CISC 322/326

Assignment 3: Report

Enhancement Proposal for Bitcoin Core

Wednesday, April 12, 2023

Group: All Around Average

Jonathan Sumabat *20js30@queensu.ca*

Lukas Boelling *lukas.boelling@queensu.ca*

Nour Mahmoud *17nhm1@queensu.ca*

Caleb Chiu *20cyjc@queensu.ca*

David Kropinski *19dsk3@queensu.ca*

Jeff Jiachuan Li *19jjl6@queensu.ca*

Table of Contents

Abstract

This paper explores the possible use of Bitcoin Core using zero-knowledge proofs in its validation of transactions. This improvement would allow greater overall security in the Bitcoin network by allowing the encryption of transactions. Two possible implementations are presented: interactive zero-knowledge proofs and non-Interactive Zero-Knowledge Proof with rollup. SAAM analysis was on both implementations to assess the potential impacts of the change. Two use cases are used to show how the new proofs would work in the system.

Introduction and Overview

A clear understanding of Bitcoin Core's architecture and functionality has been achieved through the past two reports. This upcoming report will be focused on discussing and explaining a new feature that has yet to be fully implemented by Bitcoin Core. Our group decided to propose the use of zero proof knowledge which is a protocol that improves privacy and security of transactions as well as increasing efficiency by reducing the amount of data required for processes in various applications. The report will be divided into four main sections.

In the first section, we initiate the design process by establishing two potential implementations to enact the enhancement. These two implementations are Interactive Zero-Knowledge Proofs and Non-Interactive Zero-Knowledge proofs. We further break down the implementation and discuss its functionality and the in's and out's in greater detail. The second section of the report is highlighted on the SAAM analysis on the implementations. In this phase, we will discuss the advantages and disadvantages of each implementation and identification of the stakeholders. The third section of the report covers the use cases and diagrams to help aid in the breakdown and analysis of the protocol. Finally, the last section of the report covers test cases, ensuring that our implementation is sound within Bitcoin Core's software. Following after will be a brief discussion on potential risks, lessons learned, and limitations.

Proposed Enhancement

As a group we realized that zero-knowledge proofs had not been implemented by Bitcoin Core yet even though it is prevalent in the cryptocurrency space. Zero-Knowledge proofs (ZKPs) are a widely used cryptographic technology that many other blockchains have implemented such as Tornado cash, which is a decentralized service that allows users to conduct private transactions on Ethereum. The purpose of this cryptographic protocol is to allow one party (the prover) to prove to another party (the verifier) that they are aware of a certain piece of information without disclosing any further details about that knowledge. To put it simply, ZKPs allow for the verification of a claim without revealing anything more but the fact that the claim is accurate. An example of this can be shown in authentication systems. Take into account a user who wishes to access a system without disclosing their password to the server. The user can demonstrate to the server that they are aware of the password without actually sending it by using a zero-knowledge proof. This is done by creating a unique token based on the password using a mathematical function, which can then be submitted to the server as proof of authentication. ZKPs are beneficial because they offer a mechanism to confirm information without really disclosing it. Where privacy and security are important, such as in financial transactions, medical data, and voting systems, this can be especially helpful. Individuals can do transactions or establish their identification by using ZKPs without disclosing sensitive information. This is especially useful in the cryptocurrency world and beneficial for Bitcoin Core since it would allow greater security and privacy for transactions, in addition to establishing secure communication and transactions in a variety of settings.

Implementation

Zero knowledge proofs allow transactions on the network to remain encrypted, yet still be verifiable.

Implementation 1 - Interactive Zero-Knowledge Proof:
Verifying validity requires back-and-forth communication between prover node and verifier node
Major drawbacks: Requires both parties to be online; resulting proof is only valid for the verifier node (other verifiers are not able to trust it and must construct their own).

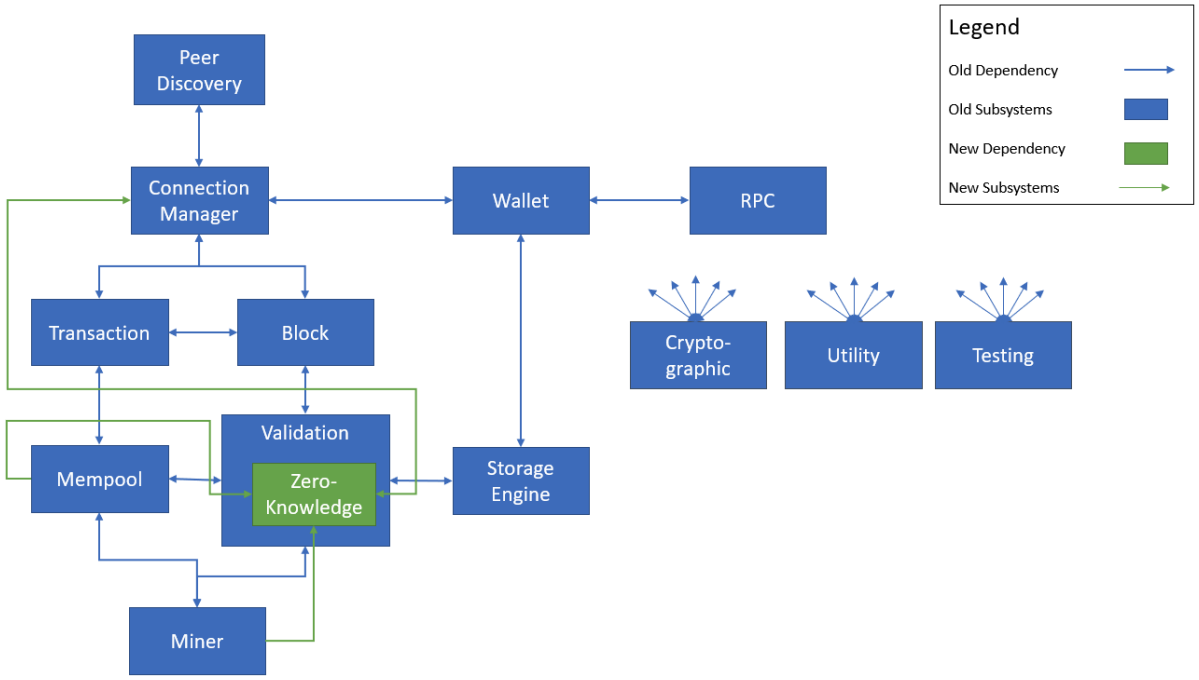

Figure 1: Proposed Conceptual Architecture for Implementation 1

Miner and Mempool both depend on the Zero-Knowledge subsystem for transaction validation. Due to the interactive nature of this protocol, a two-way dependency with the Connection Manager is required to allow the verifier to supply challenges to the prover until the verifier is satisfied.

Implementation 2 - Non-Interactive Zero-Knowledge Proof with Rollup:

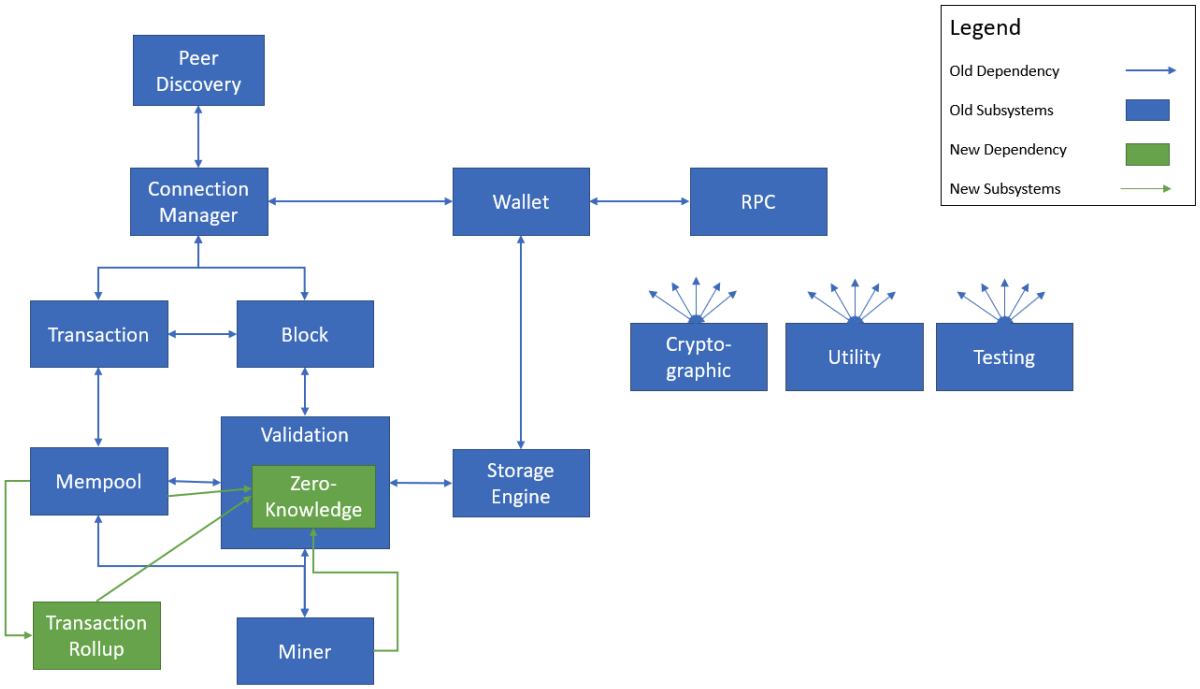

Figure 2: Proposed Conceptual Architecture for Implementation 2

Non-interactive zero-knowledge proofs work roughly as follows: Prover and verifiers have a shared public key, established through some trustworthy method. The prover has some private knowledge and constructs a proof of this. Later on any verifier who has the shared public key can use the proof to check that the prover does indeed have the knowledge they claim to have.

Transaction rollup is a method of improving blockchain throughput by aggregating a series of transactions into a single transaction. A non-interactive zero-knowledge proof is created to prove the validity of the aggregated transaction. The aggregated transaction is added to the blockchain along with its proof of validity.

Miner and Mempool both depend on the Zero-Knowledge subsystem for transaction validation. The Mempool depends on the Transaction Rollup subsystem to submit transactions to be aggregated. The Transaction Rollup subsystem depends on the Zero-Knowledge subsystem to construct zero-knowledge proofs of validity.

Impacted Directories and Files

In the validation engine the main directories that would be impacted are the validation and script directories. The validation directory deals directly with the verification of transactions and as this is a new verification method it would go here. The script directory deals with handling transactions, this would be impacted as transactions would contain different/less information in the zero-knowledge proof method.

The mempool subsystem would have its main files txmempool.h/cpp impacted in these implementations. This file uses many of the methods in the validation directory which as discussed before would be altered. Additionally with the non-interactive implementation the txmempool would also need to have interactions with the added transaction rollup subsystem.

The miner subsystem would have its miner.h/cpp files impacted for similar reasons to the previous mempool files. It needs the new validation methods and needs methods for interacting with the new rollup subsystem.

The connection manager subsystem would also be impacted as it would have to deal with the new zero-knowledge transactions and interactive messages in the case of the first implementation. The main files impacted by this would be txrequest.h/cpp and net_processing.h/cpp which helps with getting transactions and related information, the net.h/cpp files as it is the main file for dealing with the network, and the txreconciliation.h/cpp files as they deal with how a node sends information.

SAAM Analysis

Stakeholders

There are three major groups of stakeholders that the Zero-Knowledge Proof enhancement would affect: users, developers, and miners. Users include anyone who uses the Bitcoin network to send, receive, or store bitcoin. Developers are the individuals who help maintain and develop the open-source project. Miners are responsible for validating transactions and adding blocks to the blockchain, helping maintain the integrity of the Bitcoin network.

Users

The most important NFRs for users regarding the implementation of Zero-Knowledge Proofs are: security, privacy, usability, and performance. No matter which of the above approaches is used to implement Zero-Knowledge Proofs, security, privacy, and usability should function very similarly. Security must protect against any data from being forged or tampered with, ensuring that user

transactions occur safely. Privacy of user data must be well maintained, making sure no unnecessary information about a transaction or sensitive user data can be leaked. The implemented Zero-Knowledge Proofs system should also be user-friendly as it can be quite a complex and confusing topic for an average user. Providing good and clear documentation, instructions, and user interface will be crucial to the success of the new feature. As for performance, the computation and communication overhead should be efficient enough to handle thousands, if not millions, of users simultaneously. For Interactive Zero-Knowledge Proofs, the prover and verifier nodes must be able to send messages back and forth quickly and efficiently to collectively authenticate the given data. Due to this, it is important to have a compact efficient communication method along with an optimized algorithm to decipher these messages so the user can use its functionality without major delay. With Non-Interactive Zero-Knowledge Proofs, the prover only needs to send a single message to the verifier meaning less back and forth communication is required to complete a verification.

Developers

The most important NFRs for developers are the maintainability and scalability of the system. No matter which of the above implementations are used, maintainability is highly important. The enhancements should be well implemented, making sure the best practices are being followed and the appropriate tests being conducted on the code. In terms of scalability, as the Bitcoin network grows as more and more people adopt it, the system should be able to handle the increased transaction volume. For Non-Interactive Zero-Knowledge Proofs, only one message is needed to be sent between prover and verifier, meaning it is simpler and can allow for a larger number of verifications to be done concurrently. With Interactive Zero-Knowledge Proofs, many messages must be sent back and forth between prover and verifier, meaning there can potentially be immense clutter when many verifications are being done concurrently.

Miners

The most important NFRs for miners are performance and privacy of the system. In both implementations performance and privacy are highly important. For performance, miners must be able to quickly validate transactions to be able to successfully mine a block and add it to the blockchain for a reward. For privacy, miners' mining addresses and transaction details should remain protected from unauthorized access. This can be done using encryption and following good practices for data privacy to prevent any potential breaches.

Implementation Choice

Upon analyzing the different ways our two approaches impacted the different NFRs of the stakeholders, we decided the best option was idea two, Non-Interactive Zero-Knowledge Proofs. The

first option, Interactive Zero-Knowledge Proofs, seemed inefficient as many messages must be passed between prover and verifier nodes. This can create scalability problems for developers as Bitcoin is a growing technology that will require an extremely optimized communication method and algorithm to be created between prover and verifier. This does however, in theory create better security as more information by the prover must be revealed to the verifier to check whether the prover is legitimate. The whole concept of Zero-Knowledge Proofs however is already very secure, so even if Non-Interactive Zero-Knowledge Proofs are used security and privacy will be very good. Furthermore, Non-Interactive should also present better scalability and performance with less messages needed to be exchanged when performing a verification. This means that if Bitcoin ever becomes one of the main currencies in the world, with many transactions occurring simultaneously, users should be able to use it without experiencing significant delays.

Use Cases

Use Case 1: Interactive zero-knowledge proof (ZKP)

The graph illustrates the interactions among different components involved in processing a Bitcoin transaction. The process starts with the sender initiating a transaction, which is forwarded to the Transaction module. The transaction is then sent for validation to the validation engine, which initiates the Zero Knowledge Proof (ZKP) protocol in coordination with the connection manager to establish a connection between the verifier and prover. If the transaction is determined to be valid by the network, it is added to the Mempool, which is a pool of unconfirmed transactions awaiting inclusion in a block. The miner selects transactions from the Mempool to mine and adds the chosen transaction to a block using the Proof of Work (PoW) algorithm to solve a mathematical puzzle. Once the block is successfully mined, it is added to the Blockchain, completing the transaction processing use case.

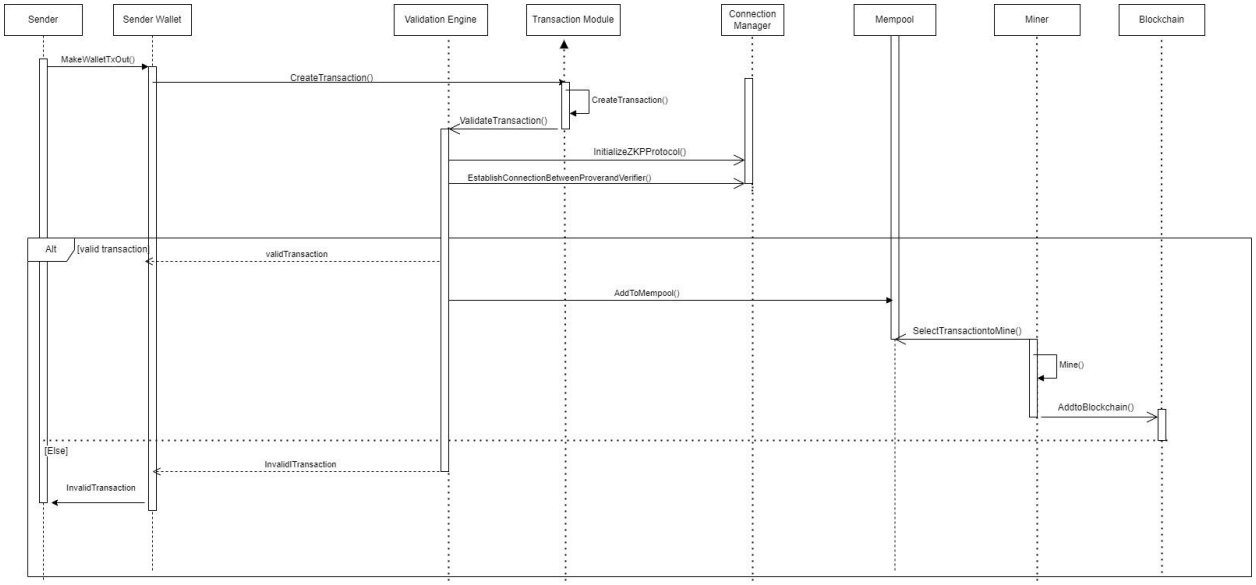

Figure 3: Use case of transaction validation for implementation 1

Use Case 2: Non-interactive zero-knowledge proof (ZKP)

The graph presents an overview of the components involved in processing a Bitcoin transaction, specifically in the context of a non-interactive zero-knowledge proof (ZKP) that is utilized to verify the validity of the aggregated transaction. The process begins with the sender initiating a transaction, which is then directed to the Transaction module. Subsequently, the transaction is added to the Mempool, which in turn submits the transactions to be aggregated to the Transaction Rollup subsystem. The Transaction Rollup subsystem initiates the ZKP protocol with the validation engine to verify the validity of the aggregated transaction. If the transaction is confirmed as valid, it is forwarded to the miner, who employs the Proof of Work (PoW) algorithm to solve a mathematical puzzle and mine a block. Once the block is successfully mined, it is appended to the Blockchain, thereby completing the transaction processing use case.

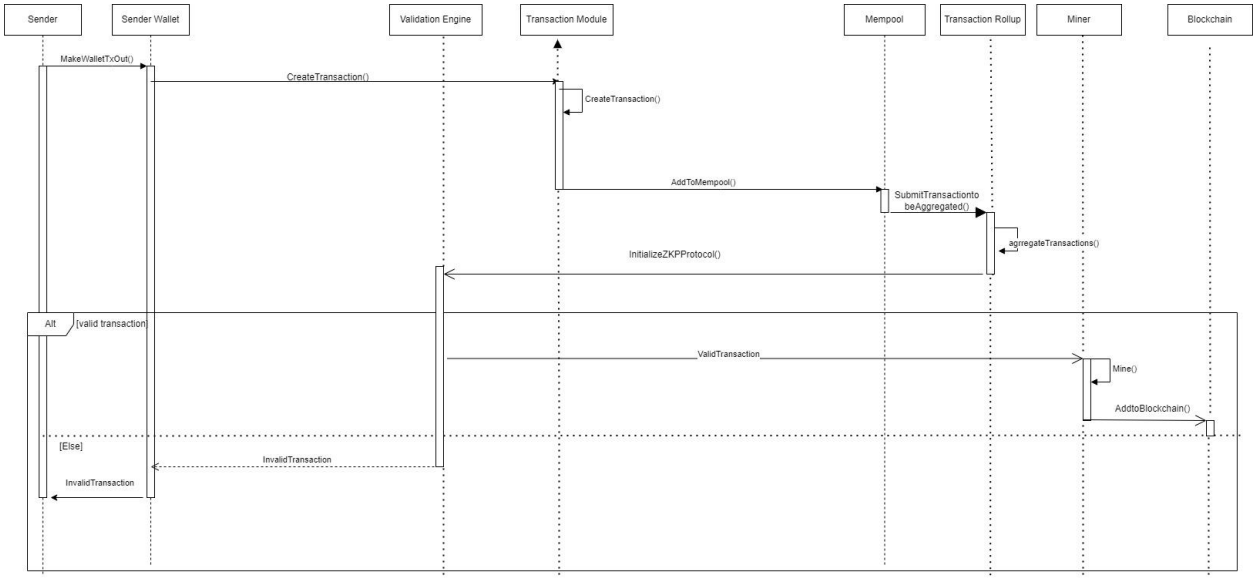

Figure 4: Use case of transaction validation for implementation 2

Testing

To test the system, it is critical to test the impact of the interactions between the suggested enhancement (Zero-Knowledge Proofs) with the pre-existing functionalities in the Bitcoin Core system. A combination of unit tests done during the development of the new feature, along with integration tests during the implementation of the feature into the Bitcoin Core system. This will help to ensure that the Zero-Knowledge Proofs enhancement behaves as intended, minimizing bugs and assuring a quality product.

Unit testing involves writing test cases for an individual component of a system to verify its functionality prior to integrating it into the system. For the Zero-Knowledge Proofs enhancement, its data transfer functionality and the selected data should be tested to ensure the transfer is securely done with the correct data. Many scenarios should be tested to make sure it works under all circumstances. By first testing the individual module before integrating into the Bitcoin Core system, errors within the module can be more easily identified and fixed, while also making sure it fulfills its functionality and NFRs.

Integration testing typically comes after unit testing. Once the individual component is deemed to be working, it must be validated on whether it works in the broader system with the other components. For the proposed enhancement, integration testing would entail reviewing the interactions between the Zero-Knowledge Proofs module with the already-existing modules in the Bitcoin Core system. Due to the previous Bitcoin Core system already having a validation engine, the

integration of the Zero-Knowledge Proofs module should have little effect on the other modules, and should not significantly change any pre-existing functionalities.

Potential Risks

Even though Zero-Knowledge proofs may be a valid solution for improvements in security and privacy on Bitcoin core's network, there are still some potential risks to consider. With the implementation of ZKP's, its important to remember that they rely on complex cryptographic algorithms which naturally increases risks of implementation errors and vulnerabilities. What this would likely require is significant changes to the current code base adding additional complexity for developers and users to understand. This could lead to errors or mistakes during implementation that may result in unintended consequences. Additionally, poor design choices provide attackers the opportunity to manipulate the proof or extract sensitive information.

With this in mind, another potential risk or complication would be managing computational resources which may impact the performance of the network as ZKP's require a large amount of computing power. A massive negative effect with implementing ZKPs is the potential for heavy computational overhead, considering that they require a lot of processing power and time to generate and verify proofs. With a computational overhead, scalability issues will follow as bandwidth and memory usage will significantly increase.

Concurrency

The interactive zero-knowledge proof implementations would introduce a new two way dependency between the connection manager module and the validation module. This adds a new concurrency between the two modules as they would pass new challenges between each other.

The non-interactive implementation would also add some new dependencies as it adds a transaction rollup module. This module would have concurrent tasks with the mempool as it would bundle transactions which the mempool would then have to use.

In both implementations the concurrency would be improved as synching with the network would be easier. Zero-knowledge proofs allows each node to download less data from the blockchain, but still be synchronized with the current blockchain state. As most modules and tasks in the Bitcoin Core system rely on the blockchain this could improve the overall system's speed.

Lessons Learned

Within completing this project, we were able to analyze and understand all major components of Bitcoin Core's system from a more objective perspective, in addition to becoming more familiar with the field of cryptography. As a result, we offered out insights into implementing a new protocol that would greatly improve the security and privacy of transactions throughout the entire system. In addition, when using SAAM, we also gained a deeper understanding of this method of architectural analysis, and its benefits in assessing the risks inherent in the architecture and early development phase.

Limitations

The major limitation with the addition of zero-knowledge proofs to the Bitcoin Core system is the increased need for computational power. The new validation method requires more computational power which for some devices could pose a problem. Smaller devices such as mobile phones may not be able to validate transactions on their own and would require the assistance of either a public resource which would require another thing to implement or a third-party which takes away from the decentralized nature of bitcoin.

Conclusions

This report proposes two different implementations of Zero-Knowledge proofs that would be used to enhance security and privacy of bitcoin transactions. With interactive Zero-Knowledge proofs both parties will be online as validity verification is dependent on back and forth communication between the prover and verifier node. The second approach is to use non-interactive Zero-Knowledge proofs where provers and verifiers have a shared public key. The prover would have some private knowledge and constructs a proof as the verifier who also has access to the shared public key may use the proof to check the prover. Implementation of these two cases would be dependent of the specific application and security requirements of the system. When large quantities of proofs need verification, Interactive Zero-Knowledge proofs may be impractical in this circumstance due to its call for back and forth communication between prover and verifier. On the other hand, non interactive Zero-Knowledge proofs call for single communication which may make them more appropriate where efficiency is prioritized. Implementation of this protocol will enhance the privacy of bitcoin transactions and security through proof of validity for a transaction without the need to reveal sensitive information, such as private keys associated with their bitcoin addresses. After proposing these two enhancements, we use SAAM to assess the risks inherent in the architecture.

References:

GeeksforGeeks. (2023, March 27). *Interactive zero knowledge proof*. GeeksforGeeks. Retrieved April 12, 2023, from https://www.geeksforgeeks.org/interactive-zero-knowledge-proof/

Jenkinson, G. (2023, March 28). *Zero-knowledge proofs coming to Bitcoin, overhauling network state validation*. Cointelegraph. Retrieved April 12, 2023, from https://cointelegraph.com/news/zero-knowledge-proofs-coming-to-bitcoin-overhauling-network-state-validation

Wikimedia Foundation. (2023, March 26). *Non-interactive zero-knowledge proof*. Wikipedia. Retrieved April 12, 2023, from https://en.wikipedia.org/wiki/Non-interactive_zero-knowledge_proof#:~:text=Unlike%20interactive%20zero%2Dknowledge%20proofs,large%20number%20of%20statements%20simultaneously.