Article

Not peer-reviewed version

A Novel Method of Secured Data Distribution Using Sharding Zkp and Zero Trust Architecture in Blockchain Multi Cloud Environment

Komala R , Arun Kumar B.R. * , Mahadeshwara Prasad , Shreyas A

Posted Date: 31 July 2024

doi: 10.20944/preprints202407.2555.v1

Keywords: cloud computing; big data; sharding concept; blockchain; smart contract; digital wallets; zero knowledge proof; zk-snarks; zero trust model; data privacy

Article

A Novel Method of Secured Data Distribution Using Sharding Zkp and Zero Trust Architecture in Blockchain Multi Cloud Environment

Komala R [1,‡], Arun Kumar B.R. [2,*,†,‡], Mahadeshwara Prasad [3,‡] and Shreyas A [4,‡]

[1] Ph.D Research Scholar, Dept. of MCA, VTU RC, BMS Institute of Technology & Management, and Assistant Professor, Department of Computer Applications, M S Ramaiah Institute of Technology, Bengaluru, India; Email: komal.uday@gmail.com.

[2] Professor, Department of Computer Science & Engineering & Research Supervisor, Department of MCA, BMS Institute of Technology & Management, Yelahanka, Bengaluru, India, ; Email : arunkumarbr@bmsit.ac.in.

[3] UG Scholar, Department of Computer Science & Engineering, BMS Institute of Technology & Management, Yelahanka, Bengaluru, India, ; Email: mahadeshwara.prasad07@gmail.com.

[4] UG Scholar, Department of Computer Science & Engineering, Sai Vidya Institute of Technology, Rajanukunte, via Yelahanka, Bengaluru, 560064, India; Email : shreyasvasista05@gmail.com

[*] Correspondence : arunkumarbr@bmsit.in; Tel.: 91-9886008210 : Arun Kumar B.R.

[*] Correspondence : arunkumarbr@bmsit.in; Tel.: 91-9886008210 : Arun Kumar B.R.

[†] Current address: Department of Computer Science & Engineering, BMS Institute of Technology and Management, Yelahanka, Bengaluru, India.

[‡] These authors contributed equally to this work.

Abstract: In the era of cloud computing, guaranteeing the safety and effectiveness of data management is of utmost importance. This investigation presents a novel approach that amalgamates sharding concept, encryption, zero-knowledge proofs (zkp), and blockchain technology for secure data retrieval and data access control to improve data security, efficiency in cloud storage and migration. Further we utilize user-specific digital wallets for secure encryption keys in order to encrypt the file before storing into the cloud. As Large files (greater than 50 MB) or Big data file (grater than 1 TB) computational complexity is more we leverage on sharding concept to enhance both space and time complexity in cloud storage, hence the large files are divided into shards and stored in different database servers. We also employed blockchain smart contract to enhance secure retrieval of the file and also a secure access method which ensures the privacy of the user. The zk-snark protocol is utilized to ensure the safe transfer of data between different cloud services. By utilizing this approach, data privacy is preserved, as only the proof of the data's authenticity is shared with the verifier at the destination cloud, rather than the actual data itself. The suggested method tackles important concerns related to data protection, privacy, and efficient resource utilization in cloud computing settings by ensuring it meets all the cloud policies require to store data. As the environment maintains privacy of the user data and by not storing the raw data of the user anywhere, the entire environment is setup as a Zero trust model.

Keywords: cloud computing; big data; sharding concept; blockchain; smart contract; digital wallets; zero knowledge proof; zk-snarks; zero trust model; data privacy

1. Introduction

In the era of digital transformation, the protection and confidentiality of user data are of utmost significance. As the amount of data being stored and managed on cloud platforms continues to grow rapidly, it is of utmost importance to guarantee the confidentiality and security of sensitive information, preventing unauthorized access. Users expect strong security measures to protect their data from breaches and leaks, ensuring the confidentiality, integrity, and availability of their information. Consequently, establishing reliable methods for safeguarding data and transferring it between cloud environments is a crucial area of study and advancement.

Prior to storing files in the cloud, it is crucial to encrypt them as a fundamental measure to safeguard data security. Also by employing the use of digital wallets which can be used for user authentication and also using the wallet's keys for encryption. By employing encryption, sensitive data is converted into an unreadable format that can only be accessed by authorized individuals. Nevertheless, when it comes to handling extensive files, encryption alone may not be enough to tackle concerns regarding efficient storage and easy access. This is where the concept of dividing data into smaller parts comes into play. Sharding entails breaking down a substantial file into smaller, more manageable fragments known as shards. Each piece of the puzzle is encrypted independently, which not only strengthens security but also enhances the speed and ease of data storage and retrieval. By spreading these encrypted fragments across various storage nodes, the system can attain improved performance, scalability, and fault tolerance. To ensure the security and privacy of user data, it is recommended to utilize blockchain smart contracts for secure data retrieval, access control, transparent system for managing data access, ensuring that file metadata cannot be tampered with. In order to add more security layer, the metadata is encrypted before sending to smart contract so that the information of file is kept hidden in the blockchain. This guarantees that only authorized individuals can access the file, safeguarding the confidentiality and integrity of the data. Moreover, smart contracts streamline access control procedures, minimizing the chances of human mistakes and unauthorized entry. This approach aligns with cloud security policies, providing a comprehensive solution that satisfies stringent privacy requirements. The incorporation of smart contracts not only strengthens the zero-trust model but also capitalizes on the inherent security features of blockchain technology to protect user data at every stage of its existence.

Zero-knowledge proofs, particularly zero-knowledge succinct non-interactive arguments of knowledge (zk-snarks), provide an effective approach for securely transferring data between cloud services. Zkps enable one party to demonstrate to another that they possess specific knowledge without disclosing the knowledge itself. Zkps are a form of zero knowledge proof. In the realm of cloud data migration, this implies that a user can produce a validation that their data has been correctly and securely decrypted without revealing the actual data to the verifier. Both the verifier and the user can interact with the smart contract to ensure data integrity of the file. The verifier can only have access to the metadata of the file to check the integrity but he doesn't have access for the raw data. By adopting this method, the data is safeguarded and kept confidential throughout the migration process. The verification process at the destination cloud can validate the accuracy of the data migration solely based on the proof, ensuring the privacy and security of the user's information. By employing zk-snarks, this approach not only improves security but also simplifies the verification process, making it an efficient and reliable solution for securely moving data to the cloud.

2. Related Work

The landscape of cloud storage and information migration has been appreciably explored in recent years, with a strong emphasis on improving security, performance, and consumer privacy including [1]–[32]. A good sized frame of studies has targeted at the implementation of superior cryptographic strategies, inclusive of encryption and sharding, to shield touchy facts and optimize garage approaches. moreover, the adoption of blockchain era and clever contracts has won traction as a means to offer transparent, tamper-proof records control and get admission to control mechanisms. These innovations goal to cope with the developing worries over facts breaches, unauthorized access, and compliance with stringent regulatory frameworks. This section critiques key contributions and findings in the field, highlighting the combination of encryption, sharding, and blockchain technology, and positioning our work in the broader context of at ease and green cloud storage answers.

Table 1. Related reference findings in our investigation.

Author	Citation	Title	Objectives	Findings
M. A. Alshammari, H. Hamdi, M. A. Mahmood, and A. A. A. El-Aziz (2024)	[1]	Cloud Computing Access Control Using Blockchain.	Secure solution for access control in cloud computing environments using blockchain.	By using blockchain technology efficient, a more secure, scalable, and Transparent access control framework can be implemented.
Ayush Thakur, Sanskar Chauhan, and Ilisha Tomar (2024)	[2]	Self-Healing Nodes with Adaptive Data-Sharding.	Improve Data Storage Efficiency In Cloud Systems.	How breaking down large datasets into smaller components enhances storage efficiency, scalability, and overall performance.
M. Almasian, A Shafieinejad (2024)	[3]	Secure cloud file sharing scheme using blockchain and aatribute-based encryption.	Leveraging blockchain technology for secure access control of the user data.	Using blockchain to implement access control as smart contract where in user can request to access his file by logging a transaction in the blockchain.
G. Sucharitha, V. Sitharamulu, S. N. Mohanty, A. Matta, and D Jose (2023)	[9]	Enhancing Secure Communication in the cloud Through Blockchain Assisted-CP-DABE.	Use of encryption to protect sensitive data.	Usage of Blockchain technology for secure key generation, and for access control while the immutability of the blockchain ensures the confidentiality of ciphertext.
D. Dhinakaran, D. Selvaraj, and N. Dharini. (2023)	[10]	Towards A Novel Privacy-Preserving Distributed Multiparty Data Outsourcing Scheme For Cloud Computing With Quantum Key Distribution.	Encryption And Distribution Techniques In Enhancing Data Privacy And Access Speed.	Leveraging encryption technique to store and migrate data from one source to another.

F. Stodt and C. Reich (2023)	[15]	A Review on Digital Wallets and Federated Service for Future of Cloud Services Identity Management.	Utilizing Digital Wallets For Encryption And Key Management.	Digital wallets can play a key role in both identity of the user as well as security of user's data.
W. Alsuwat and H. Alsuwat (2022)	[17]	A Survey on Cloud Storage System Security via Encryption Mechanisms.	Choosing efficient encryption algorithm which is suitable for cloud environment.	Searchable encryption, attribute-based, identity-based encryption, homomorphic encryption and cloud DES algorithms. Each of the above methods has some limitations and disadvantages.
G. C. Jadhav, K. I. Awale, A. A. Patil, and K. N. Rode (2022)	[18]	Cloud Cryptography.	Use of cryptographic algorithm for secure data.	When users upload or store data during cloud service, the data owner does not seem to understand the path of data transfer. Users do not know whether the data is collected, analyzed and accessed by third parties.
E. Avstein (2021)	[18]	Zero-Knowledge Cloud Storage: What is it and Why You Need it Now.	Utilization of zkcs, users can securely store data on a remote server without disclosing the actual information.	Employ zk-snarks to create zkps for secure data transmission, ensuring that encrypted information remains hidden.
R. Ragul and R. Arokia Paul Rajan (2020)	[23]	Efficient Horizontal Scaling of Databases Using Data Sharding Technique.	Enhancing cloud data protection using aes and rsa encryption.	Combining data privacy and integrity measures in cloud storage approach aligns with the project's current focus on encryption and secure data migration using zero-knowledge proofs (zkps).

F. Zhang, X. Fan, P. Zhou, and W. Zhou (2020)	[24]	Zero Knowledge Proofs for Cloud Storage Integrity Checking.	Efficient and secure storage for decentralized systems for provides valuable insights into zero knowledge proofs for cloud storage integrity checking.	Examine proof-of-replication (porep) to guarantee that storage providers store data in multiple locations, improving security and efficiency.
G. S. Mahmood, D. J. Huang, and B. A. Jaleel (2019)	[25]	A Secure Cloud Computing System by Using Encryption and Access Control Model.	Access control model that can safeguard data in cloud computing.	Employing encryption and access control to guarantee the confidentiality, integrity, and appropriate control of access to sensitive data.
E. K. K. Edris and M. Aiash (2018)	[26]	ZKPVM: A Zero-Knowledge Authentication Protocol for VMs' Live Migration in Mobile Cloud Computing.	Application of ZKPs in both contexts demonstrates their versatility and effectiveness in enhancing security protocols for cloud-based operations.	Employing ZK-SNARKs to generate Zero-Knowledge Proofs, allowing secure and privacy-preserving data migration between cloud services.
C. H. Costa, J. V. B. Moreira Filho, P. H. M. Maia, and F. C. M. B. Oliveira (2015)	[30]	Sharding By Hash Partitioning - A Database Scalability Pattern To Achieve Evenly Sharded Database Clusters,	Hashing partition method to increase the scalability of the database.	Efficient scalable and data management in cloud storage and data migration, underscores the significance of sharding in improving performance and reliability in distributed computing applications.
M. P. Patel, M. I. Hasan, and H. D. Vasava (2014)	[31]	Survey Study On Issues In Mongodb In Cloud Environment.	Security enhancement by using encryption, data fragmentation, and distributed storage methods.	Adopting robust encryption techniques and effective data fragmentation methods which can emphasizes the use of sharding concept and encryption technic for secure user data storage and sharing.

The extensive literature analysis has lead to identify several research gaps. While providing a secure cloud environment efficiency is as much important as user's security and privacy, Our architecture provides an efficient storage system using Sharding concept which plays a huge role in maintaining large files and big data while parallelly providing security for the data. With the use of smart contract it not only provides a secure access control but also stores and secures the metadata of the file which ensures immutability. Zk-SNARKS for secure data transmission from one cloud to another cloud to verify the user data without reveling his actual data provides an extra privacy and security layer for this architecture. The servers are build using zero trust architecture where the server handler has no authority over user's data providing Zero trust environment.

3. Novelty of the Work

The novelty of this investigation lies in its comprehensive integration of cutting-edge cryptographic techniques and innovative data management strategies to tackle the significant challenges of secure and efficient cloud storage and migration. The design's high-level architecture offers several distinctive contributions to the field. Figure 1 represents the high-level architecture of the secure cloud environment.

Encryption and digital wallet integration play a significant role in safeguarding user data privacy, as each file is encrypted using cryptographic keys derived from user-specific digital wallets. This guarantees that data remains protected and accessible exclusively to authorized users, offering a personalized and comprehensive security layer. By utilizing digital wallets for encryption, the system guarantees that each user possesses a distinct encryption key, thereby minimizing the chances of unauthorized access and bolstering overall data security.

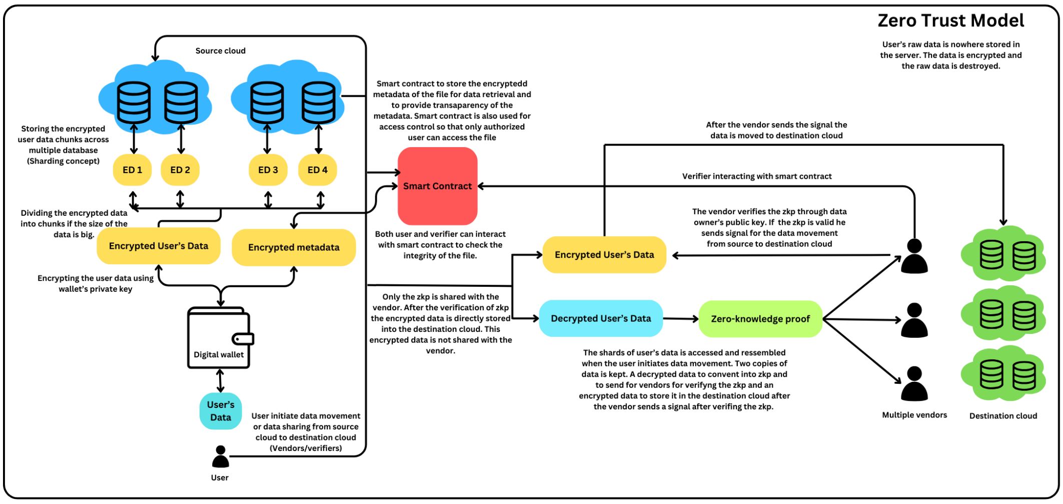
Zero Trust Model
User's raw data is nowhere stored in the server. The data is encrypted and the raw data is destroyed.
Source cloud
Storing the encrypted user data chunks across multiple database (Sharding concept)
ED 1
ED 2
ED 3
ED 4
Dividing the encrypted data into chunks if the size of the data is big.
Encrypted User's Data
Encrypted metadata
Encrypting the user data using wallet's private key
Digital wallet
User's Data
User initiate data movement or data sharing from source cloud to destination cloud (Vendors/verifiers)
User
Smart contract to store the encryptedd metadata of the file for data retrieval and to provide transaparency of the metadata. Smart contract is also used for access control so that only authorized user can access the file
Smart Contract
Both user and verifier can interact with smart contract to check the integrity of the file.
Only the zkp is shared with the vendor. After the verification of zkp the encrypted data is directly stored into the destination cloud. This encrypted data is not shared with the vendor.
After the vendor sends the signal the data is moved to destination cloud
Verifier interacting with smart contract
The vendor verifies the zkp through data owner's public key. If the zkp is valid he sends signal for the data movement from source to destination cloud
Encrypted User's Data
Decrypted User's Data
Zero-knowledge proof
The shards of user's data is accessed and ressembled when the user initiates data movement. Two copies of data is kept. A decrypted data to convent into zkp and to send for vendors for verifyng the zkp and an encrypted data to store it in the destination cloud after the vendor sends a signal after verifing the zkp.
Multiple vendors
Destination cloud

Figure 1. Novel architecture of secure cloud environment.

4. Secure Cloud Environment for Data Storage and Data Transmission

Inorder to provide a comprehensive methodology that combines cutting-edge cryptographic methods, sharding, zero-knowledge proofs (zkps) and smart contracts to guarantee secure and efficient cloud storage and data transfer. The key components of the methodology involve encrypting files using personalized digital wallets, breaking down the encrypted files into shard pieces for effective storage and Improved upload speed, and utilizing zkps for secure data verification in destination cloud. The system's architecture guarantees the protection of sensitive user data at all stages of its existence, employing a zero-trust approach to prevent any unauthorized access by server handlers.

4.1. Sharding and Encryption Modelling

As soon as a user uploads a file through the client-server interface, the file is instantly encrypted using cryptography keys stored in the user's digital wallet as Figure 2. The encrypted shards are then uploaded to the available servers simultaneously which ensures faster uploading time for large sized files or big data along with the hash of the entire file and additional metadata such as the original file name and format, are recorded.

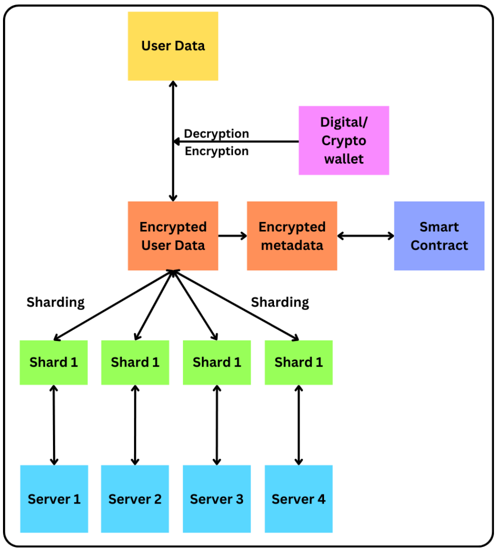

Figure 2. Flow control diagram of **sharding** and **encryption**

4.1.1. Variables and Definitions of the Model

$$\begin{aligned}
F &: \text{The original file as a binary data sequence} \\
|F| &: \text{The size of the original file in bytes} \\
n &: \text{The number of shards (pieces) the file is divided into} \\
P_i &: \text{The } i\text{-th piece (shard) of the file, where } i \in \{1, 2, \ldots, n\} \\
|P_i| &: \text{The size of the } i\text{-th piece} \\
K &: \text{The encryption key} \\
IV &: \text{The initialization vector used for encryption} \\
E_{K,IV}(P_i) &: \text{The encrypted version of the } i\text{-th piece using the encryption algorithm with key } K \text{ and IV} \\
H(x) &: \text{A cryptographic hash function applied to data } x
\end{aligned}$$

4.1.2. Sharding model

The file F is divided into n pieces, where the size of each piece P_i is ideally equal, but may vary slightly due to file size not being perfectly divisible by n. The size of each piece can be represented as:

$$|P_i| = \begin{cases} \left\lfloor \frac{|F|}{n} \right\rfloor + 1 & \text{if } i \leq |F| \mod n \\ \left\lfloor \frac{|F|}{n} \right\rfloor & \text{if } i > |F| \mod n \end{cases}$$

Where: - $|F| \mod n$ is the remainder when the file size $|F|$ is divided by the number of pieces n. - $\lfloor x \rfloor$ denotes the floor function, which rounds down to the nearest integer.

4.1.3. Encryption of Shards

Each shard P_i is encrypted using a symmetric encryption algorithm with a key K and an IV. The encrypted piece E_i is given by:

$$E_i = E_{K,IV}(P_i)$$

4.1.4. Hashing and Metadata

The unique report, each shard, and the encrypted shards may be hashed to offer a unique identifier and make certain integrity. The hashes may be denoted as follows:

$$\begin{aligned} H(F) &: \text{Hash of the original file} \\ H(P_i) &: \text{Hash of the } i\text{-th piece before encryption} \\ H(E_i) &: \text{Hash of the } i\text{-th encrypted piece} \end{aligned}$$

4.1.5. Storing Metadata

The metadata stored for the document and its shards includes the original document hash, the hashes of the shards, and the encryption information. This metadata ensures that the report can be confirmed and reassembled successfully and this metadata is encrypted and stored in blockchain through smart contract enhancing immutable state of the metadata:

$$\text{Metadata} = \{H(F), \text{file\_type}, [H(E_i), i]_{i=1}^{n}, \text{encryption\_algorithm}, \text{K}, \text{IV}\}$$

Where: - $[H(E_i), i]_{i=1}^{n}$ represents the list of hashes for all encrypted shards along with their respective indices. - file_type is the type or format of the original file. - encryption_algorithm specifies the encryption algorithm used

4.2. Data migration using ZK-SNARK

For data migration from the source cloud to the destination cloud, the system utilizes zero-knowledge proofs (zkps) to guarantee secure and private data movement as per Figure 3. The zk-SNARks construction process includes defining the circuit which represents the computation verifying the correctness of the reassembly and integrity of the shards. This circuit is then used for generating proving and verifying keys which are required for the generation of zk-SNARKs.The operations involved in this model contains two phases in order. It starts with the proving phase where the proof of the encrypted file is generated after the decryption of reassembled shards. Followed by verification phase where the verifier in the destination cloud verify the zkp data and sends the verification results to proceed for further operations.

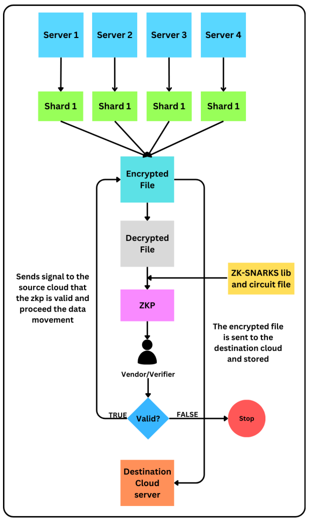
Server 1
Server 2
Server 3
Server 4
Shard 1
Shard 1
Shard 1
Shard 1
Encrypted File
Decrypted File
ZK-SNARKS lib and circuit file
Sends signal to the source cloud that the zkp is valid and proceed the data movement
ZKP
The encrypted file is sent to the destination cloud and stored
Vendor/Verifier
TRUE
Valid?
FALSE
Stop
Destination Cloud server

Figure 3. Flow control diagram of transmission of user data to destination cloud.

4.2.1. Variables and Definitions

$$\begin{aligned}
F &: \text{Original file as a binary data sequence} \\
P_i &: \text{The } i\text{-th shard (piece) of the file} \\
E_i &: \text{The } i\text{-th encrypted shard} \\
H(E_i) &: \text{Hash of the } i\text{-th encrypted shard} \\
r_i &: \text{Random value used for the proof} \\
C_i &: \text{Commitment to the } i\text{-th encrypted shard} \\
\text{PK} &: \text{Proving key} \\
\text{VK} &: \text{Verifying key} \\
\pi &: \text{zk-SNARK proof}
\end{aligned}$$

4.2.2. zk-SNARKs Construction

The setup phase generates public parameters for proof generation and verification.

- **Circuit Definition:** Define a circuit C that represents the computation verifying the correctness of the reassembly and integrity of the encrypted shards.
- **Public Parameters:** Generate proving key (PK) and verifying key (VK):

$$(\text{PK}, \text{VK}) \leftarrow \text{Setup}(C)$$

4.2.3. Proving Phase

- **Inputs and Witness:** The public input x includes the commitments C_i and the hash $H(F)$. The witness w consists of E_i and r_i.

$$x = (C_1, C_2, \ldots, C_n, H(F))$$

$$w = (E_1, E_2, \ldots, E_n, r_1, r_2, \ldots, r_n)$$

- **Proof Generation:** The proof π is generated as follows:

$$\pi \leftarrow \text{Prove}(\text{PK}, x, w)$$

4.2.4. Verification Phase

The verifier checks the proof using the verifying key.

- **Verification:** Verify the proof against the public input:

$$\text{Verify}(\text{VK}, x, \pi) \rightarrow \text{True/False}$$

4.2.5. Summary

$$\begin{aligned}
(\text{PK}, \text{VK}) &\leftarrow \text{Setup}(C) \\
\pi &\leftarrow \text{Prove}(\text{PK}, x, w) \\
\text{Verify}(\text{VK}, x, \pi) &\rightarrow \text{True/False}
\end{aligned}$$

4.3. Smart Contract for Access Control and Metadata Storage

Storing the encrypted metadata on the blockchain, the system guarantees the integrity and authenticity of the metadata, creating an unalterable record of the data. Furthermore, smart contracts are employed to regulate access control, guaranteeing that only authorized users can access and decrypt the files. By adopting this approach, organizations can not only enhance data security but also adhere to strict cloud policies that prioritize user privacy and data protection.

4.3.1. Variables and Definitions

$$\begin{aligned}
\text{Address} &: \text{Unique identifier for a user or entity in the blockchain network}\\
file_id &: \text{Unique identifier for each file}\\
\text{Metadata} &: \text{Information related to the file, including the hash of the original file,}\\
&\quad \text{shard hashes, encryption details, and file type}\\
\text{Access Control} &: \text{Permissions granted to users for accessing or interacting with the file}
\end{aligned}$$

4.3.2. Smart Contract Model for access Control

- **Storing Metadata function:** Refer Sub-section 4.1.5 to know about Metadata function.

$$\text{StoreMetadata}(file_id, H(F), \text{file\_type}, \{H(E_i)\}_{i=1}^{n}, \text{encryption\_algorithm}, \text{K}, \text{IV})$$

- **Access Function:** The smart contract function which controls the access of user data:

$$[\text{Access}_{file_id}](u) = \begin{cases} \text{True} & \text{if user } u \text{ has access} \\ \text{False} & \text{otherwise} \end{cases}$$

Where: - $file_id$ is the identifier of the file. - u is the user address.

- **Grant Access Function:** The smart contract function for granting access to a user:

$$GrantAccess(file_id, u) \rightarrow \text{True}/\text{False}$$

- **Revoke Access Function:** The smart contract function for revoking access from a user:

$$\text{RevokeAccess}(file_id, u) \rightarrow \text{True}/\text{False}$$

- **Check Access Function:** The smart contract function for checking if a user has access:

$$\text{CheckAccess}(file_id, u) \rightarrow \text{True}/\text{False}$$

5. Experimental Results and Discussion

This investigation aimed to analyze the performance of a cloud storage system by comparing the time it takes to upload files when using sharding versus a system without sharding. The main goal was to assess whether the sharding technique provides a substantial improvement in terms of uploading speed. The experiment entailed transferring substantial files using a client-server interface in two situations: with sharding, where the files were divided into smaller encrypted fragments, and without sharding, where the entire file was uploaded as a single unit.

The performance analysis was conducted in a basic local environment using Nodejs to calculate the uploading time with and without using sharding concept and also the time required to generate zkp and verifying process of zkp. The last experiment conducted was to calculated the time taken for the migration of encrypted file to the destination server. The method of implementation was done by deploying the server in 4 different port and one port for destination cloud (assumption as cloud storage). The performance was conducted on a laptop with specification having AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz with 16GB RAM of 64-bit operating system, x64-based processor of Windows 11 Operation system.

The findings of the performance evaluation indicated that the uploading time was noticeably faster when employing the sharding technique. By dividing the encrypted file into smaller fragments, each piece could be processed and uploaded independently and concurrently on multiple servers. The ability to perform multiple tasks simultaneously resulted in a significant reduction in the time it took to upload files. In contrast, the non-sharded approach necessitated the upload of the entire file as a single entity, leading to extended upload times due to the sequential processing and increased resource utilization on a single server.

The below Figure 5 represents the analysis of the uploading time vs file size on varying number of servers. As we can see the uploading time gradually decreases as the no. of servers increases, we observe a significant reduction in the time it took to upload files. As the no. of server increases not only the efficiency of the time increases but also it can tolerate some amount of faulty nodes or servers.

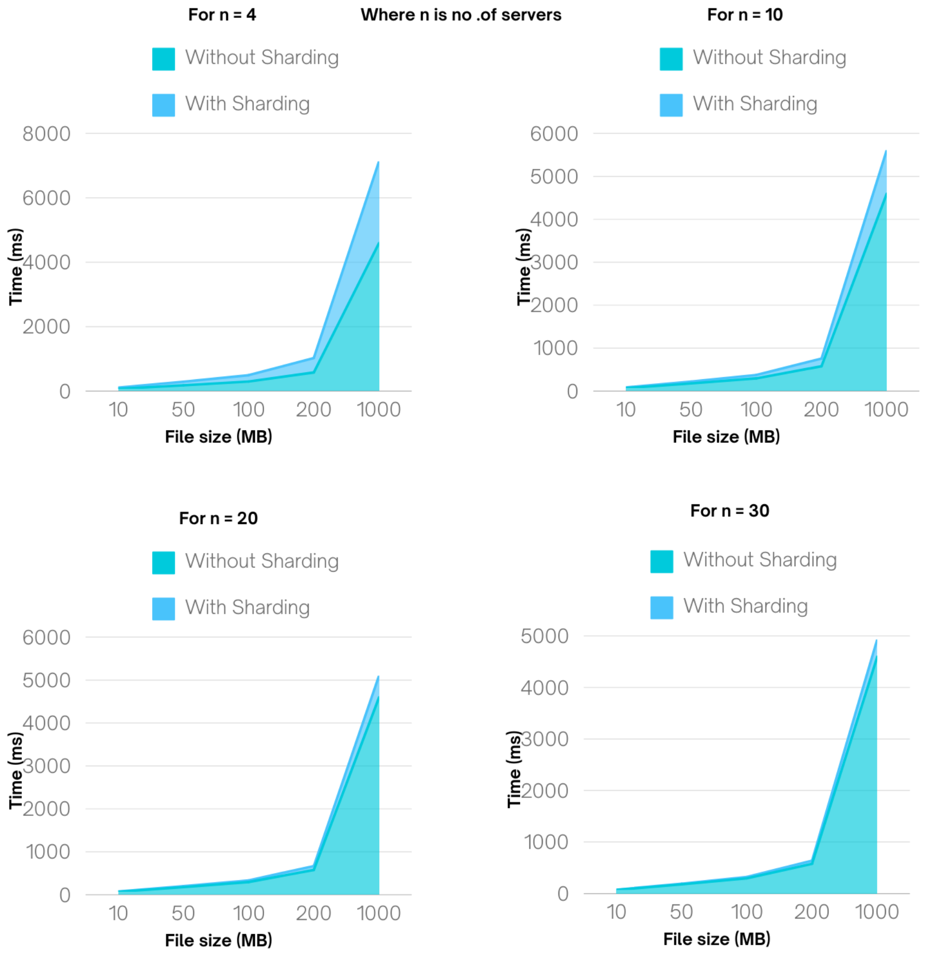

Figure 4. Analysis of time complexity for n no. of servers.

Through the comparison we can see the time function gradually decreases on increasing number of servers **n**. From the above analysis we calculated the mathematical equation which can approximately calculate the time taken on **n** number of servers.

$$T(n) = T(4) \times \frac{4}{n} \tag{1}$$

where:

- $T(n)$ is the sharding time with n servers.
- $T(4)$ is the sharding time with 4 servers.
- n is the number of servers.

Table 2. Results of Uploading time for With and Without sharding of data.

File Size (MB)	**Without Sharding (ms)**	**With Sharding, *for n=4* (ms)**	**With Sharding, *for n=10* (ms)**	**With Sharding, *for n=20* (ms)**	**With Sharding, *for n=30* (ms)**
10	69.5	40.158	16.06	8.03	5.35
50	117.57	112.25	44.9	22.45	14.97
100	295.65	197.109	78.84	39.42	26.28
200	577.76	450.16	180.60	90.03	60.02
1000	4595	2540	1016	508	338.67

We aimed to apprehend the useful resource performance and value implications of various operations inside the settlement. by categorizing transactions into sorts which include storing metadata, granting and revoking access, checking get admission to permissions, and updating metadata, we quantified the fuel intake associated with each. Our findings discovered that operations involving data storage, along with storing and updating metadata, typically ate up the maximum gas because of the higher computational and storage requirements. Conversely, operations like checking get admission to, which by and large involve study operations, ate up substantially less gasoline. This analysis affords valuable insights into optimizing clever contract design to reduce gas costs, especially for frequent transactions, thereby enhancing the general fee-effectiveness and performance of the system deployed at the Polygon blockchain.

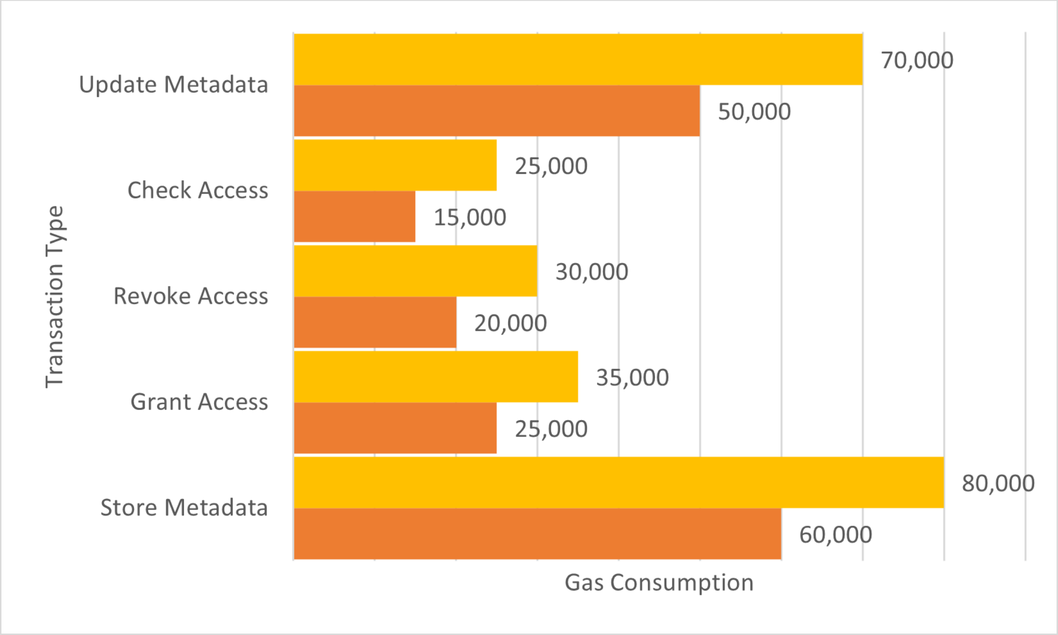

Figure 5. Gas consumption for each transaction type involved in the smart contract.

The gas consumption analysis was conducted on Remix editor where we can deploy and test our smart contracts. We deployed our smart contract on polygon network as it provides higher scalability and less gas fee. The transaction functions are designed to consume less gas while incorporating secure methods that protect against smart contract vulnerabilities.

The evaluation of our project's performance included assessing the time required to produce a zero-knowledge proof (zkp) and transmit it to the verifier in the cloud-based destination. The process starts with putting together and decrypting the encrypted file pieces, then creating the zkp using zksnarks. This phase is crucial as it guarantees the accuracy and reliability of the data being transferred without revealing the actual data. The study showed that the zkp generation and transmission to the verifier were executed effectively, requiring an appropriate amount of time. Upon receiving the zkp, the verifier promptly validated it, ensuring a quick and efficient validation process with minimal latency. This rapid validation process is crucial for ensuring a seamless and secure data migration flow, validating that the data integrity checks are thorough yet efficient.

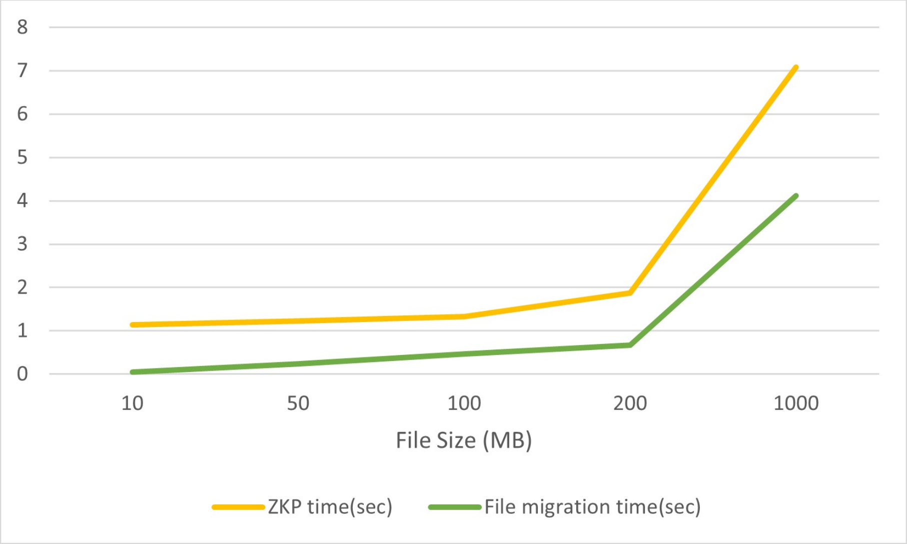

Figure 6. Zkp generation time and file migration time for different file size

The subsequent performance analysis centered around the time it took for the generation of zkp as well as the time taken for the actual migration of the file to the destination cloud. This phase included piecing together the encrypted fragments, ensuring their accuracy, and safely moving the encrypted file to the designated cloud server upon successful verification of Zkp from the destination cloud. Our analysis revealed that the file uploading process with sharding concept for big data and large file, file migration process, following the zkp validation, was carried out with remarkable efficiency. We also carried out the efficiency calculating metric by using the below formula which resulted in approximately **40.94%** for 4 servers, **76.37%** for 10 servers, **88.18%** for 20 servers and **92.12%** for 30 server more efficient when we use sharding concept over a single cloud server. As we can see as the no. of server increases efficiency rate also increases. So more the servers we get more efficient file storage and management can be achieved for big data or large files.

$$\text{Efficiency}(\%) = \left(1 - \frac{\text{Sharding Time}}{\text{Without Sharding Time}}\right) \times 100 \tag{2}$$

By adopting this streamlined approach, the encrypted file can be delivered to the destination promptly and securely, safeguarding the privacy and security of user data throughout the transmission. The integration of efficient zkp validation and optimized file migration highlights the effectiveness of our system in delivering secure and swift data migration services.

6. Conclusions

In culmination of this novel work, it presents a comprehensive framework that ensures the security and efficiency of cloud storage and data migration, utilizing cutting-edge cryptographic techniques, sharding, blockchain smart contracts and zero-knowledge proofs (zkps). By encrypting files using user-specific digital wallets and dividing the encrypted data across multiple local database servers, the system guarantees that sensitive user information remains secure and inaccessible to unauthorized individuals at all stages of its existence. By implementing a zero-trust model, where encrypted data is stored and server handlers have no direct access to raw data, data security and privacy are greatly improved.

Looking into the future, there are plans to introduce additional improvements to the system, aiming to enhance its capabilities and tackle new challenges in managing cloud data. Initially, the implementation will progress to generate zkps by utilizing cryptographic keys directly obtained from the user's digital wallet. This improvement not only strengthens the security of the proof generation process but also guarantees a personalized and comprehensive verification procedure that is specifically designed to meet the unique encryption key requirements of each individual user.

Further, deploying the smart contracts on a zero-knowledge proof (zkp) blockchain will be examined. This strategy intends to enhance the security and privacy protocols of the cloud storage system. By utilizing zkp blockchain technology, the verification of transactions and data access can be conducted without disclosing any sensitive information, ensuring the privacy of users. Utilizing smart contracts on a zkp blockchain will also improve compliance with privacy regulations and cloud policies, guaranteeing that user data is safeguarded to the highest possible standards. This future direction will play a crucial role in the creation of a more robust, secure, and privacy focused cloud storage solution.

These advancements are designed to enhance the system's ability to maintain data accuracy, protect privacy, and optimize operational efficiency in cloud environments. Through the incorporation of state-of-the-art technologies and methodologies, this investigation establishes a solid groundwork for future advancements in secure cloud data storage and migration, propelling the development of reliable and scalable cloud computing infrastructures.

References

1. M. A. Alshammari, H. Hamdi, M. A. Mahmood, and A. A. A. El-Aziz, "Cloud Computing Access Control Using Blockchain,". In *International Journal of Intelligent Systems and Applications in Engineering*, **2024**, vol. 12, no. 9s, pp. 380-390, link: https://ijisae.org/index.php/IJISAE/article/view/4329 .
2. Ayush Thakur, Sanskar Chauhan, and Ilisha Tomar, "Self-Healing Nodes With Adaptive Data-Sharding,". *arXiv preprint*, **2024**, doi: https://doi.org/10.48550/arXiv.2405.00004.
3. Mohammadpayam Almasian and Alireza Shafieinejad, "Secure cloud file sharing scheme using blockchain and attribute-based encryption ". In *Computer Standards Interface*, **2024**, vol. 87, doi: https://doi.org/10.1016/j.csi.2023.103745.
4. I. Hamid and M. Frikha, "Blockchain-Enhanced Cybersecurity and Privacy in Cloud Computing: A Systematic Literature Review,". In *Journal of Theoretical and Applied Information Technology*, **2024**, vol. 102, no. 2, link: https://www.jatit.org/volumes/Vol102No2/10Vol102No2.pdf.
5. Sagarika Behera and Jhansi Rani Prathuri, "FPGA-Based Acceleration of K-Nearest Neighbor Algorithm on Fully Homomorphic Encrypted Data". In *MDPI Cryptography*, **2024**, vol.8, no. 1, doi: https://doi.org/10.3390/cryptography8010008.
6. Chang Chen , Guoyu Yang , Zhihao Li, Fuan Xiao, Qi Chen and Jin Li, "Privacy-Preserving Multi-Party Cross-Chain Transaction Protocols". In *MDPI Cryptography*, **2024**,vol.8, no. 1, doi: https://doi.org/10.3390/cryptography8010006
7. Yinhao Jiang, Mir Ali Rezazadeh Baee, Leonie Ruth Simpson, Praveen Gauravaram, Josef Pieprzyk, Tanveer Zia, Zhen Zhao and Zung Le, "Pervasive User Data Collection from Cyberspace: Privacy Concerns and Countermeasures". In *MDPI Cryptography*, **2024**, vol.8, no. 1, doi: https://doi.org/10.3390/cryptography8010005.

8. Yuri Bespalov 1, Lyudmila Kovalchuk , Hanna Nelasa, Roman Oliynykov, and Rob Viglione, "Models for Generation of Proof Forest in zk-SNARK Based Sidechains". In *MDPI Cryptography*, **2023**, vol.7, no. 1 doi: https://doi.org/10.3390/cryptography7010014.
9. G. SUCHARITHA, VEDULA SITHARAMULU, SACHI NANDAN MOHANTY ,ANJANNA MATTA, AND DEEPA JOSE5, "Enhancing Secure Communication in the Cloud Through Blockchain Assisted-CP-DABE,". In *IEEE Xplore*, **2023**, vol. 11, pp. 99005 - 99015, doi: https://doi.org/10.1109/ACCESS.2023.3312609.
10. D. Dhinakaran, D. Selvaraj, and N. Dharini, "Towards A Novel Privacy-Preserving Distributed Multiparty Data Outsourcing Scheme For Cloud Computing With Quantum Key Distribution", In *International Journal of Intelligent Systems and Applications in Engineering* **2023**, vol. 12, no. 2, link: https://ijisae.org/index.php/IJISAE/article/view/4252.
11. H. Dubey and K. Roy, "Secure Access Control in Cloud Computing Environments: Smart Contract Blockchain,". In *Vidhyayana*, **2023**, vol. 8, no. si7, link: https://www.vidhyayanaejournal.org/journal/article/view/832.
12. Mujahid Sharif and Ayesha Said, "Blockchain and Cloud: Exploring the Intersection of Two Revolutionary Technologies", In *ResearchGate*, **2023**, link: https://www.researchgate.net/publication/372826151.
13. S. N. Prasad and C. Rekha, "Block chain based IAS protocol to enhance security and privacy in cloud computing,". In *Measurement: Sensors*, **2023**, vol. 28, p. 100813, doi: https://doi.org/10.1016/j.measen.2023.100813.
14. E. Jansirani and R. N. Kowsalya, "Analysis of ECC and ZKP Based Security Algorithms in Cloud Data,". In *Journal of Theoretical and Applied Information Technology (JATIT)*, **2023**, vol. 101, no. 16, pp. 6354-6368, link: https://www.jatit.org/volumes/Vol101No16/9Vol101No16.pdf.
15. F. Stodt and C. Reich, "A Review on Digital Wallets and Federated Service for Future of Cloud Services Identity Management,". Presented at the *Fifteenth International Conference on Advanced Service Computing (SERVICE COMPUTATION 2023), Nice, France, June 26-30*, **2023**, pp. 16-20, link: https://www.researchgate.net/publication/372108853.
16. Sachin N. Rajguru and Sumit K. Choubey, "Blockchain in Cloud Computing for Securing Documents,". In *International Research Journal of Modernization in Engineering, Technology and Science (IRJMETS)*, **2023**, vol. 5, no. 5, pp. 123-130, doi: https://www.doi.org/10.56726/IRJMETS38879.
17. W. Alsuwat and H. Alsuwat, "A Survey on Cloud Storage System Security via Encryption Mechanisms,". In *International Journal of Computer Science and Network Security*, **2022**, vol. 22, no. 6, pp. 52-61, link: http://paper.ijcsns.org/07_book/202206/20220626.pdf.
18. G. C. Jadhav, K. I. Awale, A. A. Patil, and K. N. Rode, "Cloud Cryptography,". In *International Journal of Research Publication and Reviews (IJRPR)*, **2022**, vol. 3, no. 6, pp 2200-2202, link: https://ijrpr.com/uploads/V3ISSUE6/IJRPR5029.pdf.
19. E. Avstein, "Zero-Knowledge Cloud Storage: What is it and Why You Need it Now,". In *Codemotion Magazine, Artical*, **2021**, link: https://www.codemotion.com/magazine/devops/cloud/zero-knowledge-cloud-storage-what-is-it-and-why-you-need-it-now.
20. Sudakshina Mandal, Danish Ali Khan and Sarika Jain , "Cloud-Based Zero Trust Access Control Policy: An Approach to Support Work-From-Home Driven by COVID-19 Pandemic,". In *New Generation Computing*, **2021**, vol. 39, pp 599-622. DOI: https://www.doi.org/10.1007/S00354-021-00130-6.
21. Priyanka Ghosh,"The State-of-the-Art in Zero-Knowledge Authentication Proof for Cloud". In *IEEE Xplore*, **2021**, edition: 1, pp 149 - 170, DOI: https://www.doi.org/10.1002/9781119764113.ch8.
22. I. Hamid and M. Frikha, "A Review on Cryptography in Cloud Computing ,". In *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, **2024**, vol. 6, no. 6, pp. 225-230, doi: https://doi.org/10.32628/CSEIT206639.
23. R. Ragul and R. Arokia Paul Rajan, "Efficient Horizontal Scaling of Databases Using Data Sharding Technique,". In *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, **2020**,International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, no. 5, pp. 590-593, DOI: https://www.doi.org/10.35940/ijitee.E2418.039520.
24. F. Zhang, X. Fan, P. Zhou, and W. Zhou, "Zero Knowledge Proofs for Cloud Storage Integrity Checking,". In *arXiv*, **2019**, DOI: https://doi.org/10.48550/arXiv.1912.00446.
25. G. S. Mahmood, D. J. Huang, and B. A. Jaleel, "A Secure Cloud Computing System by Using Encryption and Access Control Model,". In *Journal of Information Processing Systems*, **2019**, vol. 15, no. 3, pp. 538-549. DOI: https://doi.org/10.3745/JIPS.03.0117.

26. E. K. K. Edris and M. Aiash, "ZKPVM: A Zero-Knowledge Authentication Protocol for VMs' Live Migration in Mobile Cloud Computing,". In Proceedings of the *13th International Conference on Software Technologies (ICSOFT)*, **2018**, pp. 858-864, link: https://www.researchgate.net/profile/Mahdi-Aiash/publication/326926039.
27. A. Shaik, B. Madhurima, and M. Neelakantappa, "An Approach To Zero Knowledge Proof For Secure Data Sharing In Cloud Storage: New Direction,". In *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, **2018**, vol. 8, no. 2S, pp. 195-201, link: https://www.ijitee.org/portfolio-item/bs2703128218.
28. T. Jain and J. A. Khan, "Secure Big Data Access Control Policies for Cloud Computing Environment," In *International Journal of Innovative Research in Computer Science and Technology (IJIRCST)*, **2017**, vol. 5, no. 2, pp. 254-256, DOI: https://doi.org/10.21276/ijircst.2017.5.2.8.
29. S. Bagui and L. T. Nguyen, "Database Sharding: To Provide Fault Tolerance and Scalability of Big Data on the Cloud,". In *International Journal of Cloud Applications and Computing (IJCAC)*, **2015**, vol. 5, no. 2, pp. 36-52. DOI: https://doi.org/10.4018/IJCAC.2015040103.
30. C. H. Costa, J. V. B. Moreira Filho, P. H. M. Maia, and F. C. M. B. Oliveira, "Sharding By Hash Partitioning - A Database Scalability Pattern To Achieve Evenly Sharded Database Clusters,". In Proceedings of the *17th International Conference on Enterprise Information Systems (ICEIS), Barcelona, Spain*,, **2015**, vol. 2, pp. 313-320, DOI: https://doi.org/10.5220/0005376203130320.
31. M. P. Patel, M. I. Hasan, and H. D. Vasava, "Survey Study On Issues In Mongodb In Cloud Environment,". In *International Journal of Advanced and Innovative Research*, **2014**, vol. 3, no. 1, link: https://ijairjournal.in/wwwroot/temp/ijair/jan2014/t5.pdf.
32. Balasubramaniam. S and V. Kavitha, "A survey on data encryption tecniques in cloud computing", **2014**, vol. 13, no. 9, pp. 494-505, link: https://www.researchgate.net/publication/287914941.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.