http://dx.doi.org/10.21172/ijiet.133.17

A Secure Scheme for storing data on the cloud using Attribute-Based Signatures and Blockchain concept

Akanksha Achanti[1], Sagarika Behera[2], P Raghavendra Reddy[3]
[1,2,3]*Department of Computer Science Engineering, CMRIT, Bangalore, Karnataka, India*

Abstract- The cloud is an interesting solution for the remote storage of large amounts of data. But it is known that data stored on the cloud is vulnerable to security threats. In such an environment, securing the data that is stored is of paramount importance. This paper presents a solution for storing data on the cloud using the combined concepts of role-based access, blockchain, and attribute-based signatures for securely storing and retrieving data on the cloud.
Keywords– Cloud Computing, Blockchain, Attribute-Based Signatures, Public Key Infrastructures (PKI)

NOMENCLATURE
Description of NOTATIONS USED IN THIS PAPER

SYMBOL	DESCRIPTION
F	File
F’	Encrypted file
K	RSA public key of the system
k	RSA private key of the system
k’	Encrypted private key
HC	An MD5 hash of file
PHC	Hash of the previous block
ID	Block ID
BH	Block Header
TS	Time Stamp
BL	Block
Si	DES Symmetric key of User i
C	Cloud Storage
||	Concatenation operation

I. INTRODUCTION

In the 21st century, it has been observed that Cloud computing is a great platform for storing data online digitally. Today, the popularity of storing data online on cloud servers is increasing. This is because the cloud allows unlimited storage at reasonable cost and the data can be easily accessed from anywhere in the world. But the major bane of storing data on the cloud is that the data is vulnerable to security threats. Data breaches, hijacking of accounts, Insider threats, Malware injection, Denial of Service attacks and Data Loss are some of the possibilities that one must anticipate while storing the data on the cloud. Therefore it is imperative to store the data securely and also to provide controlled access to the data. One way to secure the data would be to use the blockchain concept. A blockchain is a growing list of cryptographically linked records/ units of data, called blocks. Every block in this chain consists of the following: hash of the previous block, the timestamp, and the data. Although it is possible to alter the records in the block chain, they can still be considered secure by design and are a great example of a system with high fault tolerance.

Additionally, the data could be made more secure in the following way. Access to the data could be given to a specific set or a subset of users. In order to further secure the data, we use Attribute-Based Signatures (ABS), a scheme that enables an entity to sign messages that has precise control over identifying information. In this scheme, a signer has a set of attributes from the authority. This entity has the ability to sign a message with a predicate. The attributes of the user satisfy the predicate mentioned above. The signature says nothing more than the fact that a single user possessing a set of attributes that satisfy the predicate has attested to the message. The signature hides the attributes used to satisfy the predicate and any identifying information about the signer that could enable an adversary to identify a user by the process of linking multiple signatures as being from the same signer.

The content of the paper is organized as follows. Section II discusses the related works. Section III describes the key technologies used in the system in detail. Section IV describes the proposed system and the associated algorithms. Section V demonstrates the implementation of the proposed scheme for a particular use case. Section VI discusses conclusions and suggests future work respectively.

II. LITERATURE REVIEW

Ilya and Sukhodolskiy et al.[2] in their paper present a prototype of a multi-user system for access control to datasets stored in an un-trusted cloud environment. Their approach provides access control over the data stored in the cloud without the provider participation. Their system uses a blockchain based decentralized ledger to provide an immutable log of all meaningful security events, such as key generation, access policy assignment, change or revocation, access request.

They propose a set of cryptographic protocols ensuring the privacy of cryptographic operations requiring secret or private keys. Maji and Prabhakaran et al. [3] introduced the Attribute-Based Signatures (ABS) scheme. In this scheme, a signature attests to a claim regarding the attributes possessed by the user, instead of the identity of the individual who endorsed the message. A strong policy of privacy is guaranteed for the signer. This is because the signature reveals nothing about the identity or attributes of the signer that is beyond what is explicitly revealed by the claim being made. They then illustrate the application of the aforementioned scheme in ABM (Attribute based messaging systems). Unlike many other attribute-based cryptographic primitives, ABS can be readily used in a multi-authority setting, wherein users can make claims involving combinations of attributes issued by independent and mutually distrusting authorities. Dalia et al [4] in their paper introduce a new Attribute Based Group Signature (ABGS) scheme that will enable us to remove a member from a group or remove some of his attributes if and when it is needed. It has been demonstrated that the client can choose his/her password freely. Herranz et al. [5], in their paper, present two attribute-based signature schemes with constant size signatures. Joshi et al [1] in their paper have developed a centralized attribute-based authorization mechanism that makes use of Attribute-Based Encryption (ABE), allowing secure access of patient records by entrusted entities. The mechanism proposed in the paper migrates the service management overhead from the patient to the hospital. This makes it easy to delegate the access authority of the patient's electronic health records to the concerned authorities, that is the medical providers. In their paper, they have described this novel ABE approach as well as the prototype system they have used to illustrate it.

III. KEY TECHNOLOGIES

This section discusses the key technologies used in this scheme. This scheme uses both symmetric as well as asymmetric key cryptography techniques, along with the concepts from blockchain and attribute-based encryption schemes. Created blocks are transferred to the cloud using FTP protocol. Hashing and RSA encryption are done for the creation of individual blocks. Hashing is done using the MD5 algorithm. The user attributes are concatenated to create the symmetric key that is used for DES encryption of the RSA private key. The user has to identify and authenticate himself by uploading this encrypted key. The user will only get access to the file only after successful verification.

3.1 Attribute-based Signatures (ABS):

Attribute-based encryption is one of the encryption techniques in which the secret key of a user as well as the ciphertext are dependent on attributes (e.g. the country in which he lives, or the kind of subscription he has, or his department and his designation). In such a system, the decryption of a ciphertext is possible only if the set of attributes of the user key matches the attributes of the ciphertext. In our scheme, the symmetric key created using the ABS scheme is used in the process of encryption of the RSA public key of the system.

3.2 Block Chain Technology:

A blockchain is an expanding list of records that are called blocks, which are interlinked with each other by the use of cryptographic techniques. Every block that is a part of this chain contains a cryptographic hash of the previous block along with a timestamp, and the transaction data, that is usually represented as a Merkle tree root hash. A Blockchain is defined as a distributed decentralized and digital registry or log that is used for recording transactions across many computers. This is done so that the record cannot be altered ex post facto without the alteration of all the following blocks and the network consensus.

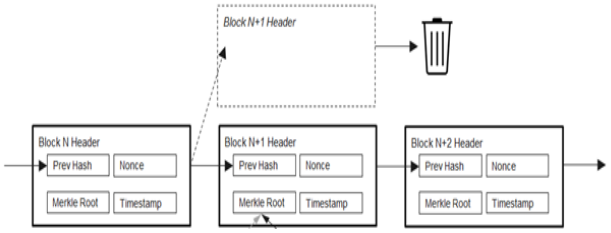

Figure 1 BlockChain Structure

This allows the entities involved to verify and audit transactions in a non-expensive manner. A database that stores the Blockchain is managed using a distributed time stamping server. The authentication is done by a mass collaboration that is powered by the collective self-interests. The result is a robust workflow where there is marginal uncertainty regarding the security of the data. The use of this Blockchain technology removes the digital asset's capability to be reproduced an infinite number of times.

IV. THE PROPOSED SYSTEM

This section elaborately describes the proposed scheme to securely share and store files. The proposed system stores the individual files uploaded by the data owner entity in the form of a chain of blocks that are stored on the cloud. Each block contains the block header that links it to the previous block and an encrypted version of the file. This encryption is done using the RSA algorithm. All files are encrypted using a single key that is stored in the database.

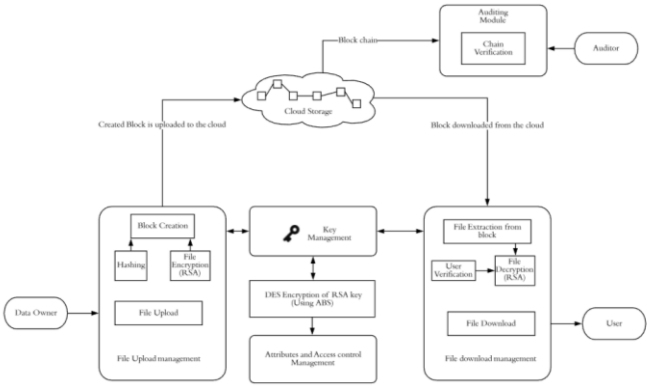

Figure 2 Proposed System Architecture

The basic architecture of the system is demonstrated in Fig 2. The system has the following set of entities:

1. Administrator: The administrator performs the functions of creating a new public key-private key pair for the encryption of files in the system; or adding or removing the existing data owners.
2. Data Owner: The data owner creates and uploads files to the system and these files later form individual blocks stored in the cloud. The data owner has the ability to add new users to the system by specifying the attributes of users. The data owner can specify the access control of each file and can restrict it to a particular subset of users.
3. User: The user can download the files that he has access permission to (set by the data owner). To authenticate and identify himself, the user is required to upload the encrypted private key that was sent to him. The private key is decrypted and is used to decrypt the desired file, which is then downloaded.
4. Auditor: The Auditor is an entity who verifies blocks of the blockchain to ensure that the data has not been tampered with. By the design of the blockchain, if any single block is modified, the connection to the next and subsequent blocks will be disrupted.

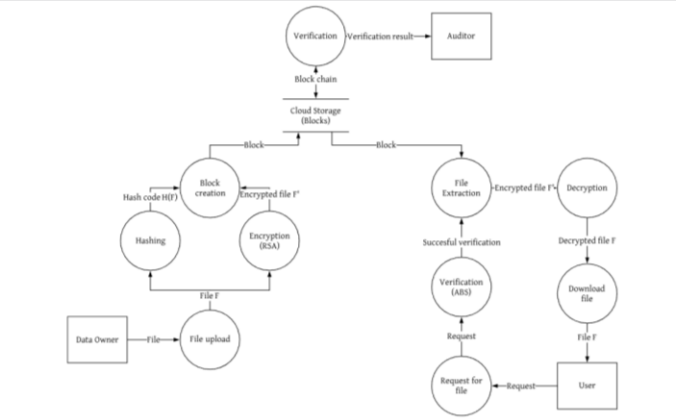

Figure 3 Dataflow diagram for the proposed system

When the data owner uploads a file, the file is first encrypted using the RSA algorithm. This encrypted file forms the block body of a block. The block header of this file contains the hash of the previous block, the timestamp, and the hash of the current file. The hashing is done using the MD5 algorithm. This block is converted to a zip file, and this file is transferred from the server to the cloud storage using FTP protocol.

A data owner can add, delete and edit details of users, who will then be given access to selected files. When a data owner creates a user, the RSA private key that is used to encrypt all the files of the system is sent to the user. This private key is sent to the user in an encrypted form. The private key is encrypted using the Attribute-Based Signature (ABS) scheme. That is, the attributes of the user (such as user id, designation, and department) are used to form the symmetric key of this particular user. This symmetric key is then used to encrypt the private key file that is sent to the user at the time of user creation.

The data owner can specify and restrict the access of the files to users of a particular department and designation, or in other words, the set of users that have the specified attributes. These access permissions can also be modified and deleted by the data owner. When a particular user that satisfies these constraints logs into the system, he can download the particular file. In order to authenticate himself, the user has to upload the encrypted private key that was sent to him. The key uploaded is then decrypted using this user's symmetric key (that was created using the attributes of this user). On decryption, the key obtained is used to decrypt the file which was stored in the encrypted form inside its block on the cloud.

4.1 Algorithm for File Upload process:

The sequence of processes associated with uploading of the file is given below. The file upload process is initiated by the data owner entity.

Input: Uploading the file

Output: Block generation and uploading of the block to the cloud

1. Upload file F to the server (by data owner)
2. Retrieve the RSA Key K from the database DB
3. Encrypt the file F
   RSAencrypt (F, K) -->F'
4. Generate the hash code HC for F
5. Fetch previous block hash code PHC
6. Generate the timestamp TS and Random number RN
7. Generate Block Header
   (BH) --> HC || PHC || TS || RN
8. Generate Block
   BL -->BH+F'
9. Connect to cloud C by establishing an FTP connection
10. Write the Block BL in C
11. Stop

On upload of the file to the system, the data owner can specify the set of attributes that a user has to possess in order to be able to download the file.

4.2 Algorithm for File Download process:

The sequence of processes associated with the download of the file is given below. The file download process is initiated by the user entity.

Input: Request for file

Output: File downloaded / display of the appropriate message.

1. User requests download of file F to the server
2. User uploads the encrypted private key k'.
3. Decrypt the key k'

DESdecrypt (k', Si) -->k

4. If k is not the RSA key of the system: Exit.
5. If user attributes do not match access control settings for file: Exit.
6. Fetch block from the cloud C by establishing FTP connection
7. Retrieve encrypted file F' from the block body
8. Decrypt the file F'

RSAdecrypt(F',k)-->F

9. Download file F to the user's system.
10. Stop

4.3 Algorithm for the Blockchain Verification process:

The sequence of processes associated with the download of the file is given below. The blockchain verification process is carried out by the auditor entity.

Input: Block ID (of starting block)

Output: Display whether or not data is corrupted.

1. Fetch the block content using first block ID
2. Unzip block content
3. Extract root hash HC
4. Get details of next block
5. While (next block exists):
    a. Fetch next block
    b. From this extract previous block hash code PHC
    c. If PHC!= root hash:
        Continue
    Else:
        Display 'chain is corrupted'
        Exit
6. Display 'chain is not corrupted'

V. RESULTS AND ANALYSIS

In this section, the implementation of the proposed scheme is demonstrated. The above scheme has been implemented in the scenario of IT system, where a user (data owner) can upload a file to the system and users of the specified department and designation (attributes) can download the file upon verification. The auditor performs the verification of the blockchain.

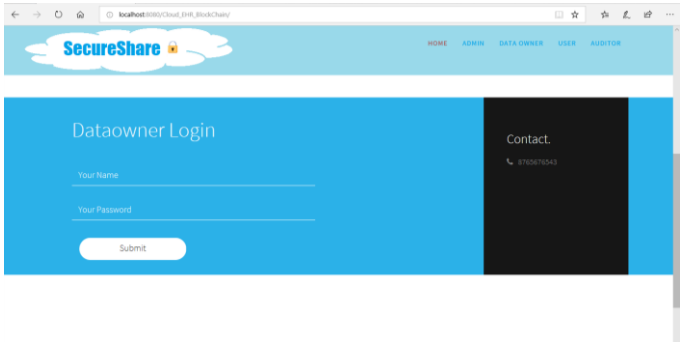

Figure 4 Data Owner Login page

http://dx.doi.org/10.21172/ijiet.133.17

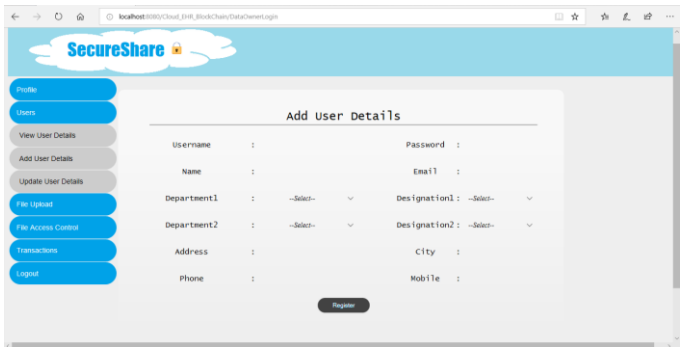

Figure 5 Adding a new user

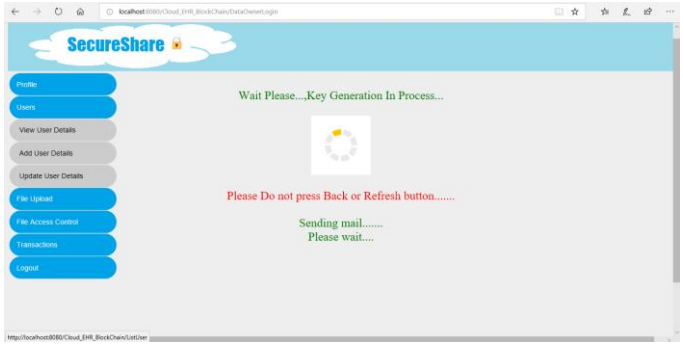

Figure 6 User creation- generation of symmetric key

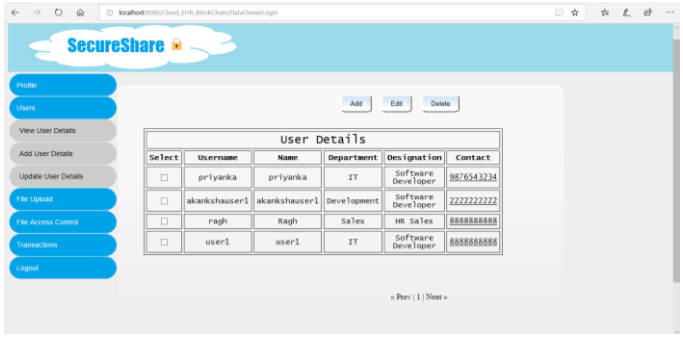

Figure 7 View and edit user details

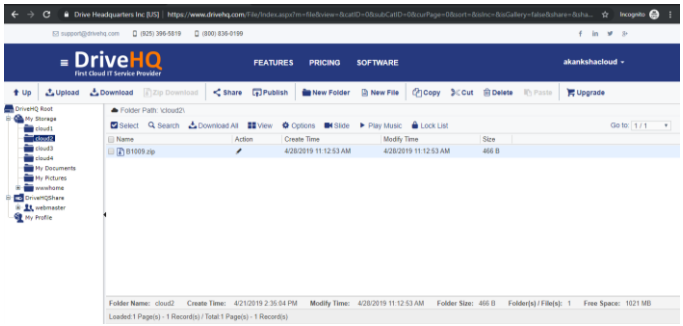

Figure 8 The file is converted to block and stored on the cloud

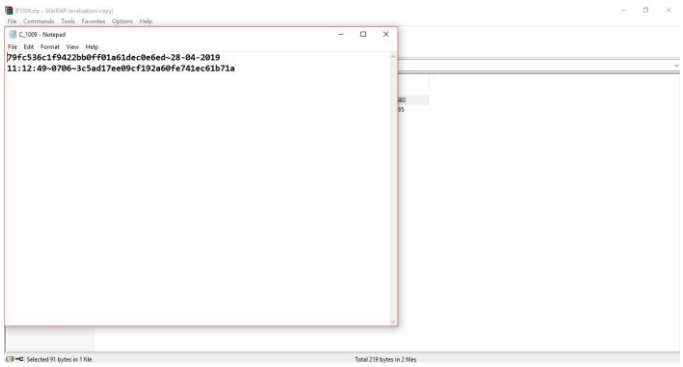

Figure 9 Block header containing the hash of the previous block, timestamp, and hash of current file

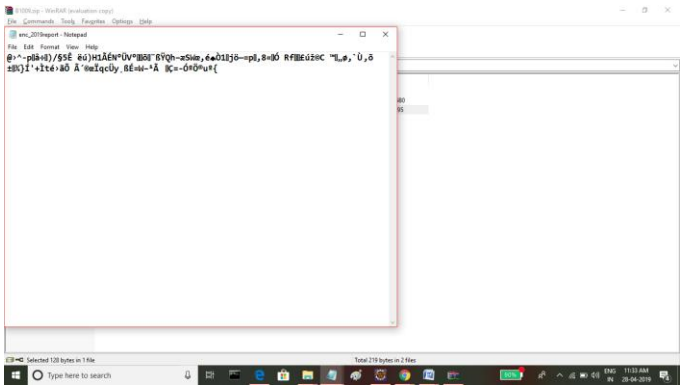

Figure 10 Block body containing the encrypted file

VI. CONCLUSION AND FUTURE SCOPE

In this paper, we present a secure way to store files on the cloud in the form of blocks in a blockchain. Each block contains the encrypted individual files. Further security has been provided by restricting the access to files to a set of users with the attributes specified by the creator of the file, or the data owner. Further, the private key used to decrypt the files is sent to the users in an encrypted format.

The future work would be to further enhance the security of the system by replacing the algorithms used with their sophisticated and secure counterparts and also to implement and test the security level of the implemented system.

VII. REFERENCE

[1] Maithilee Joshi, Karuna P. Joshi, and Tim Finin, "Attribute-Based Encryption for Secure Access to Cloud-Based EHR Systems", 2018 IEEE International Conference on Cloud Computing.

[2] Ilya Sukhodolskiy, Sergey Zapechnikov, "A blockchain-based access control system for cloud storage", 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), Moscow, 2018, pp. 1575-1578.

[3] Hemanta K. Maji, Manoj Prabhakaran, Mike Rosulek, "Attribute-Based Signatures", ECCC ISSN 1433-8092

[4] Dalia Khader, "Attribute-Based Group Signature with Revocation"

[5] Javier Herranz, Fabien Laguillaumie, Benoˆıt Libert, and Carla Ra`fols, "Short Attribute-Based Signatures for Threshold Predicates"

[6] Behrouz A Forouzan and Debdeep Mukhopadhyay, "Cryptography and Network Security", 3rd edition, TMH publisher