PRISM: A Blockchain-Enabled Reputation-Based Consensus for Enhancing Scientific Workflow Provenance

Matthew Miller*
Department of Computer Science
Marshall University
matthewmiller0110@gmail.com

Skarlet Williams*
Department of Computer Science
Boise State University
skarletwilliams@u.boisestate.com

Gaby G. Dagher
Department of Computer Science
Boise State University
gabydagher@boisestate.edu

Min Long
Department of Computer Science
Boise State University
minlong@boisestate.edu

***Abstract*—Recent surveys and reports have shed a spotlight on the disconcerting prevalence of scientific fraud, prompting the call for robust systems to uphold integrity in scientific research. In this paper, we introduce PRISM, a novel blockchain-based solution designed to address the challenges of storing provenance records for scientific workflows on a decentralized ledger. PRISM aims to enhance the reputability of scientific findings by providing a flexible and dynamic framework that accommodates the evolving nature of scientific research. We introduce a reputation-based quorum consensus protocol (POER) that involves two pivotal actors: miners and quorum members. Reputation is a central aspect of the protocol, motivating miners to provide accurate and timely results. The quorum composition dynamically adjusts after each block addition to involve the most trustworthy and effective nodes in decision-making processes. We describe the process of selecting quorum members using reputation and task sharding to efficiently divide workflow tasks among miners. Additionally, we outline the capability of PRISM to support workflow modifications, allowing researchers to adapt workflows during experiments while maintaining complete transparency and immutability. Our experimental evaluation highlights the fairness and scalability of PRISM.**

***Index Terms*—Blockchain; Scientific Workflow; Consensus Mechanism; Reputation; Provenance;**

I. INTRODUCTION

Scientific fraud is not a new phenomenon; it has been a disconcerting presence throughout the annals of scientific history. Recent surveys and reports, however, have thrown a spotlight on its worrying prevalence. An anonymous survey conducted at Dutch Universities revealed that nearly eight percent of participating scientists admitted to manipulating and/or fabricating data at least once between 2017 and 2020 [1][2]. In the medical and life science disciplines, this number rose to over ten percent. Furthermore, the National Cancer Institute reported that 0.25 percent of trial data was found to be fraudulent in 2016 [3]. These findings are reminiscent of the various instances of scientific fraud brought to light in "Betrayers of Truth" by William J. Broad in 1982 [4], which is considered one of the earliest comprehensive investigations into scientific misconduct. This historical yet still prevalent issue underscores the need for robust systems to uphold integrity in scientific research.

To effectively address this issue, it is critical to understand scientific workflows and their provenance. A scientific workflow is a sequence of ordered, non-linear tasks linked by their inputs and outputs. The provenance records are comprehensive documentation of workflow history, detailing every procedure, data origin and progression, stages of analysis, deployed parameters, intermediate computations, and data transformations [5]. Such a thorough record, provided it is clear and unambiguous, enables unaffiliated parties to reproduce and validate the original results, thereby ensuring the integrity of scientific investigations. Given the challenges and the critical need for verifiable, reproducible, and secure scientific workflow provenance, the use of blockchain technology presents an innovative solution. This technology provides an immutable, decentralized ledger to safeguard against unauthorized data modifications, ensuring the integrity of provenance records. However, existing blockchain-based solutions for workflow provenance, while promising, often assume the existence of an effective and fair consensus mechanism - a crucial component that is not sufficiently addressed in the current research.

We address this gap with the introduction of PRISM: a Provenance Reputation-based Invalidatable Scientific-workflow Mechanism. In addition to a secure, trustworthy, and fair consensus mechanism, PRISM integrates a robust consensus mechanism into a blockchain-based audit system for scientific workflow provenance and enables adaptability with workflow modification and data invalidation capabilities. This paper will outline the design of PRISM, its extensive testing process, and its contributions to combatting the ongoing

*These authors contributed equally.

challenge of scientific fraud, ultimately enhancing the integrity of scientific research.

A. Contributions

The key contributions of this paper are as follows:

1) We propose PRISM: a Provenance Reputation-based Invalidatable Scientific-workflow Mechanism. PRISM is a blockchain-based audit for scientific workflow provenance. Our system's main features provide a reliable foundation for research collaboration and empowers researchers to adapt their scientific workflows dynamically, ensuring accuracy and relevance in the face of evolving requirements and changing experimental conditions. PRISM also ensures network-wide optimization and scalability by breaking up scientific workflow tasks and allowing computations to be more efficient.
2) Proof of Earned Reputation (POER) is a quorum-based consensus protocol specifically tailored to maintain the integrity of data within the scientific workflow provenance on the blockchain. POER ensures that the quorum primarily consists of nodes with good reputations, enhancing the reliability and trustworthiness of decision-making.
3) To validate the effectiveness and reliability of our solution, we implemented our reputation-based consensus mechanism on the open-source project BlueChain[6]. Through extensive testing in simulated environments, we assessed the fairness and scalability of our design. The testing process allowed us to identify and address any potential vulnerabilities, ensuring that our solution achieves data integrity and adaptability for the scientific workflow provenance.

II. Related Work

The landscape of research leveraging blockchain technology for data provenance tracking is vast. Applications can spanning various domains such as supply chain management [9][14][15][16], IoT [17][18][19], and cloud computing [11][20][21], among others. However, the niche of scientific workflow provenance presents unique challenges and requirements that are not directly addressed by these solutions.

Several solutions specifically designed for scientific workflow provenance have emerged [7][8][9][10][12][13], each with their own merits but also limitations. DataProv [13], for instance, offers threshold-based voting systems and customizable smart contracts to validate provenance records, emphasizing the importance of reproducibility and the preservation of data lineage. ProductChain [9] and Nizamuddin *et al.* [22] propose the use of a decentralized database called IPFS (InterPlanetary File System) to store every data state for record verification between the blockchain and the database. This solution ensures consistency and reliability in capturing and managing data in a scientific workflow provenance. SciLedger [7] stands out by supporting complex workflows. The blockchain shapes the scientific workflow, allowing for branching and merging of data inputs similar to the scientific process. Similarly, SciChain [10] introduces a solution optimized for high-performance computing systems and focuses on supporting large-scale scientific workflows.

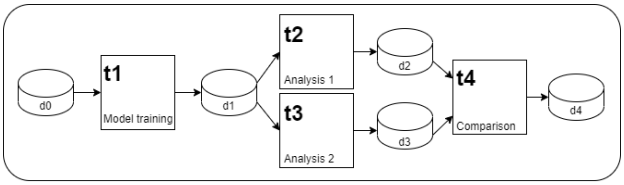

Figure 1: An example scientific workflow

Consensus mechanisms commonly used in these contexts, such as Proof of Work (PoW) and Proof of Stake (PoS), pose their own problems. For instance, PoW is prone to selfish mining, majority manipulation, and consensus delay [11]. PoS, as proposed by Tosh *et al.* in BlockCloud [11], sidesteps some of these issues but doesn't cater specifically to the scientific workflow environment. Table I highlights the pressing need for a solution that supports all types of workflows and incorporates an appropriate consensus protocol

III. Preliminaries

The following provides useful background information for the rest of this paper.

A. Blockchain

A blockchain is a distributed digital ledger that records transactions across a network of computers. Each transaction is grouped into 'blocks' that are cryptographically linked to form a 'chain'. Notable features of blockchain include transparency, security, and immutability. By removing intermediaries, it enables trustless interactions. Blockchain is applied in sectors like finance, supply chain management, and voting systems and also supports smart contracts for automated agreements.

B. Consensus Algorithms

Consensus algorithms are essential for establishing agreement within distributed networks, such as blockchains. They work by creating a unified state across the network, even when there are multiple actors with potentially conflicting inputs. Depending on the protocol, these algorithms might require either a subset of participants (nodes) to agree or participants to provide proof of specific criteria [23]. In essence, they act as a democratic system within a network, ensuring all participants adhere to the agreed-upon state of the system and thus maintain the integrity and security of the network.

C. Scientific Workflow Provenance

Scientific workflows describe the procession of processes and tasks that must be completed to satisfy the workflow [5]. This includes the origin and progression of data, stages of analysis, parameters deployed, intermediate computations, and data transformations. Figure 1 displays an example of a simple scientific workflow. A Scientific Workflow Provenance details the history of a Scientific Workflow. This encompasses all of the details of the scientific workflow, including the start date,

Table I: Comparative evaluation of main features in closely related works

Papers	Workflow Type			Provenance Consensus Protocol	Privacy Preserving	Open Source
	Simple	Complex	Dynamic			
SciLedger [7]	✓	✓	✗	✗	✗	✓
SciBlock [8]	✓	✗	✓	✗	✓	✗
ProductChain [9]	✓	✗	✗	✗	✓	✗
SciChain [10]	✓	✗	✗	POST/POET	✗	✓
BlockCloud [11]	✓	✗	✓	✗	✓	✗
BlockFlow [12]	✓	✗	✗	✗	✗	✓
DataProv [13]	✓	✗	✓	✗	✓	✗
This paper: PRISM	✓	✓	✓	POER	✗	✓

hypothesis, processes and tasks, the data output from each task, and any modifications made to the original workflow. The provenance record should be clear and unambiguous. This allows scientists to keep a record of all changes to their data, ensuring scientific integrity and reproducibility.

D. Sharding

Sharding is a technique used in distributed computing to efficiently distribute tasks or data across multiple nodes or servers by breaking them into smaller parts called "shards." This approach reduces redundancy and ensures scalability. In blockchain systems, sharding improves transaction processing and network performance [24].

IV. Problem Formulation

Scientific research relies heavily on the ability to record and verify the provenance of workflows—ensuring transparency, reproducibility, and trustworthiness of provenance records. An anonymous survey at Dutch Universities revealed that approximately eight percent of participating scientists falsified and/or fabricated data at least once between 2017 and 2020 [1][2]. This statistic rises to more than ten percent within the medical and life science research community. Furthermore, in 2016, the National Cancer Institute found 0.25 percent of trial data was fraudulent [3].

The emerging technology of blockchain holds promise in facilitating secure and decentralized provenance record storage. Tracking data provenance on the blockchain has been previously explored, but current consensus protocols employed do not adequately address the unique requirements and challenges of scientific workflow provenance. Due to this, it is essential to examine the need for a specialized consensus protocol that caters to scientific workflow provenance on the blockchain.

Existing consensus protocols such as proof of work (PoW) and proof of stake (PoS) have been widely utilized in various blockchain applications. However, these protocols were primarily designed to address the security and consensus needs of financial transactions and are not fully aligned with the distinctive characteristics of scientific workflow provenance. For example, PoW is a very computationally demanding method of consensus, and as discussed by Tosh *et al.* [11], it is not a viable consensus mechanism for this reason. Despite Tosh *et al.* [11] arguments, proof of stake is also insufficient for this task. In terms of storing research provenance on the blockchain, the amount of "stake" any scientist has in "proof of stake" would be the number of provenance records they have stored on the blockchain. Merely participating in the blockchain and considering the number of provenance records as a scientist's 'stake' is inadequate. Therefore, a variation or enhanced version of the proof of stake mechanism is required to address these limitations.

SciBlock incorporates a unique variation of the proof of stake (PoS) mechanism, termed Proof of Authority (PoA), where a scientist's real-world identity constitutes their stake in the system. It operates under the presumption that established scientists are inherently more reliable, given the potential risk to their professional reputation should their actions on the network prove untrustworthy. However, we posit that this assumption - relying on past trustworthiness as a reliable indicator of future behavior - is insufficient and somewhat simplistic. We strongly believe in the need for continuous demonstration of trustworthiness; scientists should constantly validate their credibility to maintain the trust vested in them over time. SciChain introduces two models: Proof of Scalable Traceability (POST) and Proof of Extended Traceability (POET), as outlined by Abdullah Al-Mamun *et al.* [10]. These consensus mechanisms are also insufficient because scientists must first compute the outcomes of their own provenance records. This can allow for the manipulation and collusion of malicious actors to get their provenance records added to the blockchain unfairly. There is a need for a consensus mechanism that allows other researchers to verify the provenance records without the possibility to fake their participation. The design of a more robust consensus mechanism must support the verification of results and ensure that the recorded provenance information remains immutable and tamper-proof. These specific requirements necessitate a consensus protocol tailored to the intricacies of scientific workflows.

V. SOLUTION: PRISM

A. Solution Overview

We propose PRISM, a unique solution specifically designed for storing provenance records on a decentralized ledger. PRISM addresses existing challenges in current systems by creating a flexible framework capable of evolving with the dynamic nature of scientific workflows. PRISM is a blockchain-based scientific workflow provenance mechanism for scientists to increase the reputability of their scientific findings.

In order for the PRISM structure to adequately display scientific workflow provenance on the ledger, four blocks are

needed to represent the different operations required by a workflow. These include Workflow Inception Blocks (WIBs) – marking the initiation of a workflow, Workflow Task Blocks (WTBs) – maintaining a continuous record of individual tasks within a given workflow, Data Invalidation Blocks (DIBs) – signifying tasks that have been erroneously computed, and Workflow Modification Blocks (WMBs) – permitting the alteration and reconfiguration of a workflow. The provenance of a workflow is tracked through the addition of these blocks to the ledger. For workflow task blocks, the addition to the blockchain follows a 9-step process. Figure 2 displays the typical process of adding a WTB to the chain. Each of the other blocks requires only 6 steps since they do not require miners.

1) Generate the quorum from the top x-th percentile of nodes, based on reputation. Use the previous block's data as a source of randomness.
2) Each member of the quorum establishes a communication link with every other member.
3) The quorum collectively decides on the next record to be added. This decision is based on the time elapsed since a workflow was last processed and the reputation of the scientists assigned to the workflow. Any transaction lacking required fields is automatically rejected.
4) Each quorum member contacts their adjacent nodes, designating them as miners and providing them with the task to be executed.
5) Each designated miner performs the assigned task.
6) Upon completion of the task, each miner sends the output back to their respective quorum member.
7) The quorum members collate and compare data sets in the form o, m, n, where o is the most common output from their miners, m is the percentage of their miners that produced this output o, and n is the total number of miners they assigned to the task. For the workflow task to be considered reliable, at least 66 percent of miners must have obtained the same output o across the quorum. If this condition is not met, the block is rejected.
8) Once the data is validated, the quorum finalizes and signs the block.
9) The quorum disseminates information about the new block to other nodes in the network through a process known as gossiping.

At the end of this process, whichever scientist(s) had their block computed lose a percentage of their reputation. This is to prevent malicious scientists from flooding the network with illegitimate workflows. All miners either earn or lose reputation for either correctly computing the computation or not. This information is appended to the block to allow other nodes to calculate reputations. Quorum members' reputations do not change from being a quorum member. This is specifically to ensure that high-reputation nodes do not unfairly earn more reputation without doing more work, in other words- the rich don't get richer.

B. Block Types

Within the PRISM framework, blocks are primarily classified into four distinct categories, exclusive of the genesis block. These categories encompass:

1) Workflow Inception Blocks (WIBs): These serve as the foundational building blocks, marking the initiation of a workflow.
2) Workflow Task Blocks (WTBs): These are integral for maintaining a continuous record of individual tasks within a given workflow.
3) Data Invalidation Blocks (DIBs): These signify tasks that have been erroneously computed, indicating a requisite for revision and correction.
4) Workflow Modification Blocks (WMBs): These permit the alteration and reconfiguration of a workflow, thereby fostering adaptive flexibility.

This systematic classification of block structures not only facilitates the creation and routine tracking of new workflows on the blockchain, but also ensures a streamlined ability to invalidate inaccurate tasks and dynamically modify workflows as needed.

C. Reputation-based Quorum Consensus Protocol

Our system incorporates two pivotal actors: miners and quorum members. Miners, the nodes lending their computational power to the network, execute provenance tasks on input data to yield output data. A miner's reputation within the system is based on accurate, quick, and consistent task performance. The higher a node's reputation, the greater the trust vested in it. This dynamic motivates miners to contribute valuable computations to the network.

In terms of network topology, every miner is responsible for reporting their task output to a neighboring quorum member who delegated the task to them. The responsibility of reaching a consensus on the authenticity of the miners' output rests on the shoulders of quorum members. Notably, the quorum composition is not static but is recalculated after each block is added to the ledger. This dynamic quorum is comprised of a subset of nodes with high reputation scores at the time of the block's addition. This fluidity in quorum composition ensures the involvement of the most trustworthy and effective nodes in decision-making processes. Inherently, a high-reputation node will be selected as a quorum member with greater frequency, therefore decreasing the probability that they will be selected as a miner. This has the added benefit of not creating more work for high-reputation nodes.

1) Reputation Calculation: A complex formula (1) is required to handle the various factors we wish to consider when calculating a node's reputation. We also must ensure that the formula is fair to nodes who have weaker computational systems or miners who are geographically far away from their quorum members. We resolve these requirements by assigning weights to important variables within the formula. The three major variables used to influence reputation are accuracy (A), time (T), and consistency (C).

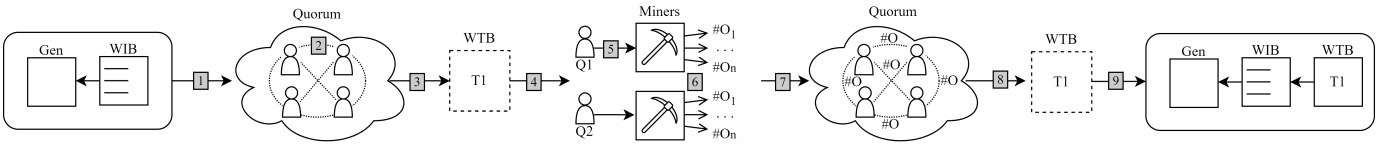

Figure 2: The addition of a block to the blockchain. 1: The quorum is generated from the ledger. 2: Quorum members establish connection to each other. 3: The quorum decides the next transaction. 4: Each quorum member assigns miners to the task. 5: Every miner computes the task. 6: Miners return their output to their quorum member. 7: The quorum decides the true output. 8: The quorum completes the block. 9: The block is added to the ledger. Steps 4-7 are exclusive to task blocks.

Workflow inception block

- Unique workflow ID
- Workflow overview
- Hypothesis
- Authorized scientists
- Public verification hash
- Reputation Updates
- Quorum Signatures

Workflow task block

- Unique workflow ID
- Task ID
- Input hashes
- Output hashes
- Validation trees
- Scientist proof
- Reputation updates
- Quorum signatures

Data invalidation block

- Unique workflow ID
- Task ID (to invalidate)
- Justification
- Validation trees
- Threshold proof
- Reputation Updates
- Quorum signatures

Workflow modification block

- Unique workflow ID
- Modified workflow
- Justification
- Threshold proof
- Reputation Updates
- Quorum signatures

Figure 3: An enumeration of each block type

Variable	Definition
R_i	Reputation of node i
n	Length of the ledger
x	Relevant block depth
α	Weight for accuracy
A	Accuracy
β	Weight for time
T	Time - minimum time
γ	Weight for consistency
C	Consistency
ϕ	Blocks participated

Table II: Notation Table for Reputation Formula (1)

$$R_i = \frac{\gamma C_i}{\phi_i}\left(\alpha \sum_{n}^{n-x} A_i + \beta \sum_{n}^{n-x} 2^{-T_i}\right) \tag{1}$$

The reputation score R_i of node i is normalized to lie in the interval $[0, 1]$. The time duration T_i signifies the time it took for i to finish mining a block, starting from when the task was first assigned to when the output is received by the quorum member. To ensure fairness when comparing different difficulties of tasks, this value is calculated relative to the fastest correct completion time for the same block by any miner. This is expressed by the formula $T_i = t_i - \min(T)$, where t_i denotes the time that miner i spent to successfully mine the block and $\min(T)$ symbolizes the quickest time any miner has taken to correctly mine the same block. This formula represents the relative efficiency of miner i when it comes to mining a block. Note that the mining process is divided based on sharding. This adds complexity to the tracking of elapsed time. In short, the time t_i for a miner i computing a sub-workflow at depth d is L_i/d, where L_i is the number of computational tasks assigned to miner i. This will be elaborated further later. The accuracy of the output from miner i is denoted by $A_i \in \{-1, 1\}$. If miner i's output aligns with the most frequent output, which is considered the correct result from the mining operation, A_i is assigned a value of 1; otherwise, it is assigned -1. The consistency of miner i is denoted by C_i and indicates the reliability of the miner in generating accurate results over time. The total number of blocks in the ledger is denoted as n. We use the term x, where $x \leq n$, to specify the number of blocks preceding the most recent one that is included in the computation of the reputation score. This introduces a 'sliding window' approach that incentives ongoing involvement from miners to maintain a high reputation score. The quantity $\phi_i \leq x$ represents the number of blocks where miner i has participated, serving to normalize the reputation calculation. Lastly, the coefficients α, β, and γ are defined such that $\alpha \in (0, 1)$, $\beta = 1 - \alpha$, and $\gamma \in [0, 1]$. These coefficients are used to assign weights to the previously mentioned variables. To discover the most equitable distribution of computational power, we experimentally adjusted these weights via a Python script to determine which weights generate a reputation distribution with the least overall standard deviation in relation to the inverse of their computational time. A small standard deviation ensures that reputation is more evenly distributed among well-behaving scientists, despite any discrepancies in computational power. Figure 4 includes the plot distribution of reputation scores depending on the different weights and the relationship of reputation compared to computational time. These analyses and our experiments revealed that the optimal values for these weights are $\alpha = 0.8$, $\beta = .2$, and $\gamma = .1$

2) Quorum Selection: Quorum members are responsible for constructing blocks, determining which tasks to perform, delegating those tasks to miners, and reaching consensus on the miners' output of a task. Due to this responsibility, it becomes critically important that the process of selecting

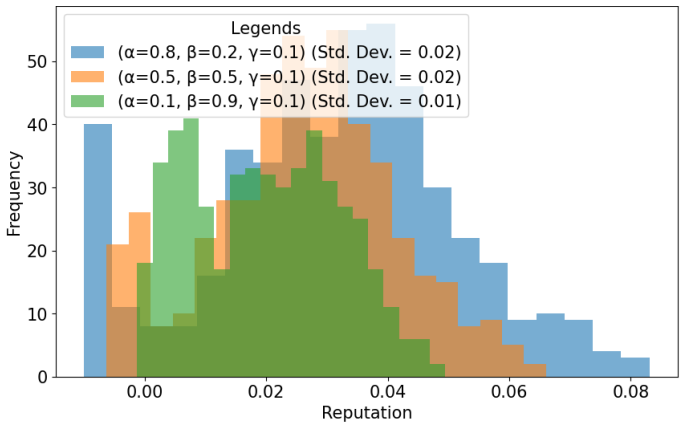

Figure 4: Reputation Distribution for Different Weight Sets. There is optimal spread for $\alpha = .8$, $\beta = .2$, and $\gamma = .1$ This means that accuracy has more impact than time- this allows low-computational systems to still earn reputation fairly.

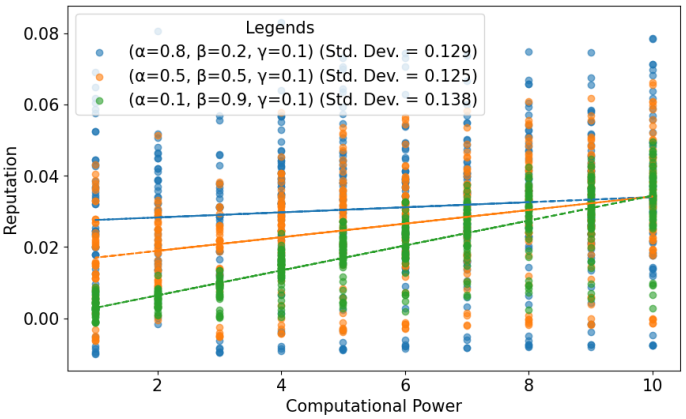

Figure 5: Reputation Against Computational Power for Different Weight Sets. There is minimal relationship between reputation and computational power for $\alpha = .8$, $\beta = .2$, and $\gamma = .1$

quorum members assures the greatest likelihood of including reliable and honest nodes in the network. It would be inappropriate to select random quorum members to handle these responsibilities. The primary objective of the reputation system is to ascertain that these quorum members have demonstrated their trustworthiness through the execution of good work, and serves as the foundation to determine which nodes are selected to participate in the quorum. We select quorum members from a subset of nodes with the highest reputation, and to ensure the same nodes are not selected as quorum members every iteration, we select a smaller random subset of these trustworthy nodes. A crutial aspect of POER is the ability to select high-reputation nodes to be responsible for reaching consensus.

Algorithm 1 outlines the pseudocode for the process of using reputation to derive a highly trustworthy quorum. First, we must obtain the reputation for every node in the network and store it in a Map with the node's unique $Address$ as the key. We sort this map in descending order (2) to get the highest-reputation nodes at the top and take a certain percentage of these to be eligible as quorum members (3). We want to find the optimal percentage of eligibility as well as the optimal percent of quorum members, so we use placeholder variables for this explanation. In our experiments, we go into more detail about choosing these values to provide maximum fairness in quorum selection. We store these new eligible nodes in a different list called eligibleNodes (4). Using Java *collections* and Java *random*, we can shuffle this list using the hash of the previous block as the seed (5). This creates a repeatable way to randomize the quorum selection. Next, another Map is created that will contain the addresses of the final selected quorum members (6). Again, quorumPercent is a placeholder value that consists of a fixed value between 0 and 1. The size of the quorum is determined by the number of eligible nodes and the percentage of quorum members we wish to choose from these nodes (7). Now that we have a shuffled, pseudo-random list of eligible nodes, we can take the first $quorumSize$ nodes in this list and add them to the quorum (8-11). With this method, we can ensure that only high-trustworthy nodes are selected as quorum members and that the same nodes are not being repeatedly selected as quorum members.

Algorithm 1 Select Quorum Members

```
1:  procedure SELECTQUORUMMEMBERS(nodes)
2:      nodes ← sortDescending(reputation)
3:      eligibleSize ← eligiblePercent × size(nodes)
4:      eligibleNodes ← first eligibleSize in nodes
5:      Shuffle(eligibleNodes, hashOfPrevBlock)
6:      Initialize new map quorum
7:      quorumSize ← quorumPercent × size(eligibleNodes)
8:      for i ← 0 to quorumSize − 1 do
9:          Add i-th entry from eligibleNodes to quorum
10:     end for
11:     return quorum
12: end procedure
```

3) Task Sharding: When the quorum reaches a decision regarding a transaction to be computed from the mempool, and if it is a WTB transaction, each quorum member takes on the task by dividing it into a unique sub-workflow consisting of distinct sub-tasks. To determine the assignment of miners to specific sub-tasks within the sub-workflow, a quorum member employs a randomization process using a seed generated by hashing their public key and the block ID together. Consequently, each sub-task within the sub-workflow is assigned to a miner. Every assigned miner is responsible for computing all the preceding sub-tasks within the given sub-workflow. For instance, the miner designated to handle sub-task 1 computes that specific sub-task and provides the corresponding input and output results to the respective quorum member. Likewise, the miner assigned to the final sub-task, sub-task n, is tasked with computing the entire sub-workflow and returning the input and

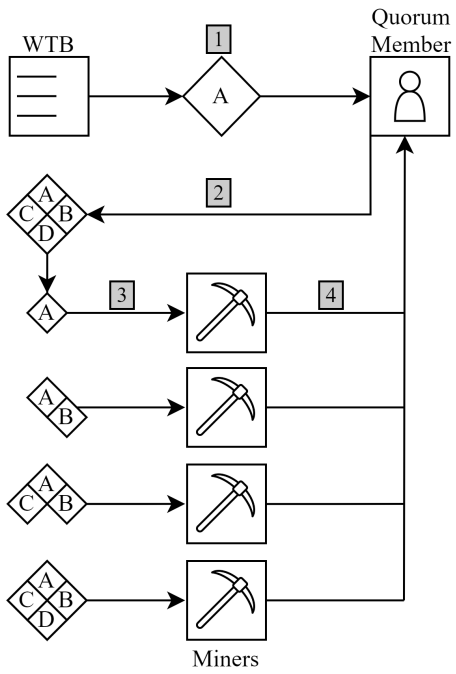

Figure 6: Sharding a workflow task into a sub-workflow. 1. The quorum member gets the task to be computed from the task block. 2. The quorum member divides a task block into a sub-workflow. 3. The quorum member assigns a miner to each of the sub-workflow tasks. 4. The miner computes their sub-task, which requires them to compute all sub-tasks that theirs is dependent on, and they return the output of all tasks they've computed to their quorum member.

output data for each sub-task back to the quorum member. This process ensures the efficient and collaborative execution of the task by the quorum members and miners, thereby facilitating the computation of complex tasks from the mempool. The utilization of randomization and unique sub-workflows enhances the security and reliability of the computation process, while the clear assignment of responsibilities ensures a well-organized and effective execution of the WTBs. This process is outlined in Figure 6.

If the miners have disagreements between inputs and outputs, the most common inputs and outputs are agreed as the correct ones. If the last two miners, those computing the most sub-tasks, have disagreements between the last of their common outputs, another miner is assigned to the maximum depth to solve the dispute. This continues as long as needed.

4) Miner Delegation: Miners are crucial to our system: they provide the computational power needed to verify tasks and ensure their repeatability. The quorum relies on these calculations to determine the most likely output of a workflow task, and this consensus is impossible without these computations. Due to the fact that quorum members are nodes with high reputation, and a node earns reputation via beneficial mining, it is essential that every node has a fair chance of being selected as a miner.

5) Miner Process: Miners are only expected to compute the workflow task assigned to them by their quorum delagatee. After a miner is contacted by one quorum member, they are locked to that quorum member so that on the off-chance that two quorum members contact the same miner, the miner is only assigned the task once. As soon as the miner receives the task to be computed by their delagatee, they should efficiently compute it and reply with their solution.

D. Workflow Modification

To support the dynamic nature of real-world scientific experiments, PRISM enables researchers to modify workflows on the ledger using Invalidation and Modification Blocks.

A use case of an invalidation block is as follows: A scientist discovers data in the ledger has been incorrectly computed. The scientist communicates with the other scientists on the workflow, and at least 50 percent of the scientists must sign on to invalidate the block. They then submit this invalidation transaction to the mempool, and a quorum will be selected and add the Data Invalidation Block to the ledger and propagate it to the network, flagging the data as invalid. Every block has hashes of two Merkle trees on them, a valid tree and an invalid tree. All workflow tasks are automatically added to the valid tree. The Data Invalidation Block added to the ledger contains updated trees, the valid tree without the invalid tasks, and the invalid tree containing the new invalid tasks. It will also contain the corrections to the incorrect computation. The workflow task that produced the invalidated data must then be recomputed. If other tasks relied on this input, they are also invalidated and must be recomputed.

A modification block is intended for use when a workflow requires adjustment in the form of addition or subtraction of workflow tasks. When a scientist realizes that the current workflow does not fully capture the importance of his experiments, he can communicate this with the other scientists on the workflow, and, similar to data invalidation blocks, must obtain 50 percent of the scientists on the workflow's signatures. The scientist then adds a modification transaction to the mempool, indicating the workflow needs to insert new tasks into the workflow for further data analysis. Upon this transaction being picked by a quorum, they add these modifications to the ledger as a Workflow Modification Block.

Such selective modifications provide researchers the flexibility to adapt workflows during experiments. The immutability of blockchain still maintains comprehensive provenance records with complete transparency into invalidation and changes.

VI. Experimental Evaluation

A. Implementation and Setup

We analyze our solution based on its performance as a simplified implementation developed atop an open-source blockchain called Bluechain. We fully implemented all features that were necessary to test PRISM. This includes reputation and reputation calculation, quorum selection, task

sharding, delegation to miners, and the addition of blocks to the blockchain. This does not include implementing scientific workflows or the different types of blocks. Task sharding was simulated with a set enumeration of sub-workflows for quorum members to randomly choose from for each block.

B. Fairness

As our main contribution is our consensus protocol, we found it important to test our method of quorum selection for fairness and to ensure equal opportunity for miners to earn reputation, while still motivating scientists to use their high-computational power machines.

1) Quorum Fairness: Quorum fairness is the fairness of the distribution of eligible quorum members across many quorum cycles. We measured this by repeatedly adding blocks to the chain and tracking who was eligible for the quorum every time. After N Iterations, we plot the distribution of how often each node was eligible for the quorum.

Observations: Figure 7 details the distribution of selection frequencies relative to varying percentages of eligibility. In our network of 104 nodes (labeled 8000-8103), we tracked the eligibility of each node to be a part of the quorum. The criterion for eligibility was being within the top 20 percent of nodes based on high reputation. The ideal situation posits that nodes with the same eligibility should be chosen an identical number of times, ensuring fair representation. Upon evaluating our quorum selection, we noticed a reasonable level of fairness amongst nodes with similar eligibility chances. For instance, nodes that were eligible to be quorum members 25 to 50 percent of the time were chosen between 8 and 9 times. Meanwhile, nodes that had an eligibility range of 0 - 25 percent were chosen between 0 and 5 times. This distribution suggests a proportionality between eligibility and selection frequency. Nodes with higher eligibility were invariably chosen as quorum members more often than those with lower eligibility rates. Notably, there were no occurrences of highly eligible nodes being overlooked for quorum membership. It is also important to note that because reputation changes as they do work, their eligibility chance will change throughout the course of adding blocks.

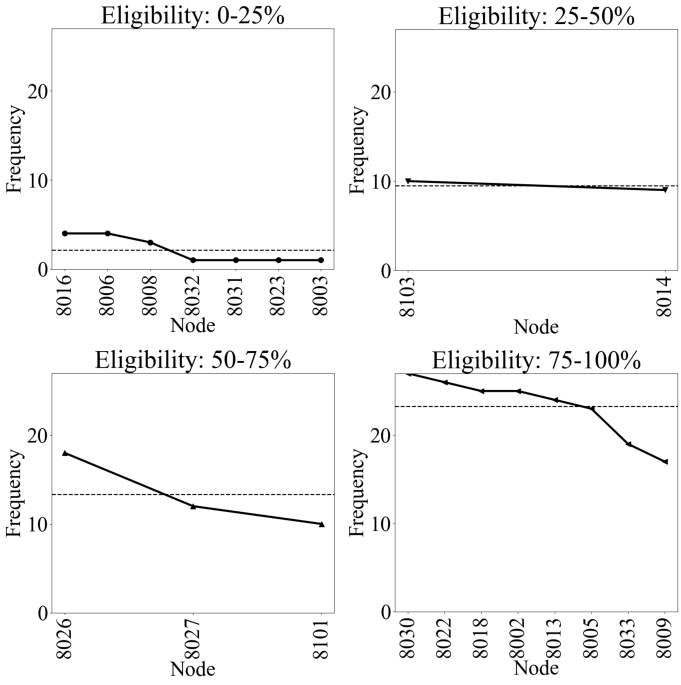

Figure 7: Fairness of node quorum selection relative to quorum eligibility. Nodes who were never eligible and thus never selected as quorum members are not shown.

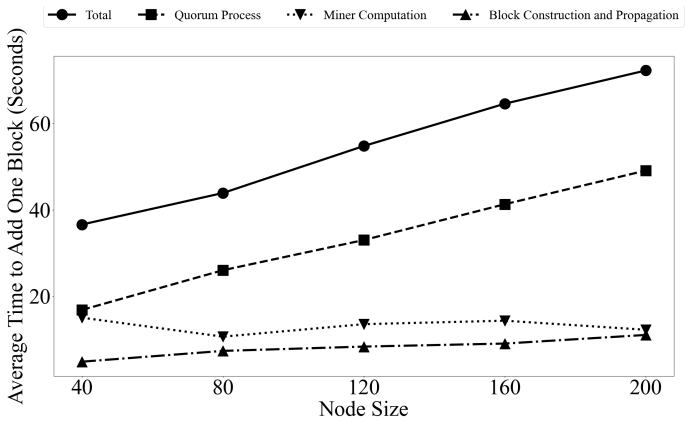

Figure 8: Scalability shown in time as the number of nodes in the network increases.

C. Scalability

Scalability is integral to any network aiming to propose a viable solution. We test the scalability of our consensus mechanism by tracking the amount of time it takes PRISM to add a block to the ledger, from the time the quorum selects the block to the time the quorum begins propagation, with different network sizes, ranging from 40 to 200. This will allow us to see how our consensus mechanism scales according to network size.

Observations: Figure 8 shows how long on average it takes for a block to get added to the ledger at different network sizes. The quorum size for this data was 10% of the network size. From Figure 8, the time is takes for blocks to be added to the ledger and the processes that must occur for that to happen scale linearly with the number of nodes in the network. Some factors such as miner computation time and quorum propagation time are constant within the system.

VII. Conclusion and Future Work

A. Conclusion

In this paper, we have presented a novel solution designed to address the challenges of storing scientific workflow provenance on a decentralized ledger. PRISM introduces a reputation-based quorum consensus protocol, where miners and quorum members play pivotal roles in ensuring the reliability and integrity of scientific workflow data. Our solution's key feature is the classification of blocks into four

distinct types: Workflow Inception Blocks (WIBs), Workflow Task Blocks (WTBs), Data Invalidation Blocks (DIBs), and Workflow Modification Blocks (WMBs). This classification enables the transparent tracking of scientific workflows on the blockchain and facilitates the identification and correction of inaccuracies through data invalidation and adaptive workflow modification. The reputation-based quorum consensus protocol ensures fair and efficient participation of nodes in the network. Quorum members, with high reputation scores, select miners for specific sub-tasks using randomization, while miners are motivated to contribute computational power to the network by earning reputation points. This dynamic and reputation-driven approach fosters a robust and secure environment for scientific workflow provenance.

B. Future Work

Future research and development for PRISM can focus on several key areas. Conducting real-world, large-scale experiments will validate scalability and further enhance its applicability. Addressing privacy concerns is essential, and exploring methods to protect sensitive data while maintaining transparency and accountability on the blockchain will be crucial. Integrating smart contracts into the system can automate and enforce aspects of workflow validation and modification, reducing reliance on manual consensus. Optimizing the reputation-based quorum consensus protocol will improve efficiency and reduce ledger overhead. Interoperability with existing scientific workflow systems and other blockchain networks should be studied to facilitate data exchange and collaboration. Decentralized storage solutions can complement PRISM, ensuring the availability and resilience of scientific workflow data. Additionally, developing more sophisticated incentive mechanisms will encourage greater participation and contribution from network nodes while ensuring fairness. By addressing these areas, PRISM can evolve into a powerful and reliable solution for storing, tracking, and validating scientific workflow provenance on decentralized ledgers, thereby enhancing the credibility and reproducibility of scientific findings in a trustless and transparent manner.

REFERENCES

[1] J. D. Vrieze, "Landmark research integrity survey finds questionable practices are surprisingly common," in *Science*, vol. Science Insider, 2021.

[2] D. Singh Chawla, "8% of researchers in dutch survey have falsified or fabricated data," in *Nature*. Nature Publishing Group, Jul 2021.

[3] R. B. Weiss, N. J. Vogelzang, B. A. Peterson, L. C. Panasci, J. T. Carpenter, M. Gavigan, K. Sartell, I. Frei, Emil, and O. R. McIntyre, "A Successful System of Scientific Data Audits for Clinical Trials: A Report From the Cancer and Leukemia Group B," *JAMA*, vol. 270, no. 4, pp. 459–464, 07 1993.

[4] W. Broad and N. Wade, *Betrayers of the Truth: Fraud and Deceit in the Halls of Science*. Simon & Schuster, 1983.

[5] S. B. Davidson and J. Freire, "Provenance and scientific workflows: Challenges and opportunities," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '08, 2008, p. 1345–1350.

[6] P. Lundquist and G. G. Dagher, "Bluechain," ISPM Research Lab, 2023. [Online]. Available: https://github.com/peytonlundquist/BlueChain

[7] R. Hoopes, H. Hardy, M. Long, and G. G. Dagher, "Sciledger: A blockchain-based scientific workflow provenance and data sharing platform," in *2022 IEEE 8th International Conference on Collaboration and Internet Computing (CIC)*, 2022, pp. 125–134.

[8] D. Fernando, S. Kulshrestha, J. D. Herath, N. Mahadik, Y. Ma, C. Bai, P. Yang, G. Yan, and S. Lu, "Sciblock: A blockchain-based tamper-proof non-repudiable storage for scientific workflow provenance," in *2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC)*, 2019, pp. 81–90.

[9] S. Malik, S. S. Kanhere, and R. Jurdak, "Productchain: Scalable blockchain framework to support provenance in supply chains," in *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, 2018, pp. 1–10.

[10] A. Al-Mamun, F. Yan, and D. Zhao, "Scichain: Blockchain-enabled lightweight and efficient data provenance for reproducible scientific computing," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 2021, pp. 1853–1858.

[11] D. K. Tosh, S. Shetty, X. Liang, C. Kamhoua, and L. Njilla, "Consensus protocols for blockchain-based data provenance: Challenges and opportunities," in *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, 2017, pp. 469–474.

[12] R. Coelho, R. Braga, J. M. N. David, M. Dantas, V. Ströele, and F. Campos, "Blockchain for reliability in collaborative scientific workflows on cloud platforms," in *2020 IEEE Symposium on Computers and Communications (ISCC)*, 2020, pp. 1–7.

[13] A. Ramachandran and D. Kantarcioglu, "Using blockchain and smart contracts for secure data provenance management," 09 2017.

[14] P. Cui, J. Dixon, U. Guin, and D. Dimase, "A blockchain-based framework for supply chain provenance," *IEEE Access*, vol. 7, pp. 157 113–157 125, 2019.

[15] K. Toyoda, P. T. Mathiopoulos, I. Sasase, and T. Ohtsuki, "A novel blockchain-based product ownership management system (poms) for anti-counterfeits in the post supply chain," *IEEE Access*, vol. 5, pp. 17 465–17 477, 2017.

[16] K. S. Loke and O. C. Ann, "Food traceability and prevention of location fraud using blockchain," in *2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC)*, 2020, pp. 1–5.

[17] J. L. Canovas Sanchez, J. B. Bernabe, and A. F.

Skarmeta, "Towards privacy preserving data provenance for the internet of things," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, 2018, pp. 41–46.

[18] U. Javaid, M. N. Aman, and B. Sikdar, "Blockpro: Blockchain based data provenance and integrity for secure iot environments," in *Proceedings of the 1st Workshop on Blockchain-Enabled Networked Sensor Systems*, ser. BlockSys'18. New York, NY, USA: Association for Computing Machinery, 2018, p. 13–18.

[19] M. S. Siddiqui, T. A. Syed, A. Nadeem, W. Nawaz, and S. S. Albouq, "Blocktrack-l: A lightweight blockchain-based provenance message tracking in iot," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 4, 2020. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2020.0110462

[20] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla, "Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability," in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, 2017, pp. 468–477.

[21] P. Abhishek, Y. Akash, and D. G. Narayan, "A scalable data provenance mechanism for cloud environment using ethereum blockchain," in *2021 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, 2021, pp. 1–6.

[22] N. Nizamuddin, K. Salah, M. Ajmal Azad, J. Arshad, and M. Rehman, "Decentralized document version control using ethereum blockchain and ipfs," *Computers & Electrical Engineering*, vol. 76, pp. 183–197, 2019.

[23] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[24] V. K, J. R, G. Kommineni, M. Tanna, and G. Prerna, "Sharding in blockchain systems: Concepts and challenges," in *2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, 2022.