SciLedger: A Blockchain-based Scientific Workflow Provenance and Data Sharing Platform

1st Reagan Hoopes*
Department of Computer Science
Utah State University
reaganhoopes@gmail.com

2nd Hamilton Hardy*
Department of Computer Science
Utah State University
hamilton.j.hardy@gmail.com

3rd Min Long
Department of Computer Science
Boise State University
minlong@boisestate.edu

4th Gaby G. Dagher
Department of Computer Science
Boise State University
gabydagher@boisestate.edu

***Abstract*—Researchers collaborating from different locations need a method to capture and store scientific workflow provenance that guarantees provenance integrity and reproducibility. As modern science is moving towards greater data accessibility, researchers also need a platform for open access data sharing. We propose SciLedger, a blockchain-based platform that provides secure, trustworthy storage for scientific workflow provenance to reduce research fabrication and falsification. SciLedger utilizes a novel invalidation mechanism that only invalidates necessary provenance records. SciLedger also allows for workflows with complex structures to be stored on a single blockchain so that researchers can utilize existing data in their scientific workflows by branching from and merging existing workflows. Our experimental results show that SciLedger provides an able solution for maintaining academic integrity and research flexibility within scientific workflows.**

***Index Terms*—Scientific Workflow; Provenance; Blockchain**

I. Introduction

A scientific workflow is a pipeline of processes conducted to reach a scientific goal, usually comprised of tasks connected by their data inputs and outputs. The provenance of a scientific workflow is the audit trail that allows for the reproducibility of scientific findings and proves the validity of a data product by computing how it came to be [1]. There are two fundamental requirements when collaborating on research projects with members at different locations: ability to capture and distribute data while also ensuring the integrity of the data collected; open data sharing.

For the first requirement, there are a variety of existing systems for managing provenance along a scientific research workflow. Systems like Dataview [2][3] use centralized cloud-based storage. Despite comprehensive data collection capabilities, data integrity can no longer be ensured if the central server is compromised. Others like Taverna [4], Kepler [5], Galaxy [6], KNIME [7], and Pegasus [8] use locally maintained storage models. While data in local storage is more secure than in cloud-based systems, having data stored locally makes collaboration among multiple institutions difficult and leaves the system vulnerable to falsification in the context of research. A survey among academic researchers found a self-reported rate of at least one instance of research fabrication or falsification between 2017 and 2020 to be estimated at around 8.3% [9]. Another investigation has been conducted recently concerning the falsification of data in an Alzheimer's research paper published in 2006 in the journal Nature. It shows that the paper in question has been cited over 2300 times and, if proven falsified, could invalidate at least a decade of research [10]. These cases illustrate critical and emerging needs for a tamper-proof way of storing scientific data provenance and but also establish a validation method among multiple parties.

For the second requirement of open data sharing that can eventually enhance the confidence scientific findings, the scientific community is striving towards greater openness on research data, code, and workflow provenance. Some government agencies such as National Science Foundation and private funders are beginning to require that researchers create plans for data management and, in some cases, establish data sharing [11].

This paper is motivated by the lack of a comprehensive solution and propose a platform that can satisfy both needs. The challenges of creating an ideal solution for scientific workflow provenance are clear due to scientific research's varying and sometimes contradictory needs. Researchers need immutability to ensure the integrity, non-reputability, and reproducibility of findings but also flexibility for adaptability in a workflow. If a researcher needs to redo a workflow task, thus invalidating an old task, an immutable system does not allow for changing or deleting records. Additionally, there are many benefits to data sharing and open accessibility, but the public nature of systems that support this openness lends itself to privacy concerns for users. Blockchains are not designed for storing large amounts of data directly on the chain, so these solutions require integrating off-chain storage.

Among various existing methods, blockchain technology with a decentralized, distributed ledger that stores the record of data processing provides a promising and reliable method for maintaining integrity of data Provenance and scientific discoveries and has been investigated recently (see Section II for

*These authors contributed equally.

details). For example, SciBlock [12] proposed a tamper-proof and non-repudiable storage for scientific workflow provenance that relies on Proof of Authority consensus. The limitations of this model comes from two facts. First, it suffers from insufficient support for data sharing and invalidation of workflow tasks through the blockchain in a realistic and complex environment, such as branching off from and merging existing workflows, as every chain houses only a single workflow. For this reason, a generic blockchain solution is usually not enough and require additional support. Second, SciBlock's invalidation mechanism invalidates all blocks before a specified execution time. If used to invalidate tasks within non-linear workflows, this method would, in many cases, result in unnecessary invalidation of scientific workflow tasks, thus requiring researchers to repeat workflow tasks needlessly.

Thus, as current research stands, there is a lack of a public, blockchain-based system catering specifically to scientific workflow provenance that allows researchers to perform complex operations on multiple workflows stored on a single blockchain.

This paper proposes SciLedger, a novel, blockchain-based solution for collecting and storing scientific workflow provenance and open data sharing. SciLedger accommodates the inclusion of multiple related and unrelated scientific workflows on a single blockchain by adding an inception block to the blockchain to indicate the beginning of each new workflow. Our public design is open access allowing for greater data sharing among researchers from existing partnerships and fostering the formation of new collaborations by branching and merging existing workflows. Additionally, SciLedger features an invalidation mechanism that allows researchers to efficiently and reliably invalidate scientific workflow tasks. Before the insertion of each new block, two separate Merkle trees are constructed. The trees commit the hashes of provenance records for the valid and invalid tasks of the given workflow separately. Both Merkle roots are then included within the provenance record that is to be included on the new block. By looking at the latest block within a workflow, users can always determine whether a provenance record is valid, invalid, or yet to be attempted. Finally, while our solution promotes accessibility for users, we also want to ensure privacy for participants in the system and that users can only perform approved actions. Using a quorum consensus mechanism and including researcher public keys on the inception block for each workflow, we ensure that while anyone can create a workflow, only authorized public keys can add workflow tasks while not requiring users to make their identities known within the system.

The contributions of this paper are as follows:

- We propose a blockchain-based solution that supports open access data sharing for scientific workflow provenance and complex workflow operations such as branching from and merging several related and unrelated scientific workflows on a single blockchain.
- We propose a novel invalidation mechanism that allows researchers to modify workflows in a way that minimizes modifications and ensures efficient verification.
- We have simulated SciLedger on a blockchain and conducted experiments to evaluate the scalability and performance of SciLedger.

The rest of the paper is organized as follows: Section II presents related works. Section III provides an overview of scientific workflows, provenance, and Merkle trees. Section IV outlines the design and capabilities of SciLedger. Experimental results are outlined in Section V and finally, our conclusions are given in Section VI.

II. Related Work

There are several works utilizing blockchain to record data provenance. Some of these works attempt to target specific fields like IoT [22][23][24], supply chain management [25][19], cloud computing [20][26], machine learning [27], and GPDR data collection compliance [18]. Others offer generic provenance collection capabilities for a variety of applications. LineageChain [15] utilizes event listeners to detect any attempts to modify data and proposes novel techniques for achieving efficient query speeds and minimizing required storage. BlockCloud [16][17] uses an approach similar to that of LineageChain for detecting data modification but also presents a consensus protocol where users stake dedicated cloud resources. ProvHL [28] is built on a private Hyperledger network and uses access control management to control specific user actions. Sifah *et al.* [29] build upon the idea of utilizing access control policies by proposing a validation mechanism where actions are contingent upon users obtaining consent from data owners. Duong and Dang [30] use a public-permissioned model to support provenance collection for open access data that can be integrated into existing open access systems. These works address the data integrity concerns of centralized solutions by taking a blockchain-based approach. However, systems designed for generic provenance applications fail to collect specific information about the data being collected as is required by scientific workflows. Additionally, except for [30], these works assume a private blockchain that eliminates the ability to preserve user privacy, meaningful validation or invalidation mechanisms, and open access data.

A variety of works propose solutions specific to scientific workflow provenance. BlockFlow [31] uses integrated event listeners for detecting data modification like Lineage Chain but builds the blockchain on top of the E-Science ECO-system. SmartProvenance [13] and DataProv [14] use threshold-based voting systems and customized smart contracts to validate provenance records based on the Open Provenance Model. Nizamuddin *et al.* [21] uses a decentralized database called IPFS to store copies of every state of data for verifying records between the blockchain and the database. SciBlock [12] introduces a timestamp-based invalidation mechanism that supports modifying workflows. Bloxberg [32] introduces a unique provenance model that includes configuration information, code, and other data specific to scientific software systems.

Table I: Comparative evaluation of main features in closely related work.

	Workflow Type		Task	Data	
Related Works	Simple	Complex	Invalidation	Open Access	Privacy-Preserving
Fernando *et al.* [12]	✓		✓		✓
Ramachandran and Kantarcioglu [13][14]	✓				✓
Ruan *et al.* [15]	✓	✓		N/A	N/A
Tosh *et al.* [16][17]	✓				✓
Neisse, Steri and Nai-Fovino [18]	✓				✓
Malik, Kenhere and Jurdak [19]	✓				✓
Liang *et al.* [20]	✓			✓	
Nizamuddin *et al.* [21]	✓	✓		✓	
Our Proposed System: SciLedger	✓	✓	✓	✓	

SciChain [33] introduces a solution optimized to support high-performance computing systems and a consensus protocol similar to that of SmartProvenance, designed to minimize communication overhead.

Table I compares SciLedger to its most closely related works and outlines the key features that set SciLedger apart. Many of these related works can sufficiently support generic provenance collection but lack the specificity needed for comprehensive solutions in scientific workflows. Even works specifically addressing scientific workflow provenance may also have limitations. SciBlock's invalidation mechanism cannot be efficiently utilized for complex workflows, as it could invalidate workflow tasks unnecessarily. Bloxberg, SmartProvenance, and DataProv do not provide the ability to invalidate workflow tasks. Finally, all of these systems lack a public blockchain that can store multiple workflows on a single blockchain, support the branching off from and merging existing workflows, and provide data sharing capabilities while maintaining privacy among users.

III. Preliminaries

A. Scientific Workflows and Provenance

Scientific workflows offer descriptions of the process for reaching a scientific goal, often expressed as a set of tasks connected by their data inputs and outputs. Provenance is the chronology of an object, including its ownership, transfer, and history. Depending on the provenance model utilized, the provenance could include several other values, including the time, location, and software used for the workflow task. In scientific workflows, provenance collection would allow users to track the changes made to a data product which is essential for project reproducibility and scientific integrity.

B. Merkle Tree

Figure 1 visualizes the structure of Merkle trees. The Merkle tree is a bottom-up constructed binary tree that uses hash pointers to commit a set of n data points while allowing for efficient verification ($\mathcal{O}(\log n)$) of membership and non-membership of a data point within the tree. The top panel specifically highlights the hash nodes and operations needed to prove that data point 8 is a member of the tree. The verifier must obtain three hash nodes (those with the red borders) and perform four hash operations (those with green borders). The last hash operation gives the root of the tree, which is called the Merkle root. If the Merkle root calculated by the verifier matches the known Merkle root, the verifier can be confident that 8 is a member of the tree in only $\mathcal{O}(\log n)$ operations. The bottom panel outlines the hash nodes and operations needed to prove that data point 4 is not a member of the tree. The closest data point above and below the data point in question must be revealed. In the case, these data points are 3 and 5. Two hash nodes (those with red borders) must be provided, and the verifier must perform five hash operations (those with green borders). If the Merkle root calculated by the verifier matches the known Merkle root, the verifier can be confident that 4 is not a member of the tree in only $\mathcal{O}(\log n)$ operations.

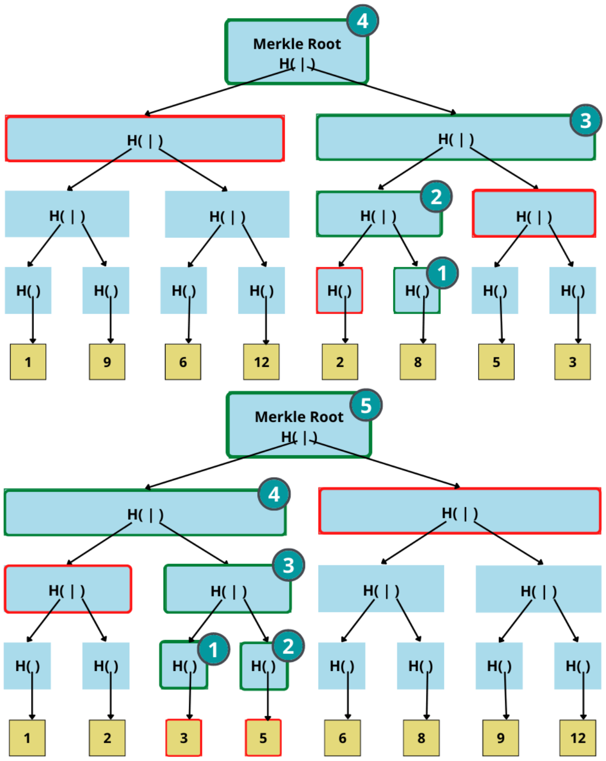


Figure 1: Membership verification using Merkle tree in $\mathcal{O}(\log n)$. *Top panel*: proving membership of data point 8. *Bottom panel*: proving non-membership of data point 4.

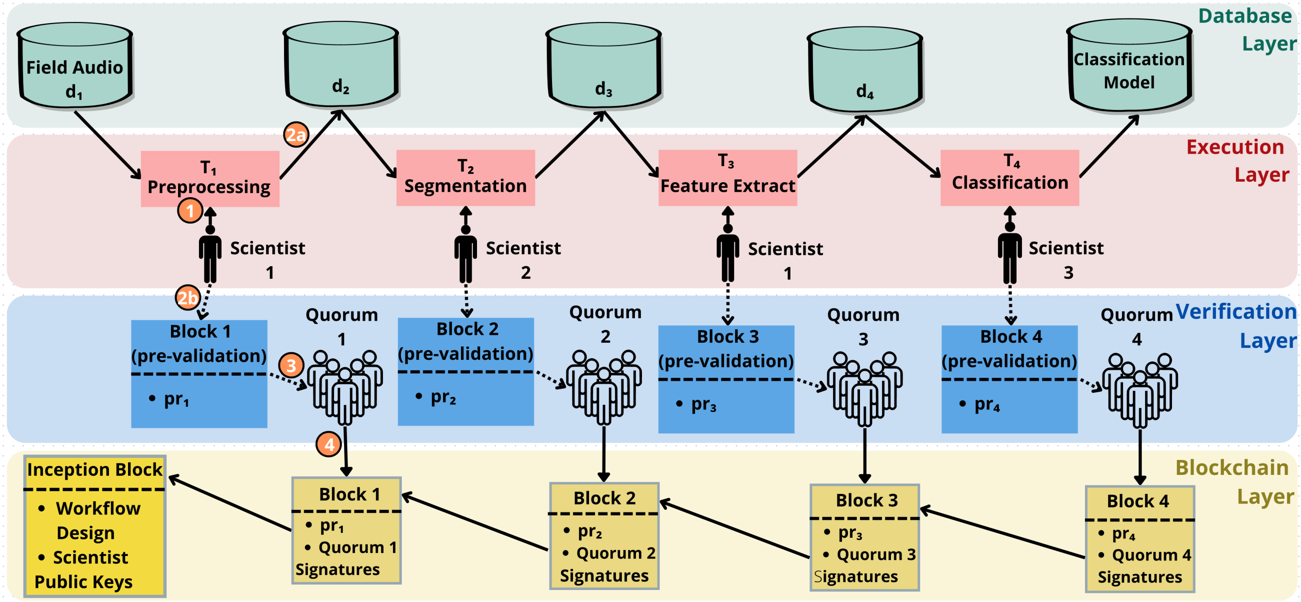


Figure 2: Schematic diagram of interworking layers in the SciLedger system, using a four-task workflow for the automated bird recognition process as an example. The workflow steps are as follows. (1) The scientist designated to perform the first workflow task, as defined in the inception block, completes the task using previously collected data as input. Upon completion, the validation and invalidation Merkle trees are constructed. A provenance record is then generated, which includes the new output data hash and the two Merkle roots. (2a) The data output, provenance record, and Merkle trees from the first workflow task are sent to be stored in the database. (2b) The scientist creates a block of the new provenance record to be submitted to the quorum. (3) The block created is sent to the selected quorum to be validated. (4) Upon successfully validating the block, the quorum members' signatures are included on the block, which is then committed to the blockchain. The data output from the workflow task is used as input for the next workflow task. Steps 1-4 are repeated until the workflow is completed and the last block is committed to the chain.

IV. SOLUTION: SciLedger

A. Solution Overview

SciLedger is a public, blockchain-based solution for scientific workflow provenance. Figure 2 visualizes the different layers of the SciLedger system utilizing a four-step workflow for the automated bird recognition process as an example.

On the execution layer of Figure 2, a four-step scientific workflow summarizing a bird recognition process [34] is visualized. The first workflow task, or T_1, uses previously recorded field audio as its data input and consists of the resampling and noise suppression of the audio. Next, the audio data d_2 resulting from T_1 is used as the data input for T_2 where audio segmentation is performed to select sections of the signal that are considered promising and eliminate segments with silence or background. Upon the completion of T_2, the updated audio d_3 is used as the input for T_3. The raw signal is transformed into a small set of representative audio features which provide compact descriptions of acoustic events of interest. Finally, T_4 classifies d_4 based on feature vectors and precomputed models, resulting in a classification model. For this example, the workflow provenance would allow users to look at any data product, such as d_4, and trace it back through each step to the origin of the data d_1.

On the blockchain layer, every block on the blockchain corresponds to a single workflow task. The relationships between the blocks correspond to the dependency-based relationships between workflow tasks. Each hash pointer stored within the provenance records of blocks points back to the block whose data output provides the given block's data input. Upon the completion or invalidation of a workflow task, the two Merkle trees that commit the provenance records of the workflow's valid and invalid tasks are reconstructed, and the new Merkle roots are included within the provenance record of the new block. We will refer to the tree committing valid records as the MT_V and its root at the R_V. Additionally, we will refer to the tree committing invalid records as the MT_I and its root at the R_I. Because of the limited storage space of the blockchain, SciLedger must integrate off-chain storage. The provenance record for each workflow task is stored within a block on the blockchain. A copy of the provenance record is stored off-chain along with the corresponding scientific data and the complete Merkle tree structures. Storing the Merkle roots on the blockchain allows users to corroborate any determination from the database on whether a provenance record is valid, invalid, or yet to be committed to the blockchain. Finally, while consensus was not the main focus of our research, the public system lends itself well to a quorum-based consensus protocol where other scientists serve as miners and validators in the system.

B. System Design

In this section, we describe the main components of the SciLedger system: Provenance (IV-B1), Off-Chain Data (IV-B2), Workflow Branching and Merging (IV-B3), and Ledger for Complex Workflows (IV-B4).

1) Provenance: SciLedger's goal is to provide a comprehensive solution for scientific workflow provenance. Provenance can include many elements, so the system design must be generic enough to be applied to various scientific research applications but also specific enough to store meaningful information regarding the scientific workflow processes. Before committing any provenance records for a workflow to the blockchain, a user must create and commit an inception block. As shown in the blockchain layer of Figure 2, the inception block contains the workflow design and scientist public keys. Workflows must be predefined, so at the start of a workflow, researchers must determine all workflow tasks and where each task is receiving and sending its data input and output, respectively. Additionally, SciLedger is designed to ensure that users can only perform approved actions within a workflow. The workflow design in the inception block specifies which scientists are approved to perform and invalidate workflow tasks. This way, when a block for a workflow task or invalidation is submitted, the quorum can determine if the user is authorized to perform this action. Once an inception block is committed to the chain, the workflow processes can begin. SciLedger stores scientific workflow provenance as provenance records. Each workflow task has an associated provenance record, the only information stored on the blockchain. SciLedger is designed so every block on the chain comprises an individual provenance record. Table II shows SciLedger's provenance records and each field they contain. The provenance record includes a unique ID for the specific task and a unique ID for the specific workflow. Together, these two values distinguish an individual provenance record from all other provenance records. The provenance record also includes the public key for the scientist who performed the task and the task execution time. These help with block validation within the quorum. The data inputs and output are hashed and included within the provenance record as well as R_V and R_I. Since the MT_V needs to commit the hashes of all valid provenance records, the hash of the provenance record for which MT_V is being constructed will contain all fields except the R_V (which is what is being created). To support custom provenance for scientists, SciLedger also allows scientists to add extra fields to the provenance records for other values they wish to be stored on the blockchain. These extra fields could represent custom workflow task relationships or a variety of text fields specific to a certain workflow.

2) Off-Chain Data: Because of limited storage capacity on the blockchain, SciLedger integrates a distributed, off-chain storage system. As shown in step 3a of Figure 2, the database receives the same provenance record that was added to the blockchain along with the data output and Merkle trees after completing a workflow task. Then, as shown in step 6 of Figure 2, the data output from a previous task is retrieved from the database and is used as a data input for the current workflow task. Because all data is stored off-chain, verification is necessary to ensure that information in the database can be trusted. The process for verification will be explained later in the system design.

Table II: Provenance Record Design

Provenance Record	
Field	**Description**
Task ID	The task's assigned identifier value
Workflow ID	The workflow's assigned identifier value
User ID	Public key belonging to the task performer
Execution Time	The task's execution time
Input Data	Hash pointer to data before modification
Output Data	Hash pointer to data after modification
R_V	Top hash for MT_V
R_I	Top hash for MT_I
Other	Extra fields for custom provenance values

3) Workflow Branching and Merging: SciLedger's design supports the inclusion of multiple workflows on a single blockchain. Figure 3 visualizes the blockchain with the corresponding blocks for four different workflows (A-D). The inception block, which has the designation ICP, is added to the blockchain before any of the blocks corresponding to workflow tasks. Since SciLedger is a public system, the ability to branch and merge research in a different direction from an existing workflow can allow researchers to avoid repeating workflow tasks that other researchers have already completed. By allowing researchers to share data, SciLedger supports a faster route to scientific discoveries. As shown in Figure 3, if a scientist wants to use the data output from workflow task T_{A_1} as input data for a new workflow, then that scientist can branch from block A_1. Workflow C is an example of this process. Additionally, a scientist can perform a merge if they want to utilize data outputs from multiple workflows as inputs for a new workflow. Workflow D is an example of merging the data outputs from workflow task T_{A_1} and workflow task T_{B_2}. Advanced operations such as branching and merging allow researchers to bypass the repetition of workflow tasks done by others and begin their workflows at the point in their research where they are performing unique tasks.

4) Ledger for Complex Workflows: An essential component of each block on the blockchain is its hash pointers. Hash pointers for each block point back to the block or blocks that supplied its data input. By giving blocks the ability to store multiple hash pointers, the blocks for a given workflow can take the form of a directed acyclic graph (DAG), allowing the blockchain to house complex workflows. A complex workflow is non-linear, meaning that workflow tasks do not have to have only one input. Figure 3 provides a visualization of the blocks corresponding to workflow B's complex workflow. Blocks B_1 and B_2 contain hash pointers to the inception block. In workflow B, the researcher may need to initially perform two separate tasks where one task does not provide input for the other. However, after completing these separate workflow tasks, their data outputs may be needed as inputs for the third workflow task, T_{B_3}. At this point, the data is merged and results in block B_3 which contains hash pointers to both B_1 and B_2. Many scientific workflow processes are non-linear and may deviate to perform unrelated tasks before combining data products in later tasks, so SciLedger is designed to accommodate complex workflows.

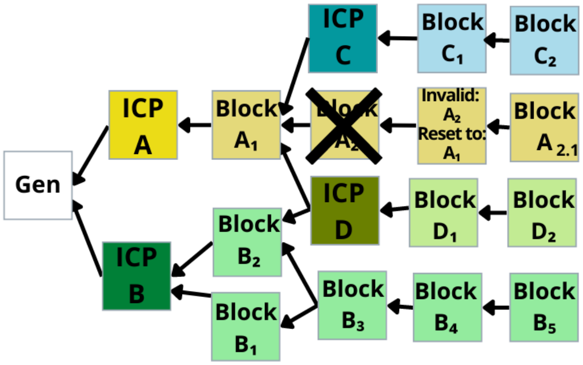


Figure 3: An illustration of the blockchain with blocks corresponding to four workflows $A - D$.

C. Invalidation

1) Purpose: To support the reproducibility of data produced from a scientific workflow, SciLedger requires scientists to pre-establish their workflow designs in the inception block. After adding a workflow task to the blockchain, scientists may need to go back and redo a workflow task. However, given the immutable nature of the blockchain, there is no way to remove or modify provenance records once added to the chain. This problem presents the need for an invalidation mechanism that works within the constraints of the immutable blockchain while also providing users the flexibility they need within scientific research processes. Allen and Mehler [35] analyzed studies using pre-established workflow designs, noting that over 60% of them concluded with null results while only 5-10% of traditional studies produced null results. They indicate that while pre-establishment curbs retroactively modifying hypotheses to fit research results, it limits the flexibility of the scientists to adapt their workflow based on the needs of the study. A goal of SciLedger is to prevent scientists from retroactively altering hypotheses to match results while avoiding the issue of null findings resulting from pre-establishing workflows. By offering an invalidation mechanism that records an immutable record of any workflow modification, we can maintain both academic integrity and research flexibility.

2) Invalidation Process: An invalidation block is added to the blockchain for an authorized researcher to invalidate a workflow task successfully. Figure 3 visualizes the blocks corresponding to workflow A. Before the invalidation of block A_2 (invalidation is indicated with an ”X”), the only blocks committed to the chain were the inception block, block A_1, and block A_2. When an invalidation must occur, the database will trace all dependencies down the blockchain from the original invalid block. Tracing dependencies allows the system to determine which blocks down the chain must also be invalidated, including blocks within the same workflow as well as blocks in other workflows. At the end of every workflow affected, an invalidation block is added. After tracing the dependencies and identifying all invalid blocks, each workflow's MT_V and MT_I are reconstructed with updated entries. The new R_V and R_I are then included on the invalidation block, which contains a hash pointer pointing back to the block in the workflow with the most recent execution time. By connecting the invalidation block to the last data state in a workflow, researchers to be sure of the most current state of their workflows and know which workflow tasks to redo accordingly. In the case of Figure 3, if block A_2 needed to be invalidated, MT_I will only commit one provenance record because A_2 did not have any dependencies committed to the chain. Once this is determined, and the Merkle roots are computed, the invalidation block is created with a hash pointer to block A_2 because it has the most recent execution time. The MT_V will commit the provenance record for A_1 as before, and MT_I will commit the provenance record for A_2. The researchers from this workflow can determine that the last good state of data is the output from A_1, so they must resume the workflow from there and redo A_2. The task IDs for workflow tasks that are redone following invalidation will be slightly altered to reflect which iteration of the task is being done. This is shown in Figure 3, where block A_2 is replaced with block $A_{2.1}$. After redoing necessary workflow tasks, researchers can continue executing additional workflow tasks as usual.

D. Block Verification

1) Merkle Tree Usage: SciLedger uses Merkle trees to confirm the validity of provenance records as determined by the database. MT_V will commit in order every valid provenance record from the given workflow and any provenance records that make up its history as a result of branching from or merging other workflows. MT_I will commit all invalidated provenance records only from the given workflow, also in order. When creating a new block, the Merkle trees will be reconstructed and used to compute the two Merkle roots that are included in the provenance record of the new block. As shown in Figure 3, workflow D utilizes data outputs from workflow A and B as input. The R_V on the last block corresponding to workflow D will commit all provenance records in workflow D as well as all of the provenance records that are a part of its history from workflows A and B, which would be the provenance records on blocks A_1 and B_2.

2) Verification Process: To verify the existence and validity of a provenance record using the blockchain, we utilize the MT_V structure from the last block of the workflow to hash the provenance records in the database and then hash pairs in order, repeatedly until we produce R_V. The R_V computed from the database can then be compared with the R_V included in the workflow's latest block. If the roots do not match, then the user knows that the database has been changed in some way, by invalidation or potentially by malicious action. To confirm that a provenance record is invalid and has not been altered due to malicious activity, we introduce the MT_I, which commits only invalidated provenance records. To verify that a record has been invalidated, we use the MT_I structure to hash invalid provenance records in the database and hash pairs to produce R_I. The resulting R_I can then be compared to the R_I contained on the latest block on the blockchain. If the roots match, the user can be sure the block has been invalidated on the blockchain rather than altered due to malicious action. Overall SciLedger's two-tree verification method ensures that users can distinguish malicious actions in the network from authorized invalidation of workflow tasks. This method preserves the scalability of the blockchain by minimizing storage overhead while ensuring the integrity of scientific workflow provenance and the corresponding scientific data.

E. Reaching Consensus in a Public Permissionless System

To support open access data, SciLedger operates as a public, permissionless system that allows researchers to share data without making their identities known to the network. This arrangement lends itself well to a quorum-based consensus. If a scientist wants to add a block to the chain, they must submit the block to the quorum with an associated smart contract. To create a quorum, we select a random group of nodes and validate a block based on the percentage of the quorum that approves the block after executing the smart contract. We have determined the ideal quorum size to be 5% of the network. Additionally, for a block to be committed to the blockchain, 70% of the quorum must approve the block. The reasoning behind these percentages is outlined in our experiments.

V. Experiments

A. Good Quorum Evaluation

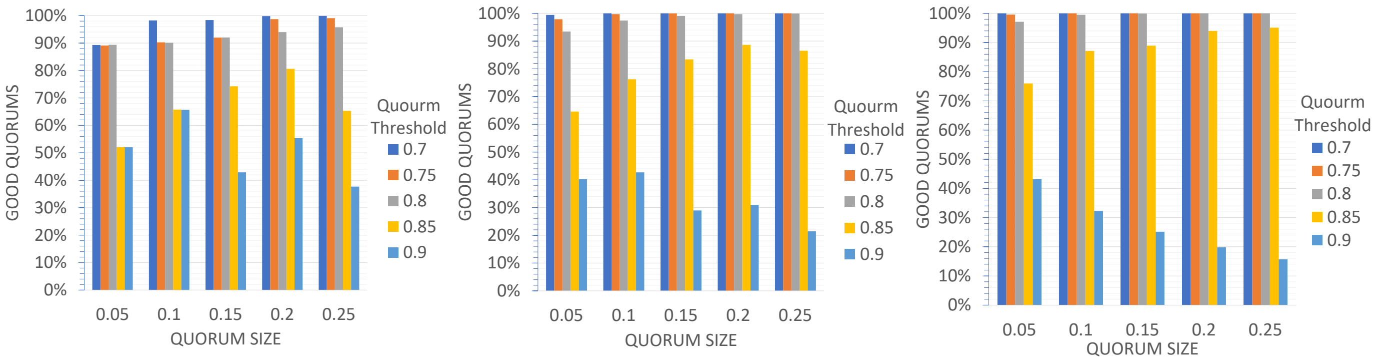
We opted to design the quorum size based on modern research's latest information concerning academic fraud. Work by Gopalakrishna *et al.* [9] analyzed a variety of anonymous surveys from Dutch researchers in all scientific fields over three years. Their conclusions determined that 8.3% of researchers fabricated or falsified data at least once during those three years [9]. These results are significantly higher than estimates from previous works by Fanelli [36] who analyzed academic fraudulency to be around 1.97% in 2009. Using Gopalakrishna *et al.* work as an upper bound while factoring in the upper limit of their confidence interval, we set a potential estimate of fraudulency in SciLedger to be approximately 12%. Using this value, we designed an experiment to algorithmically compare the percentage of good quorums among several test network sizes in Figure 4 to the size of the quorum relative to the network and the threshold a quorum needs to reach to be validated. We fix the malicious nodes in the system to be 12%. We assume the malicious nodes represent scientists looking to intentionally sabotage reaching consensus or approve bad blocks for research manipulation. A randomized Boolean list is created to model blockchain nodes with a variable size of 100, 500, and 1000 and then is used to select a variable number of entries to simulate a quorum. Within the quorum selected, the program identifies the percentage of good nodes, and if it is over the threshold, that quorum is considered good. This process is averaged 10,00 times and identifies the average percentage of good quorums achieved within that configuration.

Observations. First, we observe good quorum percentages are generally higher among a larger network size. Second, higher quorum thresholds generally decrease good quorum percentages. Third, larger overall quorum sizes generally increase good quorum percentages except in the case of a quorum threshold of 90%. In all network sizes, the percentage of good quorums with a threshold of 90% slopes downwards. Lastly, the best option where SciLedger can consistently validate correct transactions in any sized blockchain is with a quorum threshold of 70% and a quorum size of 5% of the network. This option preserves the system's scalability and reliably counteracts malicious nodes in the network.

B. Block Upload Speed

To generate sample workflows, we created a program to generate random workflows in various formations and then averaged the time required to upload the corresponding blocks. This program also combines the workflows on the blockchain to replicate SciLedger's multi-workflow capabilities and test the network speeds based on the shape of the blockchain. Randomly generating workflows ensures complex operations like branching and merging are accounted for in testing. Each workflow stores a task object which contains the workflow ID, task ID, an array with each task's parents, and the hashes of the input and output data. A Lorem Ipsum generator creates this data with a fixed data size of 2 Mb. The size of the data will only affect the performance of the SHA-256 algorithm used to hash the data. The task object also stores two Merkle trees that include hashes of all the tasks in the workflow. This information is used to construct a provenance record for the task object. The provenance record only stores the Merkle root for the Merkle tree. The provenance record is included on a block, and communication is initiated between the nodes in the network to achieve quorum consensus before it is uploaded to the simulated blockchain. The performance of this process is broken down into sections. First, the time to hash the input and output data; second, the time to construct each Merkle tree; third, the time to construct the rest of the block; and fourth, the time it takes to reach consensus within a network of varying size. Not shown here is any verification script quorum members intend to use to confirm correct data. This script's denoted α would be consistent among all network sizes but only vary depending on the number of quorum members that need to run it, meaning that its addition to this graph would be linear.

Observations. The time to hash the input and output data in different network sizes stays constant and is not affected by the network size. The Merkle tree construction time is only affected by the number of previous blocks in the network, and



(a) Network with 100 nodes.

(b) Network with 500 nodes.

(c) Network with 1000 nodes.

Figure 4: Percentage of valid quorums.

the rest of the block creation process is consistently constant time. The largest section in block upload time is the signature exchange among quorum members. Given that there is an n^2 relationship between the network size and the time it takes to reach consensus in the quorum, the overall relationship between network size and block upload time is n^2.

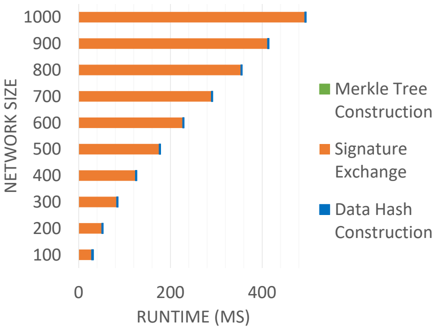


Figure 5: Block Creation Runtime.

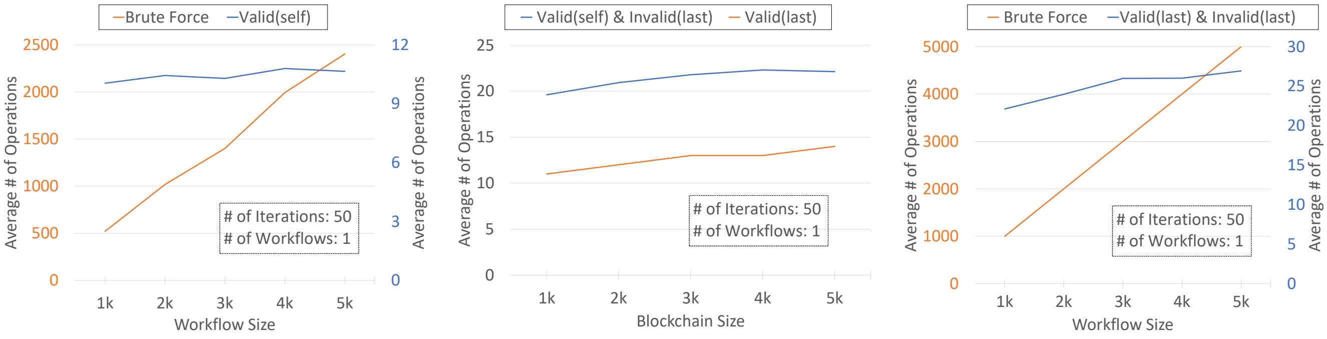
C. Provenance Record Verification Efficiency

To test provenance record verification efficiency, we create five workflows simulated on the blockchain containing 1000 to 5000 tasks, going up in increments of 1000. For each blockchain size, we select 50 random blocks and calculate the average number of operations that would be required for two different approaches to three different types of verification. Some approaches utilize the MT_V and MT_I to prove the existence or non-existence of a provenance record on the blockchain. These approaches use the number of hash operations required to obtain the Merkle root of the tree and then compare the root to the root on the blockchain. Other approaches use brute force, and the number of operations is simply the number of direct comparisons between off-chain storage and the blockchain. Figure 6a shows the average number of operations needed to verify the existence of a provenance record on the blockchain. The first approach verifies that a provenance record exists within its MT_V. The second approach is a brute force search of the blockchain until the block containing the given provenance record is found. Figure 6b shows the average number of operations needed to verify the existence and validity of a provenance record on the blockchain, meaning that the record exists and was not later invalidated. The first approach verifies the existence of a provenance record within its MT_V along with proving its non-existence within the MT_I for the last block on the chain. The second approach verifies the presence of the provenance record within the MT_V for the last block on the chain only. Figure 6c shows the average number of operations needed to verify the non-existence of a provenance record on the blockchain. The first approach is to prove the non-existence of the provenance record in both the MT_V and MT_I for the last block on the chain. The second, brute force approach iterates thorough the entire blockchain and confirms that a block containing the given provenance record does not exist anywhere on the blockchain.

Observations. To accommodate the large difference in scale between approaches, Figure 6a and Figure 6c use a secondary scale to visualize the average values of the data better. Brute force approaches for verification of existence and non-existence are significantly slower than SciLedger's Merkle tree solutions, as the number of operations is roughly the same as the number of blocks in the chain. In Figure 6b, the verification of existence and validity, the approach that verifies using only the last block's R_V was consistently faster than the approach that verifies using the R_V of the given provenance record and the R_I on the provenance record from the last block on the chain. This finding informs our decision to use only the R_V from the latest provenance record to verify existence and validity. Overall, however, the relationship between the size of the blockchain and the average number of operations to verify the existence, existence and validity, or non-existence is $\mathcal{O}(\log n)$ operations.

VI. Conclusion and Future Work

In this paper, we first identified two significant concerns facing the collection of scientific data provenance, namely



(a) Verification of Existence.

(b) Verification of Existence and Validity.

(c) Verification of Non-existence.

Figure 6: Provenance Record Verification.

academic integrity and workflow flexibility. Second, we proposed SciLedger, a blockchain-based system that supports multiple complex scientific workflows and dependency-based block invalidation. Third, we have conducted experiments that simulate SciLedger and test scalability performance and network design. Our results show that SciLedger offers a promising approach to maintaining academic integrity and research flexibility within scientific workflows.

We identify two areas for potential future research. SciLedger does not support private data that can only be viewed and used as input for new workflows by specific researchers. Future work could investigate how differential privacy in data storage supports a partially private model. SciLedger does not explicitly encourage any particular consensus protocol. Future work could investigate consensus mechanisms and novel ways of validating transactions for scientific workflow provenance.

Acknowledgement: This work is supported in part by the National Science Foundation under award number 2051127.

References

[1] 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.

[2] A. Kashlev, S. Lu, and A. Mohan, "Big data workflows: A reference architecture and the dataview system," *Services Transactions on Big Data*, vol. 4, pp. 1–19, 01 2017.

[3] A. Kashlev and S. Lu, "A system architecture for running big data workflows in the cloud," in *2014 IEEE International Conference on Services Computing*, pp. 51–58.

[4] "Taverna - apache incubator." [Online]. Available: https://incubator.apache.org/projects/taverna.html

[5] "The kepler project." [Online]. Available: https://kepler-project.org/

[6] "Galaxy community hub." [Online]. Available: https://galaxyproject.org/

[7] Aug 2022. [Online]. Available: https://www.knime.com/

[8] "Pegasus," Apr 2022. [Online]. Available: https://pegasus.isi.edu/

[9] G. Gopalakrishna, G. ter Riet, G. Vink, I. Stoop, J. M. Wicherts, and L. M. Bouter, "Prevalence of questionable research practices, research misconduct and their potential explanatory factors: A survey among academic researchers in the netherlands," *PLOS ONE*, vol. 17, pp. 1–16, 02 2022.

[10] C. Piller, "Potential fabrication in research images threatens key theory," Jul 2022. [Online]. Available: https://www.science.org/content/article/potential-fabrication-research-images-threatens-key-theory-alzheimers-disease

[11] G. Popkin, "Setting your data free," *Nature*, vol. 569, pp. 445–447, 2019.

[12] 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.

[13] A. Ramachandran and M. Kantarcioglu, "Smartprovenance: A distributed, blockchain based dataprovenance system," in *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '18, 2018, p. 35–42.

[14] A. Ramachandran and D. M. Kantarcioglu, "Using blockchain and smart contracts for secure data provenance management," 2017. [Online]. Available: https://arxiv.org/abs/1709.10000

[15] P. Ruan, G. Chen, T. T. A. Dinh, Q. Lin, B. C. Ooi, and M. Zhang, "Fine-grained, secure and efficient data provenance on blockchain systems," *Proc. VLDB Endow.*, vol. 12, no. 9, p. 975–988, May 2019.

[16] 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.

[17] D. Tosh, S. Shetty, X. Liang, C. Kamhoua, and L. L. Njilla, “Data provenance in the cloud: A blockchain-based approach,” *IEEE Consumer Electronics Magazine*, vol. 8, no. 4, pp. 38–44, 2019.

[18] R. Neisse, G. Steri, and I. Nai-Fovino, “A blockchain-based approach for data accountability and provenance tracking,” in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ser. ARES ’17, 2017.

[19] 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.

[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] 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.

[22] 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, 2018, p. 13–18.

[23] 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.

[24] M. A. Rahman, M. S. Hossain, M. S. Islam, N. A. Alrajeh, and G. Muhammad, “Secure and provenance enhanced internet of health things framework: A blockchain managed federated learning approach,” *IEEE Access*, vol. 8, pp. 205 071–205 087, 2020.

[25] 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.

[26] Y. Zhang, S. Wu, B. Jin, and J. Du, “A blockchain-based process provenance for cloud forensics,” in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, 2017, pp. 2470–2473.

[27] P. Lüthi, T. Gagnaux, and M. Gygli, “Distributed ledger for provenance tracking of artificial intelligence assets,” 2020. [Online]. Available: https://arxiv.org/abs/2002.11000

[28] A. Demichev, A. Kryukov, and N. Prikhodko, “The approach to managing provenance metadata and data access rights in distributed storage using the hyperledger blockchain platform,” in *2018 Ivannikov Ispras Open Conference (ISPRAS)*, 2018, pp. 131–136.

[29] E. B. Sifah, Q. Xia, K. O.-B. O. Agyekum, H. Xia, A. Smahi, and J. Gao, “A blockchain approach to ensuring provenance to outsourced cloud data in a sharing ecosystem,” *IEEE Systems Journal*, vol. 16, no. 1, pp. 1673–1684, 2022.

[30] T. Dang and T. Duong, “An effective and elastic blockchain-based provenance preserving solution for the open data,” *International Journal of Web Information Systems*, vol. 17, pp. 480–515, 2021.

[31] R. Coelho, R. Braga, J. M. David, F. Campos, and V. Ströele, “Blockflow: Trust in scientific provenance data,” in *Anais do XIII Brazilian e-Science Workshop*, 2019.

[32] K. Wittek, N. Wittek, J. Lawton, I. Dohndorf, A. Weinert, and A. Ionita, “A blockchain-based approach to provenance and reproducibility in research workflows,” in *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2021, pp. 1–6.

[33] 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.

[34] A. de Oliveira, T. Ventura, T. Ganchev, L. Silva, M. Marques, and K.-L. Schuchmann, “Speeding up training of automated bird recognizers by data reduction of audio features,” *PeerJ*, vol. 8, 01 2020.

[35] C. Allen and D. M. Mehler, “Open science challenges, benefits and tips in early career and beyond,” *PLOS Biology*, vol. 17, no. 5, p. 1–14, May 2019.

[36] D. Fanelli, “How many scientists fabricate and falsify research? a systematic review and meta-analysis of survey data,” *PLOS ONE*, pp. 1–11, 2009.