcf1a3d6ea, 11 May 2020.

[90] Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., and Juels, A. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE Symposium on Security and Privacy (SP)* (2020), pp. 566–583.

[91] Dale, B. Feature from tech no collateral required: How Aave brought unsecured borrowing to DeFi. *Coindesk* (24 Aug. 2020). https://www.coindesk.com/aave-unsecured-borrowing-defi.

[92] Decentralized Identity Foundation. DIF website. https://identity.foundation. [Online; accessed 30 Mar. 2021].

[93] del Castillo, M. How to track official election results on Ethereum and EOS. *Forbes* (3 Nov. 2020). https://www.forbes.com/sites/michaeldelcastillo/2020/11/03/how-to-track-official-election-results-on-ethereum-and-eos.

[94] Dierks, T., and Rescorla, E. The transport layer security (tls) protocol version 1.2.

[95] Dingledine, R., Mathewson, N., and Syverson, P. Tor: The second-generation onion router. Tech. rep., Naval Research Lab Washington DC, 2004.

[96] Dong, C., Wang, Y., Aldweesh, A., McCorry, P., and van Moorsel, A. Betrayal, distrust, and rationality: Smart counter-collusion contracts for verifiable cloud computing. In *ACM Conference on Computer and Communications Security (CCS)* (2017), pp. 211–227.

[97] Duan, S., Reiter, M. K., and Zhang, H. Secure causal atomic broadcast, revisited. In *Proc. 47th International Conference on Dependable Systems and Networks* (2017), pp. 61–72.

[98] Ellis, S., Juels, A., and Nazarov, S. Chainlink: a decentralized oracle network. https://chain.link/whitepaper, 4 Sept. 2017.

[99] Enigma Project. New to enigma? start here. *Enigma Blog*. https://blog.enigma.co/welcome-to-enigma-start-here-e65c8c9125ef, 5 Nov. 2018.

[100] Erik Marks, P. G. Ethereum Improvement Proposal (EIP) 712: Wallet add Ethereum chain RPC method. https://eips.ethereum.org/EIPS/eip-3085, 1 Nov. 2020.

[101] Eskandari, S., Moosavi, S., and Clark, J. SoK: Transparent dishonesty: front-running attacks on blockchain. In *International Conference on Financial Cryptography and Data Security* (2019), pp. 170–189.

[102] EtherCards. The EtherCards platform. https://docs.ether.cards/platform.html. [Online; accessed 30 Mar. 2021].

[103] Ethereum Foundation. Ethereum 2.0 (Eth2). https://ethereum.org/en/eth2, [Online; accessed 30 Mar. 2021].

[104] etherscan. io. Ethereum Average Block Time Chart | Etherscan. https://etherscan.io/chart/blocktime, Sep 2020. [Online; accessed 30 Mar. 2021].

[105] European Data Protection Board. Guidelines 3/2018 on the territorial scope of the GDPR (article 3) [version 2.1]. https://edpb.europa.eu/sites/edpb/files/files/file1/edpb_guidelines_3_2018_territorial_scope_after_public_consultation_en_1.pdf, 12 Nov. 2019.

[106] Eyal, I., and Sirer, E. G. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography and Data Security (FC)* (2014), pp. 436–454.

[107] Fanti, G., Venkatakrishnan, S. B., Bakshi, S., Denby, B., Bhargava, S., Miller, A., and Viswanath, P. Dandelion++ lightweight cryptocurrency networking with formal anonymity guarantees. *Proceedings of the ACM on Measurement and Analysis of Computing Systems 2*, 2 (2018), 1–35.

[108] Feldman, P. A practical scheme for non-interactive verifiable secret sharing. In *Annual Symposium on Foundations of Computer Science (FOCS)* (1987), IEEE, pp. 427–438.

[109] Felten, E. What's up with Rollup. *Medium*, Offchain Labs. https://medium.com/offchainlabs/whats-up-with-rollup-db8cd93b314e, 18 Nov. 2019.

[110] Felten, E. Front-Running as a Service. *Medium*, Offchain Labs. https://medium.com/offchainlabs/front-running-as-a-service-334c929c945a, 29 June 2020.

[111] Fisch, B. Tight proofs of space and replication. In *Advances in Cryptology (EUROCRYPT)* (2019), pp. 324–348.

[112] Fisch, B., Bonneau, J., Greco, N., and Benet, J. Scaling Proof-of-Replication for Filecoin mining. *Technical report, Stanford University* (2018).

[113] Franz, M. E unibus pluram: massive-scale software diversity as a defense mechanism. In *Proceedings of the 2010 New Security Paradigms Workshop* (2010), pp. 7–16.

[114] Gallo, A. A refresher on net present value. *Harvard Business Review 19* (2014).

[115] Garay, J., Kiayias, A., and Leonardos, N. The bitcoin backbone protocol: Analysis and applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (2015), pp. 281–310.

[116] Gennaro, R., and Goldfeder, S. Fast multiparty threshold ECDSA with fast trustless setup. In *ACM Conference on Computer and Communications Security* (2018), pp. 1179–1194.

[117] Gennaro, R., and Goldfeder, S. One round threshold ECDSA with identifiable abort. *IACR Cryptol. ePrint Arch. 2020* (2020), 540.

[118] Gennaro, R., Goldfeder, S., and Narayanan, A. Threshold-optimal dsa/ecdsa signatures and an application to bitcoin wallet security. In *International Conference on Applied Cryptography and Network Security* (2016), pp. 156–174.

[119] George, W., and Lesaege, C. An analysis of $p + \epsilon$ attacks on various models of schelling game based systems. In *Cryptoeconomic Systems (CES)* (2020).

[120] Gilad, Y., Hemo, R., Micali, S., Vlachos, G., and Zeldovich, N. Algorand: Scaling byzantine agreements for cryptocurrencies. In *ACM Symposium on Operating Systems Principles (SOSP)* (2017), pp. 51–68.

[121] Gluchowski, A. Evaluating Ethereum L2 Scaling Solutions: A Comparison Framework. *Medium* (Aug 2020).

[122] Goldwasser, S., Ben-Or, M., and Wigderson, A. Completeness theorems for non-cryptographic fault-tolerant distributed computing. In *Symposium on Theory of Computing (STOC)* (1988), pp. 1–10.

[123] Gollmann, D. *Computer security, 3rd edition.* John Wiley & Sons, 2011.

[124] Google. Serve AMP using signed exchanges. https://amp.dev/documentation/guides-and-tutorials/optimize-and-measure/signed-exchange. [Online; accessed 30 Mar. 2021].

[125] Gorokh, A., Banerjee, S., and Iyer, K. When bribes are harmless: The power and limits of collusion-resilient mechanism design. *SSRN*. https://ssrn.com/abstract=3125003, 2019.

[126] Groth, J. On the size of pairing-based non-interactive arguments. In *Advances in Cryptology (CRYPTO)* (2016), pp. 305–326.

[127] Gu, W. C., Raghuvanshi, A., and Boneh, D. Empirical measurements on pricing oracles and decentralized governance for stablecoins. *SSRN 3611231* (2020).

[128] Hu, E. Intentional access delays, market quality, and price discovery: Evidence from IEX becoming an exchange. *SSRN*. https://ssrn.com/abstract=3195001, 15 Mar. 2019.

[129] Hyperledger Project. Hyperledger Indy. https://www.hyperledger.org/use/hyperledger-indy. [Online; accessed 30 Mar. 2021].

[130] Jackson, M. O., and Wilkie, S. Endogenous games and mechanisms: Side payments among players. In *The Review of Economic Studies* (2005), vol. 72, pp. 543–566.

[131] Jansen, M., Hdhili, F., Gouiaa, R., and Qasem, Z. Do smart contract languages need to be Turing complete? In *International Congress on Blockchain and Applications* (2019), pp. 19–26.

[132] Johnson, R., Molnar, D., Song, D., and Wagner, D. Homomorphic signature schemes. In *Cryptographers' Track at the RSA Conference (CT-RSA)* (2002), pp. 244–262.

[133] Johnson, S., Scarlata, V., Rozas, C., Brickell, E., and Mckeen, F. Intel® Software Guard Extensions: EPID provisioning and attestation services, 2016. White Paper.

[134] Joint Task Force. Security and privacy controls for federal information systems and organizations. *NIST Special Publication 800*, 53 Rev. 5 (2020).

[135] Juels, A., Breidenbach, L., Coventry, A., Nazarov, S., Ellis, S., and Magauran, B. Mixicles. https://chain.link/mixicles.pdf, 2019.

[136] Juels, A., Breidenbach, L., Daian, P., Ji, Y., and Tramèr, F. Project Chicago for the study of cryptocurrencies. projectchicago.io. [Online; accessed 30 Mar. 2021].

[137] Juels, A., Breidenbach, L., and Tramèr, F. Fair Sequencing Services: Enabling a provably fair DeFi ecosystem. *Chainlink Blog*. https://blog.chain.link/chainlink-fair-sequencing-services-enabling-a-provably-fair-defi-ecosystem, 11 Sept. 2020.

[138] Juels, A., Catalano, D., and Jakobsson, M. Coercion-resistant electronic elections. In *Towards Trustworthy Elections*. Springer, 2010, pp. 37–63.

[139] Juels, A., Kosba, A., and Shi, E. The ring of Gyges: Investigating the future of criminal smart contracts. In *ACM Conference on Computer and Communications Security (ACM CCS)* (2016), pp. 283–295.

[140] Kaiko. Kaiko partners with Chainlink to bring cryptocurrency market data to smart contracts. *Kaiko Blog*. https://www.kaiko.com/blogs/latest-news/kaiko-partners-with-chainlink-to-bring-cryptocurrency-market-data-to-smart-contracts, 14 Nov. 2018.

[141] Kalodner, H., Goldfeder, S., Chen, X., Weinberg, S. M., and Felten, E. W. Arbitrum: Scalable, private smart contracts. In *USENIX Security Symposium (USENIX Security)* (2018), pp. 1353–1370.

[142] Kalodner, H. A., Carlsten, M., Ellenbogen, P., Bonneau, J., and Narayanan, A. An empirical study of Namecoin and lessons for decentralized namespace design. In *Workshop on Economics of Information Security (WEIS)* (2015).

[143] Kapilkov, M. Deutsche Telekom's T-Systems is now a Chainlink node operator. *Cointelegraph* (22 July 2020). https://www.coindesk.com/kraken-exchange-brings-its-spot-price-data-to-defi-via-new-chainlink-node.

[144] Kelkar, M., Zhang, F., Goldfeder, S., and Juels, A. Order-fairness for Byzantine consensus. In *Advances in Cryptology (CRYPTO)* (2020), pp. 451–480.

[145] Khalil, R., Gervais, A., and Felley, G. TEX - a securely scalable trustless exchange. *IACR Cryptol. ePrint Arch. 2019* (2019), 265.

[146] Kiayias, A., Russell, A., David, B., and Oliynykov, R. Ouroboros: A provably secure Proof-of-Stake blockchain protocol. In *Advances in Cryptology (CRYPTO)* (2017), pp. 357–388.

[147] Kivlighan, I. The Aave oracle network powered by Chainlink is now live! *Aave Blog*. https://medium.com/aave/the-aave-oracle-network-powered-by-chainlink-is-now-live-45bb8a5a8c4e, 9 Jan. 2020.

[148] Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., Syta, E., and Ford, B. Omniledger: A secure, scale-out, decentralized ledger via sharding. In *IEEE Symposium on Security and Privacy (SP)* (2018), pp. 583–598.

[149] Kosba, A., Miller, A., Shi, E., Wen, Z., and Papamanthou, C. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *IEEE Symposium on Security and Privacy (SP)* (2016), IEEE, pp. 839–858.

[150] Kroll, J. A., Davey, I. C., and Felten, E. W. The economics of bitcoin mining, or bitcoin in the presence of adversaries. In *Workshop on the Economics of Information Systems (WEIS)* (2013), vol. 2013, p. 11.

[151] Kursawe, K. Wendy, the good little fairness widget. *arXiv preprint arXiv:2007.08303* (2020).

[152] Lee, D., Kohlbrenner, D., Shinde, S., Asanović, K., and Song, D. Keystone: An open framework for architecting trusted execution environments. In *European Conference on Computer Systems (EuroSys)* (2020), pp. 1–16.

[153] Lev-Ari, K., Spiegelman, A., Keidar, I., and Malkhi, D. Fairledger: A fair blockchain protocol for financial institutions. *arXiv preprint arXiv:1906.03819* (2019).

[154] Levi, Y. Bancor's response to today' s smart contract vulnerability. *Bancor Blog*. https://blog.bancor.network/bancors-response-to-today-s-smart-contract-vulnerability-dc888c589fe4, 18 June 2020.

[155] Lewis, M. *Flash boys: a Wall Street revolt*. WW Norton & Company, 2014.

[156] Libra Association. Libra whitepaper v2.0. https://libra.org/en-US/white-paper, April 2020.

[157] Lu, D., Yurek, T., Kulshreshtha, S., Govind, R., Kate, A., and Miller, A. HoneyBadgerMPC and Asynchromix: Practical asynchronous MPC and its application to anonymous communication. In *ACM Conference on Computer and Communications Security (ACM CCS)* (2019), pp. 887–903.

[158] Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., and Saxena, P. A secure sharding protocol for open blockchains. In *ACM Conference on Computer and Communications Security (ACM CCS)* (2016), pp. 17–30.

[159] Manuskin, A. The fastest draw on the blockchain (BZRX example). *Medium*. https://medium.com/@amanusk/the-fastest-draw-on-the-blockchain-bzrx-example-6bd19fabdbe1, 22 July 2020.

[160] Maram, D., Malvai, H., Zhang, F., Jean-Louis, N., Frolov, A., Kell, T., Lobban, T., Moy, C., Juels, A., and Miller, A. CanDID: Can-do decentralized identity with legacy compatibility, Sybil-resistance, and accountability. In *IEEE Symposium on Security and Privacy (SP)* (2021. To appear.).

[161] Maram, S. K. D., Zhang, F., Wang, L., Low, A., Zhang, Y., Juels, A., and Song, D. CHURP: Dynamic-committee proactive secret sharing. In *ACM Conference on Computer and Communications Security (ACM CCS)* (2019), pp. 2369–2386.

[162] Marino, W. Smart-contract escape hatches: The Dao of the DAO. *Hacking, Distributed*. https://hackingdistributed.com/2016/06/22/smart-contract-escape-hatches, 22 June 2016.

[163] Matetic, S., Schneider, M., Miller, A., Juels, A., and Capkun, S. Delegatee: Brokered delegation using trusted execution environments. In *USENIX Security Symposium (USENIX Security)* (2018), pp. 1387–1403.

[164] McCorry, P., Hicks, A., and Meiklejohn, S. Smart contracts for bribing miners. In *Financial Cryptography and Data Security (FC)* (2018), pp. 3–18.

[165] McCorry, P., Hicks, A., and Meiklejohn, S. Smart contracts for bribing miners. In *Financial Cryptography (FC)* (2018).

[166] Metcalfe, B. Metcalfe's law after 40 years of ethernet. *Computer 46*, 12 (2013), 26–31.

[167] Micali, S. Algorand' s smart contract architecture. *Algorand Blog*. https://www.algorand.com/resources/blog/algorand-smart-contract-architecture, 27 May 2020.

[168] Miller, A., Bentov, I., Bakshi, S., Kumaresan, R., and McCorry, P. Sprites and state channels: Payment networks that go faster than Lightning. In *Financial Cryptography and Data Security (FC)* (2019), pp. 508–526.

[169] Miller, A., Cai, Z., and Jha, S. Smart contracts and opportunities for formal methods. In *Symposium on Leveraging Applications of Formal Methods* (2018), pp. 280–299.

[170] Moos, M. Mining pool censorship could make Zcash 'mostly unusable'. *Cryptoslate*. https://cryptoslate.com/mining-pool-censorship-zcash-unusable, 7 June 2019.

[171] Moroz, D. J., Aronoff, D. J., Lovejoy, J., Narula, N., and Parkes, D. C. Double-spend counterattacks. In *Cryptoeconomic Systems (CES)* (2020).

[172] Möser, M., Eyal, I., and Sirer, E. G. Bitcoin covenants. In *Financial Cryptography and Data Security (FC)* (2016), pp. 126–141.

[173] Narayanan, A., and Clark, J. Bitcoin's academic pedigree. *Communications of the ACM 60*, 12 (2017), 36–45.

[174] Nayak, K., Fletcher, C. W., Ren, L., Chandran, N., Lokam, S. V., Shi, E., and Goyal, V. HOP: Hardware makes obfuscation practical. In *Networks and Distributed Security Systems (NDSS)* (2017).

[175] Nazarov, S., Shukla, P., Erwin, A., and Rajput, A. Bridging the governance gap: Interoperability for blockchain and legacy systems. World Economic Forum whitepaper. https://www.weforum.org/whitepapers/bridging-the-governance-gap-interoperability-for-blockchain-and-legacy-systems, Dec. 2020.

[176] Nisan, N., Tardos, E., Roughgarden, T., and Vazirani, V. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[177] Nowak, M., and Sigmund, K. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. In *Nature* (1993), vol. 364, pp. 56–58.

[178] Osipovich, A. More exchanges add 'speed bumps,' defying high-frequency traders. *Wall Street Journal* (29 July 2019).

[179] Papadopoulos, D., Wessels, D., Huque, S., Naor, M., Včelák, J., Reyzin, L., and Goldberg, S. Making nsec5 practical for dnssec. *Cryptology ePrintArchive, Report 2017/099* (2017).

[180] Pass, R., and Shi, E. Fruitchains: A fair blockchain. In *ACM Symposium on Principles of Distributed Computing (PODC)* (2017), pp. 315–324.

[181] Pass, R., and Shi, E. Hybrid consensus: Efficient consensus in the permissionless model. In *International Symposium on Distributed Computing (DISC)* (2017), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[182] Pass, R., Shi, E., and Tramèr, F. Formal abstractions for attested execution secure processors. In *Advances in Cryptology (EUROCRYPT)* (2017), pp. 260–289.

[183] Perez, D., and Livshits, B. Broken metre: Attacking resource metering in EVM. *arXiv preprint arXiv:1909.07220* (2019).

[184] Peterson, J., Krug, J., Zoltu, M., Williams, A., and Alexander, S. Augur: a decentralized oracle and prediction market platform (v2.0). https://augur.net/whitepaper.pdf, 2019.

[185] Poon, J., and Buterin, V. Plasma: Scalable autonomous smart contracts (working draft). https://www.plasma.io/plasma.pdf, 11 Aug. 2017.

[186] Poon, J., and Dryja, T. The Bitcoin lightning network: Scalable off-chain instant payments. http://lightning.network/lightning-network-paper.pdf, 2016.

[187] Protocol Labs. Filecoin: A decentralized storage network. https://filecoin.io/filecoin.pdf, 19 July 2017.

[188] Qin, K., Zhou, L., Livshits, B., and Gervais, A. Attacking the DeFi ecosystem with flash loans for fun and profit. *arXiv preprint arXiv:2003.03810* (2020).

[189] Reiter, M. K., and Birman, K. P. How to securely replicate services. *ACM Trans. Program. Lang. Syst. 16*, 3 (May 1994), 986–1009.

[190] Ritzdorf, H., Wüst, K., Gervais, A., Felley, G., and Capkun, S. TLS-N: Non-repudiation over TLS enabling ubiquitous content signing. In *Networks and Distributed Security Systems (NDSS)* (2018).

[191] Rivest, R. L., Shamir, A., and Tauman, Y. How to leak a secret: Theory and applications of ring signatures. In *Theoretical Computer Science, Essays in Memory of Shimon Even*. Springer, 2006, pp. 164–186.

[192] Roberts, J. J. Exclusive: Coinbase buys Xapo custody for $55 million, eyes lending business. *Fortune* (15 Aug. 2019). https://fortune.com/2019/08/15/coinbase-xapo-bitcoin-custody.

[193] Roberts, J. J., and Rapp, N. Nearly 4 million Bitcoins lost forever, new study says. *Fortune* (25 Nov. 2017).

[194] Robinson, D., and Konstantopoulos, G. Ethereum is a dark forest. *Medium*. https://medium.com/@danrobinson/ethereum-is-a-dark-forest-ecc5f0505dff, 28 Aug. 2020.

[195] Rutter, K. R3 reports: If at first you don't succeed, try a decentralized KYC platform: Will blockchain technology give corporate KYC a second chance?, 22 July 2018.

[196] Saltzer, J. H., and Schroeder, M. D. The protection of information in computer systems. *Proceedings of the IEEE 63*, 9 (1975), 1278–1308.

[197] Schultz, D. A., Liskov, B., and Liskov, M. Mobile proactive secret sharing. In *ACM Symposium on Principles of Distributed Computing (PODC)* (2008), ACM, pp. 458–458.

[198] Shacham, H., and Waters, B. Compact proofs of retrievability. In *Advances in Cryptology (ASIACRYPT)* (2008), pp. 90–107.

[199] Sinclair, S. Ethereum Classic suffers second 51% attack in a week. *Coindesk* (6 Aug. 2020). https://www.coindesk.com/ethereum-classic-suffers-second-51-attack-in-a-week.

[200] StarkWare. Volition and the Emerging Data Availability spectrum. *Starkware Blog*. https://medium.com/starkware/volition-and-the-emerging-data-availability-spectrum-87e8bfa09bb, 14 June 2020.

[201] Stinson, D. R., and Strobl, R. Provably secure distributed Schnorr signatures and a (t, n)-threshold scheme for implicit certificates. In *Australasian Conference on Information Security and Privacy* (2001), pp. 417–434.

[202] Team Rocket. Snowflake to Avalanche: A novel metastable consensus protocol family for cryptocurrencies. https://ipfs.io/ipfs/QmUy4jh5mGNZvLkjies1RWM4YuvJh5o2FYopNPVYwrRVGV, 16 May 2018. [Online; accessed 30 Mar. 2021].

[203] Tezos. Proof-of-Stake in Tezos. https://tezos.gitlab.io/whitedoc/proof_of_stake.html, [Online; accessed 30 Mar. 2021].

[204] The LibraBFT Team. State machine replication in the Libra blockchain. https://developers.libra.org/docs/assets/papers/libra-consensus-state-machine-replication-in-the-libra-blockchain/2020-05-26.pdf, 1 May 2020.

[205] Tomescu, A., Abraham, I., Buterin, V., Drake, J., Feist, D., and Khovratovich, D. Aggregatable subvector commitments for stateless cryptocurrencies. *IACR Cryptol. ePrint Arch. 2020* (2020), 527.

[206] Tramèr, F., Zhang, F., Lin, H., Hubaux, J.-P., Juels, A., and Shi, E. Sealed-glass proofs: Using transparent enclaves to prove and sell knowledge. In *IEEE European Symposium on Security and Privacy (EuroS&P)* (2017), pp. 19–34.

[207] Travers, G. All Synths are now powered by Chainlink decentralised oracles. *Synthetix Blog*. https://blog.synthetix.io/all-synths-are-now-powered-by-chainlink-decentralised-oracles, 1 Sept. 2020.

[208] U.S. Securities and Exchange Commission. Updated investor bulletin: Accredited investors. https://www.investor.gov/introduction-investing/general-resources/news-alerts/alerts-bulletins/investor-bulletins/updated-3, 21 Jan. 2019.

[209] U.S. Securities and Exchange Commission. SEC modernizes the accredited investor definition. *SEC Press Release*. https://www.sec.gov/news/press-release/2020-191, 26 Aug. 2020.

[210] Van Bulck, J., Minkin, M., Weisse, O., Genkin, D., Kasikci, B., Piessens, F., Silberstein, M., Wenisch, T. F., Yarom, Y., and Strackx, R. Foreshadow: Extracting the keys to the Intel SGX kingdom with transient out-of-order execution. In *USENIX Security Symposium (USENIX Security)* (2018), pp. 991–1008.

[211] van Schaik, S., Kwong, A., Genkin, D., and Yarom, Y. SGAxe: How SGX fails in practice. sgaxe.com, 2020.

[212] van Vuuren, G. J. AES Golang encryption performance benchmarks updated. *Medium*. https://medium.com/@gerritjvv/aes-golang-encryption-performance-benchmarks-updated-bcfa3555165b, 30 June 2019.

[213] Vickrey, W. Counterspeculation, auctions, and competitive sealed tenders. In *The Journal of Finance* (1961), vol. 17, pp. 8–37.

[214] Vorick, D., and Champine, L. Sia: Simple decentralized storage. https://sia.tech/sia.pdf, 2014.

[215] W3C. Decentralized identifiers (DIDs) v1.0: Core architecture, data model, and representations. *W3C Working Draft*. https://w3c-ccg.github.io/did-spec, 4 Feb. 2021.

[216] Walton-Pocock, T. Aztec: Fast privacy with ZK^2 rollup. *Aztec Blog*. https://medium.com/aztec-protocol/aztec-fast-privacy-with-zk^2-rollup-7c742f45457, 27 Mar. 2020.

[217] Wang, X., Malozemoff, A. J., and Katz, J. EMP-toolkit: Efficient MultiParty computation toolkit. https://github.com/emp-toolkit, [Online; accessed 30 Mar. 2021].

[218] Wang, X., Ranellucci, S., and Katz, J. Global-scale secure multiparty computation. In *ACM Conference on Computer and Communications Security (ACM CCS)* (2017), pp. 39–56.

[219] Weng, C., Yang, K., Katz, J., and Wang, X. Fast, scalable, and communication-efficient zero-knowledge proofs for boolean and arithmetic circuits. Cryptology ePrint Archive, Report 2020/925, 2020. https://eprint.iacr.org/2020/925.

[220] Wesolowski, B. Efficient verifiable delay functions. *Journal of Cryptology* (2020), 1–35.

[221] Whinfrey, C., Fontaine, S., Guido, D., Daian, P., and Breidenbach, L. Failure to set gasLimit appropriately enables abuse. https://drive.google.com/file/d/1mULop1LxHJJy_uzVBdc_xFItN9ck04Jj/view, 19 Nov. 2018.

[222] Wiggins, M. W. Vigilance decrement during a simulated general aviation flight. *Applied Cognitive Psychology 25*, 2 (2011), 229–235.

[223] Wikipedia contributors. Txt record — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/TXT_record, 2020. [Online; accessed 30 Mar. 2021].

[224] Wikipedia contributors. Accelerated mobile pages — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Accelerated_Mobile_Pages, 2021. [Online; accessed 30 Mar. 2021].

[225] Wilkinson, S. Storj: A peer-to-peer cloud storage network (v1.01). https://storj.io/storj2014.pdf, 15 Dec. 2014.

[226] Wilson, D., and Ateniese, G. From pretty good to great: Enhancing PGP using Bitcoin and the blockchain. In *International conference on network and system security (NSS)* (2015), pp. 368–375.

[227] Witkowski, J., and Parkes, D. C. Peer prediction without a common prior. In *ACM Conference on Electronic Commerce (EC)* (2012), pp. 964–981.

[228] Xing, B. C., Shanahan, M., and Leslie-Hurd, R. Intel® Software Guard Extensions (Intel® SGX) software support for dynamic memory allocation inside an enclave. In *Hardware and Architectural Support for Security and Privacy*. 2016, pp. 1–9.

[229] Yasskin, J. Signed HTTP Exchanges. *W3C Internet-Draft*. https://wicg.github.io/webpackage/draft-yasskin-http-origin-signed-responses.html, 27 Jan. 2021.

[230] Yin, M., Malkhi, D., Reiter, M. K., Gueta, G. G., and Abraham, I. Hotstuff: BFT consensus with linearity and responsiveness. In *ACM Symposium on Principles of Distributed Computing (PODC)* (2019), pp. 347–356.

[231] Zamani, M., Movahedi, M., and Raykova, M. Rapidchain: Scaling blockchain via full sharding. In *ACM Conference on Computer and Communications Security (ACM CCS)* (2018), pp. 931–948.

[232] Zhang, F., Cecchetti, E., Croman, K., Juels, A., and Shi, E. Town Crier: An authenticated data feed for smart contracts. In *ACM Conference on Computer and Communications Security (ACM CCS)* (2016), pp. 270–282.

[233] Zhang, F., Maram, S. K. D., Malvai, H., Goldfeder, S., and Juels, A. DECO: Liberating web data using decentralized oracles for TLS. In *ACM Conference on Computer and Communications Security* (2020), pp. 1919–1938.

[234] Zhang, X.-Z., Liu, J.-J., and Xu, Z.-W. Tencent and Facebook data validate Metcalfe' s law. *Journal of Computer Science and Technology 30*, 2 (2015), 246–251.

[235] Zhang, Y., Setty, S., Chen, Q., Zhou, L., and Alvisi, L. Byzantine ordered consensus without Byzantine oligarchy. In *USENIX Symposium on Operating Systems Design and Implementation (OSDI)* (2020).

[236] Zhou, L., Qin, K., Torres, C. F., Le, D. V., and Gervais, A. High-frequency trading on decentralized on-chain exchanges. *arXiv preprint arXiv:2009.14021* (2020).

[237] Zimmermann, P. R. *The official PGP user's guide*. MIT Press Cambridge, 1995.

A 术语

本术语库包含白皮书中关键术语的定义。方括号表示白皮书中提到术语的章节。相应术语用粗体表示

- **经过认证的数据来源（Authenticated data origination）（ADO）** [7.1 章节]：数据源使用数字签名从 API 端认证数据来源。APO 认证数据来源，有助于增强预言机报告的数据完整性。
- **拜占庭容错（BFT）：** 区块链等分布式系统协议中使用的术语。BFT 可以在（少数）节点故意发起恶意攻击的情况下保护系统安全。拜占庭这个词也可以用来表示多数 BFT 协议采用的机制。
- **加密经济机制：** 一种特殊的经济机制，利用加密技术和数字资产在去中心化系统中创建最优平衡。
- **DECO** [3.6.2 章节]：DECO 是去中心化预言机（decentralized oracle）的缩写，是一种新型加密协议，用户（或预言机）可以用零知识证明数据来自某一 HTTPS 服务器。DECO 可以将来自 web 服务器的隐私数据通过预言机网络安全传输至链上，并无须对 web 服务器做更改。（证明者不能将数据直接发送到链上。）DECO 的功能比 Town Crier 少，无须依赖可信执行环境。
- **交易平台：** 交易数字资产的平台。去中心化交易平台（DEX）部分或全部以智能合约形式实现。
- **公允排序服务（FSS）** [第 5 章]：Chainlink 可以为区块链用户带来先到先得的创新排序服务，就像熟食店里的取号机制一样。FSS 可以有效避免抢跑或其他类型的套利交易，防止部分交易者利用自身技术优势在智能合约系统中榨取普通用户价值。
- **抢跑：** 在公链等金融系统中的一种盈利方式。交易者看到进入交易池的交易，并快速将自己的交易排在前面，以不当获取利益。这个词在区块链行业的涵义要比在传统金融行业更广，但是传统金融行业的抢跑现象甚至更加猖獗。

- **未来费用收入机会（FFO）：** Chainlink 预言机在未来服务客户所获得的净收益。由于不当操作损失的未来收入机会是一种机会成本。
- **博弈论：**用数学方式对战略互动进行研究。博弈指的就是一种这样的互动模式。在现实世界中，它规定了博弈中“参与者”可以采取的一系列行动。
- **Layer-2 系统：**为提升区块链交易吞吐量并降低延迟而开发的一类解决方案。Layer-2 系统在链下处理交易，并定期与链上进行同步。Rollup 就是其中一个例子。
- **机制/机制设计（MD）：** MD 有时也被称作“反向博弈论”，是一种设计博弈规则（即机制）的科学，激励“参与者”采取某种行为。
- **公钥基础架构（PKI）：**安全创建并管理公钥身份映射的系统。
- **Rollup**[6.3 章节]：一种计算机批量处理事务的方式，目前实现了多个变种。Rollup 可以提高交易处理速度和吞吐量，并且有望为区块链扩容，处理大量交易。Rollup 包括在链下将交易打包，并随即发送至链上。目前有两种主流的 rollup 方案：
  - *Optimistic rollup:* 这类 rollup 方案可以将打包的交易无需验证直接发送至链上。网络中的参与者可以对错误的 rollup 提出挑战，并通过质押机制为挑战者提供经济激励。
  - *zk-Rollup:* 其中的 zk 指“zero-knowledge”（零知识），这种 rollup 方案采了用简洁的证明（即压缩的加密证明或正确的计算），高效验证发送至链上的数据。
- **超线性/二次方质押影响** [9.4.2 章节]：攻击者可以通过贿赂买通理性的（即受到经济激励驱动的）预言机节点。假设攻击者的预算为 $\$B(n)$，网络中有 n 个参与的预言机节点，每个都质押固定的保证金 $\$d$。如果攻击者成功发起攻击所需的预算 $B(n)$ 增速超过 n，则质押机制实现超线性影响。二次方影响指 $B(n)$ 的增速等于 n^2。
- **权益质押** [第 9 章]：使用严格量化的经济激励增强去中心化系统安全性的方法。在权益质押机制中，参与者质押加密货币作为保证金，如果违反区块链协议，则保证金部分或全部被没收。在 Chainlink 预言机网络中，节点质押保证金，保障预言机报告质量，如果生成错误报告，则被没收保证金。

- **交易执行框架（TEF）** [第 6 章]：针对 layer-2 系统的去中心化的预言机网络框架，支持预言机并实现公允排序服务。
- **Town Crier** [3.6.2 章节]：一种预言机技术，采用可信执行环境（TEE）保障预言机数据和计算的完整性和隐私性。Town Crier 的功能比 DECO 更广，但需要依赖 TEE。
- **可信执行环境（TEE）：** 应用运行的可信执行环境，通常需要特殊硬件支持。TEE 拥有强大的安全属性，比如可以保障数据隐私和完整性，软件无法篡改。TEE 目前还在开发中，但有潜力在行业中实现巨大影响。
- **信任最小化** [第 7 章]：区块链行业常用术语，它的实际意思与字面意思正好相反。信任最小化指通过优化去中心化系统的功能来实现系统中各个参与方的诚实行为，与此同时无需盲目信任系统中的任何实体或要素。

B DON 接口：详解

附件详解了第 3 章提到的 DON 接口。附件 B.1、B.2 和 B.3 分别阐述了三种计算资源，即网络连接、计算和储存。

B.1 网络连接

预言机系统的出现是为了解决基础区块链协议的痛点，即：公链无法从 web 服务器等链下系统获取数据。[20] DON 可以为区块链和链下系统之间提供通用灵活且可扩展的应用层适配器。

这一部分，我们将讨论 DON 构建安全适配器并保障其可用性所采用的技术。我们会提到一些具体的应用案例，其中大部分技术都是在 Chainlink 白皮书中首次正式提出。这些技术可以应用于如今的 Chainlink 基础架构中，也就是说即使没有 DON 也可以实现。我们期待在现有基础架构中实现其中部分技术。

我们基于 CIA（保密性、完整性和可用性）三要素定义了适配器的安全性。

[20] 目前大多数具有价值的数据都无法在数据源 web 服务器签名。因此，单一矿工获取数据并生成区块，很容易被篡改。即使在基于委员会的共识协议中（如 [103, 120]），要实现预言机功能也存在巨大挑战，比如数据可用性低（及其对性能和交易审查方面的影响），增加不必要的复杂性，无法访问一些数据源（如内部数据库）等。

预言机系统中最重要的因素就是数据完整性。这里完整性指攻击者无法篡改从数据源传输至 DON（或反过来从 DON 传输至数据源）的数据。在某些情况下，还需实现数据保密性。保密性指攻击者只能查看协议在通道中传输的那部分数据。（我们提供了更精简的概念。）通道还应该具有可用性，也就是说应及时传输数据，响应用户需求。最后，我们还添加了一个要素，那就是可问责性。这个要素对于保障预言机系统正常运行至关重要：如果预言机系统出现故障，能够甄别（并惩罚）出错的节点。

B.1.1 完整性

传输层安全协议（TLS）是在客户端和主机之间建立安全通道的主流互联网协议，而这个协议存在一个严重的限制：它只针对两方保障数据完整性，无法面向第三方提供数据签名，证明数据来自主机或客户端。而目前提出的解决方案（如 [9, 190]）应用范围非常有限。一些传输到区块链的数据可以在数据源进行数字签名，我们在 7.1 章节讨论了实现这一功能的方式。然而，如今大部分 web 数据都没有签名。

因此，我们在 DON 中创建了特殊机制，保障传输到 DON 的报告 r 经过一个或以上的数据源签名。我们提出了三种方案，每一种都有独特的信任模型和运行模式。

为了阐述这三种方案的运行机制，我们建立了一个简单的模型，各个参与者共同保障预言机报告 $r = (\tilde{r}, \mathsf{sid}, S)$ 的数据完整性，其中 $\tilde{r}$ 的标签包括：(1) 会话识别符sid (2) 数据源识别符 S 。sid 值为发送 $\tilde{r}$ 的协议贴上了独特的标签，其中包括连上智能合约的独特标识符以及报告响应请求的独特标识符等信息。我们将“增强后”的报告统称为报告（r）。

我们用 $\rho_i = (r\,[, w])$ 指代预言机节点 $\mathcal{O}_i$ 写入$\mathcal{L}$ 的报告，其中 w 指代见证者（如有）。我们将 w （无论是否有见证者）作为记录的报告。由于参与者在，即由发布消息的参与者签名，因此写入链上账本的 ρ_i 由 $\mathcal{O}_i$ 背书。

DON 支持三种基本机制，保障数据请求和报告的完整性，即数据来自真实的数据源 S，且未经过篡改。这三种机制分别是：

- **加密证明：** 数据源 S 使用公钥 pk 对 r 签名，即 $\rho = (r, w = \mathsf{Sig}_{\mathsf{pk}}[r])$，可以直接保障消息的完整性。
- **可信执行环境（TEE）/ Town Crier：** 应用实例 A 拥有公钥 pk_A，应用在功能完备的 TEE 中运行，比如英特尔的 SGX。应用可以生成一份证明，即获得数字签名的见证者 $w = \mathsf{Sig}_{\mathsf{pk}_A}[r]$。假设 TEE 可以信任，那么记录的报 $\rho_A = (r, w = \mathsf{Sig}_{\mathsf{pk}_A}[r])$ 可保障 r 的完整性。

- **门限机制：** 任何 $f+1$ 个参与者的子集生成记录报告 $\rho_i = (r, \mathsf{Sig}_{\mathsf{pk}_i}[r])$ 或聚合的见证者 $\rho_{\mathcal{O}} = (r, \mathsf{Sig}_{\mathcal{O}}[r])$，共同认定 r 源自于真实的数据源 S。

门限机制和 TEE 可以互为补充。这两种机制可以同时使用，实现双重安全保障，只要其中一种机制正常运行，就可以保障数据完整性。

我们可以扩展对“完整性”的定义，包括记录 $\tilde{r}$ 的转变 $\mathsf{conf}(\tilde{r})$，而非只记录 $\tilde{r}$ 本身。这个定义扩展有助于实现数据隐私，下文会详细讨论。

不可信的数据源： 任何数据源 S 都有可能提供不可信的数据。我们对适配器完整性的定义并不包括数据源的完整性。预言机系统中采用的主要机制是聚合来自多个数据源的数据。比如，假设资产价格报告 r_1, r_2, r_3 分别从数据源 S_1, S_2, S_3 获取，聚合报告 $r = \mathsf{median}(r_1, r_2, r_3)$（注：取中位数）。如果其中一个数据源出错，$r$ 值将来自正确的数据源。保障正确的聚合方式涉及到计算完整性问题，附件会详细探讨。

B.1.2 隐私性

如果 $\mathcal{O}$ 能查看请求 q 以及适配器创建的报告 r，那么我们就认为适配器的隐私性较弱。然而，隐私性较弱的协议仍可以提供其他形式的隐私保护：适配器可以只向 $\mathcal{L}$ 传输 (q, r) 的部分数据，因此向观察者隐藏了 (q, r) 的部分数据，因此保护数据隐私。

相反，如果适配器能向 $\mathcal{O}$. 部分或全部隐藏 q 和 r 中的数据，我们则认为适配器具有较强的隐私性。乍一看似乎很难解决数据隐私性问题：$\mathcal{O}$ 怎么样才能处理隐藏的请求并管理隐藏的报告呢?

DON 可以使用 DECO [233] 和 Town Crier [233] 有效地在适配器中实现隐私保护。请参见附件中的例子。

DECO 和 Town Crier 旨在为区块链接入链下遗留系统，并保护数据隐私。另一个方式是对数据源的数据签名，并提供隐私保护功能，比如我们还将探索 TLS-N [190] 作为不错补充方案，并谨慎尝试 OpenID Connect [9]，这是基于 OAuth2.0 的身份数据签名协议。

B.1.3 可用性

适配器的可用性取决于它所接入的系统可用性以及所在的 DON 节点可用性。虽然每个适配器采取了不同的可用性机制，但是我们预期所有适配器会统一采用 Chainlink 现有的几项基本技术。其中一个就是冗余备份，即多个节点共同操作通道。另一

项技术是加密经济激励，提高运行速度。比如，虽然所有节点都会收取服务费，但最快将数据从链下系统传输到 DON 的节点将获得一笔奖金。

参见附件 B.1 ，了解关于 DON 外部适配器的具体内容。

B.2 计算

DON 为 layer-2 扩容方案提供支持（无论是 optimistic rollup 还是 zk-rollup），可以大幅提升智能合约的性能。DON 还采用了其他技术，进一步提升智能合约性能，并通过多种方式保护数据隐私，其中包括实现 layer-2 方案的隐私保护（如 [216]），或为其他机制增强隐私保护。

zk-SNARKs [126]、zk-STARKs [38] 以及 Bulletproofs [56, 66] 等零知识证明系统是非常强大的隐私保护工具，并将对智能合约隐私保护功能 [149] 的发展起到关键作用。值得一提的是，向预言机节点提供零知识证明，采用了“指定验证者”的新型零知识证明方案，即证明只能由一个专门的验证者进行验证，而不能公开验证 [219]。此类机制的性能远超普通的零知识证明系统。我们预期零知识证明将对适配器和可执行程序准确保密地传输预言机报告起到关键作用。但零知识证明存在一个重要的问题：证明者可以查看到保密信息。

因此，我们必须采用其他技术进一步增强数据隐私。这些技术同样可以用来实现数据完整性，即保障运算正确执行。在此，我们简要讨论一下可信执行环境（TEE）和安全多方计算（MPC）在 Chainlink 发展中的起到的作用。

B.2.1 可信执行环境（TEE）

英特尔 SGX [228, 133] 和 Keystone [152] 等可信执行环境可以让应用在受保护的环境中执行，这种环境也称 enclave。Enclave 在理论上可以保障数据完整性以及隐私。TEE 旨在实现一个非常强大且通用的功能 [182]，其中甚至包含虚拟黑盒混淆等强大的加密功能，这些功能已证明是无法仅靠密码学技术实现的 [27, 174]。

[139, 232] 提到了将智能合约放入 TEE 运行，以及后续提出的几个通用框架 [58, 80, 99]。

我们预期 TEE 将最终应用到 Chainlink 网络中，因为 TEE 是实现 Town Crier 的基础 [232]，并且 Chainlink 目前已接入基于 TEE 的区块链系统（如 [8]）。随着这项技术不断成熟，我们认为它将在区块链和 Chainlink 网络中扮演越来越关键的角色。

对于 TEE 安全性的担忧一直阻碍着它的大规模应用。因此，下文中我们来简要讨论一下这个问题。

B.2.2 TEE 的安全性

近期对英特尔 SGX 发起的攻击（注：目前英特尔是唯一拥有证明功能的 TEE）[210, 211] 引起了大众对 TEE 安全性的担忧。我们认为这项技术会不断向前发展，特别是在越来越多开源资源的支持下 [152]。

应对方案： 可以在清楚意识到 TEE 弱点的前提下部署 TEE。其中一个方式是部署 TEE 以实现深度防御，即进一步增强系统安全。这里可以用到 Town Crier：可信的预言机运营商应该有能力为客户保障数据隐私和完整性，但使用 Town Crier 可以进一步增强安全保障。同样地，TEE 还可以部署在不要求百分百安全的情况中。比如 Sealed-glass proofs [206] 可以使用 TEE 保障数据完整性，但并不能保障数据隐私性（不过最近的侧链攻击在这两个维度都破防了）。

最后，还可以开发拥有前向安全机制（forward security）的 TEE 协议，如果 TEE 被攻陷：（1）参与者可以避免使用这个 TEE，并且（2）攻击不会影响到之前的协议调用。这个方法将大幅缩小攻击者攻击 TEE 的机会窗口。比如，如果协议中的 TEE 控制用户资金，用户可以将资金转移至临时专门为 TEE 设置的地址。如果协议中的 TEE 要在长期保障数据隐私，定期密钥轮转也可以实现相同的目标。

比如，可以用 TEE 针对附件提到的保密 DeFi 实现 ConfSwitch 适配器。TEE 获取用公钥 $\mathsf{pk}_{TEE}(t)$ 加密的 (q, switch)，假设时间 t 是一周。然后生成结果 $s = \mathsf{switch}(r)$。完成后，TEE 清除所有处理过的数据。在每次新的时间段开始之前，TEE 都会删除旧密钥，创建新密钥，并将其发布到 $\mathcal{L}$ 或 MAINCHAIN 。（可以将可执行程序连通启动器，按时发布。）参与者可以按照协议在任何时候放弃。

当然，在这个模式中，用户可能遭遇零日攻击（即未公布的攻击）。然而，发起零日攻击需要一定成本，即通过发布补丁曝光攻击或使其失效。

B.2.3 B.2.3 安全多方运算（MPC）

MPC 指针对一组节点加密共享的值进行通用计算。MPC 可以实现“黑盒”功能，即基于保密数据 $x_1, \ldots, x_z$ 计算函数 g，参与者只能查看 $g(x_1, \ldots, x_z)$，而无法查看输入函数的数据或计算过程。数据输入可以加密共享，并采用提交机制或使用加密共享的密钥进行加密，以保密方式发送到链上。

虽然通用 MPC（即可以支持任何函数 g 的 MPC）非常灵活，而且近期取得了惊人发展，但相比直接在一个节点上执行计算，其计算和通讯成本非常高。比如，使用 EMP toolkit [217, 218]-等攻击模型中的高级工具的 MPC，为 WAN 中 14 个节点

计算一次 AES 大约需要 20 秒。(在线计算只需要 250 毫秒，其他都可以在预计算节点执行。) 相比之下，用目前商用级的 CPU 计算 AES 只需大约 2 纳。 [212]。

简而言之，通用 MPC 在某些情况下是可行的方案，但只能用于执行简单的计算。可行的方案普遍针对相对简单的计算任务，比如 Hyperledger Fabric 中的拍卖 [42]。

某些函数 g 可以采用更高效的协议。比如门限签名，这个方案对 BLS 签名尤其适用 [54]。2PC 是一种包含两个参与者的计算方式，利用特殊的技术（乱码电路 garbled circuits）使计算效率远高于 $n > 2$ 个参与者比如，DECO [233] 就采用了定制化和通用的 2PC 技术。

多数 MPC 机制和实现都存在不可忽视的限制。其中一个就是任何一方可以放弃协议，并查看自己的数据输出，同时匿名阻止其他参与者查看他们的数据输出。这不仅影响了计算的稳健性，还使 MPC 的公平性大打折扣，即要么所有参与者都收到收据输出，要么没有一个参与者收到。

有些机制允许识别放弃的参与者的身份（ID-MPC）(如 [37])，是一种解决问题的思路。但是 MPC 协议的运行前提还包括网络可以正常同步，如果无法同步，就无法实现数据隐私和完整性。 [157] 中提到了一个相对比较稳健的 MPC 方案，可以在异步网络中运行。

大多数 MPC 协议存在的另一个问题是无法公开地进行验证。如果诚实参与者达到了法定人数（quorum），不会造成问题，但如果是针对 7.3 和 8.3 章节中提到的 MAINCHAIN 可问责性，则会造成问题。为了保障 MPC 在大多数节点失效的情况下也能正确输出结果，就需要 [37] 这种可公开验证的方案，因此并不十分可行。

综上所述，我们预期 DON 在早期发展阶段将支持 MPC 展开门限签名和 DECO 等特殊的计算任务。我们将密切关注 MPC 的发展，明确如何更好地支持通用 MPC 的开发。

B.3 存储

传入区块链的数据可以发挥各种作用。数据可以立即传入链上智能合约，也可以存入区块链，以便未来的智能合约能够快速访问，或实现与合约执行不相关的目的，如为了链下系统审计而存档。后面两种目的所需的数据量远超智能合约所需的数据量，以及多数公链的数据存储能力。

我们预期 DON 将实现两种存储功能。其一是在账本中储存，实现方法是在 DON 中创建一个账本。其二是账本外储存，这个方案对大量数据存储非常具有吸引力。

账本外储存可能需要接入 Filecoin [187]，IPFS [40]，Storj [225] 或 Sia [214] 等去

中心化的链下系统。Filecoin 采用的协议是复制证明 (PoReps) [74, 112, 111, 187, 198]，目的是保障一份文件 F 储存了多个副本（并避免矿工操纵共识协议）。另外三个链下系统要实现这个功能，只能让终端用户用不同密钥多次加密 F，创建相应的秘文 $\tilde{F}_1$、$\tilde{F}_2$ 和 $\tilde{F}_3$。DON 可以代表用户进行加密，无需存储协议原生支持就可生成复制证明。

在与用户的信任模型兼容的情况下，还可以使用云平台等现有系统。多份复制可以增强服务稳健性 [57]，并可与去中心化存储服务形成互补。

隐私性： DON 中的 $\mathcal{O}$ 可以储存采用加密共享密钥加密的数据。正如附件 B.2 所述，DON 可以对这些数据开展保密运算。

B.4 资源定价

向使用区块链资源的用户精准定价仍然是重大挑战。资源被错误定价，不仅有可能劝退生态中真正有价值的应用，还有可能产生安全风险，因为可能会导致 DoS 攻击 [183] 以及资金盗窃等更隐晦的攻击 [221]。为 DON 中的可执行程序资源定价相对比较容易，因为 DON 节点控制着资源用量，并可以灵活应用所需的定价策略。

可行的定价策略包括基于市场定价、效仿以太坊的 gas 计价（EVM 和 WSAM [3] 中可以实现）以非图灵完备语言等基于语言的定价 [131]（还可以加入代码量这个维度）。然而，为适配器定价就要视具体应用而定，因为适配器接入的是云平台等链下资源，费用取决于具体的 DON。

C 适配器案例

接下来，我们举几个适配器的具体案例，详细阐述适配器在应用中可以实现的丰富功能，以及相关的安全概念和技术。

我们用 $\mathsf{Adapter}(\mathcal{P}, \mathcal{S})$ 表示适配器，其中 $\mathcal{P}$ 为参与者，$\mathcal{S}$ 为一组数据源（或一个数据源）。在概念模糊的情况下，我们用 $\to T$ 表示向系统 T 输入或输出的数据。其他情况下，数据输入应理解为适配器从 $\mathcal{L}$ 读取的数据，数据输出为适配器写入 $\mathcal{L}$ 的数据。

C.1 预言机作为适配器访问数据源（MediatedReport）

在某些情况下，部分节点可能可以访问设置了访问权限的数据源 S，整组节点 $\mathcal{O}$ 无法访问。比如，部分节点可能有 API 密钥或其他证书，而 O 中的其他节点则没有。

拥有权限可以访问数据源 S 的节点可以使用 DECO 或 Town Crier 代表 $\mathcal{O}$ 访问。这里案例中的适配器保密性较弱。

技术方案： 有了 DECO，$\mathcal{O}_i$ 可以使用保密证书访问 S，并向其他 S 点 $\mathcal{O}_i$ 证明它从 S 正确传输了报告 r，并且无需向 $\mathcal{O}_i$ 披露证书内容。这个方案也可以扩展至一组节点。

适配器 MediatedReport$(\mathcal{O}, S)$

输入： q

输出： $\langle \texttt{medreport} : (q, r);\ \mathcal{O}_i \rangle$

安全性：

- 可用性：$\mathcal{O}$ 无法访问 S，当 $\mathcal{O}$ 中参与者发起调用时，$\mathcal{O}_i$ 必须是可用。
- 完整性：在 DECO 的加密硬度假设中，r 是正确的，并且假设 $\mathcal{O}$ 中至少有 k 个诚实节点。

具体用例： 假设 $\mathcal{O}_i$ 从数据源 S 提交一份报告 r，$\mathcal{O}$. 中的另一个节点或多个节点对 r 值提出质疑。这种情况可以在权益质押协议的分歧机制中出现。

解决分歧的方法可以是调用 $\mathcal{O}$ 中的全部节点。然而，$\mathcal{O}$ 中的部分节点可能无法访问 S，因为 S 可能设置了权限。有了 DECO，就可以向节点证明报告 r 准确无误地从 S 传输，同时无需授权其他节点直接访问 S。

其他改进和扩展方案： Town Crier 为实现同样的适配器功能提供了另一种方案。$\mathcal{O}_i$ 的证书可以放到 $\mathcal{O}$ 单个预言机节点运行的 Town Crier 实例中，并对其应用设置适当的限制 [163]。

另外还可以对 MediatedReport 进行扩展，让 $\mathcal{O}$ 可以调用 $\mathcal{O}^*$ 中的任何节点，$\mathcal{O}^*$ 是可以访问 S 的节点子集。另外，还可以扩展 MediatedReport，涵盖 MAINCHAIN 上的节点。这类方案会对协议比较有用，如权益质押或 optimistic rollups,（如 [141]）等协议，在链上裁决链下计算。

C.2 跨账本报告（XL-Report-Read）

跨越不同 DON 交互数据可以为许多应用带来价值。这个案例中是一个简单的适配器，从 DON B 导出基础报告并导入 DON A。为了简化描述，我们假设 $\mathcal{L}_B$ 上有一份独特的报告 $\langle \texttt{report} : (q, r) \rangle$，其中 q 为任意值。

技术方案： 由于 $\mathcal{L}_B$ 上的消息采用 $\mathsf{pk}_{\mathcal{L}_B}$ 签名，因此，去中心化预言机网络 $\mathcal{O}_A$ 中的预言机节点如果知道 $\mathsf{pk}_{\mathcal{L}_B}$ 就可以轻松对消息进行验证。

适配器 XL-Report-Read$(\mathcal{O}_A, \mathcal{L}_B)$

输入： $(q, \mathsf{pk}_{\mathcal{L}_B}) \to \mathcal{L}_A$，$q$ 为请求，$\mathsf{pk}_{\mathcal{L}_B}$ 为目标 DON 的公钥。

输出： $\langle \texttt{XLreport} : (q, r); \mathsf{pk}_{\mathcal{L}_B} \rangle \to \mathcal{L}_A$.

安全性：

- 可用性：$\mathcal{O}_A$ 中至少有一个节点要能够访问 $\mathcal{L}_B$。
- 完整性：r 的正确性取决于 $\mathcal{O}_B$ 是否诚实。

具体用例： DON A 可以判断它是否充分信任 DON B 及其输出的报告。在这种情况下，成本效益更高的方式是导入部分报告，而非自己生成。

PKI 支持： 在 DON 之间安全传输报告的主要挑战是如何维护底层的公钥基础架构（PKI），即可靠地映射 DON 的最新公钥。为了在 Chainlink 中实现这个关键功能，我们计划将去中心化预言机网络及其在多有接入的区块链上的公钥都放在智能合约的目录中维护。

其他改进和扩展方案： 当然，还可以用其他方式实现跨 DON 适配器。比如，DON A 有权限访问 DON B，从 $\mathcal{L}_B$ 导入隐私数据。这种情况下，适配器就需要 $\mathcal{O}_B$ 参与。另一个例子是，DON A 有部分权限写入 DON B。这种情况下，就可以创建跨链适配器写入数据。

还可以通过其他扩展方案处理 $\mathcal{L}_B$ 上多个或模糊的报告。比如，XL-Report-Read 可以传输消息 $M = \Big((q, *), z\Big)$，z 为任意时间 $\mathcal{L}_B$ 中最高的指数。

适配器 ConfSwitch $(\mathcal{O}, \mathcal{S})$

输入： $\mathsf{Enc}_{\mathsf{pk}_{\mathcal{O}}}[q, \mathsf{switch}], \mathsf{Enc}_{\mathsf{pk}_{\mathsf{aud}}}[q, \mathsf{switch}, \alpha]$

输出： $\langle \texttt{ConfSwitch} : \mathsf{switch}(r) \rangle$, for $r = \mathsf{median}(\{r_i\}_{i:\mathcal{O}_i \in \mathcal{O}})$

安全性：

- 隐私性：适配器向 $\mathcal{O}$ 披露 Mixicle 执行细节，但只向 $\mathcal{L}$ 和 MAINCHAIN 上的观察者透露 $\mathsf{switch}(r)$。
- 可用性：可用性取决于 $\mathcal{S}$ 的可用性（不过也可以选择依赖 $\mathcal{S}$ 的一组门限子集。
- 完整性：r 的正确性取决于加密硬度假设、$\mathcal{O}$ 是否诚实以及 $\mathcal{S}$ 报告的准确性。

图 19: 适配器 ConfSwitch

C.3 保密开关（ConfSwitch ）

之前两个例子都没有在账本中实现 q 或 r 的隐私性。接下来的例子则可以实现隐私性：ConfSwitch。这个适配器只在链上披露一个 switch 函数，这个函数聚合了报告 r 的中位数，因此可以作为喂价。这个功能为在链上保护了 q 和 r 的隐私，只披露了 r 的函数。另外，它还为审计者保留了隐私数据记录（用公钥 $\mathsf{pk}_{\mathsf{aud}}$）查看）。这种审计很重要，因为使用 ConfSwitch 的交易内容无法在账本或主链上查看。审计人员可以查看的数据包括 q、switch 以及辅助数据 α（可选）。

技术方案： 为了保护 q 和 switch（以及 α）的隐私，这些数据可以用 $\mathcal{O}$ 和审计方的公钥加密。$\mathcal{O}$ 在账本外聚合报告，聚合至报告 r 中，并 $\mathcal{L}$ 中不直接写入 q 或任何报告数据。

强大的隐私保护： 使用 DECO 或 Town Crier 可以在 ConfSwitch 中实现强大的隐私保护。

其中，参与者向 $\mathcal{L}$ 发送 $c = \mathsf{comm}(q, \mathsf{switch})$。然后，其中任何一个参与者使用 DECO 或 Town Crier 收到报告，并通过零知识证明确认报告与 c 一致。由于 DECO

只能为单个报告生成证明，因此还需要额外一个步骤：用户需要证明单个报告中密文是正确无误的，然后再证明结果 $\mathsf{switch}(r)$ 是准确的，即正确提取了中位数，并将 switch 应用在报告中。

D 函数签名

函数（数字）签名机制 [59]，与传统签名一样，都使用一个主密钥（私钥）sk 对任何消息签名。然而，除此之外，主密钥还可以用来创建一个签名密钥 sk_F，其中 F 为函数，可以对 F 中的任何值 y 进行数字签名（即函数 $F(x) = y$ 中输入值 x 得出的结果 y）。

实现这个机制的一个简单方法 [59] 如下（我们做了一点修改）。新建一个密钥对 $(\mathsf{sk}', \mathsf{pk}')$。计算函数 F，创建密钥 sk_F：

$$\mathsf{sk}_F = \Big(\mathsf{sk}', \mathsf{cert} = \mathsf{Sig}_{\mathsf{sk}}(\mathsf{pk}' \parallel F)\Big),$$

其中 cert 代表 “certificate”（证书），表示 pk' 是使用函数 F 签名的有效公钥，也是主公钥 pk。

因此，可以通过以下方式用 sk_F 对消息 y 签名：

$$\mathsf{Sig}^*_{\mathsf{sk}_F}(y) = (\mathsf{cert}, \mathsf{Sig}_{\mathsf{sk}'}(y), x).$$

如需验证 $\mathsf{Sig}^*_{\mathsf{sk}_F}(y)$（即计算 $\mathsf{verify}^*\Big(\mathsf{Sig}^*_{\mathsf{sk}_F}(y)\Big)$，验证者要验证 $F(x) = y$，以及 cert 和 $\mathsf{Sig}_{\mathsf{sk}'}(y)$ 的有效性。

x 值（即证人）理论上可以是很大的（比如预言机相关的应用场景）。对验证者来说，一个在计算上更高效的方案 [59] 是用 SNARK（或 zk-SNARK 以实现保密性）替代 x，SNARK 是 x 的简洁证明。

函数签名在概念上简单直接，基本构成也非常直观。但是它为预言机网络聚合数据提供了有效的框架。

D.1 聚合数据的函数签名

案例中展示了如何利用函数签名创建证明 σ，证明值 v 是 (V, Σ) 正确的计算结果。我们用 $(\mathsf{sk}_i, \mathsf{pk}_i)$ 指代数据源 $\mathcal{S} = \{S_1, S_2, \ldots, S_{n_S}\}$ 中 S_i 的签名密钥对。假设 pk 是 $\mathcal{S}$ 的公钥，对应的私钥在数据源之间门限共享。为了简化描述，我们假设 sk_i 是 S_i 持有的部分 sk。

然后可以定义取 n_S 中位数的函数签名为有效签名的值：

$$F(V,\Sigma)=\left\{\begin{array}{ll}\mathsf{median}(V) & \text{if } \forall i\in\{1\dots n_S\},\mathsf{verify}(\mathsf{pk}_i,\sigma_i,v_i)=\texttt{true}\\ \perp & \text{otherwise.}\end{array}\right\}$$

有主密钥 sk 的函数签名机制，可以传输正确的中位数值 v 到 SC。预言机网络计算 $\sigma^*=\mathsf{Sig}^*_{\mathsf{sk}_F}(v)$，然后发送至 SC，计算 $\mathsf{verify}^*(\sigma^*,\mathsf{pk},v)$。

当然，为了更有效地应对数据源失效，F 可以设置成取至少 k_S 个有效签名值子集的中位数，门限为 $k_S\le n_S$，即从法定数量的数据源获得有效签名，下方案例中详述。

D.2 离散函数签名

使用 SNARK 函数签名计算 σ^* 需要耗费大量计算资源，可能无法支持高性能的 DON 应用。因此，我们提议采用离散函数签名机制解决这一问题，这个机制的关键是将可能签名的值限定在固定网格内。比如，如果所需的答案是一个单标量，那么网格里包含网格大小 a 的整数倍数 $a\in\mathbb{R}^+$，$a\mathbb{Z}=\{\dots,-3a,-2a,-a,0,a,2a,3a,\dots\}$。有效的签名值是对 az 的表示加上在 az 的有效签名，其中 $z\in\mathbb{Z}$。

值 az 不用是 $F(V,\Sigma)$ 的精简答案，但也比较接近了。除了附件 D.1 中的 predicate 函数 F 以外，离散函数签名还有一对参数 (δ,δ')，具体根据网格的粗细程度而定。具体如下：

- *参数* δ 限制了与 F 正确结果的最大偏差值。[21] 假设 (V,Σ)，对方签名者（即 DON）创建的签名 σ^* 与 $F(V,\Sigma)$ 正确结果的偏差不能超过 δ，即：

$$\mathsf{verify}^*(\sigma^*,\mathsf{pk},v)\bigwedge(|F(V,\Sigma)-v|>\delta)\,.$$

- *参数* δ' 表明 V 的值需要匹配到什么程度才能保障签名。如果满足下列条件：有 k_S 个签名者 $\{S_{z_i}\}_{i=1}^{k_S}$，观察结果 $\{v_{z_i}\}$，因此 $|v_{z_i}-v_{z_j}|\le\delta'$，$i$ 和 j 为任意值，总是能构建函数签名 σ^*。

因此，δ 可以大致看作是稳健性参数，而 δ' 则是完整性参数。也就是说，签名总是可以基于诚实节点的观察结果 $\mathbf{v}=(v_1,\dots,v_{k_S})$，前提是 $\mathbf{v}$ 与 V^{k_S} 空间对角线上的 δ' 一致。即 δ' 越大，可签名观察结果的集合就越大。

[21] 在更正式的安全定义中，需要在特定消息攻击（EUF-CMA）下对存在的不可逆性应用“离散”属性，具体请参照 [118]。

我们之前假设 sk_i 是 sk 的私钥共享。在例子中，我们进一步假设 Sig 是不具有交互功能的门限签名，如：BLS 签名 [54]。下方例子中，我们使用 (k_S, n_S) 门限签名，其中 $k_S \leq n_S$ ，因此 $k_S > \frac{2}{3} n_S$。

在这个机制下，我们可以针对各种不同的消息空间建立高效的函数签名。简而言之，核心理念是让数据源在各自的 v_i 空间中同时（但互相不产生交互）针对网格值 $az_i \in a\mathbb{Z}$ 计算部分 (k_S, n_S) 门限签名。这样，$\mathcal{S}$ 中的数据源共同约束了 v，但并没有直接对其开展计算。如果至少 k_S 个数据源提交了值 v_i，并且足够接近正确值，则它们就可以共同在 δ 限定的范围内创建一个门限签名，并且 $v = F(V, \Sigma)$. 有效。$v = az$ 的签名 $\mathsf{Sig}^*_{\mathsf{sk}}(v)$ 仅仅是 $az \in a\mathbb{Z}$ 的门限签名。

我们在下方示例中展示了如何取中位数。在这里，可以轻松将 $a\mathbb{Z}$ 的单元格定义为一对连续的网格点中所包含的数值区间，即 $[az, a(z+1)]$。

Example 6 (离散函数签名取中位数). 假设要签名的值为喂价（如：美元对太币的价格），可以在 $\mathbb{Z}^+$ 中表示成正整数（如用美分作单位）。函数 $F(V, \Sigma)$ 在至少 k_S 个签名的数值中取中位数，下面的结构实现了 $(\delta = a, \delta' = a)$，其中任何 $a \in \mathbb{N}$。

假设

$$\begin{aligned} \iota(v_i) &= a\lfloor v_i/a \rfloor, \\ \iota'(v_i) &= \iota(v_i) + a, \end{aligned}$$

即 $[\iota(v_i), \iota'(v_i)]$ 是网格 $a\mathbb{Z}$ 中最高的单元格。每个数据源 S_i 按以下方式提交签名 σ_i:

$$\sigma_i = (\sigma_i^{(1)}, \sigma_i^{(2)}, \sigma^{(3)}) = \Big(\mathsf{Sig}_{\mathsf{sk}_i}\big(\iota(v_i)\big), \mathsf{Sig}_{\mathsf{sk}_i}\big(\iota(v_i)\big), \mathsf{Sig}_{\mathsf{sk}_i}\big(v_i\big)\Big)$$

如果所有签名 $\sigma_i^{(j)}$ 都有效，且 $\sigma_i^{(1)}$ 和 $\sigma_i^{(2)}$ 中的消息等于 $\iota(v_i), \iota'(v_i)$，其中 v_i 是 $\sigma_i^{(3)}$ 中的消息，[22]签名 σ_i 才有效。

[22] $\sigma_i^{(3)} = \mathsf{Sig}_{\mathsf{sk}_i}(v_i)$ 并非保障正确性的必要条件，但它的存在在以下情况下可以解释观察到的中位数是什么：

- 不诚实的预言机 S_i 可以证明任意的网格单元边界，而不考虑其观察到的值 v_i ，甚至根本就没有观察到任何值 v_i ，或者
- 部分签名的聚合器可以选择一个任意 k_S 大小的 σ_i 子集，从中构建功能签名。

要优化这个结构，只需用 $\Sigma = \{(\sigma_i^{(1)}, \sigma_i^{(2)})\}$，并且表明，只要少于 $\frac{n_S}{3}$ 的答案是不诚实的，由此产生的函数签名值的中位数的可能范围就包括所有可能的实际中位数值。它需要将不准确报告的/无意义的/不存在的 V 值重新分配给任意的整数，虽然并不困难，但是比较麻烦。

如果有共同的网格点 $v = az \in a\mathbb{Z}$，并且其中有足够大的子集 $\{\sigma_i^{(1)}, \sigma_i^{(2)}\}_i$ 拥有部分签名，则可以基于 $\{\sigma_i\}_i$ 生成 $\mathsf{Sig}^*_{\mathsf{sk}}(v) = F(\{v_i\}, \{\sigma_i\})$

案例六中的机制中位数函数 $F(V, \Sigma)$ 实现了 $\delta = a$ 以及 $\delta' = a$ ，函数在至少 k_S 个签名的值中取中位数。下方引理中将详细阐述。

Lemma 1. 假设针对 $v \in a\mathbb{Z}$ 存在有效的离散函数签名，签名基于 v 的 k_S 个部分签名构建，

$$\left\{\sigma_i = (\mathsf{Sig}_{\mathsf{sk}_i}(\iota(v_i)), \mathsf{Sig}_{\mathsf{sk}_i}(\iota'(v_i)), \mathsf{Sig}_{\mathsf{sk}_i}(v_i))\right\}_{i=1}^{k_S},$$

每个 v_i 都在 $[v-a, v+a]$ 中。

证明. 由于 $\{\sigma_i\}$ 可以用于构建 v 的函数签名，每个 i 都必须符合 $v \in \{\iota(v_i), \iota'(v_i)\}$ 。因此，下列中有一个是真实的：

$$\begin{aligned} v = \iota(v_i) &= a\lfloor v_i/a \rfloor &&\implies v_i \in [v, v+a] \\ v = \iota'(v_i) &= a\lfloor v_i/a \rfloor + a &&\implies v_i \in [v-a, v] \end{aligned}$$

因此，$v_i \in [v-a, v] \cup [v, v+a] = [v-a, v+a]$。 □

Lemma 2. 假设 $\{v_i\}_{i=1}^m$ 是一个观察合集，$I \subset \{1, \ldots, m\}$ 是一个指数的子集，它构成了大多数。$\{v_i\}$ 的中位数在 $[\min(\{v_i\}_{i\in I}), \max(\{v_i\}_{i\in I})]$ 中。

证明. 在不丧失一般性的情况下，假设观察值 $v_1, \ldots, v_m$ 是被排序的。子序列

$$\arg\min_{i\in I} v_i, \ldots, \arg\max_{i\in I} v_i$$

比 $\frac{m}{2}$ 长，因为 $|I| > \frac{m}{2}$。因此，序列 $1, \ldots, m$ 的中点[23]位于子序列中，因此中位数于 $[\min(\{v_i\}_{i\in I}), \max(\{v_i\}_{i\in I})]$ 内。 □

Lemma 3. 给定一个有效的 $v \in a\mathbb{Z}$ 的离散函数签名，来自一组 n_S 签名者 $\{S_i\}$，其中三分之二以上是诚实的，以及一个签名阈值 $k_S > \frac{2}{3}n_S$，真正的中位数位于 $[v-a, v+a]$ 内。

证明. 根据引理一，诚实的签名者使用的所有 v_i 都位于 $[v-a, v+a]$ 中。虽然不诚实的签名者可以根据任何观察（甚至没有观察）在 v 上签名，但诚实的参与者构成了签名者中的大多数，因此根据引理二，观察到的中位数位于 $[v-a, v+a]$ 中的两个诚实值之间，因此其本身也位于 $[v-a, v+a]$ 中。（诚实的签名者占大多数，因为至少 $k_S > \frac{2}{3}n_S$ 个节点签名了，但是签名集合中的不诚实节点应该少于 $\frac{1}{3}n_S = \frac{1}{2} \times \frac{2}{3}n_S < \frac{k_S}{2}$） □

[23]如果 m 是偶数，则有两个中点，$\frac{m}{2}$ 和 $\frac{m}{2}+1$，如果 m 是奇数，则有一个中点，$\frac{m}{2}+1$。

Lemma 4. *例 6 中的离散函数签名机制为取中位数函数实现了 $\delta = a$。*

证明. 根据引理三，真正的中位数位于 $[v-a, v+a]$。这个区间内的所有点与 v 的距离都小于 a 个单位，所以 v 到真正的中位数的距离小于 a。也就是说，我们希望的 δ 定义中的条件是不可行的，

$$\mathsf{verify}^*(\sigma^*, \mathsf{pk}, v) \bigwedge (|F(V,\Sigma) - v| > \delta)\,.$$

实际上 $\delta = a$ 确实是不可行的。 □

Lemma 5. *例 6 中的离散函数签名方案实现了 $\delta' = a$，即给定一个至少由 k_S 个签名者组成的法定人数 $Q = \{S_i\}$，观察值 $|v_i - v_j| < a$，$S_i, S_j \in Q$，可以基于签名 $\{\sigma_i = (\mathsf{Sig}_{\mathsf{sk}_i}(\iota(v_i)), \mathsf{Sig}_{\mathsf{sk}_i}(\iota'(v_i)), \mathsf{Sig}_{\mathsf{sk}_i}(v_i))\}$ 构建离散函数签名。*

证明. 让 $I = [\min(\{v_i\}_i), \max(\{v_i\}_i)]$ 是包含 $\{v_i\}_i$ 的最小区间。由于 I 的长度，$|\max(\{v_i\}_i) - \min(\{v_i\}_i)|$ 是 $\{v_i\}$ 的两个元素之间的差，因此它小于 a。因此，I 最多包含 $a\mathbb{Z}$ 的一个点。如果它包含 v，那么 $v \in \{\iota(v_i), \iota'(v_i)\}$ 覆盖所有 I，因为每个 v_i 都位于 v 下面或上面的单元格中。因此，可以基于 σ_i 的函数签名构建一个关于 v 的函数签名。如果不是这样，它就完全位于 $a\mathbb{Z}$ 的一个单元内，而且 $\{\iota(v_i), \iota'(v_i)\}$ 对所有 i 来说都是一样的，所以可以在任一边界值上构建一个函数签名。 □

例 6 中的技术可以被看作在网格中离散化数值，将接近性测试简化为平等性测试的一种手段。就可以实现更大的灵活性（即 δ' 更大）。

丰富离散函数签名： 正如这里所描述的，一个签名 $\mathsf{Sig}^*_{\mathsf{sk}}(v)$ 是在一个网格点 $v \in a\mathbb{Z}$ 上，而 v 将接近（但不一定等于）在一组至少由 k_S 节点提供的数值上计算出的中位数 v_{med}。然而，DON 也有可能在其签名消息中包含 v_{med} 。如果 v_{med} 是由 DON 计算出来的，那么假设 DON 可以正常运行，它就是一个值得信赖的值。在一些情况下，智能合约可能希望在 $|v - v_{\mathsf{med}}| \leq a$, 的情况下使用 v_{med}，用 v 来抵御 v_{med} 出现严重偏差的风险。

E 预期性贿赂

我们的权益质押机制明确针对的一类攻击行为就是预期性贿赂（prospective bribery）。这种新型的攻击模式在 [165] 中描述的贿赂攻击模式上进行了扩展。PoW、PoS 以及无须许可的系统中都可能出现预期性贿赂，而且会影响网络安全。

在 PoW 和 PoS 区块链上，预期性贿赂攻击者会承诺给未来区块创建者一笔具体金额的贿赂，前提条件是区块满足一定预设条件。比如，贿赂者要求未来产生的区块中不包括某笔交易，或按照某个具体的顺序排序交易。

我们用“预期性”（prospective）这个词，是因为贿赂者实际上也不知道哪个节点在那一轮会被选中。然而，只要被选中的节点知道贿赂的存在，就有可能被成功收买。

为了帮助大家在权益质押机制的背景下理解预期性贿赂的影响，我们可以假设为一个预言机网络建立了权益质押机制，网络中有 n 个预言机节点，每一轮中，一个节点随机被选中（使用最新的随机数），报告下一个值。这个方案看似可以抵御贿赂攻击，因为贿赂者无法提前得知应该贿赂哪个节点。然而，贿赂者可以先向任何被选中的节点保证支付贿赂金，只有当区块返回 false 才付款（引用第 9 章的说法）。

贿赂者可以使用类似 [165] 中的智能合约来实现预期性贿赂。

F 随机选择预言机 VS 基于委员会选择预言机

预言机网络可以在公链或类公链环境中运行，随机选择子委员会 $\tilde{\mathcal{O}} \subset \mathcal{O}$ ，生成报告 r 或一系列报告。这个方法的最大优势是可以防止节点遭遇技术或贿赂攻击。假设 $\tilde{\mathcal{O}}$ 事先未知，攻击者理论上必须控制接近多半数的 $\mathcal{O}$, 而非只控制 $\tilde{\mathcal{O}}$，才能篡改报告 r。

有许多混合型共识协议采用了这个机制，如 [120, 181]，一些预言机协议也采用了这个机制，如 [25]。

然而，我们认为在许多应用中，固定的预言机节点（或节点委员会）还是有一定优势的，原因如下：

- 节点贿赂方式多样化：理性节点的可靠性基于各种激励因素，其范围超出了所质押的资产，如未来潜在收入和声誉等，第 9 章详细进行了论述。用户可能更希望选择它们认为比较可信的节点，而不是随机选择节点。即使与质押加权的随机选择机制相比，这个方案也可以更可靠地选出节点委员会，即委员会更不容易被收买。

- 节点贿赂门限：还有一个相关问题，就是除非 n 和 k 都非常大，或者问题节点占比很小，否则随机选择的 $\tilde{\mathcal{O}}$ 子集中很有可能大部分节点都在某一时刻被收买。

比如，假设 $n = 100$, $f = 10$，即只有 10% 的节点被收买，统一随机选择 $|\tilde{\mathcal{O}}| = 10$ 的子委员会。那么子委员会中大多数节点被收买的概率 ≈ 0.000009。即使系统吞吐量较低，比如每秒处理一个请求，如果为每个任务选出一个新的委员会，那么预期一天后就会出现委员会被买通的情况。

因此，假设问题节点占比 $f/n = 1/4$，并设置了一定安全参数，[41] 中的随机委员会选择机制所要求的委员会节点数量要达到 26,091 个！

因此，我们认为随机选择节点委员会不足以在未来保障 $\mathcal{O}$ 的安全性和可靠性，这个问题对我们要展开讲的下一个点非常重要。

- 隐私性：用户要访问预言机通过门限方式储存的数据，就必须通过 $\mathsf{sk}_{\mathcal{L}}$ 与节点交互。可以对动态变化的委员会节点发放 $\mathsf{sk}_{\mathcal{L}}$。委员会节点频繁变更除了会大幅增加沟通成本外，还可能导致一些问题，因为这提高了委员会节点同时出现 $f \geq k$ 的情况，并导致上文中提到的严重后果。

- 预期性贿赂：正如第 9 章所述，攻击者可以在选出委员会之前就向委员会中的节点提出贿赂，贿赂只有在节点被选中后才生效。也就是说，节点可以证明自己被选中，并且生成一份证明，以领取贿赂金。智能合约可以用 [164] 中提到的技术实现这个功能。

- 创新：预言机提供商可以在性能、准确性或特殊功能（如保密性或访问付费数据）等各个维度拥有不同优势。让用户自行选择预言机，而不是从同一个池子中随机选出同质化的预言机，这样可以促进创新发展。

- 定制化：不同用户可能对于预言机的可靠性、性能和成本有不同的需求和权衡。因此用户应该根据自己的需求灵活选择预言机。