Proposal of an Efficient Architecture for Data Protection in Cloud

[1]Jalashree D. Trivedi, [2] Prof. Amita V. Shah

[1]P.G Student, [2] Professor

[1]Department of Computer Engineering,
[1] L.D College of Engineering, Ahmedabad, India

Abstract : Cloud data storage is a service that brings several advantages for its users. However, in public cloud systems, the risks involved in the outsourcing of data storage can be a barrier to the adoption of this service by those concerned with privacy. Several cloud service providers that claim to protect user's data do not fulfil some requirements considered essential in a secure, reliable and easy to use service, raising questions about the effective security obtained. In this research we have proposed constructing a model of decentralized data storage based on client side encryption using ECC algorithm and blockchain technology with the purpose to improve the security and safety of confidential information.

Index Terms—Cloud data storage, Security, Decentralization, Encryption, Blockchain, ECC

I. INTRODUCTION

Security is one of the most difficult task to implement in cloud computing. The paper basically deals with the security issues that are experienced during the storage of data on the cloud. The cloud vendors generally store the client's data and information in cloud without following any security measures. Almost every cloud provider does not provide enough security measures to ensure the data safety and that's why clients waver keeping their data at some place which is very easy to be accessed by someone else.

Normally Cloud storage is centralized where are all data are stored on a single server or a computer so that if the server is hacked or broken, the confidential data can be accessed thereby interrupting the service. Also trusting the third party service is not recommended for confidential data transfer. Hence 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.

The overall problem can be summed up as follows: designing a network that lets a client control their data, distributes the work, and maintain security and privacy of data over cloud. These are fundamental problems which must be solved in order to fulfil the purpose of the study, and involve a number of challenges in both security and network communication. However, the challenges can be divided into different sub problems that are easier to solve. We address the following three issues:

- How can work be distributed in a decentralized network?
- How can the network verify that all nodes in network execute work correctly?
- How can security of data and metadata be maintained in the distributed network?

The main objectives of this research are : To identify problems and concerns related to the security and privacy of users when storing their data on public cloud, to investigate how to distribute work on a network of nodes with no central authority, and how to verify that the work has been executed correctly. The key focus is to conduct research in the field of decentralized applications and to identify encryption algorithm that can provide a good level of client side security of data stored over cloud.

II. Literature Review

[7] proposed an identity-based hybrid encryption method (RSA with ECC) for the outsourced computation on encrypted information in cloud computing. The identity based encryption is combined with hybrid RSA with ECC to encrypt the user data. In standardize to improve the security the proxy re encryption is utilized to encrypt the user identity and keyword. [5] proposes a scheme to build a trusted cloud storage system, which allow the user to store and access their data securely in the cloud by encrypting the data in the client side and decrypting the data after down loading from the cloud. An efficient secure storage scheme is presented in [2], which aims to provide security to end-user's data while mostly storing it to public clouds. This proposed scheme is based on the invertible Discrete Wavelet Transform (DWT) to fragment data into two or three fragments with different levels of importance and protected accordingly. As a matter of fact, the most important fragment takes the smallest amount of storage space and can be stored in a user trusted area while the less important fragments take most of the storage space and are uploaded to public clouds.

III. BLOCKCHAIN FOR DATA INTEGRITY

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.[25]

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.

Figure 1: Blockchain structure[25]

IV. ECC ALGORITHM FOR DATA SECURITY

ECC is pronounced as elliptic curve cryptography[13], developed by Neil Koblitz and Vector Miller in 1985. ECC provides better security with a smaller key size if we compare it with other asymmetric algorithms [9]. ECC 160-bit gives same level of security to data as RSA 1024-bit does. High level of security can be achieved using a small key size. ECC works on elliptic curve equation. Elliptic curve equation for binary field is written as-

$$y^2+xy=x^3+ax+b$$

where a and b are two constants, different elliptic curves will be shaped with different values of these two constants. Elliptic curve equation for prime field is given as-

$$y^2=x^3+ax+b \bmod p$$

here a and b are constants and p is a prime number. Grater the value of number p more will be the number of points generated on the elliptic curve. Large number of points on the curve gives high level of security. Elliptic curve is shown in figure 2 below-

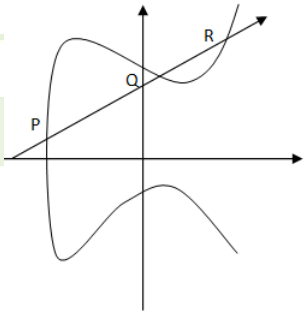

Figure 2 : Elliptic curve[13]

Elliptic Curve Encryption

Elliptic curve cryptography can be used to encrypt plaintext messages, *M*, into ciphertexts. The plaintext message *M* is encoded into a point *PM* form the finite set of points in the elliptic group, $Ep(a, b)$. The first step consists in choosing a generator point, $G \in Ep(a, b)$, such that the smallest value of *n* such that $nG = O$ is a very large prime number. The elliptic group $Ep(a, b)$ and the generator point *G* are made public. Each user select a private key, $nA < n$ and compute the public key *PA* as: $PA = nAG$. To encrypt the message point *PM* for Bob (*B*), Alice (*A*) choses a random integer *k* and compute the ciphertext *pair of points PC* using Bob's public key *PB*:

$$PC = [(kG), (PM + kPB)] \quad (1)$$

After receiving the ciphertext pair of points, *PC*, Bob multiplies the first point, (*kG*) with his private key, *nB*, and then adds the result to the second point in the ciphertext pair of points, (*PM* + *kPB*):

$$(PM + kPB) - [nB(kG)] = (PM + knBG) - [nB(kG)] = PM \quad (2)$$

which is the plaintext point, corresponding to the plaintext message *M*. Only Bob, knowing the private key *nB*, can remove *nB*(*kG*) from the second point of the ciphertext pair of point, i.e. (*PM* + *kPB*), and hence retrieve the plaintext information *PM*.

Security of ECC

The cryptographic strength of elliptic curve encryption lies in the difficulty for a cryptanalyst to determine the secret random number *k* from *kP* and *P* itself. The fastest method to solve this problem (known as the *elliptic curve logarithm problem*) is the Pollard ρ factorization method .The computational complexity for breaking the elliptic curve cryptosystem, using the Pollard ρ method, is 3.8×1010 MIPS-years (i.e. millions of instructions per second times the required number of years) or an elliptic curve key size of only 150 bits .

V. PROPOSED SYSTEM ARCHITECTURE

The overall architecture of the proposed system is depicted in Figure 3 below

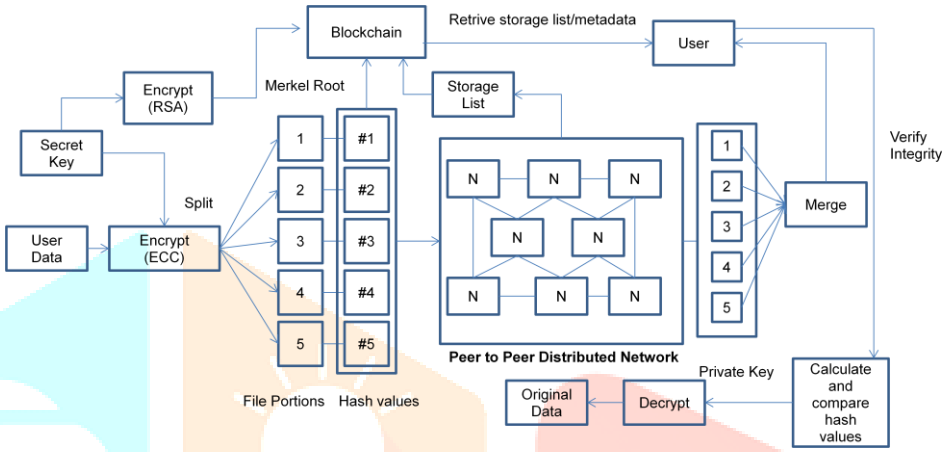

Figure 3 : Proposed System Architecture

The overall proposed system architecture is shown as above. The architecture is implemented in following sequence:

1. **Encrypting files and keys.** Encrypt the data file using a strong encryption algorithm such as ECC. For more security the secret keys used during encryption can also be encrypted with another algorithm such as RSA. The keys can then be stored along with the metadata.
2. **Hashing and division into parts.** 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.Encrypted files are divided into pieces (shards), or multiple files are combined to create a single shard.
3. **File Distribution.** Files randomly distributed across the network, along with 3 copies of each shard (3 copies of files stored - is the industry standard).
4. **Merging the parts and verifying Integrity.** The user who wants access to data has to download the file parts on his machine. He also needs to download metadata information stored over the blockchain to get access to storage location list of file parts as well as hash value of the file. He can then combine the split parts into single file and verify its integrity by comparing the hash values.
5. **Decryption.** In the end he will access the secret keys retrieved from the metadata and decrypt the data to get back the original data file.

VI. IMPLEMENTATION

The entire implementation was carried out in matlab r2017a release, on a Lenovo core i5 processor. Below are the implementation details.

Blockchain Block Structure

The first logical step is to decide the block structure. The main components are : index, timestamp, data, hash and previous hash.

Block hash

The block needs to be hashed to keep the integrity of the data. A SHA-256 is taken over the content of the block.

Generating a block

To generate a block we must know the hash of the previous block and create the rest of the required content (= index, hash, data and timestamp). Block data is something that is provided by the end-user.

Storing the blocks

A in-memory array is used to store the blockchain. The first block of the blockchain is always a so-called “genesis-block”, which is hard coded.

Validating the integrity of blocks

At any given time we must be able to validate if a block or a chain of blocks are valid in terms of integrity. This is true especially when we receive new blocks from other nodes and must decide whether to accept them or not.

Communicating with other nodes

An essential part of a node is to share and sync the blockchain with other nodes. The following rules are used to keep the network in sync.

- When a node generates a new block, it broadcasts it to the network
- When a node connects to a new peer it querys for the latest block
- When a node encounters a block that has an index larger than the current known block, it either adds the block the its current chain or querys for the full blockchain.

The user is able to interact with the node in the following ways:

- List all blocks
- Create a new block with a content given by the user
- List or add peers

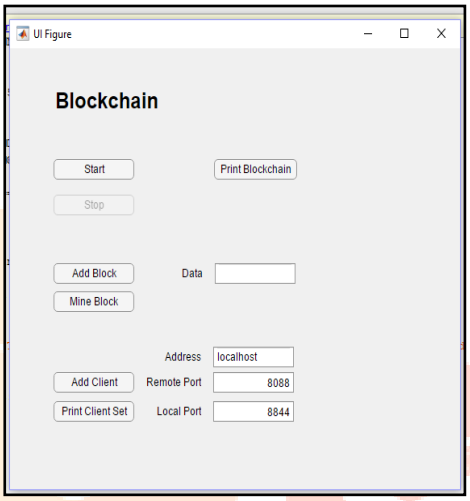

Figure 4 : Blockchain Implementation

ECC Encryption Steps

Step 1 : Obtain the plain text to send.
Step 2 : Convert it to corresponding ASCII values.
Step 3 : Use the group size to partition the ASCII values into equally sized groups.
Step 4 : Convert the obtained groups from previous step to big integer numbers with base 256.
Step 5 : Pad with 32 (blank space) at the end of the list, if the count of the list is odd.
Step 6: Select random k value, k = Random value with range 1 to $n-1$. Compute kG and kPb using Point multiplication operation.
Step 7 : Compute $Pm + kPb$ using point addition or point doubling as required.
Step 8 : Send $Pc = \{kG, Pm + kPb\}$ as cipher text to the receiver side.

ECC Decryption Steps

Step 1 : Get the cipher text Pc.
Step 2 : Get the left part kG and right part $Pm + kPb$ of the Pc separately.
Step 3 : Multiply with nB to the left part and subtract it from the right part to get Pm.

$$\{Pm + kPb\} - nBkG = Pm$$

Step 4 : The above operation will yield the big integer value which is formed by combining group of ASCII values. Convert it back to list of ASCII values.
Step 5 : Convert the list of ASCII values to its corresponding characters.

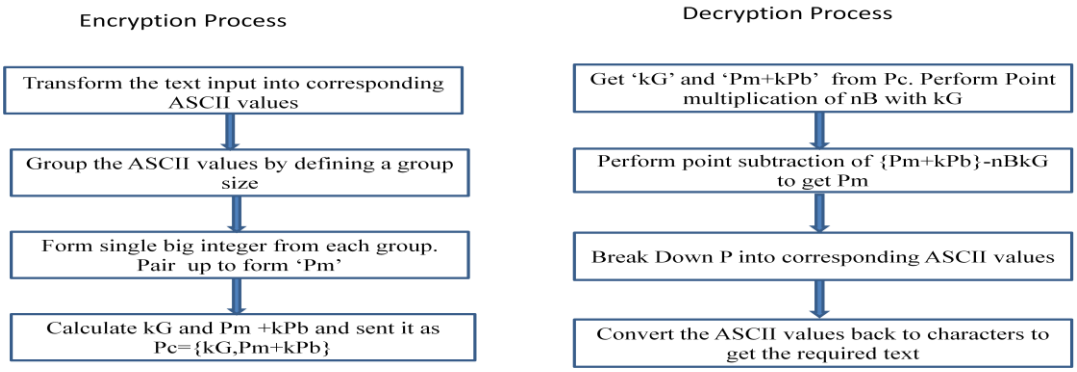

Figure 5 : Encryption Decryption process with proposed method

The figure below is a snapshot of the implemented process in matlab release R2017a,

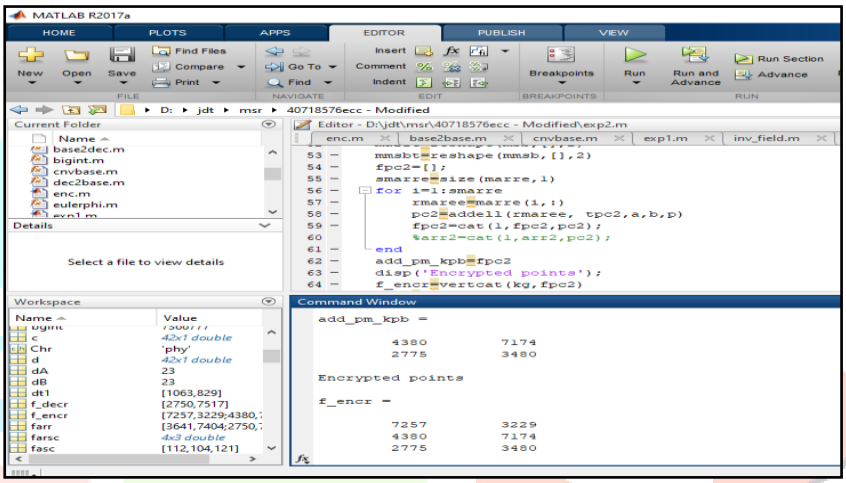

Figure 6 Implementation snapshot

VII. EXPERIMENTAL PARAMETERS

The following graphs indicate a measurable improvement in the performance of proposed system algorithm over base algorithm, where point mappings were carried out using mapping table.

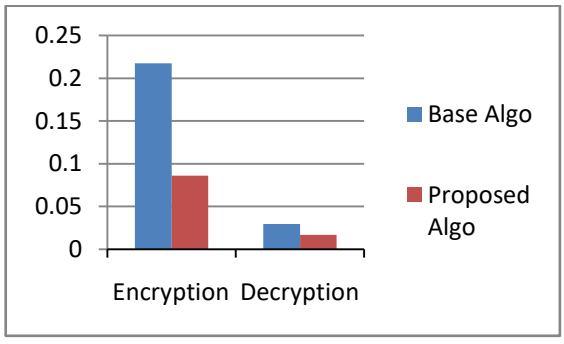

Figure 7 Encryption Decryption time comparisons (secs)

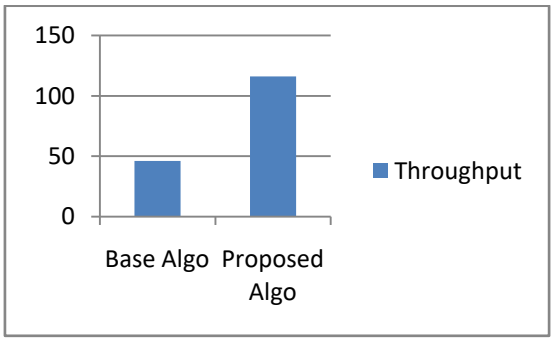

Figure 8 Throughput time comparisons (bytes/sec)

VIII. CONCLUSION

In this paper a new decentralized cloud data storage scheme is proposed. We can see from the above results that when decentralized approach for storing cloud data is implemented, it can provide data security by enforcing client side encryption with ECC algorithm. It also enhances integrity of the data by implementing the block chain technology for storing and distributing small amounts of data

IX. REFERENCES

[1] Lin JS. (2011) Cloud Data Storage with Group Collaboration Supports. In: Fong S. (eds) Networked Digital Technologies. NDT 2011. Communications in Computer and Information Science, vol 136. Springer, Berlin, Heidelberg

[2] H. Qiu, G. Memmi and H. Noura, "An Efficient Secure Storage Scheme Based on Information Fragmentation," *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, New York, NY, 2017, pp. 108-113.

[3] M. G. Jaatun, Å. A. Nyre, S. Alapnes and G. Zhao, "A farewell to trust: An approach to confidentiality control in the Cloud," *2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE)*, Chennai, 2011, pp. 1-5

[4] P. C. Chen, C. P. Freg, T. W. Hou and W. G. Teng, "Implementing RAID-3 on cloud storage for EMR system," *2010 International Computer Symposium (ICS2010)*, Tainan, 2010, pp. 850-853.

[5] X. C. Yin, Z. G. Liu and H. J. Lee, "An efficient and secured data storage scheme in cloud computing using ECC-based PKI," *16th International Conference on Advanced Communication Technology*, Pyeongchang, 2014, pp. 523-527.

[6] S. R. Pardeshi, V. J. Pawar and K. D. Kharat, "Enhancing information security in cloud computing environment using cryptographic techniques," *2016 International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, 2016, pp. 1-5.

[7] G. P. Kanna and V. Vasudevan, "Enhancing the security of user data using the keyword encryption and hybrid cryptographic algorithm in cloud," *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, Chennai, 2016, pp. 3688-3693.

[8] M. S. Kumar and M. Kumar, "A secured cloud storage technique to improve security in cloud infrastructure," *2013 International Conference on Recent Trends in Information Technology (ICRTIT)*, Chennai, 2013, pp. 97-102.

[9] M. Schnjakin, D. Korsch, M. Schoenberg and C. Meinel, "Implementation of a secure and reliable storage above the untrusted clouds," *2013 8th International Conference on Computer Science & Education*, Colombo, 2013, pp. 347-353.

[10] A. Alsirhani, P. Bodorik and S. Sampalli, "Improving Database Security in Cloud Computing by Fragmentation of Data," *2017 International Conference on Computer and Applications (ICCA)*, Doha, 2017, pp. 43-49.

[11] Babitha M. P. and K. R. R. Babu, "Secure cloud storage using AES encryption," *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, Pune, 2016, pp. 859-864

[12] Lin JS. (2011) Cloud Data Storage with Group Collaboration Supports. In: Fong S. (eds) Networked Digital Technologies. NDT 2011. Communications in Computer and Information Science, vol 136. Springer, Berlin, Heidelberg

[13] A Arjuna Rao1, K Sujatha1, A Bhavana Deepthi1, L V Rajesh1 (2017), Survey paper comparing ECC with RSA, AES and Blowfish Algorithms, International Journal on Recent and Innovation Trends in Computing and Communication , IJRITCC

[14] Omar Reyad , "Text Message Encoding Based on Elliptic Curve Cryptography and a Mapping Methodology"

[15] Md Obaidur Rahaman, Data and Information Security in Modern World

[16] https://www.uk.capgemini-consulting.com/blog/retail-banking/2016/04/blockchain-the-answer-to-secure-data in-a-cloud-based-world

[17] http://www.ybrikman.com/writing/2014/04/24/bitcoin-by-analogy

[18] http://www.zdnet.com/article/how-blockchain-is-likely-to-change-it-and-business-forever

[19] http://blog.allion.com/2014/06/testing-on-the-cloud-a-scenario-based-point-of-view-on-the-comparative-analysis-of-6-cloud-storage-services/

[20] http://www.ameerrosic.com/what-is-blockchain-finally-a-simple-guide/

[21] http://epaper.timesgroup.com/Olive/ODN/TimesOfIndia/#

[22] https://en.wikipedia.org/wiki/Blockchain

[23] https://sachi73blog.wordpress.com/2013/11/21/symmetric-encryption-vs-asymmetric-encryption/

[24] http://linuxdevices.linuxgizmos.com/primer-on-elliptical-curve-cryptography/

[25] https://medium.com/@lhartikk/a-blockchain-in-200-lines-of-code-963cc1cc0e54