ANASTASIA LABS

Mithril-Enhanced Cross-Chain Bridge Protocol

Version 1.4 – October 2024

Contents

Mithril-Enhanced Cross-Chain Bridge Protocol

1. Abstract

Blockchain bridges are essential for enabling interoperability across decentralized networks, but current solutions face challenges such as high gas costs, replay attacks, centralization risks, and throughput limitations. This paper proposes the Mithril-Enhanced Cross-Chain Bridge Protocol, adapting Mithril's certificate-based multi-signature aggregation and Merkle Patricia Forestry (MPF) state tracking to ensure secure, scalable, and low-latency cross-chain operations.

2. Introduction

2.1. A New Paradigm of Self-Sovereign Finance

Blockchain technology has introduced a transformative model in finance and technology, promoting self-sovereignty by enabling users to transact without dependence on centralized intermediaries. Since its inception, blockchain has gained substantial traction in financial applications, empowering individuals to autonomously store and transfer value, execute transactions, and accrue interest. Paradoxically, a significant proportion of this activity has been facilitated through centralized exchanges and platforms, where authorities retain control and may impose censorship. The emergence of decentralized finance (DeFi) is reshaping this landscape by providing financial services—such as lending, exchanging, and leveraging—without centralized oversight. DeFi encompasses a diverse ecosystem of decentralized applications (dApps) designed to maintain financial operations securely and transparently within the blockchain's trustless environment.

2.2. The Challenge of Interoperability

As DeFi continues to expand, a critical challenge arises: interoperability. While scalability in terms of transaction throughput is important, the seamless interaction between disparate blockchain networks is paramount for establishing a fully connected ecosystem. Currently, blockchains predominantly operate in isolation, limiting cross-chain capabilities and constraining the potential for widespread adoption. Despite Bitcoin's market dominance, no fully decentralized, universal solution for cross-chain interactions exists. This siloed nature affects not only Bitcoin but also other major blockchains, impeding the realization of the full potential of decentralized technologies.

Interoperability is essential for the growth and maturation of blockchain ecosystems. Decentralized exchanges (DEXs) require increased liquidity; lending platforms necessitate access to a diverse array of assets; and synthetic assets and derivatives benefit from high-capital resources. Achieving true interoperability can amplify network effects across chains, reduce redundancy, and foster innovation rather than replication.

Although interoperability may not resolve every challenge faced by blockchain technology, it provides a foundational step toward addressing many of them.

2.3. Enabling Universal Interoperability through Zero-Knowledge Proofs and Mithril

In this paper, we propose a method to achieve seamless interoperability across blockchains by utilizing a zero-knowledge (ZK) bridge built upon Mithril. This approach allows users to initiate complex cross-chain transactions with a single, straightforward action, thereby enabling what we term universal interoperability. By employing zero-knowledge proofs, we ensure privacy and security in transactions, while Mithril's lightweight certificates facilitate rapid and secure cross-chain communication.

2.4. Defining Universal Interoperability

Interoperability can encompass various functionalities; however, we define universal interoperability as the capability to transfer any asset across different blockchains for use in any application, all initiated through a single user transaction. For example, a user should be able to:

- Exchange Bitcoin (BTC) for Cardano (ADA) on a Cardano decentralized exchange
- Utilize ADA as collateral on the Polkadot network to mint a stablecoin
- Return the stablecoin to the Cardano network for lending purposes

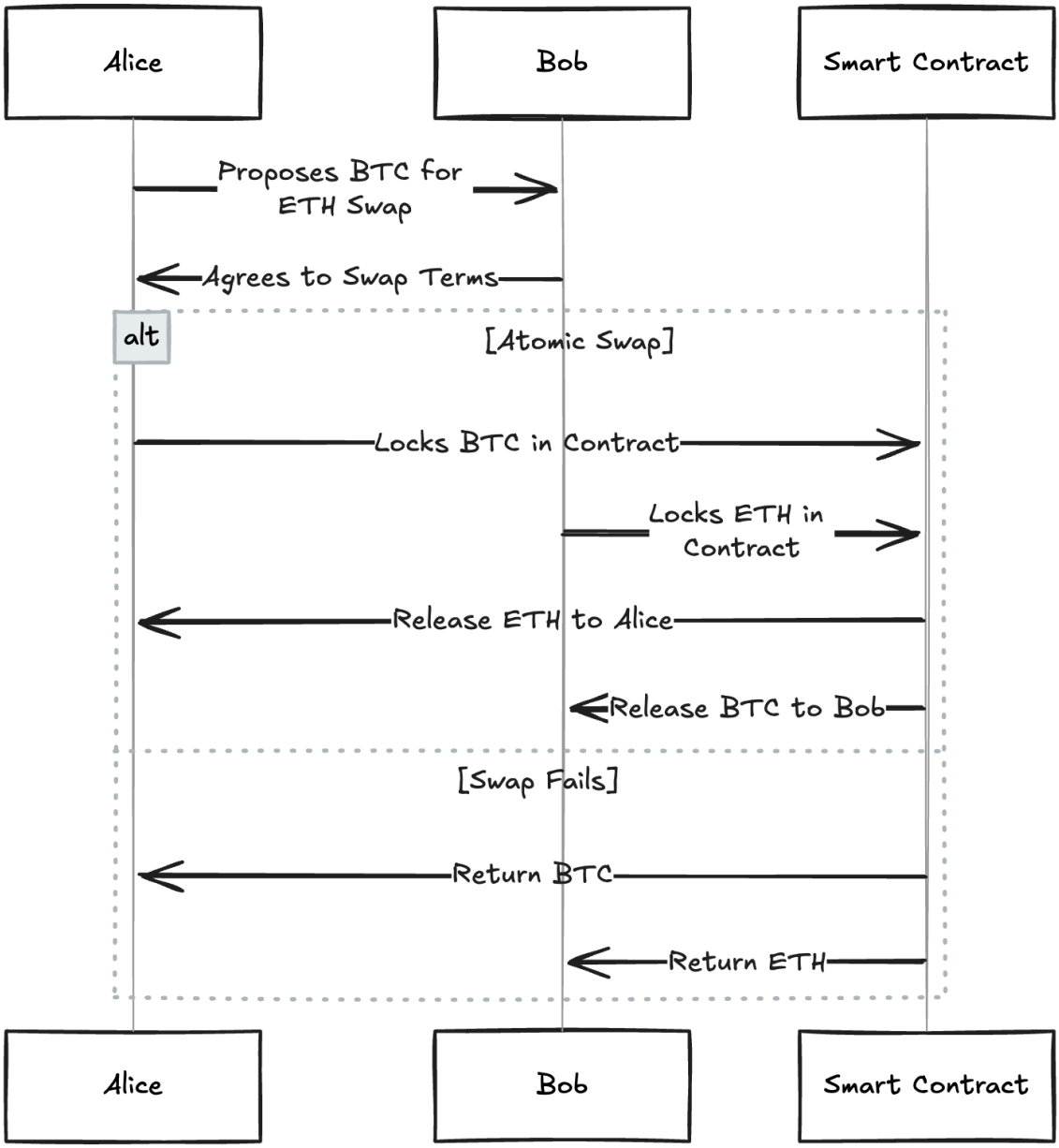
This entire process should be initiated by a single transaction from the user and executed seamlessly across multiple chains and applications. Furthermore, a universal interoperability protocol must be adaptable to future, as-yet-unimagined applications, ensuring its relevance as new use cases and technologies emerge.

3. Related work

Achieving interoperability between different blockchain networks has been a focal point of research and development in the blockchain community. Numerous solutions have been proposed, particularly aiming to bridge interactions between major platforms like Bitcoin and Ethereum. This section examines some of these existing approaches and discusses their primary limitations.

3.1. Atomic Swaps

Atomic swaps are protocols that enable two parties to exchange cryptocurrencies from different blockchains directly, without the need for a trusted intermediary. Implemented using techniques like hash time-locked contracts (HTLCs), they ensure that either both parties successfully complete the exchange or neither does. For example, if Alice wants to swap Bitcoin (BTC) with Bob's Ethereum (ETH), atomic swaps guarantee that Alice receives ETH only if Bob receives BTC, and vice versa.



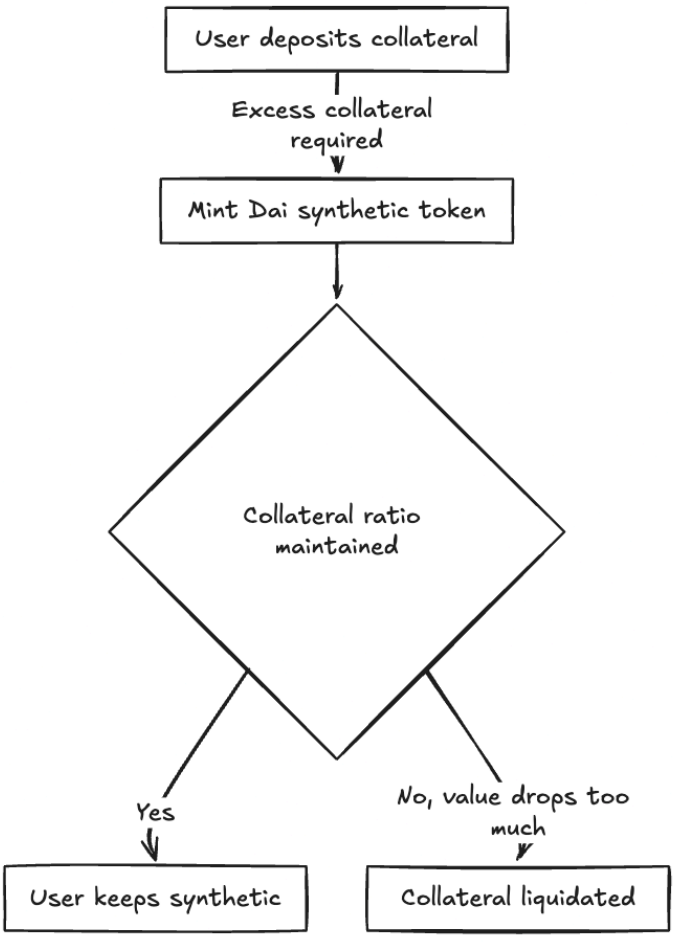
While atomic swaps offer a decentralized method for cross-chain exchanges, they face significant challenges:

- Limited Scope of Application: Atomic swaps are tailored specifically for peer-to-peer asset exchanges where both parties agree on the assets and terms in advance. This restricts their utility to simple trades and precludes more complex financial operations such as cross-chain lending, derivatives, or integration with decentralized applications (dApps). As the blockchain ecosystem evolves, there is a need for interoperability solutions that can support a wider array of use cases beyond straightforward swaps.

- Free-Option Vulnerability: The design of atomic swaps requires time constraints to ensure security across different blockchains. Participants can exploit these time windows by delaying their actions to monitor market conditions. If the market moves unfavorably, they can choose to abort the transaction without penalty. This behavior effectively grants them a risk-free option to cancel the trade if it becomes disadvantageous, undermining the fairness and reliability of the protocol. Mitigating this issue often necessitates additional mechanisms, such as collateral or reputation systems, to discourage such strategic behavior.

3.2. Synthetic Assets

Synthetic assets are financial instruments that replicate the value of other assets, allowing users to gain exposure to asset price movements without holding the actual asset. For example, a synthetic version of BTC enables users to benefit from Bitcoin's price fluctuations on platforms like Ethereum. Creating synthetic assets typically involves locking up collateral—often exceeding the value of the synthetic asset—to mitigate the risk of price volatility.



Despite their innovative nature, synthetic assets present several challenges concerning interoperability:

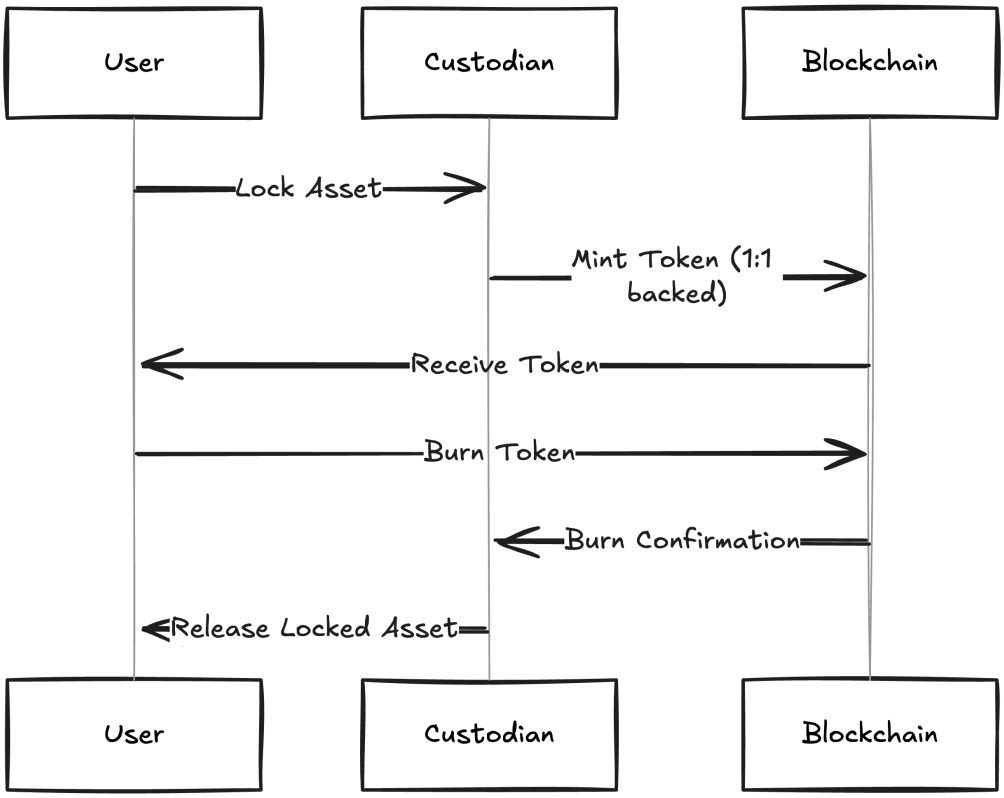
- Indirect Ownership and Redemption: Synthetic assets mirror the value of underlying assets but do not confer direct ownership. If a user wishes to convert synthetic BTC back into actual BTC, they must find a

counterparty willing to make the exchange. This dependency can introduce friction and limit the liquidity of synthetic assets.

- Isolation within Blockchains: Synthetic assets are usually confined to the blockchain where they are issued. Transferring them to another blockchain is not straightforward and requires additional interoperability layers. For instance, moving a synthetic asset from Ethereum to another network would necessitate a cross-chain protocol or bridge, adding complexity and potential security risks.
- Collateral and Liquidation Risks: The stability of synthetic assets depends on over-collateralization and liquidation mechanisms to maintain their peg to the underlying asset. During periods of extreme market volatility, these mechanisms can fail or become insufficient, leading to under-collateralization. This scenario can cause the synthetic asset to lose its intended value relationship, resulting in losses for holders and destabilizing the ecosystem.

3.3. Tokenized Asset Bridges

Tokenized asset bridges involve locking an asset on one blockchain and issuing a corresponding token on another blockchain, effectively creating a pegged representation of the asset. Examples include Wrapped Bitcoin (WBTC), imBTC, tBTC, and pBTC. Users deposit their assets with a custodian or smart contract, which then mints an equivalent amount of tokens on a different blockchain. These tokens can interact with the destination blockchain's ecosystem, and users can redeem them to retrieve the original assets.



While this method provides a practical approach to interoperability, it has notable drawbacks:

- Centralization and Trust Dependencies: Many tokenized bridges rely on centralized custodians or federations to hold and manage the locked assets. This reliance introduces trust assumptions that conflict with the decentralized principles of blockchain technology. Users must trust that custodians will securely manage the assets and act honestly, exposing them to risks such as mismanagement, fraud, or regulatory intervention.
- Complexity and User Constraints: Some protocols, like tBTC, require users to engage in complex processes involving over-collateralization and multiple transactions across different blockchains. These requirements can be cumbersome and may deter users due to the associated costs, risks, and technical barriers. Additionally, limitations such as fixed denominations restrict flexibility and usability.
- Security Assumptions and Risks: Certain solutions depend on advanced technologies like Trusted Execution Environments (TEEs) to ensure security. While TEEs offer hardware-based security features, they are not impervious to attacks. Vulnerabilities in TEEs can be exploited by adversaries, especially when large amounts of value are at stake. As the incentives for attacking these systems increase with the value they secure, relying on them can pose significant systemic risks.

3.4. Cross-Chain Asset Transfer Mechanisms

Representation tokens enable three types of cross-chain transactions:

- Lock and Mint: Sending BTC from Bitcoin to Cardano.
- Burn and Release: Sending BTC from Cardano back to Bitcoin.
- Burn and Mint: Sending BTC from Cardano to another blockchain, such as Polkadot.

3.4.1. Lock and Mint

The Lock and Mint process allows users to transfer assets from the origin blockchain (Bitcoin) to a destination blockchain (Cardano) using representation tokens.

Process Overview:

- Lock: The user locks their BTC into the custody of the bridge, ensuring it cannot be moved without consensus.
- Mint: After verification, the bridge generates a minting proof, allowing the user to mint an equivalent amount of BTC-backed tokens on Cardano.

These minted tokens maintain a one-to-one peg with the original BTC and are redeemable at any time.

Example Workflow:

1. Asset Locking: Alice initiates a Bitcoin transaction that locks 0.55 BTC into the bridge's custody.
2. Notification: Alice or the application informs the bridge about this transaction.
3. Verification: The bridge verifies the transaction details and confirms the required number of confirmations.
4. Minting Proof Generation: The bridge generates a minting proof for Alice.
5. Token Minting: Alice submits the minting proof on Cardano to mint 0.54940005 Wrapped-BTC tokens (after accounting for fees).

In this workflow, Alice only needs to perform the initial Bitcoin transaction. Subsequent steps can be automated or handled by third-party services. She can also include application-specific data in her transaction, enabling interactions with smart contracts on Cardano.

3.4.2. Burn and Release

The Burn and Release process enables users to move assets from Cardano back to Bitcoin.

Process Overview:

- Burn: The user burns their Wrapped-BTC tokens on Cardano.
- Release: After verification, the bridge releases the equivalent amount of BTC on the Bitcoin network to the specified address.

Example Workflow:

1. Token Burning: Alice burns 0.2 Wrapped-BTC tokens on Cardano, specifying her Bitcoin address for receiving the funds.
2. Event Detection: The bridge detects the burn event and confirms the necessary validations automatically.
3. Asset Release: The bridge initiates a Bitcoin transaction, transferring 0.19975 BTC (after deducting fees) to Alice's specified Bitcoin address.

Again, Alice only needs to initiate the initial burn transaction on Cardano. The bridge handles the rest, ensuring a secure and efficient transfer back to Bitcoin.

3.4.3. Burn and Mint

The Burn and Mint process allows users to transfer assets from one host blockchain (Cardano) to another (e.g., Polkadot) without interacting with the origin chain (Bitcoin).

Process Overview:

- Burn: The user burns their Wrapped-BTC tokens on Cardano.
- Mint: After verification, the bridge generates a minting proof for the destination blockchain, allowing the user to mint equivalent tokens on Polkadot.

Example Workflow:

1. Token Burning: Alice burns 0.34 Wrapped-BTC tokens on Cardano, specifying her Polkadot address where she wants to receive the tokens.
2. Event Detection: The bridge detects the burn event and completes the necessary confirmations.
3. Minting Proof Generation: The bridge generates a minting proof for Polkadot using zero-knowledge proofs and Mithril.
4. Token Minting: Alice submits the minting proof on Polkadot to mint 0.33932034 Wrapped-BTC tokens (after deducting fees).

3.5. Common bridge problems

In certain bridge designs, users are required to interact only with the source chain when performing wrapping or unwrapping requests (e.g., sending a single Cardano transaction to wrap ADA into its representation on another chain). While this simplifies the user experience compared to requiring interactions on both chains, it introduces technical challenges for both current and future implementations:

- Request Uniqueness
- Fee Structure

Without resolving the oracle problem, obtaining decentralized exchange rates between tokens is not feasible. This limitation makes it difficult to implement the proposed fee structure and user experience (UX) effectively. Transitioning to a system with a less optimal UX is undesirable.

3.5.1. Ensuring Request Uniqueness

This challenge primarily affects the Cardano contracts. Contracts on other chains, such as Cosmos, do not face this issue because marking a request as fulfilled on-chain has a constant cost per request.

Enforcing the uniqueness of requests on-chain without user interaction on the destination chain is complex, even though it is technically possible using structures like sparse Merkle trees.

In a trusted setup, validators can still misbehave due to external factors such as clock synchronization issues or data inconsistencies. If uniqueness cannot be enforced, multiple UTXOs (Unspent Transaction Outputs) referencing the same wrapping or unwrapping request on the destination chain could be created. This situation necessitates additional off-chain measures:

- Validator Coordination: Validators must coordinate to ensure only one node creates a request UTXO in a given cycle.
- Conflict Resolution: If multiple UTXOs exist for a single request, validators need to agree on which one to accept.
- Refund Mechanism: A system is required for validators to cancel and receive refunds for duplicated UTXOs.

Implementing these solutions is non-trivial and adds significant complexity to the system.

3.5.2. Fee Structure Challenges

Consider a request to wrap 100 ADA into its representation on another chain.

Wrapping Example:

- User Requirements: The user locks 100 ADA plus an additional amount X in the Locker contract. X includes:
  - Protocol Fee: A certain percentage of the 100 ADA.
  - Validators' Transaction Fee: The cost for validators to fulfill the request on the destination chain (e.g., a certain amount of ATOM on Cosmos), converted to ADA using off-chain exchange rates.
  - Minimum UTXO: X must be at least 2 ADA to cover the minimum UTXO requirement.
- Validators' Actions: Validators validate the request, mint, and send 100 wrapped ADA to the user on the destination chain.

Unwrapping Example:

- User Requirements: The user burns 100 wrapped ADA and locks an amount Y on the source chain (e.g., ATOM on Cosmos). Y includes:
  - Protocol Fee: A certain percentage of the 100 ADA.
  - Validators' Transaction Fee: The cost for validators to fulfill the request on Cardano, converted to ATOM.
  - Minimum UTXO: An additional 2 ADA (converted to ATOM) to cover the minimum UTXO when the ADA is returned.
- Validators' Actions: Validators validate the request, unlock, and send 100 ADA plus 2 ADA (for the minimum UTXO) back to the user on Cardano.

3.5.2.1. Identified Problems:

- High Minimum Fees: The inclusion of 2 ADA in all requests implies that bridging fees are always at least 2 ADA, regardless of the transaction size. This minimum fee may not be competitive for users making smaller transactions.
- Currency Exposure for Validators: Since users do not interact with the destination chain, fees must be collected in the source chain's native token (e.g., ADA or ATOM). However, validators incur transaction fees in the destination chain's native token. This exposure effectively means validators are holding the source chain's token while owing fees in the destination chain's token, leading to potential financial risks due to exchange rate fluctuations.
- Complex Earnings Calculation: Calculating validators' earnings becomes complicated without relying on stablecoins or fiat currencies, given the fluctuating exchange rates between native tokens.

3.5.2.2. Potential Solutions

To address these issues:

- Both-Chain Interaction UX: Encouraging or requiring users to interact with both the source and destination chains can distribute transaction fees more evenly and reduce the minimum fee requirements, making the bridge more attractive for users.
- Stable Fee Mechanisms: Implementing a fee structure that utilizes stablecoins or pegged assets can mitigate exchange rate risks. This method allows for predictable earnings for validators and transparent fees for users.
- Optimize UTXO Management: Exploring methods to minimize the impact of the minimum UTXO requirement, such as aggregating transactions or using specialized contracts to reduce the baseline cost for users.

4. Mithril overview

Mithril is a stake-based cryptographic protocol designed to enhance the scalability and efficiency of blockchain operations, particularly in Proof of Stake (PoS) networks like Cardano. It introduces a new primitive called stake-based threshold multisignatures (STM), also known as Mithril signatures, which allows for the aggregation of individual signatures into a single compact multisignature. This is contingent upon the collective stake supporting a given message exceeding a predefined threshold.

4.1. Stake-Based Multiparty Cryptography

- Stake Association: Participants in the network are linked to their stake, which determines their influence in the protocol.
- Adversary Model: Security is assessed against an adversary limited by the total stake it controls, rather than the number of participants.
- Scalability: Critical operations scale logarithmically with the number of participants, ensuring efficiency even as the network grows.

4.2. Stake-Based Threshold Multisignatures (STM)

- Aggregation: Individual signatures from a subset of participants are combined into a single multisignature.
- Stake Threshold: The aggregated stake must exceed a certain threshold for the multisignature to be valid.
- Random Sampling: For each message, a pseudorandom subset of participants is selected to issue individual signatures, enhancing scalability in signing, aggregation, and verification processes.

4.3. Applications

- Stakeholder Decision-Making in Proof of Work (PoW) Blockchains: Mithril can improve governance and decision-making processes in PoW networks like Bitcoin by utilizing stake-weighted signatures.
- Fast Bootstrapping of Proof of Stake (PoS) Blockchains: Mithril enables rapid synchronization of new nodes in PoS networks by providing compact and easily verifiable multisignatures, facilitating efficient bootstrapping.

4.4. Mithril Network Components

The Mithril protocol operates through three main components:

4.4.1. Mithril Aggregator

- Role: Coordinates the production of Cardano snapshot archives.
- Function: Collaborates with Mithril signers and Cardano nodes to generate certificates using Mithril multisignatures.
- Purpose: Ensures that snapshots of the blockchain can be quickly and efficiently verified.

4.4.2. Mithril Signer

- Role: Produces individual signatures.
- Function: Works alongside a Cardano node operated by a stake pool operator (SPO) holding stake in the network.
- Purpose: Contributes to the multisignature by signing messages, representing their stake proportionally.

4.4.3. Mithril Client

- Role: Verifies and restores blockchain snapshots.
- Function: Facilitates lightning-fast bootstrapping of a Cardano full node by validating multisignatures and applying the snapshot.
- Purpose: Enhances network efficiency by reducing the time and resources required for new nodes to synchronize.

4.5. Advantages of Mithril

- Scalability: Operations depend logarithmically on the number of participants, supporting large-scale networks efficiently.
- Security: The protocol is secure against adversaries controlling less stake than the required threshold.
- Efficiency: Reduces computational and bandwidth demands for nodes joining and participating in the network.

5. Mithril for the Bridge Protocol

Mithril offers several key advantages over other interoperability solutions, including threshold signatures, optimistic proofs, and centralized custodial models. Below are the primary reasons for choosing Mithril for the cross-chain bridge:

5.1. Decentralized, Stake-Weighted Governance

- Stake-based lotteries select validators dynamically, ensuring decentralization without compromising efficiency.
- Dynamic validator participation prevents centralization risks, which are common in custodial bridges and certain threshold signature schemes.

5.2. Lightweight and Scalable Multi-Signature Aggregation

- Mithril allows for compact, aggregated certificates from partial signatures, reducing the on-chain data footprint.
- This efficiency ensures that gas fees remain low even for high-frequency transactions, making the protocol ideal for real-time cross-chain operations.

5.3. Fault Tolerance and Parallel Validation

- Mithril's partial signature model enables validators to sign transactions independently, improving throughput by allowing parallel operations.
- Unlike synchronous consensus protocols, Mithril's design ensures low-latency validation without compromising security.

5.4. Security through Certificate Chains

- Certificate chaining links validated states or transactions, ensuring each step in the process is verifiable and traceable.
- This model aligns with the need for secure cross-chain bridges, preventing replay attacks and unauthorized transaction execution.

5.5. Flexibility through Customizable Messages

- Mithril's framework supports extensible message types, allowing the protocol to adapt to different ecosystems and use cases.
- This adaptability is essential for cross-chain bridges that need to interoperate between EVM-compatible and non-EVM blockchains.

In summary, Mithril's decentralization, scalability, and flexibility make it an ideal cryptographic framework for building a secure and cost-efficient cross-chain bridge.

6. Utilizing Mithril and Zero-Knowledge Proofs in Blockchain Bridging

Integrating Mithril and Zero-Knowledge (ZK) proofs into a blockchain bridge enhances security, scalability, and efficiency. Mithril provides cryptographic certificates that confirm the inclusion of transactions in the Cardano blockchain. By combining Mithril with ZK proofs, the bridge can securely and verifiably transfer assets between Cardano and other blockchains, such as Ethereum or Polkadot.

6.1. Bridging Process with Mithril and ZK Proofs

6.1.1. Transaction Initiation on Source Chain (Cardano)

- User Action: A user initiates a transaction on the Cardano network to transfer assets to another blockchain.
- Transaction Details: The transaction includes metadata such as: amount, source address, destination address, asset type.

6.1.2. Mithril Certificate Generation

- Bridge Observation: The bridge node monitors the Cardano blockchain for relevant transactions.
- Certificate Request: The bridge queries the Mithril aggregator to obtain a certificate—a cryptographic proof that confirms the transaction's inclusion in the Cardano blockchain.

- Mithril Functionality:
  - Provides a chain of certificates for transactions.
  - Ensures that a set of transactions is included in the global Cardano transaction set.

6.1.3. Metadata Retrieval

- Data Fetching: The bridge queries a Cardano node or blockchain explorer API to retrieve the transaction's metadata.
- Metadata Includes:
  - Amount transferred
  - Source address
  - Destination address
  - Asset type

6.1.4. Verification Process

6.1.4.1. Proof Validation

- Certificate Verification: The bridge validates the Mithril certificate to ensure that the transaction is part of the Cardano blockchain.
- Stake Assurance: Mithril's stake-based signatures confirm that the transaction has been endorsed by sufficient stake holders.

6.1.4.2. Metadata Validation

- Consistency Check: The fetched metadata is compared against the expected transaction details.
- Integrity Assurance: Ensures that the transaction meets the bridge's criteria (correct amount, valid addresses).

6.1.5. Zero-Knowledge Proof Computation

- ZK Framework Utilization: The bridge uses a Zero-Knowledge proof framework to generate a succinct proof.
- Proof Components:
  - Verification of the Mithril Certificate: Confirms transaction inclusion without revealing sensitive details.
  - Verification of Transaction Metadata: Validates the transaction's specifics privately.
- Purpose: Allows on-chain verification of the transaction's validity on the destination chain with the submitting proof.

6.1.6. Execution on Destination Chain

- Proof Submission: The bridge submits the ZK proof to the destination blockchain.
- On-Chain Verification:
  - Smart contracts on the destination chain verify the ZK proof.
  - Confirms both the transaction's inclusion in Cardano and the validity of its metadata.
  - Verify the bridge transaction uniqueness
- Finalization:
  - Upon successful verification, the bridge finalizes the transfer.
  - Actions may include minting equivalent tokens on the destination chain representing the transferred assets.

6.2. Responsibilities of the Bridge Node

6.2.1. Observing Bridge Transactions

- Monitoring: Continuously watches the Cardano blockchain for transactions initiating a bridge transfer.
- Filtering: Identifies and processes only those transactions relevant to the bridge.

6.2.2. Querying the Mithril Aggregator

- Certificate Acquisition: Requests certificates for observed transactions.
- Timeliness: Ensures certificates are obtained promptly to avoid delays.

6.2.3. Zero-Knowledge Proof Generation

- Certificate Verification: Confirms the validity of the Mithril certificate off-chain.
- Metadata Verification: Validates transaction details against bridge requirements.
- Proof Production: Generates a ZK proof encapsulating both verifications.

6.2.4. Submitting Fulfillment Transaction

- Transaction Creation: Constructs a transaction for the destination chain that includes the ZK proof.
- Submission: Sends the transaction to the destination blockchain network.
- Smart Contract Interaction: Triggers the verification and asset transfer logic on the destination chain.