Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.025906
Article

Tech Science Press

A Virtual Cloud Storage Architecture for Enhanced Data Security

M. Antony Joans Kumar[1,*], C. Christopher Columbus[2], E. Ben George[3] and T. Ajith Bosco Raj[4]

[1]Department of Information Technology, St. Xavier's Catholic College of Engineering, Nagercoil, 629003, Tamil Nadu, India
[2]Department of Computer Science Engineering, PSN College of Engineering and Technology, Tirunelveli, 627152, India
[3]Department of Computer Science Engineering, University of Technology and Applied Sciences, Muscat, 112, Oman
[4]Department of Electronics and Communications Engineering, PSN College of Engineering and Technology, Tirunelveli, 627152, India
*Corresponding Author: M. Antony Joans Kumar. Email: antonyjoanskumar@gmail.com

Received: 08 December 2021; Accepted: 14 January 2022

Abstract: The sensitive data stored in the public cloud by privileged users, such as corporate companies and government agencies are highly vulnerable in the hands of cloud providers and hackers. The proposed Virtual Cloud Storage Architecture is primarily concerned with data integrity and confidentiality, as well as availability. To provide confidentiality and availability, the file to be stored in cloud storage should be encrypted using an auto-generated key and then encoded into distinct chunks. Hashing the encoded chunks ensured the file integrity, and a newly proposed Circular Shift Chunk Allocation technique was used to determine the order of chunk storage. The file could be retrieved by performing the operations in reverse. Using the regenerating code, the model could regenerate the missing and corrupted chunks from the cloud. The proposed architecture adds an extra layer of security while maintaining a reasonable response time and storage capacity. Experimental results analysis show that the proposed model has been tested with storage space and response time for storage and retrieval. The VCSA model consumes 1.5x (150%) storage space. It was found that total storage required for the VCSA model is very low when compared with 2x Replication and completely satisfies the CIA model. The response time VCSA model was tested with different sized files starting from 2 to 16 MB. The response time for storing and retrieving a 2 MB file is 4.96 and 3.77 s respectively, and for a 16 MB file, the response times are 11.06 s for storage and 5.6 s for retrieval.

Keywords: Erasure code; regenerating code; cryptography; virtual cloud storage architecture; circular shift chunk allocation algorithm

1 Introduction

Cloud storage offers remote service to the users for storing and retrieving their files over the internet through cloud service providers. Cloud storage model stores data in distributed and virtualized groups of storage that are owned by third-party vendors. Various companies like Google, Amazon own massive data centers and this storage can be leased or purchased by any individual or organization [1]. This heavily eases the owners of the data from possessing their local storage and its maintenance. By keeping the data in a third-party cloud, the data owners are mostly away from the issues related to availability, dependability, and security [2]. Based on the requirements of the user, the cloud storage service providers offer virtualized resources to store the user data or files. The user may feel that they are using single storage, but physically it may be using multiple servers.

Small and Medium Enterprises (SME) need to spend a huge amount of money for setting up their own IT infrastructure to provide their services. Cloud computing technology is a saviour for these SMEs by providing a docile IT architecture that can be accessed by any device through the internet. This technology allows the users to utilize the advantages of improved efficiency of their software systems and applications.

Despite the enormous advantages of cloud computing, there are many concerns raised about the security and integrity of the data stored in the cloud, since the users, companies, and governments store their sensitive information [3]. The users of cloud computing pay for their service and expect more returns from the providers. These issues related to security may impact the prospective users to opt for the cloud model. Assuring cloud data security and integrity is the need of the hour for the service providers and users. The security model should enable the users to perform verification and validation on their data and to confirm that it is secured and not tampered with [4]. From the user's point of view, the levels of security provided should not impact the speed and the efficiency of the applications using the cloud data.

Various researchers have done a vast scale of work in the area of secure cloud data storage and the most promising literature is presented here. A threshold-based public key encryption, proxy re-encryption techniques along a decentralized erasure code on cloud data were proposed to secure the cloud data. The system also enables users to securely store a huge volume of data and also allows users to forward data very easily to another user without retrieving it [5]. Role-Based Encryption (RBE) technique uses cross-cloud storage architecture with cryptographic techniques and assigns different roles to users and privileges. The author of this study explores a mechanism that allows an organization to store data securely either in a public or private cloud-based on the importance of data [6]. A distributed storage integrity auditing method was designed using the homomorphic token and distributed erasure code. The audit uses very less computation and communication cost and assures the correctness of the data [7,8]. The privacy-preserving approach separates the normal data and the sensitive data in the file. Any sensitive data stored in the file will be masked and not available for any unauthorized users; furthermore, authorized users will have to use the key to unmask the sensitive information after retrieving it from the cloud [9].

In the Cloud-RAID model, the user data is first encrypted and distributed across cloud storage providers using Redundant Array of Independent Disks (RAID) technology to achieve confidentiality and availability of the user data [10,11]. Bandwidth and Maintenance minimum Cloud system (BMCloud) was developed using E-Code and JUDGE_STYLE algorithms. This system uses low bandwidth during recovery and also tries to address the issues related to consistency and storage cost [12]. The SaveMe model combines both the public and private cloud storage and provides security from the client-side. The model also uses encryption and encoding techniques to achieve data confidentiality and integrity [13,14]. DEPSKY system uses encryption, encoding, and replication techniques for secure cloud data storage. It used four commercial clouds and a protocol for the clients to access the service from different countries [15]. FriendBox: a hybrid Friend-to-Friend (F2F) personal storage system that allows the friends in a group called trusted friends to share the storage among themselves assuming there is no violation of data leakage [16].

The cloud of cloud model used a hybrid approach called HyRD, where the large files are erasure-coded and stored in different clouds, whereas small files and metadata are stored in high-end cloud storage using replication [17,18]. A semantic data splitting approach is used by the authors to store data both in local storage and cloud. The algorithm determines the parts of the file that are important and gets them stored in the local storage and the other parts of the file will be stored in different cloud storages [19]. In the multi-cloud storage systems model, encryption, partitioning, and encoding techniques were used to achieve security, privacy, and reliability [20]. An innovative approach uses a partitioning function to split large files into smaller parts and encrypt them. This method also uses the swarm intelligence algorithm to optimize the segregated chunks [21]. Innovative DNA-based computing technology is used to compute a 1024 bit key for securing very big files in a cloud. The key is generated based on the user's attributes, system (Media Access Control) MAC address, and many more personal attributes [22].

From the available literature, it is clear that the storage system has to use data redundancy schemes like replication and erasure code, to make the data available at any time without any loss. Replication is the most commonly used technique that stores redundant multiple copies of the same file in multiple clouds which in turn increases the storage cost. The erasure code is a much superior scheme where an object O is split into k fragments and these fragments are processed to produce k + n redundant blocks [23–25]. The original fragment can be reconstructed with any of the available k fragments. Along with erasure code, a few of the following techniques like a cloud of cloud, encoding, encryption, optimization were added advantages for the secure cloud data storage [26,27]. There are many security algorithms that can be implemented in the cloud. Cloud computing, however, requires security algorithms that allow a linear search of deciphered data that guarantees the security of the data [28]. A robust security approach using GA has been introduced for cloud data security. It is simple and easy to implement having only two main processes of crossover and mutation [29].

A Cloud storage system is a storage system over a large scale that consists of many independent storage servers. During recent years a huge change and adoption have been seen in cloud computing so security has become one of the major concerns in it. As Cloud computing works on a third-party system so security concern is there not only for customers but also for service providers.

This paper introduces an innovative technique called the Virtual Cloud Storage Architecture (VCSA) model to store and retrieve the data economically and securely using multiple clouds. In the proposed architecture, the storage of a file in multiple clouds has to go through four layers of tasks: encryption, encoding, hashing, and allocation and retrieval of the file have to pass through the same steps in reverse order. For storing the file in multiple clouds using the VCSA approach, the file was passed to the front-end cloud system and the complete file was encrypted using the auto-generated key. Then the encrypted file was encoded into different chunks using the simple regenerating codes. Next, the encoded chunks were hashed separately using the MD5 algorithm, and each chunk was allocated to one of the available private or public cloud storage using the newly proposed Circular Shift Chunk Allocation (CSCA) algorithm. The retrieval process obtained the metadata about the file and then applies the CSCA algorithm to locate the chunks from different cloud storage. The collected chunks were hashed separately and if any of the chunks were found missing or tampered with then, it was recovered using the regenerating code. The chunks were aggregated together to form a single file using the decoding process and finally, the file was decrypted to obtain the actual file. The performance of this innovative VCSA technique was fair enough when compared to the other techniques in terms of cost, time, and extended level of security. The main aim of this paper is to ensure distributed cloud data storage security to fulfill the CIA (Confidentiality, Integrity, and Availability) model. The architecture presented in this paper is easily scalable to any number of clouds and for files of any size. The availability feature of this system ensures the regeneration of chunks depending upon the number of parity chunks.

The rest of the paper is organized as below. In Section 2, the proposed virtual cloud storage method is presented and performance evaluation is given in Section 5. The paper is concluded in Section 4 with directions for future research.

2 Virtual Cloud Storage Architecture

The VCSA model is a novel approach that enables users to store their files securely in cloud storage without worrying about its security. The model integrates multiple public clouds in the background to provide a single virtual storage system for the users. The model splits the file into chunks and stores it in different clouds and can be collected and reorganized only by the user. Many of the cloud storage systems may fail or not be available, but this model helps to recover the file chunks that are even lost. The VCSA model uses 4 levels of operation to ensure the security of the data during storage and retrieval. In the first layer, a front-end private cloud system gets the user file and it is encrypted using the auto-generated key. In the second layer, the encrypted file is encoded into different chunks using the Simple Regenerating codes. In the third layer, the encoded chunks are hashed separately using the MD5 algorithm, and the corresponding message digest is stored in the metadata file. The last layer uses a special chunk allocation algorithm to decide the order of chunk storage in the available 'n' public or private clouds.

2.1 System Model

The Virtual Cloud Storage Architecture provides a simple and efficient means to store and retrieve data securely in private or public cloud storage. Fig. 1 depicts the overall architecture of the proposed VCSA model that has two major modules: file storage and retrieval. VCSA front-end uses a private cloud running with a minimum storage capacity. After successful login, the user can perform four file operations such as upload, download, list, and delete. When the user uploads a file, only the metadata will be stored in the private cloud and the data chunks are stored in different public clouds. The authentication details of all public clouds are also stored in the private cloud. This architecture ensures the confidentiality and secrecy of the stored data using the four layers of security which is impossible for any hackers to crack.

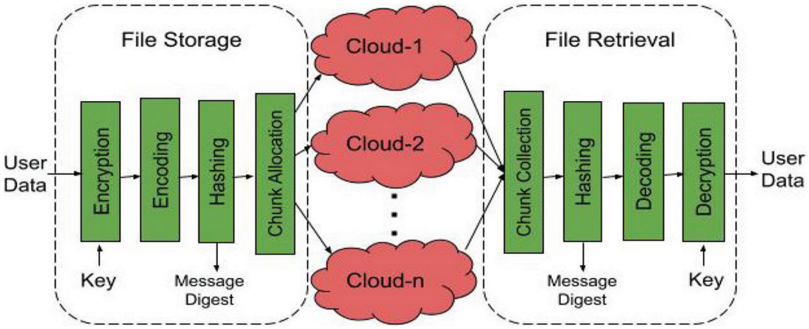

Figure 1: Virtual cloud storage architecture

2.2 File Storage Module

In most of the intercloud or cloud of cloud models, either all or few of the techniques like encryption, encoding, and partitioning processes were used to store the files securely in multiple clouds which ensure only data confidentiality and availability. Ideally, a foolproof security model should fulfill all three principles: Confidentiality, Integrity, and Availability, which is termed as CIA triad model. To guarantee that the VCSA model is a satisfying CIA model, an additional integrity check using hashing technique is added on top of other techniques. Fig. 2 demonstrates the process of file storage in the cloud using the VCSA model. The four layers of the file storage operation are explained in the below steps.

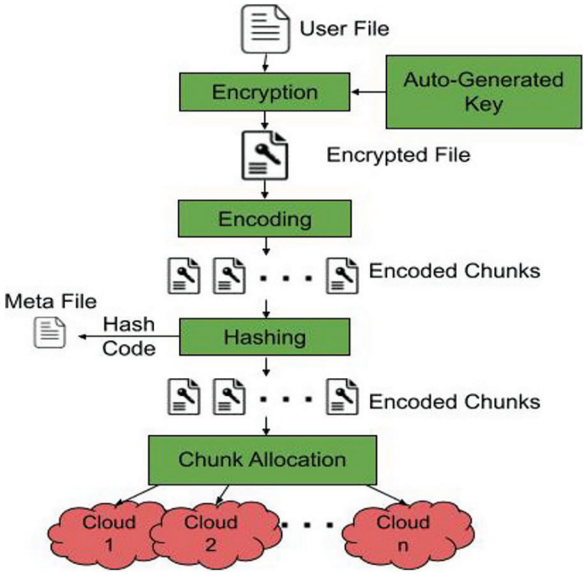

Figure 2: File storage module

Cloud storage services provide multiple levels of security like End-to-End encryption, Server-side encryption, Client-side encryption, and so on [30]. But these encryptions do not guarantee that the cloud providers are not snooping around the user data stored in their clouds. For securing user files from internal and external threats, it is mandatory to encrypt them. The Advanced Encryption Standard (AES) algorithm is one of the best and the most powerful techniques used to encipher the file [31]. The first phase of the VCSA model uses the symmetric encryption-based AES algorithm with a 128-bit key to providing enhanced security to the file. Since the algorithm uses the key, it will be cumbersome to store and secure the key used for the encryption process. To overcome this difficulty, the VCSA model uses an auto-generated key for encrypting the file to be stored in the cloud.

The encrypted file is divided into several chunks using the Maximum Distance Separable (MDS) technique [32–34]. The number of chunks will be based on the number of clouds available for storage purposes. The simple Regenerating Code (SRC) technique [35] is used to encode every chunk using the XOR operation that allows the message to be recovered. Consider a file of size ‘M’ with (n, k) property, where ‘n’ chunks with ‘k’ parities can lead to recovery of ‘k’ lost chunks. The file is initially split into ‘n’ equal parts, and then they are MDS encoded into ‘A’ and the ‘B’ parts. These ‘A’ and ‘B’ chunks are XORed to get the parity chunk ‘S’. These ‘A_i’, ‘B_i’, and ‘S_i’ chunks are stored in different clouds, where i vary from 1 to n.

After the file is encoded into different chunks, an additional level of integrity check may safeguard every chunk. Hashing technique can be used to check the integrity of every chunk and to make sure that the chunks have not been tampered. A message digest or hash code is generated for all chunks using the MD5 hashing algorithm [36,37] and the 32-digit hexadecimal number is stored in the metadata file for checking the integrity during the retrieval process.

All the hashed chunks are to be stored in different public clouds in a specific order to retrieve them easily. If the chunks are stored in sequential order, the cloud providers or hackers may easily collect the chunks, which may lead to a serious security issue. To enhance the level of security, the order of storage of chunks for every file should be different. A novel chunk allocation algorithm named Circular Shift Chunk Allocation (CSCA) algorithm was devised to store the chunks. The chunks are stored in different public or private clouds circularly depending upon the size of the chunk as explained in the below steps.

Let C_1, C_2, … C_p are the available public ‘p’ clouds and ‘cs’ is the size of a chunk.

A_1, A_2, … A_p is the first part of the data chunks.

B_1, B_2, … B_p is the remaining part of the data chunks.

S_1, S_2, … S_p is the parity chunks.

The seed value (sv) is calculated for every file using the Eq. (1)

$$sv = cs \bmod p \ldots \tag{1}$$

The following Eq. (2) illustrates about the order of chunks stored in a particular cloud C_i

$$\text{Chunks stored in cloud } C_i = \left[A_{((sv+i)\text{mod } p)+1}\right], \left[B_{((sv+i+1)\text{mod } p)+1}\right], \left[S_{((sv+i+2)\text{mod } p)+1}\right] \ldots \tag{2}$$

where i = 1,2, … p

Consider a model with 4 clouds (p = 4) and a chunk size of 50 bytes, the seed value is calculated as (sv = 50 mod 4, which is 2). The chunks stored in cloud $C_1 = [A_{((2+1)\text{mod } 4)+1}], [B_{((2+1+1)\text{mod } 4)+1}], [S_{((2+1+2)\text{mod } 4)+1}] = [A_4, B_1, S_2]$. The following Tab. 1 shows the order of chunk storage using the CSCA algorithm for the given scenario. By observing Tab. 1, the CSCA algorithm does a unique circular shift for every part of the file (A, B, S) based on the seed value.

Table 1: Chunk allocation using CSCA algorithm

Cloud1 [C_1]	Cloud2 [C_2]	Cloud3 [C_3]	Cloud4 [C_4]
A_4	A_1	A_2	A_3
B_1	B_2	B_3	B_4
S_2	S_3	S_4	S_1

2.3 File Retrieval Module

The file is stored in multiple clouds after passing through the four layers of operations. Retrieving the complete file from multiple clouds has to pass through the reverse operations of file storage as shown in Fig. 3. The cloud interface shows the list of files available in the cloud storage. Once the user selects the file to be retrieved, the cloud service collects and integrates all the required chunks from the clouds based on the metadata. The metadata contains the file details like file name, file size, chunk name, and a message digest of each chunk. If any of the required chunks are not available or tampered with, the VCSA model will rebuild them using the remaining chunks. The following steps clearly explain the file retrieval process.

The order in which the chunks are stored in the clouds varies from file to file based on the chunk size, and this gives an extra level of security to the user files. For the file to be retrieved, the chunk size (cs) information stored in the metadata is to be given to the CSCA algorithm. The CSCA algorithm will find the order of the chunks to be collected from different clouds for a specific user file. Only the data chunks will be downloaded from multiple clouds and the parity chunks will be downloaded only if rebuilding is required. For every chunk that is collected from the cloud, the message digest is computed and compared with the corresponding message digest stored in the metadata file. If both the message digests match, then it is foolproof that the chunks have not been tampered with; otherwise, the required chunks can be downloaded and rebuilt. Once all the chunks are collected, the decoding phase merges all the chunks to form a single file in order based on SRC and MDS techniques. Finally, the decoded file is decrypted using the same key used during encryption by applying the AES algorithm. Finally, the original file is downloaded from the multiple virtual clouds to the local computer or mobile device.

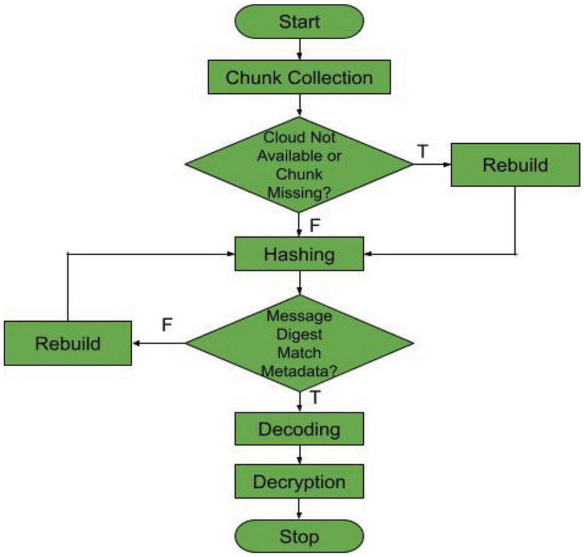

Figure 3: File retrieval process

2.4 Chunk Rebuilding

If any of the participating clouds is down or a chunk is unavailable during chunk collection or a chunk is tampered with during the hash verification, then the chunk should be rebuilt with the support of other chunks using the CSCA algorithm.

For example, if the cloud C_i is unavailable, the chunks $[A_{((sv+i)mod\ p)+1}]$, $[B_{((sv+i+1)mod\ p)+1}]$, $[S_{((sv+i+2)mod\ p)+1}]$ are found to be missing. Using the Simple Regeneration code technique and CSCA algorithm, the chunks can be recovered using the following steps.

To get chunk A

Download $B_{((sv+i)mod\ p)+1}$ from $C_{((sv+i)mod\ p)+1}$

Download $S_{((sv+i)mod\ p)+1}$ from $C_{((sv+i-1)mod\ p)+1}$

Rebuild $A_{((sv+i)mod\ p)+1} = S_{((sv+i)mod\ p)+1}$ XOR $B_{((sv+i)mod\ p)+1}$

To get chunk B

Download $A_{((sv+i+1)mod\ p)+1}$ from $C_{((sv+i+2)mod\ p)+1}$

Download $S_{((sv+i+1)mod\ p)+1}$ from $C_{((sv+i)mod\ p)+1}$

Rebuild $B_{((sv+i+1)mod\ p)+1} = S_{((sv+i+1)mod\ p)+1}$ XOR $A_{((sv+i+1)mod\ p)+1}$

To get chunk S

Download $A_{((sv+i+2)mod\ p)+1}$ from $C_{((sv+i+3)mod\ p)+1}$

Download $B_{((sv+i+2)mod\ p)+1}$ from $C_{((sv+i+2)mod\ p)+1}$

Rebuild $S_{((sv+i+2)\mathrm{mod}\ p)+1} = A_{((sv+i+1)\mathrm{mod}\ p)+1}$ XOR $B_{((sv+i+1)\mathrm{mod}\ p)+1}$

From the above example, to rebuild a chunk (for example chunk A), it is required to download 2 chunks (chunks B and S). So, to rebuild all the three chunks from cloud C_i, a total of 6 chunks of downloads are required. The regenerated chunks can be used to reconstruct the actual file. Thus, the user can download the original file securely even if a part of the file is not available or altered.

3 Evaluation

This section provides the performance evaluation of the proposed VCSA model with other similar techniques. The VCSA model satisfies the CIA model by providing high-end security for cloud files using four levels of security. In the normal setup, the complete file can be stored in a single cloud with less time and space without satisfying the CIA model. Since the prime requirement is to ensure all the attributes of the CIA model, it needs to square up with the time and space constraints. But the level of security operations should not degrade the efficiency and quality of the cloud storage. The space and the time complexities were the main parameters considered for the evaluation purpose [37–40]. This architecture is tested using a cloud computing infrastructure with HP Proliant DL385 G7 server as cloud controller configured using OpenNebula. To evaluate the VCSA model, four popular public cloud storage such as Google Drive, Microsoft OneDrive, Amazon Drive, Dropbox were used.

3.1 Storage Space

In general, a file of size ‘x’ MB takes up the same ‘x’ MB (Mega Byte). storage space when stored in a single cloud. To implement the CIA model fully or partially, the storage space of the file will increase. For the evaluation purpose, different files of sizes of 1 to 16 MB were stored in different clouds using VCSA and 2x replication techniques, and the total space occupied was recorded. The following Fig. 4 shows the percentage of storage space used by the different techniques for a file. For a file of size ‘x’, it can be observed that the 2x Replication technique takes 2x (200%) storage space since the same file is replicated in two different clouds. The VCSA model consumes 1.5x (150%) storage space. It was found that total storage required for the VCSA model is very low when compared with 2x Replication and completely satisfies the CIA model.

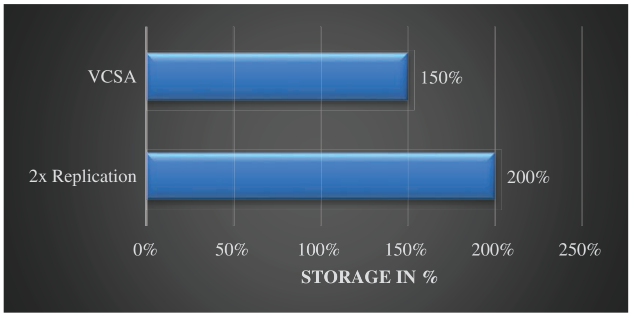

Figure 4: Storage space required for VCSA model and 2x replication

3.2 Response Time

The performance of the VCSA model is evaluated based on the time to perform both the storage and retrieval process. The response time is the total time taken for the model to completely store the file or retrieve the file back from the cloud storage without any rebuilding. The following Fig. 5 shows the comparison of response time for the storage and retrieval process using the VCSA model. The response time VCSA model was tested with different sized files starting from 2 to 16 MB. The response time for storing and retrieving a 2 MB file is 4.96 and 3.77 s respectively, and for a 16 MB file, the response times are 11.06 s for storage and 5.6 s for retrieval.

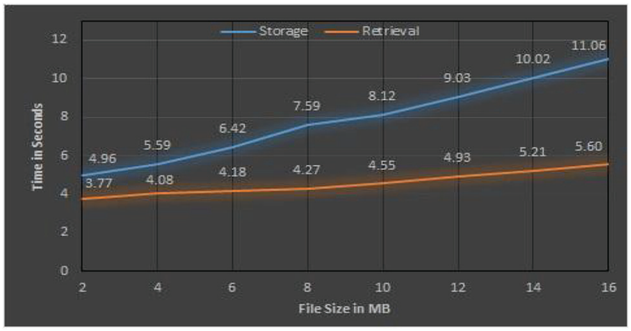

Figure 5: Response time for storage and retrieval

From the chart, it is evident that the retrieval process takes comparatively less time than the storage process. The main reason for the variation in response time is as follows: during the encoding operation, the file is split into data and parity blocks, but during the decoding operation, the parity chunks don't get downloaded. The second inference from the chart is that the response time for the file retrieval process is almost flat and has less variation for the larger files. Also, the time difference between storage and retrieval widens due to an increase in the file size which indicates that the model performs much better for large-sized files than the smaller ones. The reason for this variation in time is that the encoding operation splits the whole file into equal and smaller numbers of chunks and stores them in different clouds that almost consume the same time for small and large files.

The security of the file is directly proportional to the response time. If a file tends to comply with the CIA model, it has to use extra computation which consumes time. The following Fig. 6 shows the response time for a file size of 1 MB with different security options. In the first option "without ACI," the file is stored in the cloud using the simple upload technique that does not apply the CIA model. 'With A' represents the storage model that uses only the availability criteria, which means the file is erasure-coded and stored on the cloud that can recover any data loss. The third option "with AC" satisfies the confidentiality and availability components of the CIA model and in which the file is only encrypted and erasure-coded. The last option "with ACI" complies with the CIA model and is adopted in the VCSA model. This option uses encryption, erasure coding, hashing, and allocation. From the graph, the response time for the retrieval process is much lesser than the storage process. Among the four options, 'without ACI' has less response time and all the other options have almost similar response times. For the retrieval process, the four different security options without ACI, with A, with AC, and with ACI consumes 3.632, 3.763, 3.766, and 3.770 s respectively. From this, it is obvious that the VCSA model provides an extra level of security using hashing which takes very less additional time which can be negligible.

The third time-related evaluation is the rebuilding time. Rebuilding time is the time taken to reconstruct chunks when the chunks have gone missing or tampered. The rebuilding process only happens in rare cases, when the chunks have gone missing or altered. Fig. 7 demonstrates the response times for rebuilding various file sizes from 2 to 16 MB. This experiment is considered for a single cloud failure where it has to rebuild 3 chunks. The rebuilding of 3 chunks for a file of size 1 MB takes 0.12 s; whereas, the rebuilding time for 16 MB in 1.24 s.

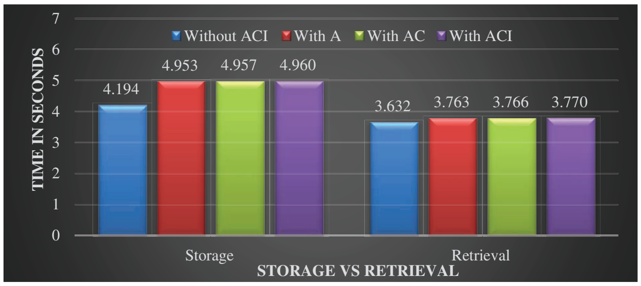

Figure 6: Response time for file storage and retrieval with different security options

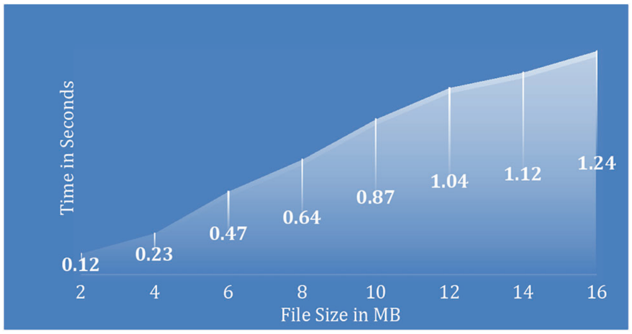

Figure 7: Rebuilding time for single cloud failure

3.3 Storage and Retrieval Overheads

The VCSA model uses four layers of operation and it is required to know about the percentage of time consumed for each layer computation. The model is tested with different file sizes from 2 to 16 MB and the storage overhead percentage is calculated as the average time for all the mentioned sizes. Fig. 8 shows the percentage of time taken for the various operations in the VCSA model during the file storage. Chunk allocation and uploading to different clouds take 75.83%, which is a major share of the total time and also varies depending on the network speed. The next time-consuming task is the Encoding operation, which takes 22.43%. Encryption time (1.01%) and Hashing time (0.73%) consume very less time when compared with other operations. It can be noted the additional integrity through hashing consumes a very less percentage of the time.

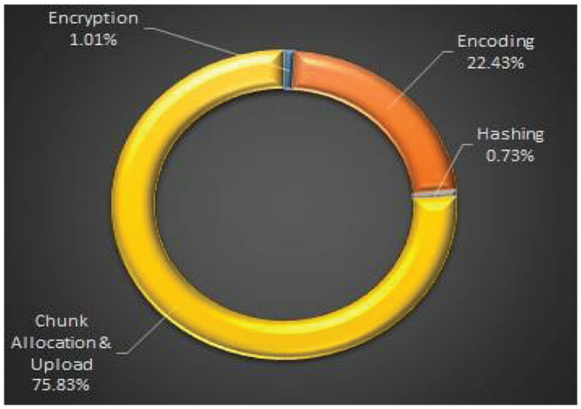

Figure 8: File storage time in percentage

The percentage of time taken for the various operations in the VCSA model during the file retrieval is given in Fig. 9. The time mentioned is the average time for carrying out the operations for file size from 2 to 16 MB. Similar to the file storage time, the chunk collection and downloading operation consume 90.85% of the total time. The remaining operations take less percentage of time with Decoding (7.45%), Decryption (0.75%), and Hash checking (0.95%). It shows that Decryption and hashing take less time and provide the much-required confidentiality and integrity for the data.

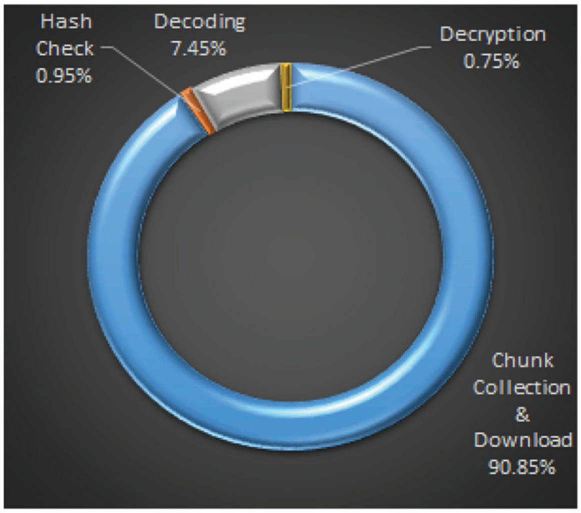

Figure 9: File retrieval time in percentage

4 Conclusion

This paper presented an approach, termed the VCSA model, to store and retrieve files in multiple clouds without compromising its objectives like usability, confidentiality, integrity, and availability. The model implemented these goals with the aid of passing the documents through four levels of operations including encryption/decryption, encoding/interpreting, hashing, and allocation/collection for storing and retrieving files securely. The VCSA model provided the required integrity with the help of hashing which ensured that the file has not been tampered. Moreover, an extra level of security to the file was provided by distributing the chunks to diverse clouds using the CSCA algorithm. The performance of the VCSA model was evaluated with different available models and found that it offered the required security with optimal time and storage. The VCSA model was more effective for files that do not require frequent read-write operations. However, more study is required for handling files that require frequent transactions. Proposed work can be enhanced by implementing the least privilege model for additional security.

Acknowledgement: The author with a deep sense of gratitude would thank the supervisor for his guidance and constant support rendered during this research.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] H. Chung, J. Park, S. Lee and C. Kang, “Digital forensic investigation of cloud storage services,” *Digital Investigation*, vol. 9, no. 2, pp. 81–95, 2012.

[2] J. Wu, L. Ping, X. Ge, Y. Wang and J. Fu, "Cloud storage as the infrastructure of cloud computing," in *Proc. Int. Conf. on Intelligent Computing and Cognitive Informatics*, Lumpur, Malaysia, IEEE, pp. 380–383, 2010.

[3] Y. Jadeja and K. Modi, "Cloud computing-concepts, architecture and challenges," in *Proc. Int. Conf. on Computing, Electronics and Electrical Technologies*, Nagercoil, India, IEEE, pp. 877–880, 2012.

[4] M. Ahmed and A. T. Litchfield, "Taxonomy for identification of security issues in cloud computing environments," *Journal of Computer Information Systems*, vol. 58, no. 1, pp. 79–88, 2018.

[5] H. Y. Lin and W. G. Tzeng, "A secure erasure code-based cloud storage system with secure data forwarding," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 6, pp. 995–1003, 2011.

[6] L. Zhou, V. Varadharajan and M. Hitchens, "Achieving secure role-based access control on encrypted data in cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 1947–1960, 2013.

[7] C. Wang, Q. Wang, K. Ren, N. Cao and W. Lou, "Toward secure and dependable storage services in cloud computing," *IEEE Transactions on Services Computing*, vol. 52, pp. 220–232, 2011.

[8] S. Han and J. Xing, "Ensuring data storage security through a novel third party auditor scheme in cloud computing," in *Proc. IEEE Int. Conf. on Cloud Computing and Intelligence Systems*, Beijing, China, IEEE, pp. 264–268, 2011.

[9] J. D. Ferrer, O. Farras, J. R. Gonzalez and D. Sanchez, "Privacy-preserving cloud computing on sensitive data: A survey of methods, products and challenges," *Computer Communications*, vol. 140, pp. 38–60, 2019.

[10] M. Schnjakin and C. Meinel, "Scrutinizing the state of cloud storage with cloud-RAID: A secure and reliable storage above the clouds," in *Proc. Int. Conf. on Cloud Computing*, Santa Clara, CA, USA, IEEE, pp. 309–318, 2013.

[11] M. Schnjakin and C. Meinel, "Evaluation of cloud-raid: A secure and reliable storage above the clouds," in *Proc. Int. Conf. on Computer Communication and Networks*, Nassau, Bahamas, IEEE, pp. 1–9, 2013.

[12] C. Yin, C. Xie, J. Wan, C. C. Hung, J. Liu *et al.,* "BMCloud: Minimizing repair bandwidth and maintenance cost in cloud storage," *Mathematical Problems in Engineering*, vol. 2013, pp. 1–11, 2013.

[13] G. Song, S. Kim and D. Seo, "Save me: Client-side aggregation of cloud storage," *IEEE Transactions on Consumer Electronics*, vol. 61, no. 3, pp. 302–310, 2015.

[14] D. Seo, S. Kim and G. Song, "Mutual exclusion method in client-side aggregation of cloud storage," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 2, pp. 185–190, 2017.

[15] A. Bessani, M. Correia, B. Quaresma, F. Andre and P. Sousa, "DepSky: Dependable and secure storage in a cloud-of-clouds," *ACM Transactions on Storage (TOS)*, vol. 9, no. 4, pp. 1–33, 2013.

[16] R. G. Tinedo, S. M. Artigas and G. P. Lopez, "Friend box: A hybrid F2F personal storage application," in *Proc. Int. Conf. on Cloud Computing*, Honolulu, HI, USA, IEEE, pp. 131–138, 2012.

[17] B. Mao, S. Wu and H. Jiang, "Exploiting workload characteristics and service diversity to improve the availability of cloud storage systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 7, pp. 2010–2021, 2015.

[18] D. Seo, S. Kim and G. Song, "Mutual exclusion strategy in a cloud-of-clouds," in *Proc. Int. Conf. on Consumer Electronics*, Las Vegas, NV, USA, IEEE, pp. 124–125, 2017.

[19] D. Sanchez and M. Batet, "Privacy-preserving data outsourcing in the cloud via semantic data splitting," *Computer Communications*, vol. 110, pp. 187–201, 2017.

[20] N. Mhaisen and Q. M. Malluhi, "Data consistency in multi-cloud storage systems with passive servers and non-communicating clients," *IEEE Access*, vol. 8, pp. 164977–164986, 2020.

[21] L. M. Vaquero, L. R. Merino and D. Moran, "Locking the sky: A survey on IaaS cloud security," *Computing*, vol. 91, no. 1, pp. 93–118, 2011.

[22] S. Namasudra, D. Devi, S. Kadry, R. Sundarasekar and A. Shanthini, "Towards DNA based data security in the cloud computing environment," *Computer Communications*, vol. 151, no. 1, pp. 539–547, 2021.

[23] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.

[24] A. Duminuco and E. Biersack, “A practical study of regenerating codes for peer-to-peer backup systems,” in *Proc. Int. Conf. on Distributed Computing Systems*, Montreal, QC, Canada, IEEE, pp. 376–384, 2009.

[25] O. Khan, R. C. Burns, J. S. Plank, W. Pierce and C. Huang, “Rethinking erasure codes for cloud file systems: Minimizing I/O for recovery and degraded reads,” in *Proc. USENIX Conf. on File and Storage Technologies (FAST)*, San Jose, CA, vol. 1, pp. 20, 2012.

[26] K. V. Rashmi, N. B. Shah and P. V. Kumar, “Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction,” *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5227–5239, 2011.

[27] D. S. Papailiopoulos, A. G. Dimakis and V. R. Cadambe, “Repair optimal erasure codes through hadamard designs,” *IEEE Transactions on Information Theory*, vol. 59, no. 5, pp. 3021–3037, 2013.

[28] S. Kamlesh Sharma and G. Nidhi Garg, “An enhanced data storage technique on cloud computing,” *Indian Journal of Data Communication and Networking (IJDCN)*, vol. 1, no. 3, pp. 1–4, 2021.

[29] M. Tahir, M. Sardaraz, Z. Mehmood and S. Muhammad, “CryptoGA: A cryptosystem based on genetic algorithm for cloud data security,” *Cluster Computing*, vol. 24, no. 2, pp. 739–752, 2021.

[30] S. Kamara and K. Lauter, “Cryptographic cloud storage,” in *Proc. Int. Conf. on Financial Cryptography and Data Security*, Berlin, Heidelberg, Springer, pp. 136–149, 2010.

[31] J. Thakur and N. Kumar, “DES, AES and blowfish: Symmetric key cryptography algorithms simulation-based performance analysis,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 1, no. 2, pp. 6–12, 2011.

[32] A. G. Dimakis, K. Ramchandran, Y. Wu and C. Suh, “A survey on network codes for distributed storage,” *Proceedings of the IEEE*, vol. 99, no. 3, pp. 476–489, 2011.

[33] J. Spillner, J. Muller and A. Schill, “Creating optimal cloud storage systems,” *Future Generation Computer Systems*, vol. 29, no. 4, pp. 1062–1072, 2013.

[34] M. Blaum, J. Brady, J. Bruck and J. Menon, “EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures,” *IEEE Transactions on Computers*, vol. 44, no. 2, pp. 192–202, 1995.

[35] D. S. Papailiopoulos, J. Luo, A. G. Dimakis, C. Huang and J. Li, “Simple regenerating codes: Network coding for cloud storage,” in *Proc. IEEE Int. Conf. on Computer Communications (INFOCOM)*, Orlando, FL, USA, pp. 2801–2805, 2012.

[36] A. M. Ali and A. K. Farhan, “A novel improvement with an effective expansion to enhance the MD5 hash function for verification of a secure e-document,” *IEEE Access*, vol. 8, pp. 80290–80304, 2020.

[37] R. Rivest and S. Dusse, “The MD5 message-digest algorithm,” *Network Working Group*, vol. 1, pp. 21, 1992.

[38] Q. Duan, “Cloud service performance evaluation: Status, challenges, and opportunities-a survey from the system modeling perspective,” *Digital Communications and Networks*, vol. 3, no. 2, pp. 101–111, 2017.

[39] A. Appathurai, G. Manogaran and N. Chilamkurti, “Trusted FPGA-based transport traffic inject, impersonate (I 2) attacks beaconing in the internet of vehicles,” *IET Networks*, vol. 8, no. 3, pp. 169–178, 2019.

[40] A. Appathurai, R. Sundarasekar, C. Raja, E. J. Alex, A. C. Palagan *et al.,* “An efficient optimal neural network-based moving vehicle detection in traffic video surveillance system,” *Circuits Systems, and Signal Processing*, vol. 39, no. 2, pp. 734–756, 2020.