Ensuring Data Integrity Using Blockchain Technology

Igor Zikratov, Alexander Kuzmin, Vladislav Akimenko, Viktor Niculichev, Lucas Yalansky
ITMO University
Saint-Petersburg, Russia
zikratov@cit.ifmo.ru, akouzmin@altirix.ru, yavlad96i@gmail.com, mail@nikva.ru, yal@tuta.io

***Abstract*—Blockchain is a relatively new technology that has shown a lot of possibilities. It emerged in 2009 as a public ledger of all Bitcoin transactions. Blockchain technology is finding applications in wide range of areas: digital assets and stocks, smart contracts, record keeping, ID systems, cloud storage, ride sharing, etc. We investigate the blockchains' activity in terms of how to store, retrieve and share files in decentralized network.**

I. Introduction

Many people are excited about cryptocurrencies like Bitcoin, but what is even more interesting is the technology that powers it.

A blockchain is essentially a distributed database of records, or public ledger of all transactions or digital events that have been executed and shared among participating parties. Each transaction in the public ledger is verified by consensus of a majority of the participants in the system. Once entered, information can never be erased. The blockchain contains a certain and verifiable record of every single transaction ever made.

Beyond currency, the blockchain can be used in smart contracts, record keeping, ID systems, cloud storage and many other areas.

In our study we decided to look closely at how to ensure data integrity using blockchain technology.

The Clark-Wilson model identified the features of an integrity-secure system. It consists of the following factors: well-formed transactions, separation of duty, authentication, audit, Principle of least privilege, objective control and control over privilege transfer [1].

We are mostly interested in creating our own version of private chain IaaS "Zeppar", that is similar to Storj and BigchainDB, except for the effort of ensuring data integrity in the process. Our model quantifies the importance of several factors that determine data integrity formulated by Clark and Wilson including well-formed transactions, separation of duty, authentication, audit, Principle of least privilege, objective control and control over privilege transfer.

II. Related works

Users of cloud systems commonly assume that if their data is encrypted before outsourcing it to the cloud, it is secure enough. Although encryption is to provide solid confidentiality against internal attacks, it does not protect that data from corruption caused by configuration errors and software bugs. There are two traditional ways of proving the integrity of data outsourced in a remote server. Checking the integrity of data can be by a client or by a third party. The first one is downloading the file and then checking the hash value. In this way, a message authentication code algorithm is used.

MAC algorithms take two inputs, which are a secret key and variable length of data, which produce one output, which is a MAC. In this way this algorithm is run on the client side. After getting a MAC, the data owner outsources those data to the cloud. For checking its integrity, the data owner downloads the outsourced data and then calculates the MAC for it and compares it with the one calculated before outsourcing that data.

By using this method accidental and intentional changes will be detected. Also, by using the key, the authenticity of data will be protected and only the one who has the key can check the data authenticity and integrity. For a large file, downloading and calculating the MAC of the file is an overwhelming process and takes a lot of time. Also, it is not practical since it consumes more bandwidth.

Therefore, there is a need for using a lighter technique, which is calculating the hashing value. The second one is to compute that hash value in the cloud by using a hash tree. In this technique, the hash tree is built from bottom to top where the leaves are the data and parents are also hashed together until the root is reached. The owner of data only stores the root. When the owner needs to check his data, he asks for just root value and compares it with the one he has. This is also to some extent is not practical because computing the hash value of a huge number of values consumes more computation. Sometimes, when the provided service is just storage without computation, the user download the file, the same as in the first case, or send it to third party, which will consume more bandwidth.

Therefore, there is a need to find a way to check data integrity while saving bandwidth and computation power. Remote data auditing, by which the data integrity or correctness of remotely stored data is investigated, has been given more attention recently.

A. Third Party Auditor

Third Party Auditor (TPA) is the person who has the skills and experience to carry out all auditing processes. TPA scheme is used for checking the data integrity. Architecture of third-party auditing their proposed scheme attains data integrity and assures the data owner of the data security. The owner is aware of all his resources on the cloud. Therefore, this scheme guarantees the integrity of data for all owner resources on the cloud.

This scheme involves the data owner in the auditing process. First, TPA uses normal auditing processes. Once they discover any modification to the data, the owner is notified about those changes. The owner checks the logs of the auditing process to validate those changes. If the owner suspects that unusual actions have happened to his data, he can check his data by himself or by another auditor assigned by him.

Therefore, the owner is always tracking any modification to his own data. There is an assigned threshold value that a response from the third party auditor should not exceed. The data owner validates all modifications lesser than or equal to this threshold. If the time exceeds this threshold, the data owner is supposed to do surprise auditing

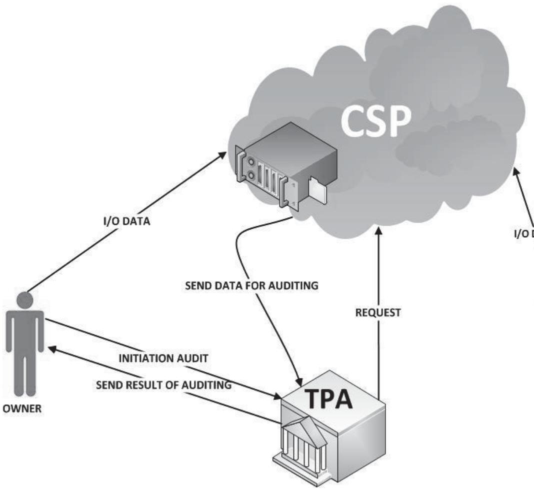

Fig. 1. TPA-method

The disadvantages of this method are the need for third-party communication channel, as well as the exposure of the "man in the middle" attacks. In addition, the emergence of a third party for data processing, this increases the risk of the implementation of a range of threats from hackers [2].

B. Provable Data Possession (PDP) and related methods

Provable Data Possession (PDP) scheme to investigate statically the correctness of the data outsourced to cloud storage without retrieving the data. In the proposed model is to check that data stored in a remote server are still in its possession and that the server has the original data without retrieving it. This model is based on probabilistic proofs by randomly choosing a set of blocks from the server to prove the possession.

They used a RSA-based homomorphic verifiable tag, which is combines tags in order to provide a message that the client can use to prove that the server has specific block regardless of whether the client has access to this specific block or not [3].

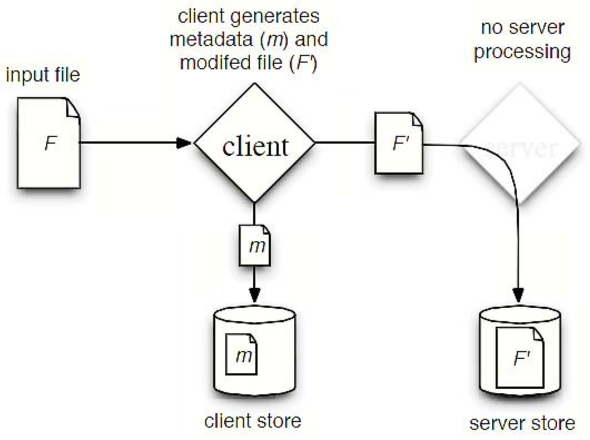

Fig. 2. Pre-process of PDP-protocol

At a pre-process step, before sending files to the cloud, user modifies data and adding metadata, at the same time maintaining in the client metadata repository.

In the future, when checking the fact of the immutability of the file saved in the cloud, the user generates a request to the server to store file metadata comparison to the repository metadata repository located in the client [4].

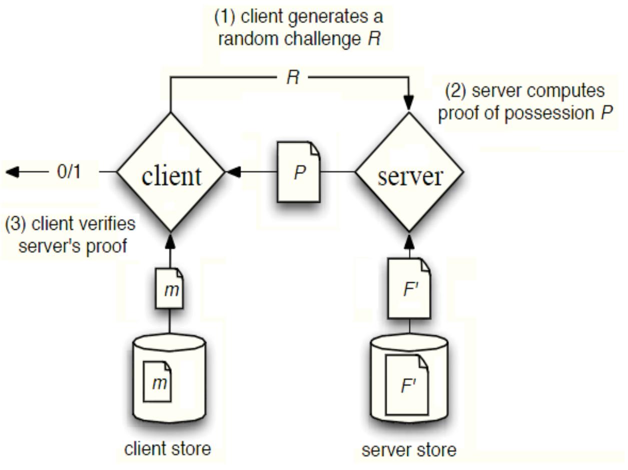
(1) client generates a random challenge R
R
(2) server computes proof of possession P
0/1
client
P
server
(3) client verifies server's proof
m
F'
m
F'
client store
server store

Fig. 3. Checking of PDP-protocol

A further development of the method is the E-PDP, which is different protocol capable of 185 times faster and generate metadata file, in fact, only limited operating speed reading/writing physical storage devices. There are also variations in the method of the PDP, such as Proof of Retrievability (POR), and the method of Prof. Ownership, The main disadvantage of the above methods is the need for the metadata store on the user side, which increases the number of potential targets for an attack. In addition, these methods make it very difficult to ensure the integrity of the dynamic files.

Cloud Storage consists of a set of virtual machines. This can be successfully applied for blockchain.

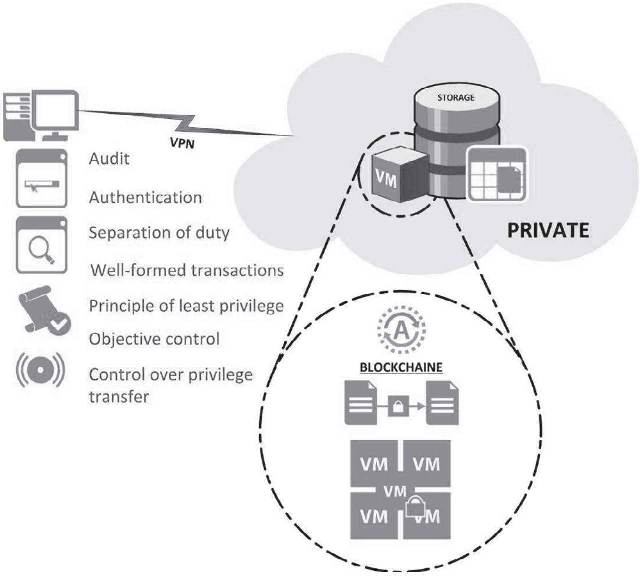

Fig. 4. Blockchain ensure data integrity method in Cloud Storage

Blockchain is a structure composed of blocks each of which are recorded transaction. The box consists of a header and the transaction list. Title block includes a hash, hash of the previous block, the transaction. Transaction, inter alia, contains an attribute within the input link to the transaction with the previous state data. As a result of the hash is irreversible, there is no algorithm for obtaining the desired result, in addition to random search. The node sends the resulting unit is connected to other nodes that test unit. If there are no errors, then the block is considered to be added to the chain, and the next block should include a hash of it.

Eligible units are sent to the network joining in a distributed database blocks. Regularly there are situations when a number of new units in different parts of the distributed network called the previous one and the same unit, ie, the chain may branch blocks. Intentionally or accidentally, you can restrict relaying information about the new blocks (for example, one of the chains can be developed within the local network). In this method the possible build-up of various parallel branches. Each of the new block can occur as the same transaction, or different, included in only one of them. In method of equal complexity and length of that preference is given to the chain, the end of which there was previously a block.

Chain blocks contains the history of ownership, which can be found. Cloud storage generally is a cluster is divided into virtual spaces allocated to each individual user. Functionality of the proposed method consists of two main parts: the part that is responsible for the formation of user transactions and the part that is responsible for maintenance of the transaction block chain. The method is applicable to both static files and dynamic. If the static files are immutable and are available for downloading or rewriting.

Thus, for a static file changes fact serves only its full rewriting. Recording a transaction takes place in a block using the method of concurrent access without locks. Separation of access between threads comes at the expense of atomic operations for each of the file types. There are three types of transactions: the creation, modification, deletion. If part, responsible for the formation of user transactions, finds that the file is modified, the transaction is created in which is specified the current hash of the file and the previous one. Previous hash is taken from the block chain ,. When creating / deleting a file created by the transaction, where the value of the previous / current hash take special values.

Working with dynamic files. The user writes various dynamic files, such as databases. Then he sets up the configuration file of a background process that monitors the change in dynamic files. When you first start a transaction file is created with the original hash file and sent to the server. It then monitors data files for changes. If changes are made unauthorized process that sent the notification to the user. Otherwise, at user-defined intervals updated hashes of the files. Concurrent user access to dynamic files is carried out by internal application modules working with dynamic files.

One of the main components of this method is that the virtual machine is engaged in transaction processing blocks chain. She receives a transaction from client programs, background, checks for validity. Notification of non-valid transactions, the creation of units of valid transactions, the spread of new units through the network of virtual machines.

III. RELEVANCE

Is blockchain relevant?

Blockchain technology is being used in all types of industries. This is not surprising because this technology provides a wide range of different features and amenities.

Some examples are:

- Scaling
- Monitoring the chain in real-time (event-based triggers)
- User isolation
- Momentary asset transfer over the network
- Immutability
- Modular core

Having assessed all the advantages of this technology. it becomes obvious that it can be used to monitor the integrity of files, which is one of the most important tasks of information security.

IV. Overview

Our method is divided into two main parts: the frontend and server. The frontend consists of the UI and programs that implement the services provided by the server. The server is responsible for processing the transactions, creating/transmitting blocks and handling all user access control.

A. Frontend

The user interface is a web application which provides the user the ability to access the provided storage space (uploading/downloading data).

B. Backend

The backend is the core of our system. It is where all of our services that ensure data integrity of the file storage are implemented.

The backend consists of several parts.

1) ABRAXAS

ABRAXAS uses the FileSystemWatcher class to get notifications from the OS about changes in the filesystem. It then creates and sends the transactions based on the data that the OS provides. ABRAXAS creates and sends transactions.

Transactions can be of 4 different types, each one corresponds to a different filesystem operation. The transactions are standardized:

- **Tx Type**: identifies the type of the transaction. **Actual Hash**: current hash of file
- **Prev Hash**: previous hash of file
- **Timestamp**: the exact time and date when the transaction was created in UNIX-time.
- **Previous/Current Path**: previous/current path to file
- **Signature**: digital cryptographic signature of the file owner.

<u>Types of transactions:</u>

a) OnCreated

This type of transaction is formed when a file is created.

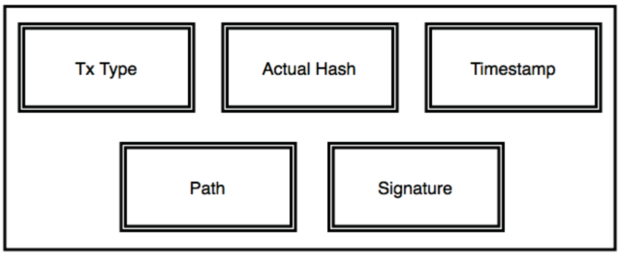

Fig. 5. Transaction scheme of OnCreated type

b) OnChanged

This type of transaction is sent when a file is changed.

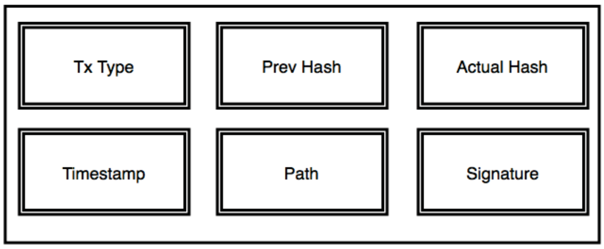

Fig. 6. Transaction scheme of OnChanged type

c) OnDeleted

This type of transaction is sent when a file is deleted.

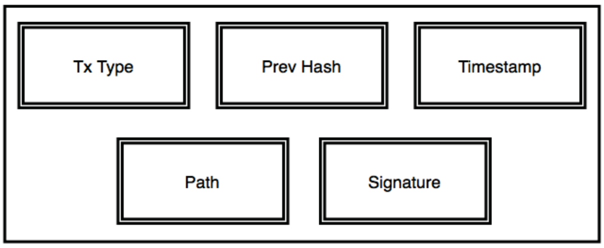

Fig. 7. Transaction scheme of OnDeleted type

d) OnRenamed

This type of transaction is sent when a file is renamed.

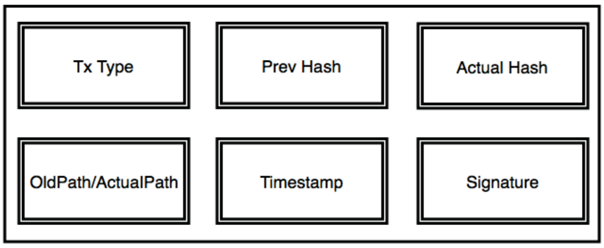

Fig. 8. Transaction scheme of OnRenamed type

2) A worker process processes the transactions and creates blocks out of them. [5]

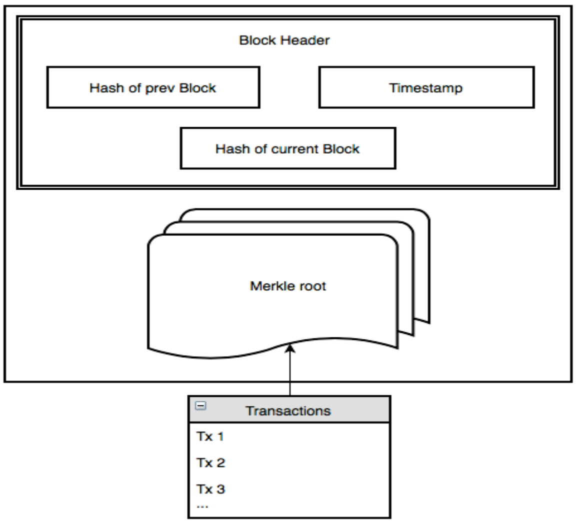

Fig. 9. Scheme of Block

V. Model description

As mentioned earlier, there are two components of the entire client system, is solely responsible for the use and experience of the responsible for all computing processes and ensuring the integrity of file storage. Now in more detail about each of them.

A. User side

The user interface is a web application which provides the user the ability to access the provided storage space (uploading/downloading files).

B. Server side

1) Session management [6]

Session management is handled on the server [7].

Users can authenticate using a login/password pair, a private/pub key pair or even a smart card.

Once authenticated, a session key is created by hashing a random nonce with the username using the GOST R 34.11-2012 (Stribog) algorithm. The session key will be used for all server-side functions until it expires.

2) FileSystemWatcher

Algorithm 1 setWatchers

```
List<FileSystemWatcher> watchers = new List<Filesystemwatcher> ();
configurations = getConfiguration(config)
foreach (Configuration conf in configurations) {
    watchers..add(new FileSystemWatcher(conf))}
return watchers.
```

3) Type of transactions

Ehe basic idea was taken from creation *Secure High-Rate Transaction Processing in Bitcoin* [8]

Algorithm 2 struct Transaction_FileChanged

```
public int OpCode;
public string OldFileHash;
public string NewFileHash;
public DateTime TransactionCreationTime;
public string signature;
```

Algorithm 3 struct Transaction_FileCreated

```
public int OpCode;
public string FileHash;
public DateTime TransactionCreationTime;
public string signature;
public string path;
```

Algorithm 4 struct Transaction_FileDeleted

```
public int OpCode;
public string FileHash;
public DateTime TransactionCreationTime;
public string signature;
```

4) Creation of transactions

Algorithm 5 createTransaction

```
Transaction t = new Transaction();
t.setOldFileHash();
t.setNewFileHash(getHash(e));
t.setTransactionCreationTime();
t.setSignature();
send(t);
```

5) Making block of transactions

Algorithm 6 createBlock

```
Block b = new Block();
foreach (Transaction t in tx) {
    if (t.isValid) {
        b.add(t)}
    else sendnotification();}
```

6) Integrity check (Cross-checking hashes with blockchain). As it mentioned in works [9].

Algorithm 6 checkIntegrity

```
hash = searchInBlockchain(file.Fullpath);
if (hash = gethash(file)) return true;
else return false;
```

VI. Conclusion

Blockchain technology can secure integrity of files stored in the database. It can be achieved through well-formed transactions, authentication, audit that blockchain provides. The amount of possible threats to data integrity can be decreased.

With one of the three main security attributes secured blockchain can be used in order to ensure the remaining two properties of data: confidentiality and availability [8].

There are limitations of using blockchain as it relies on the fact that it is mathematically impossible for a single party to game the system due to lack of needed compute power. However, with the advent of Quantum Computers, the cryptographic keys may be easy enough to crack through brute force approach within a reasonable time. This will destroy blockchain technology [10].

REFERENCES

[1] Clark, D., Wilson, D. A compassion of Commercial and Military Computer Security Policies (1987)

[2] Sultan Aldossary, William Allen. Data Security, Privacy, Availability and Integrity in Cloud Computing: Issues and Current Solutions. (IJACSA*) International Journal of Advanced Computer Science and Applications,Vol*. 7, No. 4, 2016 pp.485-498

[3] Dr.Nedhal A.Al-Saiyd, Nada Sail. Data integrity in Cloud computing security. *Journal of Theoretical and Applied Information Techno*logy. 31.12.2013. Vol. 58 №3 pp. 570-581

[4] Giuseppe Ateniese, Randal Burns. Provable Data Possession at Untrusted Stores. 14th ACM *Conference on Computer and Communications Security*(CCS 2007)

[5] Tsirlov, L. Bases of information security of the automated systems.Short course.Phoenix (2008)

[6] Wilkinson, S., Lowry J. Metadisk: Blockchain-Based Decentralized File Storage. Application (2014)

[7] Sompolinsky, Y., Zohar, A. Secure High-Rate Transaction Processing in Bitcoin (2015)

[8] Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)

[9] Zyskind, G., Nathan, O., Pentland, A. Decentralizing privacy: Using blockchain to protect personal data (2015)

[10] Andy Majot and Roman Yampolskiy, 2015. Global catastrophic risk and security implications of quantum computers. *Futures,* vol. 72 (September), pages 17-26.