2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing

ProvChain: A Blockchain-based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability

Xueping Liang[1], Sachin Shetty[2], Deepak Tosh[3], Charles Kamhoua[4], Kevin Kwiat[4], and Laurent Njilla[4]
[1] College of Engineering, Tennessee State University, Nashville, TN 37209
[2] Virginia Modeling Analysis and Simulation Center, Old Dominion University, Norfolk, VA 23529
[3] Department of Computer Science, Norfolk State University, Norfolk, VA 23504
[4]Cyber Assurance Branch, Air Force Research Laboratory, Rome, NY 13441
xliang@tnstate.edu, sshetty@odu.edu, dktosh@nsu.edu,
{charles.kamhoua.1, kevin.kwiat, laurent.njilla}@us.af.mil

***Abstract*—Cloud data provenance is metadata that records the history of the creation and operations performed on a cloud data object. Secure data provenance is crucial for data accountability, forensics and privacy. In this paper, we propose a decentralized and trusted cloud data provenance architecture using blockchain technology. Blockchain-based data provenance can provide tamper-proof records, enable the transparency of data accountability in the cloud, and help to enhance the privacy and availability of the provenance data. We make use of the cloud storage scenario and choose the cloud file as a data unit to detect user operations for collecting provenance data. We design and implement ProvChain, an architecture to collect and verify cloud data provenance, by embedding the provenance data into blockchain transactions. ProvChain operates mainly in three phases: (1) provenance data collection, (2) provenance data storage, and (3) provenance data validation. Results from performance evaluation demonstrate that ProvChain provides security features including tamper-proof provenance, user privacy and reliability with low overhead for the cloud storage applications.**

***Keywords*-Data provenance, Blockchain, Cloud Computing, Privacy, Reliability, Blockchain Cloud.**

I. Introduction

Cloud computing is widely adopted by commercial and military environment to support data storage, on demand computing and dynamic provisioning. Cloud computing environments are distributed and heterogeneous with a diversity of software and hardware components which are provided by different vendors, possibly introducing risks of vulnerabilities and incompatibility. The security assurance of intra-cloud and inter-cloud data management and transfer arises as a key issue. Cloud auditing can only be effective if all operations on the data can be tracked reliably. Provenance is a process that determines the history of a data product, starting from its original sources [1]. Assured provenance data can help detect access violations within the cloud computing infrastructure. However, developing assured data provenance remains a critical issue for cloud storage applications. Besides, provenance data may contain sensitive information about the original data and the data owners. Hence, there is a need to secure not only the cloud data but also ensure integrity and trustworthiness of provenance data. State-of-the-art cloud based provenance services are vulnerable to accidental corruption or malicious forgery of provenance data [2].

Blockchain technology has attracted interest due to a shared, distributed and fault-tolerant database that every participant in the network can share the ability to nullify adversaries by harnessing the computational capabilities of the honest nodes and information exchanged is resilient to manipulation. Blockchain network is a distributed public ledger where any single transaction is witnessed and verified by network nodes. Blockchain's decentralized architecture can be leveraged to develop an assured data provenance capability for cloud computing environment. In decentralized architecture, every node participates in the network for providing services, thereby providing better efficiency. Availability is also ensured because of blockchain's distributed characteristics. Since a centralized authority is frequently used in cloud services, there is a need to safeguard the personal data while maintaining privacy. With blockchain based cloud data provenance service, all data operations are transparently and permanently recorded. Thus, the trust between users and cloud service providers can easily be established. Furthermore, maintaining provenance can assist in improving the trust of cloud users toward cyber-threat information sharing [3] [4] to enable proactive cyber defense at a reduced security investment [5] [6].

In this paper, we present ProvChain, a blockchain based data provenance architecture to provide assurance of data operations in a cloud storage application, while enhancing privacy and availability at the same time. ProvChain records the operation history as provenance data which will be hashed into Merkle tree nodes [7]. A list of hashes of provenance data will constitute a Merkle tree and the tree

Approved for Public Release; Distribution Unlimited : 88ABW-2016-6385 Dated: 09 Dec 2016.

978-1-5090-6611-7/17 $31.00 © 2017 IEEE
DOI 10.1109/CCGRID.2017.8

root node will be anchored to a blockchain transaction. A list of blockchain transactions will be used to form a block and the block needs to be confirmed by a set of nodes in order to be included in the blockchain. An attempt to modify a provenance data record will require an adversary to locate the transaction and the block. Blockchain's underlying cryptographic theory will allow to modify a block record only if the adversary can present a longer chain of blocks than the rest of miners' blockchain, which is quite difficult to achieve. By leveraging the global-scale computing power of blockchain network, the blockchain based data provenance can provide integrity and trustworthiness. In our architecture, we keep the hashed identity of users in order to protect their privacy from rest of the nodes in blockchain network.

The rest of the paper is organized as follows. Section II provides an overview of the state-of-the art data provenance efforts and blockchain technology. Section III describes the design of ProvChain, our blockchain based data provenance architecture. The detailed implementation is given in Section IV. Performance evaluation of ProvChain is presented in Section V. Finally, we conclude in Section VI.

II. Background and Related work

A. Data provenance

Data provenance is very critical for cloud computing system administrators to debug break-ins to the system or network. Cloud computing environments are typically characterized by data transfers between diverse system and network components. These data exchanges could take place within a data center or across federated data centers. The data does not usually follow the same path due to multiples copies of the data and diversity of paths taken to ensure resilience. This design adds degree of difficulty for administrators to accurately identify the origin of attack, what software and/or hardware components caused the attack, and the impacts of the attack. Security violations needed to be identified at a fine granularity and provenance can assist. Current state-of-the art provenance systems in the cloud support the above tasks through logging and auditing technologies. These technologies are not effective in cloud computing systems, which are complex in nature, due to several layers of interoperating software and hardware components spread across geographical and organizational boundaries. To identify the origin, cause and impact of security violations in cloud infrastructures will require collection of forensics and logs from the diverse and disparate sources which is an insurmountable task. At the same time, logs only provide a sequential history of actions related to every application. The provenance data provides the history of the origins of all changes to a data object, list of components that have either forwarded or processed the object and users who have viewed and/or modified the object and has enhanced requirements for assurance.

Researchers have presented several data provenance related efforts. PASS is the first scheme to address the collection and maintenance of provenance data at the operation system level [8]. A file provenance system [9] is proposed to collect provenance data by intercepting file system calls below the virtual file system, which requires changes to operating systems. For cloud data provenance, S2Logger [10], was developed as an end to end data tracking tool which provides both file-level and block-level provenance in kernel space. In addition to data provenance techniques and tools, the security of provenance data and user privacy has also been explored. Asghar *et al*. [11] proposed a secure data provenance solution in the cloud, which adopts two-folder encryption method to enhance privacy albeit at a higher computation cost. SPROVE [12] protects provenance data confidentiality and integrity using encryption and digital signature. However, SPROVE does not possess provenance data querying capability. Progger [13] is a kernel-level logging tool which can provide log tamper-evidence at the expense of user privacy. There are also efforts which use provenance data for managing cloud environment, such as, discovery of usage patterns for cloud resources, popularized resource reuse and fault management [14].

B. Blockchain

Blockchain technology has attracted tremendous interest from wide range of stakeholders, which include finance, healthcare, utilities, real estate and government agencies. Blockchains are shared, distributed and fault-tolerant database that every participant in the network can share, but no entity can control. The technology is designed to operate in a highly contested environment against adversaries who are determined to compromise. Blockchains assume the presence of adversaries in the network and nullify the adversarial strategies by harnessing the computational capabilities of the honest nodes and information exchanged is resilient to manipulation and destruction. The reconciliation process between entities is sped up due to absence of trusted central authority or intermediary. Tampering of blockchains are extremely challenging due to use of a cryptographic data structure and no reliance of secrets. The blockchain networks are fault tolerant which allows nodes to eliminate compromised nodes. Despite this, there are several vulnerabilities exist [15], which could potentially disrupt the integrity of blockchain. However, it requires the malicious node to have enormous computational power to conduct attacks, which may not be even cost worthy.

The decentralization and security characteristics of blockchain have attracted researchers to develop various applications such as smart contracts, distributed DNS, and identity management etc. Besides Bitcoin, Ethereum [16] is also designed on top of public blockchain for simple and quick development of decentralized applications with per-address transaction model. Multichain [17] provides

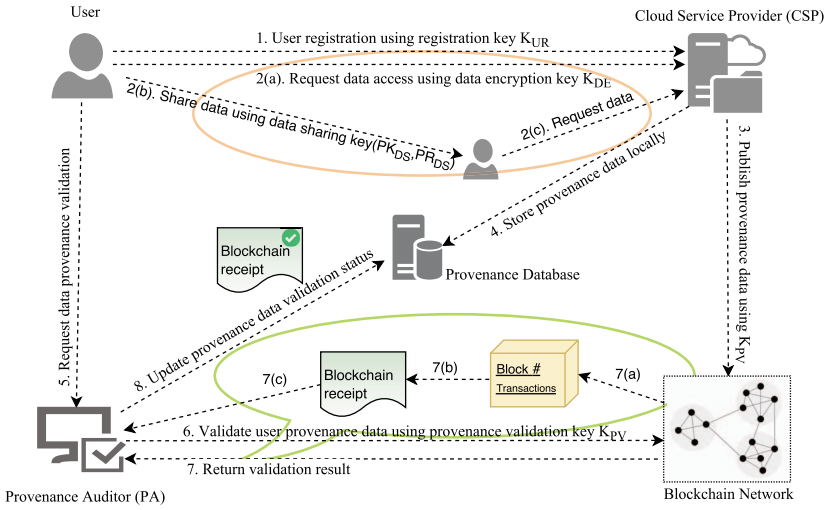


Figure 1: ProvChain System Interaction.

an open-source permissioned blockchain network, where developers can host their blockchain on a private cloud architecture. Multichain uses per-output transaction model and can handle high throughput [18]. Tierion [19] provides a platform for uploading and publishing data records into the Blockchain network. With public APIs available, Tierion is convenient for integrating applications that demand need of blockchain. Developers can post metadata using HTTP request into Tierion data store and fetch record information. Each data record has a record ID which can be used to retrieve the blockchain receipt generated based on the blockchain transactions. The blockchain receipt contains the transaction ID which will be used to locate a transaction and the block that hosts the transaction. In this way, the data record posted on the blockchain cannot be tampered and the integrity is assured.

The Blockstack Labs from Princeton University proposed a decentralized PKI service on top of Namecoin and a blockchain based naming and storage system [20]. Blockchain application in information-centric network for name based security of content distribution has also been proposed [21]. Enigma is a decentralized computation platform with guaranteed privacy which uses blockchain network to control the network, manage access control and identity, and create tamper-proof log of events [22]. Guardtime provides industrial-scale blockchain services using Keyless Signature Infrastructure (KSI) and secure one-way hash function, which is quantum-immune in contrast to RSA [23]. Guardtime also proposed a blockchain standard for digital identity and a protocol for authentication and digital signature which provides a simplified mechanism for revocation management and long-term validity [24].

III. ProvChain Architecture

ProvChain is a data provenance architecture built on a blockchain which will provide the ability to audit data operations for cloud storage. ProvChain achieves the following four objectives.

- **Real-time Cloud Data Provenance**: User operations are monitored in real time to collect provenance data, which will further support access control policy enforcement [25] and intrusion detection.
- **Tamper-proof Environment**: Data provenance record is collected and then published to the blockchain network which protects the provenance data. All data on the blockchain is shared among the nodes. ProvChain builds a public time-stamped log of all user operations on cloud data without the presence of a trusted third party. Every provenance entry is assigned a blockchain receipt for future validation.
- **Enhanced Privacy Preservation**: Data provenance record is associated with a hashed user ID to preserve privacy so that blockchain network node cannot correlated data records associated with a specific user. Provenance auditor can access provenance data owned by the user but can never identify the true owner. Only the service provider can link each record with the owner of the record data.
- **Provenance Data Validation**: Data provenance record is published globally on blockchain network, where a number of blockchain nodes provide confirmation for every block. ProvChain uses blockchain receipt to validate every provenance data entry.

To achieve the above objectives, we adopt the below methods to design ProvChain's architecture.

- Monitor user activities in real time using hooks and listeners so that every user operation on files will be collected and recorded for generating provenance data.

Table I: Structure Definition of a File Provenance Entry

RecordID	DateTime	Username	Filename	AffectedUser	Action	TxHash	BlockHash	Validation
001	2016-9-7 10:00:00	A	X	None	Create at CloudA/A	m-bits	n-bits	True
002	2016-9-7 11:00:00	A	X	None	Copy to CloudB/A/backup	m-bits	n-bits	True
003	2016-9-7 12:00:00	A	X	None	Read at CloudA/A	m-bits	n-bits	True
004	2016-9-7 13:00:00	A	X	B	Share to User B	m-bits	n-bits	True
005	2016-9-7 14:00:00	A	Y	None	Write to CloudA/Y	m-bits	n-bits	True
006	2016-9-9 11:00:00	B	Z	A	Copy from CloudA/A to CloudB/B	m-bits	n-bits	True
007	2016-9-9 12:00:00	B	Z	A	Read at CloudB/B	m-bits	n-bits	True
008	2016-9-9 13:00:00	B	Z	A	Write at CloudB/B	m-bits	n-bits	True
009	2016-9-9 15:00:00	A	X	B	Share X to public	m-bits	n-bits	True
010	2016-9-9 14:00:00	B	Z	None	Delete from CloudB/B	m-bits	n-bits	True

- Store all hashed data operations in form of blocks in the blockchain. Every node on the blockchain can verify the operations by mining the block so that data provenance is authentic and tamper-proof.
- Hash the user ID while publishing provenance data so that the blockchain network and the provenance auditor cannot determine user identity and the data operations.
- Provenance auditor validates provenance data by retrieving transactions from the blockchain network by using blockchain receipt which contains block and transaction information.

A. Architecture Overview

An overview of ProvChain architecture is illustrated in Figure 1. Following are the critical components of ProvChain.

- **Cloud User**. A user, who owns its data and has sharing relationship on other users' data, may opt for the provenance service, where the provenance data is stored on the public blockchain. Data changes made by the user can be monitored and validated by blockchain nodes, but the nodes may not know about details of other users' activities. The provenance data will not expose real user identity.
- **Cloud Service Provider (CSP)**. The cloud service provider offers a cloud storage service and is responsible for user registration. A CSP can benefit from our system in the following aspects. First, they can audit the data changes all the time, and they can learn a lot about data operations performed by all the users to better improve their service. Through provenance data, they can also detect intrusion from anomalous behaviours. Besides, they can protect their own daily records just like normal users. As far as business aspects, they can gain brand reputation from using blockchain provenance services since they provide trustworthiness.
- **Provenance Database**. The provenance database records all provenance data on the blockchain network, which is used for detecting malicious behaviors. All data records are anonymized.
- **Provenance Auditor (PA)**. PA can retrieve all the provenance data from the blockchain into the provenance database and validate the blockchain receipt. The PA maintains the provenance database but cannot correlate the provenance entry to the data owner.
- **Blockchain Network**. The blockchain network consists of globally participating nodes. All the provenance data will be recorded in form of blocks and verified by blockchain nodes.

B. Preliminaries and Concepts

ProvChain uses cloud file as data unit and monitors file operations to provide data provenance service. After each file operation is detected, a provenance entry will be generated. The cloud service provider will upload the provenance entry onto the blockchain network. In this section, we describe the details on file provenance use case and block structure.

File Provenance Use Case. For each file provenance, we can record activities, such as, file creation, file modification, file copy, file share and file delete. Examples are shown in Table I. A file can be created by user A, which refers to origin of file X. Then user A copies file X to another location, probably for backup or other reasons. The read and write operation of user A on file X can also be recorded. If a user B asks for sharing file X from user A, there will also be a record both on user A and user B. User A shares the file X at pre-defined location and user B creates a new file Y from the shared file X. Then user B can operate on file Y just the same as user A on file X, such as read and write operations. If user B deletes the file, there will be a record for deletion. At some point of time, user A decides to make the file X public so that the file access is changed. Anyone accesses it will also create a new file at their own respective location. History of files (different versions of file) can be backed up for future use.

Block Structure. ProvChain uses blockchain network to provide data record verification and resist against tampering. The block structure is composed of two parts, block header and a list of transactions. The main attributes in the header are block hash, height, confirmations, nonce and Merkle root. Block hash is computed using the previous block hash and a nonce. The height represents the block index in the global blockchain network. The confirmation number of the block indicates the number of nodes that have mined

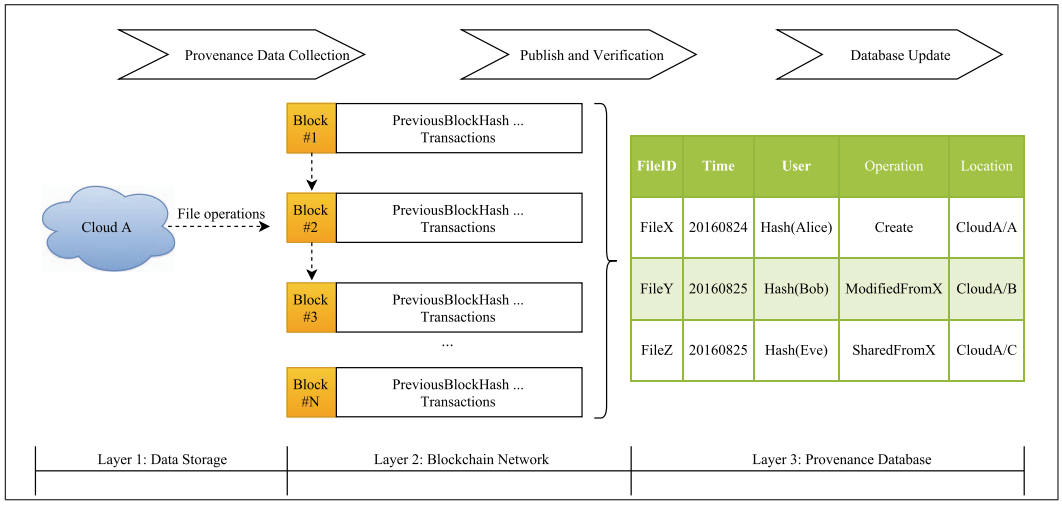


Figure 2: ProvChain System Architecture.

this block and the nonce is used by blockchain nodes to check the integrity of the block. The Merkle root is the root of binary hash tree created out of all the transactions in a block. Transaction lists come after the block header. Each transaction has a hash, with inputs and outputs. In ProvChain, each data record is hashed into a Merkle tree node. The Merkle tree root node will be anchored to one transaction in a certain block.

C. Threat Model

Here, we analyze the potential vulnerabilities in ProvChain. The cloud service provider offers data provenance service as well as cloud storage service, which allow user to store data on the cloud platform and provide the option to enable the data provenance service. The cloud service provider cannot guarantee that data records will remain unchanged due to known vulnerabilities in hypervisors and cloud operating systems. Once the data provenance service is enabled, the user will be able to trace the data and the provenance auditor is allowed to access all the provenance data. However, the provenance auditor cannot be completely trusted. The adversary can potentially access or modify user data and/or user provenance data. Since ProvChain's main objective is to protect provenance data, we assume that user data stored on the cloud is encrypted and is not accessible to anyone without the decryption key.

D. Key Establishment

To use ProvChain, users are required to register the service and create their credentials. For cloud storage applications, users generate data encryption key pairs to encrypt their cloud data for confidentiality. If the user wants to share a file, a data sharing key will be provided. For provenance data, cloud service provider generates key pairs to encrypt provenance data for privacy considerations, because provenance data will be further uploaded and published to the blockchain network. We describe each of key as follows.

- **User Registration Key** K_{UR}. In ProvChain, user needs to register the cloud storage service to store data on the cloud. We denote the key as K_{UR}. Every time user wants to operate on cloud data, the registration key is needed.
- **Data Encryption Key** K_{DE}. After registration, the user generates an encryption key K_{DE}, for encrypting all the data stored in the cloud. When a file is created, user has the option to encrypt the file, which limits the file access only to key holders.
- **Data Sharing Public/Private Key Pair** (PK_{DS}, PR_{DS}). For data sharing, a public/private key pair will be generated, denoted as (PK_{DS}, PR_{DS}). For common cases, the private key is used to generate a signature from the owner, while the public key is used by others to verify the data ownership. When users share the data with others, they share the private key for data ownership changes.
- **Provenance Verification Key** K_{PV}. Each block on the blockchain holds several provenance data entries and provenance data entry is produced upon detection of a file operation. Every data operation will trigger the cloud service provider to generate a key K_{PV} to encrypt the provenance data. The key will be shared with PA if the user assigns a provenance auditor to audit the provenance data.

IV. ProvChain Implementation

The implementation of ProvChain is conducted using a three layer architecture, comprising of data storage layer, blockchain layer, and provenance database layer, as in Figure 2. The functions for each layer are described as follows.

- **Data Storage Layer**. ProvChain is implemented to support cloud storage applications. Here we use one

cloud service provider but our architecture can be scaled to multiple providers.

- **Blockchain Network Layer**. We use blockchain network to record each provenance data entry. Each block can record multiple data operations. Here we use file as a data unit, so we record each file operation with username and file name. File access operations include Create, Share, Change and Delete.
- **Provenance Database Layer**. We build an extended database locally for recording the file operation as well as querying. In ProvChain, the service provider can assign a provenance auditor to verify the data from the blockchain network. The response is a blockchain receipt that gets validated and appended in the database.

There are three phases in the life cycle of data provenance for ProvChain, namely, provenance data collection, provenance data storage, and provenance data validation.

A. Provenance Data Collection and Storage

Once user performs actions on the data files stored in the cloud, the corresponding operations get recorded. The operation can be denoted in a metadata, including all the attributes mentioned in Table I. Note for this phase, only RecordID, Date and Time, Username, Filename, AffectedUser, and Action attributes are recorded. The transaction hash, block hash and validation field will be collected after provenance auditor queries the blockchain network. The AffectedUser attribute is considered in two cases. One is data modification in which the same user is operating on the data, using the data encryption key, where there are no affected users other than the user itself. The other case is data sharing, where user shares a file with someone else. In second case, the attribute, AffectedUser, in the file operation metadata, will include all the users in the sharing group.

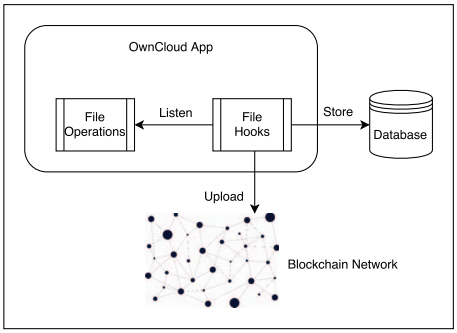


Figure 3: Provenance Data Collection and Storage.

ProvChain is built on top of an open source application called ownCloud [26] to collect the provenance data. OwnCloud provides both web-based cloud storage services and desktop client, similar to Dropbox and Google Drive, which provides user control of personal data and universal file access to all of the data seamlessly. Besides, ownCloud is flexible and developers can utilize their functions to develop various applications on top of it. In order to collect provenance data, we use hooks to listen to file operations in ownCloud web interface. After an operation is monitored, the record is generated, which will be uploaded to the blockchain network and stored in the provenance database. Figure 3 shows the architecture of our provenance data collection and storage.

For provenance data storage, we use Tierion API [19] to publish data records to blockchain network. We take file change operation as an example to demonstrate the original provenance data in JSON format as follows.

```
{
"app":"files",
"type":"file_changed",
"affecteduser":"test",
"user":"test",
"timestamp":"1475679929",
"subject":"changed_self",
"message":"",
"messageparams":"[]",
"priority":"30",
"object_type":"files",
"object_id":"142",
"object_name":"66.txt",
"link":"/apps/files/"
}
```

For privacy consideration, ProvChain hashes user name. In that case, the provenance auditor cannot know which user each provenance data belongs to. Only the service provider can relate each user with the hashed user name since the provider keeps a list of user names. ProvChain also keeps the provenance data in a local provenance database for further update and validation.

For publishing data records to blockchain network, we adopt Chainpoint standard [27]. Chainpoint proposes a scalable protocol for publishing data records on the blockchain and generating blockchain receipts. According to Chainpoint 2.0, data records are hashed so that each Merkle tree can host a number of records, as is shown in Figure 4. The target hash of the specific record and the path to the Merkle root constitute the Merkle proof of the provenance data. The Merkle root for each Merkle tree is related to one transaction in the blockchain network.

B. Provenance Data Validation

To validate the data records that are published in the blockchain network, the provenance auditor requests the blockchain receipt via Tierion API. The blockchain receipt contains information of the blockchain transaction and the Merkle proof used to validate the transaction. Figure 4 is a sample blockchain receipt. We reconstruct the Merkle tree from the blockchain receipt. Each provenance record

Figure 4: Blockchain Receipt and Merkle Tree.

is stored along with other records in the blockchain network as a transaction, which is accessible in blockchain Block Explorer [28]. Since the transaction attribute height represents the block index, we can find the exact block information as well. Both information are shown in Figure 5. Algorithm 1 is used to validate the blockchain receipt by the provenance auditor. In the algorithm, the proof, merkleRoot and targetHash in the blockchain receipt are inputs and the output is a validation result. If true is returned, then the data record is validated based on the fact that the transaction and block is authentic. If false is returned, it means the block is tampered and the data record is forged. Note all the hashes are handled in binary format. The anchors in the receipt indicates how the data record is anchored.

After the validation of the blockchain receipt, the provenance auditor can update the data record in the provenance database by filling in the remaining attributes including transaction hash, block hash and validation result. If the validation result is true, then the provenance auditor can make sure that the provenance data is authentic. If the result is false, then the provenance auditor will report to service provider that a tamper has happened.

Algorithm 1 Blockchain Receipt Validation Algorithm

procedure VALIDATE(*proof, merkleRoot, targetHash*)
  nodeNum ← number of Merkle tree nodes in *proof*
  $h \leftarrow$ *targetHash*
  $i \leftarrow 0$
  while $i <$ *nodeNum* **do**
    if *proof*$(i).key = right$ **then**
      $h \leftarrow$ *hash(h + proof(i).value).*
    else
      $h \leftarrow$ *hash(proof(i).value + h).*
    end if
    $i \leftarrow i + 1$
  end while
  if $h =$ *merkleRoot* **then return** true
  return false
  end if
end procedure

V. Evaluation

A. Summary of ProvChain's capabilities

Prior to providing the performance evaluation of ProvChain, we summarize the capabilities.

- ProvChain provides a real-time auditing for all data access in the cloud storage application. We use file as a data unit and all the operations on the cloud data objects are audited as well as recorded using blockchain. In this way, evidence for all cloud data access events can be collected and monitored.
- For each of the access record, we transform the provenance data and upload the record to the blockchain network. By doing so, we create an unalterable fingerprint of file operations, with secure and permanent record keeping as well as tamper-proof timestamp. Any changes to the blockchain will be detected by validating the blockchain receipt. Once the data record is published, no one can maliciously rewrite or alter the records without exposure.
- By utilizing blockchain network, we reduce the need for trust. There is no need to trust the owner of the remote computers involved in the blockchain network, thus removing the requirement for a trusted third party. Even the cloud service provider is not trusted for keeping the provenance data record. With decentralization, data records are confirmed and validated by continual system cross checking among computing nodes. Besides, the decentralized method ensures the integrity of data records and each of the data record has a copy with each node in the blockchain network, thereby resisting against any DDoS attack. Besides, there is no single point failure problem since no single machine holds all the data record.

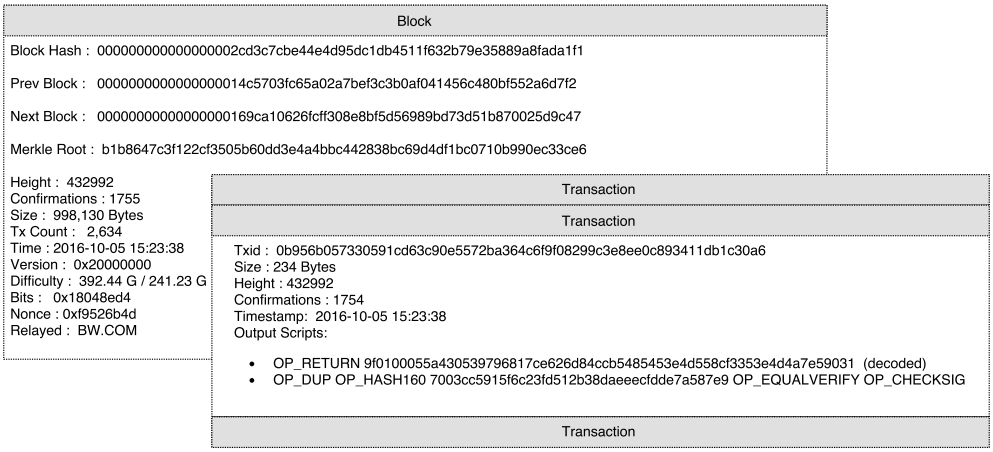


Figure 5: Transaction and Block Information.

- Users can subscribe to the data provenance service while preserving their privacy. User access records are anonymized in the blockchain network. The provenance auditor cannot learn user activities. Anonymity is preserved in two aspects. For one hand, user identity will not be linked to provenance data entries since the user ID is hashed. For the other hand, the unlinkability between each user is also achieved, especially for provenance of shared data.

B. Performance and Overhead

For provenance collection, we use Apache Jmeter [29] to assess the performance of the provenance enabled ownCloud application. We use file create operation as a use case for our performance evaluation. The evaluation for other file operations follow the same procedures. We perform file create with random file names and file contents for 500 repetitions in Jmeter [30]. The file size ranges from 1KB to 2MB. Figure 6 shows the average response time of both provenance enabled ownCloud and non provenance ownCloud. Provenance service brings an average of 6.49% of total overhead against original ownCloud application in terms of the response time, which is acceptable considering the security features it provides. Besides, with the file size increases, the overhead is generally not as much as it is when the file size is smaller, since the larger file size is, the more time will be spent on transmitting the file itself and the less time for provenance service.

Figure 7 shows the throughput for both original ownCloud 7(a) and provenance enabled ownCloud 7(b). We choose 64KB as the file size to assess the performance where only one server is responsible for the provenance service regardless of the production environment which comprises of a web server and services for load balancing and network flow optimization. The results show that both systems have the same amount of traffic received, however there is a difference in amount of traffic sent. The provenance enabled ownCloud has a comparable transaction rate as depicted in

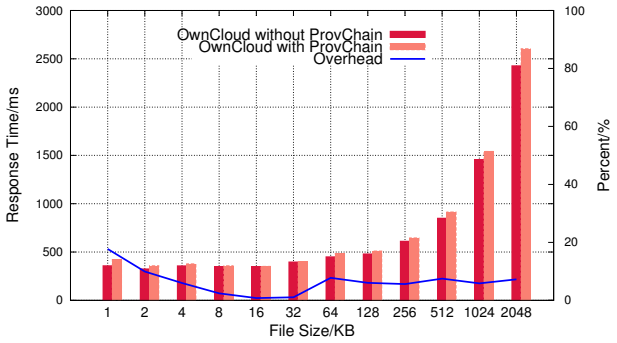
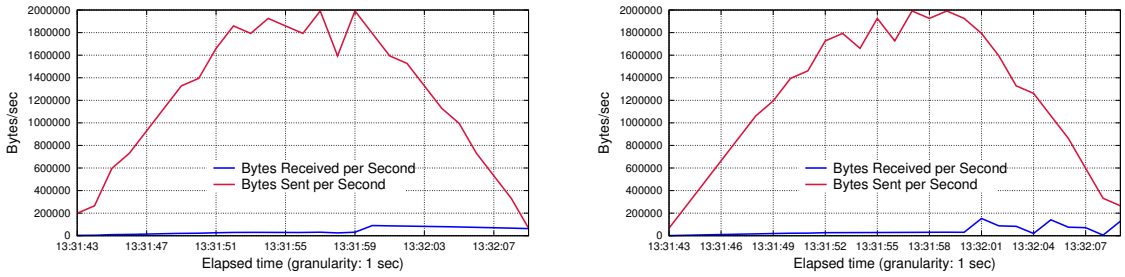


Figure 6: Average Response Time with Different File Size.

Figure 8. Overall, the transaction time distribution is considered acceptable as shown in Figure 9. More evaluations can be conducted with varying file types, operations and file sharing status. For provenance retrieval, we focus on the efficiency of requesting blockchain receipt for each of the provenance data entry. In our experiment, we query 10 records each time with a total size of 1.004KB, which uses an average time of 221ms. For each retrieval of blockchain receipt, we record the retrieval time for different file operations. Performance test for provenance data storage follows the similar way. Table II is the provenance retrieval overhead, from which we can conclude that our retrieval methods have a low overhead for the cloud storage system.

Table II: Overhead for provenance data retrieval

RecordType	Size of Data Transferred	Average Time Cost
File Create	1.07KB	0.838s
File Change	1.06KB	0.676s
File Delete	1.07KB	0.675s
File Share	1.07KB	0.790s



(a) Owncloud without ProvChain

(b) Owncloud with ProvChain

Figure 7: Bytes Throughput over Time.

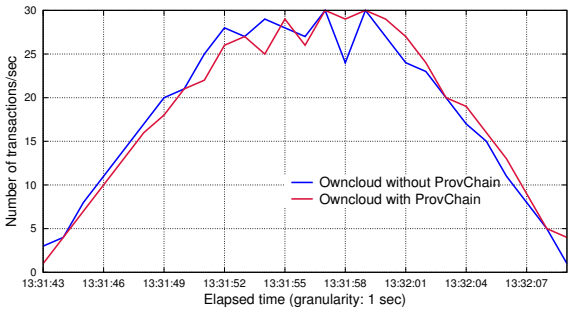


Figure 8: Number of Transactions per Second.

VI. Conclusions and Future Work

In this paper, we present the design and implementation of ProvChain, a blockchain based data provenance system for cloud auditing, with preserved user privacy and increased availability. Using blockchain technology, we make the record with unalterable timestamp and generate blockchain receipt for each of the data records for validation. Based on the current work, we can extend the system to various use cases where globally verified proof is needed. Instead of file as the data unit, we can also use other granularity such as data chunk in cloud storage. Our evaluation shows that provenance enabled ownCloud brings a low overhead.

As for the rewards of blockchain miners, cloud users may have to pay for a fee to enable data provenance services by cloud service provider. The service provider can then pay for the blockchain network. In this way, miners can be paid for continuous mining on blocks and validation of block authenticity. The fee can be determined depending on different level of data usage of each user.

Currently we collect provenance data inside one cloud service provider and one cloud application. For future work, we plan to develop ProvChain for federated cloud provider. Cloud storage applications on federated cloud providers will require the need to address interoperability, cross-provider data sharing and management. We will collect data provenance across different cloud providers and different cloud storage applications to provide better provenance services and enhance data security. For provenance validation, we currently use the Tierion API to validate the blockchain receipt. For future work, we will implement the validation on top of an open source architecture that will not only improve overall performance but also security and flexibility. We will use the collected provenance data to check for access control violations [31] which will in return provide better protection for the cloud storage application.

Acknowledgment

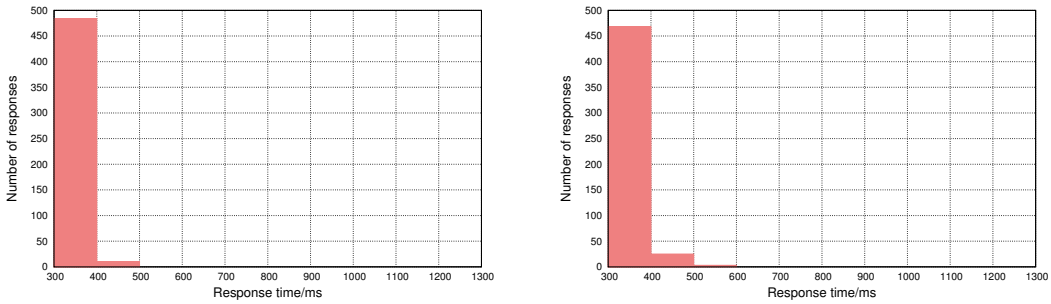
The author Xueping Liang is a visiting student at Tennessee State University and would like to thank State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100093, China and University of Chinese Academy of Sciences, Beijing, 100190, China where she is completing her PhD Study. This work is supported by Air Force Material Command award FA8750-16-0301 and Office of the Assistant Secretary of Defense for Research and Engineering (OASD (R&E)) agreement FA8750-15-2-0120

References

[1] Y. L. Simmhan, B. Plale, and D. Gannon, "A survey of data provenance in e-science," *ACM Sigmod Record*, vol. 34, no. 3, pp. 31–36, 2005.

[2] B. Lee, A. Awad, and M. Awad, "Towards secure provenance in the cloud: A survey," in *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*. IEEE, 2015, pp. 577–582.

[3] D. Tosh, S. Sengupta, C. A. Kamhoua, and K. A. Kwiat, "Establishing evolutionary game models for cyber security information exchange (cybex)," *Journal of Computer and System Sciences*, 2016.



(a) Owncloud without ProvChain

(b) Owncloud with ProvChain

Figure 9: Response Time Distribution.

[4] C. Kamhoua, A. Martin, D. K. Tosh, K. Kwiat, C. Heitzenrater, and S. Sengupta, "Cyber-threats information sharing in cloud computing: A game theoretic approach," in *IEEE 2nd International Conference on Cyber Security and Cloud Computing (CSCloud)*, 2015, pp. 382–389.

[5] D. K. Tosh, S. Sengupta, S. Mukhopadhyay, C. Kamhoua, and K. Kwiat, "Game theoretic modeling to enforce security information sharing among firms," in *IEEE 2nd International Conference on Cyber Security and Cloud Computing (CSCloud)*, 2015, pp. 7–12.

[6] D. K. Tosh, M. Molloy, S. Sengupta, C. A. Kamhoua, and K. A. Kwiat, "Cyber-investment and cyber-information exchange decision modeling," in *IEEE 7th Intl. Symposium on Cyberspace Safety and Security*, 2015, pp. 1219–1224.

[7] R. C. Merkle, "Protocols for public key cryptosystems," in *Security and Privacy, 1980 IEEE Symposium on*, April 1980, pp. 122–122.

[8] K.-K. Muniswamy-Reddy, D. A. Holland, U. Braun, and M. I. Seltzer, "Provenance-aware storage systems." in *USENIX Annual Technical Conference, General Track*, 2006, pp. 43–56.

[9] S. Sultana and E. Bertino, "A file provenance system," in *Proceedings of the third ACM conference on Data and application security and privacy*. ACM, 2013, pp. 153–156.

[10] C. H. Suen, R. K. Ko, Y. S. Tan, P. Jagadpramana, and B. S. Lee, "S2logger: End-to-end data tracking mechanism for cloud data provenance," in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, 2013, pp. 594–602.

[11] M. R. Asghar, M. Ion, G. Russello, and B. Crispo, "Securing data provenance in the cloud," in *Open Problems in Network Security*. Springer, 2012, pp. 145–160.

[12] R. Hasan, R. Sion, and M. Winslett, "Sprov 2.0: A highly-configurable platform-independent library for secure provenance," *ACM, CCS, Chicago, IL, USA*, 2009.

[13] R. K. Ko and M. A. Will, "Progger: An efficient, tamper-evident kernel-space logger for cloud data provenance tracking," in *2014 IEEE 7th International Conference on Cloud Computing*. IEEE, 2014, pp. 881–889.

[14] M. Imran and H. Hlavacs, "Applications of provenance data for cloud infrastructure," in *Semantics, Knowledge and Grids (SKG), 2012 Eighth International Conference on*. IEEE, 2012, pp. 16–23.

[15] D. K. Tosh, S. Shetty, X. Liang, C. Kamhoua, K. Kwiat, and L. Njilla, "Security implications of blockchain cloud with analysis of block withholding attack," in *Intl. Symposium on Cluster, Cloud and Grid Computing*. IEEE/ACM, 2017.

[16] "Ethereum project," https://www.ethereum.org/.

[17] "Multichain private blockchain white paper," http://www.multichain.com/download/MultiChain-White-Paper.pdf.

[18] "Design rationale," https://github.com/ethereum/wiki/wiki/Design-Rationale.

[19] "Tierion api," https://tierion.com/app/api.

[20] M. Ali, J. Nelson, R. Shea, and M. J. Freedman, "Blockstack: A global naming and storage system secured by blockchains," in *2016 USENIX Annual Technical Conference (USENIX ATC 16)*, 2016.

[21] N. Fotiou and G. C. Polyzos, "Decentralized name-based security for content distribution using blockchains," in *Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on*. IEEE, 2016, pp. 415–420.

[22] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: Decentralized computation platform with guaranteed privacy," *arXiv preprint arXiv:1506.03471*, 2015.

[23] A. Buldas, A. Kroonmaa, and R. Laanoja, "Keyless signatures infrastructure: How to build global distributed hash-trees," in *Nordic Conference on Secure IT Systems*. Springer, 2013, pp. 313–320.

[24] A. Buldas, R. Laanoja, and A. Truu, "Efficient implementation of keyless signatures with hash sequence authentication." *IACR Cryptology ePrint Archive*, vol. 2014, p. 689, 2014.

[25] D. Nguyen, J. Park, and R. Sandhu, "Dependency path patterns as the foundation of access control in provenance-aware systems." in *TaPP*, 2012.

[26] "Owncloud," https://owncloud.org/.

[27] "Chainpoint: A scalable protocol for anchoring data in the blockchain and generating blockchain receipts," http://www.chainpoint.org/.

[28] "Bitcoin block explorer," https://btc.com/.

[29] "Apache jmeter," jmeter.apache.org/.

[30] K. Udayasuryan, O. Oliver, and B. Chris, "Owncloud enterprise edition on ibm elastic storage server: A performance and sizing study for large user number scenarios," https://owncloud.com/wp-content/uploads/2016/03/WP-ownCloud-Enterprise-Edition-on-IBM-Elastic-Storage-Server-10Mar16.pdf.

[31] T. Ma, H. Wang, J. Cao, J. Yong, and Y. Zhao, "Access control management with provenance in healthcare environments," in *Computer Supported Cooperative Work in Design (CSCWD), 2016 IEEE 20th International Conference on*. IEEE, 2016, pp. 545–550.