International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-4, April 2019

Block Chain Based Decentralized Cloud Storage

G. Abinaya, Preksha Kothari, Alex Pavithran KP, Manasi Biswas, Farheen Khan

***Abstract:** This paper revolves around Block chain technology, a decentralized implementation. Cloud storage is traditionally known for it's centralized system. Here is our take on using Block Chain technology as a decentralized cloud storage system. Wherein we use resiliency attacks to display the benefits of Block chain based decentralized cloud storage by performing key operations on our proposed decentralized structure. This work aims to construct a model of decentralized data storage technology based on Block chain with the purpose to improve the security and confidentiality of information. Therefore, in this paper we will analyze the existing centralized platform for data storage for the shortcomings, from which it is possible to give by building decentralized data storage on the basis of Block chain technology.*

***Keyword:** Block chain, Decentralized Cloud Storage, Merkle tree, Resiliency.*

I. INTRODUCTION

Blockchain - a distributed database, in which you can store any data or transactions. In blockchain stores information across a network of personal computers, and the result is not only decentralized, but the allocated space. Consequently, neither the company nor the people, nor any other trusted party is not the owner of the network. All people can use the system, and thus to maintain its operation, so one person is very difficult to crack or completely destroy the network [2]. People are using their personal computers to store bundles of other user data, called blocks ("blocks") in the chronological chain ("chain"), hence the name "blockchain" literally "a chain of blocks." Blockchain-revolution can be divided into 3 categories: 1 Blockchain. 0 (cryptocurrency bitcoin), 2.0 (smart contracts) and 3.0 (the application). Bitcoin - is both a digital cash, currency and online payment system in which encryption methods provide control of aggregate monetary units and the confirmation of the transfer of funds, work without intermediaries and national central banks. Bitcoin was first established 9 years ago, January 9, 2008, to be precise an unknown person or group of people under the pseudonym Satoshi Nakamoto. Payments made in a decentralized virtual currency of any cryptocurrency like Bitcoin is written in such a public journal entries stored in the shared network on users' computers this cryptocurrency and continuously available for viewing. Thus, any transaction performed cryptocurrency impossible to alter or forge. Blockchain, At this point, if the user is interested in the "cloud" storage of files he has three options: 1) download the data to your own server 2) to take advantage of centralized services such as Google Drive or Dropbox, or 3) a network for decentralized storage of files, like Freenet. [4]. Each of these options have their drawbacks; at first too high costs of customization and technical support, the second relies on an intermediary service owner, and often expensive, the third variant is characterized by a low rate service and is limited by the amount of storage space as well as the users of the service provided (or not at all) available memory for use at their discretion. Thanks to modern advanced information technology there is a fourth option, which will provide more space for storing data and high-quality service in a decentralized environment. This work is intended to prove the hypothesis that the option for data storage is the most secure. It is assumed that the data store based on technology blockchain invulnerable to attempts to disclosure of confidential information, as all data is protected using cryptographic techniques. Cryptographic encryption algorithms ensure high security of data against attempts of forgery or alteration of their content. Another undoubted advantage of decentralized storage in terms of security is the fact that the data is spread to be divided into parts of the network and encrypted on the client side by cryptographic algorithms. Consequently, data all users are distributed across the network in an encrypted form in multiple copies in the safe from hacking free-space computer hard drives of people around the world [5]. This paper includes a study of such a distributed data store, wherein the data storage is performed by entering into a contract between intelligent network nodes. Users in such a decentralized environment fall into two categories: host, renting storage space, and user-farmer, providing space of their own hard drive. For the provision of free farmer node receives the award from host-tenant as cryptocurrency

The objectives of the study

- Examine the scientific literature on the technology Blockchain and storage;
- To analyze the security methods to ensure data in a decentralized environment to blockchain;
- Develop a model of traditional and decentralized storage;
- A comparative analysis of conventional and decentralized cloud storage based on blockchain;

Manuscript published on 30 April 2019.
* Correspondence Author (s)
G. Abinaya*, Department of CSE DEPT, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.
Preksha Kothari, CSE DEPT, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.
Alex Pavithran Kp, CSE DEPT, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.
Manasi Biswas, CSE DEPT, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.
Farheen Khan, CSE DEPT, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.

- To develop a model of interaction scenarios nodes in the form of diagrams in order UML notation

The object of research this work is to model a data warehouse that provides the highest possible data security and availability.

II. EXISTING SYSTEM (CENTRALIZED CLOUD STORAGE)

A. The Vulnerability of Modern Cloud Storage

When it comes to the confidentiality of data stored in the cloud, there are two potential problems. The first problem, the authors of the book released, is to control access to data which consists of authentication and authorization [8]. As a rule, suppliers of "cloud" services usually use weak user authentication mechanisms (eg. Login and password) and authorization control, which is not sufficiently secure. The second problem is that often people are concerned about the "cloud" services raises the question of how the data is stored in the cloud is actually protected if they are encrypted and, if so, what encryption algorithm and to what extent it is reliableIf the cloud provider encrypts the data, then the question is the degree of reliability of the algorithm used. Not all encryption algorithms are created equal, and not all provide sufficient information security. symmetric encryption is used for cloud storage, which uses the same key for encryption and decryption (Figure 1.) This type of encryption has the highest bulk data encryption speed. With symmetric encryption of data on the level of security is also affected by the key length, and key management. The larger the key, the higher the security level, but at the same time with an increase in the key length is increased and the intensity calculations, which may strain capacity of computer processors.

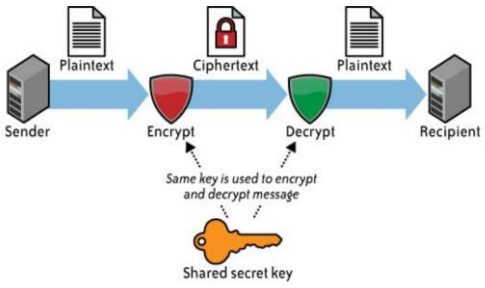

Figure 1

At an international conference on information security (RSA conference) in March 2016 the organization Cloud Security Alliance (CSA) identified 12 of the most serious threats to the security of cloud [10]:

1. sdata Leakage
2. Hacking accounts and authentication bypass
3. Theft accounts
4. The vulnerability of the systems used
5. Theft accounts
6. Malicious Insiders
7. targeted cyber attacks
8. Permanent data loss
9. lack of awareness
10. Abuse of cloud services
11. D DoS-attacks
12. Collaborative technologies, general risks

CSA explains the appeal of cloud-based services for attackers that today more and more data is transferred to the cloud. According to experts of information security organization CSA data leakage and other threats are caused by weak authentication, weak passwords, or wrong key and certificate management. Solving security authentication problems can be one-time passwords, such as authentication using the phone or smart card (token). In connection with the indispensability of the user interface in the cloud infrastructure, security and availability of cloud-based system depends on the degree of maturity of access control, and encryption in the API. According to CSA in the case of cooperation with a third party that uses a proprietary API that significantly increase the risks because it requires the transfer of user information, such as username and password. CSA emphasizes the importance of the use of encryption mechanisms in the storage systems. CSA at the conference also raises the issue of permanent data loss, which is becoming more acute because of the volatile backup capabilities (eg. Due to incomplete uptime). The current model of cloud storage, which is performed through centralized institutions, is not safe from the viewpoint of problems of confidentiality, integrity and availability. Cloud storage are faced with a growing number of vulnerabilities, which can be solved with the help of blockchain technology. This paper discusses a possible solution to these problems through research and development of a decentralized model based on cloud storage Blockchain technology.

III. PROPOSED SYSTEM (DECENTRALIZED SYSTEM)

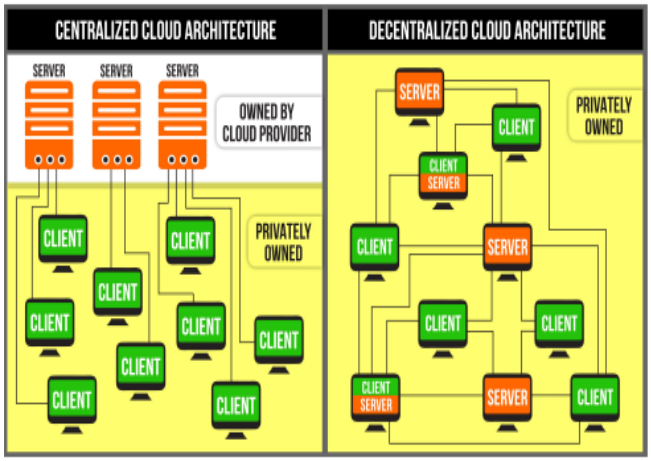

Figure 2 **Decentralized storage on block chain**

The distributed network known as peer to peer (ad-hoc) network if network members provide and share the hardware resources. Concept model of decentralized P2P storage involves the creation of an ad hoc network of users by providing its own free storage space. In order to understand, to understand why much safer to store data in a decentralized repository, based on blockchain technology, it is important to consider what iteration of files held after download.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

To ensure maximum data security the following steps must be completed:

A. **Encrypting files.** Once the file is encrypted (the most commonly used SHA-256), a unique identifier and method for detecting unauthorized access to files is its hash. Every iteration of the file changes its hash thus possible to check files without having direct access to it.

B. **Division into parts.** Encrypted files are divided into pieces (shards), or multiple files are combined to create a single shard.

C. **File Distribution.** Files randomly distributed across the network, along with 3 copies of each shard (3 copies of files stored - is the industry standard)

The process of dividing the ordered file in the scheme below:

The process of dividing the encrypted file is the most reliable mechanism of confidentiality, integrity and availability. The data must be encrypted on the client side to the fission process. Thus, the contents of the file is not available any cloud services provider or owner of the computer on which the data is stored. The user has full control over the private key, and thus is the only one with access to the files. It is important to note that the client can also verify the authenticity and integrity of the file by using its hash. File hash, along with the location of 3 copies of the files stored in blockchain immediately after downloading the file to the network. With the growth of the kit of parts (shard) the network becomes much more difficult to find any given set of parts of the files, without prior knowledge of their locations. This means that data security is proportional to the square of the size of the network [16]. Creating 3 copies of each piece of data (shards) is the most effective way of ensuring data availability. Therefore, it can be argued that the availability of data in a decentralized peer to peer environment is proportional to the number of nodes stored the data.

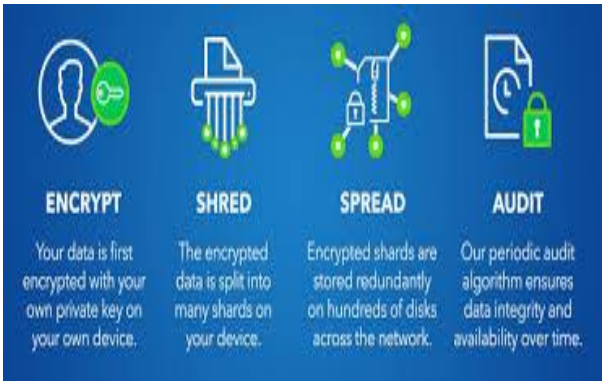

Figure 3

IV. PROPOSED METHODOLOGY

A. Proposed Structure

a. Merkle Tree

With the increasing use of decentralized storage model to blockchain, a transition to a blockchain that stores the root of the Merkle tree. Merkle tree - a special way of storing large amounts of data, which combines pieces of data into blocks, then hashes the blocks, then get as a result of new hashes are combined and hashed until until a root hash. That is the root hash and to be kept blockchain. Merkle tree essentially represents a binary tree, where each unit always consists of two blocks or hashes [18]. Using this blockchain that keeps the root element of a large number of data elements allow proof of the existence of millions of files of the same transaction. Diagram - blockchain storing in itself, the root element Merkle tree shown in Figure

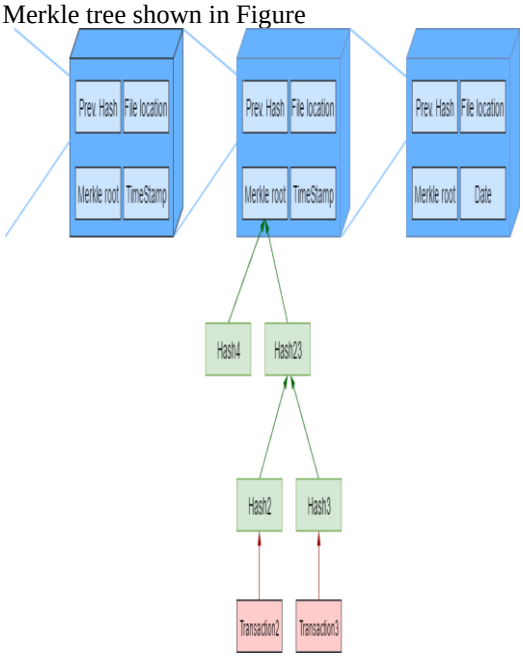

Figure 4. Blockchain stored root element Merkle tree.

B. Encryption Algorithm

- Our encryption choice is authenticated encryption, with support for both the AES-GCM cipher and the Salsa20 and Poly1305 combination NaCl calls “Secretbox”. Authenticated encryption is used so that the user can know if anything has tampered with the data.
- Data is encrypted in blocks of small batches of stripes, recommended to be 4KB or less .
- While the same encryption key is used for every encryption batch in a segment, segments may have different encryption keys.
- However, the nonce for each encryption batch must be monotonically incrementing from the previous batch throughout the entire segment. The nonce wraps around to 0 if the counter reaches the maximum representable nonce.
- To prevent reordering attacks, the starting nonce of each segment is deterministically chosen based on the segment number.
- When multiple segments are uploaded in parallel, such as in the case of Amazon S3’s multipart-upload feature, the starting nonce for each segment can be calculated from the starting nonce of the file and the segment number.
- This scheme protects the content of the data from the storage node housing the data. The data owner retains complete control over the encryption key, and thus over access to the data. Paths are also encrypted.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

- Like BIP32, the encryption is hierarchical and deterministic, and each path component is encrypted separately. To explain how we do this, we start with a scheme for determining a secret value for each path component.
- **Let's say a given path p has unencrypted path components p1 ,p2, . . . ,pn and we want to determine an encrypted path e with path components e1 , e2, . . . , en.**
- **We assume a predetermined root secret, s0. This root secret is chosen by the user and, like all other encryption secrets, never leaves the client computer. We recursively define si = HMAC(si–1,pi). A key K(si) .**
- **Concrete implementation can be deterministically generated from si . We then define the encrypted path component ei = enc(K(si–1),pi), such that the new path e is e1 , e2, . . . , en. HMAC-SHA256 or HMACSHA512 are used for key derivations. This construction allows a client to share access to some subtree of the path without access to its parents or other paths of the same depth. For example, suppose a client would like to share access to all paths with the same prefix p1 ,p2,p3 with another client. The client would give the other client e1 , e2, e3 and s3**.
- This allows the client to decrypt and access any arbitrary e4, as K(s3) is known to them, without allowing the client to decrypt e3 or earlier.
- More generally in this case, the client could decrypt and access any arbitrary ei , if and only if i > 3. Path encryption is enabled by default but is otherwise optional, as encrypted paths make efficient sorted path listing challenging.
- When path encryption is in use (a per bucket feature), objects are sorted by their encrypted path name, which is deterministic but otherwise relatively unhelpful when the client application is interested in sorted, unencrypted paths.
- **For this reason, users can opt out of path encryption. When path encryption is disabled, unencrypted paths are only revealed to the user's chosen Satellite, but not to the storage nodes. Storage nodes continue to have no information about the path and metadata of the pieces they store.**

i. HMAC

$$\mathrm{HMAC}(K,m) = \mathrm{H}\left(\left(K' \oplus opad\right) \| \mathrm{H}\left(\left(K' \oplus ipad\right) \| m\right)\right)$$

$$K' = \begin{cases} \mathrm{H}(K) & K \text{ is larger than block size} \\ K & \text{otherwise} \end{cases}$$

Where

H is a cryptographic hash function

m is the message to be authenticated

K is the secret key

K' is a block-sized key derived from the secret key, *K*; either by padding to the right with 0s up to the block size, or by hashing down to less than the block size first and then padding to the right with zeros

‖ denotes concatenation

⊕ denotes bitwise exclusive or (XOR)

opad is the block-sized outer padding, consisting of repeated bytes valued 0x5c

ipad is the block-sized inner padding, consisting of repeated bytes valued 0x3

Implementation

```
Function hmac
  Inputs:
    key:       Bytes     //array of bytes
    message:   Bytes     //array of bytes to be hashed
    hash:      Function  //the hash function to use (e.g. SHA-1)
    blockSize: Integer   //the block size of the underlying hash function (e.g. 64 bytes for SHA-1)
    outputSize: Integer  //the output size of the underlying hash function (e.g. 20 bytes for SHA-1)

  //Keys longer than blockSize are shortened by hashing them
  if (length(key) > blockSize) then
    key ← hash(key) //Key becomes outputSize bytes long

  //Keys shorter than blockSize are padded to blockSize by padding with zeros on the right
  if (length(key) < blockSize) then
    key ← Pad(key, blockSize) //pad key with zeros to make it blockSize bytes long

  o_key_pad = key xor [0x5c * blockSize]   //Outer padded key
  i_key_pad = key xor [0x36 * blockSize]   //Inner padded key

  return hash(o_key_pad ‖ hash(i_key_pad ‖ message)) //Where ‖ is concatenation
```

ii. SHA 256

- The initial version of the SHA-256 algorithm was created by the US National Security Agency in the spring of 2002.
- A few months later, the national metrological University published the newly-announced encryption Protocol in the FIPS PUB 180-2 secure data processing standard adopted at the Federal level.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

- In the winter of 2004 it was replenished with the second version of the algorithm.

Over the next 3 years, the NSA issued a second-generation Sha patent under Royalty-free license. This is what gave rise to the use of technology in civilian areas.

This Protocol works with information broken down into pieces of 512 bits (or 64 bytes in other words). It produces its cryptographic "mixing" and then issues a 256-bit hash code. The algorithm includes a relatively simple round, which is repeated 64 times.

Specifications

SHA-256 has quite good technical parameters:

- **block size indicator** (byte): 64.
- **maximum allowed message length** (bytes): 33.
- **characteristics of the message digest size** (bytes): 32.
- **the standard word size** (bytes): 4.
- **internal position length parameter** (bytes): 32.
- **the number of iterations in one cycle**: 64.
- **the speed achieved by the Protocol** (MiB/s): approximately 140.

The **Sha-256 algorithm** is based on the Merkle-Damgard construction method, according to which the initial index is divided into blocks immediately after the change is made, and those, in turn, into 16 words.

SHA-256 is used in several different parts of the Bitcoin network:

1. Mining uses **SHA-256 as the proof-of-work algorithm**.
2. SHA-256 is used in the creation of bitcoin addresses to improve security and privacy.

C. Blockchain as implementation

Decentralized storage on blockchain as a tool to ensure the confidentiality, integrity and availability of data

The main advantage of using blockchain in storage - storage is the formation of contracts is not based on trust. Unlike data storage on blockchain traditional cloud storage requires a user to trust service providers, sometimes in the form of an additional payment for the provision of better protection. Thus, the provider is the owner of files and completely takes control of them. The current model of cloud storage, which operates through a centralized institutions that are entrusted with personal information is unreliable due to the fact that the information scammers can replicate and destroy data stored on centralized servers. Another vulnerability in the traditional storage model - a payment mechanisms, which are neither private nor secure, as the majority of online payment technologies keep the information on the payer and recipient. Below is a diagram vulnerability traditional model data store. Considering the drawing model, you will notice that there are some sensitive points for attacks c purpose of obtaining unauthorized access to personal data.Problem of having easily identifiable central points of attack of the traditional model of the cloud of data storage can be solved by decentralization. There are three reasons for decentralization is the most appropriate in the field of data storage:

A. **fault tolerance** - due to the fact that decentralized systems do not rely on the central nervous system, and many of the individual components, they are less prone to random failures.
B. **Resistant** - decentralized systems is much more expensive to attack, destroy, since there is no sensitive center point, which is cheaper and more attractive for the attack.
C. **Resistance to the conspiracy** - in a decentralized system is much more difficult to organize the collusion between its members for personal gain, whereas in various centralized systems (government, large companies) of its members are subject to collusion, which carry them to a personal benefit, but harm the system.

Confidentiality Due to the fact that an attacker cannot gain access to the contents of a file even if the existence of the file is known, the data is protected from unauthorized access. Because the file is encrypted before transfer to the network, a model of decentralized storage is resistant to attacks by a trusted third party, such storage may therefore be used in any network, including open and unencrypted networks Wi-Fi. The figure below shows an exemplary pattern diagram of decentralized data storage based blockchain. Immediately noticeable that in contrast to the traditional model, decentralized model is not sensitive points of possible attacks in connection with:

- Encrypted transfer of data to the network;
- Dividing data into pieces;
- Using transaction registry entries - blockchain recording of transactions with the data;
- Abandonment of a trusted third party who has control over user data.

Availability and integrity. While data security is maintained by encrypting the user side, while the data integrity and availability checking supported on reconstitution tethers (proof-of-retrievability) [19].
Checking for recovery is the most reliable method of checking the state of mathematical data "directly from the cloud." Evidence of data recoverability guarantee the existence of certain parts of the file on the remote host. In the proof of the possession of one node to another provides a unique hash of the stored file with the help of which is checked whether there had any iteration of the file [20]. Thus, the second node may ensure integrity and availability of the file. If the data source does not pass by recoverability, data can be restored from the next node, Concept model of decentralized storage on a mandatory basis should contain the above-described evidence in the form of periodic inspections of pieces of data to provide information to the network nodes that the file is available and has not been altered. is a schematic explanation of the principle of verification of data on the subject of availability and integrity.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

Member of the network, sending data for storage in a decentralized network, it can be assured of their safety, as a permanent audit of availability of units and the integrity of data files provides complete control over the data.

D. Proof for Resiliency of the Proposed Systems Work Flow on Key Activities

Resiliency Decentralized Storage Based Blockchain

Since the relationship of the decentralized network nodes is not based on trust, and this hidden information as a set of pieces of data cannot be transferred to a third-party site is unreliable, the owners of the data required to conclude contracts smart, pre-processing of the data, to issue and verify the audits, provide payments, etc. Despite the fact that for all of these functions require high performance and significant infrastructure, the most important is the fact that the decentralized storage on blockchain provides full fault tolerance. This section presents an analysis of the interactions of the decentralized network nodes in form of diagrams the sequence of actions in UML notation.

i) *Scheme interaction of nodes in the decentralized*

a(i). Repository

In order to test the hypothesis about the advantage of decentralized storage on blockchain over traditional cloud storage me was analyzed and developed the source code model nodes behavior of algorithms in such a network in the form of sequence diagrams in UML notation. In this section, 5units interaction scenarios will be described:

- Creating a storage contract
- Performing storage contract
- Formation files redundancy (backup)
- Checking the integrity and availability of data (audit files)
- request for Information

The diagrams unit provides storage space - Farmer, and the node borrowing - Renter. Communication between the decentralized network nodes is performed via the user interface in Scheme - UI (User Interface). The following are the interaction model nodes as sequence diagrams reflecting approval scenario storage and execution of contracts, the formation of backup files, and data files audit request. It should be noted that the communication between the nodes ordered in the diagram below, occur automatically in the decentralized system through the user interface. From the node, borrowing space, registration is required in the system, means getting the private key to sign, and send the data to the storage after a successful search for "neighbors" and at the conclusion of the contract system. The node which is the storage location, should, in turn, be registered in the system and provide a place. Schemes designed to prove the safety and reliability of decentralized data storage system using blockchain as metadata repository and user interface for communication between nodes.

a(ii). Creating a storage Contract

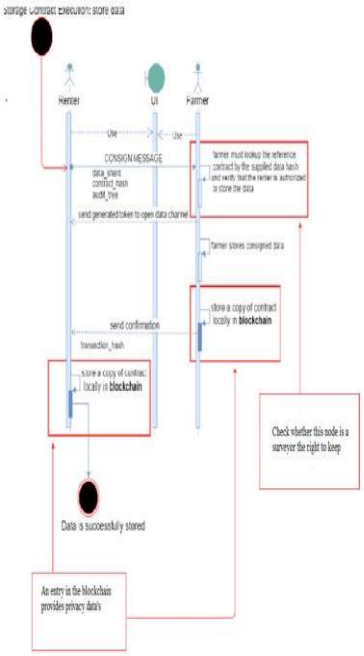

Figure5. Matching storage contract.

Creating storage unit contract is finding, which will satisfy the proposed node renting conditions of the contract. To start from the node, the borrowing of storage space, a message is sent with the aim of publishing contract. Upon receipt of a request for the publication of "Farmer" contract performs the following actions:

1. Checks whether the already received request for publication (Publish message) from a given node, if yes - Error message will seem node wishing to store data.

2. If-borrowing node does not send a request for a contract previously and contract terms are satisfactory "Farmer" sends a message to the contract offer (Offer message). The farmer can also request to change the terms of the contract, and then to poison the sentence. The operation takes place until the terms of the contract will not be acceptable for both sides.

In agreeing the contract, the farmer sends the following parameters Offer message: own ID and signature, as well as on the destination of the payment information ("Renter" pay farmers after each audit data).

If the terms of the contract indicated in the message Offer are satisfactory for "tenant" he sends his signature, waiting for the signature of the farmer, and after which both parties retain a copy of the contract in blockchain.

In contrast to the decentralized storage, traditional storage, as a rule, does not carry out any operations related to checking whether data already placed in storage. Also, in today's cloud storage there is no public register of transaction records like blockchain, which is responsible for maintaining the confidentiality of the data.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

a(iii). Performing storage contract (transfer of data for storage)

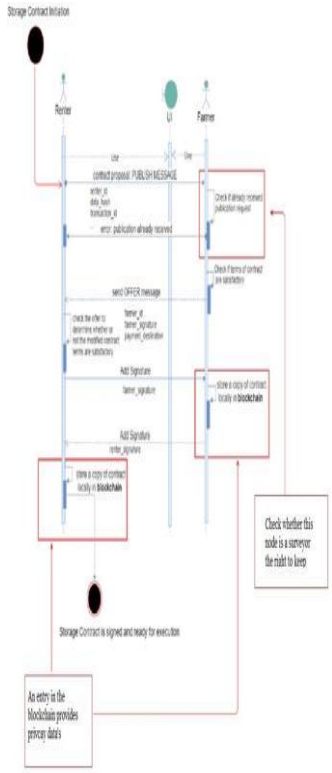

Figure6. Data transfer to storage.

Performing storage contract is essentially a transfer of data from the storage node-node tenant-lessee. As soon as the contract is signed by both parties (Figure 8) from the host-tenant sends a message that includes a link to the contract and the data itself sent to storage. Also, in the transmission message (Consign) contained audit tree for recording each hashes verify data integrity. Upon receipt of the above message "farmer" checks all hashes to make sure that the host-tenant has the right to store this data. After checking node borrows comes generated cryptographic token (key) to open a channel for direct transmission of data to the storage. The farmer keeps at the data and automatically records the transaction in the hash blockchain, and then sends a confirmation message to the node-tenant containing a hash of the transaction. "Rent" stores the hash transaction blockchain then can be considered successful sending of data.

Traditional cloud storage does not imply the validation rules for storage as data is stored in one place and there is no such thing as a storage contract.

a(iv). Redundancy files(Back up)

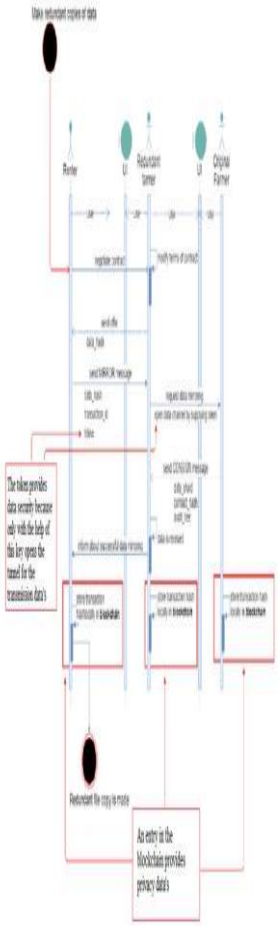

Figure 7. Create backup files.

Maintaining redundancy, that is, backup data ensures fault tolerance. For host-tenant desirable to have multiple backups of each shard in a network with several farmers in the event that one of the farmers under contract leaves the network, lose data or somehow violates the terms of the storage contract. Figure 10. The scenario data mirroring from the original tenant farmer excessive. Before going mirroring itself, the knot, borrowing the storage place must agree on the terms of the contract with an excess storage of farming. Once the contract is signed by both parties, rent a node sends a message (MIRROR) excessive farmer of the need to copy data from another farmer. After that, by the excess of the farmer is sent a request to transfer data to another farmer. As a result, the source transmits data excessive farmer farmer in the form of message transmission (CONSIGN), comprising shard data hash tree and contract audit elements. Upon receipt of the data set in the form of shards excess farmer sends an acknowledgment to the node-tenant successful mirroring data. Once the hash of the backup transaction data recorded locally in blockchain each side, creating a redundant process is completed. Upon receipt of the data set in the form of shards excess farmer sends an acknowledgment to the node-tenant successful mirroring data.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

Once the hash of the backup transaction data recorded locally in blockchain each side, creating a redundant process is completed. Upon receipt of the data set in the form of shards excess farmer sends an acknowledgment to the node-tenant successful mirroring data. Once the hash of the backup transaction data recorded locally in blockchain each side, creating a redundant process is completed.

Not all cloud storage creates additional copies of files in the event of failure of the server that stores the master data. Also, as a rule, modern cloud storage is meant sharing files links, so that any user who has the link can access the file. In contrast to the decentralized storage, traditional storage systems, there is no such thing as the use of a particular cryptographic key for data.

a(v). Checking the Integrity and Availability of files

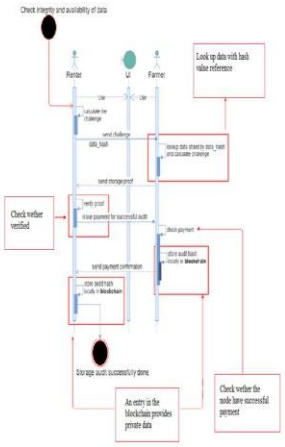

Figure 8. The audit files.

Conduct an audit of the stored files is requested from the farmer's proof that he still fulfills the conditions of the contract, without the need of the whole shards data. To do this, the part of the farmer rents node provided one of the previously calculated secret tasks. Therefore, a farmer receiving a request to audit data and compares the hash obtained conducts calculation tasks using it. As a result, the farmer sends the hash, which should correspond to that which was provided in the form of an audit element tree, which is so-called "call» («challenge»). If the file has not been subjected to any iteration, its hash is unchanged and therefore the problem will be solved properly. Upon receipt of solving the problem, node-tenant checks the received hash merkle audit tree and sends the farmer payment for successfully completed verification of data integrity. As soon as the farmer receives a payment, it stores a hash transaction audit blockchain and sends rents node confirmation in the form of a hash of the audit, which, in turn, will be recorded in blockchain host-tenant.

Traditional cloud storage provides users regularly check the integrity of files, as a decentralized store. As noted before, the data request to the cloud modern systems occurs on a link that is inherently insecure, since every person has a given link may access the data. The decentralized storage on blockchain retrieval of data is carried out exclusively on their cryptographic hash, and only node-tenant has access to their own files.

a(vi). Request for Information

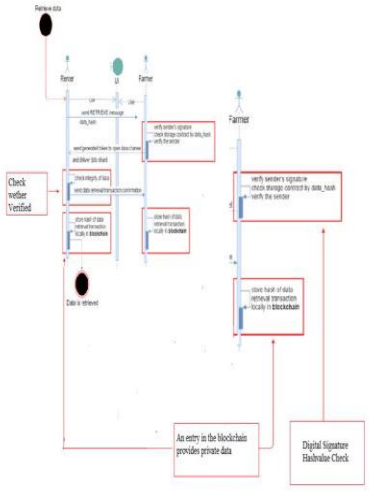

Figure 9. Query data.

To restore the data from the host-tenant sends a message about the data returned (RETRIEVE), containing a hash of the recoverable data. Upon receipt of such a message farmer checks the signature of the sender, check the storage contract on the resulting hash and thereby determines whether the sender is entitled to obtain the required data. After checking the message, the farmer sends a cryptographic token (key) to open a channel for data and directly transfers the required shard. After receiving the data, rent node checks the integrity of the data and sends a message confirming the successful recovery of data to the farmer. The data recovery process may be considered complete after the hash of the transaction recorded locally on both sides.

Doing modeling the relationship of nodes in the decentralized network can be seen in the following:

- Check hashes transactions and data at each step of the scenarios provides complete data privacy. Since the hash is unique, only the true owner of the file can have access to them.
- Record each transaction and changes the data in blockchain ensures that no one iteration of a file can not pass unnoticed.
- Through regular audits user file can be confident in the integrity of its own files.

V.RESULTS AND ANALYSIS

Under this section, an analysis of the traditional cloud storage Dropbox and decentralized data storage based on blockchain technology, Storj. The purpose of this analysis is proof that decentralized storage based on blockchain technology provides greater security of stored user data on three key aspects: confidentiality, availability and integrity.

A. Comparison criteria:

Peer connection - This criterion describes the technical aspects of the system P2P, since there are many different forms of how peers can be connected in P2P networks.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

Another important aspect of the comparison is the physical storage location data in the network. Under this criterion it will also be described as a network of users communicate and what opportunities exist Communications.

Encryption - The presence of encryption is an important criterion for the user when choosing a data warehouse, as it cryptographic algorithms ensure complete confidentiality of customer data.

Trust and integrity. - This criterion defines the level of trust between peers, as well as the existence of mechanisms to verify the integrity of the data for any unauthorized iterations files. Payment plan - The system is cheaper, the more likely that the user will choose the system.

Justice - Distribution of files between peers should be made in a balanced manner, so as overloading files from one network node is undesirable because it can lead to a problem of a single point of failure.

Resiliency and availability – The system must handle network failures and prevent data loss in the event of any of the network nodes. Also, users will always be interested in the immediate receipt of the file the long wait is not desirable when it comes to the storage of personal data

Provision of services -The criterion characterizes ease using user interface, and the ability to build their applications on top of the system for advanced users.

Availability source - *This* aspect is most important is for application developers, which in the case of open source can make changes in the program and thus contribute to future releases of the system.

User base - The number of users, as a rule, is the first indicator of the quality system to potential customers. Moreover, the system is open source, and a large user base inspires more confidence than a system with a small number of users. However, there is a possibility that a large user base makes the system more attractive point of attack for hackers.

1.Condition - The system should be available for use or at least affordable in the near future

Centralized Vs Decentralized

	DropBox	**Storj**
Peer Connection	Not	Unstructured
Encryption	Files are encrypted, Locks are stored in centralized server	Through encryption type determined user
Trust and Integrity	No mechanism insuring integrity. Members rely on links to other user at their discreation.	Audit “heartbeat”
The fee for use	Free/Pay	Pay
Justice	Does not apply to this system fair	Does apply due to distribution files evidence
Resiliency and Availability	Server is an vulnerable place(sensitive point of failure)	Create three backups file recovery process by the network replication
Provision of Services	Application/ Website	Application/Website
Availability Source	Closed	Open
User Base	>300 million	>3 million
Status	Available	Available

B. Implementation risks

Further at section will describes the risks introduction decentralized storage on blockchain in terms of possible attacks. Consequently, the following attack, which is a threat to the decentralized systems:

i) «Sybil» Attack

It is known that other peer decentralized distributed systems are particularly vulnerable to attacks sybil type [41]. The bottom line of attack is to create such a large number of nodes that will grab or delete messages in the network. The nodes in a decentralized network of neighbors, usually selected from a uniform distribution on a particular ID pool, and most messages are sent at least three neighbors. If an attacker controls 50% of the network nodes, he successfully isolates only 12.5% honest nodes [16]. While reliability and performance in this case will worsen, the network will operate as long as a large part of the network will not contain malicious components.

ii) «Google» attack.

The term «Google attack" was first coined by the developer of the project Bitcoin Core Peter Todd (Peter Todd) to describe a coordinated attack on the large corporation or organization network, which controls a large amount of computing power or storage space. Google attack is hypothetical option Sybil attack, carried out with the huge resources of the organization. Google has been taken as an example, which currently has the greatest number of data stored in the "cloud." According to the sources [27], [28] and [29] the Google at the moment keeps about 8000 PB data. Concept decentralized storage introduces the concept of user space or unused space collective consumers. According to the source [27] storage space in a decentralized network can increase to 250,000 PB when users start to use this system as a major cloud storage system. But even if such a large company like Google will suspend all of its services for the attack on the network, it is impossible to surpass the resources of the user space, as it has a prospect several times the space currently Google. Despite this, the threat of «Google attack» and there is a risk for any decentralized storage systems, since it is difficult to predict the actions of the attacker having more resources.

iii) Attack «eclipse» (Eclipse)

With this attack, the attacker tries to isolate a node or set of nodes so that all outbound connections or communications concerned malicious nodes attack *Eclipse* are difficult to identify,since malicious nodes, usually,operating normally, only the "eclipsing", that is hiding some important messages or information. This problem in modern storage systems, decentralized as SIA and Storj solved by using as the hashes of public keys to identify nodes. [16], [42]. To "eclipse" any node in the network, an attacker must re-generate the key pair, until it finds three keys whose hashes will be closer to the destination node than any other diligent uzelsosed.

The complexity of this attack is proportional to the number of nodes in the network, as in essence resembles the work of recoverability of evidence [16]. Consequently, the best way to protect against attacks such as *Eclipse* It is to increase the number of nodes in the network. It is becoming too expensive for a large network attack "eclipse" [15]

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

C. Sharding

The proposed Decentralized network can further benefit from sharding. Sharding describes the procedure of splitting a file into multiple fractions. This step is performed by the client and is optional, yet in many cases recommended. Figure illustrates the sharding process with a file that is being encrypted before it is sharded.

i) Advantages of sharding files

Sharding files brings two main advantage: privacy and performance. The uploaders privacy is increased, because none of the involved farmers stores any entire file, but just a part of it. In this sense, even if a malicious farmer would somehow achieve to decrypt the uploaded data, the farmer would still only be able to read part of the original file, not knowing where the other parts of the file are stored. The performance of the data transfer (uploads as well as downloads) can be positively influenced by uploading shards simultaneously to different farmers. In this case, the transfer speed is equal to the transfer speed of the slowest farmer multiplied by the size of the shards

ii) Implementation of Sharding

(1) The encrypting-first sequence

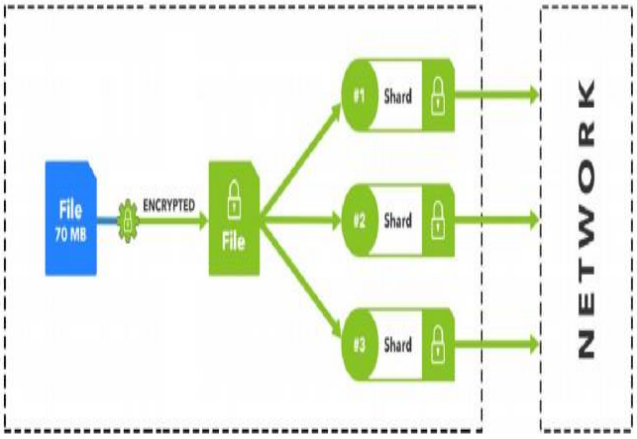

Figure 10.

(2) The sharding-first sequence

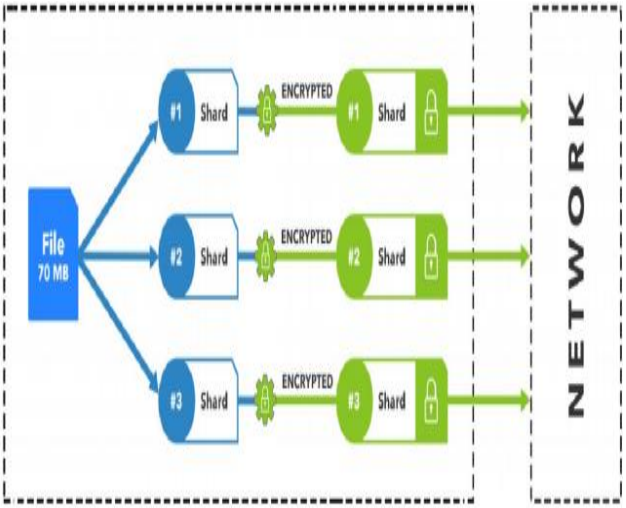

Figure 11.

iii) Performance measure of Sharding

Comparison of performances between unsharded and sharded upload in a Decentralized cloud storage

FILE UPLOADED ENTIRELY				FILE UPLOAD AS SHARDS			
No.	file size	transfer	time	No.	shard size	transfer	time
1	32 MB	10 Mbit/s	3.2 s.	1	8 MB	10 Mbit/s	0.8 s.
				2	8 MB	25 Mbit/s	0.32 s.
				3	8 MB	4 Mbit/s	2 s.
				4	8 MB	5 Mbit/s	1.6 s.
Total			3.2 seconds	Total			2 seconds

Comparing performances between normally sharded and highly sharded uploads

REASONABLE NUMBER OF SHARDS				INAPPROPRIATE NUMBER OF SHARDS			
No.	file size	transfer	time	No.	shard size	transfer	time
1	8 MB	10 Mbit/s	0.8 s.	1	0.1 MB	10 Mbit/s	0.01 s.
2	8 MB	25 Mbit/s	0.32 s.	2	0.1 MB	4 Mbit/s	0.025 s.
3	8 MB	4 Mbit/s	2 s.			...	
4	8 MB	5 Mbit/s	1.6 s.	183	0.1 MB	0.01 Mbit/s	10 s.
						...	
				319	0.1 MB	25 Mbit/s	0.004 s.
				320	0.1 MB	5 Mbit/s	0.02 s.
Total			2 seconds	Total			10 seconds

D. Proof of advantages against threats

Decentralized storage in terms of iterations of data to send to the storage. By modeling algorithms are presented graphically encryption process, division, verification and recording of data in blockchain. It is important to note that all transactions carried out with the data held on the client side, as a result of the user and provides full control over the files. Conducted an audit units and maintaining files and redundancy provide continuous access to data and the impossibility losses files.the thus, decentralized storage provides maximum data protection with point of view threethe main cloud storage problems: confidentiality, availability and integrity. Comparative characteristics of the traditional systems and decentralized storage of selected criteria on the safety revealed definite advantage over Storj Dropbox. Septic Storj system is more tolerant and provides continuous availability to ensure data integrity by data audits. Also in comparison with decentralized system Dropbox blockchain as Storj, completely waives Operation, encryption, since the encryption and division of data is on the client side.

VI. CONCLUSION AND FUTURE WORKS

The main problem of today's cloud storage is the existence of a mediator, a provider of cloud services. Trusted third party, usually has access to the data because it stores encryption keys.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

Data is stored on centralized servers, which are attractive points for malicious attacks. In this case, the safety and security of the data is questionable, but in this age of information is an important factor, both for individual users and for large organizations. This WRC investigates the decentralized model of the cloud data storage based blockchain, identifying advantages over traditional cloud storage for data security in terms of confidentiality, availability and integrity.In the course of writing the thesis studied the scientific literature on the topic blockchain and storage of both traditional and decentralized. disadvantages of modern cloud-based solutions. Also, an analysis of the security methods to ensure data in a decentralized environment, and inspection units and files for recoverability. It was also carried out a comparative analysis of conventional and decentralized storage solution. To address the decentralized network from the perspective of fault tolerance and modeling interaction scenarios of nodes through the user interface in the network in the form of flow charts in UML notation. Scheme of interaction sites demonstrate how data and your identity are protected. Writing metadata about the transaction in blockchain, it becomes possible to check the data with the help of transaction hashes and signatures. Consequently, it can be argued that decentralized systems are immune to unauthorized access to data.

After conducting research and developing models of decentralized storage logical to draw the following conclusions regarding data storage, providing the highest possible data security:

- Secure storage should work in an environment in which the relationship between the parties are not built on trust. Maximum data security is ensured only if no one except the user does not hold the key to access the data.
- To achieve reliability of the storage system should be decentralized and should share information in an encrypted form between independent storage nodes.
- Secure storage system must necessarily include monitoring of nodes for failover and verifying data integrity and availability.

During this final qualifying work the following tasks were carried out:

- Studied the scientific literature concerning the Blockchain technology and data storage;
- The analysis methods for ensuring security of data in a decentralized environment to blockchain;
- The models of traditional and decentralized storage;
- A comparative analysis of conventional and decentralized cloud storage based on blockchain;
- The models of interaction scenarios between nodes in the form of diagrams in sequence notation user UML.

Consequently, we were able to fully achieve the goal of the work, in identifying the benefits of a decentralized model of the data warehouse based on blockchain technology in comparison with traditional cloud storage.

REFERENCES

1. Internet Live Stats [electronic resource] - URL: http://www.internetlivestats.com/internet-users/
2. Vorontsov EA, EG Meleshenko Blockchain: panacea or a threat for the storage and transmission of information // Synergy Sciences. Number 2016. 5. - pp 93 - 101. - URL:http://synergy-journal.ru/archive/article0044
3. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2009. - URL: https://bitcoin.org/bitcoin.pdf
4. V. Buterin.Secret sharing and erasure coding: A guide for the aspiring dropbox decentralizer. 2014. - URL:https://blog.ethereum.org/2014/08/16/secret-sharing-erasurecoding-guide-aspiring-dropbox-decentralizer/
5. hitesh Malviya Reinventing Cloud with Blockchain // White Paper. 2016.
6. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder Bitcoin and Cryptocurrency Technologies. 2016. - 308 c.
7. Venkata Narasimha Inukollu 1, Sailaja Arsi1 and Srinivasa Rao Ravuri Security Issues Associated With Big Data In Cloud Computing // International Journal of Network Security & Its Applications (IJNSA) 2014. – Vol.6 – №3.
8. Tim Mather Cloud Computing and Privacy. 2009. – 335 c.
9. Michael Crosby, Nachiappan, Pradan Pattanayak, Sanjeev Verma, Vignesh Kalyanaraman AIR Applied Innovation Review Issue. 2016. – №2.
10. Fahmida Y. Rashid The dirty dozen: 12 cloud security threats [// InfoWorld. 2016. 18.05.2017)
11. Zach Herbert Why blockchains are the future of cloud storage [// Sia blog. 2017. (Дата обращения: 18.05.2017)
12. David Vorick Decentralization in Light of the Recent Amazon S3 Downtime [Электронный ресурс] // Sia blog. 2017. (18.05.2017)
13. Jonathan Howell Getting Started with Private, Decentralized Cloud Storage [Electronic resource] // Sia blog. 2017 (reference date: 18/05/2017)
14. Shawn Wilkinson, Jim Lowry Metadisk: Blockchain-Based Decentralized File Storage Application // Whitepaper.2014.
15. shawn Wilkinson Storj: A Peer-to-Peer Cloud Storage Network // Whitepaper 2014. - v1.01
16. shawn Wilkinson Storj: A Peer-to-Peer Cloud Storage Network // Whitepaper.2016 - v2.0
17. Nick Lambert and Benjamin Bollen "The SAFE Network a New, Decentralised Internet" // Paper for the final symposium of the ADAM project. 2014.
18. vitalik Buterin Merkling in Ethereum [Electronic resource] // Ethereum blog. 2015. - URL:https://blog.ethereum.org/2015/11/15/merkling-in-ethereum/ (Reference date: 18/05/2017)
19. Cawrey. How bitcoins technology could revolutionize intellectual property rights [electronic resource] // CoinDesk. 2014. - URL: http://www.coindesk.com/how-block-chaintechnology-is-working-to-transform-intellectual-property (Reference date: 18/05/2017)
20. M. Araoz.What is Proof of existence? [Electronic resource] - 2014. - URL: http://www.proofofexistence.com/about
21. roy Thomas Architectural Styles and the Design of Network-based Software Architectures // DISSERTATION in Information and Computer Science Fielding. 2000.
22. johanna Amann Redundancy and Access Permissions in Decentralized File Systems // Die Dissertation wurde bei der Technischen Universität München Eingereicht und durch die Fakultät für Informatik. 2011.
23. Alexandros G. Dimakis Network Coding for Distributed Storage Systems // IEEE Transactions on Information Theory. 2010. - VOL. 56 - № 9.
24. Filecoin: A Cryptocurrency Operated File Storage Network. [Electronic resource] - 2014 - URL: filecoin.io
25. Hall. Storj core [electronic resource] - 2016. - URL: http://storj.github.io/core/
26. V. ButerinA next-generation smart contract and decentralized application platform, (2014). URL https://github.com/ethereum/wiki/wiki/White-Paper
27. How Big Is The Cloud? [Electronic resource] / Storj: official website. - URL:http://blog.storj.io/post/95376799893/how-big-is-the-cloud. (Reference date: 16.05.2017).
28. Google's Data Centers on Punch Cards [electronic resource] / WhatIf. - URL:https://what-if.xkcd.com/63/ (Reference date: 16.05.2017).

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

29. colin Carson How Much Data Does Google Store? [Electronic resource] // Cirrus Insight. 2016. - URL:https://www.cirrusinsight.com/blog/much-data-google-store
30. (Reference date: 16.05.2017).
31. Burkhard Stiller Communication Systems VIII University of Zurich Department of Informatics. 2015. p.73-84.
32. Nick Lambert, Qi Ma and David Irvine Safecoin: The Decentralised Network Token. 2015.
33. Chunming Rong Beyond lightning: A survey on security challenges in cloud computing // Department of Electrical Engineering and Computer Science, University of Stavange. 2012.
34. Nir Kshetri Privacy and security issues in cloud computing: The role of institutions and institutional evolution // Bryan School of Business and Economics, The University of North Carolina. 2012.
35. Quick D and Choo K-K R. Google Drive: Forensic Analysis of Cloud Storage Data Remnants // Journal of Network and Computer Applications. 2013.
36. Jay J.Wylie, Michael W., Bigrigg, John D. Strunk Survivable Information Storage Systems // Computer. 2000. Volume 33, Issue 8, p. 61-68

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)