Provision mechanism of authenticity of data origin in cloud environment based on Blockchain technology

Pavel Stetsenko[1], Gennadiy Khalimov[2],

[1]Kharkiv National University of Radioelectronics, Nauky Ave. 14, Kharkiv, 61166, Ukraine,
[2]Kharkiv National University of Radioelectronics, Nauky Ave. 14, Kharkiv, 61166, Ukraine

Abstract. This work is devoted to the problem of ensuring authenticity of data origin in cloud environments. A Blockchain-based protocol is proposed to solve this problem. Proposed protocol makes possible to use logging data as evidence in cybercriminal cases and mitigate possibility to modify logs by attackers. Scope of usage of proposed protocol is mainly internal audits in cloud environments.

Keywords. Blockchain, cloud environment, data origin, internal audit, cloud service provider

1 Introduction

At this time, the usage of cloud computing services is reaching a new level in various commercial and military spheres to ensure reliable data storage and dynamic "elastic" provisioning of resources for computing "on demand" of the cloud customers. Securing management and data transfer within and between the clouds is one of the key challenges for organizations, which implement cloud approach to their business. Cloud auditing can only be effective when all data operations can be reliably tracked. Ensuring the authenticity of the history of data origin is a process that determines the history of a data object, starting from its creation [1]. Provisioning of the authenticity of data origin can help detect malicious activity in architectures built on the cloud platform basis [2].

Blockchain technology has aroused interest due to a common, distributed and fault-tolerant data store that allows participants to counteract malicious attempts by using the computing capabilities of honest part of network nodes; and in which the exchange of information is resistant to manipulation. The Blockchain network is a distributed open transaction ledger in which network nodes verify and confirm any individual transaction. The decentralized Blockchain architecture can be used to develop the ability to provide an assurance of the authenticity of data origin for a cloud computing environment. In a decentralized architecture, each node is involved in a network to provide services, which provides higher efficiency. Accessibility is also ensured by distributed characteristics of Blockchain technology. Because cloud services often use centralized authority, there is a need to protect personal data while maintaining confidentiality. If there was a service for ensuring the authenticity of the cloud data origin based on Blockchain technology, transparent and constant recording would be ensured for all cloud operations. In addition, maintaining a history of data origin can help increase the confidence of cloud users in sharing information about cyberthreats [3, 4] in order to provide proactive cyber defense at a lower security cost [5,6].

This paper presents a Blochchain-based protocol for ensuring the authenticity of the history of data origin for cloud environments. The architecture proposed in the work records the operations with each data object and stores them as a history of origin, which is then hashed in the Merkle tree [7]. The list of data origin hashes will be the Merkle tree, and the root node of the tree will be tied to the Blockchain transaction. The list of transactions will be used to form the block, and the block must be confirmed by a set of nodes in the Blockchain network in order to be included in the Blockchain transaction ledger. Attempting to modify the record of data origin will require the attacker to locate the transaction and the block in the ledger. The underlying cryptography in Blockchain technology will only allow a block record to be modified if an attacker can submit a longer version of the Blockchain transaction ledger than the rest of the fair network, which is quite difficult to achieve. Using the global computing capabilities of the network, the Blockchain-based authenticity service for cloud data origin can provide integrity and authenticity. The protocol proposed in this paper preserves the confidentiality of users.

2 General theoretical information and related publications

2.1 History of data origin in the cloud

The history of the data origin describes the creation and modification chronology of the content of the object. From the point of view of information security, the history of data origin refers to the audit process, which keeps track of all operations performed with data generated by the work process. In the context of the modern use of Blockchain technology, an example of the history of data origin may be the ability to track the redistribution of funds in a distributed publicly available cryptocurrency transaction ledger, which contains all operations related to the asset data. The history of data origin based on Blockchain technology can provide features such as a verifiable event log in the cloud computing environment, creation and transfer of ownership of digital assets, consensus and cryptography-based identification.

Cloud computing environments are dynamic and heterogeneous, and include several different software and hardware components that are produced by different vendors and require interaction. Since companies, regardless of whether they are private or state-owned, use cloud computing as a platform for storing, processing, and providing services, data protection in the cloud has become a top priority for cloud providers. Ensuring data transfer within cloud architectures and between cloud architectures of different providers is often a prerequisite. The classic data guarantee aims to ensure the confidentiality, integrity and availability of data content. However, ensuring the authenticity of the data origin (where it came from and by whom it was created or modified) is currently a problem in cloud environments. Tracking every critical data item in the cloud can ensure the confidentiality, integrity, and availability of data content. This process, called confirmation of the history of the origin of the data, will record every transaction in the cloud data, so that information about all data operations can be obtained at any time to confirm their accuracy. The origin of the data has the potential to prevent internal attacks and network intrusion scenarios by identifying the exact sources that cause the state of the data object to become abnormal. Ensuring the authenticity of the history of the data origin in the cloud for all data processed in the cloud infrastructure will allow for secure distributed data computing, data and transaction exchange, detection of internal attacks, reproduction of research results and determination of the exact source of intrusions into the system or network. The current level of authenticity of the history of the origin of data in the cloud does not provide such guarantees, and there is a need to develop methods to solve this problem.

The history of data origin will play an important role for cloud security engineers when debugging hacks of a system or network or performing digital forensics. Cloud computing environments are typically characterized by the transfer of data between different system and network components. Data usually does not follow the same path because of the many copies of the data and the variety of paths used to ensure system stability. Such a diverse data streams creates a certain difficulty for security engineers to correctly and accurately respond to a possible security incident, determine which software and / or hardware components contained vulnerabilities that led to a successful attack, the source and the surface of the attack, as well as the attack blast-radius. The history of the data origin in the cloud can be a key tool for identifying security incidents with a high degree of granularity and evidence. Modern data ownership cloud-systems support the above tasks using logging and auditing technologies. These technologies are inefficient in cloud computing systems, which are complex in nature due to several levels of interaction between software and hardware components, covering various geographical and organizational boundaries. To identify and eliminate malicious actions in the cloud requires analysis of data and logs from a diverse and heterogeneous set of sources for a limited time period using digital forensics, which is an insurmountable task. Although the exchange of information related to cyberthreats may be one of the options for achieving situational awareness of the cloud attack surface with less investment, this approach is prone to information forgery threats [3-5]. A reliable history of data origin will help to track all operations performed on each data object in the cloud, and Blockchain technology will guarantee data authenticity and integrity.

Cloud computing systems usually consist of several nodes (physical machines) that host one or more virtual machines (VMs). Each virtual machine has an owner and includes such components as software (application resources) and data. Running software on the VM and exchanging information with the VM results in several artifacts, such as variables, intermediate data output, and final output artifacts. All of them are of interest and are important for establishing the history of data origin. Blockchain technology provides such an opportunity and has many necessary functions, as well as properties for efficiently establishing the data origin in the cloud environment. Blockchain is a peer-to-peer registration system in which information representing the origin of physical, virtual and application resources can be stored openly for transparent verification and audit. Thus, transparency and cost-effectiveness are ensured, and access control and confidentiality for individual register users are ensured using encryption methods, where individual users can see only those parts of the Blockchain ledger that are associated with their data. In addition, Blockchain technology provides much-needed features that take place to be part of cloud platforms, including asset transfer and source determination [8].

This section presents some previous researches in the field this work denoted to – authenticity of data origin in cloud environments. The first scheme designed to collect and maintain information about the origin of data was called PASS and it worked at the operating system level [9]. A comprehensive data tracking tool called S2Logger was developed to ensure the origin of cloud data and it works both at the file level and at the block level in kernel space [10]. It was proposed to ensure the authenticity of the history of data origin by using of encryption, which in turn entailed lower performance and higher computational cost [11]. A file system with the ability to collect information about the origin of data by intercepting calls to the file system below the virtual file system, which requires changes to operating systems, was proposed in [12]. A kernel-level logging tool, Progger, which can provide evidence of unauthorized access by violating user privacy, was proposed in [13]. It also explores the use of data origin history for cloud management, for example, discovering cloud resource usage patterns for resource reuse and failure management [14]. Another example of the use of encryption and digital signature to ensure the confidentiality and integrity of the history of data origin is SPROVE [15]. However, this tool does not have the ability to query origin data.

2.2 Blockchain Technology

Blockchain technology has caused great interest in various fields of activity – financial and agricultural sectors, healthcare, utilities, government, real estate, etc. The network architecture of Blockchain-systems is built on a common distributed peer-to-peer platform in which each participant can use the system's functionality, but no single user can control it. Blockchain technology assumes the presence of cybercriminals in the network and mitigates attack strategies based on the advantage in the hashing computational power of honest participants in the system and on the exchange of information between network nodes that is resistant to modification and destruction. The process of achieving consensus in the system is performed without a trusted central authority or intermediary, which speeds up the operation in the Blockchain system in comparison with traditional centralized systems. Forgery of data in the Blockchain ledger is extremely difficult due to the use of the cryptographic data structure and the lack of authenticity of secrets. Blockchain networks are fault tolerant, which allows nodes to eliminate compromised nodes. Despite this, several vulnerabilities could potentially violate the integrity of data in the Blockchain transaction ledger [16]. However, this requires a huge amount of computational power for the attacker to carry out attacks, which may deprive the attacker of any benefit in carrying out the attack.

The decentralization and security features of Blockchain technology have attracted researchers to develop various applications, such as smart contracts, identity management and distributed DNS.

Hyperledger is an open source Blockchain project supported by The Linux Foundation, which includes leaders in finance, banking, IoT, supply chain, manufacturing and technology fields [17]. Hyperledger Fabric is an architecture that provides a high degree of confidentiality, flexibility, and scalability on top of the Hyperledger platform, supporting plug-in implementations of various user components [18]. Developers can capitalize on Fabric infrastructure by integrating custom and custom methods on an open platform.

The Multichain project provides an open source Blockchain network where developers can place their Blockchain applications on top of a private cloud architecture [19]. Multichain uses a transaction model for each output and can work with high throughput [8]. The transaction model for each output means that the input of each transaction has some connection with the output of the previous transaction. Using different addresses for the same user, this model provides a higher degree of confidentiality. Developers can use different types of assets for different types of transactions due to built-in multicurrency functionality in Multichain. In addition, the Multichain project added two functions - Blockchain messaging and database synchronization.

The Ethereum platform, on the contrary, is designed for simple and fast development of Blockchain applications, which is one of the most outstanding features of the transaction model for each address. In addition, this saves space, since each transaction requires only one signature, one link and one exit. In addition to the Bitcoin cryptocurrency ecosystem, the Ethereum decentralized platform was also developed on top of the public Blockchain ledger conception for simple and quick development of decentralized applications [20]. To implement the function of transmitting values and rewards of participants, Ethereum has an internal cryptocurrency called Ether. This platform provides smart contract features that can be implemented via Solidity or other high-level programming languages. On Blockchain networks, smart contracts are compiled in binary format and are able to work on the Ethereum virtual machine (EVM). The Ethereum platform adopts a transaction model for each address, and each single transaction is independent, which means that the transaction simply transfers assets between participating nodes.

The Tierion project, which is used to implement the protocol proposed in the work, provides a platform for downloading and publishing data records on the Blockchain network [21]. Due to its open application programming interfaces (APIs), Tierion is convenient for integrating applications that require a Blockchain archi-

tecture. Developers can publish metadata via an HTTP request to the Tierion datastore and retrieve information about records. Each data record has an identifier that can be used to receive a transaction-generated Blockchain receipt. The Blockchain receipt contains the transaction identifier that will be used to determine the location of the transaction, and the block in which the transaction is located. Thus, the data record placed in the Blockchain ledger cannot be falsified, and integrity is guaranteed.

Guardtime provides Blockchain services on an industrial scale using the Keyless Signature Infrastructure (KSI) and a secure one-way hash function, which is post-quantum in contrast to the asymmetric cryptographic algorithm (RSA) [22]. Guardtime also proposed the Blockchain standard for digital identification and a protocol for authentication and digital signature, providing a simplified feedback management protocol and long-term validity [23]. Enigma is a decentralized computing platform with guaranteed confidentiality that uses Blockchain technology for network management, access control and identification, and also creates an event log protected from unauthorized access [24]. A decentralized public key infrastructure (DPKI) on top of Namecoin and a naming and storage system based on Blockchain technology were proposed in [25]. It has also been proposed to use Blockchain technology in information networks to protect the security of name-based content distribution [26].

2.3 Blockchain technology and ensuring the authenticity of the history of data origin

Blockchain technology provides the ability and many of the necessary functionality and features for effective ensuring the authenticity of the history of data origin in cloud environments. Blockchain technology is a peer-to-peer transaction ledger system in which reliable information about the origin of physical, virtual and application resources can be stored publicly for transparent verification and audit. In addition, one of the main concerns here is non-repudiation. Thus, the implementation of Blockchain technology in the cloud can lead to the task of ensuring the authenticity and non-repudiation of the history of data origin, when cloud nodes implicitly create a distributed network for recording data origin in a distributed and fault-tolerant Blockchain ledger protected by a strict cryptographic protocols. This distributed ledger must be updated by all nodes in the cloud based on a strictly regulated protocol for achieving consensus, the development of which for the cloud is a rather difficult task.

3 Architecture of the Blockchain protocol for ensuring the authenticity of data origin

This section presents the architecture of the Blockchain protocol for ensuring the authenticity of data origin in cloud environments. The proposed architecture allows achieving the following goals:

1. Real-time authenticity of the history of cloud data origin – user operations are monitored in real time to collect information about the history of origin, which will further support the application of access control policies and intrusion detection systems [27]. However, a delay occurs when placing records in blocks and processing them by the Blockchain network but capturing data events is real time process.
2. Protection against unauthorized access – a reliable history of the data origin is collected and then published to the Blockchain ledger to achieve data integrity. Then all data are distributed between nodes. The architecture provides creation of a public log with all user operations on cloud data with time stamps and without a trusted third party. A special construction called “Blockchain receipt” is assigned to each record for further verification. Moreover, according to the principle of least privilege an access to proposed protocol can be configured more granularly with cloud Identity and Access Management.
3. Increased confidentiality. Each entry in the data origin history is associated with a hashed user identifier in order to maintain its confidentiality, so that no Blockchain network node can match data records associated with a specific user. The data origin auditor can access information related to the user, but can never determine his identity. Only a service provider (cloud provider) can associate identifiers with the actual owners of each record in the data origin history. As for regulation compliance – the proposed protocol is focused mainly on internal audit and operates with data generated by employees of the organization with cloud access, and GDPR or CCPA are focused on the privacy of customer's data i.e. consumers not employees.
4. Confirmation of the authenticity of the data origin history in the cloud – a record of the history of the origin of data is published globally in the Blockchain network, where several nodes provide confirmation for each block. To check each record of the history of data origin, a Blockchain receipt is used.

The following methods were used in the architecture development to achieve the goals mentioned above:

– *real-time monitoring of user actions* using interceptors and listeners, so that each user file operation will be collected and recorded to obtain a history of data origin;
– *storing all hashed data in the form of blocks in the Blockchain transactional ledger*. Each node in the system can verify operations by analyzing the block so that the origin of the data is reliable and protected from falsification;
– *hashing the user ID* when adding data to the Blockchain ledger so that the network and the auditor cannot determine the identity of the user and operations with the data.

The cloud auditor of data origin history performs verification by extracting transactions from the Blockchain network using the Blockchain receipt, which contains information about the block and transactions.

3.1 Architecture Overview

Key architecture components are listed below:

- **Cloud user** – a user who owns his data, has distributed connections with other users and can choose the Blockchain-authenticity service for the authenticity of cloud data origin, in which information about the history of origin is stored in a public Blockchain ledger. User data changes can be monitored and checked by Blockchain nodes, but nodes may not be aware of the details of other user actions. It is important to note that the information about the history of data origin contained in the open Blockchain ledger does not allow to unambiguously establishing the user's identity.
- **Cloud Service Provider (CSP).** The cloud service provider offers the cloud storage service and is responsible for registering users. CSP can benefit from the proposed architecture in the following aspects: it becomes possible to constantly check data changes and data operations performed by all users in order to better develop the proposed functionality; Using a reliable history of data origin to detect intrusions and malicious actions within the system. With regard to business aspects, it is possible to increase the brand reputation by providing a Blockchain service to ensure the authenticity of data origin. The possibility of providing a PaaS-managed Blockchain-authenticity service for data origin for an additional fee is also possible.
- **Database of the history of data origin** – the database stores all information about the history of the origin of data in the Blockchain network, which is used to detect malicious behavior. All entries in the database are anonymous.
- **Provenance Auditor (PA)** – can extract all information about the history of data origin from the open Blockchain ledger to the database and confirm the Blockchain receipt. The auditor is responsible for maintaining the database, but at the same time cannot correlate the record from the history of data origin with the owner of this data.
- **Blockchain network** – consists of globally participating nodes. All data operations will be placed in the history of origin in the form of blocks and checked by the nodes of the Blockchain network.

3.2 Background and approach concept

The proposed architecture uses a cloud file as a unit of data and monitors file operations to provide the Blockchain-based service for the authenticity of cloud data origin. After detecting each file operation, an origin history record is generated. The cloud service provider then uploads an origin history record to the Blockchain network. It is important to note that the system can be scaled by increasing the number of nodes in the Blockchain network (scaling-out) or by deploying more powerful nodes with the same number of them (scaling-up), the database component with origin history can be scaled in the same way. This section describes in detail the scenario of using a cloud file as data unit and the structure of the Blockchain block.

Cloud file usage scenario. To keep track of the history of each file in the cloud, actions such as creating, modifying, copying, sharing, and deleting the file are recorded. User A can create a file that references the source of file X. Then user A copies file X to another location, for example, for backup or other reasons. The read and write operation of user A on file X can also be recorded. If user B asks user A to share file X, an entry will be created for user A and user B. User A shares file X at a predetermined location and user B creates a new file Y from shared file X. Then user B can work with file Y, just like user A with file X, for example, with read and write operations. If user B deletes the file, an entry will be created for deletion. At some point, user A decides to make file X public in order to change access to the file. Anyone who gets access to it will also create a new file in the appropriate place. An approach called Versioning is applied to keep the history of different versions of the file for future use.

Block structure – the architecture proposed in the work uses Blockchain technology to ensure authentication of data records and prevent falsification. The block structure consists of two parts – the block header and the list of transactions (operations). The main attributes in the header are block hash, height, confirmations, nonce value and Merkle tree root. The hash value of a block is calculated using the hash of the previous block and a one-time number. Height is the block index in the global Blockchain network. The value of block confirmations indicates the number of nodes that performed the mining process on this block, and the one-time number is used by Blockchain nodes to verify the integrity of the block. The root of the Merkle tree is the root of the binary hash tree created from all transactions in the block. Transaction lists follow the block header. Each transaction has a hash with inputs and outputs. Each data record is hashed into a Merkle tree node. The root node of the Merkle tree will be bound to one transaction in a specific block.

3.3 Threat Model

This section presents an analysis of potential vulnerabilities in the proposed architecture. The cloud service provider provides the ability to enable the Blockchain service for the authenticity of data origin, as well as a cloud storage service that allows users to store data on a cloud platform. A cloud service provider cannot guarantee that data records will remain unchanged due to known vulnerabilities in hypervisors and cloud operating systems. As soon as the Data Origin Blockchain Autheticity service is enabled, the user will be able to track the data, and the auditor will have access to all information about the history of data origin in the cloud. However, the provenance auditor cannot be fully trusted. An attacker could potentially gain access to or modify user data and / or information about the origin of user data. Since the main purpose of the architecture proposed in this work is to protect information about the history of data origin in the cloud, user data should be stored in the cloud in encrypted form and be accessible only if there is a decryption key.

3.4 Key activation

In order to use the Blockchain service for authenticity of data origin, users must enable the service and create their credentials. To ensure confidentiality in cloud storage applications, users must generate key pairs to encrypt their cloud data using the key management services provided by cloud providers (for example, AWS Key Management Service or Azure Key Vault). If the user wants to share the file, a key for data exchange will be provided. For information on the history of data origin, the cloud provider generates key pairs for privacy reasons, since this information will be published in the open Blockchain ledger in the future. The purpose of the keys in the proposed architecture is described as follows.

- User registration key (K_{UR}) – is necessary for the user to register the cloud storage service. Each time a user wants to work with cloud data, a registration key will be required.
- Data encryption key (K_{DE}) – after registration, the user generates an encryption key K_{DE} to encrypt all data stored in the cloud. When a file is created, the user has the ability to encrypt the file, which restricts access to the file only to key owners.
- Key pair for data sharing (K_{DS}^{Pub} , K_{DS}^{Priv}) – in general, the private key is used to generate signature data by the owner, and the public key is used to verify data ownership. When the data owner permits data sharing, he transfers K_{DS}^{Priv} to another user.

The key to verify the authenticity of the history of data origin (K_{PV}) – each block in the Blockchain register contains several records of the history of data origin; and data on the origin is entered upon detection of each new operation with the file. Each data transaction causes the cloud service provider to generate a key K_{PV} to encrypt information about the history of origin. K_{PV} will then be transferred to the auditor of the history of origin, if the user designates it for the audit.

4 Implementation of Blockchain-authenticity of the history of data origin in the cloud

Implementation of the cloud-based Blockchain service proposed in the work is carried out using a three-tier architecture, consisting of a data storage level, a Blockchain network level, and a database level for storing data origin history. It includes three phases: collecting information about the data origin history, checking and adding it to the Blockchain ledger and updating the database with new historical records (Fig. 1).

Each level is designed to perform the following functions:

- Data Storage Layer – designed to support cloud storage applications. In this case, one cloud provider (mono-cloud architecture) is used, however, the possibility of implementing a multi-cloud approach is supported.
- Blockchain network level – designed to record each operation in the history of data origin. Each block can record several data operations. In this example, the file is used as a unit of data; therefore, each file operation with the user name and file name is recorded. File access operations include Create, Share, Modify, and Delete.

Database level – designed to store records of file operations and write queries. It is created in the most isolated segment of cloud architecture and can be based on PaaS relational database service such as AWS RDS or Azure SQL Database and on PaaS non-relational database service such as AWS DocumentDB or Azure Cosmos DB. The cloud provider appoints an origin history auditor to verify data from the Blockchain ledger. The result of the verification is a Blockchain receipt, which is added to the databases and serves as a guarantee of the authenticity of the record.

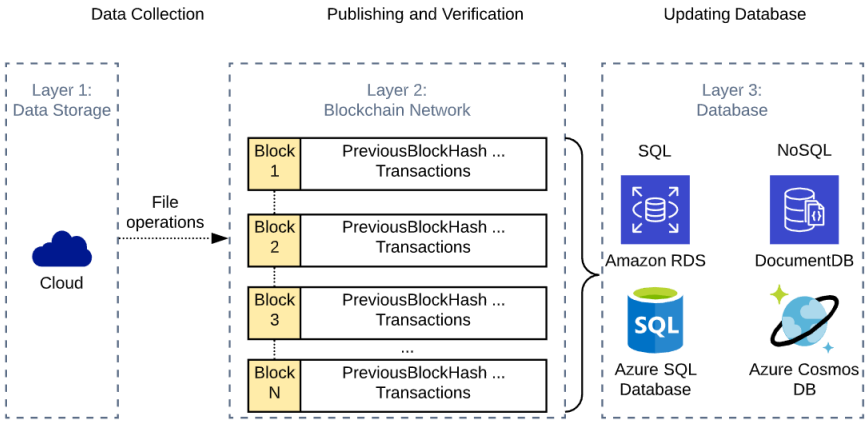

Fig. 1. – High-level representation of the architecture of the Blockchain-authenticity protocol

4.1 Collection and storage of information about the data origin history

When a user performs actions on data files stored in the cloud, the corresponding operations are recorded. An operation can be indicated in metadata, including all file attributes. It should be noted that for this stage only the attributes RecordID, date and time (timestamp), username, file name, registered user (AffectedUser) and action are recorded. The transaction hash, block hash and verification field will be collected after the auditor makes a request to the Blockchain network. The AffectedUser attribute is considered in two cases. One of them is data modification, in which the same user works with data using a data encryption key when there are no affected users except this user himself. Another case is data exchange, when a user shares a file with someone else. In the second case, this attribute in the file operation metadata will include all users in the shared group.

This paper presents an architecture built on the basis of an open source application called ownCloud to demonstrate the capabilities of Blockchain-authenticity of data origin [28]. ownCloud is a proprietary server for synchronizing and sharing files. OwnCloud provides both cloud storage web services and a PC client, similar to Dropbox and Google Drive, which provide the user with control over personal data and universal access to files for all data. In addition, ownCloud is flexible, and developers can use their features to develop various applications on top of it, allowing authorized users to enable and disable features, set policies, back up and manage access. The server also manages and protects API access for its ownCloud client and developers, while providing the internal processor needed to provide high-performance file sharing services.

To collect information about the history of data origin, hooks are used to listen to file operations in the ownCloud application web interface. After monitoring the operation, a record will be generated, which is then up-

loaded to the Blockchain network and stored in the origin history database. The process of collecting and storing information about the history of data origin is shown in Figure 2. An example of collecting information about a file change in the original JSON format is presented at Figure 3.

To store information about the history of origin after data collection in this implementation, the Tierion API is used to publish data records in the Blockchain network [21]. Tierion provides an API primarily for collecting data and for managing data stores and records in your personal account. Accessing the Tierion Data API requires an API key, which is required with every request for API data. Granting credentials should contain the X-Username and X-Api-Key headers for each data store owned by the user account. In addition to using the API data to create the record, it is possible to submit the HTML form directly to Tierion, since ownCloud is based on web technologies, and information about the history of data origin comes from the website. This approach is easier to implement for demonstration purposes.

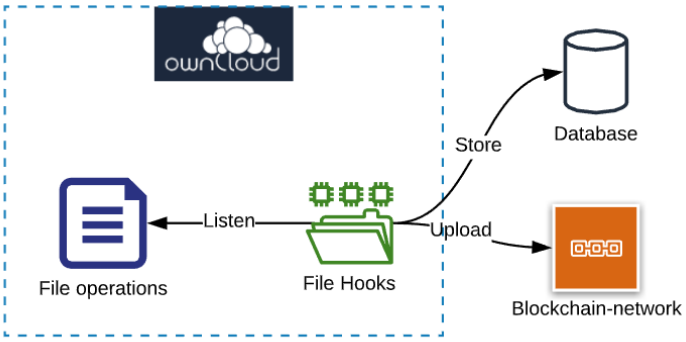

Fig. 2. – The process of collecting and storing information on the history of data origin

To store information about the history of origin after data collection in this implementation, the Tierion API is used to publish data records in the Blockchain network [21]. Tierion provides an API primarily for collecting data and for managing data stores and records in your personal account. Accessing the Tierion Data API requires an API key, which is required with every request for API data. Granting credentials should contain the X-Username and X-Api-Key headers for each data store owned by the user account. In addition to using the API data to create the record, it is possible to submit the HTML form directly to Tierion, since ownCloud is based on web technologies, and information about the history of data origin comes from the website. This approach is easier to implement for demonstration purposes.

```
{
    "app"           : "files",
    "type"          : "file_changed",
    "affecteduser"  : "testuser",
    "user"          : "testuser",
    "timestamp"     : "124235234",
    "subject"       : "self_changed",
    "message"       : "",
    "messageparams" : "[]",
    "priority"      : "30",
    "object_type"   : "files",
    "object_id"     : "1425",
    "object_name"   : "test.txt",
    "link"          : "apps/files/test/"
}
```

Fig. 3. – An example of collecting information about a file change in the original JSON format

To ensure confidentiality, the username is hashed. Thus, the provenance auditor cannot know to which user the data belongs. Only a cloud provider can associate each user with a hash value because the provider stores a list of usernames. The proposed architecture involves storing information about the data origin in a database for subsequent updates and checks. The publication of data records in the Blockchain network is based on the Chainpoint standard [29]. Chainpoint is an open standard for creating timestamps for any data, files or series of

events, which offers a scalable protocol for publishing data records in the Blockchain ledger and generating Blockchain receipts. By binding an unlimited amount of data to several Blockchain ledgers and checking the integrity and existence of data without using a third trusted party, the Chainpoint standard is widely used in Blockchain applications. According to Chainpoint 2.0, data records are hashed, so each Merkle tree can contain a large number of records, as shown in Figure 4.

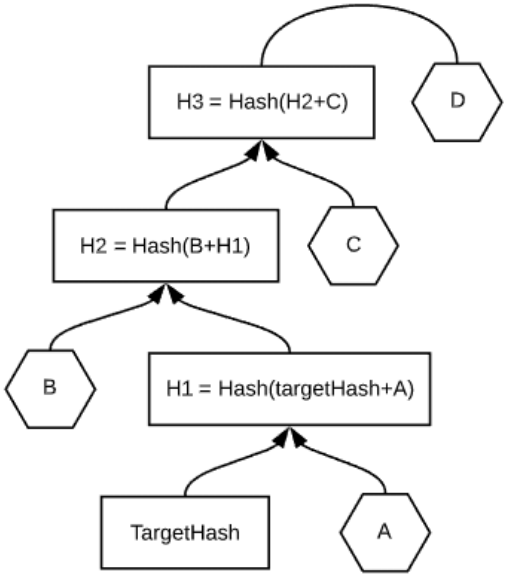

Fig. 4. – Scheme of the Merkle tree

The target hash of a particular record and the path to the Merkle root provide evidence of the authenticity of the data origin history, which is a JSON-LD document containing information for cryptographic verification that part of the data is tied to the Blockchain ledger. This proves that the data existed at the time of their binding. The root for each Merkle tree is associated with one transaction in the Blockchain network.

4.2 Verification of data origin

To check data records published on the Blockchain network, the data origin auditor requests a Blockchain receipt through the Tierion API. The data API offers a way to validate Blockchain receipts. Before verification, an API call to the GET method is used to request a record along with a Blockchain receipt.

The request header should include Content-Type: application/x-www-form-urlencoded or Content-Type: application/json to set the data format to receive. API calls to request data are made via the HTTPS protocol. The Blockchain receipt contains information about the Blockchain transaction and the evidence of authenticity used to verify the transaction. An example of a Blockchain receipt is shown in Figure 5.


```
{
 "@context": "https://w3id.org/chainpoint/v2",
 "type": "ChainpointSHA256v2",
 "targetHash": "82e46ffd212d680b3e1a169e6a8b5947298 5ac55398b8740832fe94fd5e5fd63",
 "merkleRoot": "9f0100055a430539796817ce626d84ccb5485453e4d558cf3353e4d4a7e59031",
 "proof": [
    {
       "right": "0f6117e8bddd7fdc713aa5365e74aafe34f5cc31fd654ed84ea37976d873c087"
    },
    {
       "left": "f860e7697ba57d944d925f311cce786e6d20833071d1c16e6e5fef3fc4749c96"
    },
    {
       "right": "de4b5b29183d193b95905ae9741a928ab056cbbbefb9a537ac9282fe180c78bd"
    },
    {
       "right": "e75da94bc44a3a9778b2ec7a5ffd58e4a622d4ce4c20676215eb88a4764bb335"
    }
 ],
 "anchors": [
    {
       "type": "BTCOpReturn",
       "sourceId": "0b956b057330591cd63c90e5572ba364c6f9f08299c3e8ee0c893411db1c30a6"
    }
 ]
}
```


Fig. 5. – Blockchain receipt

The Merkle tree can be restored from a Blockchain receipt. Each record of origin is stored together with other records in the Blockchain network as a transaction, which is available in the Blockchain Block Explorer [30].

Since the transaction attribute “Height” corresponds to the block index, you can find the exact information about the block. An example of the data contained in the transaction and the Blockchain block is shown in Figure 6.

An API call to the POST method is used to check the format and contents of the Blockchain receipt and to confirm that the root of the Merkle tree of one record is stored in the Blockchain ledger.

The algorithm for the auditor to verify the history of the origin of the Blockchain receipt data is shown in Figure 7.

In the algorithm, proof of authenticity, the root of the Merkle tree (merkleRoot) and the target hash value (targetHash) are the input parameters to the Blockchain receipt, and the output is the result of the check. If true is returned, then the data record is considered verified and authentic based on the fact that the transaction and the block are genuine. If false is returned, it means that the block and data record have been tampered. It should be noted that according to the requirements of Chainpoint 2.0, all hash values used in the construction of Merkle trees and evidences are processed in binary format. Anchor points in a Blockchain receipt indicate how a data record is bound. Verification of the Blockchain receipt confirms that the content is valid and authentic. In particular, the verification process confirms the following four elements: Blockchain receipt is a JSON document with correct formatting; all required fields are present; targetHash, merkleRoot, and proof are valid; and the anchor point of the merkleRoot value to the specified location(s) is correct.

Fig. 6. – Transaction and Blockchain Block Data

```
1   Validate(proof,merkleRoot,targetHash)
2   SET nodeNum = number of Merkle tree nodes in proof
3   SET h = targetHash;
4   SET i = 0;
5   WHILE (i < nodeNum) DO {
6       IF (proof(i).key=right) THEN {
7           h = hash(h+proof(i).value)
8       } END
9       ELSE {
10          h = hash(proof(i).value+h)
11      } END
12      i=i+1
13  } END
14  IF (h = merkleRoot) THEN {
15      RETURN true
16  } END
17  RETURN false
```

Fig. 7. – Blockchain receipt verification algorithm

After checking the Blockchain receipt, the data origin auditor can update the entry in the origin history database by filling out the remaining attributes, including the transaction hash, block hash and verification result. If the result of the audit is correct, then the auditor can verify that the origin data is authentic. If the result is false, the auditor informs the service provider that there has been an invasion to the system.

A high-level representation of the architecture for confirming the history of data origin in the cloud based on Blockchain technology is presented in Figure 8.

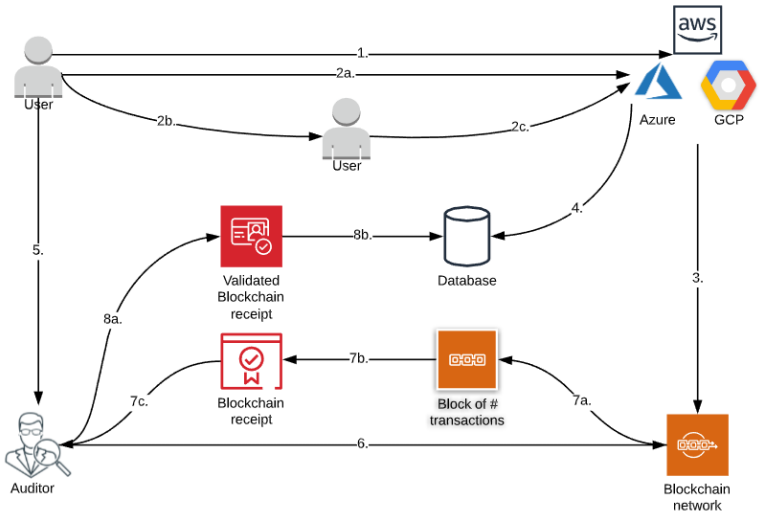

Fig. 8. – Architecture of the Blockchain protocol for authenticity of data origin in the cloud

Work process:

1. User registration using the key K_{UR}.
2. Work with data:
   2a Request access to data using a key K_{DE}.
   2b. Providing access to data to another user using a key pair $K_{DS}^{Pub}, K_{DS}^{Priv}$.
   2c. Data request.
3. Publication of information on the history of data origin in the Blockchain network using a key K_{PV}.
4. Storage of information on the history of data origin in the database.
5. Request for verification by the auditor of the authenticity of information about the history of data origin.
6. The auditor's request to the Blockchain network to obtain data for verification using the key K_{PV}.
7. The verification process.
   7a. Getting a Blockchain block with transactions.
   7b. Authentication.
   7c. Blockchain receipt update.
8. Updating the database.
   8a. Sending a verified Blockchain receipt.
   8b. Updating the status of checking the history of data origin in the database.

5 Directions for future research

This paper presents the development and implementation of the Blockchain protocol of the authenticity of data origin for cloud auditing while maintaining user privacy and increased accessibility. Usage of Blockchain technology allows recording data with an invariable time stamp and generating a Blockchain receipt for each of the data records for verification. The cloud implementation of the proposed system allows ensuring the stability of operation, fault tolerance, elasticity and scalability. The implementation discussed in this paper can be taken as a basis for various applications – for the implementation of a more secure cloud-based data and security incident management system (SIEM), offering to users as an option for existing cloud-based logging services (for example, for AWS CloudTrail or Azure Monitor) Blockchain-validity of journal entries, etc. Instead of a file, described protocol can use another granularity as a data unit, such as a data block in a cloud object storage (AWS Simple Storage Service or Azure Blob Storage).

The work did not sufficiently cover the process of calculating rewards for the process of adding new blocks to the Blockchain ledger – mining. A cloud service provider can provide Blockcahin credibility for an additional fee, which, in turn, will be aimed at maintaining the Blockchain network and providing rewards for the successful addition of new blocks. The fee can be determined depending on the amount of data for which the user wants

to provide Blockchain validity or on the number of validation checks for each user. In the context of this issue, the cost of an individual Blockchain receipt and the cost of maintaining the infrastructure should be calculated.

In this paper, the process of collecting information about the history of data origin in the cloud environment of one provider (mono-cloud) and one cloud application was considered. The urgent task is to expand the proposed system to the level of multi-cloud, which will require the solution of problems of interaction, data sharing between different cloud providers and their management.

For future work, the analysis and improvement of the overall performance of the safety and authenticity of the system is also relevant. Especially database level is considered as a direction for future research to investigate performance impact of choosing cloud database service in large cloud environments with massive amounts of data operations. Another area for future research is the possibility of introducing automation and machine learning into the system to provide the functionality of the security orchestration, automation and response (SOAR) system for the automatic response to security incidents and verification of access control violations. Collected data can be used for creation of behavioral patterns, which in turn can be used for developing of automated event-driven security responses by using ML and serverless tools. Automatically generated access control rules will be better used to detect malicious behavior and prevent intrusions, which will provide better protection for cloud applications.

Moreover, the proposed architecture can be applied to increase the security of IoT systems in the cloud, where a large number of mobile devices are responsible for collecting and processing data. In the context of IoT cloud architectures performance and throughput of Blockchain-based protocol become especially important due to big amounts of data received from various devices.

6 Conclusions

The proposed protocol for providing Blockchain authenticity provides real-time auditing for access to all data in a cloud storage application. A file was used as a data unit, verification of all operations with cloud data objects was implemented, as well as recording using Blockchain technology. In this way, information on all cloud access events can be collected and analyzed.

Information about the history of data origin is converted and uploaded to the Blockchain network for each file operation. Thus, a reliable and unchanging fingerprint of file operations is created with a safe and constant update, as well as an unchanged time stamp for each operation. Any malicious intervention in the Blockchain ledger will be detected when checking the Blockchain receipt. Once a data record is published, no one can rewrite or modify the record without disclosing it.

Using a Blockchain network avoids the need for a trusted party. The architecture also helps to avoid the need for trust in a cloud service provider when storing data origin. In decentralization, data records are confirmed and verified by continuous cross-checking by the system between computing nodes. In addition, the decentralized method ensures the integrity of data records, and each data record has a copy on each node in the Blockchain network, thus ensuring resistance to DDoS attacks. In addition, there is no single point of failure, since the loss of one or even several nodes in the network does not lead to the loss of all data and the same is for database component variety of High Availability and Disaster Recovery options in the cloud especially for PaaS services.

Users can subscribe to the Blockchain data authenticity service while maintaining their privacy. User access records are anonymized on the Blockchain network, and the auditor cannot study user actions. Anonymity persists in two aspects. On the one hand, the user ID will not be associated with origin data records, since the user ID is hashed. On the other hand, incoherence is achieved between each user, especially for the history of data origin, to which several users have access.

References

1. Simmhan, Y. L., Plale, B., Gannon, D.: A survey of data provenance in e-science. ACM Sigmod Record, vol. 34, no. 3, pp. 31-36 (2005)
2. Lee, B., Awad, A., Awad, M.: Towards secure provenance in the cloud. A survey, in 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), IEEE, pp. 577-582 (2015)
3. Tosh, D., Sengupta, S., Kamhoua, C A., Kwiat, K. A.: Establishing evolutionary game models. CYBEX, Journal of Computer and System Sciences, vol. 98, pp. 27-52 (2018)
4. Kamhoua, C., Martin, A., Tosh, D. K. Kwiat, K. Heitzenrater, C., Sengupta, S.: Cyber-threats information sharing in cloud computing: A game theoretic approach. In IEEE 2nd International Conference on Cyber Security and Cloud Computing, pp. 382-389 (2015)

5. Tosh, D. K., Sengupta, S. Mukhopadhyay, S., Kamhoua, C., Kwiat, K.: Game theoretic modeling to enforce security information sharing among firms. In IEEE 2nd International Conference on Cyber Security and Cloud Computing (CSCloud), pp. 7-12 (2015)
6. Tosh, D. K., Molloy, Sengupta, M. S., Kamhoua, C. A. Kwiat, K A.: Cyber-investment and cyber-information exchange decision modeling. In IEEE 7th International Symposium on Cyberspace Safety and Security, pp. 1219-1224 (2015)
7. Merkle, R. C.: Protocols for public key cryptosystems. In IEEE Symposium on Security and Privacy, April 1980, P. 122 (1980)
8. Sharif, A.: Design rationale. [Online]. Available: https://github.com/ethereum/wiki/wiki/Design-Rationale (2018).
9. Muniswamy-Reddy, K.-K., Holland, D. Braun, A., U., Seltzer, M. L.: Provenance-aware storage systems. In USENIX Annual Technical Conference, General Track, pp. 43-56 (2006)
10. Suen, H. Ko, R. K., Tan, Y. S., Jagadpramana, P., Lee, B. S.: S2logger: End-to-end data tracking mechanism for cloud data provenance. In 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, IEEE, pp. 594-602 (2013)
11. Asghar, M. R., lon, M., Russello, G., Crispo, B.: Securing data provenance in the cloud. In Open Problems in Network Security. Springer, pp. 145-160 (2012)
12. Sultana S., Bertino, E.: A file provenance system. In Proceedings of the Third ACM Conference on Data and Application Security and Privacy, ACM, pp. 153-156 (2013)
13. Ko, R. K. Will, M. A.: Progger: An efficient, tamper-eviden kernel-space logger for cloud data provenance tracking. In IEEE 7th International Conference on Cloud Computing, IEEE, pp. 881-889 (2014)
14. Imran, M., Hlavacs, H.: Applications of provenance data for cloud infrastructure. In Eighth International Conference on Semantics, Knowledge and Grids (SKG), IEEE, pp. 16-23 (2012)
15. Hasan, R., Sion, R., Winslett, M.: Sprove 2.0: A highly-configurable platform-independent library for secure provenance. ACM, CCS, Chicago, IL, USA (2009)
16. Tosh, D. K., Shetty, S., Liang, X., Kamhoua, C., Kwiat, K., Njilla, L.: Security implications of Blockchain cloud with analysis of block withholding attack. In International Symposium on Cluster; Cloud and Grid Computing, IEEE/ACM, Madrid (2017).
17. The Linux Foundation. "Hyperledger-blockchain technologies for business." [Online]. Available: https://www.hyperledger.org/ (2018)
18. Cachin, C.: Architecture of the Hyperledger Blockchain fabric. In Workshop on Distributed Cryptocurrencies and Consensus Ledgers (2016)
19. Greenspan, G.: Multichain private Blockchain white paper." [Online]. Available: http://www.multichain.com/download/Multichain.White Paper.pdf (2015)
20. Ethereum project. [Online]. Available: https://www.ethereum.org/ (2018)
21. Tierion, "Tierion api. [Online] Available: https://github.com/chainpoint/chainpoint-node/wiki/chainpoint-Node-Api:-How-to-Create-a-Chainpoint-Proof.
22. Buldas, A., Kroonmaa, A., Laanoja R.: Keyless signatures infrastructure How to build global distributed hash-trees. In Nordic Conference on Secure 17 Systems Springer, pp. 313-320 (2013)
23. Buldas, A., Laanoja, R., Truu, A.: Efficient implementation of keyless signatures with hash sequence authentication. [Online]. IACR Cryptology ePrint Archive, vol. 2014, p. 689. Available: https://eprint.iacr.org/2014/689.pdf (2014)
24. Zyskind, G., Nathan, O., Pentland, A.: Enigma: Decentralized computation platform with guaranteed privacy. [Online]. Available: https://arxiv.org/pdf/1506.03471.pdf (2015)
25. Ali, M., Nelson, J. Shea, R., Freedman, M. J.: Blockstack: A global naming and storage system secured by Blockchains. In USENIX Annual Technical Conference (USENIX ATC 16), (2016)
26. Fotiou, N. Polyzos, G. C.: Decentralized name-based security for content distribution using Blockchains. In IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 415-420 (2016).
27. Nguyen, D., Park, J., Sandhu, R.: Dependency path patterns as the foundation of access control in provenance-aware systems. [Online]. TaPP. Available: https:l/www.usenix.org'system/files conference/tapplztappl2-final23.pdf (2012)
28. ownCloud [Online]. Available: https://owncloud.org/ (2018)
29. Chainpoint: A scalable protocol for anchoring data in the Blockchain and generating Blockchain receipts. [Online]. Available: http://www.chainpoint.org/ (2018)
30. Bitcoin block explorer. [Online]. Available: https://btc.com/ (2018)