

On the Security of Wallet Nodes in the Cardano Blockchain

ALEX SIERKOV

1 INTRODUCTION

The performance characteristics of personal wallet nodes are often not analyzed when their security is discussed. This note argues that they can be as important for user security as the traditionally analyzed resistances against various attacks. Building on this notion, this note proposes a more balanced security approach for personal wallet nodes, targeting primarily the Cardano blockchain and its only full-data wallet, Daedalus [2]. This approach considers both the resistances against known attacks and the synchronization performance, providing a more balanced and thus more optimal solution for personal wallet nodes.

1.1 Value of time

The value of assets stored in personal wallets can only be realized by transacting with other parties. For this reason, the ability to transact reliably and at the exact time when a need or an opportunity arises can be as important as the assets' safety.

Cryptocurrency markets are very volatile, and the value of assets can change by double-digit percentages in mere hours. This volatility can be even more significant during extraordinary events, such as major exchange faults¹ or the introduction of unfavorable regulations.² Owners who synchronize their wallets only sporadically can lose a significant portion of their portfolio's value due to the need to wait several hours for synchronization. The frequency of such events is no less than the frequency of certain attack types, such as long-range attacks, against which security is normally analyzed.

Furthermore, the speed with which a wallet is able to synchronize with the blockchain directly influences the speed at which some attacks are detected, thus directly influencing the security of a wallet node.

2 VALIDATIONS PERFORMED BY CARDANO NODE

Cardano Node [3] is the software operating all block-producing nodes in the Cardano blockchain as of 2024. It contains the status-quo implementation of Cardano specifications and is used by the Daedalus wallet, the only full-data wallet in Cardano as of 2024.

The blockchain validations performed by Cardano Node can be split into two parts:

- Block-level validations, such as the verification of block header and block body hashes, the coherency of previous block hashes, the monotonicity of slot numbers and their origination in the present time, and the correctness of block signatures, and block-leadership eligibility. Since these validations are necessary to establish consensus, they will be called consensus validations moving forward.
- Transaction-level validations, such as the correctness of various certificates³ and the correctness of transaction witnesses and invariants.⁴

¹The Mt. Gox security breach in 2011.

²The Chinese Central Bank prohibiting Bitcoin transactions in 2013.

³Pool registration, stake delegation, etc.

⁴That transaction outputs are used only once, that the sum of all inputs equals the sum of outputs minus the transaction fee, etc.

The following subsections argue that in most practical cases involving personal wallet nodes transaction validation is redundant because consensus validation also proves the transactions to be valid.

2.1 Presence of an honest majority

The fundamental property of many consensus algorithms, including Cardano's Ouroboros Praos [8] and Ouroboros Genesis [6], is their reliance on the presence of an honest majority. The loss of this property has catastrophic consequences on the security of any account holder and is considered the accepted risk of using blockchain technology. This is the status quo for the past, the present, and the foreseeable future.

Given that the risk of 51% attacks is unavoidable, a question arises if there is a way to leverage this property positively to offset its burden: Once a given block is confirmed⁵ by a large enough relative stake, one can prove that at least one honest node⁶ must be present in the set of signers; thus, the validity of all transaction witnesses and transaction-level certificates becomes a mathematical certainty. This means that for wallet nodes, nodes that do not include new transactions into blocks but simply follow the chain, the revalidation of transaction witnesses is a redundant effort with the exception of the tail of the chain, which will be later discussed in this note.

2.2 Dominance of consensus over transaction validation

To further highlight the dominance of consensus validation over transaction-level validation, it is important to note that many contemporary attacks against individual wallet operators, such as cryptocurrency exchanges, rely on hidden future chain reorganizations rather than on fake cryptographic certificates. For this reason, having perfectly valid transaction data does not guarantee safety. On the other hand, consensus validation, with a confirming relative stake significant enough to prove processing by at least one honest node paired with chain density metrics, which are further discussed below, come as close as possible to providing a guarantee in blockchain technology.

3 PROPOSED METHOD

The proposed method relies on parallelization techniques to accelerate blockchain synchronization presented in [12] and on the recently described properties of the presence of an honest majority and the dominance of consensus over transaction validation to create a method that has both an outstanding synchronization performance, which is an order of magnitude better than that of Cardano Node,⁷ and its security against typical attacks not worse than that of Daedalus.

This performance advantage allows for much better availability to perform transactions for users who synchronize their wallets only sporadically. Since reduced availability means reduced security and since the resistance against typical attacks is not worse, this method results in improved practical security overall.

3.1 The core and the tail

A consequence of relying on relative-signed-stake behind a block to prove the processing by an honest node is that there is a boundary separating blocks for which such proof is possible and the ones for which it is not. Moving forward, these two parts of the blockchain will be called the core and the tail, respectively.

⁵by being referenced from subsequent blocks

⁶which follows full protocol specifications

⁷These measurements are presented in the same Parallelized Ouroboros Praos [12] paper.

It is not always possible for users to wait until the information that they would like to rely on gets into the core part of the chain. In such situations, a number of additional actions can be performed to increase the confidence in the information in the tail:

- Dynamically query a stake-weighted sample of active stake pools, confirming the presence of a given block.
- Compute chain density metrics on the tail.
- Validate tail transactions that are in the transaction dependency graph of a given wallet.

All of the above activities can be performed interactively since they take time negligible in comparison to the time necessary to synchronize and validate the core part of the chain. Having collected the above information, a wallet node can help users make much more informed decisions about the safety of relying on certain transaction-level information in the tail.

It is important to emphasize that while validating transaction-related certificates and invariants in the tail is good for protection against naive attacks, such as the ones relying on fake cryptographic credentials, it is not sufficient to guarantee security due to the presence of chain-reorganizing attacks. Thus, transaction validation in the tail is only a minor factor of the overall security.

3.2 Dominance of optimistic parallel pre-processing

The proposed method relies on the optimistic parallel preprocessing of blockchain data to accelerate the validation. At first glance, this may seem like an additional opportunity for a resource-exhaustion attack. However, in practice, what matters is the speed at which the method confirms the validity of blocks as measured by the time to detection of invalid data, which, due to the work done by the parallel preprocessing, happens much faster. On the other hand, Daedalus not only wastes available hardware resources without leveraging them in optimistic scenarios, which occur much more frequently, but also validates the blockchain slower, thus giving potential attackers bigger leverage from supplying fake data.

3.3 Chain density metrics

Two groups of metrics can help evaluate the state of the chain and detect the likelihood of an ongoing attack:

- Minimum density—helps determine whether there may be an ongoing attack that started a while ago and can result in a major chain reorganization in the future;
- Tail density—helps to evaluate if there is an ongoing attack against the wallet node, such as an Eclipse attack, during which an attacker can filter valid blocks and provide blocks not known to the rest of the network.

The active use of these metrics, along with the dynamic collection of additional information about the security of the tail, helps provide users with an extra level of confidence when they decide to act based on information in the tail.

3.4 Prevention of personal information leakage

To reduce the risk of personal attacks against a given high-net-worth wallet node, it is valuable to reduce the sharing of information about its worth with the rest of the world. However, the transaction exploration user interface of Daedalus currently redirects users to an external website, Cardano Explorer [1]. Such requests can potentially share sensitive information, allowing one to guess the relative worth of a wallet node by the requested transaction ids and their metadata and connect it with the node's IP address, simplifying the profiling of targets for future attacks.

For this reason, a personal wallet that provides a local blockchain explorer and, thus, does not send unnecessary requests to third-party websites improves security. The proposed method offers such an implementation based on parallel indexing and history reconstruction techniques presented in [10].

4 RESISTANCE AGAINST COMMON ATTACK TYPES

Since the consensus validation for the core of the chain is equivalent to all validations performed by Cardano Node, the differences in attack responses can only be present for attacks focused on manipulating the most recent transactions. Typical attacks [9] grouped by their method of influence are analyzed below in the context of tail security: dynamic querying, tail density metrics, and tail transaction validation.

4.1 Whole network attacks

In whole network attacks, such as Sybil or long-range attacks, attackers possess many valid cryptographic identities. Therefore, the validation of transactions will not help; however, dynamic querying in the case in which the full list of stake pools is still available on all branches of the chain may identify drastic differences between branches. In addition, chain density metrics should be able to indicate the presence of an attack too.

4.2 Chain-reorganizing and networking-based attacks

In networking-based attacks, such as Eclipse, DDoS or network partitioning, an attacker can take partial and even full control of all incoming and outgoing network messages of a wallet node. The attacker often uses these capabilities in a way that will lead to a reorganization of the chain history as the attacked node sees it once the attack is over, such as in Race attack. In particular, the attacker can omit valid blockchain blocks and add new blocks that are not shared with the rest of the network. Furthermore, the attacker is likely to possess valid cryptographic credentials for at least some nodes of the network.

In such situations, validation of transactions of the tail is ineffective, but chain density metrics may indicate problems. However, an attacker could manipulate only one latest block, and, in that case, remain undetectable for the density metrics. Dynamic querying, on the other hand, may reveal a high level of inaccessibility of other nodes and, therefore, indicate network partitioning. However, an attacker may try to emulate these queried hosts since the Cardano network protocol [7] currently does not require network protocol responses to be cryptographically signed. Clearly, signing protocol responses with a key that can be easily tied to a given network node would be a welcome change to the protocol and improve the effectiveness of dynamic querying for attack detection.

4.3 Wallet attacks

The proposed method only changes the mechanisms for blockchain data delivery and wallet transaction history reconstruction. However, the mechanisms for private key generation and transaction submission remain the same. For this reason, there is no difference in how this group of attacks, such as Phishing or Dictionary attacks, affects this method compared to Daedalus.

5 DISCUSSION

5.1 Ouroboros Genesis

Cardano currently expects a transition from Ouroboros Praos [6] to Ouroboros Genesis [5] as its consensus algorithm. The only difference between Ouroboros Praos and Ouroboros Genesis is the

chain selection rule. Praos uses the maximum chain length criterium while Genesis relies on maximum local chain density, which is better suited for working in the fully peer-to-peer environment and can better protect against long-range attacks.

Since block signatures, leadership eligibility validation, and other pieces of consensus validation remain the same, all parallelization techniques presented in [12] are unmodified and retain their performance. The only thing that needs to change is the choice of which chain branch to validate first, which is a trivial implementation detail.

5.2 Peer-to-peer networking

At the same time, with the transition to Ouroboros Genesis, Cardano plans to transition to a fully peer-to-peer model. For the proposed method, this means querying a random set of peers to synchronize with their view of the chain in a way similar to the dynamic querying presented for improving tail security. This change is fully compatible with the method since it can reuse almost the same software components and simply waits for its prioritization, which is planned to take place around the time of the upgrade.

5.3 Compressing proxies

The Cardano network protocol currently does not support the compressed transfer of blockchain data, leading to its nodes using an excessive amount of network bandwidth during synchronization. For this reason, the current implementation of Parallelized Ouroboros Praos relies on compressing proxies to noticeably reduce the amount of data transferred over the network: the compression ratio currently lies at around 4.5x. The proposed method can work by directly downloading uncompressed data from the Cardano stake pool nodes using the current network protocol; however, that would increase the requirements for user Internet connections and the necessary cumulative bandwidth of Cardano stake pool nodes to support such transfers, placing this costly burden on the shoulders of stake pools. For the above reasons, the use of compressing proxies seems a very fair choice for the demo of the technology.

Furthermore, the use of compressing proxies does not present risks to Cardano's decentralization since they serve purely as a content delivery network in which the requested data is identified by the hash of its content. This is analogous to the pattern of including a hash of data on the chain and relying on IPFS or other services for the users to retrieve the data. Moreover, this process is exactly the same as that of Mithril, which distributes the actual blockchain snapshots using Google's CDN. The only difference is that compressing proxies automatically add new data following Cardano's mainnet. [11] explains how the model of compressing proxies can be further decentralized while waiting for the support of compression in the Cardano network protocol.

To summarize, the use of compressing proxies is simply a matter of convenience and not a fundamental requirement of the method.

5.4 Present and future

The current implementation of Parallelized Ouroboros Praos performs full consensus validation that, as this note explains, provides for the core of the chain exactly the same security properties as the full validation performed by Cardano Node. This level of security is already better than that of Mithril, which is discussed below.

As to tail security, the current implementation relies on the dynamic querying of a centralized service provided by the IOG, the developers of Cardano, `relays-new.cardano-mainnet.iohk.io`. This is the status quo source for Daedalus in its non-peer-to-peer configuration. A conversion to dynamic querying of a sample of peers is planned around the time of Cardano's upgrade to Ouroboros Genesis.

Also, the current implementation does not support validation of script transaction witnesses at this time. The reason for that is that for a public demonstration of the method, one can synchronize the core of the chain with the proposed method and the tail with the Cardano Node. Given that the time necessary for the synchronization of the tail is negligible, this does not undermine the presented performance benefits. Furthermore, the initial analysis of Plutus byte code suggests that implementing script validation is fully feasible within a reasonable time frame.

5.5 Mithril

Mithril [4, 6] is a beautiful technology but it can only really shine in use cases with no alternative ways to verify the signed data, such as voting, or where the time of alternative data validation dominates the time of data transmission. However, the use case of bulk synchronization of blockchain data is not one of these cases. Blockchain data already contains all the necessary certificates for validation, and the security of directly validating all blockchain blocks outweighs the security of validating only a sample of signer signatures, as in Mithril. Furthermore, blockchain data is large in size, and its transfer time dominates verification time, as shown in [12].

Finally, the current lack of support for partial synchronization makes Mithril impractical for the use in personal wallets that synchronize their chain copy only sporadically. [12] has shown that even in eight-CPU-core configurations, which are available in entry-level laptops, Parallelized Ouroboros Praos achieves more than ten times better performance than Mithril when the synchronization period lies between six months and five days of blockchain data.

Last but not least, Mithril currently has not achieved its target level of signer adoption, further weakening its security properties.

5.6 Refinements to the Cardano network protocol

Even though the following refinements are not necessary for the method to work, their presence would be beneficial both for the method and the Cardano blockchain overall:

- Compressed data transfers (discussed in this note and in [11]);
- Cryptographically signed network protocol responses (discussed in this note);
- Hierarchical blockchain data discovery (discussed in [11]).

5.7 Cardano Node on request

The presented method is new and requires extensive testing. Thus, its deployment side-by-side with Cardano Node is proposed as an option in the Deadalus wallet. This enables users both to test the technology and choose an active validation method based on their current use case.

6 CONCLUSION

The note highlights the influence of performance on the security of personal wallet nodes, and presents an approach to the security of personal wallets that is better balanced with regard to user needs than the status quo implementation of the Cardano blockchain, the Daedalus wallet.

The proposed method achieves an order of magnitude better synchronization performance and thus provides users with the improved security and convenience of much quicker availability to transact with the blockchain. The method maintains the same resistance against typical attacks, with the exception of 51% attack. However, 51% attack is the fundamental weakness of all blockchains and not a specific weakness of this method.

In addition, the proposed method includes its own implementation of blockchain explorer functionality that does not leak explored data to the Internet, further improving the practical security of the method.

REFERENCES

- [1] 2024. Cardano Blockchain Explorer. <https://explorer.cardano.org/en>.
- [2] 2024. Daedalus Wallet. <https://daedaluswallet.io/>.
- [3] 2024. GitHub: Cardano Node source code repository. <https://github.com/IntersectMBO/cardano-node>
- [4] 2024. GitHub: Mithril source code repository. <https://github.com/input-output-hk/mithril>
- [5] Christian Badertscher, Peter Gazi, Aggelos Kiayias, Alexander Russell, and Vassilis Zikas. 2018. Ouroboros Genesis: Composable Proof-of-Stake Blockchains with Dynamic Availability. *IACR Cryptol. ePrint Arch.* (2018), 378. <https://eprint.iacr.org/2018/378>
- [6] Pyrros Chaidos and Aggelos Kiayias. 2021. Mithril: Stake-based Threshold Multisignatures. *IACR Cryptol. ePrint Arch.* (2021), 916. <https://eprint.iacr.org/2021/916>
- [7] Duncan Coutts, Neil Davies, Karl Knutsson, Marc Fontaine, and Alex Vieth Armando Santos, Marcin Szamotulski. 2024. The Shelley Networking Protocol Version 1.3.0. (2024). <https://input-output-hk.github.io/ouroboros-network/pdfs/network-spec/network-spec.pdf>
- [8] Bernardo Machado David, Peter Gazi, Aggelos Kiayias, and Alexander Russell. 2017. Ouroboros Praos: An adaptively-secure, semi-synchronous proof-of-stake protocol. *IACR Cryptol. ePrint Arch.* (2017), 573. <http://eprint.iacr.org/2017/573>
- [9] Muhammad Saad, Jeffrey Spaulding, Laurent Njilla, Charles A. Kamhoua, Sachin Shetty, DaeHun Nyang, and David Mohaisen. 2020. Exploring the Attack Surface of Blockchain: A Comprehensive Survey. *IEEE Commun. Surv. Tutorials* 22, 3 (2020), 1977–2008. <https://doi.org/10.1109/COMST.2020.2975999>
- [10] Alex Sierkov. 2023. Highly Parallel Reconstruction of Wallet History in the Cardano Blockchain. (2023). https://github.com/sierkov/daedalus-turbo/blob/main/doc/2023_Sierkov_WalletHistoryReconstruction.pdf
- [11] Alex Sierkov. 2023. Scalability of Bulk Synchronization in the Cardano Blockchain. (2023). https://github.com/sierkov/daedalus-turbo/blob/main/doc/2023_Sierkov_CardanoBulkSynchronization.pdf
- [12] Alex Sierkov. 2024. Parallelized Ouroboros Praos. (2024). <https://github.com/sierkov/daedalus-turbo/blob/main/doc/2024-sierkov-parallelized-ouroboros-praos.pdf>

revised March 22, 2024