Anti-fraud system for blockchain networks

Master's Thesis submitted to the

Constructor Institute

in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Software Engineering

presented by

Timur Mustafin

under the supervision of

Prof. Bertrand Meyer

co-supervised by

Eugene Aseev

June 2023

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Timur Mustafin
Schaffhausen, 28 June 2023

Abstract

This study addresses the critical issue of user security and safety within the blockchain ecosystem, which is often compromised due to the pseudo-anonymous nature of blockchain addresses. By leveraging historical on-chain data, the research develops and implements a system for evaluating and rating blockchain addresses, thereby reducing the risk of fraud and enhancing user trust in blockchain transactions. The study also involves the implementation of a voting smart contract, further contributing to the security measures within the blockchain ecosystem. The findings of this research could have significant implications for user adoption and the overall user experience within the blockchain ecosystem. However, it's important to note that the findings of this research do not directly contribute to enhancing the reliability of the blockchain system. Future research in this area could focus on user testing and further refinement of the rating algorithm, as well as exploring other potential factors that could improve UX and safety of blockchain systems for the end users.

Contents

Contents **iv**

1 Introduction **1**

2 Literature Review **3**
2.1 Web3 Architecture & Principals . . . 3
2.2 Blockchain Technology . . . 4
2.2.1 Design . . . 4
2.2.2 Public Blockchain Systems . . . 4
2.2.3 Private blockchain systems . . . 5
2.3 Security Concerns and Attacks on Blockchain Systems . . . 5
2.4 DNS hijacking and frontend attacks . . . 5
2.5 Heuristic Rating Systems . . . 6
2.6 Decentralized Autonomous Organizations (DAOs) . . . 6
2.7 Research Gap . . . 7

3 Research Methodology **9**
3.1 Description of the Proposed System . . . 9
3.1.1 Indexer structure . . . 9
3.1.2 Data model . . . 9
3.1.3 The Algorithm for Calculating Heuristic Rating . . . 11
3.1.4 Rule 1. Age of the account . . . 11
3.1.5 Rule 2. No mixers . . . 13
3.1.6 Rule 3. Contract deployment . . . 13
3.1.7 Rule 4. NFT holders . . . 13
3.1.8 DAO Setup for Rules Calculation and Enforcement . . . 13
3.1.9 On-Chain Verification Tool for an Address Rating . . . 13

4 Evaluation and Results **15**
4.1 Addresses for verification set . . . 15
4.2 Testing and Evaluation on Verification Set . . . 15
4.3 On-chain verification tool . . . 15

5 Discussion **19**

6 Conclusion **21**
6.1 Future work . . . 21

A DAO voting contract **23**

B Score state contract **25**

Bibliography cited **27**

Chapter 1

Introduction

In recent years, we have witnessed a significant rise in the popularity of public blockchain systems, notably Ethereum, Avalanche, and Polygon, primarily because of the enhanced privacy they offer to users. These systems offer pseudo-anonymity by permitting users to display only public addresses, derived from private keys, that are not inherently linked to any identifiable personal data. This attribute, though a remarkable stride towards privacy, has been increasingly exploited by malevolent actors who leverage this anonymity to execute a variety of attacks, such as frontend assaults and basic social engineering exploits.

The primary objective of this research is to enhance the safety and security of users within the blockchain ecosystem. By leveraging historical on-chain data, the study aims to provide a mechanism for evaluating and rating blockchain addresses, thereby reducing the risk of fraud and enhancing user trust in blockchain transactions. This could potentially lead to increased user adoption of blockchain technology and contribute to its continued growth and development.

This study fills a significant research gap in the field of blockchain technology. While much research has been conducted on the technical aspects of blockchain, less attention has been paid to the user experience, particularly in terms of safety and security. By focusing on the issue of fraud due to the pseudo-anonymous nature of blockchain addresses and utilizing historical on-chain data for evaluation, this study contributes to a more holistic understanding of blockchain technology and its implications for users.

The significance of this study lies in its potential to enhance user safety and security within the blockchain ecosystem. By addressing the issue of fraud due to the pseudo-anonymous nature of blockchain addresses and using historical on-chain data for evaluation, the study could have far-reaching implications for user adoption and the overall user experience within the blockchain ecosystem.

The following chapters will provide a detailed overview of the research methodology, the evaluation and results, and a discussion and conclusion based on the research objectives and results.

Chapter 2

Literature Review

2.1 Web3 Architecture & Principals

Web3, also known as the decentralized web, stands as the next stage in the evolution of the internet. After the "read-only" Web 1.0 and "read-write" Web 2.0, Web3.0 is aiming to be a "read-write-own" internet, giving users control and ownership over their online interactions and assets [1].

Web3.0 architecture stands on the following core principles[1]:

- **Decentralization:** Web3.0 breaks the monopoly of centralized entities over the internet and shares the ownership with its users and builders.
- **Permissionless:** Web3.0 enables equal access for everyone without exclusion.
- **Native Payments:** Web3.0 uses cryptocurrency for transactions, eliminating the dependency on banks and traditional payment processors.
- **Trustless:** Rather than relying on third-parties, Web3.0 employs incentives and economic mechanisms for operations.

Web3.0 introduces several key features [2]:

- **Ownership:** Users gain unprecedented control over their digital assets, enabled by technologies like non-fungible tokens (NFTs).
- **Censorship Resistance:** Web3.0 allows users to maintain their data and reputation even when they leave a platform, mitigating the risk of censorship.
- **Decentralized Autonomous Organizations (DAOs):** DAOs enable decentralized decision-making and ownership of a platform through tokens that work like company shares.
- **Identity:** Users can control their digital identity using an Ethereum address and ENS profile, making it secure, censorship-resistant, and anonymous.
- **Native Payments:** Web3.0 facilitates direct online transactions through cryptocurrencies.

2.2 Blockchain Technology

2.2.1 Design

Blockchain is a database running on a number of computers, the data is stored in a consequent groups ("blocks"), each one containing a set of new state transitions ("transactions") and a hash of a previous block. The new block production is handled by network miners or validators, and each single one of them has to agree on the new block. For consensus on how the new block should look like, a consensus algorithm is used [3]. This design prevents data tempering, although it does not illuminate attacks completely [4].

2.2.2 Public Blockchain Systems

In terms of data structure and internal organization, blockchain systems are essentially a set of consequent blocks, containing a batch of data (usually, transactions) and a hash of the previous block. Public blockchain systems also provide with decentralization: unlike centralized databases, there is no single authority which is capable of data repudiation, and new block production is done through consensus mechanism, which is yet not a bulletproof solution, but makes attacks extremely non-practical to carry out.

The first public blockchain system which successfully attracted attention and some significant adoption, both user and governmental, was Bitcoin (BTC). The original white paper by Satoshi Nakamoto proposed a decentralized peer-to-peer electronic cash system, which operates without the need for a central authority and uses cryptographic proof and consensus algorithm backed by computational hardness to prevent double-spending.

Although Bitcoin has an option to have simple programs to run on-chain, practically-wise, it was not enough. In order to allow decentralized permissionless program execution, Vitalik Buterin proposed Ethereum: a blockchain-based platform that enables developers to build and deploy smart contracts and decentralized applications. As a part of the system to execute smart contracts in a deterministic way, an Ethereum Virtual Machine (EVM) was created.

Regardless of the success of Ethereum as a decentralized computing platform (often referred to as "mainnet"), it is struggling to become the most adopted one, mainly due to high gas fees. This brought up the problem of "Ethereum scaling", which has seen in implementations of Layer two solutions (L2) like Polygon, Optimism etc. These systems mainly focus on solving the high costs of transactions on mainnet, by maintaining their own separate state and history, and committing only Merkle tree proofs on mainnet.

As an alternative to EVM-based chains and L2s, completely alternative chains were developed:

- Polkadot, acting as a multichain platform that enables different blockchains to interoperate and share information in a secure and scalable way, with a focus on governance, staking, and bonding.

- The Solana blockchain proposes a high-performance, permissionless blockchain that uses a novel timestamp system called Proof of History, along with other optimizations, to achieve high speed and scalability for decentralized applications and cryptocurrencies.

- The Internet Computer Protocol (ICP), proposes a decentralized and non-proprietary network to host the next generation of software and services, aiming to extend the func-

tionality of the public Internet, so it can become the world's primary compute platform, replacing traditional IT infrastructure.

- The Open Network (TON) is aiming to build a decentralized ecosystem to act as a foundation for different decentralized apps like TON DNS, TON Storage etc. It introduces a concept of asynchronous messages and focuses on building utility while achieving high TPS with low fees.

2.2.3 Private blockchain systems

A private blockchain system, also often referred to as a permissioned blockchain or Distributed Ledger Technology (DLT), is a type of blockchain network where participation is restricted to specific entities. Unlike public blockchains where anyone can join, validate transactions, and create new blocks, private blockchains require an invitation and must be validated by either the network starter or by a set of rules put in place by the network starter. Examples are:

- Corda [5] is a private, permissioned blockchain platform specifically designed for managing legal contracts and other shared data between businesses, with a focus on ensuring privacy and security, enabling direct peer-to-peer transactions, and providing interoperability across business networks.
- Hyperledger Fabric [6] is an open-source, enterprise-grade platform that enables the development of permissioned blockchains, offering modularity and versatility for a broad set of industry use cases, including finance, banking, supply chain and more, by featuring configurable and pluggable components.

2.3 Security Concerns and Attacks on Blockchain Systems

When it comes to security, there are many vectors to consider: cryptographic level attacks, consensus attacks, denial of service attacks, which mostly target the blockchain system security itself. The goal is to overwrite the system's state into the attacker's favor. These types of attacks are broadly covered in the literature [7], although we would like to focus on DNS hijaking and frontend attacks.

2.4 DNS hijacking and frontend attacks

Recent years have seen several attacks targeting blockchain companies and decentralized applications (DApps). Curve Finance, a decentralized trading platform, confirmed reports of a frontend attack where hackers compromised the website or domain name to divert users or their transactions to a malicious destination, causing a loss of approximately $570,000 in ETH [8].

Similarly, Kyber Network, a multi-chain DeFi protocol, experienced an exploit on its frontend leading to a loss of $265,000. The attacker inserted malicious code into KyberSwap's frontend, demonstrating the severity of these frontend vulnerabilities [9].

DNS hijacking has also been prevalent in the blockchain sphere. For instance, Convex Finance reported a DNS hijacking incident where users were prompted to approve malicious contracts on their site [10]. Cream Finance and PancakeSwap have reported similar incidents,

warning users about possible compromised DNS and advising not to input any sensitive information, such as seed phrases or private keys, on their websites [11], [12].

The question of insurance coverage in relation to frontend attacks came into the spotlight following a major hack on BadgerDAO. Nexus Mutual, a DeFi insurance protocol, faced controversy when it decided not to cover the $120M loss resulting from what was believed to be a frontend attack. The company argued that covering such incidents could potentially allow malicious actors to exploit the system [13].

The frequent occurrence of frontend attacks and DNS hijacking underscores the critical need for robust security measures and practices within blockchain platforms, along with the need for continuous user education on safe practices to prevent the loss of sensitive information. It also brings forth issues regarding insurance coverage for such incidents, which will shape the future of DeFi insurance [13].

Table 2.1. Financial Impact of Notable Frontend Blockchain Attacks

Platform	**Date**	**Financial Impact (USD)**
Curve Finance	August 10, 2022	$570,000
Kyber Network	September 1, 2022	$265,000
Convex Finance	June 24, 2022	Unknown
Cream Finance	March 15, 2021	Unknown
Pancake Swap	March 15, 2021	Unknown
BadgerDAO	December 1, 2021	$120,000,000

2.5 Heuristic Rating Systems

There are several proposed solutions to protect users of blockchain systems from human mistakes, scams and malicious actors. Shaker, M., Shams Aliee, F., & Fotohi, R. (2021) [14] proposed an online rating system using blockchain technologies to allow user-voted on-chain rating, based on EVM-compatible smart contract. Also, on-chain rating is proposed to be used in online education by Garg, A., Kumar, P., Madhukar, M., Loyola-González, O., & Kumar, M. (2022)[15]. The group proposes a blockchain-based online education content ranking system which offers a decentralized, trustworthy review and ranking process, leveraging independent subject-matter experts to enhance the integrity and validity of course ratings, and includes a novel parity score to compare similar courses across different platforms, effectively addressing current issues like data manipulation and lack of transparency in centralized rating systems. The solution is based on Hyperledger Fabric [6].

2.6 Decentralized Autonomous Organizations (DAOs)

A Decentralized Autonomous Organization (DAO) [16] is a blockchain-based entity that enables people to coordinate and govern themselves mediated by a set of self-executing rules deployed on a public blockchain through smart contracts and consensus protocols, allowing it to make decisions, manage resources, and execute tasks without centralized governance or a management team.

The first successful example was The DAO [17], which acted as a venture fund for other projects. Other DAO like MakerDAO [18] succeeded in issuing and maintaining their own cryptocurrency DAI [19].

2.7 Research Gap

This research aims to address a gap in research, allowing on-chain rating algorithm to be defined by a DAO, which would illuminate centralized approaches of score computation algorithm design in existing solutions. The aim is to create a more secure blockchain environment by highlighting reliable addresses, thereby reducing the potential impact of scams and enhancing the overall security of public blockchain systems.

Chapter 3

Research Methodology

3.1 Description of the Proposed System

We propose to create an open-source off-chain system to calculate on-chain rating for addresses based on their historical activity and provide on-chain tool for rating verification and enforcement. Te rules for calculation of such a rating will be defined by DAO, and the open-source nature of it will allow anyone to verify correctness. The goal is to allow users to have more confidence in verifying trustworthiness of the parties they collaborate with on-chain.

This chapter contains a detailed explanation of the system design, with the following subchapters:

- Indexer structure and data model.
- The algorithm for calculating heuristic rating.
- DAO setup for rules calculation and enforcement.
- On-chain verification tool for an address rating.
- Testing and Evaluation on verification set.

3.1.1 Indexer structure

In order to compute rating for each of the address, it is required to access each single historical transaction in the network and apply rating calculation rules. The pseudocode 1 demonstrates a high-level procedure on how the processing is done.

3.1.2 Data model

In our indexer, unlike popular EVM chains, the state and data stored in the blockchain will be lightweight and only store state transitions of the account's history Table 3.1 and the latest state Table 3.2.

This structure allows tracing rating changes occurred cross-chain (chain_id table), as well as when they occurred (block_number table). The more lightweight table allows cheap access to the data, including lookup and building of a Merkle tree.

Algorithm 1 Indexing Blockchain and calculating rating

```
1:  procedure ComputeRating(Blockchain)
2:      let address_transaction_counter = new Map()
3:      for i = 0 to length(Blockchain) - 1 do
4:          block = Blockchain[i]
5:          for j = 0 to length(block.transactions) - 1 do
6:              transaction = block.transactions[j]
7:              ▷ new_rating calculation by DAO defined rules
8:              address_info[transaction.address].rating = updateRating(new_rating)
9:          end for
10:     end for
11:     return address_transaction_counter
12: end procedure
```

Table 3.1. Historical Rating Change Table

Name	Size	Comment
chain_id	u256	EVM chain id
block_number	u64	EVM block number
address	20 bytes	EVM on-chain address
rating_diff	i32	change in the rating of this address
rule_id	u32	index of the rule which caused change in the rating

Table 3.2. Rating Table

Name	Size	Comment
address	20 bytes	EVM on-chain address
rating_score	i64	change in the rating of this address

3.1.3 The Algorithm for Calculating Heuristic Rating

The heart of the system is an algorithm that calculates a heuristic rating for each address. The rating is determined by evaluating the address's historical activity on the blockchain. Our data model efficiently captures and structures the relevant data for this calculation. All rules are assigned the same weight, and the total rating is calculated as a sum of all the outputs from each of the rule's functions. Each sore function would take block height as an argument.

$$\text{Total Score} = \sum_{i=1}^{n} \text{Score}_i$$

For the research and demonstration purposes, a set of 4 rules are used, Table 3.3.

Table 3.3. Blockchain Rating Calculation Rules

Rule Number	**Description**
1	The age of the account, determined by the difference between the latest block and the occurrence of the first transaction.
2	No interactions with known mixers. If the account has interacted with known mixers, the rating will be impacted negatively.
3	Awards for contract deployment.
4	Additional points are added for well-known non-fungible tokens (NFTs) associated with the account.

3.1.4 Rule 1. Age of the account

As one of the indicators of trust, it's proposed to determine the age of the account. Age is defined in amount of blocks passed from the first known transaction.

Required properties:

1. Punish new accounts and reward older ones
2. Age of the account should not grow linearly with time
3. The difference between "old" and "new" accounts should not be strictly tighten to one value, but loosen over a specified time period.

The function cannot be simply linear, because otherwise age of the account would simply overweight any other parameter and older accounts would be much more "trusted" than newer ones. In order to solve this, we can take a function which would punish to a certain extent new accounts and reward older ones, but the older the age of the account is, the slower growth should be. One of the functions which could fit the requirements might be $\arctan(x)$ (Figure 3.1), the range of the function is $(-\pi/2; \pi/2)$ with $\arctan(0) = 0$, which does not satisfy requirements 1 and 3.

Requirements 1 and 2 can be achieved by modifying the plain arc tangent:

Requirement 1: In order to punish newer accounts more, we can simply move the function by a α amount of blocks to the right. This will force accounts newer than α blocks to have negative rating.

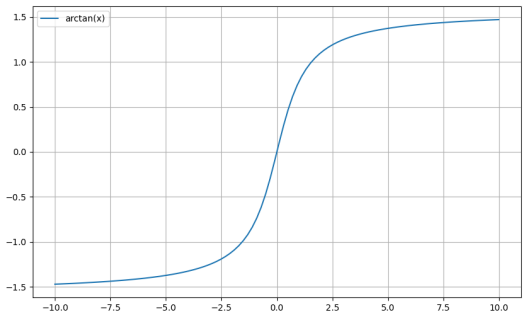

Figure 3.1. Plot of $\arctan(x)$

Requirement 3: The points of slope change we can visually observe approximately between $x = -2.5$ and $x = 2.5$ is the exact "transition time" which is required to be adjustable. In order to find the exact points, we can take a second derivative:

$$\arctan''(x) = \frac{-(2x)}{(1+x^2)^2}$$

Extrema points:

$$x_{min} = \frac{-1}{\sqrt{3}}$$

$$x_{max} = \frac{1}{\sqrt{3}}$$

And in order to adjust the transition time, we should solve the following problem:

$$f(x) = \begin{cases} \arctan(\frac{x_1-b}{c}) = \arctan(\frac{-1}{\sqrt{3}}, \\ \arctan(\frac{x_2-b}{c} = \arctan(\frac{1}{\sqrt{3}}, \end{cases}$$

where x_1 is the start of the transition period and x_2 is the end.

$$f(x) = \begin{cases} c = \frac{\sqrt{3}}{2}(x_2 - x_1), \\ b = x_2 - \frac{c}{\sqrt{3}} \end{cases}$$

x_1 – beginning of the transition period, x_2 – end of the transition period.
The choice of this exact function is driven by the facts:

- The arctan function is defined on the all Ox axis, making it simpler to understand.
- By itself, the function doesn't give huge benefit from having too old account, and bad actions are still going to hurt the rating, but at the same time and old account receives trust by default.
- It has a "transition period" between bad rating and good old one.

3.1.5 Rule 2. No mixers

A blockchain mixer is a service that enhances the privacy of blockchain transactions by muddling the transaction trail. It achieves this by combining the user's transaction with other transactions, making it difficult to trace the origins and destinations of any specific transaction. These kinds of services do not allow understanding if the parity is trustworthy or not, so we decided to punish interactions with them. Additionally, one of such services was sanctioned [20].

3.1.6 Rule 3. Contract deployment

Deployment of smart-contracts demonstrates active participation and contribution to the network's ecosystem. It showcases the deployer's commitment and understanding of the blockchain, enhancing trust and credibility within the community. So, we would like to reward such actions.

3.1.7 Rule 4. NFT holders

Non-fungible tokens (NFTs) are tokens that represent ownership of unique items [21], including digital ones. Well-known NFTs demonstrate the account owner's commitment to the ecosystem and more likely to be a trustworthy individual, since it has a financial commitment to the ecosystem.

3.1.8 DAO Setup for Rules Calculation and Enforcement

In order to ensure that our system is transparent and community ruled, we propose setting up a Decentralized Autonomous Organization (DAO). This DAO will be responsible for determining the rules for calculating the heuristic rating. It will also enforce these rules, providing an additional layer of trust in the system.

The DAO members can propose and vote on rules to be accepted or not. The reference implementation of such a smart contract is in the appendix A, and proposal structure is described in Table 3.4

Field	Description
ipfsHash	A string that represents the immutable IPFS[22] hash of the rule. This allows us to store complex and large data without bloating the Ethereum blockchain. only IPFS hash is stored on-chain, and then fetch the actual content off-chain.
voteCount	An unsigned integer that represents the total number of votes that this proposal has received. Each vote is weighted by the number of tokens the voter holds at the time of voting.

Table 3.4. Description of the Proposal struct

3.1.9 On-Chain Verification Tool for an Address Rating

Our system includes an on-chain tool that users can use to verify an address's rating. An ability to verify the rating on-chain is implemented with use of Merkle tree root and Merkle proof: a

state contract stores Merkle roots and allows verification of Merkle proofs B.1 on-chain. Example usage in a claim method, guarded by minimal required score:

```
function claim(bytes32 root, uint256 score, bytes32[] calldata proof) external {
    address user = msg.sender;
    require(score >= minScore, "Score too low");
    ScoreState.Leaf memory leaf = ScoreState.Leaf(user, score);
    require(scoringSystem.verify(root, proof, leaf), "Invalid proof");
    // success
}
```

Listing 3.1. Rating usage example in a contract

Chapter 4

Evaluation and Results

4.1 Addresses for verification set

For verification set, all addresses which had a transaction between blocks **17200000** and **17210000** on Ethereum blockchain were collected, it results in a verification set of **474171** unique addresses which had either incoming or outgoing transaction.

4.2 Testing and Evaluation on Verification Set

For the test setup, it was assumed that the current block height is *17220000*.

Age distribution for the accounts is shown on Figure 4.1. It shows a significant amount of accounts being recently created.

After the indexing, rating distribution on Figure 4.2, where the high frequency of data points around 0 from both positive and negative sides are shown, indicating that most of the accounts were not falling under any other rule for rating calculation except for the age.

Data points with negative rating are accounts interacted with mixers, while data points on the right are the ones which had either NFT tokens, either contract deployments.

Such information can help new users to understand which account is trustworthy and interactions with which ones should be more careful.

4.3 On-chain verification tool

In order to verify and enforce the account rating, a two-component system was developed:

1. off-chain API which returns an address's rating and along a Merkle proof which can be later verified on-chain

2. a smart-contract which holds several recent Merkle roots and a method to verify proofs.

The UI of the example off-chain application is showed on Figure 4.3. This application uses an off-chain API, which provides with user rating. If the rating is positive, it can be used to claim a reward on-chain using smart-contract, Figure 4.4 and transaction on Figure 4.5.

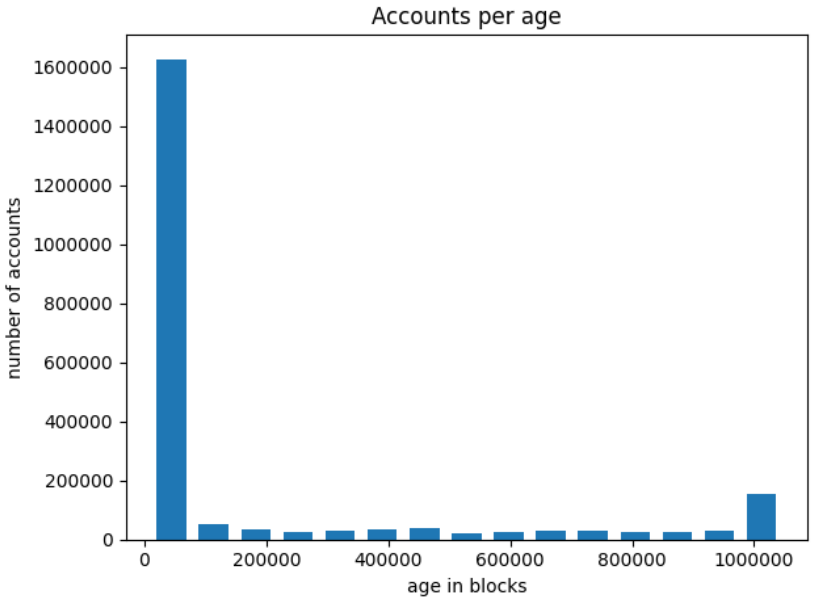

Figure 4.1. Age distribution among test set

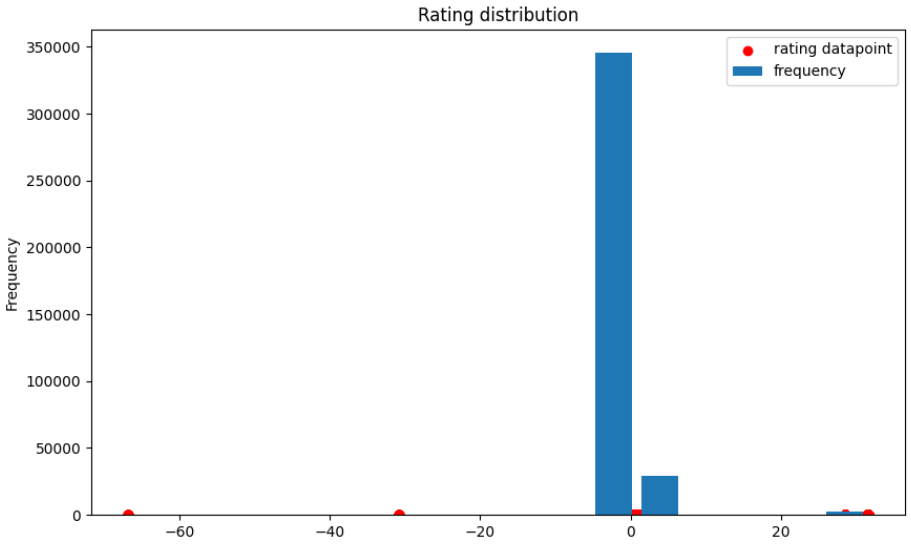

Figure 4.2. Rating distribution among test set

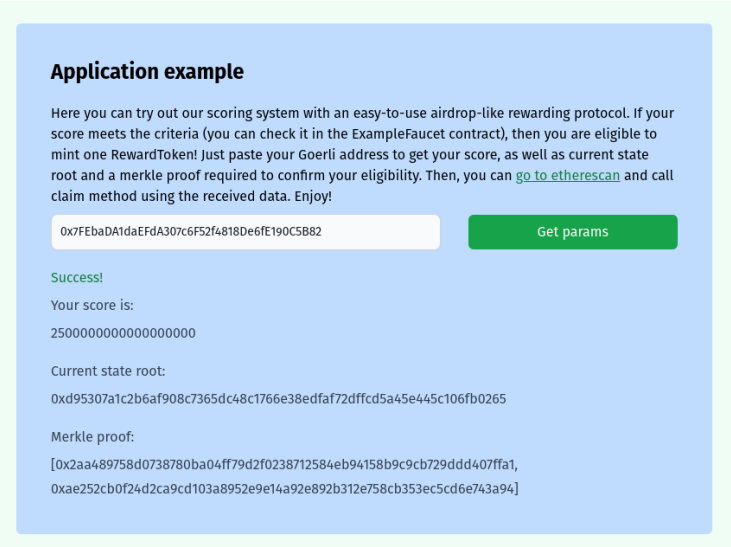

Figure 4.3. Example off-chain rating enabled application

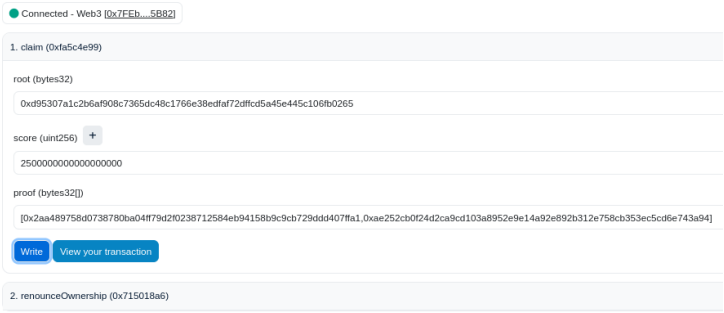

Figure 4.4. The process of claiming rating-gated token, providing the proof

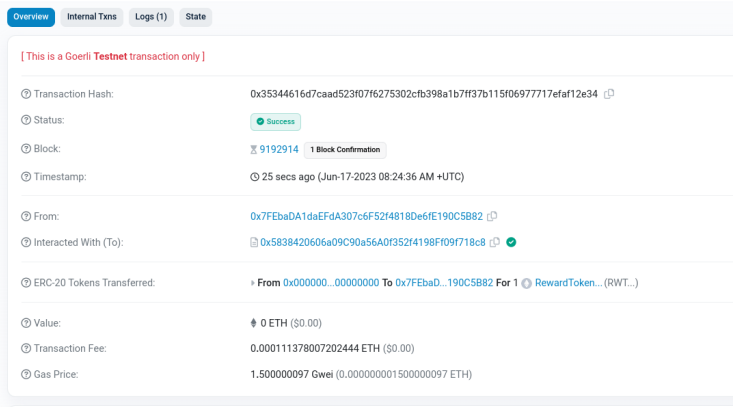

Figure 4.5. Transaction of successfully claimed reward

Chapter 5

Discussion

The results of this research have provided with a proof of concept tool and theoretical background, creating a trustless system for defining and assigning ratings to a blockchain address to fight common fraud.

Our findings suggest that the proposed system, which includes an indexer structure, a data model, an algorithm for calculating heuristic rating, rules for account age, no mixers, contract deployment, and NFT holders, a DAO setup for rules calculation and enforcement, and an on-chain verification tool for an address rating, is capable of achieving this goal. This aligns with the growing body of research emphasizing the need for more secure and transparent systems in the blockchain environment.

However, our study is not without limitations, missing one of the key aspects: user testing. User testing is a crucial aspect of system development, as it provides insights into how the system performs in real-world scenarios and how users interact with it. Without this testing, there may be issues or challenges that were not addressed during the development process.

Another drawback of the research is that chosen rating assigning rules setup is not ideal, where age of the account is the only one rule which can be applied to the majority of accounts. That might be a limitation of the dataset or approach, and further research is required.

Despite these limitations, our study provides a valuable contribution to the adoption of blockchain technology. It demonstrates a practical approach to enhancing security and trust within the blockchain ecosystem, addressing customer safety and not deep-technical problems.

Future research could build upon our findings by conducting extensive user testing to identify potential issues and improve the system. Additionally, future studies could explore other factors that could be incorporated into the rating algorithm to provide a more comprehensive and accurate rating system. Also, the work on creating trustless indexing environment is required.

In conclusion, our research has advanced the understanding of blockchain security by demonstrating a feasible approach to creating a trustless system for defining and assigning ratings to blockchain addresses. We recommend that future research in this area focuses on user testing and further refinement of the rating algorithm.

Chapter 6

Conclusion

The study conducted has provided significant insights into the evaluation of blockchain addresses. The research was centered around the collection and analysis of addresses which had a transaction between specific blocks on the Ethereum blockchain. This resulted in a comprehensive verification set of unique addresses, which were then subjected to further testing and evaluation.

The implications of these findings are substantial, particularly in terms of user adoption and user-facing security and safety within the blockchain ecosystem. By providing a mechanism for evaluating and rating blockchain addresses, users can have greater confidence in the transactions they engage in. This could potentially lead to increased user adoption as the perceived risk associated with blockchain transactions is reduced.

Furthermore, the enhanced user-facing security and safety could also contribute to a more robust blockchain environment. Users can be more assured of the safety of their transactions, reducing the potential impact of scams and enhancing the overall user experience within the blockchain ecosystem.

However, it's important to note that while these findings can enhance user-facing security and safety, they do not necessarily affect the reliability of the blockchain system. The reliability of a blockchain system is typically determined by factors such as the robustness of the underlying technology, the stability of the network, and the resilience of the system against potential attacks or failures. These aspects were not the focus of this study and thus, the findings of this research do not directly contribute to enhancing the reliability of the blockchain system.

In conclusion, this research has made a valuable contribution to the field of blockchain technology by providing a feasible approach for evaluating and rating blockchain addresses. This could have far-reaching implications for user adoption and user-facing security and safety within the blockchain ecosystem. Future research in this area could focus on user testing and further refinement of the rating algorithm, as well as exploring other potential factors that could enhance the reliability of the blockchain system.

6.1 Future work

As the continuation of this research, the following might be considered:

- User verification

- New rules development and verification
- Implementation of decentralized indexing as a separate network or an L2 solution

Appendix A

DAO voting contract

```
pragma solidity >=0.8.0 <0.9.0;

import "@openzeppelin/contracts/token/ERC20/IERC20.sol";

contract Voting {

    struct Proposal {
        string ipfsHash; // IPFS hash of the rule
        uint voteCount; // number of accumulated votes
    }

    IERC20 public token;
    Proposal[] public proposals;
    mapping(address => uint) public votes;
    uint public votingEnds;

    event NewRule(string ipfsHash, uint voteCount);
    event Voted(address voter, uint proposal, uint voteCount);

    constructor(address _tokenAddress, uint _votingPeriod, string[] memory _proposals) {
        token = IERC20(_tokenAddress);
        votingEnds = block.timestamp + _votingPeriod;

        for (uint i = 0; i < _proposals.length; i++) {
            proposals.push(Proposal({
                ipfsHash: _proposals[i],
                voteCount: 0
            }));
        }
    }

    function vote(uint proposal) external {
        require(block.timestamp < votingEnds, "Voting period is over");
        require(proposal < proposals.length, "Invalid proposal");

        uint balance = token.balanceOf(msg.sender);
        require(balance > votes[msg.sender], "No available tokens for more voting");

        proposals[proposal].voteCount += balance - votes[msg.sender];
        votes[msg.sender] = balance;
```

```
41
42          emit Voted(msg.sender, proposal, balance - votes[msg.sender]);
43      }
44
45      function winningProposal() public view returns (uint winningProposal_, uint winningVoteCount_
            ) {
46          uint winningVoteCount = 0;
47          for (uint p = 0; p < proposals.length; p++) {
48              if (proposals[p].voteCount > winningVoteCount) {
49                  winningVoteCount = proposals[p].voteCount;
50                  winningProposal_ = p;
51              }
52          }
53          winningVoteCount_ = winningVoteCount;
54      }
55
56      function winningIpfsHash() public view returns (string memory winningIpfsHash_) {
57          (uint winningProposal_, ) = winningProposal();
58          winningIpfsHash_ = proposals[winningProposal_].ipfsHash;
59      }
60  }
```

Listing A.1. Voting Smart Contract

Appendix B

Score state contract

```
1  pragma solidity ^0.8.13;
2
3  import "@openzeppelin/contracts/access/Ownable.sol";
4  import "@openzeppelin/contracts/utils/cryptography/MerkleProof.sol";
5
6  contract ScoreState is Ownable {
7      mapping(uint256 => bytes32) public roots;
8      uint32 public constant STATE_HISTORY_SIZE = 5;
9      uint32 public curTreeIndex = 0;
10
11     struct Leaf {
12         address addr;
13         uint256 score;
14     }
15
16     constructor(bytes32 root) {
17         roots[0] = root;
18     }
19
20     function updateRoot(bytes32 root) external onlyOwner {
21         uint32 rootIndex = (curTreeIndex + 1) % STATE_HISTORY_SIZE;
22         roots[rootIndex] = root;
23         curTreeIndex++;
24     }
25
26     function verify(
27         bytes32 root,
28         bytes32[] calldata proof,
29         Leaf calldata leaf
30     ) external view returns (bool) {
31         require(checkRoot(root), "Unknown root");
32         bytes32 leafHash = hashLeaf(leaf);
33         // ‘verifyCalldata‘ is a cheaper version of ‘verify‘
34         return MerkleProof.verifyCalldata(proof, root, leafHash);
35     }
36
37     function checkRoot(bytes32 root) public view returns (bool) {
38         if (root == 0) {
39             return false;
40         }
41         for (uint32 j = 0; j < STATE_HISTORY_SIZE; j++) {
```

```
            uint32 index = curTreeIndex - j;
            if (root == roots[index]) {
                return true;
            }
            if (index == 0) {
                break;
            }
        }
        return false;
    }

    function latestRoot() public view returns (bytes32) {
        return roots[curTreeIndex];
    }

    function hashLeaf(Leaf calldata leaf) internal pure returns (bytes32) {
        return
            keccak256(
                bytes.concat(keccak256(abi.encode(leaf.addr, leaf.score)))
            );
    }
}
```

Listing B.1. Score state contract

Bibliography cited

[1] *Introduction to web3*, https://ethereum.org/en/web3/.

[2] *Web2 vs web3*, https://ethereum.org/en/developers/docs/web2-vs-web3/.

[3] *Consensus mechanisms*, https://ethereum.org/en/developers/docs/consensus-mechanisms/.

[4] S. Sayeed and H. Marco-Gisbert, "Assessing blockchain consensus and security mechanisms against the 51% attack," *Applied sciences*, vol. 9, no. 9, p. 1788, 2019.

[5] *Corda*, https://corda.net/.

[6] *Hyperledger fabric*, https://www.hyperledger.org/use/fabric.

[7] M. Saad, J. Spaulding, L. Njilla, *et al.*, "Exploring the attack surface of blockchain: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1977–2008, 2020.

[8] *Ethereum defi exchange curve suffers frontend hack*, https://decrypt.co/107120/ethereum-defi-exchange-curve-frontend-hack-hijack.

[9] *Defi protocol kyber network suffers frontend hack, loses $265k*, https://cryptoslate.com/defi-protocol-kyber-network-suffers-frontend-hack-loses-265k/.

[10] *Convex finance hack*, https://twitter.com/ConvexFinance/status/1540104036229185536.

[11] C. Finance, *Cream finance*, https://twitter.com/CreamdotFinance/status/1371448627663491088, 2021.

[12] P. Swap, *Pancake swap*, https://twitter.com/PancakeSwap/status/1371471934999777281, 2021.

[13] O. Fernau, *Defi insurer nexus mutual weighs whether to pay out on badgerdao hack*, https://thedefiant.io/badgerdao-hack-insurance-payout/, 2021.

[14] M. Shaker, F. Shams Aliee, and R. Fotohi, "Online rating system development using blockchain-based distributed ledger technology," *Wireless Networks*, vol. 27, no. 3, pp. 1715–1737, 2021.

[15] A. Garg, P. Kumar, M. Madhukar, O. Loyola-González, and M. Kumar, "Blockchain-based online education content ranking," *Education and information technologies*, pp. 1–23, 2022.

[16] S. Hassan and P. De Filippi, "Decentralized autonomous organization," *Internet Policy Review*, vol. 10, no. 2, pp. 1–10, 2021.

[17] *The dao*, https://www.economist.com/finance-and-economics/2016/05/19/the-dao-of-accrue.

[18] *Makerdao*, https://makerdao.com/en/.

[19] *Dai cryptocurrency*, https://makerdao.com/en/whitepaper/.

[20] *Ofac's tornado cash sanctions*, https://www.sanctions.io/blog/ofacs-tornado-cash-sanctions.

[21] *Non-fungible tokens*, https://ethereum.org/en/nft/.

[22] *Ipfs*, https://ipfs.tech/.