Tezos: 一个自我修复的加密账本
目的聲明

L.M Goodman

August 3, 2014

"Laissez faire les propriétaires."
— Pierre-Joseph Proudhon

摘要

随着比特币的普适化，市场开始出现一些去中心化的加密货币，也称竞争币。其中具备代表性的有以太坊，CryptoNote, 以及零币，这些都对加密货币领域内做出了独特的贡献。尽管很多替代货币有着自己的创新，但是由于彼此孤立，没有办法来接纳其它货币的创新来让自己变得更加成功。我们希望通过设计和实现 Tezos，一个普适的且能够自我进化的加密账本，来补救这种无意义的消耗和浪费。

Tezos 可以实例化任何一个区块链协议。其种子协议规定了一个流程，该流程支持让持币者决定针对协议改进的审批，包括对改进过程自身的改进。针对 Tezos 的升级在测试环境下筹划，允许持币者撤销潜在的可能出现问题的修正。

Tezos 的哲学由 Peter Suber' s Nomic[1] 启发，这是一个完全建立在内省性规则集合之上的游戏。

在本文内，我们希望阐述 Tezos 的潜在优势，以及选择股权证明机制实现共识协议和 OCaml 语言编码的原因。

目录

1 动机

我们开发 Tezos 的动机是希望通过 Tezos 来解决比特币 [2] 的四个问题:

- “硬分叉” 问题，也就是比特币因为协调问题而产生的不能够动态创新的问题。
- 成本和中心化问题，导致这个问题的主要原因是比特币的工作证明机制。
- 比特币交易语言的有限表达问题，致使智能合约出现在其他竞争链上。
- 关于加密货币实现的安全隐患。

1.1 协议分叉问题

1.1.1 持续创新

为了让比特币持续化创新，很多的开发者和企业家开发了很多的竞争加密货币（所谓的山寨币）。尽管很多加密货币只是对比特币的原始代码的简单模仿[1]，但其中不乏一些有所针对地做了一些有意义的改善。例如，莱特币引进了高内存要求的工作证明机制[2]，以及更短的区块确认时间。类似的，以太坊具有有状态的合约和图灵完备的交易语言 [3]。其它的创新包括为隐私隐藏环签名 (CryptoNote)[4] 以及使用 SNARK(ZeroCash)[5] 实现的不可追踪交易。

这些山寨币的崛起促使在软件创新上的更加激烈的竞争。但很多哈耶克式增长的支持者们忽略了一个重要的问题，那就是加密货币要想成为货币的一种有效形式，那么它必须首先成为一个稳定的价值存储工具。账本内的创新通过保护和强化基础设施网络，从而为该货币赋予和保存价值。

[1] 喔，非原创

[2] scrypt mining ASICs 已经可用

为了更好的展示这个问题，让我们来比较一个加密货币和智能手机。当我们购买一部智能手机的时候，消费者支付的是手机的某些特性功能，例如播放音乐，查阅邮件的能力，和朋友互发短信的能力，以及打电话的能力。

几周以后又一款新的智能手机上市了，往往带有更新更强的功能。尽管老型号手机用户会希望得到这些新型号，但新型号的手机并不会让老手机不能使用。

如果新的手机变得不能够和老型号手机无缝通信，不能够做到向下兼容，那么这种关系就会改变。每款型号手机的价值就仅限于使用该型号手机的人群。

加密货币存在和智能手机兼容性类似的问题。其价值来源于底层网络，或者赋予货币价值的用户数量。从这点上说，一个新的创新货币要么因为不能够建立强大的网络而失败，要么以损害原有加密货币价值为代价获得成功。如果一款智能手机不能和旧款兼容，那么该手机可能没有任何创新，也可能具备颠覆性的创新，强制旧版本过时。

侧链是一个让新货币的价值和比特币进行锚定实现两者之间的互换的尝试，其保证了侧链与比特币之间的兼容性。不幸的是，这样的关系不确定能有足够的灵活性来满足那些和比特币非常不同的货币。迄今唯一的替代方案就是对协议进行分叉。

1.1.2 分叉经济学

要理解分叉经济学，一定要首先理解货币的价值首先是一个社会共识。我们很容易认将加密货币等价于规则和账本，其实它们的价值源自它们被广泛接受为货币。尽管这样的观点看上去似乎是循环辩论，但却没有任何的矛盾。从一个博弈论的观点来看，一个币只要被广泛接受就可以成为稳定的储值工具。作为一个账本，比特币只是一系列的 1 和 0。如何对待这些数量的选择是一个完全基于社会的共识，而不是协议本身。

对协议的改变被称为分叉。它们之所以被称为是分叉是因为原则上，用户有权来保存旧的协议。所以，当分叉发生的时候，货币被一分为二，成为一个老的版本和一个新的版本。

一个成功的分叉并不仅仅对软件工程学有要求，也要求一定数量用户的配合。这样的配合在实践上特别难以实现。事实上分叉之后出现两个账本，用户将面对二选一的问题，也就是如何衡量不同分支的价值。

这是一个所谓的协调博弈，用户往往选择他们所认为其他用户被期待

选择的分支。所以，这些用户很可能也会使用同样的策略来基于同样的原因来做出选择。经济学家 Thomas Schelling 对这些博弈进行过仔细的研究，并提出了一些原创的观点 [6]。

不幸的是，Schelling 最优并不保证总是被绝大所数的持币者所选择，这只是一个默认选择。默认选择可以是跟随核心开发团队的意见，它也可以是跟随监管的裁定。

一个攻击者可以通过改变社会共识来操控货币来达成各种意图和目的。如果一旦币价随着共识的转移而崩溃，那么和最初的共识协议保持一致的选择将变得毫无意义。[3]

核心开发团队是潜在的非常危险的中心化威胁。尽管用户可以对一个开源项目进行分叉，但是这个能力在那些给有足够能力的强大到足够改变社会共识的攻击者面前所能提供的保护是微不足道的。即使我们认定所选择无条件的信任的开发团队总是善意的，这仍然是一个系统的短板，可以被攻击者所利用。

Tezos 通过完全的去中心化分叉来解决中心化所带来的系统弱点，通过使用自己的加密账本来让持币者在分叉链上进行协调。这让协调成为可能，并且让原则变得神圣不可侵犯 - 那就是如果不是内部发生的分叉，那么这个分叉将不被承认。最终通过改变共识来攻击协议变得更加困难。

假设一个很受欢迎的开发者宣布他有计划在不通过协议的内部章程的前提下分叉 Tezos，其他用户会问 “为什么他会试图绕过这个过程？” 持币者群体会发问。更加重要的是因为开发者知道自己不能通过这种方式在 Tezos 内建立一个新的共识所以也不会这样做。

这也给投资者释放了信号，他们所偏好的共识将是拒绝这个分支，按照 Schelling 点原理，这个分支会被拒绝，不管这个开发者多么有影响力。

1.2 劳动证明机制的缺陷

劳动证明机制的问题。比特币所运用的劳动证明机制 (PoW) 是一个巧妙的利益平衡机制，其目的是解决了双花问题。虽然在排除矿工联合劫持网络以外的场景有很好的理论特性，在实践中这个机制有很多问题。

[3]一个论点是永远不会有超过两千一百万个比特币，因为如果一个 fork 提高这个上限，那么它就不再是比特币了。这种观点是没有道理的，比特币是共识所达成的。

1.2.1 挖矿算力集中化

以 PoW 作为加密货币的基础有几个问题。其中的一个，在 2014 年特别突出的问题是矿池的中心化。这把算力集中在少数的几个人手里。

劳动证明机制是去中心化的原因是用户不用去信任来维护货币系统安全的系统。但是，私底下，比特币系统内，所有的用户必须相信一个或者两个矿池是善意的。

一个矿工拥有超过 50% 算力的结果是可以对系统发动 51% 攻击 [7]。这让攻击者可以让转账停顿，让已经发生的转账逆转，偷走最近挖出的币，以及进行双花 [8]。

一个中心化的印钞机构完全可以和一个有矿工控制着 51% 的算力的网络那样安全，而且更加节省资源。如果一个中心化的印钞机构不能够被比特币用户所接受，同理，他们也不应该接受矿工的中心化。

这个中心化不是一个巧合。大的矿池获得稳定的收益，因此可以获得大量用户。而这个增长进一步提升他们的市场份额，让他们的收益波动变得更低。

让事情变得更加糟糕的是，ghash.io, 一个大矿池声称自己的一个商业模式是为那些交高手续费的转账者提供优先服务，更加剧了大矿工赚更多的钱的趋势。不幸的是，p2pool 没有能够吸引大量的算力，因为大多数的矿工因为方便自私地选择了中心化的矿池。

很多人认为市场的中心化是被过度夸张了。这样的论点犯了现实经济的过度概括的问题。真正的商业在一个极具变化的环境中进行竞争，而创新的破坏让现有的领导者感到压力。真正的商业需要本地知识，他们会面对组织问题，以及主要代理（principal agent）问题。比特币挖矿是一个纯粹的人工合成的经济类型，其中心是算力，而算力是一个完全可互换的商品。把基于传统商业类比比特币是错误的，因为这个单一的环境完全没有传统经济那么复杂，丰富。[4]

很多经济学观点认为自然垄断者没有动机来滥用自己的垄断地位。这个观点也可以被运用在比特币挖矿行业 - 毕竟，哪一个占据主导地位的矿工会冒着让自己投资减损的风险来破坏这个货币？然而不幸的是，这仍然产生了一个巨大的系统风险，因为拥有大算力的矿工可以被不诚实的攻击者所

[4]很可能会有新技术将会取代 ASICs, 正如它们已经取代了 FPGA 电路板一样。然而，这种创新的速度没有快到足够可以让矿工在长时间内形成一个主导地位。这样的创新会让一小帮的新人通过掌握新技术收益，或者那些掌握足够资金的人来重复这个过程的人。

利用。一个破坏网络的双花攻击的代价不会高过颠覆几个大矿池的费用。

曾经有人提议对协议进行修改，让矿池的组织者无法想当然地默认用户会诚实，然而这些提议仅仅让矿池不能够从匿名参与者那里集中算力，而参与者仍然没有有效报复手段。矿池也可以采用非匿名的方式进行组织。矿池主运行算力，而参与者仅仅持有股份，或者组织者可以追踪那些不诚实的成员，要求他们加入一个可以被辨识的身份在他们要进行哈希的区块中。这样的结果是增加匿名的挖矿操作的变化，并且让算力更加集中在少数几个挖矿集团手里。

而以 Tezos 为代表的 PoS 机制，则可以免收这类问题的困扰。如果有人持有多数算力，那么它必然持有多数的货币，这样的结果是如果要发动 51 攻击所要承担的代价也更加的大，而且也意味着更好的激励体制。

1.2.2 坏的动机

PoW 面临一个更加深刻的问题是很难让矿工和股东之间达成利益一致。

长期来看，挖矿的总利润应该是支付给矿工的所有的交易手续费之和。因为矿工们彼此竞争来产生哈希，花在挖矿上的钱最终将比利润略小，而花在转账上的费用依赖于对转账的供给和需求。所能接受交易的数量受区块的大小所限制，而且区块大小是固定的。

不幸的是，我们有理由预期交易的需求会降到非常低的水平。为减少交易确认的等待时间，尤其是小金额交易，人们很可能会利用基于第三方信任的链下交易机制。支付处理器之间可能并不需要频繁的信誉验证。

这种情况不仅仅是经济上的最好选择，也是基于比特币支持的低转账速度下不得已的解决办法。区块链转账必须要和链下转账竞争，花在转账上的费用最终将逼近它的成本价，而现代互联网基础设施的飞速发展意味着这个值最终将趋近 0。

试图强加一个最低的转账费用可能会将导致问题恶化，并让用户更多依赖于链下转账。随着支付转账费的数量下降，矿工的收益也随着下降，而发动 51 攻击的成本也会下降。简单来说，区块链劳动证明 PoW 的安全面临一个公地问题 [9]。核心开发 Mike Hearn 已经建议使用特殊的转账来给挖矿提供资助，即一种 pledge 的融资方式 [10]。一个健壮的货币应该不需要假定节点无恶意才能够安全运作。

股权证明机制 (PoS) 解决了这些问题，通过重新配置矿工和用户之间

的关系：按照其定义，矿工也同时是持币人，因此也希望转账的费用比较低。于此同时，因为 PoS 挖矿并不是基于资源消耗，所有的交易的成本（不管是直接的费用还是间接的通货膨胀）都被矿工所承担，而他们不需要进行财富消耗的竞争就可以负担运营成本。

1.2.3 成本

一个替代方案是像狗狗币 [11] 一样让挖矿收益变为永久性。然而随之而来的问题是用户使用成本的增加，矿工收益没有增加，造成对整个系统的净损失。确实，随着矿工彼此竞争来生产哈希，他们投资的钱将比最终的收益少一点，长期来看，他们的收益将等同于他们转账手续费。挖矿的成本将被每个人所承担。

更严重的是，真正的经济商品（fabs 时间，电，工程投入）都因为 PoS 挖矿机制而不能参与实体经济。在 2014 年六月，比特币的全年的贬值率大约为 10%，平均每天因为维护这个系统要消耗 2.16 百万美元，这并不能让一个算力集中在 ghash.io 手里的系统更加安全。

PoW 的支持者坚持其安全保障在于攻击代价比一个攻击者所愿意花费的金额要高，这个代价随着货币的价值的提升而不断升高。

PoS 减少了浪费的资源，但没有降低攻击的代价。它在货币升值的同时自动地让攻击的代价升高了。因为你要证明你挖矿的时候不是摧毁现有的资源而是证明提供现有资源，一个 PoS 货币并不会因为变得更受欢迎而消耗更多资源。

1.2.4 控制

一个 PoW 系统让矿工而不是持币人成为系统的主人。分叉要求大多数矿工支持。这导致了潜在的利益冲突。大多数的矿工可以决定是否劫持区块链来通过协议分叉来增加挖矿回报。更严重的是，他们将会让这个非常低效浪费的系统来让他们长期存在，这并不符合用户的利益。

1.3 智能合约

尽管比特币允许智能合约，但大多数的 opcodes 已经不能使用，其潜在功能也受到限制。以太坊提出智能合约系统，其关键性创新在于：1、脚

本语言是图灵完备的；2、用有状态账户替代比特币的未花掉输出（unspent outputs）。

尽管重点是在语言的图灵完备性，但第二个特性是到目前为止最有意思的和强大的。在比特币系统里，一个 output 可以被认为是只有两种状态：花掉和没有被花掉。在以太坊，账户（被密钥所保护的）维护一个余额，一个合约代码，和一个数据存储。一个账户存储的状态可以由向该账户的交易改变。该交易规定传入合约代码的币的数量和参数。

图灵完备脚本语言的一个会坏处是需要执行脚本的潜在步骤是无限的，该属性在通过情况下是不可计算的。

为解决这个问题，以太坊提出矿工在验证每笔交易时引入一定的交易费用，该费用与执行合约的复杂性和步数成比例。

但是，为了让区块链变得安全，*all* 活跃节点需要验证交易。一个恶意的矿工可以在其区块里包含一个交易，该交易内的程序是一个死循环，并且给自己支付特别高昂的费用来确认这笔交易。其它的矿工将会浪费很长时间来确认这笔交易。更糟糕的是，他们可以拖延，并且不确认。实际上，大多数的有趣的智能合约按照很简单的商业逻辑进行实现，并不需要特别复杂的计算。

我们的解决方案是对单个交易中程序允许执行的最大步数加一个限制。区块有一个大小的限制，用于限制交易数量，在每个区块运算步数上也有一个上限。这个限制让针对 CPU 使用的 DOS 攻击变得无效。同时，合法用户可以发起多个交易来允许超过单个交易步长限制的更多步数。矿工可以决定忽略太长执行的交易，如果他们觉得被包含的费用过于低。因为 Tezos 的协议可以被修改，这个上限可以在未来修订中提升，同时随着需求的变化，新的加密原语也可以被添加到脚本语言。

1.4 正确性

比特币是八十亿美元市值的市场。正如安全研究员 Dan Kaminsky 所解释的，比特币看上去像一个安全的噩梦。一个C++ 的代码基础外加一个定制的二元协议节点通过互联网互联并保存电子现金 - 这样的配置听上去似乎是一个完全的灾难。C++ 语言有各种内存腐败的 bug。当他们被通过互联网链接在一起的时候，这将产生可被远程攻击者所利用的弱点。如果攻击者足够聪明到可以发现系统弱点，那么电子现金是一个直接利益动机。

幸运的是，比特币的实现到目前为止被证明是足够的健壮的，在绝大多

数情况下。在 2010 年的八月份，一个系统的漏洞让攻击者通过一个 0.5 个币的 input 创造两个 92233720368.54 币的输出。更近的一次是大量的漏洞例如 heartbleed 漏洞被在 OpenSSL 的代码库中被发现。这些个弱点有一个共同点 - 他们产生的原因是因为 `C` 或者`C++` 这样的编程语言并不检查他们执行的 operation。基于效率的考虑，他们都可以获得进入权限。虽然比特币并没有被这些问题所摧毁，但他们确实让系统安全收到不断困扰。

有些语言都没有这问题。OCaml 是一个功能上的程序语言，由 INRIA 在 1996 年所开发出来, 而它自己也是建立在之前的努力之上的。它的速度和`C++` 差不多，它在各个指标上大致和最快的编程语言不相上下 [12]。更重要的是 OCaml 是非常强的类型, 并提供非常强大的类型推理系统，语法和句法，包括强大的模式搭配和高排序模块，这让这个编程语言很容易集成和正确地表述底层区块链协议的类型。

OCaml 的语法十分严谨，并且其很多的子集都已经被完全形式化 [13], 这也消除了关于修改行为的任意歧义。

此外，Coq 作为一个最高级的证明检测软件，可以从证明中提取出 OCaml 代码。随着 Tezos 变得越来越成熟，它将能够从正确性的数学证明中自动化地提取协议代码的关键部分。

很多软件失败的例子。例如 heartbleed 漏洞就导致数百万美元的损失。在 2013 年，一个高频交易机构 Knight 资本因为一个 bug 导致 5 亿美元的损失。在 1996 年，一个 arithmetic 的溢出 bug 导致 Ariane5 火箭爆炸，导致 70 亿美元火箭开发成本报废，火箭费用和运载物品的合计代价估值约为五亿美元。

所有的这些漏洞都可以通过形式化验证避免。形式化验证近些年已经发生了天翻地覆的变革，是时候在真实系统环境中加以运用了。

2 抽象的区块链

Tezos 试图以最广泛的方式表示一个区块链的协议，并且试图保持作为一个原始协议的有效性。一个区块链的目标是表征一个正在被并发修改的唯一状态。为了被避免两个同步发生修改的冲突, 它使用账本表征状态，也就是一系列被运用到初始状态的转化操作。这些个转化是区块链的 “区块”, 而且 — 在比特币里面 — 这个状态主要是未被花费的输出的集合。因为区块由许多并发节点异步创建，形成区块树。每个叶子节点都代表一个可能的

状态，以及一个不同的区块链。比特币认定只有一个分支是有效的分支，而最有效的分支就是那个拥有最高难度的分支。区块，就像它的名字所暗示的那样，实际上是把多个操作绑定在一起（在比特币系统中，称为交易）。这些操作随后被应用到状态上。

2.1 三层协议

非常重要的一点是要区分加密账簿中的三个协议：网络协议，转账协议，以及共识协议。

其中元壳所扮演的角色是以一种透明的方式处理这个网络的协议，并把转账和共识协议指派给抽象的实现。

2.1.1 网络协议

比特币的网络协议本质上是一个 Gossip 网络，允许交易广播，以及区块下载和发布，以及节点发现，等等。这是最为集中的开发区域。例如，在 2012 年通过 BIP0037 引入布隆过滤器加速简单的支付验证，不需要下载全部区块链的客户端。

这些对网络协议的改变相对而言不是那么有争议。可能起初存在对这些改变可行度的争议，但是所有的参与者的利益都在总体上一致的。

这些改变并不需要同步发生。一个人可以找到一种方式将比特币的交易上融入到自己宠物猫的图片内。如果足够的人开始用这种方式来发布他们的交易，那么矿工就会开始解析猫的图片来查询他们添加到区块链的交易。

尽管一个健康的网络需要有兼容性，协议上的创新竞争通常会让一个加密货币变得更加茁壮。

2.1.2 转账协议

交易协议描述什么致使转账生效。这在比特币上是通过脚本实现。首先，比特币是由矿工通过挖矿生产出来。之后将脚本增添在这些币之上。

这样的一个脚本也就是所谓的 “未花掉输出”。交易通过提供脚本估值为真的参数合并输出。这些参数可以理解为作为锁的密钥和脚本。

在简单的转账中，这些脚本仅仅是做签名的审核，但是也存在更复杂的脚本。这些输出被添加在一起并被放置到一组新的输出之中。如果被花掉的输出的数量比分配的大，那么矿工可以获得差额。

和那些对网络协议的改变相比，对转账协议的改变更加富有争议性。尽管一小部分人就可以通过使用猫图片广播算法单方面的改变转账协议，但是要对交易协议本身进行修改则要复杂的多。这样的改变通常并不影响区块的有效性，因此仅仅要求大多数的矿工的同意。这些通常被称为“软分叉”。

那些相对不那么有争议性的改变在这里有更多的机会被实现。而对交易可锻性的修改，零币的引进则是一个转账协议层面的改变，则很有可能产生争议。

2.1.3 共识协议

比特币的共识协议规定了共识需要被建立在难度最高的链上，并包含矿工接受收益的细则。它允许矿工从 coinbase 上获取交易，规定了难度调整规则，有效块的标准，以及哪个是“主链”的一部分。

这是目前最核心，也是最难修改的协议，通常需要“‘分叉”来对老的区块进行排斥。对于 PoW 系统而言，目前所依赖的是 SHA256 加密算法。

2.2 网络壳

Tezos 将这三个协议进行了分离。这个转账协议和共识协议在一个封闭的模块中被实现，置入一个负责维护区块链的通用的网络壳内。

为了让协议变得更加通用，我们定义了以下的接口。我们让区块链来代表当前的经济“状态”，在 Tezos 中命名为 **Context**。这可能包括大量账户余额和其它信息，例如当前的区块高度。区块被视为将旧状态转换为新状态的操作。

在这个方面，一个协议可以被描述为仅有两个函数：

- **apply** 需要上下文（Context）和区块作为参数。返回一个有效的 Context 或无效结果（应该是区块无效）

- **score** 需要上下文（Context）作为参数，返回得分，该得分可以让我们来对比不同区块链的叶子节点来决定哪支是主链。在比特币里，我们仅仅是简单地记录总的难度，或者是上下文中的链，并返回这个值。

值得注意的是，这两个函数可以实现任何一个基于区块链的加密账本。除此以外，我们针对上下文本身附加那些函数，并暴露给协议以下两个函数：

- **set_test_protocol** 用一个新的协议来替代测试网中使用的协议。（特别是通过持币者投票选举出的那个）

- **promote_test_protocol** 用目前的测试协议来替代目前的协议。

这两个操作让协议来确认自己的替代品。尽管种子协议依赖于一个简单的绝大多数投票通过机制，未来我们可以引入更复杂的投票规则。例如，股东可以通过投票来要求特定的特性被加入到未来的协议中区。这个可以通过在协议内融入一个证明检测机制并且要求每一个修改包含一个合规性证明，来加以实现。

3 股权证明

Tezos 可以实现任何一个类型的区块链算法：包括 PoW，PoS 甚至中心化的方式。由于 PoW 的固有缺陷，Tezos 的种子协议目前按照 PoS 的方式实现。PoS 系统毫无疑问面临很多的理论上的瓶颈，这里我们要解释一下我们是如何来解决这些问题的。[5]

3.1 股权证明可行吗?

任何一个 PoS 的系统都存在严重的理论阻碍。主要反对观点包括以下：一个新的用户下载一个客户端，并且首次和网络互连。他接受到两个区块树，从最开始的创世块开始。两条分支展现出一个非常茁壮的经济活动，但是它们代表了两个完全不同的历史。一个非常明确的被攻击者所塑造，但是怎样才能辨别哪条是真正的链条呢?

以比特币为例，正宗的区块链是哪个代表着最大工作量证明的分支。这并不意味着重写历史是完全不可能的。但是这样做是非常高的成本，特别是当一个人的算力挖出块的时候。在一个 PoS 系统中，区块被股东进行签名，一个前股东-他已经退出，可以使用他的旧的签名来无成本的分叉区块链，这被称为 Nothing at stake 问题。

3.2 缓解负面影响的措施

尽管理论上的批评似乎是无丝毫漏洞，但是还是存在有效的消除影响的手段的。一个很重要的洞见是这里有两种分叉。一个非常深，可以重写一

[5]我们的技术白皮书提供了一个全面的技术性的对 PoS 的描述

个相当久的历史，一个是那些只是试图进行双花的。在表面上，两者之间只是存在一个数量上的不同，但是在实际上，激励、动机以及缓解方案都有很大的不同。

没有任何一个系统可以做到无条件的安全，比特币做不到，公钥加密学也做不到。系统的安全措施是针对特定威胁模型设计的。这个模型如何精准地对应现实是一个非常精细和实际的问题。

3.2.1 审核点

不规则的检查点可以有效地避免一个长链重组问题。检查点可以成为一种黑客攻击方式。就像 Ben Laurie 所指出的那样，比特币对检查点的使用有损于它的去中心化 [14]。

但是实际上，永久的或者是半永久的检查点都是没有问题的。人类现有的机构有足够的能力在几个月的时间内针对一个单独的哈希形成一个共识。这个哈希可以在全球主要的报纸上发表，刻在餐桌上，喷涂在桥上，以及被编程在歌曲里，被标记在新的水泥平面上，纹在你的宠物雪貂身上，有数不清的方式可以来记录这个检查点，让作假变得不可能。而且，它可以在几分钟内形成共识并在一个中心化的系统内获得更加安全的实现。

3.2.2 统计检测

转账可以指向属于最长链的区块，因此可以间接地来签这个链。如果一个攻击者试图复制一个长链，那么只能在最后检查点后生成他控制的币的转账当一个长的有效的链显示一个相对大额转账时，存在方法可以以统计学的方式甄别真伪。

这些技术通常被称为 TAPOS，对于短的链，因为样本太少所以不足够进行有统计学意义的测试，所以不是特别有效。但是和一个技术同时使用是，这个方法可以用来处理短期的分叉，产生一个对两类分叉都足够健壮的合成选择程序。

3.3 Nothing-At-Stake 问题

一个有趣的解决问题的方式被 Vitalik Buterin 通过 Slasher[15] 代码提出。但 Slasher 仍然未摆脱 PoW，并且假定了一个可行链的长度。

这里我们引入来一个创新的观点，就是对双重签名者的惩戒。如果对一个交易的签名延迟，这个交易可以被任何的双花企图被侦测到的时候被撤回。这足以防止一个自私的用户为了获得这个分支成功所带来的收益而机会主义地试图签署一个分支加。然而，一旦一个奖励被支付，这个动机也就不复存在了。因此我们使用一个延迟的方法来让 TAPOS 变得在统计学意义上有效，或者为了让检查点变得更加有效。

为了让用户表现的更加诚实，我们推出了一个 ticker 系统。一个潜在的矿工必须烧掉一定量的币，让他来行使挖矿权。这个数量的币被自动地返还给他，如果他没有能够进行挖矿，或者耽搁了很长一段时间。

为了让用户在长时间与互联网联通时不暴露私钥，需要每次使用一个不同的签名密钥。

3.4 威胁模型

没有任何一个系统可以做到无条件的安全，比特币做不到，公钥加密学也做不到。系统的安全措施是针对特定威胁模型设计的。这个模型如何精准地对应现实是一个非常精细和实际的问题。

比特币的确提供了一个有趣的保证，试图容忍没有道德立场的自私的参与者. 只要矿工不进行勾结，就不必要来假定某个参与者是诚实的，而仅仅假定他们想赚更多的钱而不是摧毁这个网络。然而，“非勾结” 成为一个非常重要的条件。而比特币所谓 “不需要信任” 这个特性往往被过度宣传却没有人真正去了解。

通过引入检查点机制，PoS 系统同样可以实现类似比特币的安全特性。

事实上，理论上仍存在这种可能，就是一个攻击者可以购买前持币人的旧密钥对网络进行攻击，而这样对他们没有任何不利后果。在这种情况下，需要一个更强健的假设，那就是参与者，也就是多数的目前或者从前的股权持有者，都不能够在一个针对网络攻击中被容易地贿赂。这种情况下，“股权” 在股权证明的角色是避免恶意攻击者在共识集群中做出不利的选择。

4 可能的发展

在这个部分中，我们将探索一些特有价值的想法，并将其整合到 Tezos 的规划中去。

4.1 隐私保护转账

最紧急的协议升级将引入交易的隐私保证手段。我们知道两种方式来实现这个。一个是环形签名，另一个是非交互式的零知识证明机制。

4.1.1 环形签名

CryptoNote 建造了一个协议，使用环形签名来保证隐私。用户可以能够在花币的同时不暴露 N 个地址中的那个花了这些币的地址。双花者将会暴露，而转账也被证实为无效。这个和 Coin-join 协议 *without* 相类似，需要参与交易混洗的地址之间的合作。

环形签名的一个主要的优势是他们的设置相对简单，比起 NIZKPK, 他们也更加依赖成熟的加密算法，这让他们更加能够经受时间的检验。

4.1.2 非互动零知识证明

Matthew Green 和其他人建议使用 NIZKPK 来实现区块链虚拟货币的转账不可追踪性。他们最新的成果是能够在 Merkle 树上维护带有附加秘密信息的币集合。提交的币通过在树中提供一个附加秘密的币被重新认定。这使用一个相对新的原语 SNARKs，来建立可以被高效验证的非常小的证明。

这个技术是有吸引力的，但是也有一些问题。加密学的原语比较新，并且没有被像椭圆曲线加密法那样被严密地验证过。

第二，这些证明的构建依赖于 CRS 模型。这意味着需要一个被信任的设施，尽管使用安全的多元计算可能会减少这样设置被攻破的风险。

4.2 修改规则

4.2.1 宪政主义

可能通过在协议内集成一个检验器的方法让任何修改携带一个形式证明，来保证它们对特定特性的接纳。这在事实上执行了某种类似宪法的特性。

4.2.2 Futarchy

Robin Hanson 曾经倡议我们对价值进行投票来决定我们的信仰 [16]。根据他的理论，最有价值的体系是多数人的共识，而哪些政策对实现那些价

值是最优的则被留给了预测市场。

这个系统可以在 Tezos 上得以提现。持有人可以先在一个受信任的代表价值被满足的信息收集器进行投票。投票的内容可以是 Tezos 币和一系列国际货币的汇率。一个内部的预测市场可以被建构出来来估测各种代码修改对这个指标的改变。对这些合约的做市可以收到赞助，通过给做市商发行代币来改善价格发现和流动性问题。在最后，被认为是最可能用来改善指标的方案被系统自动添加。

4.3 解决搭便车问题

当集体成员都会通过行动带来好处，但是没有一个单个成员可以通过单独采取行动而自己获得好处时，就会产生所谓的集体行动问题。这也被称为搭便车问题。这里有加密货币所有者可以提升平台的影响力以及来保护平台的合法合规性。

4.3.1 提高知名度

在 2014 年七月，比特币的市值已经达到了八亿美元。如果仅仅是花费比特币货币基数的 0.05% 用于慈善事业，比特币就可以每周捐赠一百万美元。假如在 2014 年比特币按照这个数量进行慈善捐赠，那么比特币的市值将不止上涨区区 0.6%。我们认为这个答案是显然是一定会。比特币持币者可以赢得尊重的同时获得经济收益。

然而，比特币的使用者不能够达成一致。这个种类的共同行动问题可以被 Tezos 所解决。一个协议的修改案可以建立一个制度，让所有的持币人来进行每个月的投票，决定在几个地址上的 0.05% 货币总量如何被花掉。持币人的投票共识的内容可以是避免对某个地址进行投票，避免稀释，也可以是如何更加好的进行慈善捐款。

4.3.2 集资创新

创新募资可以通过在协议内建立赏金机制来实现。一个协议可以定义单元测试和自动。

相反的，一个创新者设计了一个新的协议可以在协议里的给自己一个奖励。尽管他的协议可能会被复制，造成奖励被取消，但持币用户的共识很可

能会是把奖励发给最初的创造者。持币者们将不太可能会因为拒绝一个非常合理的奖金而选择背叛。

结论

在本文，我们这里给你展示了目前加密货币所面临的问题，而 Tezos 则恰恰给出了一个好的解决方案。尽管通过发明一种新的货币的方法来阻止加密货币过度碎片化的方式略带讽刺意味，Tezos 真正的目的是成为最后的一种加密货币。

不管其它的协议产生任何的创新，Tezos 的持币人都有权利接受这些新的创新。Tezos 解决共性问题及利用 OCaml 上轻易实现协议的能力都将使得 Tezos 成为最具活性的加密货币之一。

参考文献

[1] Peter Suber. Nomic: A game of self-amendment. `http://legacy.earlham.edu/~peters/writing/nomic.htm`, 1982.

[2] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. `https://bitcoin.org/bitcoin.pdf`, 2008.

[3] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. `https://github.com/ethereum/wiki/wiki/%5BEnglish%5D-White-Paper`, 2014.

[4] Nicolas van Saberhagen. Cryptonote v 2.0. `https://cryptonote.org/whitepaper.pdf`, 2013.

[5] Matthew Green et al. Zerocash: Decentralized anonymous payments from bitcoin. `http://zerocash-project.org/media/pdf/zerocash-extended-20140518.pdf`, 2014.

[6] Thomas Schelling. *The Strategy of conflict*. Cambridge: Harvard University Press, 1960.

[7] Bitcoin Wiki. Weaknesses. `https://en.bitcoin.it/wiki/Attacks#Attacker_has_a_lot_of_computing_power`, 2014.

[8] Gaving Andresen. Centralized mining. http://bitcoinfoundation.org/centralized-mining/, 2014.

[9] Bitcoin Wiki. Tragedy of the commons. https://en.bitcoin.it/wiki/Tragedy_of_the_Commons, 2014.

[10] Bitcoin Wiki. Dominant assurance contracts. https://en.bitcoin.it/wiki/Dominant_Assurance_Contracts, 2014.

[11] Simon de la Rouviere. Not actually capped at 100 billion? https://github.com/dogecoin/dogecoin/issues/23, 2013.

[12] Debian project. Computer language benchmarks game. http://benchmarksgame.alioth.debian.org/u32/index.html, 2014.

[13] Scott Owens. A sound semantics for ocaml light. http://www.cl.cam.ac.uk/~so294/ocaml/paper.pdf, 2008.

[14] Ben Laurie. Decentralised currencies are probably impossible, but let's at least make them efficient. http://www.links.org/files/decentralised-currencies.pdf, 2011.

[15] Vitalik Buterin. Slasher: A punitive proof-of-stake algorithm. https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm/, 2014.

[16] Robin Hanson. Shall we vote on values, but bet on beliefs? http://mason.gmu.edu/~rhanson/futarchy2013.pdf, 2013.