星云开发者激励协议紫皮书

星云研究院

2018 年 10 月
版本号:1.0.0

目录

1 概要 1

2 背景介绍 3

2.1 DApp 开发者激励 3

2.2 星云指数 4

2.3 投票机制 5

3 开发者激励模型 5

3.1 模型表示 6

3.2 投票行为 7

3.3 采样时间 8

4 开发者激励协议 9

4.1 投票效用与分贡献值的计算 9

4.2 排名分的计算 11

4.3 最终奖励的计算 11

5 性质分析 12

5.1 收买用户 12

5.2 恶意分拆 14

5.3 女巫攻击 15

6 开发者激励协议的实现 15

6.1 如何发放奖励? 15

6.2 开发者激励协议的更新 16

7 未来工作 16

7.1 多维投票行为 16

7.2 DApp 之间的调用 16

附录 A 证明 20

A.1 特征 3 证明 20

A.2 特征 4 证明 20

A.3 推论 2 证明 23

A.4 性质 5 证明 23

附录 B 修订记录 24

B.1 2019.4.8 修订 24

1 概要

一般而言，在传统的软件开发行业，开发者在开发平台（如 Windows[1]，Linux[2]，macOS[3], iOS[4]，Android[5]等）上开发相应的应用，并由此获利。开发者获利的方式主要包括薪资、售卖个人开发的软件以及在免费分发的软件中插入广告流量等。

然而，现有利益分配模式下，应用平台方在软件的使用中所获得的利益并没有被公平的分配给相应的开发者。例如，某用户需要使用 Sketch 而不得不购买搭载 macOS 的 Mac 设备[6]，并由此付出费用，尽管 Sketch 的开发者可以向用户收取使用费用，但是通过售卖 Mac 设备而获利的 Apple 公司[7]并不会因此额外向 Sketch 的开发者分享利益；与之相类似，某用户需要使用 AutoCAD[8]软件时，只能选择搭载 macOS 或 Windows 的设备，由此需要支付给 Apple 或 Microsoft[9]及相关企业的费用亦并未向 AutoCAD 的开发者分享。用户选择一个平台的原因多种多样，其中一个关键性的影响因素是平台所支持的应用，也就是说，一个应用平台的发展，离不开其上的优秀应用。基于以上考量，应用平台方无视开发者利益的做法，一定程度上侵害了开发者的利益。

在区块链领域，去中心化应用（Decentralized Application，DApp）开发者的利益同样被漠视。2014 年，以太坊提出在区块链上运行图灵完备的智能合约（Smart Contract），使得区块链从单纯的数字货币支付网络升级为了去中心化应用平台。然而，DApp 开发者的获利方式与传统的软件开发行业相比，并无明显区别，DApp 开发者并不能从去中心化应用平台或区块链系统的发展中获利。

可以抽象的认为，区块链中出块奖励代表了区块链系统发展中新增的价值，而出块奖励的分发决定着去中心化系统的激励方向。我们认为，区块链系统新增的价值本质上来源于新增的用户数据所蕴含的价值，这些新增的价值应该公平的分发给为系统的新增价值做出贡献的各方，其中就包括 DApp 开发者。然而我们看到的现实是，在以比特币为代表的区块链系统中，出块奖励被发放给了矿工节点；在诸多基于 PoS（Proof of Stake）的区块链系统中，出块奖励被发放给了系统代币的持有者；与之相

[1] https://www.microsoft.com/en-us/windows
[2] https://en.wikipedia.org/wiki/Linux
[3] https://en.wikipedia.org/wiki/MacOS
[4] https://en.wikipedia.org/wiki/IOS
[5] https://en.wikipedia.org/wiki/Android
[6] https://www.sketchapp.com/support/requirements/other-platforms/
[7] https://en.wikipedia.org/wiki/Apple_Inc.
[8] https://en.wikipedia.org/wiki/AutoCAD
[9] https://en.wikipedia.org/wiki/Microsoft

伴随的是，在诸多区块链系统中，DApp 开发者的利益一定程度上都被漠视或侵害了。

一般而言，一个去中心化应用可以理解为为了实现特定功能的一系列智能合约集合。智能合约是一种旨在以信息化方式传播、验证或执行合同的计算机协议。智能合约允许在没有第三方的情况下进行可信交易[10]。从技术架构来看，大部分 DApp 通常以智能合约作为后端，同时采用常见的前端技术与之交互，DApp 形态既可以是传统个人电脑客户端，也可以是移动应用或者网页应用。

我们认为，去中心化应用平台、DApp 开发者及 DApp 用户，这三者是互相促进、共生共荣的关系。首先，去中心化应用平台的出现，扩大了区块链开发者这一群体，越来越多的开发者尝试开发满足不同需求的 DApp，并从 DApp 的开发中获益；其次，DApp 开发者提供了丰富多样的 DApp，扩大了区块链的应用场景，为区块链带来了更多的增量用户；最后，DApp 用户驱动着去中心化应用平台的不断的优化、升级，增加去中心化应用平台之上代币的流通性，使得整个区块链系统得以发展。因此更进一步的，我们认为，DApp 开发者的利益应该得到公平的分配，并给予保证，这是区块链作为去中心化应用平台维持可持续发展的关键所在。

需要注意的是，本文所述的开发者仅指去中心化应用平台之上的 DApp 开发者，不特指星云链上的 DApp 开发者，亦不包括区块链系统本身的开发者，因此，如无歧义，下文所述的开发者皆指 DApp 开发者。特别的，DApp 开发者在身份上可能同时持有一定数量的代币，而 DApp 开发者因为持币而带来的任何收益都不能等同的认为享受到了区块链系统发展带来的增值，其作为 DApp 开发者的利益依然可能是被漠视甚至被侵害的。

让应用开发者在平台的发展中公平获益，在技术上并不容易实现。一方面，在传统的软件开发行业中，平台的发展及获利状况及由中心化的组织掌握，平台之上的应用开发者无从知晓或参与相应的利益分配；另一方面，应用开发者为平台发展所做出的贡献难以量化，利益分配难以做到公平公正。而在以区块链为基础的去中心化应用平台上，这一状况有望得到改善。区块链的价值来源于代币的流通性，DApp 的使用情况被公开记录在区块链上，得益于此，**基于 DApp 的使用情况，量化 DApp 的使用对一个区块链系统的发展所做出的贡献，并进一步给予开发者公平的激励**，是必要且可行的。

理想情况下，对于 DApp 开发者的激励需要满足如下基本性质：

- 公平性：对于 DApp 开发者的激励，需要保证相对客观，即每个 DApp 需要被公平的对待，其使用情况需要被真实的衡量，并摒弃可能存在的操纵行为。
- 有效性：对于 DApp 开发者的激励，需要真实反映用户的偏好，即获得高激励的 DApp 是活跃用户喜欢且经常使用的，而获得低激励的 DApp 是鲜有用户问

[10] https://en.wikipedia.org/wiki/Smart_contract

津的。

本文提出 DApp 开发者激励协议（Developer Incentive Protocol，DIP），试图给予开发者激励，让 DApp 开发者能够公平的在去中心化应用平台的发展中获益。我们深知，用户对于 DApp 的真实评价是主观且多维的，一个理想的 DApp 开发者激励协议可能是不存在的，本文给出的 DApp 开发者激励协议依然会存在各种不足，其中仍然包含了我们对于 DApp 开发者的偏好。然而，在抵抗操纵及保证开发者利益之间，本文所做出的权衡依旧富有创新性，即在保证 DApp 开发者利益的前提下，在抵抗操作方面，做出了最大限度的努力。

本文提出的开发者激励协议基于已有的星云指数（Nebulas Rank）[1]，由于星云指数的相关性质，开发者激励协议能够在一定程度上很好的保证上述性质。直观来说，开发者激励协议将用户对 DApp 的使用情况简化为用户根据自己的喜好对 DApp 进行投票的问题，用户的投票总数和用户的星云指数相关，而对 DApp 的使用则为对 DApp 的投票，最终根据投票结果，对开发者给予相应比例的激励。

本紫皮书在给出开发者激励协议的理论模型之外，还对其抵抗操纵的性质进行了分析，并对系统中如何实现开发者激励协议给出了必要说明，例如，如何对开发者激励协议进行必要的调整及更新，从而对开发者激励协议的实际落地给出了具体的工作方向。

> 特殊提示：本开发者激励协议紫皮书作为专项讨论开发者激励协议的紫皮书，对星云技术白皮书（2018 年 4 月发布的 1.02 版本）[2] 中开发者激励协议相关章节进行了大幅度的升级和拓展。相对于一年前的概念论证，经过一年的深入思考与实际验证，我们有信心和能力设计出更为严谨的算法，并对开发者激励协议的更多实际细节问题提供明确的解决方案或方向。

2 背景介绍

本文提出的开发者激励协议，参考了诸多前人的工作，也在我们之前的工作的基础上，做了一定的延伸，此处，给出相关工作，这些研究工作对本文的工作具有重要的参考和指导作用。

2.1 DApp 开发者激励

据我们所知，目前，各个基于区块链的去中心化应用平台并未提供任何长效的 DApp 开发者激励机制。作为区块链 2.0 的代表，以太坊突破性地提供了图灵完备的

智能合约，其上已经拥有一定数量的 DApp，涵盖了游戏、博彩、众筹、借贷等众多类型，其中以 2017 年底的以太猫（CryptoKitties）和 2018 年中的 Fomo3D [3] 最为出名，两者一度引发了以太坊网络交易拥塞。实际上，大量 DApp 正如同前两者一样，只能通过向用户收取费用而获利，并不能从以太坊市值的增长或以太坊的挖矿奖励中获利。

在 DApp 开发者激励缺失的情况下，DApp 的应用场景亦受到了一定程度的影响。例如，潜在的、免费的 DApp 会由于难以收到相应的回报而流产，从而导致 DApp 的数量、质量以及多样性等各个方面受到影响；相对的，公平的、有效的 DApp 开发者激励机制，可以使得开发者能够专注于 DApp 的开发，进一步促进整个区块链生态的繁荣、可持续发展。

一定程度上，各个新兴的区块链系统认识到了激励机制对于构建区块链生态的必要性。例如，在星云链开展的激励计划中，总共产生了超过 6781 个 DApp，并且大量优秀开发团队得以走向前台并获得高额投资 [4]。其他类似公链也随之效仿推出了短期的基于中心化管理的激励活动，此类激励活动以对社区进行宣传为主要目的，在这类活动中官方主观评价占据主导因素，并且缺乏长期持续性。

2.2 星云指数

星云指数（Nebulas Rank，NR） [1] 给出了每个账户对经济总量的贡献度，具有良好的抗操纵性。特别地，星云指数给出了 Wilbur 函数，具有以下两个性质

特征 1. 对于任意大于 0 的两个输入变量 x_1，x_2，其计算函数之和小于其和的计算函数。

$$f(x_1 + x_2) > f(x_1) + f(x_2) \quad x_1 > 0, x_2 > 0 \tag{1}$$

特征 2. 当任意大于 0 的两个输入变量 x_1，x_2 趋近于无穷大时，其计算函数之和趋近于其和的计算函数。

$$\lim_{x_1 \to \infty, x_2 \to \infty} f(x_1 + x_2) = f(x_1) + f(x_2) \quad x_1 > 0, x_2 > 0 \tag{2}$$

上述两个性质作为星云指数的基础，同样为开发者激励协议提供了良好的抗操纵性。

2.3 投票机制

如前所述，在开发者激励协议中，用户使用 DApp 的过程，可以看做一个用户给 DApp 投票的过程，其后的激励机制则类似排名算法。关于投票系统及排名算法，各个领域已有大量的相关工作，因此我们参考了相关的工作并举例如下。其中最著名的结果之一是所谓的 Arrow 定理 [5]。其指出不存在任何一个排名算法能够同时满足非独裁性，帕累托有效性（Pareto Efficiency，即排名结果符合大多数人的利益）以及无关侯选者独立性（Independent of irrelevant alternatives，即两个候选者的排名相对关系不会受第三者影响）。这说明任何排名算法都不可能面面俱到。本文的开发者激励协议将更多的侧重于重要程度较高的以及广为人知的属性。

在现实生活中存在大量需要用到排名算法的场景。其中一个典型的、和本文类似的例子为亚马逊以及淘宝平台中买家对卖家（商户）的评分。好评率较高的商户将被推荐系统排在靠前的位置从而获得较高的关注度和点击率。特别地，这类电商平台存在着和女巫攻击类似的问题，即刷单问题：商户可以以各种手段雇佣大量买家账号为其给五星好评。就目前而言，即便是此类中心化平台方刷单的手段大部分为通过机器学习手段判断真实用户和虚假用户 [6, 7, 8]。然而实际表明此类方法效果并不理想。[9] 指出甚至人工识别都不能有效判别此类账户。 [10] 从机制设计的角度给出了一个消除商户刷单动机的算法，虽然和本文的模型不同但具有一定的借鉴意义。

[11] 介绍了网络社区给帖子排名的算法，结合了用户的投票数以及随时间衰减的过程。 [12] 介绍了 Reddit 上帖子的排名算法，引入了考虑了用户可以投反对票的情形。 [13] 介绍了 Reddit 关于评论的排名算法，将置信区间考虑了进去。IMDB [14] 上对电影的排名引入了贝叶斯平均的思想，可以拉近不同电影之间因投票人数的产生的差异。

得益于星云指数的抗作弊性，本文设计的用户激励协议能够降低虚假用户或操纵行为带来的收益。故本文设计的重点在于将用户的 NR 值通过交互行为转移到 DApp 的评分上来。

3 开发者激励模型

开发者激励协议，DIP，包括两个环节：DApp 评分以及开发者激励分配。

首先对于构建一个优秀的排名系统，其意义在于为第三方开发者提供了方便且高效的应用推广平台，同时也能为用户提供可信的推荐环境。正如目前 App Store 平台，优秀 App 在排行榜上拥有更显著的位置进而能受到更多用户的关注，而用户能通过排行榜上直接获取高质量的 App 无疑能提高其体验。更进一步地，App 排名也可以应用于关键词搜索中，类似于搜索引擎和电商平台中的搜索功能，与关键词相关

的候选 DApp 将按排名分顺序展示在搜索结果中，增加用户对搜索结果的满意程度。

另一方面，正如我们在第 2章所提到的，DIP 旨在为优秀 DApp 的开发者提供奖励，这进一步增加了开发者设计优秀 DApp 的动机，对整个生态的开发起到了促进作用。因此 DIP 第二个环节则是根据 DApp 评分排名提出公平的奖励分配机制。

3.1 模型表示

首先我们给出 DIP 模型中涉及的符号表示。

- $\mathcal{A} = \{a_1, a_2, ..., a_m\}$ 表示一定时间内所有参与评选的投票用户的集合，注意的是只有在一定时间内调用过任何 DApp 的外部账户（External Owned Account）才会被定义为投票用户。同时期所有用户集合表示为

$$\mathcal{A}^* = \{a_1, a_2, ..., a_m, a_{m+1}, ..., a_{m^*}\}$$

  ，即有 $m^* - m$ 个用户没有调用任何智能合约。

- $\mathcal{D} = \{d_1, ..., d_n\}$ 表示一定时间内所有 DApp 的集合。

- $e_{ij}, i = 1, 2, ..., m, j = 1, 2, ..., n$ 表示在一定时间内用户 a_i 对 DApp d_j 的总调用次数。鉴于区块链系统所具有的公开性，去中心化等特性，因此 DIP 评分模型和传统中心化的应用商城评分系统有所不同。简单地说，DIP 在去中心化环境中基于用户调用行为对 DApp 进行评分，具体描述见下一节。

- $\Gamma_i, i = 1, 2, ..., m$ 表示参与评选的用户 a_i 在一定时间内投票效用。 [1] 中已经证明 NR 值是衡量一个用户的有效价值尺度，故我们也把其用作 DIP 模型中用于决定用户投票效用的重要指标。

- $\Gamma_{ij}, i = 1, 2, ..., m, j = 1, 2, ..., n$ 表示用户 a_i 对 DApp d_j 的分贡献值。可以理解为 a_i 愿意为 d_j 投的票数。

- $S_j, j = 1, 2, ..., n$，表示 DApp d_j 的排名分，由该 DApp 从用户获得的所有分贡献值决定。直观地，排名分的高低直接决定了 DApp 在排行榜上所处的位置。

- M 表示开发者奖励总奖金池数量，来源于出块奖励。实际发放的奖励总额会根据该阶段社区参与度来适当进行加权调整。

- $U_j, j = 1, 2, ..., n$，表示 DApp d_j 开发者最终获得的奖励，这个奖励由奖励总额以及所有 DApp 的排名分共同决定。

综上，用户和 DApp 之间的交互可以用图 1中的二分图来表示。

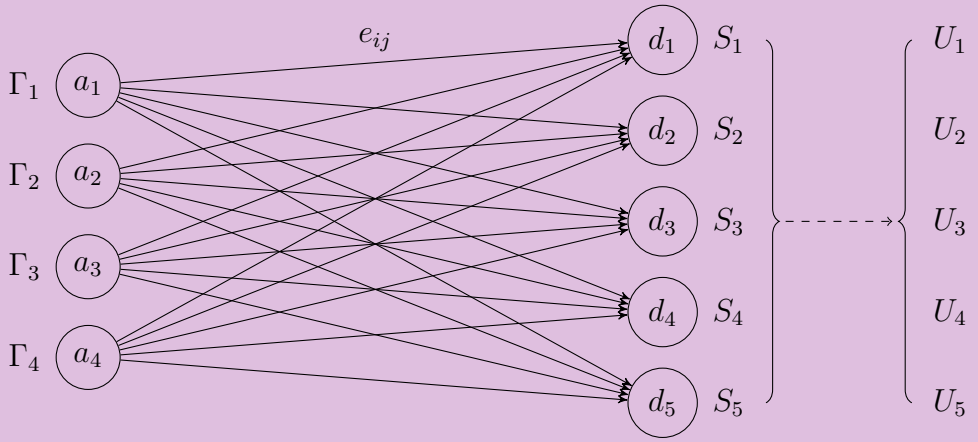

图 1: 用户与 DApp 间的交互

3.2 投票行为

在中心化的 App Store[11]中，系统会记录 App 的下载量等信息，后者也是应用评分的主要参考因素之一。而在区块链场景下，用户对于应用的使用会直接反应在调用合约地址行为上，例如表现为一段时间内用户 a_i 对 DApp d_j 的总调用次数 e_{ij}。DIP 通过用户调用行为作为用户的投票数据来源，相比传统的下载量信息具有如下优点：

- 用户调用智能合约行为记录在链上，很难篡改，相比中心化的下载量统计方式更加公开透明；
- 调用次数相比下载量信息更加细粒度，下载量只能记录用户一次性行为，而优秀的 DApp 应具有用户黏性，因此调用次数更能反映用户的真实使用情况。

实际上用户在调用智能合约时，还有其他信息可获取，例如调用智能合约过程中的 gas 消耗以及在调用过程中可能涉及到的资金交互，而 DIP 没有采取上述两者作为参考依据。

首先，用户每次调用智能合约 gas 消耗量是与智能合约内部的执行语句相关，而后者与 DApp 本身质量没有任何相关性。同时，目前星云系统中智能合约调用 gas 消耗开销平均只有 10^{-8} 数量级的 NAS，可以忽略不计。

而不考虑调用合约过程中资金转移的原因在于有效防作弊手段的缺失。直觉上来说，用户在使用 DApp 过程中愿意额外支付 token 确实能提高该 DApp 的被认同度。但实际情况下，这笔交易中 token 的最终流向存在以下三种可能。

1. Token 最终归智能合约即 DApp 开发者所有。这种情况可以认为用户自愿付出这些钱给 DApp，但此时 DApp 开发者相当于已经从中获益，再提高其排名获

[11] https://en.wikipedia.org/wiki/App_store

取额外奖励的意义不大。

2. DApp 机制设计需要资金流通，例如博彩类 DApp，其与用户之间存在大量资金交互。这是一种正常现象，此时用户与 DApp 的资金交互的动机主要在于用户希望通过这类交易获利，而并不能反应 DApp 本身质量，不应据此提高 DApp 排名。

3. DApp 开发者承诺所有为其投入的钱最终会返还到用户手里。这本质上是一种作弊手段，而据此提高 DApp 排名将会助长此种作弊手段。

实际上在不解析智能合约代码的情况下也无法判断用户和智能合约地址的资金交互属于上述哪一种情况，并且上述任何一种情况都存在不介入排名的理由，故 DIP 最终的算法独立于用户和 DApp 间的资金交互。

在 DIP 模型中，一个用户 $a_i \in \mathcal{A}$ 本质上是一个账户地址，正如 [1] 所提到的，一个用户实际上可以控制多个账户地址，由于建立新的账户地址是没有成本的，用户可以伪造出多个受他控制的地址进行投票，进行女巫攻击（Sybil Attack）。类似地，DApp 开发者也可以选择将自己的 DApp 分成多个地址，即将一个 DApp 拆分成多个低质量的 DApp，并且获得这些 DApp 的总奖励。同时一个 DApp 开发者也可以线下收买用户为他进行投票。

在设计 DIP 时，我们分析了上述作弊行为，并且设计了对应的解决方案。关于 DIP 抗作弊分析详见章节 5。

3.3 采样时间

在 3.1小节中，我们介绍了将使用 NR 值作为决定用户投票权重的重要指标。根据星云指数黄皮书定义 [1]，DIP 中的数据采样周期远大于 NR 的更新周期，这意味着系统在统计用户调用行为的过程中，用户自身 NR 可能会发生变化甚至较大波动。

直觉上，最简单的策略是将 DIP 中用户调用行为采样周期和 NR 更新周期同步，然而实际上在较短的采样周期内（如一天），大部分用户调用 DApp 次数并不高，在用户行为稀疏的情形下对 DApp 评分意义不大，同时无法保证在 5章节所描述的相关特性。

因此我们需要适当延长数据采样周期，收集足够的用户调用合约行为，同时保证在采样周期内用户 NR 值波动不会太大。对于某个账户，其星云指数在一段时间内的变化如图 3.3所示。这里我们将一次评选过程划分成若干阶段，根据所有用户 NR 变化的数据，然后取整数 t 使得绝大部分用户的 NR 值在 t 天内 NR 值的方差小于某个阈值 τ。我们把连续 t 天作为一个采样周期，通过取这一周期内用户的平均 NR 值以

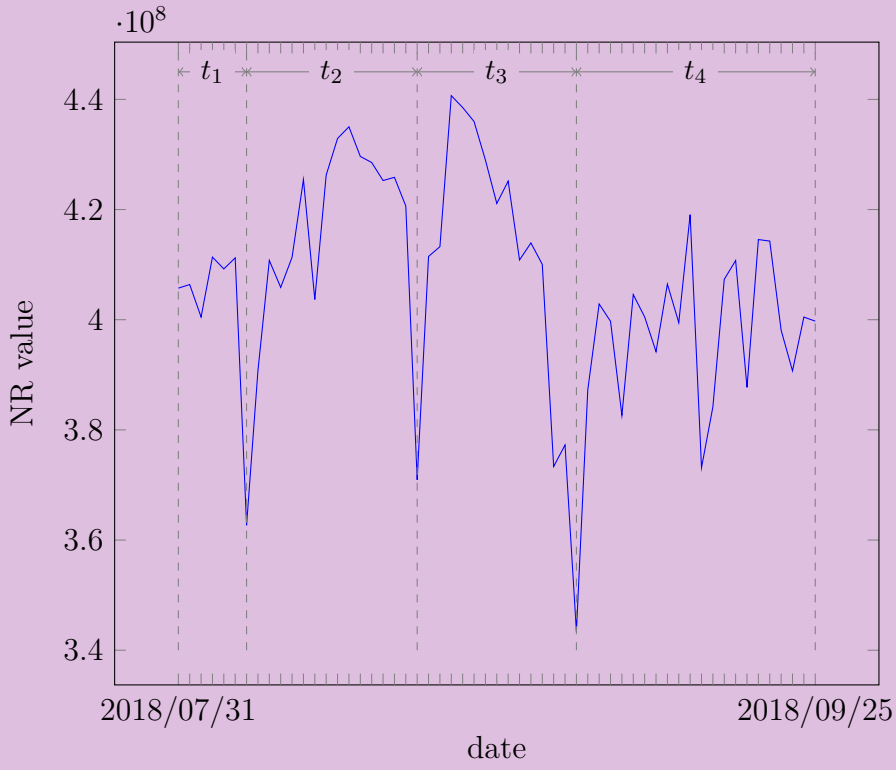

图 2: Nebulas 某主网账户的 NR 变化图。主网地址为 n1Ugq21nif8BQ8uw81SwXHK6DHqeTEmPRhj。

及调用 DApp 数据来计算 DApp 的排名分及开发者最终奖励，并把一个评选周期内所有阶段的数据取其平均值作为最终结果。

4 开发者激励协议

基于上一章节的模型，本章主要介绍开发者激励协议。DIP 包括 DApp 评分和激励分配两个环节，具体地，从用户调用行为到开发者获取激励包括调用次数到投票效用、投票效用到分贡献值、分贡献值到排名分以及排名分到最终奖励四次转换。

4.1 投票效用与分贡献值的计算

对于任何一个用户 a_i，我们将 Γ_i 定义为该用户的投票效用，即可以理解为用户手中握有的选票总张数。在 [1] 中已经证明了星云指数能够有效衡量账户的价值，因此在 DIP 中将使用星云指数衡量用户的投票效用。对于用户 a_i 而言，其投票总效用

可以表示为用户 a_i 的 NR 值函数：

$$\Gamma_i = f(\mathcal{C}(a_i)) \tag{3}$$

其中 $\mathcal{C}(a_i)$ 表示为用户 a_i 的星云指数。

通常地我们希望 f 单调递增，即星云指数高的用户可以获得更高的投票权。这里我们给出符合条件的函数：

$$f(\mathcal{C}(a_i)) = \mathcal{C}^2(a_i) \tag{4}$$

即

$$\Gamma_i = \mathcal{C}^2(a_i) \tag{5}$$

该表达式具有一些良好特性，例如抗女巫攻击，具体分析见第 5章节。

接下来我们讨论投票效用分配机制，根据 5小节，Γ_{ij} 表示为用户 a_i 对 DApp d_j 的分贡献值，其定义为

$$\Gamma_{ij} = \frac{e_{ij}}{e_{i0} + \sum_{j=1}^{n} e_{ij}} \Gamma_i \tag{6}$$

式 6可以理解为对 d_j 调用次数在总调用次数上的占比。这里 e_{i0} 表示 a_i 对于不属于上述任何一个 DApp 的调用次数。用户 a_i 可以任意调整 e_{i0} 以及 e_{ij} 的数值。

由于 e_{i0} 的引入，不难得出

$$\sum_{j=1}^{n} \Gamma_{ij} \leq \Gamma_i$$

式 6需要达到的目的仅仅是让用户可以任意分配自己的星云指数用于投票（即任意选择分贡献值）。实际上有些 DApp 可能采取强行增加调用次数的手段（比如规定必须调用两次才生效），但因为用户调用次数和星云指数都是可见的，用户仍然可以通过调整调用次数来达到自己期望达到的贡献值分配方式。

引入 e_{i0} 的意义在于，为了保证用户的个人理性（individual rationality），即用户参加此次活动不会损失利益，我们不强制用户投出所有的选票。用户可以选择性的行使部分投票权或完全弃权，通过适量增加 e_{i0} 的值。[12]这适用于用户觉得质量上乘的 DApp 太少的情况。

[12] e_{i0} 的实现可以通过官方设立一个空智能合约，不含任何实际效用。用户可以调用该智能合约任意次数。

4.2 排名分的计算

在得到用户的分贡献值后，我们可以计算 DApp 的排名分。给定所有的分贡献值 $\Gamma_{ij}, i = 1, 2, ..., m, j = 1, 2, ..., n$，我们将 DApp d_j 的排名分定义为关于所有调用用户的分贡献值的函数：

$$S_j = g(\Gamma_{1j}, \Gamma_{2j}, ..., \Gamma_{mj}) \tag{7}$$

同样地，这里我们给出一个符合条件的函数：

$$g(\Gamma_{1j}, \Gamma_{2j}, ..., \Gamma_{mj}) = \sum_{i=1}^{m} \sqrt{\Gamma_{ij}} \tag{8}$$

即 DApp 的排名分为所有调用其用户的分贡献值的开方和。对于用户 a_i 而言，其只投票给一个 DApp 时（在不弃权情况下），他的投票分贡献值之和为 $\sqrt{\Gamma_i}$。而当他将票分散给不同的 DApp 时，根据开方函数 $\sqrt{a+b} < \sqrt{a} + \sqrt{b}$ 的性质，他投票的总效用会提高。这意味着他接触了更多的 DApp，而这也是我们的系统所鼓励的。在 5章节将会给出这样构造排名分的方法的详细分析以及性质证明，类似的相关分析在论文 [15] 中也有提及。

当排名分 S_j 给出之后，即可以根据排名分对 DApp 进行排名。例如在星云 NAS nano 客户端[13]中，排名分高的 DApp 将被放在更显著的位置，也会受到更多的关注。

4.3 最终奖励的计算

DIP 对于用户而言提供了可信的 DApp 排名[14]，而对于开发者而言，我们还需要根据排名分进行奖励分配。

给定所有 DApp 的排名分 $S_i, i = 1, 2, ..., n$，定义 DApp d_j 开发者的奖励为

$$U_j = \frac{S_j^2}{\sum_{k=1}^{n} S_k^2} \cdot \lambda M \tag{9}$$

其中 M 是来源于出块奖励的奖金池总数，λ 定义为参与系数，即我们希望参与评选

[13] https://nano.nebulas.io/

[14] 我们假设用户只关心心目中的 DApp 的排名，而不会关心 DApp 开发者具体分配到多少奖励。

的用户总 NR 值越多给的奖励总额越大，具体定义为

$$\lambda = \min\{\frac{\Gamma_p}{\alpha\Gamma_s} \cdot \min\{\frac{\beta\Gamma_p^2}{\sigma^2(\Gamma_p)}, 1\}, 1\} \tag{10}$$

其中

$$\Gamma_p = \sum_{i=1}^{m}(\Gamma_i - \Gamma_{i0}), \quad \Gamma_{i0} = \frac{e_{i0}\Gamma_i}{e_{i0} + \sum_{j=1}^{n} e_{ij}}$$

即 Γ_p 为参与投票的用户有效分贡献之和，同时

$$\Gamma_s = \sum_{i=1}^{m^*}\Gamma_i,$$

即 Γ_s 为社区所有用户投票效用值（NR 值的平方）的总和。σ 为参与投票用户有效分贡献的标准差（方差的开方），

$$\sigma^2(\Gamma_p) = \sum_{i=1}^{m}(\Gamma_i - \frac{1}{m}\Gamma_p)^2$$

其最大值为 $\frac{(m-1)^2}{m^2}\Gamma_p^2$。$\alpha, \beta < 1$ 为可调的参数。

引入参与系数目的是期望参与活动的用户总效用达到某个阈值（社区总效用的 α 倍），以及让参与评选用户效用值的方差限定在某个范围内，防止出现少量高 NR 值用户带大量虚假账户的情况。两者可互补，即参与投票用户 NR 总值够高时可以不考虑方差的影响。

5 性质分析

在介绍完开发者激励协议后，这一章我们具体分析在实际情况下可能出现的作弊情况以及 DIP 所具有的抗作弊性质。分别从用户和开发者的角度来看，存在的作弊行为主要包括收买用户，恶意分拆 DApp，女巫攻击等等。

5.1 收买用户

所谓收买是指 DApp 开发者通过各种手段例如贿赂用户从而获得用户的全部投票权，这种现象在现实生活中也非常普遍。这里我们假设所有正常用户都是利益最大化的。我们认为正常用户关心的是他心目中 DApp 在排行榜上所处的排名，而不关心

DApp 开发者最终会获得多少奖金。亦即，每个用户乐于最大限度的提升自己心目中优秀 DApp 的加权排名分。我们的二阶排名算法保证了下面这个特征：

特征 3. 在 DIP 模型中，对于一个利益最大化的用户，一般而言，他会将手中的票投给多个不同的 DApp。

我们用如下模型来具体说明：

不失一般性，假设用户 a_i 对所有 DApp 的价值权重为 $b_{i1}, b_{i2}, ..., b_{in}$(可理解为用户对不同 DApp 的真实倾向)，采用公式 8的形式，则该用户最终分配的分贡献值满足

$$\frac{b_{i1}}{\sqrt{\Gamma_{i1}}} = \frac{b_{i2}}{\sqrt{\Gamma_{i2}}} = \cdots = \frac{b_{in}}{\sqrt{\Gamma_{in}}}$$

即对于用户 a_i 而言，其贡献值分配策略会与其对 DApp 的真实评价相符，具体证明见附录 A.1。

传统投票统计模型中常常使用更常见的线性排名分算法，即

$$g(\Gamma_{1j}, \Gamma_{2j}, ..., \Gamma_{mj}) = \sum_{i=1}^{m} \Gamma_{ij}$$

在这种模型下，理性用户则只会将所有票投给自己最喜爱的 DApp。相比之下，公式 8更能促进用户和 DApp 之间的交互，其根本原因在于根号函数的特性，即用户把票投给多个 DApp 才能最大化利用其投票效用，在论文 [15] 中也有类似的分析。综上，用户会倾向于投给多个 DApp 的同时保证自己最喜欢 DApp 的领先性，即上面的比例等式。

在实际生活中，传统线性投票模型通常会限制用户为单个目标投票的最大票数，从而强制让用户投给多个目标。而我们的算法则通过激励从本质上达到同样的目的，并且具有更加简洁优美的数学表达形式。

同时，基于上述特性，我们可以得出关于 DIP 具有抗收买特性的推论：

推论 1. 被收买的用户对 DApp 排名分的总贡献远远小于正常的用户。

对于已经被收买的用户 a_i，最多给收买其的 DApp 开发者提供 $\sqrt{\Gamma_i}$ 分贡献值。而一个正常未被收买的用户，假设其期望投票 K 个 DApp($K > 1$) 而不是被收买只投票给某一个 DApp[15]，在其对这些 DApp 的价值权重是分布较为均匀的情况下，用户 a_i 给所有 DApp 带来的总排名分提升大约在 $O(\sqrt{K\Gamma_i})$ 这个级别，即，未被收买

[15] K 反应的是该用户给出具有区分度的投票的 DApp 数目，通常是大于 1 的，只要该用户对 DApp 价值权重分布不是太极端，即只喜欢某个特定的 DApp 而对其他 DApp 打分都趋近于 0。

用户起到的作用是被收买用户的 $O(\sqrt{K})$ 倍，这样就一定程度上提高了开发者收买用户的代价。

5.2 恶意分拆

对于开发者而言，另一种作弊行为就是恶意分拆。所谓恶意分拆是指 DApp 开发者将自己的 DApp 强行分拆成若干个 DApp 以获得所有分拆的 DApp 的总奖励。直觉上，这种分拆会增加参与激励的候选 DApp 数量进而增加总奖励。然而我们的模型能保证这种情况不会发生。这里我们认为 DApp 开发者关心的是最终获得的总奖金。同时也有一定的高排名带来隐性收益。

具体而言，作为最终的奖励分配策略，公式 9的凹函数性能保证如下特性：

特征 4. 在所有投票者均为正常用户的情况下，开发者通过 DApp 分拆不会提升他的收益。

这里假设正常用户存在下列两种情形：其一用户面对 DApp 分拆选择简单的把原本投给分拆前 DApp 的票分散开来（分给拆分后的 DApp），这种行为一般表现为同一个应用拥有不同的智能合约调用地址；其二，假设拆分之前用户对该 DApp 的价值权重为 c，对两个拆分之后的 DApp 的价值权重分别为 a 和 b，可以得出 $c \geq a + b$。这个可以理解为 DApp 拆分之后两者的质量将大幅下降且缺乏联动性，导致两者的质量和比原 DApp 还要差。在这两类情况下，开发者不会提升其收益，具体证明见附录 A.2。

进一步地，开发者还可采取的作弊手段为同时进行 DApp 分拆和收买，例如，先分拆成 K 个 DApp，然后让被收买的用户均匀的将票投在自己拆分的 K 个 DApp 上以实现效用最大化。我们有下面的推论防止此类情形发生：

推论 2. 即使引入被收买者的情况下，开发者进行 DApp 恶意分拆不会提升他的收益。

具体证明见附录 A.3。

值得注意的是，开发者将 DApp 进行拆分同时也会降低在排行榜上的排名，从而减少排名带来的隐形收益。综上所述，我们的算法能从本质上防止 DApp 分拆的进攻手段。

当然对于开发了多个不同 DApp 的开发者，多个应用之间并不存在拆分或镜像关系，因此其收益不受影响。

5.3 女巫攻击

广义上的女巫攻击 (Sybil Attack) 是指攻击者通过创建大量假名标识来破坏对等网络的信誉系统，使用其获得虚假的高重要性评分 [16]。在星云指数黄皮书中，已经分析了星云指数对通过创建大量账户提升星云指数这类作弊行为具有良好的抵抗性 [1]。因此在 DIP 的排名算法下，用户无法通过创建大量账户来获取更高的 NR 值，即

$$\mathcal{C}(c) > \mathcal{C}(a) + \mathcal{C}(b)$$

其中 c 为原账户，a,b 为拆分后子账户。按照公式 6定义进而其投票效用也满足如下约束，

$$\sqrt{\Gamma_{a+b}} > \sqrt{\Gamma_a} + \sqrt{\Gamma_b} \tag{11}$$

假设用户进行女巫攻击的目的是为了提升某个特定 DApp 的排名分以及其开发者的最终奖励。根据上述约束我们可得出下面的推论:

特征 5. 对于任何用户，进行女巫攻击不会增加他所投的 DApp 的排名分以及最终奖励。

进而抗女巫攻击的性质得到了保证。

6 开发者激励协议的实现

开发者激励协议的完整实现不在本文的讨论范围内，此处仅讨论系统实现中所需要解决的一些关键问题。

6.1 如何发放奖励?

为了发放奖励，需要一个专用的奖励发放账户 D，同时，每个出块的奖励按照固定的比例发送到 D。

开发者所应得的奖励会定期发送[16]，为了在链上发送相应的奖励，需要发出账户的私钥对发出的交易进行签名，因此，为了奖励发出账户 D 的安全，需要对该账户进行特殊处理。

[16]奖励发送时间间隔等于 3.3小节中的采样时间。

首先，系统需要增加一个用于发放奖励的交易类型，`dip` 交易，`dip` 交易中包含了一个 DApp 开发者账户获得奖励的数量及所在区块高度等信息。其次，系统将拒绝 D 发出的除 `dip` 交易之外的一切交易，以保证无人能从 D 中提出用于奖励的代币。最后，区块链系统中的验证节点需要对 `dip` 交易进行验证，具体来说，验证节点需要在本地重新执行开发者激励协议，并验证 `dip` 交易中的数据是否与本地的计算结果一致。

通过上述方式，既能保证开发者奖励的正常发放，又能保证奖励发出账户 D 的账户安全。

6.2 开发者激励协议的更新

我们深知，开发者激励协议是和整个生态紧密相关的，随着生态的不断变化，开发者激励协议的计算也需要不断更新，尤其是其中的各个参数。如何快速地更新开发者激励协议的计算非常关键。对此，我们将通过星云原力（Nebulas Force）来保障核心星云指数计算的更新迭代。

我们会更新区块结构，新的区块结构中将包含开发者激励协议的算法及参数 (以 LLVM IR 形式)，星云虚拟机（NVM）作为算法的执行引擎，从区块中获得开发者激励协议的算法及参数，并执行算法，在节点内获得账户的奖励代币数量。

在算法或参数需要更新时，我们将和社区一起协作，让新的区块中包含入最新的算法及参数，从而保证整个更新过程的及时性及平滑性，亦避免了可能到来的分叉。

7 未来工作

7.1 多维投票行为

在 3.2小节中，我们采用了用户调用行为作为 DApp 排名参考依据，我们同时也给出了不采用调用过程中涉及的资金交互作为参考依据的原因。在未来工作中，在能够解析智能合约调用的情况下，我们可以更细粒度地区分调用过程中的资金转移原因，则可以尝试引入资金交互这一特性作为 DApp 排名参考依据。

7.2 DApp 之间的调用

DApp 的排名目前主要来自于用户的投票效用（即用户星云指数），这种传递过程来自用户调用合约行为。而更复杂的调用行为，例如智能合约直接的内部调用，也

可能进一步传递用户投票效用。因此在接下来的工作中，我们可以给定所有 DApp 排名分初始值，随后根据 DApp 间调用关系图运行 Page Rank 算法 [17] 求得最终排名分。

参考文献

[1] "Nebulas yellowpaper." https://nebulas.io/docs/NebulasYellowpaperZh.pdf.

[2] "Nebulas whitepaper." https://nebulas.io/docs/NebulasTechnicalWhitepaperZh.pdf.

[3] "Fomo3d." https://exitscam.me/shakedown.

[4] "Nebulas incentive." https://nebulas.io/incentive_spec.html.

[5] K. J. Arrow, "An extension of the basic theorems of classical welfare economics," tech. rep., STANFORD UNIVERSITY STANFORD United States, 1951.

[6] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh, "Spotting opinion spammers using behavioral footprints," in Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 632–640, ACM, 2013.

[7] N. Jindal and B. Liu, "Opinion spam and analysis," in Proceedings of the 2008 international conference on web search and data mining, pp. 219–230, ACM, 2008.

[8] K.-H. Yoo and U. Gretzel, "Comparison of deceptive and truthful travel reviews," Information and communication technologies in tourism 2009, pp. 37–47, 2009.

[9] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 309–319, Association for Computational Linguistics, 2011.

[10] Q. Cai, A. Filos-Ratsikas, C. Liu, and P. Tang, "Mechanism design for personalized recommender systems," in Proceedings of the 10th ACM Conference on Recommender Systems, pp. 159–166, ACM, 2016.

[11] A. Salihefendic, "How hacker news ranking algorithm works," 2010.

[12] A. Salihefendic, "How reddit ranking algorithms work," Hacking and Gonzo, vol. 23, 2010.

[13] E. Miller, "How not to sort by average rating," 2009.

[14] "IMDB." https://www.imdb.com/chart/top.

[15] V. Buterin, Z. Hitzig, and E. G. Weyl, "Liberal radicalism: Formal rules for a society neutral among communities," arXiv preprint arXiv:1809.06421, 2018.

[16] D. Quercia and S. Hailes, "Sybil attacks against mobile users: friends and foes to the rescue," in INFOCOM, 2010 Proceedings IEEE, pp. 1–5, IEEE, 2010.

[17] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web.," tech. rep., Stanford InfoLab, 1999.

附录 A　证明

A.1　特征 3证明

证明. 不失一般性，假设用户 a_i 对所有 DApp 的价值权重分别为 $b_{i1}, b_{i2}, ..., b_{in}$。这些均为定值。假设用户 a_i 对所有 DApp 的分贡献值分别为 $\Gamma_{i1}, ..., \Gamma_{in}$，这些为可调整的变量。

用户 a_i 的优化目标为他所提供的加权总排名分，定义为

$$w_i = \sum_{j=1}^{n} b_{ij}\sqrt{\Gamma_{ij}}$$

根据柯西不等式可得

$$w_i = \sum_{j=1}^{n} b_{ij}\sqrt{\Gamma_{ij}} \leq (\sum_{j=1}^{n} b_{ij}^2)(\sum_{j=1}^{n} \Gamma_{ij}) \leq (\sum_{j=1}^{n} b_{ij}^2)\Gamma_i$$

上式最右边为定值。等号成立当且仅当

$$\frac{b_{i1}^2}{\Gamma_{i1}} = \frac{b_{i2}^2}{\Gamma_{i2}} = \cdots = \frac{b_{in}^2}{\Gamma_{in}}$$

故命题得证。

□

A.2　特征 4证明

证明. 不失一般性，假设 d_1 开发者将其 DApp 拆分成 2 个 DApp。对于章节 5.2中第二种情形的正常用户 a_i，即，假设他对拆分之前所有 DApp 的价值权重分别为 $b_{i1}, b_{i2}, ..., b_{in}$，对拆分出的 2 个 DApp 价值权重为 b'_{i1}, b'_{i2}，根据我们的假设有 $b_{i1} \geq b'_{i1} + b'_{i2}$。

我们接下来计算拆分之前的 a_1 提供的分贡献值。计 $H_i = \sum_{j=2}^{n} b_{ij}^2$，根据特

征 3的结论及合分比定理有

$$\frac{\Gamma_{i1}}{b_{i1}^2} = \frac{\sum_{j=1}^{n}\Gamma_{ij}}{\sum_{j=1}^{n} b_{ij}^2} = \frac{\Gamma_i}{b_{i1}^2 + H_i}$$

故

$$\Gamma_{i1} = \frac{b_{i1}^2 \Gamma_i}{b_{i1}^2 + H_i}$$

类似的可得到拆分之后 a_i 对拆分的第 t 个 DApp 的分贡献值为（定义为 $\Gamma'_{it}, t = 1, 2$）

注意到 $b_{i1}^2 \geq (b\text{`}_{i1} + b\text{'}_{i2})^2 > b_{i1}'^2 + b_{i2}'^2$，可得

$$\Gamma_{i1} > \Gamma'_{i1} + \Gamma'_{i2}$$

上述分析了对于一个足够聪明的理性用户选择的分贡献值满足的条件。事实上，对于大部分普通用户，他们属于章节 5.2中第一种情形，即，他们面对 DApp 拆分所采取的方式通常为简单的将原本投给拆分前的 DApp 的所有票分散开来。无论哪种情况，均有

$$\Gamma_{i1} \geq \Gamma'_{i1} + \Gamma'_{i2}$$

定义 S'_1, S'_2 分别为 d_1 拆分之后的两个 DApp 的排名分，根据定义，有

$$S'_1 = \sum_{i=1}^{m} \sqrt{\Gamma'_{i1}}, \quad S'_2 = \sum_{i=1}^{m} \sqrt{\Gamma'_{i2}}, \quad S_1 = \sum_{i=1}^{m} \sqrt{\Gamma_{i1}}$$

定义 U'_1 为拆分之后 d_1 开发者的最终奖励，则

$$U'_1 = \frac{S_1'^2 + S_2'^2}{S_1'^2 + S_2'^2 + \sum_{j=2}^{n} S_j^2}\lambda M, \quad U_1 = \frac{S_1^2}{S_1^2 + \sum_{j=2}^{n} S_j^2}\lambda M$$

注意到在固定 $S_2, ..., S_n$ 的情况下，

$$U_1 \geq U'_1 \Leftrightarrow S_1^2 \geq S_1'^2 + S_2'^2$$

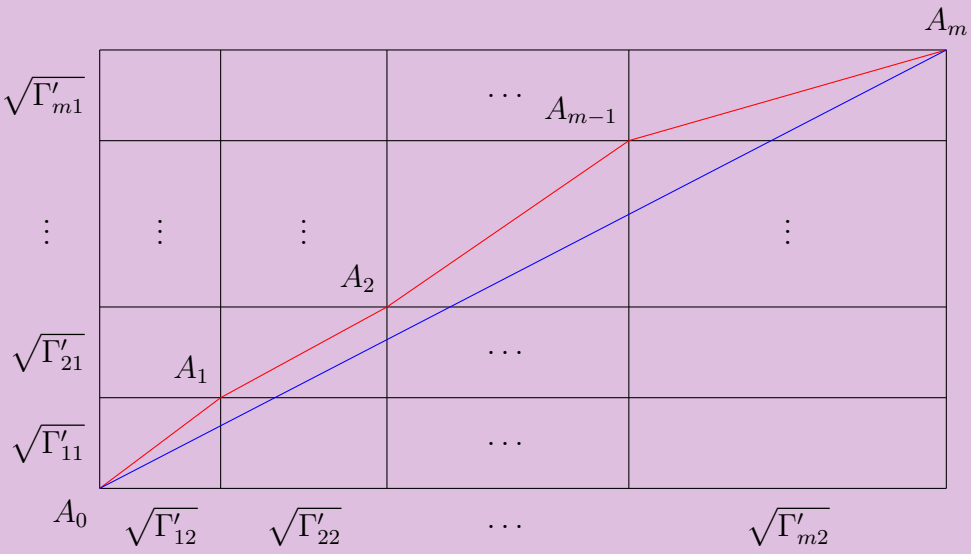

图 3: 最短路证明

所以，为了比较 d_1 开发者在拆分前后的收益，我们只需要比较下面两个量

$$S_1^2 = (\sum_{i=1}^{m} \sqrt{\Gamma_{i1}})^2, \quad S_1'^2 + S_2'^2 = (\sum_{i=1}^{m} \sqrt{\Gamma'_{i1}})^2 + (\sum_{i=1}^{m} \sqrt{\Gamma'_{i2}})^2$$

事实上，$S_1^2 \geq S_1'^2 + S_2'^2$ 可由最短路性质得出。

如图 3，构造一个网格，其长和宽均被分成了 m 段，其第 i 段长度分别为 $\sqrt{\Gamma'_{i1}}$ 和 $\sqrt{\Gamma'_{i2}}$。

那么，$S_1'^2 + S_2'^2 = A_0A_m^2$，即，恰好等于图中蓝线的长度的平方。而

$$S_1^2 = (\sum_{i=1}^{m} \sqrt{\Gamma_{i1}})^2 > (\sum_{i=1}^{m} \sqrt{\Gamma'_{i1} + \Gamma'_{i2}})^2 = (\sum_{i=1}^{m} A_{i-1}A_i)^2$$

即所有红线长度之和的平方。根据两点之间线段最短可得 $S_1^2 > S_1'^2 + S_2'^2$。

对于拆分成 $k > 2$ 个 DApp 的情形，只需转化成逐次拆分然后每次应用 $k = 2$ 时的结论即可。

故命题得证。 □

A.3 推论 2证明

证明. 对于一个被 d_1 开发者收买的用户，在 d_1 进行拆分之前，可以将该用户等价于一个价值权重向量为 $(1,0,0,...,0)$ 的正常用户（该用户将所有票都投给 d_1）。而拆分成 k 个 DApp 之后，假设被收买用户对这 k 个 DApp 的分贡献值为 $\Gamma_{t1},...,\Gamma_{tk}$，其和为定值。根据特征 3的证明中柯西不等式取等号的条件，可将该用户等价于一个价值权重向量为 $(\sqrt{\Gamma_{t1}}/C,\sqrt{\Gamma_{t2}}/C,...,\sqrt{\Gamma_{tk}}/C,0,0,...,0)$ 的正常用户，其中 $C=\sum_{j=1}^{k}\sqrt{\Gamma_{tj}}$。即对拆分出的 DApp 按某种比例分配权重，所有其他 DApp 权重为 0 [17]。此时因为

$$\sum_{j=1}^{k}\sqrt{\Gamma_{tj}}/C=1$$

可规约为特征 2 的情况，即变成满足假设的正常用户的情形（特征 4）。故命题得证。 □

A.4 性质 5证明

证明. 我们先考虑某个用户将其账户拆分成两个子账户的情况。固定其他用户的行为，假设 c 为该用户原账户，a,b 为拆分后子账户。S,S' 分别为拆分前后该用户指定 DApp 的排名分。U,U' 分别为拆分前后该用户的最终奖励。根据定义，有

$$S=\sqrt{\Gamma_c}+O,\quad S'=\sqrt{\Gamma_a}+\sqrt{\Gamma_b}+O$$

其中 O 为其他用户对该 DApp 的贡献值之和，是个定值。

由 11可得 $S<S'$，即该 DApp 排名分不会增加。

同时，根据定义有

$$U=\frac{S}{S+P}\lambda M,\quad U'=\frac{S'}{S'+P}\lambda M$$

[17]注意将所有价值权重同时扩大若干倍对结论没有影响，因为最终用户分贡献值总和只与权重占据的比例有关。

其中 P 为其他 DApp 的排名分的平方和，是个定值，故

$$U \geq U' \Leftrightarrow S \geq S'$$

由 $S < S'$ 可得 $U \leq U'$，即该 DApp 最终奖励不会增加。

对于拆分成 $k > 2$ 个子账户的情况，只需转化成逐次拆分然后每次应用 $k = 2$ 时的结论即可。 □

附录 B 修订记录

B.1 2019.4.8 修订

- 将 NR 值到投票总效用的转换函数（4）修改为 $f(\mathcal{C}(a_i)) = \mathcal{C}(a_i)$
- 将参与系数 λ 计算函数（10）修改为 $\min\{\frac{0.008}{1-r}, 1\}$，其中 $r = \frac{\Gamma_p}{\Gamma_s}$
- 修正若干错误。