Citrea的ZK-Rollup: 比特币未尽之事

最近，宣称第一个采用ZK-Rollup 方案的Bitcoin Layer-2项目，Citrea，在即将上线测试网之际，受到了来自社区的疑问。其宣传Citrea 采用了一些欺骗性的话术，将它们的项目进行了过度包装。在这个风口浪尖，毫无疑问就是研究Citrea 和BitVM 最好的时机。

一切的起因是：ZKP 在基于以太坊的Layer2 上大显身手，使得许多项目都在寄希望于ZKP 是否能够同样的用于Bitcoin Layer-2 上。显然，大量的开发者（特别是最挑剔的那些人）对比特币Layer2 的发展非常不看好，而它们的担忧也是很有道理的。常规的Layer2, 如基于以太坊的optimisic, zkrollup, 都是通过链下计算，链上验证的方式，对一层网络的数据进行压缩，以及对计算资源进行扩展。单由于Bitcoin 网络并不支持复杂逻辑的validation，因此这个方案难以实现。因此，大部分声称自己采用了零知识证明技术的Bitcoin Layer-2 项目，都采用了一种叫做Sovereign Rollup 的方案。

The Sovereign Rollup

长话短说，Sovereign Rollup 的关键在于：bitcoin 网络只负责将关键的承诺信息用铭文的方式存储在Bitcoin Network 中，而验证全部由链下进行（客户端验证）。总而言之，比特币网络中的数据，解释权属于Layer2 网络。对于一个不清楚规则的参与者，这些比特币网络中的数据毫无意义。但是如果参与者清楚这些数据的使用方式，那么任何一个用户都能够自己建设节点。从这个角度看，Sovereign Rollup 的确具有相当的积极意义，它能够达到Layer2 网络的一些基本要求，即数据的finalized 能够被比特币共识所保护，同时其Layer2 的解析也可以是去中心化的。

听起来第二种方案似乎是第一种方案的妥协，但这点事实上存在极大的争论。大量方案二的支持者认为，Bitcoin 本来就不必和Ethereum 一样：第一种方案的支持者，事实上依然在用以太坊的观点看待比特币，即，它们认为数据的存储以及验证，都应该发生在Layer-1。但事实上这并不是必要的，因为数据的验证非常浪费Layer-1 的效率。在理想的情况下，任何一个智能合约都应该只由一个节点执行，而其他节点只需要验证执行的结果以及其对应的证据就足够了。强制要求所有Layer-1 的节点进行验证，本身就并不一定是最佳答案。



sovereign rollup原理: 在链上存证，链下验证

这种哲学正是Sovereign Rollup 的核心，即基于客户端验证的思想。总之，妥协也好，创新也罢，就在大家已经准备接受Bitcoin 独有的Layer-2 技术路线时，BitVM 诞生了。

BitVM

关于BitVM 到底是什么，对于社区中的大部分人而言是神秘的：BitVM 似乎提出了一个能够在一层进行ZKP 验证的方案，但是这个方案又最终需要用乐观Rollup 的方法去执行。早期的BitVM 还提出要在比特币上模拟CPU，以实现一切可能的计算 - 听起来就像在Minecraft 里面使用红石建造计算机一样。这些怪异的说法让多数人对其嗤之以鼻，并采用了更加实际的路线如DLC 或是账户托管来保护Layer-2 的资产安全。然而，也有一部分项目认为BitVM 是Layer-2 的未来，并且投入了大量的研究，例如今天的主角Citrea。显然，搞清楚BitVM 究竟是什么至关重要。

为了弄清楚BitVM 的真相，我们将采用最直接的方式：分析BitVM 的Github Repository。总体来看，BitVM 仓库的核心包括：

一个虚拟的Bitcoin Script 运行环境。

用于拼接Bitcoin Script 脚本的辅助函数。

目前BitVM 的构建，主要是通过将Bitcoin 的Opcode 进行组合，来实现更加复杂的计算；而并非采用初始版本中提到的，使用OP\_XOR 去构建计算组件。显然，这是一个更加实际的路线。让我们用一个实际的例子来看它到底是如何工作的：通过将Opcode 进行组合，进行uint32 类型的整数计算。



BitVM实际上很像是积木：由基础的OP\_CODE 拼接而成

在这里例子中，我们能够发现两种不同的元素: Opcode，以及已经被封装过的一系列新的Opcode。其中，带有OP\_ 前缀的是比特币脚本原生的Opcode，而诸如u8\_add\_carry 这样的函数，则代表着已经被封装过的，由BitVM 自定义的新Opcode。显然，这个函数u32\_add 本身，也会被用于其他进一步的Opcode 构造。

这听起来有些小儿科，但不要小看这些功能，它实际上是进一步构造ZKP 的基础；通过u32 的计算，就能够逐步构造bigint，乃至于bn256 的计算，最终得到证明构造系统。并且从仓库看来，它们已经得到了许多值得一提的成就：它们已经能够构建基于Groth16 的验证函数！看起来，它们离最终构建出可用的ZKP 的距离已经不遥远了。

Citrea与BitVM

在了解到了BitVM 的现状之后，我们当然乐见其成，这对整个Bitcoin Layer-2 的生态都有着极为积极的意义。但是我们也同时注意到，Bitcoin Layer-2 的爆发，和BitVM 的开发步调之间，存在着一定的不同步。毕竟时间不等人，宣称自己采用了BitVM 的项目，其实更多应该被理解为“将会采用BitVM”，毕竟尽管成果喜人，但它还暂时不能够投入商用。这样看来，Citrea 宣称自己为第一个Bitcoin 上的zk-rollup，就难免引起争议了。



在BitVM 正式完成前，Citrea 是一种自主Rollup

我们首先最关心的问题是：目前Citrea与BitVM，乃至于ZKP 之间的关系到底是怎样的？显然，最好的办法依旧是深入Citrea 项目的代码，来观察目前它们的进展。在分析了它们测试网的官方仓库之后，我们发现了一些有趣的事情：

（1）Sovereign Rollup

Citrea 目前依旧是一个典型的Sovereign Rollup，而并没有实现真正等效于Ethereum Layer-2 的zk-rollup。关于这点，事实上Citrea 在它们的测试网公告中也提到，即目前BitVM 与它们的核心组件，Clamentiane (https://www.blog.citrea.xyz/unveiling-clementine/)，还没有完成。当前的情况是，它们会将Citrea 网络中的账户状态保存为一个默克尔树根，并且在比特币上，通过Inscription 的方式，同时记录默克尔树根的变化以及其对应的ZKPs。显然，在比特币上进行铭刻是无需任何限制的；因此，这些ZKPs 和默克尔树根，需要由Citrea 的节点进行主动的验证。与此同时，比特币只负责对其状态进行最终化，但并不负责保证其正确性。因此很遗憾，Citrea 目前依旧是典型的Sovereign Rollup；其BitVM 参与的部分也仅在于节点对ZKPs 进行主动验证的部分。



自主验证的局限：无法确保链上证据的有效性

（2）RISC0

Citrea 对于它们的ZKPs 以及zkEVM 部分，采用了RISC0 提供的SDK 作为它们构建的基础。RISC0 是一个2022年诞生的，比较新的ZKPs 解决方案。RISC0 和Cairo 有着非常类似的愿景 - 通过构建一个zkVM，令ZKPs 能够进行任何通用计算；但是相比于Cairo，RISC0 本身并不是一种语言，它可以直接通过Rust 当中提供的SDK 进行构建。之后，再通过其定制的zkVM 进行计算。除去RISC0，基于zk-STARK 的Cairo VM 也被Bitcoin Layer-2 所青睐。我们注意到不少项目都在采用这种通用计算+zkVM 的模式，来替代zk 语言 + zk Prover + zk Verifier 的模式。这是由于Bitcoin 中暂时无法提供zk Verifier，因此对于Sovereign Rollup 而言，直接采用zkVM 无疑是更加经济和高效的办法。

结语

首先关于Citrea 最近的争议，它们的团队很明确的指出了，其对BitVM 的兼容还尚未完成；这个结论代表了能够完全继承比特币安全性的Rollup 还没有出现。但是在观察整个项目的过程中，我们也看到了基于Bitcoin Layer-2，还有着连续不断地创新；同时，还有大量的项目在前仆后继的挑战比特币的zk-rollup，如Bitlayer，Alpenlabs，BVM network 等，这些都在驱动着我们持续跟踪和分析比特币当中Rollup 的进展。