从OP\_CAT到状态证明和BitVM，如何让比特币支持ZK？

作者：Janos Nick，Blockstream

编译：白丁 & Faust，极客Web3

摘要：这篇文章简明扼但又一针见血的指出了如何让比特币支持ZK验证功能，涉及的具体话题包含比特币UTXO和脚本在功能性上的缺陷、Taproot和OP\_CAT以及BitVM和Chain State Proof等概念的大致内容。文中抛出了比较鲜明的观点：

比特币协议引入ZK是必然趋势，对此有两种路线：一种是让比特币脚本直接支持SNARK验证，需要借助OP\_CAT操作码，而OP\_CAT最终通过的概率很大；第二种路线是基于BitVM的，需要引入欺诈证明的方式，而ZeroSync团队还针对性的提出了Chain State Proofs来降低节点客户端验证历史数据的成本。



正文：为了更深刻的理解比特币，我们最好把它当做一个社会系统来看待。比特币早期启动的时候，开发者就确定了比特币节点需要运行的软件程序，就像确定了一套社会系统所遵循的规则。比特币这个社会系统之所以能够稳定运转，是因为大家在“比特币的本质是什么”、“应该是什么“等关键问题上存在某种共识。当然，共识的达成并不容易，人们在面对上述问题时，仍然存在广泛且不断演化的分歧。



这可以追溯到比特币的历史来源问题。当初中本聪在发布比特币白皮书时，曾说："我在研究一套全新的电子支付系统，这套系统完全是P2P的，不需要依赖于任何第三方"。这段话发表于密码朋克邮件列表（一个成立于1992年的电子邮件讨论组，由一群关注隐私保护和密码学技术的密码学家及技术爱好者组成）。

然而，比特币在产品设计层面限制了数据吞吐量。其单位时间内能处理的交易笔数有限，如果待处理的交易数量快速上升，用户为了快点交易成功，就会发起价格战，迅速拉高付出的手续费。比特币网络内手续费最高的单笔交易出现在2024年出块奖励减半后，一笔上链优先级中等的交易手续费达到150美元。可以说，比特币网络昂贵的交易手续费已成为一个难题。

为了解决交易手续费问题，人们把很多资源投入到闪电网络开发工作中。但根据一篇发布于2016年的论文，闪电网络在实践中最多只能支持数千万用户，无法实现其全球支付系统的愿景。

除了交易手续费过于昂贵外，还有一个问题，就是比特币始终无法实现其愿景中想达到的匿名性。中本聪在曾在密码朋克邮件讨论组中指出，比特币具有隐私保护功能，交易发起者可以是完全匿名的。然而，虽然交易发起人不需要KYC，但比特币链上的交易数据泄露了许多信息，很大程度上暴露了用户隐私。

虽然有一些附带隐私功能的钱包客户端一定程度解决了上述问题，但这些钱包客户端的开发者却面临着大大小小的威胁。比如，Samourai CoinJoin钱包的开发者在2024年4月被FBI逮捕，而在一周后，Wasabi钱包的开发者就关闭了他们的CoinJoin协调组件。显然，这些所谓的隐私钱包并不完全值得用户信任。

总结下来，直到今天比特币的许多理念还远未实现，相关技术仍处于不断发展中。即使这样，比特币社区中很多人还认为比特币的协议设计应该保持不变，但也有很多人像我一样，热衷于对比特币做出改进。那么，比特币应该向何种方向改进呢?



针对上述问题，比特币社区中有很多提案，理论效果最好的应该是和ZK及SNARKs相关的。借助ZK和SNARKs，可以实现如下特性：

1. 显著改善隐私性：使用同态Peterson承诺对交易数额和Range Proof显著提高用户隐私性（如Blockstream的 Element侧链中所做的）；通过可链接签名（如Monero ）隐藏交易痕迹；实现真正的私密交易（如 Zcash）。

2. 提高交易吞吐量

其实有很多技术手段可以解决比特币身上存在的问题，但为什么直到今天，这些技术仍没有被加入到比特币协议中？这是因为比特币协议很难被修改。比特币生态里没有类似于以太坊基金会的组织，对协议的任何修改都需要社区达到高度共识才行，这里涉及到大量的博弈和权力制衡，所以不同于以太坊那样每年都会有EVM操作码的更新，比特币协议自问世以来，变更就非常少。

其实，协议难以被修改在某种程度上是好事，如果修改比特币协议很容易，那么对其进行恶意更改和攻击也会很容易。这就引出了一个问题：在不改变比特币协议设计的情况下，有什么手段可以改善比特币的性能？



要回答这个问题，我们要先回顾关于比特币的知识。如果我们要将比特币转账给别人，需要先创建一笔交易，将其广播到比特币网络中。交易的输出数据会说明转账的BTC数额，BTC接收者可以再创建一笔新的交易，来花费收到的BTC。此后，这笔新的交易又会产生新的输出数据，并将BTC发送给其他人。

这里要注意的是，比特币没有以太坊那样的全局状态，特别是没有账户的状态，只有交易输出数据。每笔交易的输出有两种状态：已被接收者花费或未被花费。未被花费的交易输出就是我们熟悉的UTXO。

当然，除了关联的BTC数额外，每个交易输出都有一段附加程序，用一种叫比特币脚本的语言来编写。谁能向这段程序出示正确的证明Witness，谁就能花费这笔交易输出（UTXO）。比特币脚本本身是一种基于堆栈的编程语言，包含一系列操作码，前述UTXO的附加程序往往由多个操作码组成，它们基于堆栈完成计算并将结果再放回堆栈。

常见的比特币脚本有很多种类型，从比特币启动之初就已经存在了。举个例子，比特币中最常见的脚本程序由公钥+检查数字签名的操作码组成。该操作码规定，要花费/解锁某个UTXO，必须出示对应公钥的数字签名。



这里我们来总结一下比特币脚本的功能。首先比特币脚本能做什么？

比特币脚本不能做什么？

在比特币早期版本中，上述脚本中”不能做“的事情，有一些其实可以做，但部分功能后来被中本聪禁用了，原因是中本聪发现这些操作码存在漏洞。例如，可以把堆栈中2个元素合并的操作码OP\_CAT可以被用于远程攻击比特币节点致其崩溃，中本聪出于谨慎，禁用了OP\_CAT，其他的一些操作码也遭到了禁用。

那么，比特币脚本是否可以验证SNARK？理论上虽然比特币脚本非图灵完备，但它的基本操作足以验证任何计算，可在实践中SNARK验证还是无法实现，因为验证步骤所需的程序尺寸超过比特币的最大区块限制——4MB。

也许我们可以尽量在大型有限域中进行算术运算，但这样的成本非常高，比如BitVM实现的两个254位整数的乘法，相关的比特币脚本尺寸达到近8KB。

而且，在没有OP\_CAT的情况下验证Merkle证明的成本也很高，因为这需要类似于for循环的操作。



那么再回到前面的问题：为什么我们不能简单地改变比特币协议，添加功能更强大的操作码？

正如之前提到的，要在新的协议规则上达成多数共识非常困难，因为比特币生态没有中心化决策者，任何对比特币脚本的改进提案都有很多反对意见，大家的立场、角度都是不同的。在比特币网络中，没有很好的办法来衡量社区是否达成了多数共识，在这种情况下强行推动更新，会导致链分叉。

当然，比特币也不是完全一成不变的，最近的更新是2017年的SegWit和2021年的Taproot。



Taproot升级改变了许多规则，从理论发布到真正被激活落地，花了三年半。Taproot被启用的关键因素是：它没有改变现有安全假设，并对比特币协议做出了明显改进。例如，它允许使用 Schnorr签名代替ECDSA，二者都是基于离散对数假设，并使用相同的椭圆曲线，但前者比后者效率更高、计算量更少。

进一步说，Taproot对比特币的改进主要分为以下三部分：

第一，Taproot降低了有大量选择性分支的脚本的验证成本，可以让比特币支持更复杂的程序；

第二，Taproot缩减了需要在链上揭示的脚本数据，你可以将多段脚本程序组装为一棵Merkle树，每段脚本位于不同的叶子上，如果你要触发某段脚本，只需揭示出示其所在的叶子及Merkle证明；

第三，Taproot还添加了其他的机制设计。



话说回来，既然比特币有Tarpoot这样添加较强大功能的先例，为什么不添加一个专用的操作码来验证SNARK呢？这是因为添加一个所谓的OP\_SNARK操作码与Taproot升级有很大不同。

首先，OP\_SNARK的设计思路有很多，很难让大多数人都支持某种单一方案；其次，如果这类提案通过了，所有比特币节点都要支持该特定的OP\_SNARK方案，这将增加巨大的技术负担。

此外，OP\_SNARK本身的复杂性也是不小的挑战。如果不包括测试的话，Taproot只添加了约1600行代码，这是人们可以接受的，而相比之下，OP\_SNARK包含的代码要复杂得多。

再有，谁来审核OP\_SNARK操作码该不该被激活？如何在没有几个人理解其细节的情况下在比特币生态内获得共识？这些都是问题。因此综合来看，OP\_SNARK升级是不会在短时间内发生的。



但是，还有其他途径可以在比特币脚本中验证SNARK。我们可以添加较为简单的操作码使比特币脚本功能性更强大，让人们可以在脚本中实现SNARK验证器程序。但事实上用比特币脚本语言编写SNARK验证程序的难度很大。

因此，Blockstream研究团队正在开发Simplicity，这是一种旨在替代比特币脚本的编程语言。Simplicity专为区块链共识系统设计，故意设计成不图灵完备的，易于静态分析和形式化验证。



下面我们要谈及一个非常简单但又很重量级的提案，它可以使比特币脚本变得更强大，即 OP\_CAT操作码。前面我们提到，OP\_CAT存在于比特币的最初版本中，但这个操作码可以在特定条件下让比特币节点被DOS攻击，所以被中本聪禁用，现在比特币社区中有一些人想重新启用它。

OP\_CAT的功能是弹出堆栈顶部的两个元素，将它们连接起来，然后再放回堆栈。这听起来非常简单，但却能给比特币脚本带来巨大的功能改进。

例如，比特币脚本程序本来无法访问链上交易的金额等状态信息，但有了OP\_CAT这将成为可能；OP\_CAT还可以用于验证Merkle证明。总之OP\_CAT是属于底层操作码级别的升级，会衍生出非常多的新功能，很多人都提出过使用 OP\_CAT能达成的效果。

而OP\_CAT是否有助于在脚本中验证SNARK呢？答案是有帮助，因为支持验证Merkle证明就有助于验证基于FRI的SNARK，而OP\_CAT可以支持这点。在过去，涉及SNARK验证的脚本程序可能尺寸太大而无法放入比特币区块，有了OP\_CAT则可以压缩程序大小。

过去OP\_CAT已经被讨论很多年，越来越多的人认识到它在交易检查(introspection)中的作用。与其他提案相比，OP\_CAT的优势在于它以前在比特币脚本中存在过，因此更容易在社区中达成共识。然而，OP\_CAT启用后也可能导致某些人的MEV收益受损，所以比特币社区对其还未达成共识。

综上，比特币可能会有一个潜在路径，通过启用如OP\_CAT这样的简单操作码，让大家可以用比特币脚本中验证SNARK。另外值得一提的是，最近有一个名为“Great Script Restoration”的提案，启用了乘法操作码，允许所有算术操作码以任意精度操作。



此外，当我们考虑OP\_CAT对比特币网络的影响时，可以考察它通过后对比特币节点运行者的影响。为了使比特币具有抗审查性和去中心化，比特币社区希望尽可能多的人运行节点验证数据。如果比特币支持了SNARK验证操作，运行比特币节点的成本仍不会显著增加，这对于比特币的安全性和抗审查性并无多少危害。

目前，一个比特币区块可以包含最多4MB的数据，预计每10分钟挖出一个区块，几乎所有的区块都可以填满比特币脚本和Witness见证(类似于数字签名）。折算下来，每个区块目前最多可以包含80K次签名验证，平均每个区块支持7K到10K次签名验证，我的2020年版英特尔CPU验证一个比特币区块平均耗时3.2 秒。当然，影响区块验证速度的不只有签名验证的耗时。

此外，如果比特币交易日后支持ZK化，就算因此延长交易生成时间似乎也无伤大雅。对于用于长期存储资产的硬件钱包而言，它们往往带有屏幕且体积并不大，功能是存储密钥并生成签名。硬件钱包的CPU一般比较弱，如240MHz双核CPU，并带有一定内存，在签署比特币交易时响应非常迅速。



我做了一个小调查，询问用户能接受的签名设备生成证明的最长延时，许多人可以接受较长的等待时间，特别是能获得显著收益的时候。所以如果我们将ZK引入比特币交易中，似乎也没有太多麻烦。

上面我们花了很多篇幅来讨论应如何变更比特币的底层设计，但其实还有不少无需变更比特币就可以实现的应用场景。这里我想强调一个与BitVM相关的应用——Chain State Proofs，它结合了ZK，可以证明区块哈希的有效性。



这项技术给比特币带来了什么变化？首先，有了Chain State Proofs，就可以压缩比特币历数据的同步与验证工作量，大幅度降低运行节点的成本。目前在一台硬件好的设备上从创世区块同步并验证到最新的比特币区块，需要5小时30分钟，而在树莓派级别的设备上则需要几天，如果引入状态证明则可以大幅度压缩这个耗时。其次，链状态证明是与BitVM可以用到的重要部分，会对BitVM的实现有推动作用。

ZeroSync团队对Chain State Proofs进行了深入研究，并创建了一种更轻量的“header chain Proofs”，这种方案结合ZK，只证明比特币区块头的有效性，以此构成一条”header chain“，包含比特币历史上全部的85万个区块头，并针对每个区块头生成80字节的哈希。

这种方案需要对每个比特币区块头进行双重SHA-256计算以验证对应的PoW证明。ZeroSync的使用STARKs来生成比特币的header chain Proof，生成证明的成本约为4000美元，用我的浏览器验证该证明则只需要3秒钟。



但由于不包含区块中交易内容的验证过程，header chain proof只能假设拥有最多POW证明的区块链是有效的，并默认让比特币客户端同步这条链上的最新区块。这种场景下，虽然攻击者可以创建包含无效交易的区块，并在该区块之后添加更多区块，并生成header chain proof来蒙蔽同步历史数据的比特币客户端，但这样做的话攻击成本极其昂贵，并且会被现有的比特币全节点客户端直接揭穿。



然而，尽管这种攻击场景的成功率很低，但如果可以让攻击者窃取巨大金额的BTC，那么header chain proof就无法被认作是万无一失的方案。如果我们想证明完整的链状态，就需要直接证明比特币区块中所有的内容都有效，包括基于secp256k1椭圆曲线的ECDSA和Schnorr签名验证。

比特币每个月的历史区块中都可以包含3000万个签名，历史上总计包含25亿次签名运算，以及大量的SHA-256运算。这样下来比特币网络每个月产生的区块数据约7GB，所有历史数据总计超过650GB。而实际情况中这个数字可能是2到3倍。



现在我们再来看BitVM。BitVM使得比特币可以验证任何计算任务，是无需变更协议来实现 SNARK验证的最佳路径。BitVM使用两种技术绕开了比特币区块对脚本大小的限制。首先，它使用了Taproot MerkleTree的脚本结构；

其次，它启用了可以跨单个脚本访问的KV存储方案，允许连接到超级多的脚本程序。不过，比特币协议不强制保证上述KV存储方案的完整性，BitVM需要通过欺诈证明来检查恶意Prover，如果Prover发布无效声明或有问题的KV存储，其他人可以在比特币链上发起一笔交易，表明Prover行为不当，并取走其事先质押的资产。



总结一下，比特币正面临重大挑战，大家提出了各种方案解决这些问题，然而，这些提案不会很快被比特币社区采纳，对协议的改变也非短期内能完成的，这既是好事也是坏事，这也意味着比特币是去中心化的、较为安全的。

比特币社区许多人对SNARK/STARK的潜力感到兴奋。在中长期内实现SNARK验证的最可行方法，很大概率是BitVM，但它需要更多的研发投入才能在实践中发挥作用；

重新启用OP\_CAT操作码也是一种思路，但需要证明重启该操作码的收益远大于风险，并调查哪些简单的操作码可以允许在比特币脚本中验证SNARK，或者探究类似OP\_CAT的功能可以实现什么场景。无论选择哪种方案，比特币社区的最终宗旨一定是让产品变得实用，支持更多可落地的场景。