MJSAT

Malaysian Journal of Science and Advanced Technology

journal homepage: https://mjsat.com.my/

A Secure and Efficient Data Distribution System in a Multi-Cloud Environment

Ghassan Sabeeh Mahmood[1], Noor Hasan Hassoon [2], and Hazim Noman Abed*[1]

[1] *Dept. of Computer, College of Science, University of Diyala, Diyala, Iraq.*
[2] *Dept. Computer, College of Education for Pure Science, University of Diyala, Diyala, Iraq.*

KEYWORDS

Multi-cloud
Distribution Confidentiality
Integrity
Update

ARTICLE HISTORY

Received 9 August 2021
Received in revised form 27 August 2021
Accepted 30 August 2021
Available online 2 September 2021

ABSTRACT

Cloud computing is a model for providing online services. These services can be accessed on a pay-per-use basis from anywhere at any time. However, data security in the cloud has become a very big problem. A major concern about data security is the cloud service provider (CSP), which has access to data, which in turn increases user concerns and reduces cloud capacity in many areas such as medical record databases and financial data. The problem of data security mainly affects the use of cloud computing by users and organizations. The current research on data protection has been done in multi-cloud computing so that CSPs do not have direct access to the data. In this paper, a secure and efficient cloud data system is proposed in order to ensure data confidentiality and integrity by encrypting the image using DNA encoding combined with the logistical map and spatial map. The encrypted image is then divided into chunks before being stored in the multi-cloud to ensure the confidentiality of the data. The hash value is calculated using the Message Digest (MD5) algorithm to verify the integrity of the data. The proposed system also supports efficient updating of outsourced data. For the security analysis, many experiments were conducted to ensure the confidentiality and privacy of data in a multi-cloud environment, where files of different sizes were used, and the analysis showed that the proposed system is more secure. On the other hand, conducted several tests to analyse performance with dynamic data support and latency, where the analysis showed that the results are at the required level. Finally, through a comprehensive analysis of security and performance, the proposed system is proven to be more secure in a multi-cloud environment and that the system is highly efficient.

1. INTRODUCTION

Recently, cloud technology is one of the biggest advances in IT. This has become popular among organizations and individuals because of its economic features [1,2]. This technology conveniently enables sharing a set of computing resources on-demand network access, like services, servers, networks, applications, and storage [3,4]. However, organizations and individuals may regard security matters as a barrier when using cloud storage [5]. The concern can be attributed to the fear that a single cloud may not meet user requirements due to its limited capacity, and even a single cloud may fail to deliver the necessary services if disrupted. Besides, the use of a single cloud has been detected for threats to sensitive data security as well as privacy.

Storing users' files on the cloud has become very popular, but reliability and privacy worries continue. CSPs have experienced great data leaks, for example, credit card information from Home Depot as well as celebrity photos on iCloud, and interruptions Amazon's S3 service went down for an hour in 2013, which affected sites like Flipboard and Instagram. As well as in 2014 Dropbox went down for three hours [20]. By contrast, the multi-cloud may present more advantages and solve some of the security issues concerned with the use of the single cloud [6]. One of the features that

*Corresponding author:
E-mail address: Hazim Noman Abed < hazim_numan@sciences.uodiyala.edu.iq >.

need to be improved is to secure distributed data in a multi-cloud [7]. Distributed storage may involve high opportunities for malicious attacks or abuses [8], such as attacks through data transmissions. Presently, unexpected operations often happen on the side of the cloud server that is essentially restricted by laws and regulations.

Therefore, several systems [10,11,12,17,18] have been proposed to achieve data security in the multi-cloud. Yesilyurt et al. [10] proposed a system that uses the method of data anonymization in the cloud model and the integrity of the stored information to secure the data. They were able to spread the data across the multi-cloud but they could not achieve the data confidentiality efficiently nor did they achieve the data integrity. Whereas Shivanna et al. [11] proposed a dual encryption system to save data and enable access to resources on cloud platforms. They were able to achieve the confidentiality of the data, but they did not explain how they were able to achieve the integrity of the data, in addition to the fact that they did not achieve the data updates. The same applies to Bala et al. [12] introduced a biometric-inspired homogeneous encryption method for transferring data across cloud environments. They were also able to achieve the issue of data confidentiality, but they did not achieve both the integrity of the data and the data updates. Each of the proposed systems in [17,18] included the process of separating data and distributing it to different clouds as well as the data recovery, but these systems did not take into account the security of cloud data and the process of updating the data. Therefore, due to users' lack of confidence in the cloud computing environment and the problems mentioned in the above systems in terms of lack of confidentiality and integrity, In addition to supporting only static data or not achieving secure data retrieval, these factors motivated the proposal of this paper.

So, in this paper, a secure and efficient data distribution system in a multi-cloud environment is proposed to prevent cloud service providers from directly accessing original user data to achieve data confidentiality and data integrity. The rest of this paper is planned as follows. In Section 2, the related work is presented. The methodology is presented in Section 3. Results and discussion of the proposed system are presented in Section 4. Finally, the conclusion of the paper is introduced in Section 5.

2. RELATED WORK

Confidentiality and data integrity are two of the main critical security issues related to user data. This section reviews previous contributions to cloud computing security.

Mendel et al. [9] proposed a system in cloud environments where used the steganography method to inserted private data by using the pixel mapping technique, whereas they used the genetic algorithm in encryption and decryption. Vengadapurvaja et al. [13] proposed a homomorphic encryption method to encrypt medical images for cloud computing security. Leistikow and Tavangarian [14] proposed a technique that consisted of analyzing the images initially and identifying sensitive information and then separating these images into two parts (according to sensitive and non-sensitive information). Finally, distribute the sensitive data to the private cloud and non-sensitive data to the public cloud. Lee et al. [15] Relying on data splitting into the cloud for large data processing, and to increase the efficiency of data splitting, they have proposed in their system a splitting technique called SPA. [14,15] relied on pure data division techniques in their work and did not investigate data updating, data retrieval and data integration.

Y. Singh et al. in [16] used a system for making decisions through data distribution. The result of this system is to reduce the cost of the payment required to cloud service providers. However, this model did not take into account other factors, such as the CSP performance, and the calculation of data storage time. In this system, the volume of data to be separated and distributed to the cloud was based on the user rating. Consequently, data segmentation is difficult for users without using an appropriate method. Also, the absence of a data update approach in this system makes it unsuitable for real environments. K. Latha et al [19] suggested a multi-cloud scheme depending on the block data security, by naming the bytes in the data block using defined numbers (Galois field) and applying the logarithm to this defined area to mix the bytes in each block. Then they encrypted the resulting data by block-based data. Next, they distributed data nonlinearly on multiple clouds. Although the system succeeded in data confidentiality but does not support data integrity as well as data update. To address key challenges in maintaining the confidentiality, integrity, and privacy of data in a multi-cloud environment. This paper suggests a secure system compared to existing systems. Table 1 illustrates the comparison between current systems and the proposed system, this table shows the schemes can only partially achieve the plan goal of the proposed system.

Table 1. Comparison of the systems based on criteria of multi-cloud.

References	Data Splitting	Data Encryption	Data Distribution	Data Verification	Data Updating	Data Retrieval
Ateniese [7]	Yes	No	Yes	No	No	No
Yesilyurt [10]	Yes	No	Yes	No	No	No
Shivanna [11]	Yes	Yes	Yes	No	No	No
Bala [12]	Yes	Yes	Yes	No	No	Yes
El-Booz [3]	Yes	Yes	Yes	No	No	Yes
The proposed system	Yes	Yes	Yes	Yes	Yes	Yes

3. METHODOLOGY

The proposed system for securing data can be allocated into five models respectively data security, data distribution, data verification, data update, and data retrieval. These models make the system complete for the real environment. In the next subsections, each model and its function will be described in additional detail. The suggested system is illustrated in Fig. 1 and Fig. 2.

.

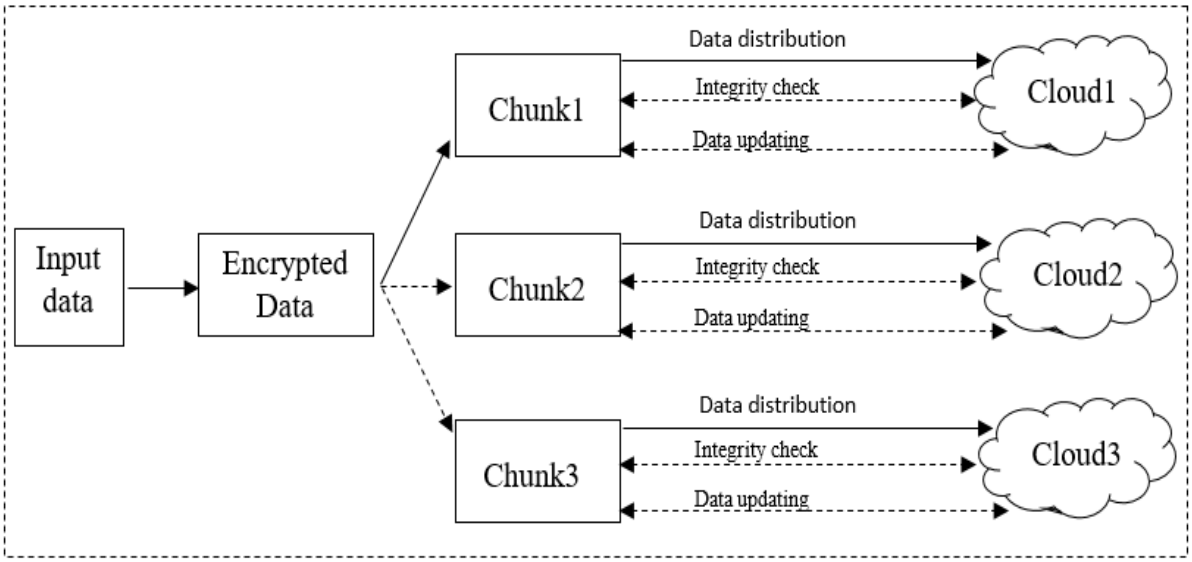


Fig. 1. Data distribution system workflow.

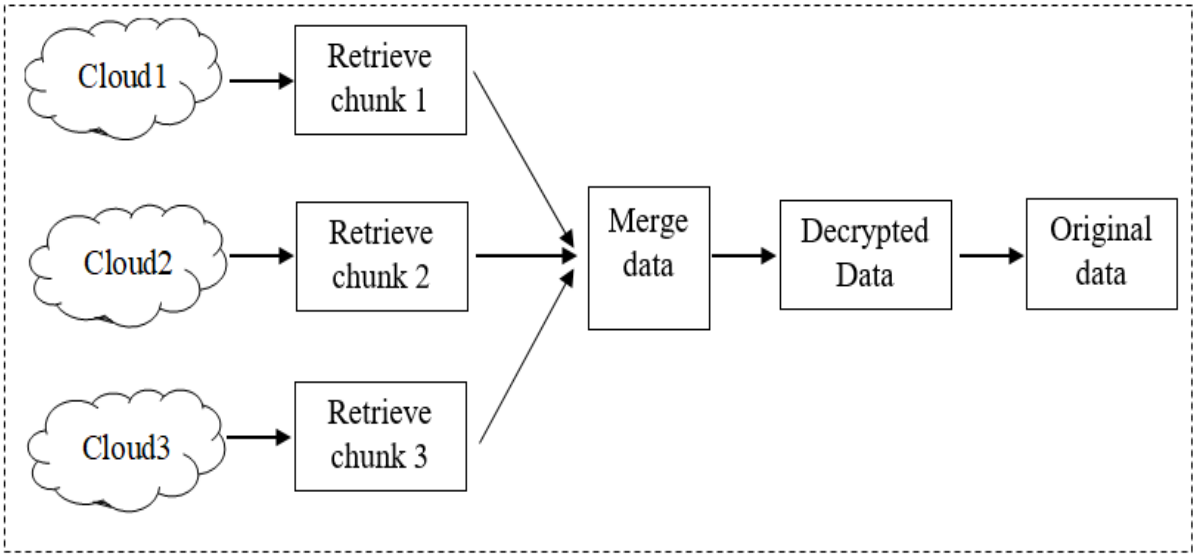


Fig. 2. Data retrieval system workflow.

3.1 Data Security Model

The first model in the proposed system is designed to improve data security and achieve data confidentiality; therefore the CSP cannot access the data directly. Ping Liu et al [21] suggested a color-coded system depending on DNA encryption joint with the logistical and spatial map that will use in the proposed model to encrypt the data before distributing the data in the multi-cloud to improve security and ensure data confidentiality. The encrypted algorithm first, performs the scrambling via the logistic map of the RGB components. After then, XOR runs between the pixel components and the sequence array organized through a map of spatial. Next, realized the addition of the RGB components by adding DNA, after encoding and performing a complementary process using a DNA sequence array organized by a map of spatial. Later, images are obtained after decoding DNA. Lastly, it obtains the encrypted images by reconfiguring the RGB components. The logistic map is highly sensitive to the values of the initial parameter, and a

small change will result from an enormous alteration, the meaning is defined through Eq. (1)

$$Yi + 1 = M * Y * (1 - Yi)\ M \in [0,4]\ Y[0,1] \quad (1)$$

Currently, DNA computing is permeating the area of cryptography. Cryptograms of DNA uses DNA as data carriers and benefit from biotechnology to realize encryption. DNA involves many kinds of deoxyribonucleic acid (A, G, C, and T). There are twenty-four encoding systems to denote the situation using 2 bits; eight of them are consistent with the supplementary bases rules set out in Table 2.

Table 2. The pairing rules.

	1	2	3	4	5	6	7	8
A	00	00	01	01	10	10	11	11
T	11	11	10	10	01	01	00	00
G	01	10	00	11	00	11	01	10
C	10	01	11	00	11	00	10	01

This algorithm used four bits of sequences of DNA to denote the values of the pixel. For instance, the value of the pixel is 231, its binary array is [1 1 1 0 0 1 1 1], the DNA sequence is [T C G T] coded by the first coding rule. On the other hand, the chaotic method is a deterministic random process in a non-linear dynamic scheme with the variables (m and n) in the chaotic mapping space. The sequence of spatial chaos of the m variable is more complicated, more difficult, and more random to predict sequences of chaotic. The differential method of the two-dim scheme of spatial generalization is set out in Eq. (2).

$$Yj+1, i + \omega Yj, i+1 = 1 - (\beta (1 + \omega) \times Yj, i)\ 2, \quad (2)$$

After the encryption process to the original image, the proposed system will divide the encrypted image into chunks based on the number of rows and naming the chunks. The user of the cloud selects a security level in the proposed system to be kept in the chunks in the cloud. The first level is denoted by the security level. The second level is denoted by a high-security level. The third level is denoted by a top-security level. For example, if the second level is selected, then the data are divided into two different chunks. In the proposed system, the size of the data fragment must not be larger than the size limit of the cloud storage file where the fragment is stored, as well as the amount of space available in the cloud. The process of this model is accomplished in the proposed algorithm called a data protection algorithm. The main parts of this algorithm are as follows:

Algorithm 1 Data Protection Algorithm

Input: the original image
Output: divided image (chunks)
1: Extract the RGB channels, and set these channels averages as logistic method initial values.
2: Create logistic sequences LS by Eq. (1).
3: Sort LS to obtain index sequence IS by Eq. (3).
$$[Sort\ StdY\ i, Flag\ i] = sort\ (Y\ i), \quad (3)$$
4: For each pixel, sort IS to confuse image pixel by Eq. (4).
$$R,G,B(scr(:,1)) = Sort\ Std\ L\ i(F\ lag\ i(:,1), :), \quad (4)$$
5: XOR between the random sequences Ch (i) and confused image sequence.
$$Ch\ (i) = f\ ix\ (mod\ (Std\ L\ i \times 1012, 256)), \quad (5)$$
6: Coding of DNA by using Table 2;
7: Get spatial Ch using (2) and then by using Eq.(6);
$$Ch = f\ ix\ (mod\ (B \times 1012, 4)), \quad (6)$$
8: Coding of Ch using Table 3;
9: Sequence Ch + DNA to get the encrypted image;
10: Split the encrypted image into multi-chunks depending on the security level (l) by the number of rows (r), naming the chunks and returns the chunks (C) by using the function: C= splitData (encrypted image,l,r).

Table 3. Coding of random sequences.

	A	G	C	T
A	A	G	C	T
G	G	C	T	A
C	C	T	A	G
T	T	A	G	C

3.2 Data Distribution Model

Data distribution is a significant process and should behave security methods for consumers through the cloud-based data distribution process, which needs additional attention. Data distribution refers to the second model in the suggested scheme and is designed to store data fragments in multiple clouds. Initially, the user creates the metadata information table. This table is stored as a dynamic table as shown in Eq.(7). Which stores user ID, which creating the data, cloud ID, the original file name, the chunk ID that will be stored in clouds, chunk ID stored in the multi-cloud, number of chunks, as well as time, and size.

Metadata= <user ID, cloud ID, file name, chunk ID, number of chunks, chunk ID in clouds, size, time> (7)

Afterwards, the chunks are stored in the corresponding multi-clouds depending on the metadata table. This process is accomplished using the data distributed algorithm. The main parts of this algorithm are as follows:

Algorithm 2 Data Distributed Algorithm

Input: The chunks to be uploaded into multi-clouds
Output: Distributed chunks, and metadata
1: For each chunk do
2: Creates the metadata table using the function: md=getMeta (chunks)
3: Stores the chunks in the corresponding multi-clouds by the function: distribute (chunk,md).
4: End for
5: Updates the information of the metadata table using the function: updateMeta (md,chunk)
6: upload the metadata table to the cloud by using the function: uploadMeta(md)

3.3 Data Integrity Verification Model

Verification is an important part of this proposed system, and specifically, if the CSP is not fully trusted, the integrity of cloud computing users' data must be protected. There should be a method that can be used to verify that the data returned from the cloud has not been modified. In this subsection, the data owner checks the data on the cloud. So that the hash value of the uploaded chunk to the cloud is calculated to preserve the data. This is done by the data owner with the help

of MD5 before uploading chunks to the cloud. In the future, the data owner can verify their data through the verification algorithm. When the owner requests verification, the hash value of the data in the cloud is calculated. The previously calculated hash value is then matched with the hash value to verify the integrity of the data. If the values match, the data in the cloud is secure and has not been tampered with or modified. The main parts of this process are accomplished using the data verification algorithm, as follows:

Algorithm 3 Data Verification Algorithm

Require: h1 and h2 {represent two hash values}
Ensure: the chunk is secure
1: Data owner calculates the hash value before uploading chunks to the cloud using MD5, using the function: h1= hash (chunk)
2: Requests data integrity verification by the data owner;
3: Calculate the current data hash value in the cloud, using the function: h2= hash (chunk)
4: The computed hash value is compared with the current hash value using the comparison function: comparison (h1,h2)
5: If the values (h1, h2) are matched, then the data present in the cloud is secure and has not been tampered.

3.4 Data updating Model to Achieve Dynamic System

In cloud computing storage, a user may want to modify data chunks kept in the clouds, from its current data to new data. This process is referred to as data modification. Besides, occasionally may need to be deleted data chunks. Deletion of data is a public process in which a user deletes data segments based on the metadata table and then updates the table. Therefore, the deletion process is a special situation of the data update process. In this model, data updating is supporting with integrity protection, including insertion, modification, and deletion. The main parts of the data updating algorithm are as follows:

Algorithm 4 Data Updating Algorithm

Input: The chunks to be updated, metadata
Output: Update the chunk and metadata
1: Execute algorithm 3 (Data Verification Algorithm) and read the metadata table; depending on this table, send a demand to the specified cloud;
2: The server of the cloud authenticates the request. If the request passes a challenge, then the results are returned; otherwise, the request is rejected;
3: Execute algorithm 1 (Data Protection Algorithm) to prepare secure chunks that can be inserted or Modified;
4: Interchange between the new chunk and the chunk saved in the cloud, and update the metadata table using the following functions: interchange (localchunk,cloudchunk), updateMeta (md,chunk)
5: Delete the chunk that needs to delete, and update the table using the following functions: delete (chunk), updateMeta (md,chunk).

3.5 Data Retrieval Model

Data retrieval is the reverse process of data distribution. During the data retrieval process, the chunks from multiple clouds are downloaded and reassembled into a complete file. After that, decrypted it to get the original image. It should be noted that decryption is the reverse encryption method. The secure retrieval algorithm is aimed to allow users to get the data by retrieval the chunks from the clouds. The main stages of the data recovery algorithm proposed for this system are explained as follows:

Algorithm 5 Data Retrieval Algorithm

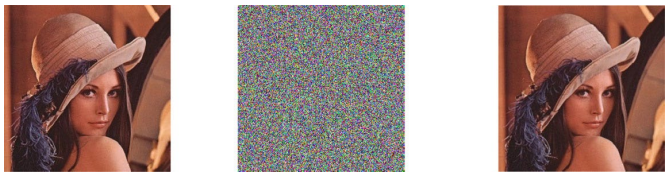
Input: Metadata
Output: The original image
1: Reads the metadata table using the function: readMeta (md)
2: Based on the metadata table, sends a demand to the specified cloud;
3: The server of the cloud authenticates the demand. If the demand passes a challenge, then the results are returned, and the download function is used otherwise, the demand is rejected: download(chunk,md)
4: The divided chunks are reconstructed to obtain the encrypted image using the function: encrypted image= mirage (chunks)
5: The data are decrypted to obtain the original image using the function: decrypt(image)

4. RESULTS AND DISCUSSION

In the proposed system, the prototype is a python-based platform on a laptop running a 2.53 GHz processor and 8 GB memory. The prototype consists of a set of basic components: The system user interface, data protection based on encryption and partitioning, selection of the number of chunks to be uploaded to the cloud-based on the required level of protection, upload and download of chunks of data to and from selected CSPs, metadata information table, and CSPs to be distributed, The experiments use public cloud storage services (Google Drive, One Drive, and Dropbox) to verify data confidentiality, integrity and support dynamic processes. The proposed system ensures that it meets user requirements regarding integrity, privacy, and latency. This section provides a recognized evaluation of the proposed system by dividing it into three parts as follows.

4.1 Guarantee of Data Privacy and Confidentiality

In the proposed system, more than one level of data security is proposed, relying first on data encryption and then the fragmentation of encrypted data into parts according to the level of security required by the user and then distribute these parts on the multi-cloud to ensure a great level of security. This requires the attacker to possess all parts of the file in addition to the algorithm used to decrypt the data Because the proposed system is based on the principle of distributing data on separate CSPs. It is unlikely that users' data has been compromised. Initially and on the client-side, performance analysis and encryption algorithm results will be represented. The original image in this experiment is shown in figure 5. Initial keys are selected for this experiment as ω=-0.05, β=1.75, A(1,1)=0.27, A(1,2)=0.83, A(2,1)=-0.45. Fig. 3 illustration the encryption and decryption outcomes.



Original image Encrypted image Decrypted image

Fig. 3. Original, encryption and decryption images.

The best encryption method must be able to resist wholly types of recognized attacks. Therefore, different security analysis was performed on the encryption algorithm, such as (secret keyspace, information entropy, correlation coefficient). Any good system in which the encryption method must be sensitive to encryption keys. Also, the size of the keys must be large enough to resist a brute force attack. Therefore, the encryption key includes the initial value of the spatial map A (1,1), A (1,2), A (2.1), β, and ω, and the precision is 10−15, so the key size is greater than 10135. Therefore, the size of the key is sufficient to resist the types of attacks. The entropy of information is an idea utilized in the theory of information to measure the information amount. The more organized a scheme is, the lesser the entropy of information. Thus, entropy is also an evaluation criterion for the level of method confusion. The entropy is represented in Eq. (8).

$$H = -\sum_{i=0}^{255} p(i) \log_2 p(i), \qquad (8)$$

where (i) represent the gray value, whereas P(i) represent the appearance probability of (i). Rendering to the equation (8), the perfect entropy=8 can be obtained. Therefore, the encrypted image entropy must be near eight. Table 4 illustrates the entropy of several systems.

Table 4. Information entropy.

References	Information entropy
Liu HJ [22]	7.9832
Liu LL [23]	7.9877
Proposed algorithm	7.9913

The image generally has a high redundancy of data, so pixels have great associations with the adjacent pixels. Therefore, the best encryption technique is necessary to be able to break these correlations. From Table 5 the correlations can be seen between the components of R, G, and B so that the outcome of the original data is closer to one, but the result of the encrypted data is closer to zero than Wang's scheme [24]. This verifies that the encrypted image has a very low correlation.

Table 5. Position correlation between RGB.

Correlation	R, G channels	R, B channels	B, G channels
Original image	0.9361	0.8067	0.9581
Encrypted image	0.0022	-0.0084	-0.0080
Wang [24]	-0.003803	-0.050968	0.012267

After data is encrypted it will be split into a set of chunks according to the security level. These chunks are then uploaded to multi-clouds. Their performance is presented in Fig. 4. The results of experiments in this figure demonstrating that when the chunks number increase, the time it takes to split the file increases substantially with the number of the chunk. Therefore, Split data into large chunks lead to degraded performance, causing the system to become impractical in practice. This result led to a three-chunk security level determination in the proposed system so that the highest level of protection from three chunks can be guaranteed at acceptable performance rates. This also applies to reintegrate distributed chunks after they are recovered from multi-cloud . Cloud service providers handle cloud users based on the amount of data uploaded to the cloud or downloaded as well as stored in the cloud. Table 6 shows the proposed system uploads and downloads, taking into account three data unit sizes: 100 KB, 200 KB, and 300 KB. This table includes the execution time for the operations that are performed (upload and download).

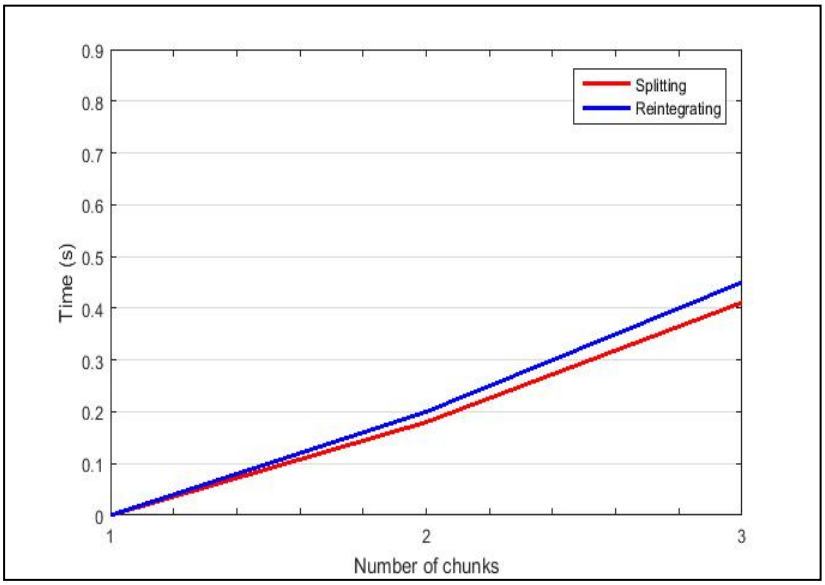


Fig. 4. Performance of data splitting, reintegrating.

All required estimates were computed depending on the values required via the three clouds with this proposed system. Uploading and downloading operations are performed on a cloud that requires less execution time. Most time drawing will be used to the flexibility that makes the proposed system workable in practice. Based on the results presented in Table 6, the proposed system for working with multiple fasteners provides a good time for implementation in terms of data uploads as well as data downloads as compared to the other three clouds (Google Drive, One Drive, and Dropbox). This can be seen in the case of three chunks, which provide maximum data protection, this makes the suggested scheme efficient and flexible so that the time difference is very small compared to the other three clouds for uploading and downloading.

Table 6. Execution time of upload and download for all clouds.

Operations	Size KB	The proposed system			Google Drive	One Drive	Dropbox
		One chunk	Two chunks	Three chunks			
Upload	100	0.31	0.49	0.72	0.31	0.42	0.45
	200	0.64	0.87	1.4	0.64	0.71	0.72
	300	0.89	1.31	1.98	0.89	0.92	0.95
Download	100	0.41	0.62	0.89	0.41	0.51	0.53
	200	0.82	0.98	1.62	0.82	0.87	0.89
	300	1.23	1.71	2.31	1.23	1.34	1.38

On the other hand, the experience can show a relative discrepancy in the performance and effectiveness of CSPs when accessed from various regions. This means no cloud service provider can cover different regions in the same manner. In addition to one more feature of the proposed system, is that the uploading and downloading of the required data can be adapted to utilize the finest cloud for a given position.

4.2 Data integrity guarantee

This is one of the most important parts of the proposed system to ensure the integrity of user data. In this subsection, the communication burden resulting from the data integrity model in the proposed system is two transmissions, the first one for the challenge and the second for its reply. Also, because the proposed system performs verification without downloading data. Therefore, the shortest query and response is illustrated and therefore the complexity of communication in the proposed system is asymmetric constant O (1), this situation proves the efficiency of the proposed integrity model and reduces latency.

Further, the experience shows that the cost of communication due to the distribution of divided data for multi-cloud and retrieval of that data is relatively acceptable with another cloud. When the data is divided into three parts depending on the security level proposed in the proposed system, the time taken to load the three parts of the multi-cloud is (0.72) and the time to download those parts is (0.89) when the data size is (100) KB. Whereas the time it takes to upload (100) KB to (Google Drive, One Drive, Dropbox) are (0.31, 0.42, 0.45) respectively, and the time to download this data from (Google Drive, One Drive, Dropbox) are (0.41, 0.51, 0.53) respectively. This result for the distribution of divided data gives the efficiency of the proposed system compared to the other clouds.

the MD5 algorithm is adopted which allows the data owner at any time to perform data validation. So that the hash value of the data is matched. If these values match, the data is secure. On the other hand, when a user stores data on more than one cloud, the cloud server will likely attack the data. However, in the proposed scheme, the hash value is calculated for each chunk and as shown, they are unique for each chunk of each file. Therefore, the proposed system ensures the integrity of cloud data.

4.3 Efficient system by dynamic support and latency

A good system must be characterized by efficiency; The efficiency of the proposed system shows the integrity of user data with less value of computation and communication. Besides the system selects the number of pieces that should be scattered, as well as the number of CSPs to reduce latency and the system supports updating operations, including insertion, modification, and deletion of outsourced data.

1) Reduce latency to prove the efficiency

The data integrity verification cost versus the number of data chunks are computed in Table 7. As shown in Table 7, data chunks are taken for 100 to 300 KB. For (100 KB) of data chunk the owner requires around (0.122) seconds and the server requires around (0.131) seconds. The cost of the data integrity consists of the time the hash values are created and verified. The time to create hash values is simply to generate hash values for data chunks. The computation of each of the terms O (1) takes time and the expression consists of the product of many chunks that have been challenged. Therefore, the time taken in the computation will be O (n), so that n represents the number of chunks.

Table 7. The computation cost of data integrity.

Operations	Size (KB)	Integrity check (s)
On client	100	0.122
	200	0.157
	300	0.190
On server	100	0.131
	200	0.163
	300	0.196

2) Data updating efficiency

Data chunks are updated in multi-clouds in the proposed system to make it more efficient and acceptable in terms of performance and functionality. The empirical results in Fig. 5 show that the suggested scheme only lets the owner alter the data so data validation in the process of updating is necessary and helps to maintain time in the event of manipulation of data. In addition to preserving original data and detecting manipulation before updating this data. Most applications do not include frequent data updates. Thus, it is unlikely that the data owner becomes the performance bottleneck.

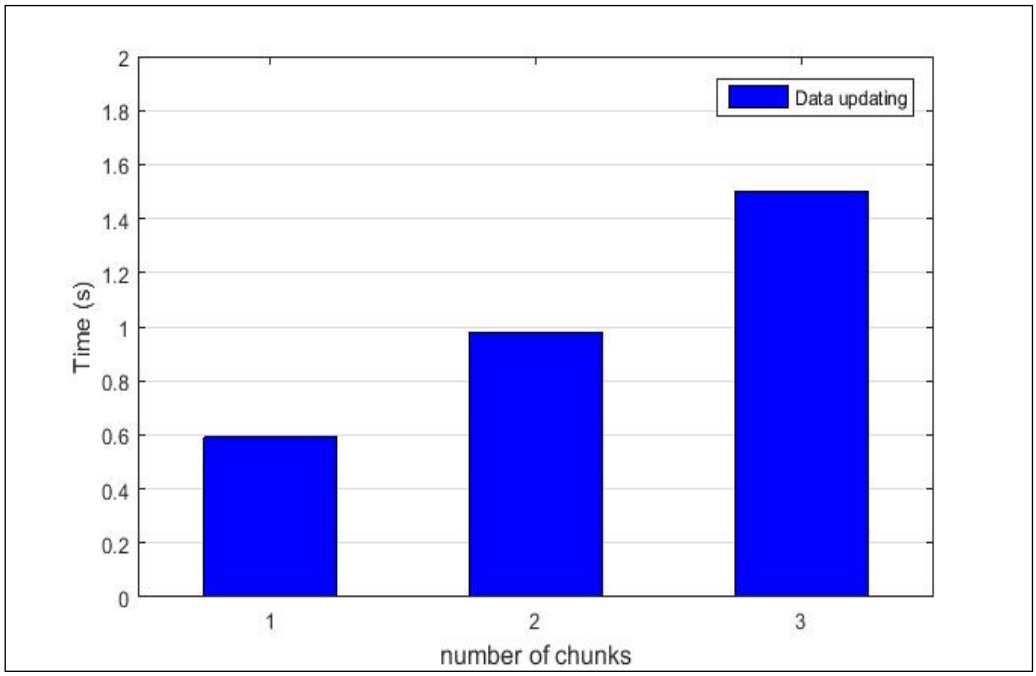


Fig. 5. Performance of data updating.

5. CONCLUSION

The proposed system is based on multi-cloud for users to meet the basic requirements of privacy, confidentiality, data integrity, and optimum efficiency. The proposed system addresses many key challenges, including the distribution of data over the multi-cloud, selecting the number of parts of data to be distributed to meet the level of protection, choosing a better cloud service provider, as well as challenges in improving privacy and integrity. The multi-cloud system distributes data into the multi-cloud, guaranteeing privacy by separating data therefore that no cloud service provider can rebuild distributed data. Furthermore, the proposed system improves data integrity to reduce latency in data integrity verification. The proposed system allows data updates for cloud users to increase system efficiency in the real environment. Future work will cover the availability of data because any data center failure will result data retrieval failure.

REFERENCES

[1] Mahmood, G. S., Huang, D. J., & Jaleel, B. A. (2019). A secure cloud computing system by using encryption and access control model. Journal of Information Processing Systems, 15(3), 538-549.

[2] Gokilavani, N., & Bharathi, B. (2021). Multi-Objective based test case selection and prioritization for distributed cloud environment. Microprocessors and Microsystems, 82, 103964.

[3] El-Booz, S. A., Attiya, G., & El-Fishawy, N. (2016). A secure cloud storage system combining time-based one-time password and automatic blocker protocol. EURASIP Journal on Information Security, 2016(1), 1-13.

[4] George, S. S., & Pramila, R. S. (2021). A review of different techniques in cloud computing. Materials Today: Proceedings.

[5] Shen, J., Yang, H., Vijayakumar, P., & Kumar, N. (2021). A Privacy-Preserving and Untraceable Group Data Sharing Scheme in Cloud Computing. IEEE Transactions on Dependable and Secure Computing.

[6] Mohammed, M. H. (2021). Bio-inspired approach and integrity check mechanism for secure data storage in multi-cloud environment. J.of Ambient Intelligence and Humanized Computing, 1-9.

[7] Ateniese, G., Fu, K., Green, M., & Hohenberger, S. (2006). Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Transactions on Information and System Security (TISSEC), 9(1), 1-30.

[8] Gai, K., Qiu, M., Tao, L., & Zhu, Y. (2016). Intrusion detection techniques for mobile cloud computing in heterogeneous 5G. Security and communication networks, 9(16), 3049-3058.

[9] Mandal, S., & Bhattacharyya, S. (2015, October). Secret data sharing in cloud environment using steganography and encryption using GA. In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT) (pp. 1469-1474). IEEE.

[10] Yesilyurt, M., & Yalman, Y. (2016). New approach for ensuring cloud computing security: using data hiding methods. Sādhanā, 41(11), 1289-1298.

[11] Shivanna, K., Deva, S. P., & Santoshkumar, M. (2017). Privacy preservation in cloud computing with double encryption method. In Computer Communication, Networking and Internet Security (pp. 125-133). Springer, Singapore.

[12] Bala, Y., & Malik, A. (2018). Biometric inspired homomorphic encryption algorithm for secured cloud computing. In Nature inspired computing (pp. 13-21). Springer, Singapore.

[13] Vengadapurvaja, A. M., Nisha, G., Aarthy, R., & Sasikaladevi, N. (2017). An efficient homomorphic medical image encryption algorithm for cloud storage security. Procedia computer science, 115, 643-650.

[14] Leistikow, R., & Tavangarian, D. (2013, March). Secure picture data partitioning for cloud computing services. In 2013 27th International Conference on Advanced Information Networking and Applications Workshops (pp. 668-671). IEEE.

[15] Lee, K., Liu, L., Tang, Y., Zhang, Q., & Zhou, Y. (2013, June). Efficient and customizable data partitioning framework for distributed big RDF data processing in the cloud. In 2013 IEEE Sixth International Conference on Cloud Computing (pp. 327-334). IEEE.

[16] Singh, Y., Kandah, F., & Zhang, W. (2011, April). A secured cost-effective multi-cloud storage in cloud computing. In 2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 619-624). IEEE.

[17] Oliveira, P. F., Lima, L., Vinhoza, T. T., Barros, J., & Médard, M. (2012). Coding for trusted storage in untrusted networks. IEEE Transactions on Information Forensics and Security, 7(6), 1890-1899.

[18] Balasaraswathi, V. R., & Manikandan, S. (2014, May). Enhanced security for multi-cloud storage using cryptographic data splitting with dynamic approach. In 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies (pp. 1190-1194).

[19] Latha, K., & Sheela, T. (2019). Block based data security and data distribution on multi cloud environment. Journal of Ambient Intelligence and Humanized Computing, 1-7.

[20] Chung, J. Y., Joe-Wong, C., Ha, S., Hong, J. W. K., & Chiang, M. (2015, April). CYRUS: Towards client-defined cloud storage. In Proceedings of the Tenth European Conference on Computer Systems (pp. 1-16).

[21] Liu, P., Zhang, T., & Li, X. (2019). A new color image encryption algorithm based on DNA and spatial chaotic map. Multimedia Tools and Applications, 78(11), 14823-14835.

[22] Liu, H., & Wang, X. (2010). Color image encryption based on one-time keys and robust chaotic maps. Computers & Mathematics with Applications, 59(10), 3320-3327.

[23] Liu, L., Zhang, Q., & Wei, X. (2012). A RGB image encryption algorithm based on DNA encoding and chaos map. Computers & Electrical Engineering, 38(5), 1240-1248.

[24] Wang, X., Liu, L., & Zhang, Y. (2015). A novel chaotic block image encryption algorithm based on dynamic random growth technique. Optics and Lasers in.