Whitepaper

00/ Abstract

The surge in Bitcoin's recognition as a store of value has sparked a substantial demand for secure custody solutions. Despite this, the trustless custody landscape has seen limited innovation since Bitcoin solidified its role. Custodial platforms remain the preference for institutional funds, while the DeFi ecosystem leans towards simplicity with wrapped Bitcoins (WBTC), secured by an 11-of-15 multisignature scheme. This paper introduces a fresh perspective, proposing an inventive solution to this challenge. By leveraging advanced Bitcoin technologies, including timelocks, taproot, and Merkelized Abstract Syntax Trees (MAST), we aim to redefine the approach to secure and efficient Bitcoin custody.

01/ Introducing BitSig

BitSig is an open-source smart vault platform designed to facilitate secure and decentralized Bitcoin custody. The core concept behind BitSig involves the implementation of degrading multisignature quorum vaults. BitSig vaults are a collection of programmable smart contracts written in Bitcoin script that govern pooled Bitcoin holdings. The governance of the Bitcoins held within these vaults is determined through voting proposals, which require approval by a predetermined quorum of signatories. Each signatory possesses a specific voting power and collaborates to reach a consensus that satisfies an unlocking threshold. This threshold progressively decreases over time, representing BitSig's innovative approach to enhancing security.

The introduction of BitSig's degrading vault mechanism creates opportunities for securing substantial amounts of Bitcoin, valued in the billions of dollars, natively on the Bitcoin network. Furthermore, BitSig offers the convenience of interaction with these secured assets using Ethereum Virtual Machine (EVM)-compatible web3 wallets, thereby bridging the gap between Bitcoin and the broader blockchain ecosystem.

02/ Introduction

BitSig stands as a non-custodial vault service provider, introducing a revolutionary approach to distributed BTC custody while retaining ownership of funds. The platform orchestrates the distribution and management of immutable bitcoin-native vault contracts on the Bitcoin network, ensuring trustless execution of vault policies. Each vault contract holds custody of pooled assets, unlocking funds based on a preset quorum and, if needed, degrading its quorum threshold over time to safeguard against potential loss if signatories lose access to their keys.

Fueling BitSig vaults is Bitcoin taproot multisig technology, accommodating up to 999 signatories per vault. Each signatory is assigned a customizable voting power in percentage format. Collaboration among signatories is key, working to meet the unlocking threshold for fund withdrawal. Signatories contribute to the unlocking process by voting on withdrawal proposals, signing off with their voting power, also referred to as unlocking power. The deployment of a BitSig vault involves a vault initiator who sets parameters such as signatory members, voting power, unlocking thresholds, and optional degrading measures.

To ensure accessibility and eliminate coordination discrepancies in pooled assets, BitSig utilizes an EVM network chosen by the vault initiator as the data availability and coordination layer for Bitcoin vaults. This EVM approach is selected for its reliability in providing redeem script backups and effectively coordinating vault events. The mirror contract serves as the coordination contract on the EVM network, making BitSig vaults accessible through web3 wallets. Notably, signatories never engage directly with the vault contract on the Bitcoin network but instead use mirror contracts to coordinate vault events. In the event of a mature withdrawal proposal with sufficient votes, signatures amassed within the mirror contract are mirrored into the witness field of the vault contract, triggering a Bitcoin transaction.

03/ Bitcoin script

Bitcoin Script, as a critical component of Bitcoin's transaction system, employs a stack-based programming language, distinguishing it significantly from Ethereum's Ethereum Virtual Machine (EVM). While the EVM is engineered for Turing completeness and the seamless integration of smart contracts, Bitcoin Script adheres to a deliberate simplicity and operates on a fundamentally distinct premise.

Comparable to the EVM, Bitcoin Script employs a stack for the storage and manipulation of values. However, unlike the EVM, Bitcoin Script exclusively relies on the stack as its sole data repository. Consequently, Bitcoin Script necessitates additional effort to store multiple values for future contract execution. Furthermore, the challenge arises when seeking to persistently store and modify values across contract executions, as this capability is notably absent. This divergence constitutes the paramount distinction between Ethereum's smart contracts, characterized by a stateful model, and Bitcoin's, characterized by a stateless model. Ethereum aligns itself with the conventional imperative paradigm of mutable data, whereas Bitcoin aligns with the functional paradigm, emphasizing immutable data.

Bitcoin transactions originate from indivisible Bitcoin units, referred to as transaction outputs. These outputs become Unspent Transaction Outputs (UTXOs) when available for use. UTXOs are secured through a locking script, which defines the conditions required for their expenditure. To spend a UTXO, an unlocking script must be provided. These scripts undergo joint execution, and the transaction is deemed valid only if their execution proceeds without errors, yielding a TRUE result.

04/ Taproot Activation

Bitcoin's scripting system imbues Unspent Transaction Outputs (UTXOs) with programmable attributes, rendering bitcoins spendable exclusively when in compliance with predetermined conditions and constraints. This programmability feature has been a foundational characteristic of Bitcoin since its inception. Nevertheless, antecedent to the Taproot upgrade, certain limitations constrained the complexity of Bitcoin's smart contracts. Most notably, these constraints encompassed a maximum of m-of-15 multisignature policies, a limit of 201 non-push opcodes, and a script size ceiling of 10,000 bytes. The initiation of the Taproot soft-fork in late 2021 marked a seminal moment in the evolution of the Bitcoin network, ushering in the unprecedented potential to construct highly efficient and intricate m-of-n spending policies.

Presently, Bitcoin employs the Elliptic Curve Digital Signature Algorithm (ECDSA) for transaction signing. Taproot introduces a novel signing method known as Schnorr signatures. Schnorr signatures not only facilitate space-efficient optimizations but also deliver faster verification, resulting in reduced resource demands for maintaining a full node. This efficiency enhancement is particularly significant should Taproot garner widespread adoption. Beyond the semantic implications of the new Schnorr signing algorithm, the Taproot upgrade liberates the Bitcoin ecosystem from several resource-related constraints.

The amalgamation of the novel signing algorithm, the removal of resource limitations, the introduction of timelocks, and the incorporation of 32-bit arithmetic operations has rendered the Bitcoin network sufficiently expressive to support advanced smart vault constructions. BitSig, an open-source smart vault platform, harnesses the capabilities afforded by the Taproot upgrade.

In tapscript, the scripting language for taproot-based bitcoin contracts, BitSig vaults undergo a shift from ECDSA to Schnorr for signature evaluation. This transition leverages the efficiency gains associated with Schnorr Signatures. Unlike the traditional one-pubkey-one-vote approach, BitSig employs a voting-power-based scheme, where each signatory pubkey signifies a unique voting power.

05/ Mirror Contract

Serving as the data availability layer for the vault contract, the mirror contract has the flexibility to be deployed on any EVM network. All user interactions are facilitated through the mirror contract, including the initiation, editing, and finalization of vaults, as well as the initiation of withdrawal requests and the voting on such requests. Primarily, the mirror contract focuses on data storage, with minimal computation limited to sanity checks during data entry deserialization. This contract's main function is to store data that the vault contract can later access. Notably, the mirror contract stores vault instances, allowing anyone to create a new instance as needed.

06/ Role of Signatures and Signatories

Signatures:

- Signatures are accumulated within the mirror contract.
- They are mirrored into the witness field of the vault contract.
- Signatures exclusively commit to transaction templates based on vaults' UTXOs.
- Signatures become invalid when attached to a bitcoin transaction with an incorrect destination address and amount.

Signatories:

- Signatories never natively interact with the vault contract on the Bitcoin network.
- They use mirror contracts to coordinate vault events, such as creating a new vault or voting for a withdrawal proposal.
- Signatories coordinate to meet the unlocking threshold for fund withdrawal.
- Each signatory votes on a withdrawal proposal by signing it off with their voting power, also known as the unlocking power.
- The party deploying a Bitlock vault is called the vault initiator, who determines signatory members, their voting power, the unlocking threshold, and optional degrading parameters when setting up the vault.