解读DLC.Link：创新性跨链解决方案 激活比特币生态潜力

作者：Bixin Ventures

### 引言

作为BTC早期持有者和坚定支持者，Bixin Ventures始终致力于BTC生态。我们认为BTC资产在各链上的优秀DeFi协议中应发挥更大作用，获得更多收益。因此，高效的跨链桥是必需的，所以我们去年初期就投资了DLC.link。

随着铭文资产兴起及比特币数据流网络扩展，比特币应用增多。面对其有限的原生智能合约功能，Bixin Ventures视DLC.Link为创新解决方案。FTX崩溃后，市场急需安全、去中心化的比特币替代品。DLC.Link与我们愿景相符，致力于建立安全、无信任生态系统，推广比特币在多种场景的应用。

## DLC.Link 技术解析

### DLC 技术如何运用Taproot升级中的Schnorr签名

比特币在Taproot升级引入了Schnorr签名，这是一种比ECDSA更为安全的替代方案，用于创建和验证密钥。Schnorr签名具有线性特性，它可以通过结合公钥和签名有效地一次验证多个交易[1]。谨慎日志合约 (DLC) 充分利用了Tarproot中已经实现的Schnorr和PTLC技术。因为这两项技术已经是比特币网络的原生功能，所以在使用DLC时，无需依赖L2或侧链。相反，DLC技术由整个比特币网络的算力保护，使其更为安全可靠。

在使用 DLC 技术进行借贷的情景中，我们通常采用Schnorr签名来处理有条件的结果。在这个过程中，双方各自生成一个独特的密钥，将其与双方各自的长期密钥结合，形成一个“公钥”，并提前准备好各个潜在的结果对应的签名，比如偿还或清算。接下来，一个去中心化的见证者网络通过谨慎日志号码来确认现实世界的结果，解锁预先准备的签名，从而在区块链上触发合约执行[2]。这展示了DLC技术在实际应用中的普适性、灵活性和安全性。

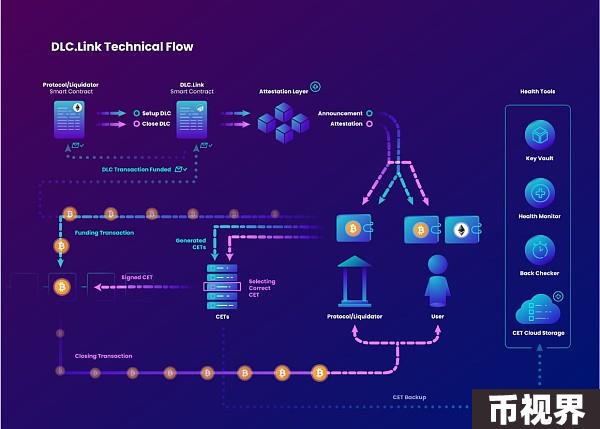
### DLC 技术如何在没有验证者的情况下确保链上比特币的安全

谨慎日志合约(DLC)不再需要外部验证者，而是直接继承比特币的安全性，确保链上安全[2]。在DLC.Link架构中，见证者(Attestor)网络是一种创新的设计。作为链下服务的见证者，他们通过智能合约积极监听区块链，以JSON的格式公开宣布DLC事件。从数据源（另一个区块链上的智能合约）获取对应的结果后，见证者对其进行证明并签署。为增强安全性，DLC.Link采用了5-of-7的见证者多签，降低了见证者和合约一方的共谋风险。DLC.Link管理合约会随机选取白名单中的无需信任的见证者[3]。设计上还允许第三方，组织运行见证者节点，从而愈加促进去中心化。去中心化见证者网络避免了单点故障，有效地促进了DLC的创建和签名请求。这一全面的设计不仅保护了比特币交易的完整性、安全性和灵活性，也证明DLC.Link才是整合BTC与去中心化金融(DeFi)的理想基础设施。

### FROST 签名如何推进DLC技术的安全性

DLC.Link 还整合了 FROST（Flexible Round-Optimized Schnorr Threshold Signatures）以提升安全性并简化见证者管理。FROST 的密钥再分享功能，赋予了见证者网络的动态管理能力，允许参与者的迅速调整而不影响服务的完整性[4]。在设置或更改见证者时，系统采用协作的方式生成密钥。整个过程同时创建了共享密钥和每个 Attestor 的单独密钥。这种密钥生成机制的灵活性确保了以后可以安全地修改见证者网络，而无需中断服务。

当区块链事件触发 DLC 合约的创建时，见证者中的协调员会广播 create-dlc 命令，从而生成共识支持的DLC公告。同样，当要关闭 DLC，区块链事件促使协调员安排 DLC最终见证的创建，其中每个见证者的签名确保了合约的安全性和可验证的结束。这一先进的升级引入了对见证者网络的动态调整能力，并显著降低了计算复杂性，同时保持了严格的安全标准。



## DLC.Link 的产品介绍

### dlcBTC

dlcBTC将于2024年初推出，站在去中心化金融(DeFi)的最前沿，通过谨慎日志合约(DLCs)，在以太坊网络上提供无需托管的比特币。使用dlcBTC，用户可以“自我封装”存款，BTC由自己锁定，由认证者保护锁定机制。dlcBTC存款人不向DLC.Link或任何第三方发送BTC。

在这一开创性的过程中，用户使用DLC安全地将他们的比特币锁定到DLC.Link中，从而创建等值的dlcBTC。DLC作为比特币区块链上的数字保险箱，在用户和协议之间建立了预先签署的合约。更加安全的是，密钥分发采用了2-of-2多签名UTXO机制，其中一个密钥由用户持有，另一个密钥在可信的证明者节点网络中智能地分发。

跨链桥由于其许多漏洞而名声不佳。DLCs是一个独特的解决方案，因为它解决了跨链通信的问题，而没有桥接的安全问题。与wBTC等桥接资产不同，dlcBTC不需要中介机构，而是选择由比特币网络的整个哈希率保护资产。dlcBTC v1将于2024年第一季度推出，将引入白名单访问、严格的安全措施、多功能交易对以及与现有DeFi协议的无缝集成等功能，标志着去中心化金融领域的重大进步[5]。

我们相信，在比特币之后，最去中心化的网络是以太坊。因此，通过dlcBTC，我们致力于将原生BTC引入到以太坊网络的DeFi，而无需构建另一个规模小、未经测试且更加危险的 L2 或解决方案。铸造的dlcBTC代币在知名的DeFi平台(如Curve和AAVE)中作为抵押品使用，并使用户能够积极参与DeFi活动，如投资、贷款和对冲等。

## DLC.Link 在比特币生态中的广泛应用

DLC.Link在更广泛的比特币生态系统中是一股变革力量，通过其谨慎日志合约(DLC)提供了一系列应用。这项创新对包括信贷交易台、首席经纪人、数字资产托管方、交易所和场外交易台在内的各方，都具有特殊意义。这一整合允许托管人将金融服务扩展到自我托管的比特币客户，支持对本地比特币进行托管和有条件的即时转移，以及向矿工和其他自托管比特币客户提供借款服务。

DLC.Link在去中心化金融（DeFi）中的作用同样广泛而有影响力。其整合促进了基于链上比特币资产的抵押贷款和稳定币的发展，为借贷协议提供了优势。此外，DLC.Link同样可以无缝扩展到跨链协议，以实现在不同区块链之间转移本地比特币。当下BRC-20资产火热，DLC.Link同样可以接纳它们进入跨链桥接 [6]。此外，它还支持在NFT市场（如OpenSea）和其他基于智能合约的平台中进行铭文资产的交易和借贷，为DeFi带来新的用例。



## 结语

Bixin Ventures之所以在早期种子轮就战略性地投资了DLC.Link，是希望其与去中心化金融（DeFi）的无缝集成，帮助解决比特币生态系统面临的挑战。DLC技术，克服了比特币的智能合约限制，更好地利用比特币强大的安全性，提供了一个更安全的解决方案。最近Taproot升级中的Schnorr Signatures更使得DLC.Link能够建立见证者网络，而FROST的集成进一步完善了见证者的动态管理。dlcBTC的推出同样应对了对安全、去中心化的自包装比特币日益增长的需求。此外，DLC.Link在其他多种产品应用层也暗含巨大潜力，包括CeFi中的自托管存款和DeFi中的跨链铭文。Bixin Ventures 战略性地投资了DLC.Link，展现了致力于塑造去中心化金融（DeFi）的决心。我们认可了DLC.Link在加密领域的创新贡献，并预见到它在为整个更广泛的加密生态系统提供比特币流动性方面将发挥关键作用。

参考

[1] Namcios. Why You Should Care About Taproot, The Next Major Bitcoin Upgrade. Bitcoin Magazine. Retrieved from https://bitcoinmagazine.com/technical/short-bitcoin-taproot-explainer

[2] Dryja, T. Discreet Log Contracts. Retrieved from https://adiabat.github.io/dlc.pdf

[3] DLC.Link. How DLCs Secure Bitcoin On Chain without Validators. Retrieved from https://www.dlc.link/blog/how-dlcs-secure-bitcoin-on-chain-without-validators

[4] Komlo, C., & Goldberg, I. FROST: Flexible Round-Optimized Schnorr Threshold Signatures. Retrieved from https://eprint.iacr.org/2020/852.pdf

[5] DLC.Link. DLC.Link Introduces dlcBTC. Retrieved from https://www.dlc.link/blog/dlc-link-introduces-dlcbtc-a-game-changing-bitcoin-bridge-for-trustless-defi-operations

[6] DLC.Link. DLC.Link Docs Generic Use Cases. Retrieved from https://docs.dlc.link/applications/generic-use-cases