区块链激励设计与分析

牛建宇和陈峰

摘要 《比特币白皮书》引入了区块链技术，以实现不依赖中央机构的去中心化电子现金系统。该技术的一大创新是激励设计，参与节点通过在最长的链上创建区块来获得奖励。激励设计对于安全的区块链系统至关重要，这一点在最近的许多著作中都有所体现。在本章中，我们将仔细研究三种有影响力的区块链协议的激励设计，包括比特币、以太坊和比特币-NG。对于每种协议，我们都介绍了潜在的基于激励的攻击，并通过一些理论结果来描述这些攻击的影响。我们希望，在我们的旅程结束时，读者能够对区块链激励设计和分析有更深入的了解。

1 导言

2008 年，中本聪在题为《比特币：点对点电子现金系统》的开创性论文中发明了比特币[24]。比特币旨在实现一种去中心化的电子现金系统，该系统不依赖中央机构发行货币和处理交易。一年后，基于中本聪开发的开源代码的比特币网络正式启动。现在，作为最大的加密货币，比特币按美元/比特币汇率计算的市值已超过 1 万亿美元。比特币的巨大成功在很大程度上归功于两项技术创新：所谓的*中本共识*（NC）协议，该协议可以实现比特币与其他加密货币之间的 "无缝连接"。

J.牛 (✉)
中国深圳南方科技大学
电子邮件：niujy@sustech.edu.cn

C.冯
加拿大基洛纳不列颠哥伦比亚大学（奥卡纳根校区） 电子邮件：
chen.feng@ubc.ca

D.D. A. Tran 等人（编），《*区块链手册*》，Springer Optimization and Its Applications 194, https://doi.org/10.1007/978-3-031-07535-3_4

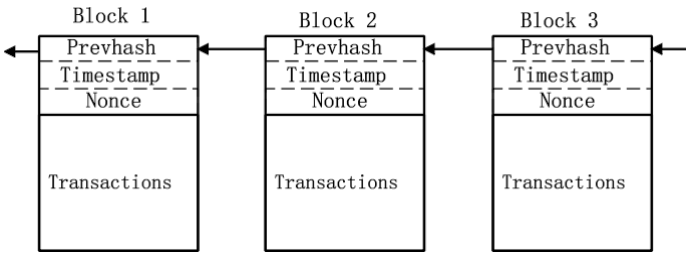

图 1 区块链数据结构示意图。每个区块包含一批交易、上一个区块的哈希值和其他元数据

NC协议是一种公开、不可更改的分布式账本，如今被广泛地称为区块链。NC 协议已在其他章节中讨论过。在本章中，我们将重点讨论区块链的激励设计和分析。本章主要基于我们之前对以太坊[12]和比特币-NG[27]的激励分析工作，以及第一作者的博士论文。

在区块链中，用户可以创建和交换交易，以修改他们的货币金额，新的交易可以分批进入区块进行处理。特别是，每个区块都通过加密哈希值与前一个区块链接，形成用户接受的区块链（这也是区块链名称的由来）。图 1 是比特币区块链数据结构的简单示意图。除了交易和哈希值参考，每个区块还包含时间戳、nonce（即随机字符串）和其他元数据。在这里，nonce 是通过解决计算难题产生的，也就是通常所说的工作量证明（PoW）。搜索有效非密钥的过程也称为挖矿，参与者称为矿工。矿工遵循最长链规则（LCR），就区块的递增序列达成一致。也就是说，矿工总是延长他们收到的最长链。非正式地讲，LCR 和 PoW 构成了中本共识。尽管简单，但只要一半以上的计算能力由遵守协议的诚实矿工控制，NC 就能让区块链安全运行。

在参与基于 NC 的区块链时，矿工必须支付计算硬件（如 CPU、GPU 或 ASIC）、电费和其他费用。经济成本使得任何矿工自愿支持该协议都是不切实际的。为了解决这一难题，（公共）区块链系统通常依赖于其激励设计，矿工可以为区块链中的每个区块获得区块奖励，即一定数量的自行发行的代币。此外，矿工还可以获得区块中包含的所有交易的交易费[24]。这些奖励可以支付挖矿成本，使挖矿有利可图，激励矿工贡献尽可能多的计算能力。这里的区块链协议采用了一个隐含的假设，即所有矿工都是理性的[16, 24]。如前所述，NC 的安全性依赖于这样一个假设，即大部分计算能力都是理性的。

由诚实的矿工控制。为保证诚实的合理性，数控系统应确保*激励相容*，即矿工只要偏离协议，就会遭受经济损失。换句话说，如果没有激励兼容性，理性矿工就会为了获得更高的收益而偏离协议，从而导致上述假设不成立，进而威胁系统安全。这些都说明了区块链激励设计的重要性。在本章中，我们将围绕三个有影响力的区块链协议，展开区块链激励设计与分析之旅：比特币、以太坊和比特币-NG。以太坊和比特币-NG是比特币的两个变种；它们都是基于NC和比特币的激励设计而设计的。对于每种协议，我们都提出了潜在的基于激励的攻击，并通过几个理论结果来说明这些攻击的影响。我们希望，在我们的旅程结束时，读者能够对区块链激励设计和分析有更深入的了解。

本章接下来的内容安排如下。在第 2 节中，我们介绍了比特币的奖励设计和分析，特别是来自 [11] 的自私挖矿分析。然后，在第 3 节中，我们对以太坊进行了激励分析，重点比较了以太坊与比特币的奖励设计。3，并在第 4 节中对 Bitcoin-NG 进行全面的激励分析。4.我们在第 5 节中提供了有关区块链激励设计和分析的进一步解读，并在第 6 节中对本章进行了总结。

2 比特币的激励设计与分析

在比特币白皮书中，没有协调的 矿工被建议遵守协议，即服从数控来挖掘区块并立即发布他们的区块[24]。这样，所有矿工都会得到公平的回报；他们获得的收入与其计算能力成正比。这也鼓励矿工在协议中投入更多计算能力。然而，在现实中，矿工们加入矿池并分享奖励，以保持稳定的收入。如今，比特币排名前六的矿池几乎控制了总计算能力的 75%。[1]计算能力集中化的趋势不仅与比特币的目标（即去中心化的电子现金系统）背道而驰，还引发了对激励问题的担忧。特别是，许多研究表明，一组串通一气、背离协议的矿工可能会从诚实挖矿中获得大于其公平份额的收益 [11, 15, 17, 25, 33]。这种行为被称为*自私挖矿*，相应的矿工被称为自私矿工。

[1] 矿池统计：https://btc.com/stats/pool。

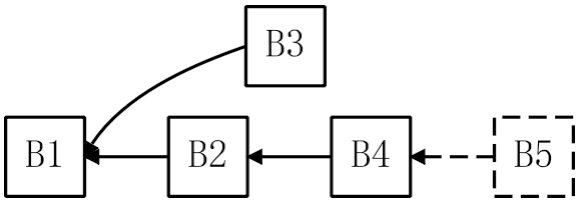

图 2 链式结构图

矿工。矿工尝试在其观察到的最长链上生成一个区块

2.1 比特币概述

比特币依靠中本共识（NC），使一群分散且互不信任的参与者在一个透明且不可更改的账本（也称为区块链）上达成协议。区块链是由哈希值链接的区块列表，每个区块包含一批有序交易。为了让所有参与者在相同的区块链上达成一致，NC 利用了两个组成部分：工作证明（PoW）机制和最长链规则（LCR）。NC 的每个参与者（也称为矿工）从网络中收集有效和未确认的交易，然后将这些交易排序并打包成一个区块。一个有效的区块需要包含一个工作证明，也就是说，它的所有者需要找到一个 nonce 值（即一个可改变的数据文件），使得该区块的哈希值具有所需的前导零[24]。前导零的长度也被称为挖掘难度，它决定了每次计算尝试找到有效 nonce 的概率。此外，系统还可以调整挖掘难度，以确保平均每 10 分钟挖掘一次新区块。

新区块生成后，将立即向整个网络广播。理想情况下，在下一个区块产生之前，所有参与者都应接受该区块。在现实中，两个新区块可能会同时被挖出，从而导致分叉，即两个 "子 "区块共享一个共同的 "父 "区块。为了解决这种分叉，诚实的矿工总是接受最长的链为有效链，并在其最后一个区块之后进行挖矿。特别是，如果多条链的长度相同，矿工会选择在最先收到的那条链之后挖矿。分叉不会永远持续下去，因为最长的分支将在竞争中获胜，并被所有矿工接受。所有最长矿工链的共同前缀称为*主链*。在比特币中，区块矿工将获得区块奖励（如果其区块最终被纳入系统主链）以及作为另一种奖励的交易费。这些奖励鼓励矿工将其计算资源投入到系统中。

2.2 比特币的自私挖矿

自私挖矿的想法最早是在比特币论坛上提出的[15]。后来，Eyal 和 Sirer 正式描述并分析了自私挖矿攻击 [11]。在这种

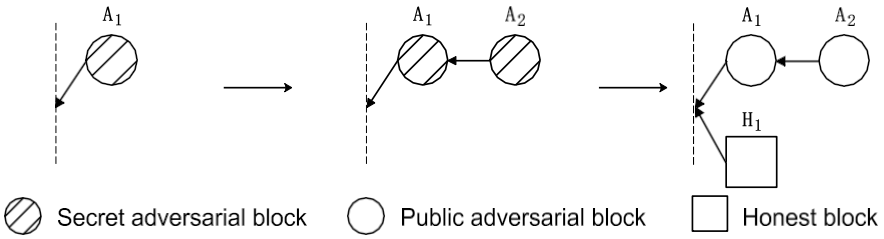

图 3 比特币中自私挖矿攻击的第一个例子

攻击时，自私的矿工首先将其新发现的区块私有化，创建一个隐藏分支。当一些诚实的矿工挖出一个新区块时，自私的矿工就会立即公布一些私有区块，试图使自己的分支成为最长的分支。这样一来，一些诚实矿工的区块就会被遗弃（因为没有被纳入最长的链中），而自私矿工可以获得比诚实矿工更多的区块奖励。让我们用两个例子来更好地说明攻击的原理。图 3 显示了这样一种情况：自私矿工首先在最长链上成功挖出两个连续的区块 A_1 和 A_2，然后在收到诚实矿工的区块 H_1 时发布这两个区块。这样，自私的矿工不仅能获得两个区块的奖励（区块 A_1 和 A_2），还能让诚实的区块 H_1 被所有矿工抛弃。如果自私的矿工总能重现这种情况，他们肯定能获得比诚实挖矿更多的区块奖励。

然而，自私的矿工并不总是像上述情况那样，率先挖出两个区块，并使自己的区块被所有矿工接受。图 4 显示了这样一种情况：自私的矿工成功挖出一个区块 A_1，然后发布这个区块，以匹配诚实的区块 H_1。这两个区块形成了两个长度相同的分叉分支，矿工们无法决定哪个是最长的。为解决分叉竞争问题，矿工应继续挖矿，直到更长的分叉获胜。如前所述，在这种情况下，诚实的矿工会在收到第一个分支后挖矿，而自私的矿工则会在自己的分支后挖矿。因此，下一个区块可进一步分为三种子情况：

(a) 自私矿工在自己的区块之后生成下一个区块 A_2 并发布。通过 LCR，所有自私矿工的区块都被诚实矿工接受，而诚实区块 H_1 则被放弃。

(b) 一些诚实的矿工在上一个诚实的区块 H_1 之后产生下一个区块 H_2。如果自私的矿工接受这两个诚实的区块 H_1 和 H_2 并在它们之后挖矿，其区块 A_1 将被放弃。

(c) 在自私矿工的区块 A_1 之后，一些诚实的矿工会生成下一个区块 H_2。区

块 A_1 和 H_2 都会被接受，而区块 H_1 则会被 LCR 放弃。

从这个例子中，我们可以发现自私矿工的第一个区块 A_1 有被放弃的风险（如图 4c 所示）。换句话说，自私的矿工可能会损失

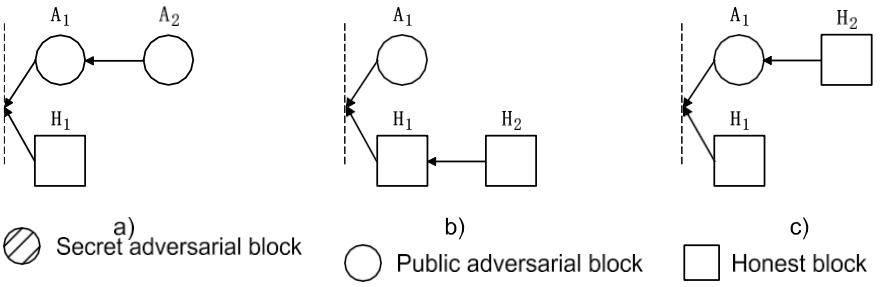

图 4 比特币中自私挖矿攻击的第二个例子

因此，这种攻击并不总是有利可图的。此外，我们还可以看到，自私挖矿攻击能否成功获利取决于两个参数：自私矿工所拥有的计算能力分数（用 α 表示）和平局期间在自私矿工区块上工作的诚实计算能力分数（用 γ 表示）。当 α 增加时，自私矿工领先两个区块并在平局期间赢得分叉竞赛的可能性也会增加（在这种情况下，自私矿工没有损失）。另一方面，当 γ越大，自私矿工的区块被放弃的可能性就越小。特别是，当 $\gamma = 1$ 时（所有诚实矿工首先收到自私矿工的区块 A_1），自私矿工的区块 A_1 总是被所有矿工接受。

上述例子只是说明了自私挖掘攻击的几种情况。我们可以进一步设计一种涵盖所有情况的完整自私挖矿策略。在接下来的章节中，我们将介绍 [11] 中的自私挖矿策略，并给出相关的理论结果。另外，需要注意的是，上述策略在不同的 α 和 γ 条件下并非最优。例如，在图 4b 中，当 α 较小时，自私矿工的最优策略是接受这两个诚实区块。但是，当 α 较大时，自私矿工的最优策略可能是在其区块 A_1 上挖矿，试图迎头赶上（也称为顽固挖矿 [25]）。我们建议对最优自私挖矿策略感兴趣的读者参阅[17, 25, 33]，了解更多详情。需要注意的是，由于挖矿难度的调整，区块生成率保持在每 10 分钟一个区块，因此在有攻击或无攻击的时期内，挖出的区块数量（和区块奖励）是固定的。因此，奖励分数越高，挖矿收入就越高。

2.3 比特币自私挖矿的理论结果

让我们从比特币自私挖矿的系统模型开始。该系统包含两种类型的矿工：一

种是诚实的矿工，他们遵守协议；另一种是自私的矿工，他们可能偏离协议以获取最大利润。让 α 表示自私矿工控制的计算能力的比例，β 表示诚实矿工控制的总计算能力的比例。在不失一般性的前提下，所有自私的

假定矿工由一个自私的矿工控制。挖矿过程被模拟为速率为 f 的泊松过程。因此，自私矿工以速率 αf 生成区块，诚实矿工以速率 βf 生成区块。诚实矿工产生的区块称为*诚实区块*，自私矿工产生的区块称为*对抗*区块。

在以下分析中，诚实区块到达所有矿工的时间被假定为可以忽略不计。这是因为区块间隔（即比特币中的平均 10 分钟）远大于传播延迟（通常为几十秒）。如前所述，γ 用来表示诚实矿工在自私矿工（而不是诚实矿工）产生的区块上挖矿的比例，无论何时

他们会观察到一个由两个等长分支组成的分叉。这里，γ 表示自私矿工的通信能力，假定范围为 [0，1]。此 外 ，由于一个区块中的交易费用通常比区块

因此，在以下分析中不考虑它们。

接下来，我们介绍 [11] 中的自私挖矿策略，如算法 1 所示。我们用 $L_s\ (t)$ (resp., $L_h\ (t)$) 表示自私矿工（resp., honest miners）在时间 t 看到的私有分支（resp., public branches）的长度。一开始，我们假设自私矿工和诚实矿工拥有同一条链的共识（第 1 行）。当自私矿工挖出一个新区块时（见第 2 行至第 8 行），它将

算法 1 比特币的自私挖矿策略

关于协商一致
1: $(l_s\ ,\ l_h\ ) \leftarrow (0,\ 0)$

上 自私的矿工挖出一个新区块 2：L_s
$\leftarrow L_s\ + 1$
3: **如果** $(L_s\ ,\ L_h\ ) = (2,\ 1)$ **那么**
4: 公布其私人分支机构
5: $(L_s\ ,\ L_h\ ) \leftarrow (0,\ 0)$ （因为所有矿工都达成了共识） 6:
否则
7: keep mining on its private
branch 8: **end if**

一些诚实的矿工挖出一个新区块 9：L_h
$\leftarrow L_h\ + 1$
10: **if** $L_s\ < L_h$ **then**
11: keep mining on this new
block 12： $(l_s\ ,\ l_h\ ) \leftarrow (0,$
$0)$
13: **else if** $L_s\ = L_h$ **then**
14: publish the block of the private
branch 15： $(l_s\ ,\ l_h\ ) \leftarrow (1,\ 1)$
16: **else if** $L_s\ = L_h\ + 1$ **then**
17: 公布其私人分支机构

18: $(L_s, L_h) \leftarrow (0, 0)$ （因为所有矿工都达成了共识）

19: **否则**

20:在其私有分支中 发布第一个未发布的区块

21： 设置 $(L_s, L_h) = (L_s - L_h, 0)$

22: **如果结束**

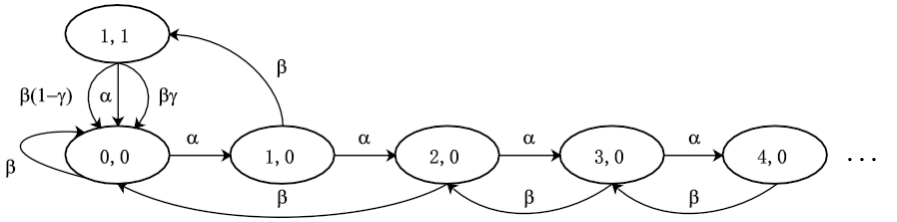

图 5 比特币自私挖矿的马尔可夫过程

保持该区块的私有性，并继续在其私有分支上挖矿，直到其优势非常有限（即 $(L_s, L_h) = (2, 1)$），这将在后面讨论。

当一些诚实的矿工挖出一个新区块时，公共分支的长度将我们有以下几种情况。情况 (1) 如果新的公共分支比私有分支长，自私的矿工就会采用公共分支，而上挖矿。(这就是为什么自私矿工会设置 $(L_s, L_h) = (0, 0)$ 的原因）。情况 (2) 如果新的公共分支与私有分支长度相同，自私矿工将立即公布其私有区块，希望尽可能多的诚实矿工选择其私有分支（因为在公布私有分支时，诚实矿工会看到两个长度相同的分支）。情况 (3) 如果新的公共分支比私有分支短一个，自私的矿工会公布其私有分支，这样所有诚实的矿工都会采用私有分支。情况 (4) 如果新的公共分支比私有分支短至少两个，自私矿工将公布第一个未公布的区块，因为自私矿工仍有明显优势。此外，由于自私矿工最终会在竞争中获胜（其所有区块为最长链），(L_s, L_h) 设为 $(L_s - L_h, 0)$。

通过上述自私挖掘策略，我们可以用 $(L_s(t), L_h(t))$ 来捕捉时*的*系统状态。如图 5 所示，很容易验证 $(L_s(t), L_h(t))$ 在我们的自私挖掘策略下是一个马尔可夫过程。[2]此外，过程 $(L_s(t), L_h(t))$ 是正循环过程，因此具有唯一的静态分布。

设 $\{\pi_{i,j}\}$ 为马尔可夫过程 $(L_s(t), L_h(t))$ 的稳态分布。通过求解上述马尔可夫过程，我们可以计算出稳态分布为如下面的 Lemma 所示。

定理 1（[11]） *给定算法 1 中的自私挖掘策略，状态的静态分布为*

[2] 在 Eyal 和 Sirer [11]中，自私挖掘过程的系统状态仅用 $L_s(t)$。在这里，我们使用 $(L_s(t), L_h(t))$ 作为系统状态，以便与以太坊的系统状态保持一致。

$$\pi_{0,0} = \frac{1 - 2\alpha}{2\alpha^3 - 4\alpha^2 + 1},$$
$$\pi_{1,0} = \frac{\alpha - 2\alpha^2}{2\alpha^3 - 4\alpha^2 + 1},$$
$$\pi_{1,1} = \frac{(1 - \alpha)(\alpha - 2\alpha)^2}{1 - 4\alpha^2 + 2\alpha^3},$$
$$\pi_{i,0} = \left(\frac{\alpha}{1 - \alpha}\right)^{k-1} \frac{\alpha - 2\alpha^2}{1 - 4\alpha^2 + 2\alpha^3}, \text{ 对于 } i \geq 1、 \quad (1)$$

有了状态空间的静态分布，我们接下来就可以分析自私矿工和诚实矿工分别获得的收益。为此

我们首先需要确定每个状态转换中已发布区块的奖励。例如，在从 (2，0) → (0，0) 的状态转换中，有一个诚实区块和两个敌对区块被公布。根据 LCR，自私的矿工将

而诚实的矿工则一无所获（见图 3 中的情况）。为简洁起见，我们不提供奖励分析，有兴趣的读者可参阅 [11]，了解更多详情。最后，我们可以通过下面的定理推导出自私的 矿工的相对收益。

定理 1（[11]） *给定算法 1 中的自私挖矿策略，自私矿工的相对收益为* $\frac{\alpha(1-\alpha)^2(4\alpha+\gamma(1-2\alpha))-\alpha^3}{1-\alpha(1+(2-\alpha)\alpha)}$。

当定理 1 中给出的自私矿工收益大于 α 时，自私矿工从自私挖矿策略中获得的收益将高于从诚实挖矿策略中获得的收益。

在此，我们通常假设 $0 \leq \alpha < 1/2$。因为当$\alpha \geq 1/2$时，自私的矿工可能会发起著名的双重花费攻击，从而获得更多收益[24]。我们可以

根据下面的推论，我们可以得出α*的*范围，从而使自私的采矿行为有利可图。

推论 1（[11]） *对于给定的 γ，拥有 α 部分计算能力的自私矿工可以在以下范围内获得比诚实矿工更大的收益：*

$$\frac{1 - \gamma}{3 - 2\gamma} < \alpha < \frac{1}{2}. \quad (2)$$

为了更好地理解定理 1 和推论 1，我们首先在图 6a 中绘制了不同 γ 下自私矿工的相对收益，α 在 0 到 0.5 之间。如前所述，当自私矿工只有一个私人区块并发布该区块以匹配诚实区块时，它就有可能失去区块奖励。在这种情况下，如果自私的矿工总是比诚实的矿工更快地传播其区块

(即 $\gamma = 1$），所有矿工都会在自私矿工的区块上挖矿。这意味着，自私矿工在启动自私挖矿策略时不承担任何风险，因此，$\gamma = 1$ 时，所有矿工都会在自私矿工的区块上挖矿。

任何计算能力的自私矿工都可以通过启动自私挖矿策略获益。计算能力的最小部分（称为

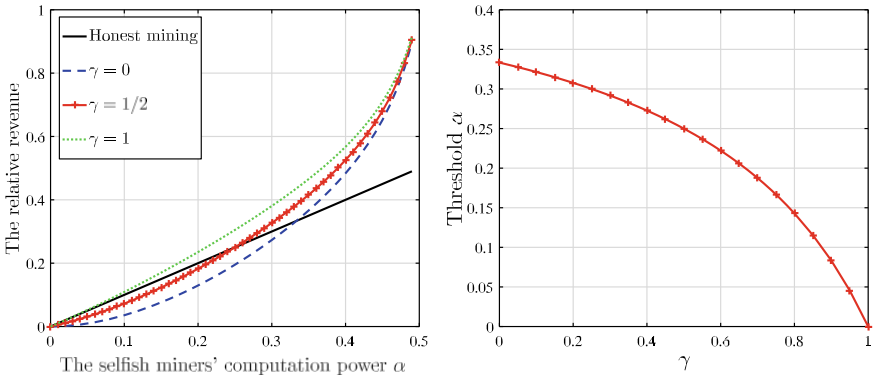

图 6 比特币中自私挖矿攻击的结果 **a** 不同 α 时自私矿工的相对收益。

相反，当 $\gamma = 0$ 时（诚实的矿工总是先发布并传播自己的区块），阈值为 $1/3$ 。因此

计算能力超过三分之一的自私矿工可以从自私挖矿攻击中获得更多收益。自私挖矿策略的阈值如图 6b 所示。

3 以太坊中的激励设计与分析

以太坊是市值第二大的加密货币，也是当今运行智能合约的最大去中心化平台。以太坊目前使用 NC 变体作为其基础共识，但其奖励设计与比特币不同，它提供了额外的叔侄两种奖励。[3]在本节中，我们将重点研究以太坊中的自私挖矿攻击，尤其是分析这两种奖励的影响。

3.1 以太坊概述

以太坊是一个基于区块链的分布式平台，可运行智能合约。粗略地说，智能合约是在图灵完备环境中定义的一组函数。以太坊的用户被称为客户端。客户可以发布交易来创建新合约，向社区或其他客户发送以太币（以太坊内部加密货币），或调用合约的某些功能。客户的交易随后被矿工收集到区块中。以太坊的矿工采用一种变体 NC

[3] 以太坊计划逐步用权益证明（PoS）取代 PoW。

表 1 以太坊和比特币的挖矿奖励

	以太坊	比特币	目的
区块奖励	C	C	补偿矿工的开采成本
奖励叔叔	C	×	减少采矿业的集中化趋势
奖励侄子	C	×	鼓励矿工参考叔叔区块
交易费（天然气成本）	C	C	交易执行；抵御网络攻击

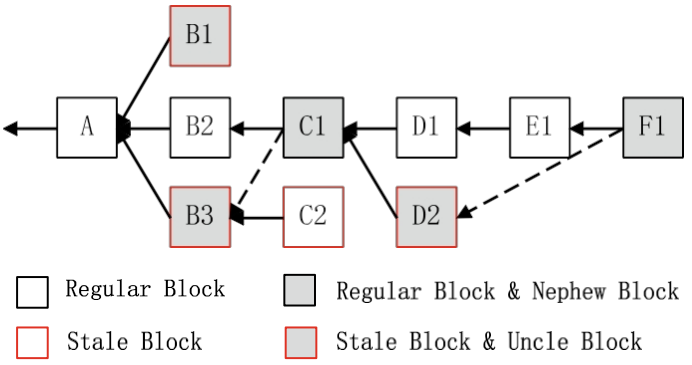

图 7 以太坊中的不同区块类型。这里，常规区块包括{ *A,* B2, *C* 1, D1, *E* 1, *F 1*}，陈旧区块包括{B*1,* B3, *C* 2, *D2*}。同样，叔叔区块为{B1、B3、D2*}*，侄子区块为{*C* 1、*F 1*}。叔叔区块 B3（叔叔区块 *D2*，相应的）的引用距离为 1（2，相应的）。

以就不断增长的区块序列（即区块链）达成协议。[4]此外，以太坊在参数设置上也与比特币不同；以太坊将区块大小设置为 20 KB 左右，区块间隔为 10-20 秒，而不是比特币的 1 MB 区块大小和 10 分钟区块间隔，以提高交易吞吐量。

以太坊的奖励有四种类型，即气体成本、区块奖励、叔叔奖励和侄子奖励[5, 9]，如表 1 所示。一方面，以太坊提供气体成本和区块奖励，这与比特币中的奖励类似。具体来说，气体成本用于奖励矿工在区块中包含和执行交易，相当于比特币中的交易费。此外，与比特币一样，以太坊的矿工一旦将其区块纳入所有矿工都接受的区块链中，就可以获得区块奖励。另一方面，以太坊引入了两种新奖励：叔侄奖励。

[4] 虽然以太坊声称应用了最重子树规则 [35]，但它似乎应用了最长链规则来选择主链 [17]。

为了解释这些奖励，我们将介绍*常规*区块和*陈旧区块的*概念。如果一个区块包含在主链中，则称为常规区块，否则称为陈旧区块。此外，陈旧区块是主链的 "直系子区块"。换句话说，"叔叔区块 "的父区块总是普通区块。如果一个 "叔叔区块 "通过引用链接被某个未来的常规区块（称为 "侄子区块"）引用，它就会获得一定的奖励。叔块和侄块的示意图见图 7。叔叔奖励的值取决于叔叔和侄子区块之间的 "距离"。由于所有区块都形成了一棵树，因此这个距离是明确定义的。例如，在图 7 中，叔叔区块 B3（叔叔区块 *D2*，下同）与其侄子区块之间的距离为 1（2，下同）。在以太坊中，如果距离为 1，则舅舅奖励为（静态）区块奖励的$\frac{7}{8}$；如果距离为 2，则舅舅奖励为区块奖励的$\frac{6}{8}$；以此类推。一旦距离大于 6，舅舅奖励将为零。相比之下，侄子的奖励总是$\frac{1}{32}$的区块奖励。

3.2 奖励设计及其对自私采矿的影响

在以太坊中，叔侄奖励最初是为了解决挖矿中心化的偏差--矿工们组成或加入一些大矿池。拥有巨大计算能力的矿池不太可能产生陈旧区块，挖矿的利润也会更高。因此，奖励陈旧区块会降低矿池的优势[8]，使其对小矿工的吸引力降低。

不幸的是，叔侄奖励也降低了发起自私挖矿的成本，从而降低了系统的安全级别。为了说明这一点，让我们用一个例子来说明自私的矿工如何从这些奖励中获益。回想一下自私的矿工先生产一个区块，然后诚实的矿工再生产两个区块的情况，如图 8a 所示。在比特币中，自私的矿工将接受诚实的区块 H_1 和 H_2，并通过算法 1 中的自私挖矿策略失去区块 A_1 的奖励。相比之下，在以太坊中，这个对抗性区块 A_1 可以被随后的诚实区块 A_2 引用，自私的矿工可以因此获得叔叔奖励（区块奖励的 7/*8*，因为区块距离为 1），如图 8b 所示。额外的叔叔奖励减少了自私矿工的损失。此外，自私矿工还可以通过引用叔叔区块获得侄子奖励。下一节，我们将通过 [12] 中的分析结果，系统地说明这些奖励对自私挖矿攻击的影响。

3.3 以太坊中自私挖矿的理论成果

我们采用第 2.3 节中介绍的模型。让 α 表示自私矿工控制的计算能力的比例，β 表示总计算能力的比例。

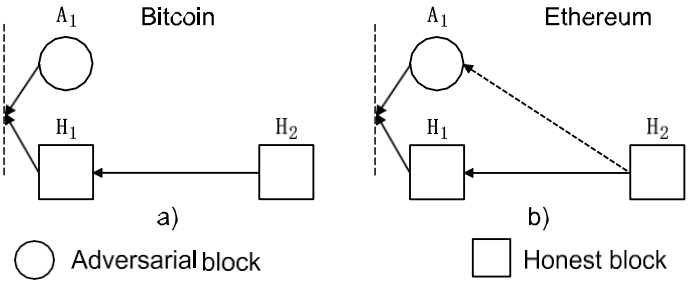

图 8 比特币和以太坊中自私矿工所获奖励的简单比较

由诚实矿工控制的权力。在不失一般性的前提下，假设所有自私的矿工都由一个自私的矿工控制。挖矿过程被建模为速率为 f 的泊松过程。因此，自私的矿工以速率 αf 生成区块，诚实的矿工以速率 βf 生成区块。我们用 K_s 、K_u 和 K_n 分别表示静态奖励、叔叔奖励和侄子奖励。不失一般性、

算法 2 以太坊中的自私挖矿策略

关于协商一致

1: $(l_s\ ,\ l_h\ ) \leftarrow (0,\ 0)$

关于自私的矿工开采新矿块

2: 根据其私有分支引用所有（未引用的）叔叔区块 3: $L_s \leftarrow L_s + 1$

4: **如果** $(L_s\ ,\ L_h\ ) = (2,\ 1)$ **那么**

5: 公布其私人分支机构

6: $(L_s\ ,\ L_h\ ) \leftarrow (0,\ 0)$ （因为所有矿工都达成了共识） 7:

否则

8: keep mining on its private

branch 9: **end if**

关于一些诚实的矿工开采出一个新区块

10:矿工根据其公共分支引用所有（未引用的）叔叔区块 11： $L_h \leftarrow L_h + 1$

12: **if** $L_s < L_h$ **then**

13: $(l_s\ ,\ l_h\ ) \leftarrow (0,\ 0)$

14: keep mining on this new

block 15： **else if** $L_s = L_h$

then

16: Publish the last block of the private

branch 17： **else if** $L_s = L_h + 1$ **then**

18: 公布其私人分支机构

19: $(L_s\ ,\ L_h\ ) \leftarrow (0,\ 0)$ （因为所有矿工都达成了共识） 20

： **否则**

21: 发布其私人分支中第一个未发布的区块

22： 设置 $(L_s\ ,\ L_h\ ) = (L_s - L_h + 1,\ 1)$ 如果新区块是在公共分支上开采的，而公共分支是私有分支的前缀

23: **end if**

我们假设 $K_s = 1$，因此 K_u (K_n , resp.) 代表叔叔奖励（侄子奖励，respect.）与静态奖励的比率。特别是，$K_n < K_u < 1$，而 K_u 是一个距离的函数。

现在我们在算法 2 中描述诚实矿工和自私矿工的挖矿策略，该算法基于算法 1 中比特币的自私挖矿策略。同样，我们用 $L_s(t)$ ($L_h(t)$) 表示自私矿工（或诚实矿工）在 t 时刻看到的私有分支（或公共分支）的长度。算法 1 和算法 2 的主要区别在于，在挖掘区块时，矿工会尽可能多地加入与（未引用的）叔叔区块的引用链接，从而获得尽可能多的叔叔和侄子奖励（见第 2 行和第 10 行）。此外，为了追踪自私矿工和诚实矿工赢得的叔侄奖励，私人和公共分支的长度也会被保留下来（见第 21 和 22 行）。更多详情请读者参阅 [26]。

通过该算法，我们可以用 $(L_s(t), L_h(t))$ 来捕捉 t 时刻的系统状态。在自私挖掘策略下，状态 $(L_s(t), L_h(t))$ 演化为马尔可夫过程，如图 9 所示。通过与图 5 中自私挖掘攻击的马尔可夫过程比较，不难看出系统状态的复杂性增加了，这是因为叔和侄子的奖励。同样，过程 $(L_s(t), L_h(t))$ 是正循环的，因此它有唯一的静态分布。设 $\{\pi_{i,j}\}$ 为马尔可夫过程 $(L_s(t), L_h(t))$ 的稳态分布。通过求解马尔可夫过程，我们可以得到静态的的状态分布。

定理 2（[12]） *给出算法 2 中的自私挖掘策略，状态的静态分布为*

$$\pi_{0,0} = \frac{1-2\alpha}{2\alpha^3 - 4\alpha^2 + 1}, \quad \pi_{1,1} = \alpha - \alpha^2 \quad \pi_{0,0}, \quad \pi_{i,0} = \alpha\, \pi^{i}_{,0,0}, \text{ for } i \geq 1$$

$$\pi_{i,j} = \alpha^{i} (1-\alpha)^{j} (1-\gamma)^{j} \quad f(i,j,j)\pi_{0,0} + \alpha^{i-j} \gamma (1-\gamma)^{j-1} \frac{1}{(1-\alpha)^{i-j-1}} - 1 \quad \pi_{,0,0}$$

$$\gamma (1-\gamma)^{j-1} \sum_{k=1} \alpha^{i-k} (1-\alpha)^{j-k} \quad f(i,j,j-k)\pi_{0,0}, \text{ for } i \geq j+2, j \geq 1.$$

函数 $f(x, y, z)$ 是多重求和，定义如下

$$f(x,y,z) = \begin{cases} \sum_{s_x} \quad {}^{s}{}_{s_z} \quad {}^{2} \\ S_z = Y+2 \; S_{z-1} \quad .. \quad 1, \; z \geq 1, \; x \geq y+2, \\ =Y+1 \qquad\qquad s_1 = y-z+3 \\ \qquad\qquad z \\ c0, \qquad\qquad \text{否则。} \end{cases} \tag{3}$$

接下来，我们对每个状态转换进行奖励分析。在此，我们采用概率方法跟

踪各种区块奖励 [26]。具体来说，在每个状态转换中，都会出现一个新区块（由诚实的矿工或自私的矿工挖出）。由于新区块的 "命运 "取决于系统的演化，因此无法确定新区块刚刚创建时的相关奖励数量。因此，新区块的预期奖励是根据可能出现的情况计算出来的。

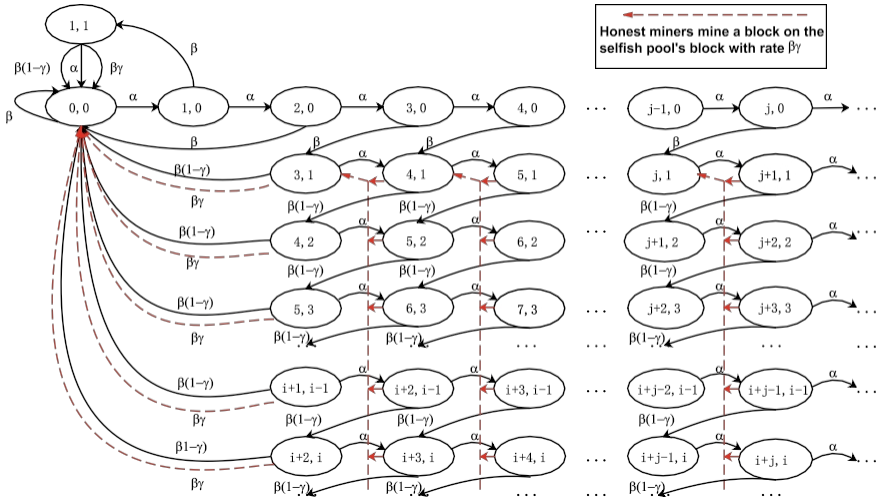

图 9 以太坊中自私挖矿的马尔可夫过程

未来事件。相比之下，比特币中的自私挖矿分析跟踪的是与状态转换（其命运已经确定）相关的已发布区块，而不是新区块，因此可以计算出准确的奖励[11]。这就产生了以下两个问题：

1. 跟踪已发布的区块有什么问题？
2. 如何计算新区块创建时的预期回报？

要回答第一个问题，我们应该注意到，跟踪发布的区块并没有提供足够的信息来计算叔侄奖励。请回顾第 3.1 节。回顾第 3.1 节，已发布的普通区块可以通过引用未发布的叔叔区块来获得侄子奖励。侄子奖励的数量取决于未发布的叔叔区块的数量。因此，有必要跟踪系统中所有未完成的叔叔区块及其深度信息（这是确定叔叔奖励数量所必需的）。这大大增加了状态空间的复杂性。

要回答第二个问题，我们将注意到，只需利用以下信息计算一个新区块的预期奖励：它成为普通区块的概率、它成为叔叔区块的概率、它与潜在侄子区块的距离（如果它确实成为叔叔区块）。也许有点出人意料的是，所有这些信息都可以在算法 2 中的自私挖掘策略生成新区块时确定。

为了更好地理解这种概率方法，我们可以看一个简单的例子，如图 10 所示。假设自私的矿工已经挖出了两个区块，并在时间 t 时将其保密。然后，

某个诚实的矿工生成了一个新区块（图 10 中的情况 a）。根据算法 2，自私矿工会立即公布其私有分支。因此，这个新区块将以概率 1.此外，我们还可以看到，这个区块的距离为 2，其电位为

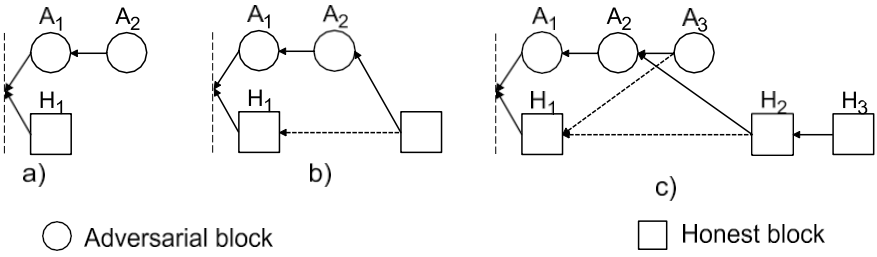

图 10 概率奖励计算方法简图

侄子区块。因此，这个新区块将获得 $K_u(2)$ 的叔叔奖励。同样，它的潜在侄子区块将获得 $K_n(2)$ 的侄子奖励。此外，这种奖励将属于一些诚实的矿工，概率为 β（图 10 中的情况 b）和 $\alpha\beta^2(1-\gamma)$（图 10 中的情况 c）。显然，自私矿工获得该奖励的概率为 $1-\beta(1+\alpha\beta(1-\gamma))$。(详见附录 B 中的*案例 7* [12]。）因此，与这个新区块相关的预期奖励总共为 $K_u(2)+K_n(2)$，其中 $K_u(2)+K_n(2)\beta(1+\alpha\beta(1-\gamma))$ 奖励将属于诚实的矿工（其余奖励将属于自私的矿工）。我们将感兴趣的

读者可参阅 [12] 了解奖励分析的更多详情。

通过奖励分析，很容易计算出区块奖励 r_b^s (resp., r_b^h), uncle 奖励 r_u^s (resp., r_u^h), 而外甥奖励 r_n^s (resp., r_n^h) 给自私的矿工 (resp.....)、诚实的矿工）的定理。

定理 2（[12]） *给出算法 2 中的自私挖矿策略，自私矿工和诚实矿工的奖励分别为*

$$r_b^s = \frac{\alpha(1-\alpha)^2(4\alpha+\gamma(1-2\alpha))-\alpha^3}{2\alpha^3-4\alpha^2+1},$$

$$r_b^h = \frac{(1-2\alpha)(1-\alpha)(\alpha(1-\alpha)(2-\gamma)+1)}{2\alpha^3-4\alpha^2+1},$$

$$r_u^s = \frac{(1-2\alpha)(1-\alpha)^{2\alpha}(1-\gamma)}{2\alpha^3-4\alpha^2+1}K_u(1),$$

$$r_u^h = (\alpha\beta+\beta^2\gamma)K_u(1)\pi_{+1,0}+\sum_{i=2}^{\infty}\beta K_u(i)\pi_{+i,0}+\sum_{i=2}^{\infty}\sum_{j=1}^{\infty}\beta\gamma K_u(i)\pi_{i+j,j}\ 、$$

$$r_n^s = \alpha\beta K_s(1)\pi+\beta_{1,0}\sum_{i=2}^{\infty}\sum_{j=1}^{\infty}\gamma^{i-1}(\alpha-\alpha\beta^2(1-\gamma))K_s(i)\pi_{i+j,j}\ 、$$

$$r_n^h = \alpha\beta^2(1-\gamma)K_s(1)\pi_{0,0}+\beta^2\gamma K_s(1)\pi+\beta_{1,0}+\sum_{i=2}^{\infty}\sum_{j=1}^{\infty}\gamma^i(1+\alpha\beta(1-\gamma))K_s(i)\pi_{i+j,j}\ 。$$

有了这个定理，就很容易得出公式（4）中自私矿工的绝对收入 $U_s(\alpha,\gamma)$。请注意，绝对收益等同于比特币中的相对收益（即份额 $R_s(\alpha,\gamma)$），但它

不同于比特币中的相对收益。

以太坊由于存在叔侄奖励。这是因为比特币会调整挖矿难度，使常规区块以稳定的速度生成，比如每个时间单位生成一个区块。因此，不管有没有私心挖矿，长期平均总收入都固定为每个时间单位一个区块奖励。这就使得弃权收入等同于相对收入。以太坊的情况则不同。即使常规区块的生成速度稳定，平均总收入仍然取决于大叔区块的生成速度，而大叔区块的生成速度会受到自私挖矿攻击的影响。事实上，在拜占庭硬分叉之前，以太坊在调整难度级别时并没有考虑舅舅区块的生成率。因此，有两种情况：(1) 普通区块的生成率为每个时间单位 1 个区块；(2) 普通区块和叔叔区块的生成率为每个时间单位 1 个区块。

在上述分析中，常规块生成率为 $r_b^s + r_b^h$，小于大于 1，如前所述。因此，可以重新调整时间，使常规区块生成率为每个时间单位生成 1 个区块。在这种情况下，自私池的长期绝对收益为

$$U_s(\alpha, \gamma) = \frac{R_b^s + R_u^s + R_n^s}{r_b^s + r_b^h}, \tag{4}$$

而诚实矿工的长期绝对收入是

$$U_h(\alpha, \gamma) = \frac{R_b^h + R_u^h + R_n^h}{r_b^s + r_b^h}。 \tag{5}$$

同样，也可以对时间进行缩放，使普通区块和叔叔区块的生成率为每个时间单位 1 个区块，并相应地定义自我区块池和诚实矿工的长期绝对收入。最后，可以推导出在以太坊中使自私挖矿有利可图的计算能力阈值。具体来说，如果自私矿工遵循挖矿协议，其长期平均绝对收益将为α，因为网络延迟可以忽略不计（因此不会出现陈旧区块）。如果矿工采用算法 2 中的自私挖矿策略，则其长期绝对收益由 $U_s(\alpha, \gamma)$ 得出，可能大于 α。

$\min_\alpha \{U_s(\alpha, \gamma) > \alpha\}$ $\min_\alpha$ 和 $\{R_s(\alpha, \gamma) > \alpha\}$ 通过数值计算可[illegible]得出两种情况下的{U $(\alpha, \gamma) > \alpha$}。

为了更好地显示结果，我们在图 11a 中绘制了自私矿工在 α 为 0 至 0.5 时的预期绝对收入。在这里，由于采用了统一的平局打破策略，$\gamma = 1/2$。从中我们可以看出，当自私矿工控制了超过 0.163 计算能力，它可以从自私的挖矿攻击中获得比从诚实挖矿。在比特币中，阈值比这更低（即当 $\gamma = 1/2$ 时为 0.25）。换句话说，额外的叔侄奖励降低了系统的安全性。

水平。图 11b 显示了不同 γ 时的攻击阈值。我们特别计算了两种情况下的阈值：(1) 正常区块生成率为每个时间单位生成 1 个区块；(2) 正常区块和叔叔区块生成率为每个时间单位生成 1 个区块。

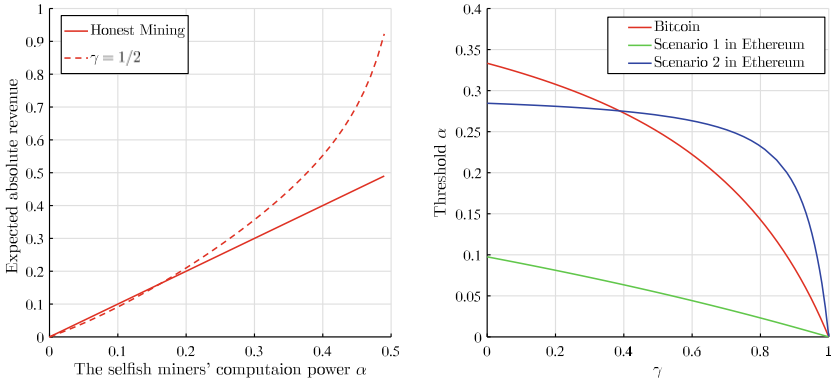

图 11 以太坊中自私挖矿攻击的结果 **a** 不同 α 时自私矿工的相对收益。

时间单位。从图 11b 可以看出，γ 越大，自私挖矿盈利所需的哈希值就越低。特别是当γ=1时，比特币和以太坊的自私挖矿无论哈希值大小都能盈利。除此之外

结果显示，以太坊在场景 1 中的哈希值阈值总是低于比特币。相比之下，当γ≥0.39时，场景2中的哈希值阈值高于比特币。这是因为 γ *越大*，挖出的区块就越多。
诚实的矿工都是 "叔叔 "区块。然而，在方案 2 中，额外引用的
叔叔区块会降低普通区块的生成率，导致自私矿池的区块奖励减少。因此，自私矿池需要有更高的哈希值，才能使自私挖矿有利可图。这表明，在算法 2 所给出的挖矿策略下，以太坊应在难度调整中考虑舅舅区块。

4 Bitcoin-NG 中的激励设计与分析

比特币自诞生以来，一直存在吞吐量低（即每秒 7 笔交易）和延迟长（即确认交易需要约一小时）的问题。糟糕的性能严重阻碍了区块链的应用，因此人们提出了许多可扩展的区块链协议[1, 10, 22, 29, 35, 42]。其中，Bitcoin-NG（下一代）是最早也是最杰出的基于 NC 的区块链之一，其吞吐量接近*最优*[10]。比特币-NG 创造性地采用了两种类型的区块：（1）*关键区块*，它与比特币中的传统区块非常相似，只是不携带任何交易；（2）*微区块*，它携带交易。每个关键区块都是通过 NC 中的领导者选举过程（通常称为挖矿过程）生成的。每个领导者可以发布多个微区块并收取交易费用，直到下一

个关键区块生成。与比特币不同、

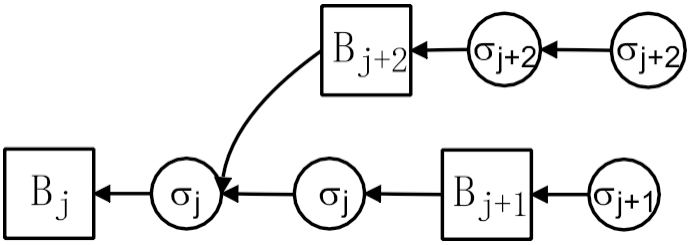

图 12 Bitcoin-NG 的示意图。方形（分别为圆形）区块表示关键区块（分别为微型区块）。微块由三个密钥块矿工 B_j，B_{j+1}，B_{j+2} 签发，他们的签名分别是 σ_j，σ_{j+1}，σ_{j+2}。

Bitcoin-NG 将领导者选举和交易序列化分离开来。直观地说，正是这种解耦使得 Bitcoin-NG 的吞吐量接近最优，因为微块的生产速度可以达到网络容量。在本节中，我们将重点讨论 Bitcoin-NG 的激励设计，特别是研究这种解耦思想对基于激励的攻击的影响。

4.1 Bitcoin-NG 概览

在比特币中，区块的挖掘有两个功能：（1）通过 NC 选举领导者（即有效区块的所有者）；（2）对交易进行排序和验证。通过区分区块功能，Bitcoin-NG 将领导者选举与交易序列化分离开来。具体来说，Bitcoin-NG 使用通过 PoW 挖掘出的关键区块，以稳定的速率（如每 100 秒一个关键区块）选举领导者。每个领导者可以以另一种速率（通常高于密钥块速率，例如每 20 秒一个微块）产生包含未确认交易的多个微块。简而言之，关键区块与比特币中的传统区块非常相似，只是它不包含任何交易。另一方面，微块包含交易，但不包含任何工作证明。虽然微块的生成率通常比密钥块的生成率要高得多，但必须对其进行约束，以防止敌对领导者用微块淹没系统。这种解耦使 Bitcoin-NG 能够在两个连续的密钥块之间处理许多微块，从而大大提高了交易吞吐量。图 12 展示了这两种类型的区块。Bitcoin-NG 采用了与比特币类似的分叉选择规则。在 Bitcoin-NG 中，微区块没有权重，甚至没有二级索引供矿工选择开采哪个关键区块。例如，在图 12 中，有两个关键区块数量相同但微区块数量不同的分叉分支。矿工将这两个分叉分支视为等同，采用统一的平分规则选择其中一个分支，然后在该分支中最新的微区块之

后进行挖矿[10]。总之，诚实的矿工仍然按照 LCR 选择一个 "正确的 "关键区块（即仅由关键区块组成的最长链中的最后一个关键区块），然后在该分支上挖矿。

关键区块矿工产生的最新微区块。因此，在没有微块的情况下，关键区块的挖掘过程与比特币的挖掘过程是一样的。比特币中的自私挖矿攻击也可以用来攻击 Bitcoin-NG 中的关键区块。与比特币类似，Bitcoin-NG 也提供两种奖励，即关键区块奖励和交易费。如果每个矿工通过成功解决 PoW 谜题挖掘出一个关键区块，并且其关键区块最终出现在最长的链上，那么它就能获得一个关键区块奖励。挖出一个关键区块后，矿工还可以通过在自己的微区块中包含尽可能多的交易（不超过微区块大小限制）来获得交易费。

4.2 微块及其基于激励的攻击

在 Bitcoin-NG 中，矿工应在其微区块中包含尽可能多的交易，并发布这些微区块以赢得交易费。这就是所谓的交易包含规则。此外，矿工应尽可能多地接受前一个关键区块矿工发布的微区块，并在最新收到的微区块上挖矿，即遵守最长链延伸规则。不难看出，当所有矿工都遵守这两条规则时，就会有更多产生的微区块被纳入区块链，比特币-NG 就能实现更好的交易吞吐量。相反，自私的矿工可能会破坏交易收录规则和最长链延伸规则，以从交易费中获取最大利润，如下所述：

- **交易包含攻击。**当自私的矿工发布一个关键区块并生成多个微区块时，它会将最后几个微区块保密。这

  也就是说，自私的矿工继续在其最新的微区块链上挖矿，而诚实的矿工只能在最后公布的微区块链上挖矿。图 13 显示了这样一种情况：自私的矿工扣留了其在关键区块 B_j 之后开采的部分微区块，而诚实的矿工则在自私矿工的最后一个公开微区块上挖矿。如果微区块的交易费主要归下一个关键区块所有者所有，那么这种攻击就会受到激励。
- **最长链扩展攻击。**当自私矿工采用诚实密钥区块时，它可以拒绝部分（或全部）微区块，直接在最后一个被接受的区块上挖矿。

  微区块（或最后一个密钥区块）。换句话说，自私的矿工拒绝接受上一个诚实的密钥-区块矿工在这些微区块中发布的交易。图 14 展示了这种攻击。如果交易费主要归当前密钥区块所有者所有，那么这种攻击就会受到激励。

从上述情况不难看出，微区块中的交易费既不能归下一个键块所有者所有，也不能主要归当前键块所有者所有。因此，为了抵御这两种攻击，Bitcoin-NG 将两个连续的关键区块矿工之间的微区块中包含的交易费用分为两部分

部分。第一个密钥块挖掘者获得 r 分数（$r \in [0, 1]$），而第二个密钥块挖掘者获得

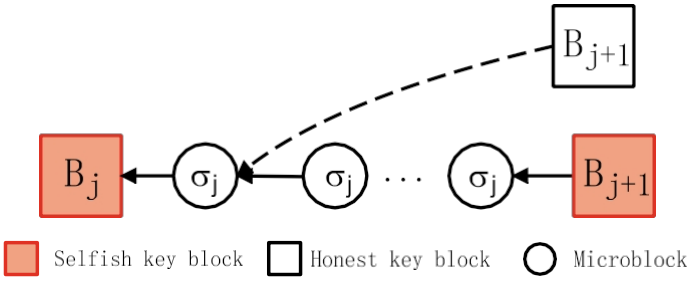

图 13 交易包含攻击的示例。自私的 B_j 之后的前两个微块已经公布，因此对诚实的矿工来说是公开的。其他微区块则不公开。虚线方形区块表示未来挖出的区块

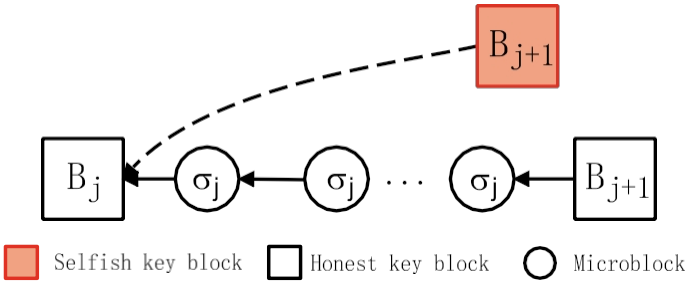

图 14 最长链扩展攻击示例。自私的矿工拒绝所有微区块，并在诚实的 B_j 上开采其关键区块。虚线方形区块表示未来挖出的区块

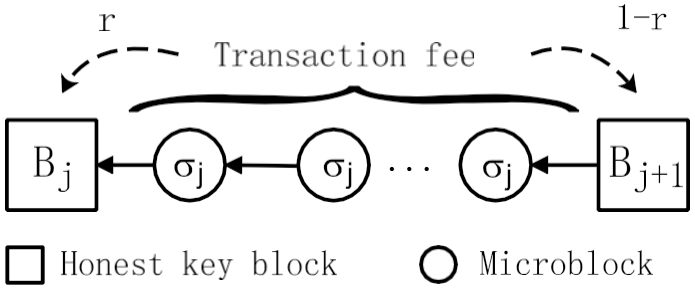

图 15 Bitcoin-NG 费用分配规则

获得剩余的 1 - r 分数。图 15 展示了这一费用分配规则。下面的分析说明了如何决定 r 值来抵制微块

采矿攻击。

抵御交易包容攻击。在这种攻击中，自私的矿工可以持有一个微区块，以避免与后续的密钥区块矿工分享交易费用。(请注意，自私的（分别是诚实的）矿工挖出一个区块的概率是 α（respec- α）。

即 $\beta = 1 - \alpha$）。为了保证发起上述攻击的自私矿工的平均收益小于其应得收益，分配比例 r 应满足以下条件

$$\alpha \times 1 + (1-\alpha) \times \alpha \times (1-r) < r, \quad (6)$$

（100% 赢；输 100%，但我是在 txn 之后输的）

因此 $r > \frac{1-\alpha}{2+\alpha-\alpha^2}$。这一比例要求鼓励自私的矿工将交易放在公共微区块中。

后来，Yin 等人[41] 发现上述计算忽略了一种情况：现任领导者可以再次当选为下一任领导者，并额外获得交易费的$\alpha(1-r)$部分。因此，分配比例 r 应满足

$$\alpha \times 1 + (1-\alpha) \times \alpha \times (1-r) < r + \alpha(1-r), \quad (7)$$

（100% 赢；输 100%，但我是在 txn 之后输的）

因此 $r > \frac{\alpha}{1-\alpha}$。

抵御最长链扩展攻击 为了增加某些交易的收入，自私的矿工可以忽略诚实微区块中的这些交易，而在前一个微区块中挖矿。之后，如果自私矿工挖到了一个关键区块，它就可以把这些交易放到自己的微区块中。为了抵御这种攻击，自私矿工在这种情况下的收益必须小于遵守最长链扩展规则所获得的收益。因此，我们有

$$\alpha \times 1 + \alpha^2 \times (1-r) < \alpha(1-r), \quad (8)$$

（我的下一个关键区块；开采第三块钥匙；我的微型块）

这导致 $r < 1-\alpha$。考虑到上限，分布为比率 r 满足 $\frac{1-\alpha-\alpha^2}{2+\alpha-\alpha^2} < r < 1-\alpha$。特别是，当 α 小于 25% 时，我们得出 $\frac{1-\alpha}{2-\alpha}$。因此，在 Bitcoin-NG [10] 中选择 $r = 40\%$。

讨论。上述分析提供了关于 r 值的理论结果，但它也存在以下问题有几个局限性。为了更好地理解它，让我们重放最长链扩展攻击，如图 16 所示。为了更好地说明分析的局限性，这里做了两个简化：1）每个领导者只允许创建一个微块；2）每个微块只允许包含一笔交易。考虑这样一种情况：诚实的矿工生成一个关键区块 B_j 以及一个包含以下内容的微区块

交易 tx。如果自私的矿工遵守最长链扩展规则，并以概率 α 找到下一个关键区块，它将获得交易费的 $1-r$ 部分（相当于公式 (8) 中的最后一项）。然而，自私的矿工可以

直接在关键区块 B_j 上挖矿，希望赢得更高的交易费 tx。如果自私的矿工恰好以 α 的概率创建了下一个关键区块 B_{j+1}，那么它就可以通过在其微区块中包含 tx 来赢得交易费的 r 部分（这部分交易费为 0.5%）。

对应于公式 (8) 中的第一项）。如果自私的矿工幸运地挖到了下一个连续的关键区块 B_{j+2}，它将赢得剩余 1 - r 的交易费。综合所有条件可得出公式 (8)。

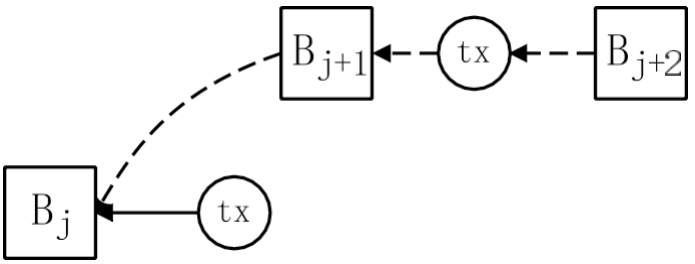

图 16 一个简单的例子说明了前面分析的局限性。自私的矿工直接在区块 B_j 上挖矿，并试图将交易 *tx* 纳入其未来的微型区块中

上述例子的分析有两个局限性。首先，如果自私的矿工只是希望从目标交易 *tx* 中获得更高的费用，那么简单的分析是非常合理的。然而，在现实中，自私的矿工通常希望从所有交易中增加收入，而不仅仅是目标交易。如果自私矿工将该策略应用于所有交易，而不是单个目标交易，那么它将很快耗尽其未来微块的空间。因此，自私的矿工无法在其微区块中包含其他交易，从而损失了相关的交易费。换句话说，上述分析忽略了交易规模和微区块容量的影响，这放大了自私矿工从攻击中获得的潜在收入。另一方面，上述分析对于高额费用的鲸鱼交易非常有效，因为这种交易很少见，所以我们不必担心空间问题。但是，在当前的区块链系统中，大多数交易的费用都很低。

其次，假定自私矿工总是采用诚实矿工的关键区块，并立即发布其新的关键区块（即对关键区块的诚实挖掘）。换句话说，它不考虑关键区块自私挖掘的影响。只有当自私挖矿者的计算能力小于使关键区块自私挖矿有利可图的阈值时，这一假设才是合理的，因为关键区块的最优挖矿策略确实是诚实挖矿[10, 11, 33]。然而，一旦自私矿工的计算能力α超过阈值，自私矿工就有动机发起关键区块挖掘攻击，因此关键区块自私挖掘的影响就不容忽视了。下面，我们将通过[27]中的激励分析来解决上述两个局限性。

4.3 微块挖掘的理论成果

我们使用第 2.3 节中的挖矿模型。让 α 表示由自私矿工控制的计算能力比例，β 表示由诚实矿工控制的总计算能力比例。在不失一般性的前提下，假定所有自私矿工都由一个自私矿工控制。密钥块挖掘过程被建模为速率为 f 的

泊松过程。因此，自私矿工生成密钥块的速率为 $f\alpha$，诚实矿工生成密钥块的速率为 $f\beta$。矿工

每个关键区块的挖矿者成为领导者，可以以恒定的速率 v 发布一系列包含尽可能多交易（最大微区块大小）的微区块，直到挖出下一个关键区块。具体来说，由诚实（分别是自私）矿工挖出的区块（包括关键区块和微区块）被称为*诚实*（分别是*自私*）区块。

按照比特币的网络模型[11, 12]，我们假设诚实的矿工通过底层网络完全连接，诚实的矿工在 Bitcoin-NG 中广播一个密钥区块或微区块所花费的时间可以忽略不计。[5]此外，我们假设自私的矿工在看到新的诚实密钥区块后，可以立即广播其私有区块。

我们根据交易费用假设了两类交易：收费高的 "鲸鱼 "交易和收费低的普通交易。此外，我们还假设绝大多数交易都是普通交易。我们假设交易规模是固定的，因此微区块中包含的最大交易数量也是固定的。我们还假设矿工有足够多的待处理交易可以被纳入微区块。[6]如果一个微区块只包含常规交易，我们将其称为常规微区块。此外，我们将常规微区块中包含的总交易费用称为微区块费用，并使用 R_t 表示。此外，我们使用 R_b 表示密钥块奖励。让 $k = R_b / R_t$ 表示区块奖励与微区块费用的比率。这个比率 k 的范围是 $(0, \infty)$。当 k 接近 0（分别为 ∞）时，意味着交易费（分别为密钥区块奖励）在奖励中占主导地位。不同的 k 值显示了奖励的不同影响

在比特币-NG 系统上。由于鲸鱼交易非常罕见，它们占用的微块空间很小。因此，我们可以忽略其空间需求（即使在网络容量限制下），并应用第 4.2 节中的分析。4.2.

接下来，我们介绍微型区块的激励分析。特别是，该分析不考虑自私的关键区块挖掘。也就是说，它假设自私的矿工总是采用诚实矿工的关键区块，并立即发布自己的新关键区块。这一假设很快就会得到证明，并将在第 4.4 节中通过考虑微区块和关键区块的联合挖掘而得到放宽。4.4.此外，由于密钥区块的发布延迟可以忽略不计，因此在下面的分析中也不考虑分叉密钥区块。

我们考虑的是自私矿工和诚实矿工在时间间隔 $[0, t]$ 内正常交易的交易费收入。在不失一般性的前提下，我们假设存在一个区块 B_0，自私的矿工和诚实的矿工都可以通过该区块进行交易。

矿工都同意在起始时间开采。(例如，B_0 可以是创世区块。）设 $M(t)$ 为时间区间 $[0, t]$ 内挖出的关键区块数量。让 X_i $(i \in [0, M(t)])$ 表示一个指示性随机

变量。

如下所述，第 i - 个密钥块是一个自私的密钥块：

---

[5] 对于关键数据块来说，这一假设是合理的，因为两个连续关键数据块的到达时间往往比数据块传播延迟大得多。另一方面，对于微块来说，这一假设可以放宽，我们将在下文中说明。

[6] 这一假设在比特币和类似以太坊的公共区块链中是合理的。例如，mempool 可视化网站 [21] 显示，2021 年 5 月的待处理交易数量约为 136k。

$$X_i = \begin{cases} 1, \text{selfish key block} \\ 0, \text{诚实的密钥块。} \end{cases}$$

在不失一般性的前提下，我们假设区块 B_0 是一个诚实的关键区块。对于其他密钥块，它是自私密钥块的可能性等于 α。

挖出一个密钥区块后，其所有者可以以恒定的速率 v 发布一系列微区块，直到挖出下一个密钥区块。这里，速率 v 反映了网络容量约束。让 Y_i 表示第 i - 个关键区块和第 $(i+1)$- 个关键区块之间的间隔。因此，第 i 个和第 $(i+1)$- 个关键区块之间产生的微区块数量为 vY_i。此外，每个微块包含的总费用为 R_t，因为我们只

这里考虑的是常规交易。现在我们准备计算合适的 r 值，以抵御针对常规交易的两种微块攻击。

抵御交易包容攻击。在这种攻击中，自私的矿工会隐藏其在关键区块之后生成的部分微区块，但继续在微区块链顶端挖矿。因此，诚实的矿工会直接在自私矿工的微区块链上挖矿。

最后公布的区块。让 ρ 表示两个连续关键区块之间所有自私微区块中未公布微区块的比例。其中，$\rho = 1$ 表示自私矿工隐藏了它在两个连续关键区块之间生成的所有微区块。因此，如果任意两个连续的关键区块满足 $(X_i, X_{i+1}) = (1, 0)$，它们之间有 $(1-\rho)vY_i$ 微块，从诚实的矿工；否则，就有 vY_i 个微型区块。

让 Z_i 表示一个指标随机变量，如果 $\{X_i = 1, X_{i+1} = 0\}$ 则等于 1，否则等于 0。接下来，让 $Z = \sum_{i=1}^{M(t)-1} Z_i$。假设 $M(t) = m$。下面的 Lemma 将帮助我们高概率地约束 Z 的值。

定理 3 *对于 m 个连续的密钥块，块对的数量 $(X_i, X_{i+1}) = (1, 0)$ 有如下切尔诺夫型约束：对于 $0 < \delta < 1$、*

$$\Pr(|Z - \alpha\beta(m-1)| > \delta\alpha\beta(m-1)) < e^{(-\blacktriangle\delta^2\alpha\beta m)}. \quad (9)$$

该 Lemma 表明，随着 m 的增加，密钥对 $(X_i, X_{i+1}) = (1, 0)$ 的数量很有可能介于 $(1-\delta)\alpha\beta m$ 和 $(1+\delta)\alpha\beta m$ 之间。

接下来，我们计算大 m 时自私矿工的相对收益。在一方面，Σ所有矿工的交易费总额为 $\sum_{i=1}^{m-1} (vY_i R_t - \rho v Z_i Y_i R_t)$.要了解这一点，请注意有 $\sum_{i=1}^{m-1} vY_i$ 微块产生的相关交易费用 $\sum_{i=1}^{m-1} vY_i R_t$。还请注意，一旦 $Z_i = 1$，有 ρvY_i 个微块没有包含在最长链中，原因是交易包含攻击。因此，相关的交易费损失为 $\sum_{i=1}^{m-1} \rho v Z_i Y_i R_t$。另一方面，自私矿工的总交易费为由 $\sum_{i=1}^{m-1} (\alpha v Y_i R_t - r\rho v Z_i Y_i R_t)$ 给出。要了解这一点，请注意，在没有任何攻击的情况下，$\sum_{i=1}$

自私矿工可以获得总交易费用的 α 部分，其计算公式为 $^{m-1}\alpha v Y_i R_t$。还需要注意的是，在交易包含攻击中，自私的矿工将损失 r

交易费用总损失的一部分。综合上述分析，我们可以得出以下关于大 m 的 Lemma。

下式 **4** *自私矿工的相对收入 u 趋近于 $\frac{\alpha-r\alpha\beta\rho}{1-\alpha\beta\rho}$，且具有较高的当 $m \to \infty$ 时的概率。*

该 Lemma 指出，对于较大的 m，自私矿工的相对收益为 $\frac{(\alpha-r\alpha\beta\rho)}{(1-\alpha\beta\rho)}$。回想一下，密钥块生成过程是一个速率为 f 的泊松过程，因此 $M(t)$ 是一个泊松到达过程。因此，当 t 趋于无穷大时，$M(t)/t \to f$ 大概率成立。因此，在很大概率上，自私矿工在 [0, t]期间的最大相对收益为

$$u = \max_{0\le\rho\le 1} \frac{\alpha - r\alpha\beta\rho}{1 - \alpha\beta\rho} = r + \max_{0\le\rho\le 1} \frac{\alpha - r}{1 - \alpha\beta\rho}. \tag{10}$$

如果 $r \le \alpha$，则最优 $\rho = 1$，相应的

$$u = r + \frac{\alpha - r}{1 - \alpha\beta}。$$

在这种情况下，u 总是大于 α，因为 $1 - \alpha\beta < 1$。这意味着自私的矿工利用这种攻击总是能获得大于其公平份额的相对收益。另一方面，如果 $r > \alpha$，则最优 $\rho = 0$，$u = \alpha$。这意味着自私矿工能获得的最大相对收益是诚实挖矿（即 $\rho = 0$）。因此，我们应该设置 $r > \alpha$ 以保证对手无法从交易包含攻击中获得更多收益。

抵御最长链扩展攻击。在这种攻击中，自私的矿工可以绕过一些诚实的微区块，直接在旧的诚实区块上挖矿。同样，让 ρ 表示被拒绝的微区块部分。具体来说，$\rho = 1$ 意味着
自私的矿工拒绝所有诚实的微型区块，直接在最后一个诚实的密钥上挖矿块。更确切地说，如果两个连续的关键区块是 $(X_i, X_{i+1}) = (0, 1)$，则有 $(1 - \rho)vY_i$ 诚实的微区块被最长链接受。让 K_i 表示一个指标随机变量，如果 $\{X_i = 0, X_{i+1} = 1\}$ 则等于 1，否则等于 0。让 $K = \sum_{i=1}^{m-1} K_i$。下面的 Lemma 可以帮助我们约束期望值
的 K 的 m 个区块。

定理 5 *对于 m 个图块序列，图块对 $(X_i, X_{i+1}) = (0, 1)$ 的个数具有如下切尔诺夫式约束：对于 $0 < \delta < 1$、*

$$\Pr(|K - \alpha\beta(m-1)| > \delta\alpha\beta(m-1)) < e^{(-\blacktriangle\delta^2\alpha\beta m)}. \tag{11}$$

接下来，我们计算大 m 时自私矿工的相对收益。一方面，所有矿工的交易费总额为
$\sum_{i=1}^{m-1}(vY_i R_t - \rho v K_i Y_i R_t)$.要了解这一点，请回想一下，有 $\sum_{i=1}^{m-1} vY_i$ 微块

产生的相关交易费用 $\sum_{i=1}^{m-1} \rho v Y_i R_t$。一旦 $K_i = 1$，由于最长链扩展攻击，就会有 Y_i 个微块不包括在最长链中。因此，相关的交易费用损失为 $\sum_{i=1}^{m-1} \rho v K_i Y_i R_t$。另一方面，自私矿工的总交易费用为 $\sum_{i=1}^{m-1} (\alpha v Y_i R_t - r\rho v Z_i Y_i R_t)$。要了解这一点，请回顾一下，在没有任何攻击的情况下，自私的矿工可以获得总交易费用的 α 部分，其计算公式为 $\sum_{i=1}^{m-1} \alpha v Y_i R_t$。随着最长的链延伸攻击，自私的矿工将损失 $1 - r$ 分数的交易费损失总额位居第二。综合上述分析，我们对于较大的 m，有如下 Lemma

下式 **6** *自私矿工的相对收入 μ 收敛到* $\frac{\alpha-(1-r)\alpha\beta\rho}{1-\alpha\beta\rho}$*，并具有较高的*
当 $m \to \infty$ *时的概率。*

该 Lemma 表明，对于大 m，自私矿工的相对收益为 $\frac{\alpha-(1-r)\alpha\beta\rho}{1-\alpha\beta\rho}$。与前面的分析类似，我们可以证明，当 $t \to \infty$ 时，自私矿工在 $[0, t]$ 期间的最大相对收益很有可能是

$$u = \max_{0\le\rho\le1} \frac{\alpha - (1 - r)\alpha\beta\rho}{1 - \alpha\beta\rho} = 1 - r + \max_{0\le\rho\le1} \frac{r - \beta}{1 - \alpha\beta\rho}. \tag{12}$$

如果 r≥ β，则最优 ρ = 1，相应的

$$u = 1 - r + \frac{r - \beta}{1 - \alpha\beta}。$$

在这种情况下，u 总是大于 α，因为 $1 - \alpha\beta < 1$。这意味着自私的矿工总是可以通过发起这种攻击获得大于其公平份额的相对收益。另一方面，如果 $r < \beta$，则最优 $\rho = 0$，$u = \alpha$。这意味着自私矿工能获得的最大相对收益是诚实挖矿（即 $\rho = 0$）。因此，我们应该设置 $r < \beta$，以保证对手无法从最长链扩展攻击中获得更多收益。将两个激励子

在交易包容机制和最长链延伸机制的作用下，R

需要满足

$$\alpha < r < \beta。$$

讨论。文献[10, 41]中的分析可以用来约束鲸鱼交易的拆分比率 r，而上述分析可以为网络容量限制下的常规交易比率 r 提供新的约束。这些界限如图 17 所示。从图中可以看出，新的界限 $\alpha < r < \beta$（对于常规交易）包含前两个界限 $1 - \frac{1-\alpha}{2} < r < \frac{1-\alpha}{1+\alpha-\alpha^2}$ 和 $\frac{\alpha}{2-\alpha} < r < \frac{1-\alpha}{2-\alpha}$（为鲸鱼交易）。这就产生了几个有趣的影响。

首先，引入网络容量限制并不会增加保持比特币-NG激励兼容性的难度。

这是因为鲸鱼交易的约束与之前的约束相同，而普通交易的约束

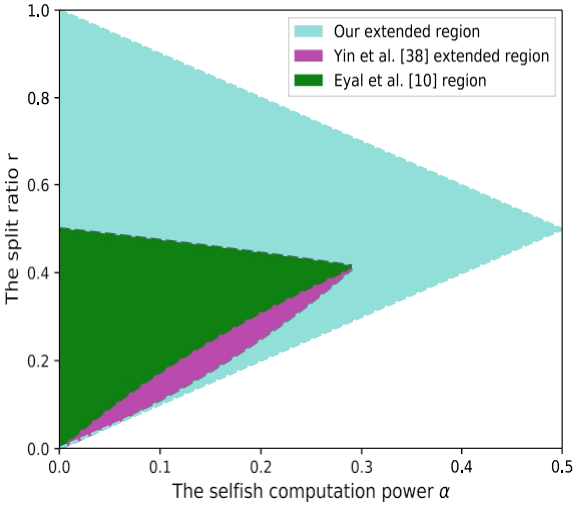

图 17 交易费比较分配比例

包含了前几条。其次，当 α 小于 29% 时，我们可以找到一个满足所有约束的 r 值。这意味着，即使在网络容量受限的情况下，比特币-NG 的激励兼容性也能适用于所有类型的交易。第三，当α大于29%时，我们无法找到满足所有约束的r值，因为鲸鱼交易的两个约束都失效了。这意味着比特币-NG的激励兼容性只能在常规交易中保持，而不能在鲸鱼交易中保持。换句话说，鲸鱼交易的存在可能会导致整个系统在该机制下的不稳定性。因此，应该设计一些相应的防御机制。

4.4 比特币-NG 中微块和密钥块挖掘的理论成果

在本节中，我们将对微区块和密钥区块挖掘进行联合分析。特别是，我们采用马尔可夫决策过程（MDP）来模拟各种自私的挖掘策略。为了使我们的分析具有可操作性，我们做了两个简化。首先，假设密钥块间隔为 $1/f$，因此在两个连续密钥块之间产生的微块数量为 v/f（注意，假设密钥块间隔遵循指数分布，均值为 $1/f$）。其次，只有二元选择：在交易包含攻击中公布或隐藏所有自私的微块，以及在交易拒绝攻击中接受或拒绝所有诚实的微块。

最长的链延伸攻击。(这与第 4.3 节中 $\rho = 0$ 或 1 的事实是一致的）。

MDP 模型由 4 元组 $M := (S, A, P, R)$ 表示，其中 S 是状态空间，A 是行动空间，P 是随机状态转换矩阵，R

是奖励矩阵。具体来说，S包含了自私挖矿过程中所有可能的状态；A包含了每个状态下的可用操作（例如，自私矿工发布或隐藏区块）；P包含了根据所采取的操作从当前状态到下一个状态的转换概率；R记录了当有一些状态转换时，自私矿工能获得多少奖励。表 2 展示了比特币-NG 中自私挖矿的 MDP。请注意，假设区块的传输没有延迟（见第 2.3 节），因此分析中不考虑分叉。下面我们将讨论 4 元组的各个组成部分：
行动自私的矿工有八种可用的行动。

- **采用和包含。**自私矿工接受所有诚实的密钥区块和相关的诚实微区块。换句话说，自私矿工将挖掘其密钥

  块，并放弃其私有链。这种行为被称为 "采纳"。
- **采纳和排除。**自私矿工接受所有诚实密钥区块和微区块，但最后一个诚实密钥区块之后产生的微区块除外。

  具体来说，自私矿工直接在最后一个诚实密钥区块上挖矿，这被称为

  adoptE。
- **覆盖和发布。**只要其私有链比诚实链长，自私的矿工就会发布其所有关键区块和相应的微区块。
  链的长度由密钥块计算。此操作表示为覆盖。
- **覆盖和隐藏。**自私矿工会公布其所有密钥区块和微区块，但在最后一个自私密钥区块之后开采的微区块除外。
  私有链比诚实链长。这一操作被称为 overrideH。
- **匹配和发布。**当诚实的矿工发现一个新的关键区块时，自私的矿工就会发布其相同高度的关键区块以及在其之后构建的微区块。

  这个关键区块。当自私的矿工提前拥有一个区块时，就可以采取这种行动

  ，这种行动被称为 "匹配"。

表 2 最佳自私挖掘的状态转换和奖励矩阵（Prob.）

State × Action	State	Probability	Reward	Condition
$(l_a, l_h, \cdot, S_h)$, adopt	$(1, 0, \text{noTie}, H_{\text{in}})$	α	$(l_h, l_h, 0, 0)$	–
$(l_a, l_h, \cdot, S_p)$, adopt	$(1, 0, \text{noTie}, H_{\text{in}})$	α	$(l_h, l_h - 1 + (1-r), 0, r)$	–
$(l_a, l_h, \cdot, \{H_{\text{in}}, H_{\text{ex}}\})$, adopt	$(0, 1, \text{noTie}, H_{\text{in}})$	$1-\alpha$	$(l_h, l_h \_ 1, 0, 0)$	–

表 2（续）

国家 x 行动	国家	概率	奖励	KOIIdltIOII
$(l_a, l_h, \cdot, S_h)$, adoptE			$(l_h, l_h, 0, 0)$	
(%#l z'z S p) z adoptE	(1, 0, noTie, *Hex)*		*(15, 15* - 1 + (1 - *r*), 0, *r)*	
$(l_a, l_h, \cdot,$ $\{H_{\text{in}}, H_{\text{ex}}\})$, adoptE	$(0, 1, \text{noTie}, H_{\text{ex}})$	$1-\alpha$	*(1q,1* - 1, 0, 0)	
$(l_a, l_h, \cdot, H_{\text{ex}})$, 否决	*hi.* — i,,0,		(0，0，*磅*+ 1，英尺 + 1)	
(/" //"-, up,)、覆盖	noTie, *Sp)*	1 - e	(0, *r,* lb + 1, lb + (1 - *r))*	*p*|>
(l z l z ' zp z th p) 否决	*ip - lb* - 1、1, 不对 *S p)*		(0, 0, *磅*+ 1, 英尺)	
l #t/- - //ex)- overrideH	hi.- '〃0,	α	(0, 0, */t,* + 1, ft + 1)	
(lg，lq，-，hfiq）、覆盖 H	noTie, *S)*	1 - e	(0, *r*, l_h + 1, l_h + (1 − *r*))	$l_a > l_h$
	ip - lb - 1, 1、不 *Sh)*		(0, 0, *lb* + 1, lb)	
t@p t§# # pp Sfi)) 覆盖 H		α 1 - a	(0, 0, 0, 0)	-
	(lg + 1, *lp*、无铁）	α	(0, 0, 0, 0)	
(I "lb, noTie, -）、等待		p,/§ -I- 1、无铁）		
(第 15 页，无、比赛	*(1* + 1, *英尺)* tie, Eli,) fin),			
(/" *lb,* tie, Eli,)、Ws"	*(1 p - 15,* 1, noTie, *S p)*	v(-)	(0, , ^ , ^- 1 + (1- r))	*Ia Iti*
	(lq, lz + 1、NoTie, fin)	i — g) i —	(0, 0, 0, 0)	
(1g、1α match fex）。	*(q1* + 1, lb、死 *bbb*	α	(0, 0, 0, 0)	
(1 , 1 , tie,fex)	($l_a - l_h$, 1, HOTie, *S p)*	y(1 - a)	(0, 0, lb, *1 - 1)*	$_a$ >

等待

表 2（续）

国家 x 行动	国家	概率	奖励	0ndl11OIl
	(1 , lh + 1 、 noTie, *Hex)*	(1 - y)(1 - e)	(0, 0, 0, 0)	
l" lb, noTie、 \S , *St*）），匹配	*(LP*+1，LB，平局，-)	a	(0, 0, 0, 0)	$la \geq l_h$
	(FP - *15,* 1、无铁，*S p)*	$\gamma(1-\alpha)$	(0, 0, 0, 0)	
	$(l_a, l_h+1,$ 无铁）	$(1-\gamma)(1-\alpha)$	(0, 0, 0, 0)	
(1g, lh , noTie, Eli,) matchH	*(1* + 1，磅 *tie', H;g)*	α		
(1 p, lh tie', ff¡,)、WE1D	*(1 p - 15,* 1, noTie, *S)*	y(1 - e)	(0, *r,* ft, */t,* - 1 + (1 - *r)) la* ñ *15*	
	(lq, lz + 1、NoTie, fin)	i - g)(i -)	(0, 0, 0, 0)	
(1 p, 1g, naTie, fex) matchH *(1 , 1 ,* tie', fex)-等待	*(lq* + 1, ft、tie', *Her)*	e	(0, 0, 0, 0)	$l_a \geq l_h$
	noTie, *Sh)*	'0 -")	$(0, 0, l_h, l_h-1)$	
	$(l_a, l_h+$ 1, 无领带，*Hex)*	(1−r)1−○	(0,0,0,0)	
*(l*页，*15*，*无领带*、	*(q1*+1, lb、领带，-	α	(0, 0, 0, 0)	$l_a \geq l_h$
$(l_a, l_h,$ tie′, $\{S_p, S_h\}$), wait	noTie, S_h)	y(1 - a)	(0, 0, *1 ,1)*	
	$(l_a, l_h+I$ 1, noTie, -) ,1、	(1 - y)(1 - e)	(0, 0, 0, 0)	

$(l_a, l_h,$ tie′, · 归还 „	归还		*(f" 磅，* 领带，-)	1
$(l_a, l_h, \cdot, S_h)$, 归还	$(l_a, l_h, *, S_p)$	1		
$(l_a, l_h, \cdot, H_{ex})$,	$(l_a, l_h, *, H_{in})$	1		

(0, 0, 0, 0)	(0, 0, 0, 0) - 0	*Ih*	(0, 0, 0, 0)	fq = 0

• 表示状态元素在状态转换中保持不变。

- **匹配和隐藏。**当诚实的矿工生成一个新的密钥块时，自私的矿工会公布其相同高度的密钥块，同时隐藏微块

  在该关键区块之后构建。当自私的矿工提前拥有一个区块时，也可以采取这一行动。这一操作被称为 matchH。
- 在这种情况下，自私矿工不会发布任何新的密钥区块和微区块，而是继续在自己的私有链上挖矿，直到出现新的密钥区块和微区块。
  找到相应的微块。
- **还原。**自私矿工会恢复之前的行动。具体地说，当没有诚实密钥区块被挖出后，自私矿工可以公布其隐藏的微区块。

  其区块；一旦诚实区块上没有挖出自私密钥区块，自私矿工就可以将诚实的微区块（在之前的决定中决定排除）包括在内，或将诚实的微区块（在之前的决定中决定包括在内）排除在外。

采用、覆盖匹配和等待行动包括对自私开采关键区块的所有可能行动，而隐藏、发布和还原行动则包括对交易包含和微区块最长链扩展攻击的所有可能行动。请注意，在匹配行动中，自私的矿工会发布其相同高度的关键区块，以匹配诚实矿工产生的关键区块。因此，会出现两个长度相同的分叉分支。在 Bitcoin-NG 中，诚实的矿工采用统一的平分规则来选择在哪个分支上挖矿。其中，引入变量 γ *来*表示在自私矿工分支上挖矿的诚实矿工的比例。

状态空间 状态空间 S 也由 4 个元组组成
$(l_a, l_h,$ fork, lastMicroBlock$)$。

- l_a 代表自私矿工在最后一个共同祖先密钥块之后挖掘的矿链长度。更准确地说，最后一个共同祖先密钥块

  是自私矿工和*所有*诚实矿工都接受的最长链中的最后一个密钥块，一旦自私矿工采用公有链或所有诚实矿工采用自私矿工的链，该密钥块就会被更新。此外，链长度由该分支中的自私密钥块计算。
- l_h 是最后一个共同祖先密钥块之后的公有链长度。自私的矿工和诚实的矿工都可以查看这条链。
- **fork**。字段分叉得到三种可能的值，分别称为 noTie、tie 和 tie^{r}。具体来说，tie 表示自私矿工发布了 l_h 个自私密钥块和相应的微块；tie^{r} 表示自私矿工发布了 l_h *个*自私密钥块和相应的微块，但最后一个自私密钥之后的微块除外。

  块；noTie 表示没有两个长度相等的公共分支。
- **lastMicroBlock**。该字段还包括四种可能的值，分别为 H_{in}, H_{ex}、

S_p 和 S_h。具体来说，H_{in}（分别为 H_{ex}）代表的共同祖先是是一个诚实的关键区块，相应的微区块被自私矿工接受（或拒绝）。而代表共同祖先的 S_p（或 S_h）则是一个自私的关键区块，所挖掘的相应微区块会被自私的矿工公布（或分别隐藏）。

状态转换与奖励。在状态转换中，自私矿工和诚实矿工的奖励可以用 4 元组 (R_h, T_h, R_a, T_a) 表示。具体来说，R_h （分别为 R_a ）是诚实（分别为自私）矿工的密钥块奖励，而 T_h （分别为 T_a ）是诚实（分别为自私）矿工的交易费。

回顾一下，交易有两种类型。这里主要分析常规交易，稍后将讨论鲸鱼交易。还可以回顾一下，普通微块的微块费用用 R_t 表示。为方便起见，每个字段只记录矿工赢得的密钥区块奖励或交易费（以 v/f 个微区块为一个单位的总交易费）的数量，而不记录奖励的数量。更重要的是，在决定下一个祖先密钥区块之前，共同祖先密钥区块之后的微区块中包含的交易费不会分配给矿工。这是因为这些交易费会受到自私矿工未来某些行为的影响（见第 4.3 节）。

在 "采纳 "或 "采纳E "行动中，自私的矿工接受 l_h 诚实的关键区块以及在这些关键区块之前开采的微型区块。诚实矿工获得 $l_h R_b$ 密钥区块奖励和 $(l_h - 1)v/fR_t$ 交易费。在覆盖或覆盖H行动中、
自私矿工发布 $l_h + 1$ 个自私密钥区块。诚实的矿工接受这些
关键区块和在关键区块之前产生的微型区块。因此，自私的矿工会获得 $(l_h + 1)R_b$ 个关键区块奖励和 $l_h v/fR_t$ 交易费。在匹配行动中，下一个状态取决于下一个关键区块是否由
$(1-\gamma)(1-\alpha)$)，或在自私分支上采矿的左侧诚实矿工(w.p. $\gamma(1-\alpha)$)。在后一种情况下，自私矿工实际上覆盖了诚实矿工的分支。它可以获得 $l_h R_b$ 密钥块奖励和 $(l_h - 1)v/fR_t$ 交易奖励。
费用。请注意，γ 的值由所采用的分叉方案决定（例如，$\gamma = 0.5$
统一的决胜局政策）。

一旦共同祖先密钥块发生变化，将分配上一个祖先密钥块之后产生的微块中的交易费。有两种情况：

- 上一个共同祖先密钥块由诚实矿工挖掘。这种情况又可分为两种子情况：(1) 下一个关键区块是由诚实的挖矿者挖出的；(2) 下一个关键区块是由诚实的挖矿者挖出的。

  矿工，诚实矿工获得 v/fR_t 交易费；（2）下一个关键区块由自私矿工开采，且 **lastMicroBlock** = H_{in} ，诚实矿工获得 rv/fR_t 交易费，自私矿工获得 $(1 - r)v/fR_t$ 交易费。
- 前一个共同祖先密钥块由自私的矿工开采。这
  情况可进一步分为两个子情况：(1) 下一个关键区块被挖掘出来
  由自私矿工开采，自私矿工获得 v/fR_t 交易费；(2) 下一个关键区块由一些诚实矿工开采，且 **lastMicroBlock** = S_{p} ，自私矿工获得 rv/fR_t 交

易费，诚实矿工获得 $(1 - r)v/f R_t$
交易费。

请注意，由于鲸鱼交易是罕见的、不可预测的，因此微块费用可以建模为取两个值的随机变量：R_t 或 R_t 加上以下费用

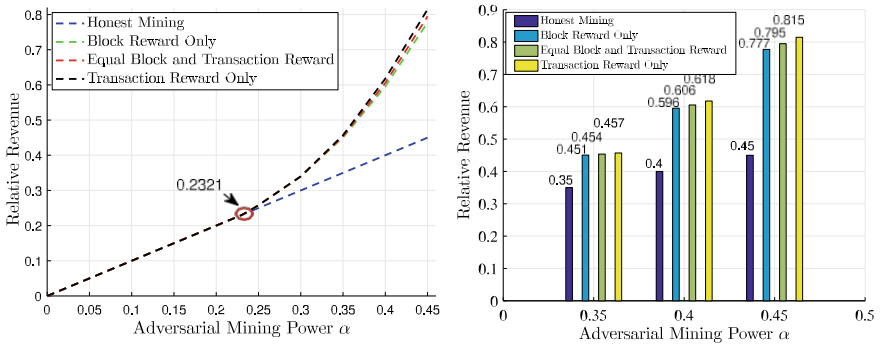

图 18 以太坊中自私挖矿攻击的结果 **a** 不同 α 时自私矿工的相对收益。

鲸鱼交易。让 $\bar{R}_t$ 成为预期的微块费用。显然，$\bar{R}_t > R_t$。鲸鱼交易的长期效应是将比率 k 从 R_b/R_t 降至 $R_b/\bar{R}_t$。

这种效应会略微增加自私矿工的相对收入。

利用 MDP 工具箱[7]求解上述 MDP 模型，我们可以数值求出每种情况下的最优策略，并得到自私矿工的相对收益。为了更好地说明结果，我们首先绘制了

如图 18a 所示，当 $r = 0.4$（Bitcoin-NG [10]）时，自私矿工的奖励为不同的 α。这里考虑了三种奖励设置：$k \to 0$、$k = v/f$ 和 $k \to \infty$。具体来说，在第一种情况下，交易费主导矿工的收入；

在第二种情况下，两个连续关键区块之间的 v/f 个微区块中包含的交易费与一个关键区块奖励的权重相同；在第三种情况下，关键区块奖励主导矿工收入。请注意，关键区块奖励占主导地位的情况与比特币的奖励分布类似，也就是说，微区块架构不会对系统产生影响。从图中可以看出，在这三种情况下，自私挖矿盈利的阈值都是 23.21%，与比特币中自私挖矿的阈值相同。换句话说，通过

采用合适的 r（即 $\alpha < r < 1 - \alpha$），与比特币相比，Bitcoin- NG 中的微块架构不会影响系统的安全性。此外，自私的

即使当 $\alpha > 29\%$ 时，三种情况下矿工的收入仍然相同。这验证了第 4.3 节中的分析，并证明在适当的设置下，Bitcoin-NG 与比特币一样具有弹性。当 $\alpha > 35\%$ 时，图 18b 展示了三种情况下自私矿工的收入与诚实矿工收入之间的差异。不难看出，在交易费占主导地位的情况下，自私的矿工可以获得最高的收入。这意味着微块架构可以略微增加自私矿工的收入。

5 更多阅读

在本节中，我们将为感兴趣的读者提供更多关于区块链激励设计和分析的作品。

比特币。Eyal 和 Sirer 是最早正式分析比特币自私挖矿的人之一 [11]。然而，他们提出的自私挖矿策略并不是最优的。后来，Sapirshtein 等人[33]和 Nayak 等人[25]证明，通过采用优化策略，当诚实矿工采用统一的打破平局防御时，使自私挖矿有利可图的计算能力阈值可降至 23.2%（而不是[11]中的 25%）。在 [17] 中，Gervais 等人进一步将分析扩展到比特币的多个变种，包括 Dogecoin、Litecoin 和 Ethereum。与这些关于区块奖励的研究不同，[6] 对交易费的自私挖矿策略进行了研究。研究结果表明，即使攻击者的计算能力较弱、网络连接较差，也能从攻击中获得更多收益。在此基础上，Tsabary 和 Eyal [37] 还研究了区块奖励和交易费之间的比特币差距博弈。在 [19] 中，分析自私挖矿时考虑了传播延迟。

除了激励分析，还有几项关于防御比特币中自私挖矿攻击的研究。Heilman 提出了一种名为 "新鲜度优先"（Freshness Preferred）的防御机制 [20]，通过使用可信方发布的最新不可伪造时间戳，可以将阈值提高到 32%。Bahack在[2]中引入了分叉惩罚规则，使自私挖矿无利可图。具体来说，系统中的每个矿工都可以在其区块中包含分叉证据。一旦确认，矿工可以获得获胜分支总奖励的一半。Solat 和 Potop-Butucaru [34] 提出了一种名为 ZeroBlock 的解决方案，它可以在不使用可伪造时间戳的情况下，使自私矿工的区块过期并被所有诚实矿工拒绝。在 [43] 中，提出了一种向后兼容的防御机制，称为加权 FRP，其中考虑的是分叉链的权重而不是长度。在 [28] 中，Pass 和 Shi 提出了 Fruitchains，它将奖励分配给所有最近的果实，这些果实是区块挖掘的平行产物。与 Fruitchains 类似，Szalachowski 等人[36]提出了一种名为 StrongChain 的新协议，该协议允许矿工发布弱解决方案，即具有较高挖掘难度目标的解决方案。矿工可以在其区块中包含弱解决方案，并始终选择区块和弱解决方案加权数最多的链进行挖矿。Bissias 和 Levine[3] 提出了 Bobtail，它能让矿工发布并收集所有目标较高的 PoW 解决方案，直到 k 个最小哈希值的平均值低于某个目标。**以太坊**在 [30] 中，Ritz 和

Zugenmaier 进行了大量模拟，研究以太坊中的自私挖矿。Wang 等人[38] 分析了以太坊中的两种顽固挖矿行为。Yang等人[40]分析了以太坊中不完善网络对自私挖矿的影响。此外，以太坊在 EIP1559 [4]中更新了交易费机制，这使得它与比特币的交易费机制大不相同。关于这种新的交易机制，已有多项研究[13, 23, 31, 32]。

其他区块链。Yin 等人[41]对比特币- NG 的激励分析进行了扩展，考虑了原论文忽略的一种情况[10]。后来，Wang 在

等人[39] 研究了高级自私挖矿策略，即顽固挖矿策略。[39]考虑了高级自私挖矿策略，即顽固挖矿策略，当攻击者可能操纵两个诚实方之间的微区块链时。Fooladgaret 等人[14]在 Algorand[18]中模拟了策略交互场景中的参与成本和获得的奖励。他们发现 Algorand 中的奖励分享方法并非纳什均衡，并提出了一种新的奖励机制来解决这一问题。

6 结论

在本章中，我们将重温三个有影响力的区块链协议的激励设计和分析：比特币、以太坊和比特币-NG。其中，与比特币相比，以太坊引入了两个新奖励（即叔侄区块奖励），而比特币-NG则重新设计了交易费用分配规则，以适应新的共识架构。通过研究，我们首先发现激励设计与系统安全密切相关。因此，在采用激励机制前应仔细评估激励机制的设计。其次，我们发现新的共识协议也需要新的激励设计。因此，区块链协议应共同考虑和评估共识算法和激励机制。第三，我们发现现有的激励分析可能无法适用于新的设计，因此需要进行量身定制的分析。然而，建模和理论分析的过程很难评估每个区块链协议的激励设计。因此，深度强化学习（DRL）等人工智能驱动的方法可能有助于自动分析激励设计并进行分析。我们注意到在这个方向上已经有了一些工作，关于这些自动分析的讨论将作为未来的工作。

参考资料

1. Bagaria, V., Kannan, S., Tse, D., Fanti, G., Viswanath, P.：解构区块链以接近物理极限（2018 年）。arXiv 预印本 arXiv:1810.08092
2. Bahack, L.: 计算能力不足一半的理论比特币攻击（草案）(arXiv preprint arXiv:1312.7013
3. Bissias，G.，Levine，B.N.：Bobtail：通过低方差挖掘提高区块链安全性。In：网络与分布式系统安全研讨会（NDSS），NDSS '20 (2020)
4. Buterin, V., Conner, E., Dudley, R., Slipper, M., Norden, I., Bakhta, A.: EIP-1559: fee market 修改为 ETH 1.0 链（2019 年）。https://eips.ethereum.org/EIPS/eip-1559
5. Buterin, V., et al：下一代智能合约和去中心化应用平台。白皮书（2014 年）
6. Carlsten, M., Kalodner, H., Matthew Weinberg, S., Narayanan, A.: 论比特币的不稳定性

没有区块奖励。In：2016 ACM SIGSAC 计算机与通信安全会议论文集，CCS '16，第 154-167 页。ACM (2016)

7. Chadès, I., Chapron, G., Cros, M.-J., Garcia, F., Sabbadin, R.: Mdptoolbox: a multi-platform 解决随机动态编程问题的工具箱。生态学》**37**（9），916-920 (2014)
8. 以太坊。设计原理：大叔激励机制。Github (2018)

9. 以太坊挖矿奖励（2018 年）
10. Eyal, I., Gencer, A.E., Sirer, E.G., Van Renesse, R.: Bitcoin-NG: a scalable blockchain protocol.In: 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)、
pp.45-59 (2016)
11. Eyal, I., Sirer, E.G.: 多数是不够的：比特币挖矿是脆弱的。Commun.ACM **61**(7), 95-102 (2018)
12. Feng, C., Niu, J.: 以太坊中的自私挖矿。In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp.
13. Ferreira, M.V.X., Moroz, D.J., Parkes, D.C., Stern, M..：ArXiv preprint arXiv:2103.14144
14. Fooladgar, M., Manshaei, M.H., Jadliwala, M., Rahman, M.A.: On incentive compatible role- based reward distribution in algorand.In: 2020 50th Annual IEEE/IFIP International Confer- ence on Dependable Systems and Networks (DSN), pp.IEEE (2020)
15. 论坛，比特币：采矿卡特尔攻击，2010 年 12 月
16. Gervais, A., Karame, G.O., Capkun, V., Capkun, S.: Is Bitcoin a decentralized currency?IEEE Secur.隐私 **12**(3), 54-60 (2014)
17. Gervais, A., Karame, G.O., Wüst, K., Glykantzis, V., Ritzdorf, H., Capkun, S.: On the security and performance of proof of work blockchains.In：Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16, pp.ACM (2016)
18. Gilad, Y., Hemo, R., Micali, S., Vlachos, G., Zeldovich, N.: Algorand: Scaling Byzantine agreements for cryptocurrenies.In：第 26 届操作系统原理研讨会论文集，SOSP '17，第 51-68 页。ACM (2017)
19. Göbel, J., Keeler, H.P., Krzesinski, A.E., Taylor, P.G.: Bitcoin blockchain dynamics: the selfish- mine strategy in the presence of propagation delay.Perform.Eval.**104**, 23-41 (2016)
20. Heilman, E.: 一个阻止自私矿工的怪招：新鲜比特币，诚实矿工的解决方案。In：国际金融密码学与数据安全会议，第 161-162 页。Springer (2014)
21. Hoenicke, J.: 未确认的交易计数 (mempool) (2020)
22. Kogias, E.K., Jovanovic, P., Gailly, N., Khoffi, I., Gasser, L., Bryan, F.: Enhancing bitcoin security and performance with strong consistency via collective signing.In: 25th USENIX Security Symposium (USENIX Security 16), pp.
23. Leonardos, S., Monnot, B., Reijsbergen, D., Skoulakis, S., Georgios, P.： ArXiv preprint arXiv:2102.10567
24. 中本聪：比特币：点对点电子现金系统。工作文件（2008 年）
25. Nayak, K., Kumar, S., Miller, A., Shi, E.: Stubborn mining: generalizing selfish mining and combining with an eclipse attack.In: 2016 IEEE European Symposium on Security and Privacy (EuroS P), pp.IEEE (2016)
26. Niu, J., Feng, C., Dau, H., Huang, Y.-C., Jingge, Z.：中本共识分析，重温（2019）。arXiv preprint arXiv:1910.08510
27. Niu, J., Wang, Z., Gai, F., Feng, C.: Bitcoin-NG 的激励分析，重温。In：Performance Evaluation：国际期刊》，第 144 卷，第 102-144 页。爱思唯尔 (2020)
28. Pass, R., Shi, E.: Fruitchains: a fair blockchain.In：ACM 分布式计算原理研讨会论文集，PODC '17，第 315-324 页。ACM (2017)
29. Pass, R., Shi, E.: Hybrid consensus: efficient consensus in the permissionless model.In：31st International Symposium on Distributed Computing (DISC 2017), vol. 91, pp.
30. Ritz, F., Zugenmaier, A.: 以太坊中叔叔奖励对自私挖矿的影响。In: 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS PW), pp.IEEE (2018)
31. Roughgarden, T.: 以太坊区块链的交易费机制设计：ArXiv preprint arXiv:2012.00854
32. Roughgarden, T.: Transaction fee mechanism design (2021). arXiv preprint arXiv:2106.01340

33. Sapirshtein, A., Sompolinsky, Y., Zohar, A.: 比特币中的最优自私挖矿策略。In：国际金融密码学与数据安全会议，第 515-532 页。Springer (2016)

34. Solat, S., Potop-Butucaru, M.: Zeroblock: preventing selfish mining in Bitcoin.技术报告，索邦大学（2016 年）
35. Sompolinsky, Y., Zohar, A.: 比特币中的安全高速交易处理。In：金融密码学与数据安全》，第 507-527 页。Springer (2015)
36. Szalachowski, P., Reijsbergen, D., Homoliak, I., Sun, S.: Strongchain: transparent and collaborative proof-of-work consensus.In：第 28 届 USENIX 安全研讨会论文集，SEC'19，第 819-836 页（2019 年）
37. Tsabary, I., Eyal, I.: The gap game.In：2018 ACM SIGSAC 计算机与通信安全会议论文集，CCS' 18，第 713-728 页（2018 年）
38. Wang, Z., Liu, J., Qianhong, W., Zhang, Y., Hui, Yu., Zhou, Z.：不完善以太坊网络中自私和顽固挖矿对大叔区块影响的分析评估。Comput.Secur.**87**, 101581 (2019)
39. Wang, Z., Liu, J., Zhang, Z., Zhang, Y., Yin, J., Yu, H., Liu, W.: A combined micro-block chain truncation attack on Bitcoin-NG.In：信息安全与隐私》，第 322-339 页。Springer (2019)
40. Yang, R., Chang, X., Mišic´, J., B Mišic´, V.: Assessing blockchain selfish mining in an imperfect network: honest and selfish miner views.Comput.Secur.**97**, 101956 (2020)
41. Yin, J., Wang, C., Zhang, Z., Liu, J.: Revisiting the incentive mechanism of Bitcoin-ng.In. Information Security and Privacy, pp：信息安全与隐私》，第 706-719 页。Springer (2018)
42. Yu, H., Nikolic, I., Hou, R., Saxena, P..：OHIE：区块链扩展变得简单。In：Proceedings of the 41th IEEE Symposium on Security and Privacy, S&P.IEEE (2020)
43. Zhang, R., Preneel, B.: Publish or perish: a backward compatible defense against selfish mining in Bitcoin.In：RSA 会议密码学家专场，第 277-292 页。Springer (2017)