Global Scientific JOURNALS

GSJ: Volume 9, Issue 1, January 2021, Online: ISSN 2320-9186
www.globalscientificjournal.com

IMPROVEMENTS TO THE CONFIDENTIALITY AND INTEGRITY OF DATA STORED IN CLOUD STORAGE

Dr. Amjad Farooq, Abdul Rehman

Dr. Amjad Farooq
Working as Associate Professor in University of Engineering & Technology Lahore, Pakistan, PH-0092-03004174386. E-mail: amjadfarooq@uet.edu.pk

Abdul Rehman
Student in Computer Science and Engineering Department University of Engineering & Technology Lahore, Pakistan, PH-0092-03214507593. E-mail: mscsstudent417@gmail.com

KeyWords

Access Control List, Network Commited Storage, Elastic Block Storage, Third Party Auditor, Network Commited Storage, Message Digest, Secure Hash Algorithm, Data Encryption Standards, Alpha Numeric Encryption, Information Life Cycle.

ABSTRACT

Cloud Computing is an emerging technology, which is adopted by most of the organizations and IT firms. Cloud computing permits organizations to gain access to the computing resources without bearing the pain of capital investment. It means “Pay for What You Use”. The cost of having large amount of data on local storage is a headache for most of the organizations as it needs huge capital investment. Cloud environment is considered untrusted as it is accessible through internet. Therefore, users have security and integrity concerns about the data stored in cloud storage. To overcome these hurdles a method is presented, which encourages to develop application to store and retrieve data to the cloud, furthermore, it also uses cryptographic and hashing algorithms to enhance the security and integrity of data stored in cloud storage. Major of the research conducted in this area lacks confidentiality and integrity mechanism at application level. The purpose of this research is to propose a new method for storing data in to the cloud storage through application and testing the efficiency of the proposed method.

1. Introduction

Cloud computing is a recent technology that uses the Internet, central servers to organize the data and applications, which the user can access. Cloud computing allows individual users and other business peoples to use application without the necessity to install in their computer. They can access their files, which is located in other computer using Internet. This technology allows for more inefficient computing by centralizing storage, processing memory, and bandwidth. Cloud computing comes in three categories such as Software as a Service (SaaS), Infrastructure as a service (IaaS), Platform as a Service (PaaS). The SaaS provides application software which the user can use. The Paas provides the platform for the user to do his operation. The Iaas provide physical or virtual devices for user. As cloud computing is popu-lar and in demand similarly cloud storage technology has greater demand. Cloud storage is a virtualized storage area over a network basis. It provides services on the basis of QoS assured. Cloud storage consist of many resources but yet act as single system. It has greater fault tolerance by redundancy. As the data generated by IT sectors are dramatically growing one can't just update the hardware frequently instead cloud storage is adopted which is a better choice. We can use cloud storage for different purpose just backing up our home desktopdata into cloud storage or as an archive to maintain data for regulatory. Cloud storage allows user to access broad range of application and re-sources immediately, which are hosted by others. Fig.1 shows a simple cloud storage architecture.

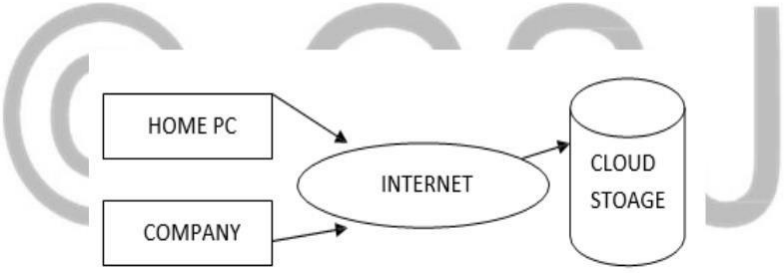

Fig. 1 Simple Cloud Storage [1]

Cloud storage architectures are mainly about delivering storage on demand in a highly scalable and multi-tenant way. Basically, cloud storage architectures contain of a front end that exports an API to communicate with the backend storage. traditional storage systems, this API is the SCSI protocol; nonetheless in the cloud, these are evolving protocols. At this layer, there are web service, filebased Internet SCSI or iSCSI front ends. This layer is the first communication point between the user and the service provider. Users access the services using their credentials.

The midpoint component layer is called storage controller that interconnects and communicates from the front API to the backend storages. This layer has a variety of features such as replication, traditional data placement algorithms with geographical location. Finally, the back-end consists of physical storage for data. This may be a central protocol that runs dedicated programs or a traditional back-end to the physical disks. Fig 2 shows the relationship between the aforementioned layers.

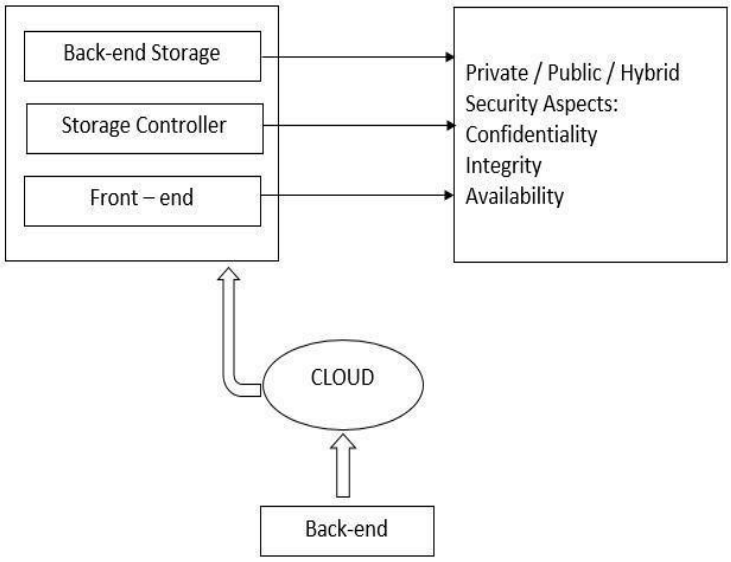

Fig. 2 Generic Cloud Storage Architecture [2]

Security is the protection of information assets through the use of technology, processes, and training. Cloud storage is a service that includes inherent vulnerabilities, but these have never discouraged users from taking advantage of its economies and flexibilities. With adoption of a cloud model, users lose control over physical security. Users raised concerns whether their data are accessed by unauthorized person since there are many user sharing the resources over the cloud.

Sharing the cloud with other users possesses risks and concerns over security. Security overall covers mainly three aspects: Confidentiality, Integrity and Availability (CIA). These aspects are the topmost considerations in designing a security measure to ensure maximum protection. Fig 3 shows the relationship between security aspects and security challenges.

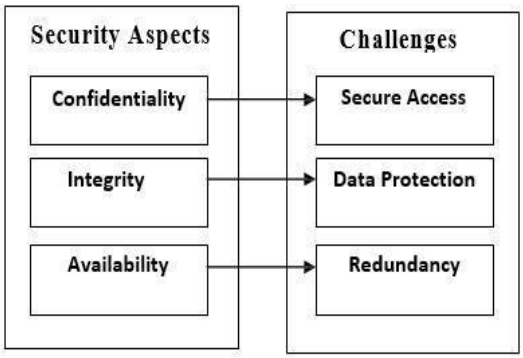

Fig.3 Cloud Storage Security Aspects and Challenges [3]

In this research a methodology is proposed to enhance the confidentiality and integrity of the data stored in cloud storage. The proposed method aims to handle these issues at application level, thus preventing unauthorized users from accessing information.

2. LITERATURE SURVEY

In 2018 D. Hyseni and A. Luma proposed a method for storing data in cloud storage. Proposed model deals with different scenarios depending on the data sensitivity. Model offers reliability of data by applying security mechanism at application level. Method involves usage of different cryptographic algorithms to enhance the security of data. Security depends on file encryption and file partitioning. To enhance the security, the proposed model offers three encryption strategies named as symmetric schema, asymmetric schema and hybrid schema. For measurement three symmetric cryptographic algorithms DES, TripleDES and AES, asymmetric algorithms RSA, Deffi-Hellmen, AlGamal were used. Hybrid Schema is the mixture of symmetric and asymmetric algorithms. There could be two approaches to achieve security: partition then encrypt or encrypt then partition, author have preferred partitioned then encrypt. A file is partitioned into fixed sized blocks after that encryption is performed for each block individually by using either symmetric, asymmetric or hybrid schema. These file blocks are then uploaded to the cloud and their mapping information is stored in local cloud. From proposed encryption strategies symmetric encryption is proved to be more efficient. The method aims to enhance the confidentiality of the data. [4]

Vijayalakshmi and N. Veeraragavan focuses on the confidentiality of data by proposing a secure unified model for data confidentiality. The systems consider the cloud storage untrusted as to be owned by the third party. The communication between cloud service provider and the client is bound to application which is responsible for providing the confidentiality from untrusted user and also from the third party cloud service provider. By hiding the details of file storage (location, cryptographic algorithm used etc.) the application makes the underly- ing storage details vague to the user. The approach classifies data into two groups numeric data and alphanumeric data. The heart of the approach is encryption and obfuscation function. If data is alphanumeric then AES encryption is applied followed by obfuscation function. If the data is numeric then only obfuscation function is applied. The results are referred to as encrypted an obfuscated data which is then stored in cloud storage. For retrieval same sequence of operations are applied in reverse order. The resulting model has benefits over the previous work as client has no control over the data as all the storage details are hidden. Furthermore, the proposed model uses AES en- cryption algorithm. [5]

Geethamani and Ranjani proposed a method which consists of five modules namely registration, upload file, admin, download and key generation. Registration module use to register a particular user. Upload file module is used to upload file and its relevant meta data in encrypted form. Admin module is responsible for managing application level issues such as resolving password recovery, key distribution, maintaining and sending auditing information etc. Download module is used to download file and its meta data to verify its integrity. Key

generation module is responsible for generating the key. The working of the system starts with the user registration, which is preceded by the key generation to make user able to save files in cloud storage. User can request key, password recovery and auditing information through admin module. Upload module is responsible for uploading file and its meta data to the cloud storage and archive storage (third party dedicated storage service). Download modules downloads a file and compare its meta data if it is correct file is delivered to the user. In case of modifications the file is retrieved from the archive (dedicated third party storage). [6]

A Valerian and C. Nadunagyu delivered an approach to enhance the confidentiality of cloud data storage. The methodology focuses on the development of application following a strategy to store data in cloud. The resulting algorithm starts by taking a salt value from the user. The target file is encrypted using the salt-key mixture. Where the key is generated randomly for the encryption algorithms. By generating separate key-salt pair for each file randomly increases confidentiality of user data. The final cipher text is further reversed to further en- hance the security of the data. Author prefers AES encryption schemes for proposed systems due to is strength and efficiency as reported by the author to be more than the other symmetric and asymmetric cryptographic schemes. [7]

R. Kulkarni and V. Waghmare proposed an entirely different technique for securing data over the cloud using behaviour profile and decoy technology. By monitoring abnormal access pattern to the cloud. In this system whenever a user tries to get a legitimate users data a decoy file viewer is created that looks the similar like the original file viewer. The system works by presenting a user with application interface. Whenever an intruder tries to reach a user's account and tries to login to the system by using hit and trial username-password combination a decoy view is generated which looks similar to the users account. For every attempt there exists a decoy view. This technique not only keeps the intruders busy but also makes it difficult to identify the original user's account. [8]

In 2017 E. Agrawal and P. Ram proposed an encryption algorithm to enhance the security of the data stored in cloud. The technique is symmetric substitution which uses a random number. The proposed algorithm starts by reading the input text. The key value is appended in front of the file. After finding the ASCII code for each text character it is converted into its corresponding binary value which is then com- plemented. The complemented binary is then converted into its corresponding decimal number. The decimal value is then divided by 4 and after finding the equivalent ASCII code for quotient. Finally, quotient is merged with remainder to get the cipher text. Finally cipher text is stored in cloud.

The decryption of the proposed algorithm is slightly different than the encryption process. First the algorithm takes the cipher text and

splits the twodigit cipher text into single - single digit. After multiplying first digit with 4 and adding the second digit into the multiplication result. After performing complement operation on the results the decimal value is obtained, which is then converted into its corresponding ASCII value. Added key value is removed to get the plaintext. [9]

In 2016 S. Ksasy and E. Takieldeen works to enhance the security of the data by offering cryptographic algorithm known as cryptobin. The algorithm starts by taking a byte from the message. After taking the input from user for key generation the following equation is applied to get the key

(Any number) MOD 8 ..... (I)

Equation (I) generates a number from 0 to 7. The output of the equation will be used to allocate which bit of each byte the system will swap (1 to 0 and vice versa). Another number is used to determine the interval length. Message encryption starts by dividing each byte into two halves as shown in Fig 4. The first half of each byte remains unchanged whereas the NOT of second part is taken. The encryption is performed by merging these two parts again. The decryption is the reverse of the encryption process.

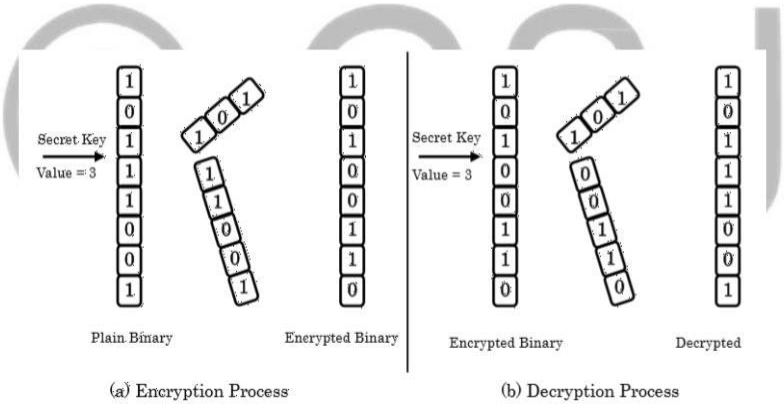

Fig. 4 Advanced Cryptographic System for Binary Codes [10]

M. Naik and P. Tungare proposed color cryptography using substitution method. By keeping in view the weaknesses of symmetric and asymmetric cryptography author have proposed a new technique to encrypt data which is based on colour schemes. The system is based on symmetric encryption which is based on encrypting text into colour images. Each character of the message is encrypted into a colour block. The user enters a message which is the actual plaintext. The channel is used from three colours red, green and blue(RGB). All the characters are converted into colour blocks. Each character present in the plaintext is replaced with one decillions colours in the world. This technique prevents from birthday paradox, meet-in-the-middle and brute force attack. Fig 5 shows the working of colour cryptography.

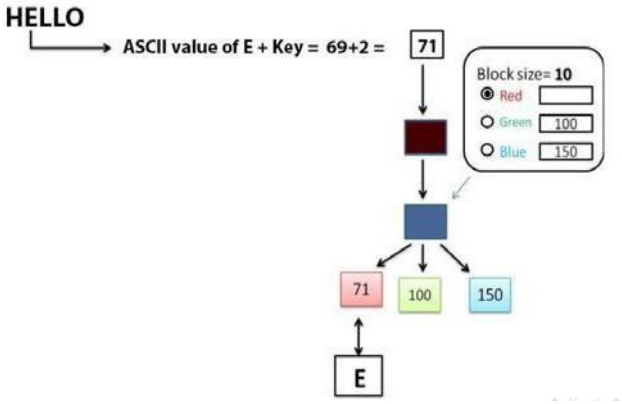

Fig. 5 Working of Colour Cryptography [11]

3. PROPOSED METHODOLOGY

In this research an application level methodology is proposed to enhance the confidentility and integrity of data stored in cloud storage furthermore, an auditing mechanism is presented. In the proposed approach following steps are carried out to improve the confidentiality and integrity of data.

3.1 HIGH LEVEL WOR FLOW

- Administrator logins to the drop box account through application to upload download file (publicly share data).
- After authentication he uploads or remove files through cloud storage.
- User or consumer can only acccess to the files uploaded in cloud storage through application.
- Access to the cloud storage is restricted through application which completely hides the file information (where it is placed and how it is stored). Fig.6 represents high level work flow.

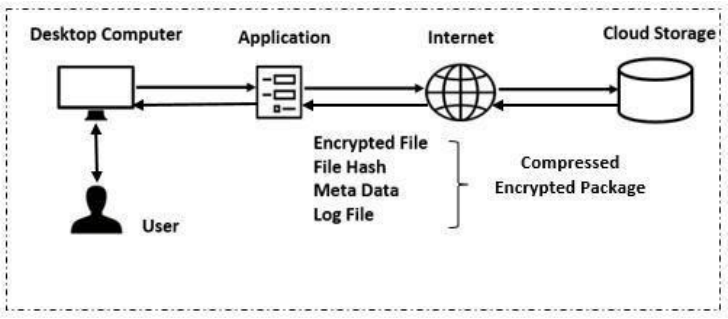

Fig. 6 High Level Flow of Proposed Methodology

3.2 STORING FILE, FILE META-DATA and HASH OF FILE

- Against each data file say F a log file LF is generate. The purpose of log file is to keep track of the users who have accessed data, which is used later for auditing purpose.
- The compression and encryption is performed on data file F using 7zip compression and AES-128 to reduce the size of the data file and to put the data into a form that cannot be easily understand.
- The hash of the data file F is generated using SHA-256 to later validate the integrity of the data.
- A meta- file is generated which stores:

- Data File Name (to hide the data file name).
- File Size (used later to check the file originality or any modification).
- Last Modified Date (to catch any unauthorized modification).

- Meta-file is compressed and encrypted to reduce the size of its contents and for data unpredictability.
- Using SHA-256 hash of the meta-file is generated which is later used to check the integrity.
- In the same way the compressed encrypted version of log file LF is generated using 7zip and AES-128.
- Hash of the log file is generated using SHA-256.
- The corresponding six files generated against a data file are uploaded to the cloud storage in *compressed encrypted package* form.

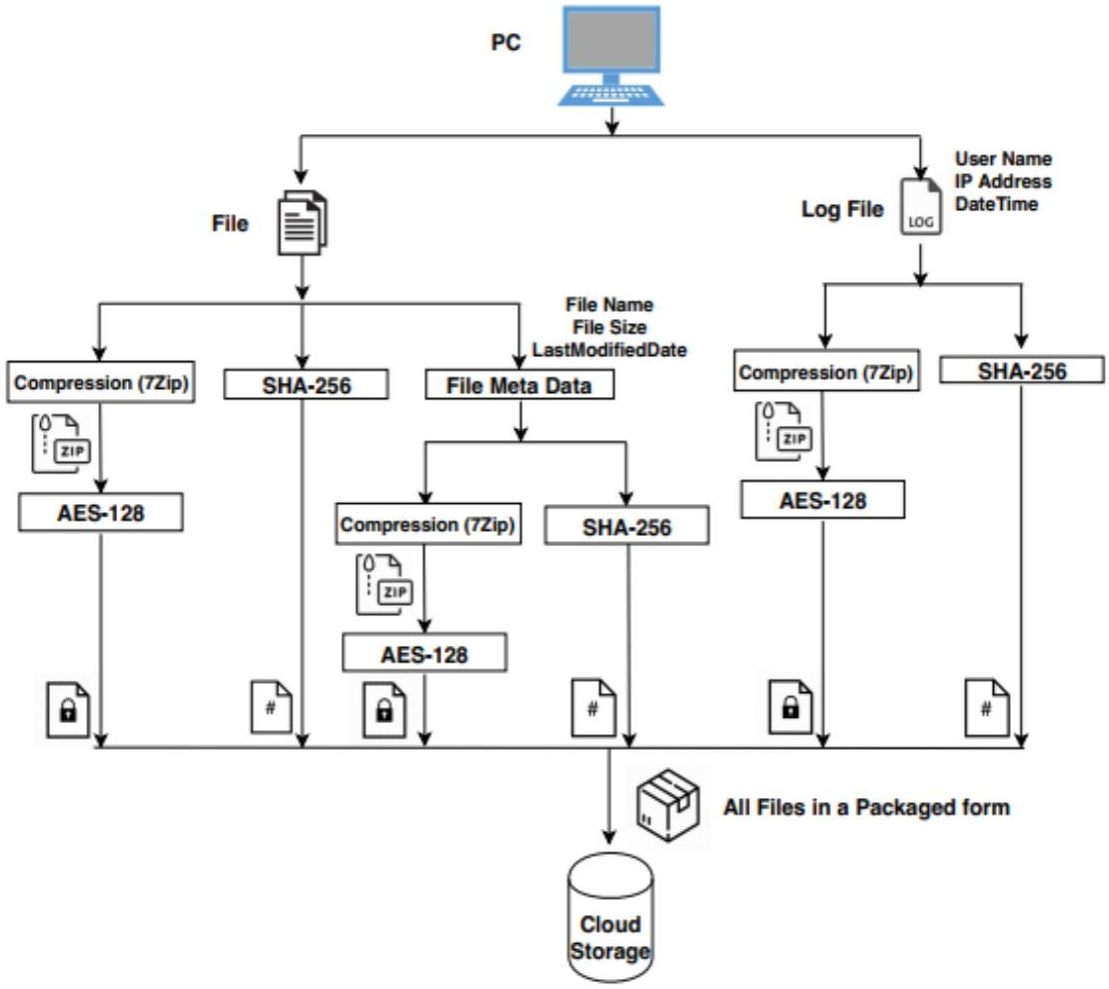

Fig. 7 File Uploading Via Proposed Approach

3.3 RETRIEVING FILE, FILE META-DATA AND HASH OF FILE

1. Download the file package from cloud storage.
2. Decrypt and Un-compress data file, meta-file and log file.
3. After generating hash of meta-file using SHA-256 compare it with the stored hash of meta file.
4. Generate the hash of data file F and compare it with the stored hash.

5. Compare data file with meta-file contents (File Name, File Size, Last Modified Date).
6. Generate the hash of log file and compare it with the stored hash.
7. If steps 1-7 succeeds file integrity is preserved.
8. In case of failure at any step send a request to the admin for file and stop.

Fig. 8 shows file retrieval and its consumption.

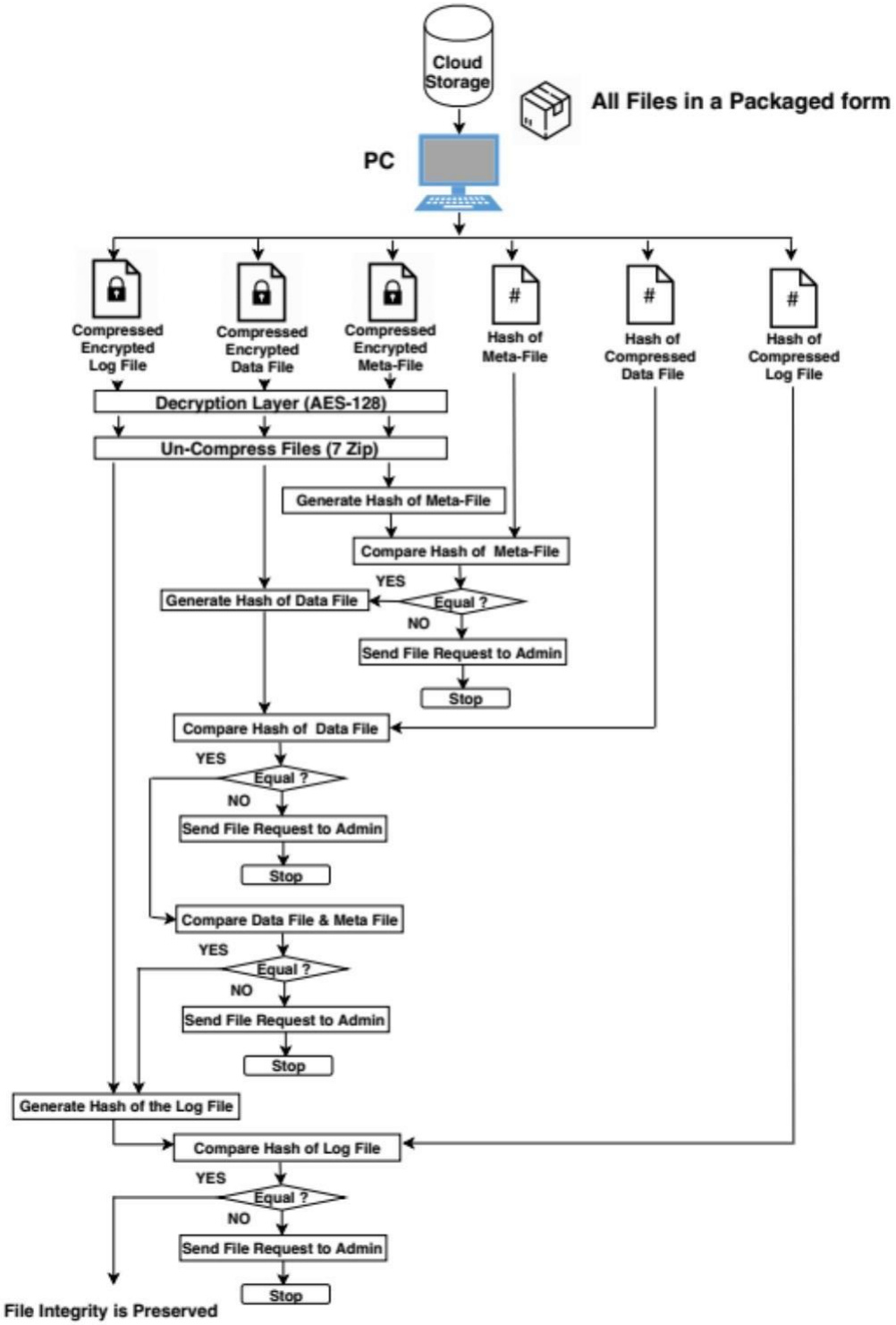

Fig. 8 File Downloading Via Proposed Approach

4. RESULTS AND CONCLUSIONS

4.1 RESULTS

The fisrt step was to check the optimization in terms of space that we have achieved through compression. Although there are six files generated against a single file in proposed approach but the size of these six files is less than the 20 % of the original file size. Table. 1 contains the results.

File Size	Compressed Encrypted Data File size	Compressed Encrypted Meta File Size	Compressed Encrypted Log File Size	Data File Hash Size	Meta File Hash Size	Log File Hash Size	Compressed Encrypted Package Size
1 MB	1.12 KB	88 Bytes	88 Bytes	8 Bytes	8 Bytes	8 Bytes	1.32 KB
2 MB	1.33 KB	87 Bytes	89 Bytes	8 Bytes	8 Bytes	8 Bytes	1.53 KB
3 MB	2.59 KB	85 Bytes	91 Bytes	8 Bytes	8 Bytes	8 Bytes	2.79 KB
4 MB	3.19 KB	90 Bytes	92 Bytes	8 Bytes	8 Bytes	8 Bytes	3.39 KB
5 MB	3.75 KB	95 Bytes	93 Bytes	8 Bytes	8 Bytes	8 Bytes	3.96 KB

Table. 1 Comparison of Original File Size to the Compressed Encrypted File Package

Fig. 9 shows the original file with its corresponding compressed encrypted package size. It can be clearly seen that compressed encrypted package size grows very slowly as compared to the original file size.

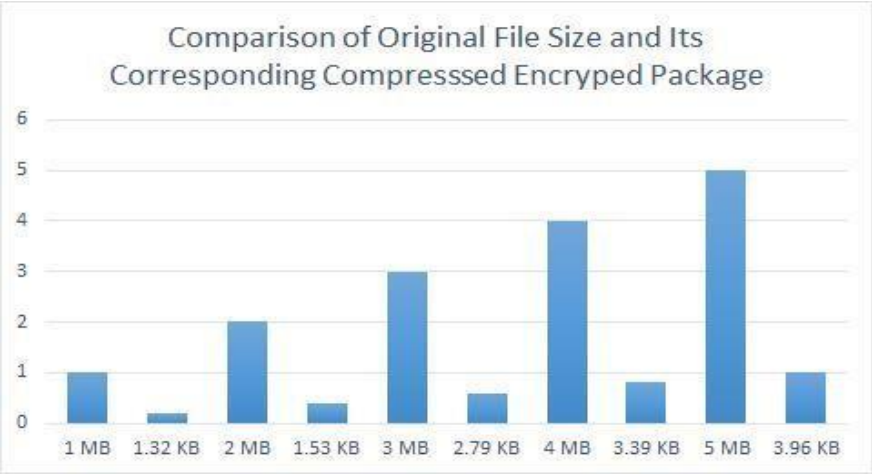

Fig. 9 Comparison of the Original File Size with its Corresponding Compressed Encrypted Package Size.

The second step was to find the running time of the proposed approach was experimented. Table. 2 shows the results of the experiment with different file sizes. After a keen analysis the running time of the proposed approach was found to be ≈ *Log (n)* where n is file size in bytes.

File Size	Time to Compress (Milliseconds)			Time To Encrypt (Milliseconds)			Time To Generate Hash (Milliseconds)			Time to Upload Package (Millis)	Total Execu-tion Time (Millis)
	7 Zip			AES			SHA-256				
	Data File	Meta File	Log File	Data File	Meta File	Log File	Data File	Meta File	Log File		
1 MB	1.21	0.10	0.12	1.25	0.13	0.13	1.22	0.15	0.16	1.10	5.57
2 MB	1.25	0.15	0.19	1.26	0.15	0.16	1.24	0.12	0.19	1.22	5.93
3 MB	1.27	0.17	0.20	1.28	0.18	0.19	1.28	0.15	0.21	1.25	6.18
4 MB	1.31	0.19	0.22	1.30	0.19	0.21	1.32	0.18	0.25	1.28	6.45
5 MB	1.41	0.22	0.25	1.32	0.22	0.25	1.35	0.22	0.28	1.31	6.83

Table. 2 Application Running Time for Different File Sizes

Fig. 10 is the graphical representation of application running time for different file sizes.

Fig. 10 Performance Analysis Application Running Time for Different File Sizes

In third step proposed approach was experimented to compare the uploading time of original file with its compressed encrypted package through application. The results were gain in performance through application which is clear in Table. 3 and Fig 11-12.

Original File	Uploading Time Through Cloud Interface (Milliseconds)	Downloading Time Through Cloud Interface (Milliseconds)	Corresponding Compressed Encrypted Package Size	Time To Upload File Through Application (Milliseconds)	Time to Download File Through Application (Milliseconds)
1 MB	11.56	9.91	1.32 KB	5.57	4.39
2 MB	13.62	10.02	1.53 KB	5.93	4.95
3 MB	15.01	11.23	2.79 KB	6.18	5.20
4 MB	17.52	11.25	3.39 KB	6.45	5.59
5 MB	20.22	13.30	3.96 KB	6.83	5.92

Table. 3 Comparison of Uploading and Downloading Time of Original File and Compressed Encrypted Package

Fig. 11 and 12, is the graphical representation of gain in performance of file uploading and downloading time.

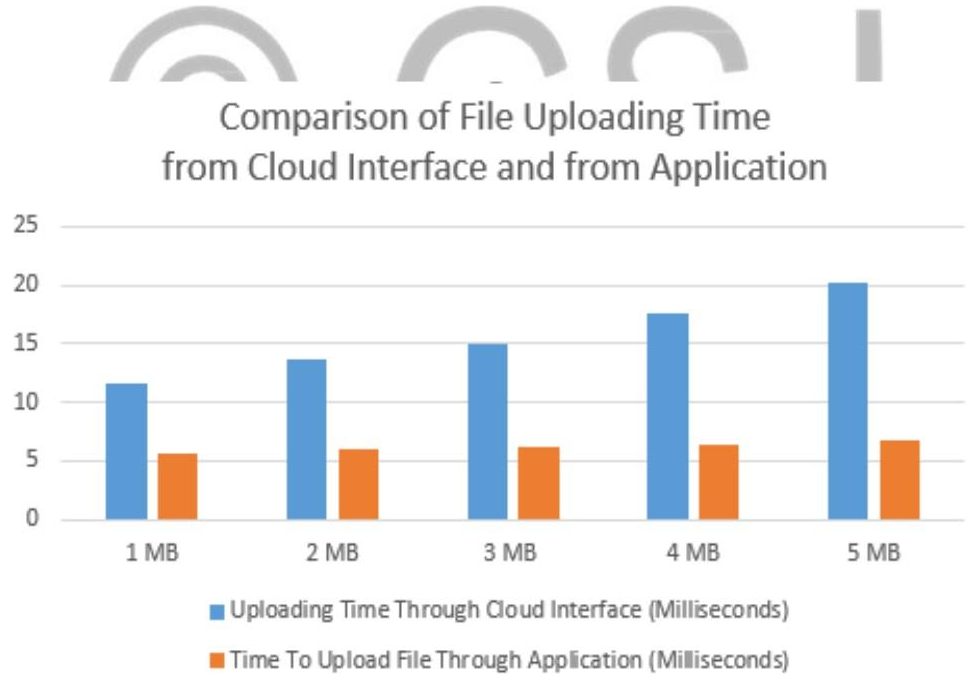

Fig. 11 Comparison of Uploading Time of Original File through Application and Cloud provided Interface

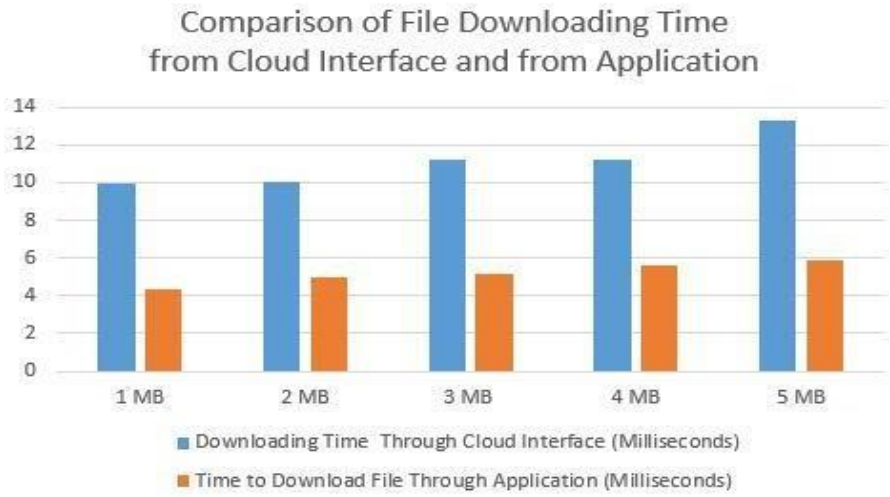

Fig. 12 Comparison of Downloading Time of Original File through Application and Cloud provided Interface

4.2 CONCLUSION

In this research a methodology has been proposed to enhance the confidentiality and integrity of the data stored in cloud storage. A strong encryption and integrity checking mechanism is devised to ensure data confidentiality and integrity at application level, which is achieved by using a strategy to place the file in cloud storage and hiding these details from others. Furthermore, an auditing mechanism is further provided to keep track of the users who have accessed the file by avoiding time consuming and costly TPA (Third Party Auditing) solutions. At the end the efficiency of the proposed approach is further discussed that is *application running time ≈ Log (n) where n is file size in bytes*.

4.3 FUTURE WORK

In future work we may extend the proposed approach for multimedia objects (Audio, Video and Graphics …etc.) files.

REFERENCES

[1]. N. Samreen and N. Khatri, "Introduction to Cloud Computing", in IRJET, Vol.5, pp. 2395-2405, 2019.

[2]. V. Suresh and M. Kumar, "An Efficient and Secure Data Storage Operations in Mobile Cloud Computing", in IJSRSET, Vol. 4, pp. 1385-1390,2018.

[3]. A. Venkatesh and M. Eastaff, "A Study of Data Storage Security Issues in Cloud Computing", in IJSRCSEIT, Vol. 3, pp. 1741-1745, 2018.

[4]. D. Hyseni and A. Luma, "Proposed Model to Increase Security of Sensitive Data in Cloud Computing", in IJACSA, Vol. 9, pp. 203-210, 2018.

[5]. A. Vijayalakshmi and N. Veeraragavan, "Unified Model for Cloud Data Confidentiality", in AJSAT, Vol. 7, pp. 23-27, 2018.

[6]. Geethamani and Ranjani, "Preserving Privacy in Public Auditing for Data Storage Security in Cloud Computing", in IJSRCSEIT, Vol. 3, pp. 1757-1762, in 2018.

[7]. A Valerian and C. Nadunagyu, "Improvement to the Confidentiality of Cloud Data", in IJRSC, Vol 3, pp. 156-169, 2018.

[8]. T. Kulkarni andV. Waghmare," Security Implementation in Cloud Computing using Behavior Profiling and Decoy Technology" in WJTER,

Vol.3, pp. 108-113, 2018.
[9]. E. Agrawal and P. Ram, “Cryptography Based Security for Cloud Computing System”, in IJARC, Vol. 8, pp. 2193-2197, 2017.
[10]. S. Ksasy and E. Takieldeen, “Advanced Cryptographic Algorithm System for Binary Codes by Means of Mathematical Equation”, in ICIC International, Vol. 10, pp. 1-8, 2016.
[11]. M. Naik and P. Tungare, “Color Cryptography using Substitution method”, in IRJET, Vol. 3, pp. 941-944, 2016.