SECURE GROUP MANAGEMENT FOR CLOUD-BASED SHARED DATA ENABLES PUBLIC AUDITING AND PROTECTS USER PRIVACY

#1**PADIGELA SUSMITHA,**
#2**Dr.D.SRINIVAS REDDY,** ***Associate Professor,***
#3**Dr.V.BAPUJI,** ***Associate Professor& HOD,***
Department of Master of Computer Applications,
VAAGESWARI COLLEGE OF ENGINEERING, KARIMNAGAR, TELANGANA.

ABSTRACT: Cloud storage enables users to store their data remotely and access high-quality apps and services on demand from a shared pool of reconfigurable computing resources, eliminating the need to manage and retain their data locally. However, because users no longer physically control the outsourced data, ensuring its integrity in cloud computing is difficult, especially for users with low computational capacity. Furthermore, users should not have to worry about cloud storage integrity; rather, they should be able to access it as if it were local. As a result, providing public auditability for cloud storage is critical so that users may rely on a third party auditor (TPA) to check the accuracy of outsourced data while feeling secure. The auditing technique should not generate any additional online constraints for users or vulnerabilities affecting user data privacy in order to appropriately and successfully construct a TPA. We present a private public auditing mechanism for a secure cloud storage system in this research. We broaden our investigation so that the TPA can audit multiple consumers effectively and concurrently. The recommended solutions are both provably secure and exceptionally effective, according to a rigorous security and performance assessment.

Index Terms—Data storage, privacy-preserving, public auditability, cryptographic protocols, cloud computing.

1. INTRODUCTION

Cloud computing has been conceptualized as the forthcoming enterprise information technology (IT) framework, owing to its extensive array of unparalleled benefits in the history of IT. These advantages include on-demand self-service, ubiquitous network access, location-independent resource pooling, rapid resource elasticity, usage-based pricing, and risk transference. The advent of cloud computing has brought about significant disruptions, leading to a comprehensive revolution in the manner in which enterprises employ information technology. The process of centralizing data or outsourcing it to the cloud is a fundamental element of this paradigm shift. From the standpoint of users, encompassing both individuals and IT companies, the practice of remotely storing data in a flexible and on-demand manner through cloud technology presents appealing benefits.

These advantages include alleviating the burden of storage management, enabling universal data access across different geographical locations, and eliminating the need for capital expenditures associated with hardware, software, and personnel maintenance, among various other advantages. The advent of cloud computing has rendered these advantages increasingly enticing, yet it has also unveiled novel and formidable security risks to users' outsourced data. As cloud service providers (CSPs) are separate administrative organizations, the act of data outsourcing involves relinquishing full authority over the fate of the user's data. The accuracy of cloud-based data is degraded as a result of the considerations outlined below.

Cloud infrastructures are subject to a diverse range of internal and external data integrity concerns, however they are notably more robust and reliable compared to personal PCs. Allegations have been made regarding instances of service disruptions and security breaches that have impacted prominent cloud providers. Furthermore, cloud service providers (CSPs) have the potential to deceive cloud customers in regards to the actual whereabouts of their outsourced data due to many factors. For example, a Cloud Service Provider (CSP) may opt to remove data that has remained inactive for an extended period in order to reduce costs. Alternatively, the CSP can choose to conceal occurrences of data loss as a means of safeguarding its standing in the industry.

In summary, although the practice of outsourcing data to the cloud offers cost-efficiency for extensive data storage over an extended period, it does not guarantee instant assurances pertaining to the availability and integrity of the data. Failure to address this matter comprehensively could potentially hinder the successful implementation of the cloud architecture. Due to the absence of physical possession of data storage by consumers, the direct application of classical cryptographic primitives for safeguarding data security is rendered impractical.

The impracticality of downloading the entire dataset at once for the purpose of validating its integrity arises from the significant expenses associated with I/O and network transmission. Furthermore, the mere detection of data corruption upon data viewing often proves inadequate, as it fails to offer consumers a guarantee of the accuracy of previously unaccessed content and may occur too late to rectify data loss or harm. The verification of data accuracy in a cloud environment can pose challenges and incur significant costs for cloud users, due to the substantial volume of outsourced data and the limited resource capacity available to them. Furthermore, it is vital to minimize the expenses associated with cloud storage to ensure that users can effortlessly access and utilize their data without the need for extensive efforts.

It would be advantageous, for instance, if users could access data without concerns over its integrity before to or subsequent to retrieval. Moreover, it is worth noting that numerous users have the ability to access a shared cloud storage system, much like the scenario observed in a typical office environment. In order to streamline administrative processes, it is recommended that the cloud server exclusively processes verification requests originating from a singular designated entity.

The provision of public auditing services for cloud-based data storage is of utmost importance in order to enable customers to engage an impartial third-party auditor (TPA) when necessary for the purpose of scrutinizing outsourced data. By implementing this approach, it will guarantee the preservation of data integrity in its whole, while simultaneously optimizing the utilization of computational resources and mitigating the burden on the internet infrastructure for clients utilizing cloud services. Users now have a more accessible and cost-effective alternative to guarantee the precision of their cloud storage. This is made possible through the involvement of a trusted third-party auditor (TPA), who possesses the expertise and capabilities that users may lack. The TPA assumes the responsibility of periodically verifying the integrity of all data saved in the cloud on behalf of the users. Furthermore, the outcomes of third-party audits (TPA) can be advantageous not only for consumers in evaluating the potential risks associated with their cloud data services, but also for cloud service providers in enhancing their cloud-based service platform.

Additionally, these audit results have the potential to serve as a basis for independent arbitration. The provision of public auditing services will play a vital role in facilitating the comprehensive growth of the cloud economy. This is because users will want a reliable mechanism to evaluate potential risks and establish a sense of confidence and reliance in cloud services. In contemporary discussions surrounding system and security paradigms, the notion of public auditability has emerged as a significant consideration for ensuring the integrity of data stored remotely. Public auditability enables a third party, apart from the user, to verify the precision of data that is stored in a remote location. However, a significant portion of these procedures prove ineffective in safeguarding consumer data during external audits.

In practice, it is possible for auditors to be provided with access to user information. The aforementioned critical issue has a substantial influence on the security of diverse cloud computing platforms. From a data privacy standpoint, users who possess their own data and depend exclusively on a third-party auditor (TPA) for ensuring the security of their data storage are concerned about the potential introduction of new vulnerabilities that could lead to illegal information leaking, so compromising their data security.

The disclosure of outsourced data to third parties is prohibited by various statutes in the United States, including the Health Insurance Portability and Accountability Act (HIPAA). One potential approach to address this privacy concern is to employ data encryption before outsourcing.

However, it is important to note that this measure serves as an additional safeguard and should be complemented by the privacy-preserving public auditing strategy put forward in this research. In the absence of a well-defined auditing protocol, encryption alone is insufficient to prevent the unauthorized disclosure of data to third parties during the auditing process.

Rather than effectively addressing the matter of data privacy, it merely simplifies it to the administration of encryption keys. The ongoing threat of illegal data leakage remains a concern as a result of the potential revelation of decryption keys. This study aims to address the challenge of establishing a third-party auditing protocol that upholds privacy and operates autonomously from data encryption.

Our research on data storage represents one of the pioneering efforts in the field of cloud computing, since it introduces a novel approach to public auditing that effectively safeguards user privacy. Furthermore, as a result of the widespread adoption of cloud computing, Third-Party Auditors (TPAs) may experience a surge in their auditing responsibilities from diverse user groups. Given the time-consuming and labor-intensive nature of individually auditing these growing duties, there arises a need to facilitate the TPA's ability to efficiently carry out various auditing tasks in a batch manner, that is, simultaneously. In order to surmount these challenges, the utilization of the public key based homomorphic linear authenticator (referred to as HLA) technique is employed.

The use of Third Party Auditors (TPA) enables the conduction of audits without the need for a local data copy, resulting in a notable reduction in communication and processing overhead when compared to traditional data auditing approaches. Through the integration of the Human Leukocyte Antigen (HLA) with a random masking technique, our proposed protocol guarantees the prevention of any information disclosure to the Third Party Auditor (TPA) regarding the data kept on the cloud server during the auditing procedure.

The architecture employed for bulk auditing in our system also leverages the aggregate and algebraic capabilities of the authenticator. The contribution we have made can be observed through three distinct characteristics: In this paper, we propose a protocol designed to safeguard privacy during auditing processes. Additionally, we advocate for the implementation of a public data storage security auditing system inside the realm of cloud computing. Through the utilization of our system, an external auditor possesses the capability to conduct an analysis of a user's cloud-based outsourced data without initially discerning the specific contents of this data.

Our concept is a pioneering effort in the field of cloud computing, as it is the only known approach to offer scalable and efficient public auditing capabilities. The solution we propose is designed to facilitate batch auditing, allowing the Trusted Third Party (TPA) to efficiently carry out many auditing duties simultaneously, as delegated by different users.

2. RELATED WORK

Ateniese was the first to incorporate the concept of public auditability into their "provable data possession" (PDP) technique for establishing ownership of data files held on untrusted storage systems. The proposed approach for auditing outsourced data uses a subset of randomly selected file blocks in conjunction with homomorphic linear authenticators based on the RSA algorithm. An external auditor need access to the linear combination of sampled blocks for a scheme to be publicly auditable. If their protocol is utilized directly, auditors may gain access to user data due to a lack of privacy transparency. Juels established the notion of "proof of retrievability" (PoR), which uses spot-checking and error-correcting codes to ensure the accessibility of data in distant archive service systems.

There is a cap on the number of audit problems that a user can tackle, and the fundamental structure that they rely on to do so is not open to public scrutiny. Although it describes a basic Merkle-tree structure for public Proofs of Retrievability (PoR), this technique is only applicable to encrypted data. Dodis explores multiple Proof of Retrievability (PoR) systems to ensure auditability in a private setting. Shacham enhanced the aforementioned security framework's Proof-of-Retrievability (PoR) method by using BLS signatures, which provide additional security assurances. Using BLS signatures, which have been shown to be secure, to produce homomorphic linear authenticators is conceptually comparable.

The implementation of the complex BLS design results in a transparent and easily auditable system. As was previously mentioned, the authors' method does not offer audits that protect users' privacy. According to Shah (year), the use of trusted third-party administrators (TPAs) can significantly boost cloud storage security. After decrypting the data, symmetric-keyed hashes are created and given to an auditor for inspection. The auditor verifies that the server actually possesses the decryption key it claims to possess, in addition to verifying the file's integrity.

This approach is only applicable to encrypted data, is auditable, has a finite number of applications, and may expose users to network congestion if and when the supply of keyed hashes is exhausted. Ateniese provides a largely dynamic adaptation of the preceding PDP technique mentioned in the cited literature using simply symmetric key cryptography and a small number of audits. Partially dynamic data storage with error localisation and equivalence support is explored by Wang et al. inside a distributed setting.

A mechanism to provide full data dynamics and enable public auditability is proposed by Wang in a future study, which involves the combination of Merkle Hash Trees (MHTs) and Boneh-Lynn-Shacham-based Homomorphic Linear Authenticators (BLAs). Erway has created a method based on skip lists that allows for both reversible operations and verifiable data ownership. Both of these methods are vulnerable to audits that compromise users' privacy since they rely on linear combinations of sampled blocks for verification. The aforementioned techniques may enhance audit productivity and data dependability, but they fall short of wholly satisfying the condition for protecting user privacy during public audits in the cloud. Also, group auditing, which can significantly cut down on the TPA's computing costs even when dealing with a large number of audit delegations, is not used in any of these approaches.

3. SYSTEM DESIGN

As has been shown, a public auditing method is crucial to calming the concerns of users whose data is stored in the cloud. In this section, we suggest a public auditing protocol to use a third party to conduct audits on behalf of data owners as necessary while maintaining confidentiality. The client, the third-party auditor (TPA), and the cloud service provider are the three main players in the proposed protocol, as shown in Figure 1. The cloud service provider (CSP) is the business whose servers your information will be stored on. Owners of data that are interested in using a cloud storage service are the customers. TPA is the government agency that is tasked with conducting audits of public records. The goals of developing the suggested auditing protocol can be summed up as follows:

1. **Public audibility**: A Third Party Auditor (TPA) is used to ensure the security of data kept in the cloud, without disrupting the online activity of clients.
2. **Storage Correctness**: The TPA (Third-Party Auditor) can check for signs of data corruption within individual blocks of data to verify the completeness of the data.
3. **Improving data availability**: The method entails shoring up defenses against attacks like denial of service (DOS) that might disrupt legitimate data access.
4. **Preserving data confidentiality:** This includes protecting data at all times, including while it is stored in the cloud and when it is being examined by independent auditors (TPAs).
5. **Efficiency**: The time it takes to perform an audit and for data to be stored in the cloud should be reduced as much as possible.

The proposed system accommodates both semi-trusted and fully-trusted TPA deployments. In the semi-trusted mode, the TPA only conducts the auditing process on the user's behalf when specifically requested to do so. In contrast, in Fully Trusted mode, the Trusted Platform Agent (TPA) acts as a go-between for the client or data owner and the cloud service provider. Simply put, the cloud service provider and the customer are not in constant communication with one another. Figure 3 shows how the client is assigned the mutual responsibility of deciding which files to upload to the cloud. After that, the file is split into pieces of uniform size.

The MD5 algorithm is then applied, resulting in a hash value for each block that is 128 bits in length. The hashes are then concatenated into a single string that is fed into the AES cipher as its input. Furthermore, it is the client's responsibility to keep the encryption keys safe on their own machine. There are two possible methods for archiving information on the cloud. In the first case, the TPA is acting in a SemiTrusted capacity, which places the onus of uploading the content into the cloud storage with the client rather than the TPA. In the latter case, with the TPA operating in Fully Trusted mode, the client sends the encrypted data to the TPA and chooses between many cloud storage options. The TPA is then accountable for transferring the information to the predetermined cloud repository.

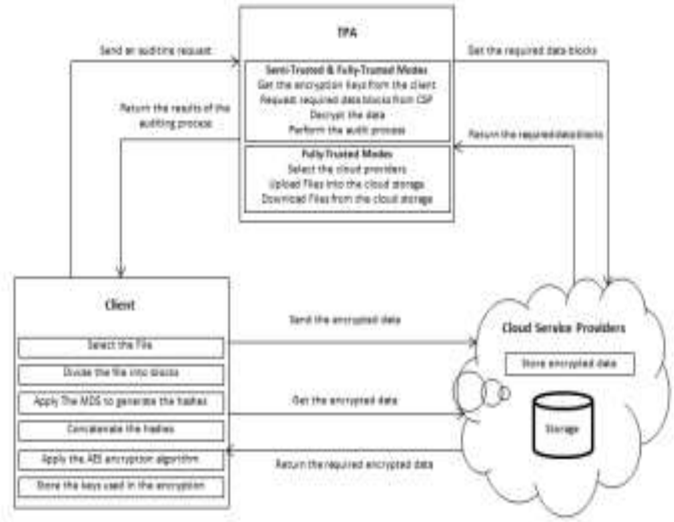
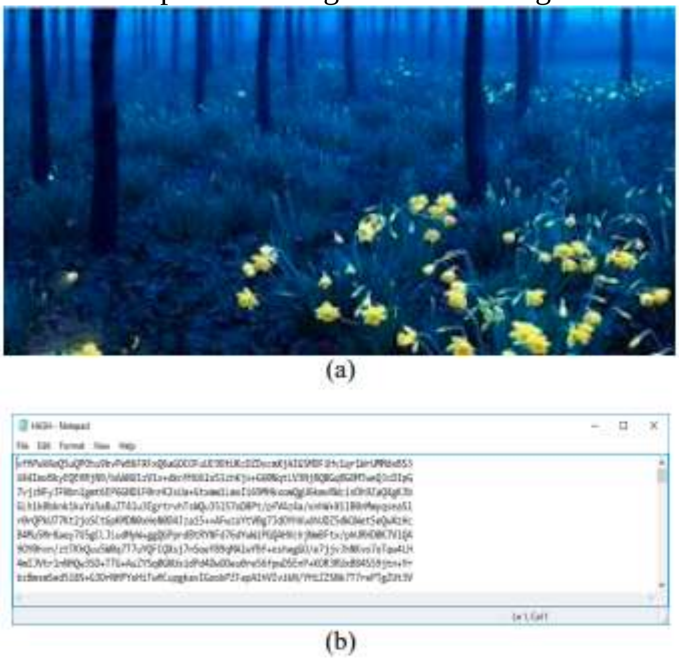

TPA
Semi-Trusted & Fully-Trusted Modes
Get the encryption keys from the client
Request required data blocks from CSP
Decrypt the data
Perform the audit process
Fully Trusted Modes
Select the cloud providers
Upload Files into the cloud storage
Download Files from the cloud storage
Send an auditing request
Return the results of the auditing process
Get the required data blocks
Return the required data blocks
Client
Select the File
Divide the file into blocks
Apply The MD5 to generate the hashes
Concatenate the hashes
Apply the AES encryption algorithm
Store the keys used in the encryption
Send the encrypted data
Get the encrypted data
Return the required encrypted data
Cloud Service Providers
Store encrypted data
Storage

Figure 1: The architecture of the proposed auditing mechanism

On the client side, the MD5 algorithm is most commonly used for two purposes. Although MD5 is well-known as an encryption method, in the proposed method, it serves primarily as a data-compression tool. By using compression, the AES encryption process and the round-trip time between the client and the cloud service provider can be sped up. Figures 2a and 2b show a real-world example of using the MD5 algorithm.



(a)

(b)

Figure 2a is a jpg color image with a resolution of 280 * 147 and total size of 22.4 KB.

Figure 4a shows the image size in bytes, and Figure 2b shows the concatenated hashes of the image. AES was chosen as the primary encryption method for the proposed auditing mechanism due to its resistance to common attacks, high performance, and ease of implementation. Implementing a double encryption protocol ensures that sensitive information remains secret and makes it more challenging for outside parties to access the data.

The client independently sends the encrypted data to the selected cloud storage and stores the encryption keys in a private cloud if a semi-trusted Third Party Auditor (TPA) is used. If the client uses a TPA that can be completely relied upon, however, the TPA will be responsible for sending the encrypted data to the customer's preferred storage location. The steps involved in putting data into a cloud-based storage system are shown in Figure 5. The proposed layout makes use of multiple clouds, or the "multi-cloud paradigm." A private cloud is used to keep track of encryption keys, while encrypted data blocks are kept on a public cloud. In addition, using many public clouds, often known as multi-cloud, can increase data availability in the event of a cloud service provider outage. To do this, encrypted data blocks are stored in the public cloud and made accessible via other cloud service providers. After encrypted data has been saved to the cloud storage platform, the client can contact the TPA to begin the auditing process by giving the TPA the file id, cloud provider id, concatenated hashes, and the AES encryption key.

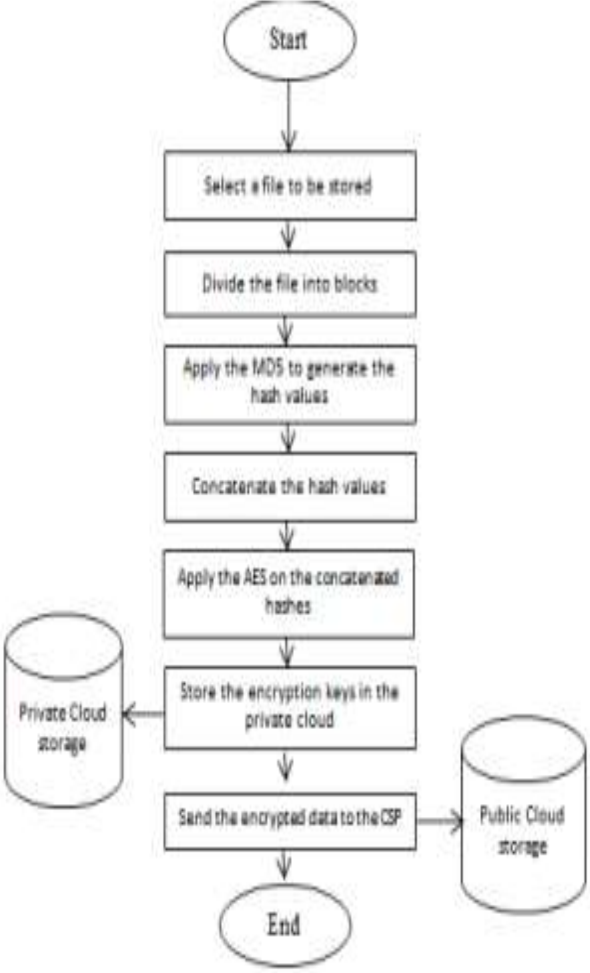

Start
Select a file to be stored
Divide the file into blocks
Apply the MD5 to generate the hash values
Concatenate the hash values
Apply the AES on the concatenated hashes
Store the encryption keys in the private cloud
Private Cloud storage
Send the encrypted data to the CSP
Public Cloud storage
End

Figure 3: Flow chart of the store data process

Customers in a Semi-Trusted Third Party Administrator (TPA) environment have access only to a subset of services, including enrollment, authentication, and auditing. The customer must register and create an account before they may begin using the TPA's services. Therefore, a valid login is necessary. The client is free to submit an auditing request to the TPA once they have completed the signup and login processes. In response to a request from a client, the Third Party Auditor (TPA) contacts the cloud service provider and asks them to provide over the relevant information. The Trusted Processing Agent (TPA) retrieves the encrypted data from the cloud provider, decrypts it with the client-supplied encryption key, compares the hashes, and then sends the results back to the client. By comparing hash values, the TPA demonstrates its competence in carrying out the auditing procedure without having access to the audited material. The auditing process flowchart is shown in Figure 4.

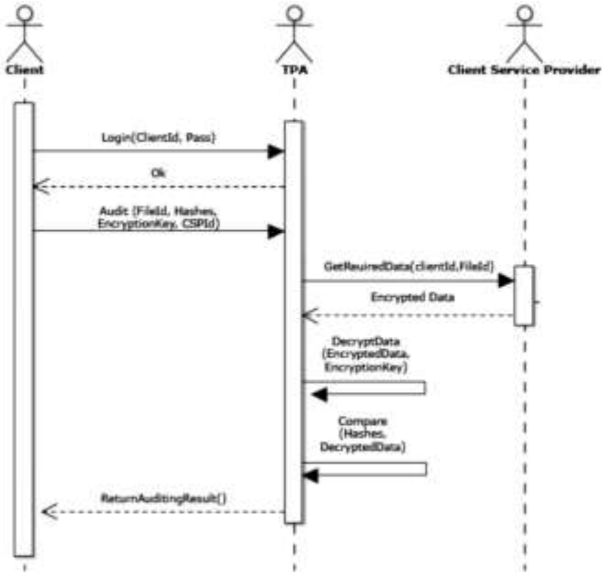

Client
TPA
Client Service Provider
Login(ClientId, Pass)
Ok
Audit (FileId, Hashes, EncryptionKey, CSPId)
GetReuiredData(clientId,FileId)
Encrypted Data
DecryptData (EncryptedData, EncryptionKey)
Compare (Hashes, DecryptedData)
ReturnAuditingResult()

Figure 4: Sequence diagram for the audit data process

The Fully-Trusted Third Party Auditor (TPA) provides a Graphical User Interface (GUI) that lets the user choose the appropriate CSPs for storing encrypted data, thus negating the necessity for direct contact between the client and the CSPs. Through a graphical user interface (GUI), customers can ask the TPA to retrieve the necessary information from cloud storage.

4. SYSTEM ANALYSIS

The proposed research will look into the use of a TPA built inside a web service. Users who have signed up for and logged into a special website created for this purpose are granted access to this service. The proposed architecture provides the data owner with two different entry points to the TPA.

In the first strategy, the TPA is accessible via multiple web browsers, including Chrome, Firefox, Internet Explorer, and others, for the convenience of the data owner or customer. The second strategy involves the data owner or customer using an Android smartphone to access the third-party auditor (TPA) via a mobile application. Netbeans was used as the IDE, Java, JSP, Java servlet, HTML, Java Script, MySQL, and Jason were all employed in the creation of the proposed system. The following hardware was used in the study alongside Microsoft's Windows 7 operating system for the client environment.

A Pentium IV processor clocking in @ 2.6 GHz powers the computing system.

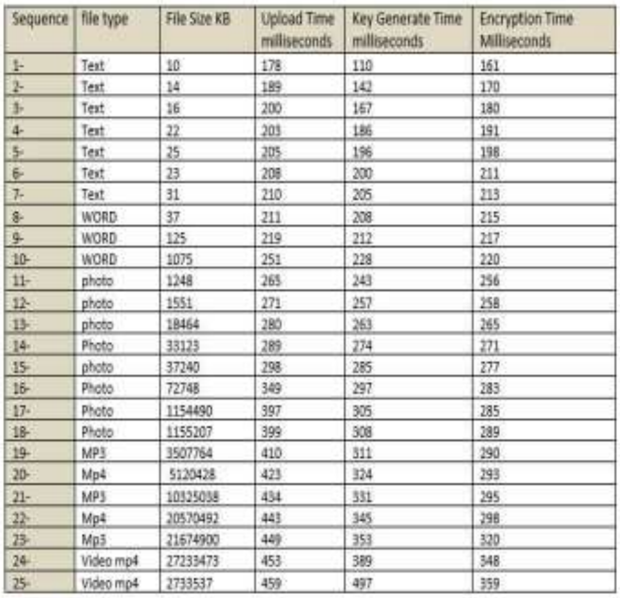
The computer has 512 MB of RAM, all of which is Double Data Rate (DDR).

There are a total of 150 distinct files that make up the dataset used in the ensuing studies. Table 1 provides an in-depth breakdown of the main file formats, their respective numbers, and their originating sources. The files range in size from a few hundred bytes to over 1.5 gigabytes.

sequence	file type	Number of files	View source
1	Document File	37	Created using Microsoft Word.
2	Text File	25	Created using Notepad.
3	Image File	35	internet
4	PDF File	14	Internet
5	Video File	24	YouTube
6	Audio File	3	internet
7	PowerPoint File	12	Created using Microsoft PowerPoint

Table 1. The details of the files set used to evaluate the performance of the TPA

In this section, we will discuss the results of the experiments we ran to see how effective the proposed method was. Table 2 displays the dataset used in the first three studies. Table 2 details the 25 separate files that make up the data collection procedure and their individual sizes and types. The first three checks are run in order to calculate the total time spent uploading, creating keys, and encrypting.



Sequence	file type	File Size KB	Upload Time milliseconds	Key Generate Time milliseconds	Encryption Time Milliseconds
1-	Text	10	178	110	161
2-	Text	14	189	142	170
3-	Text	16	200	167	180
4-	Text	22	203	186	191
5-	Text	25	205	196	198
6-	Text	23	208	200	211
7-	Text	31	210	205	213
8-	WORD	37	211	208	215
9-	WORD	125	219	212	217
10-	WORD	1075	251	228	220
11-	photo	1248	265	243	256
12-	photo	1551	271	257	258
13-	photo	18464	280	263	265
14-	Photo	33123	289	274	271
15-	photo	37240	298	285	277
16-	Photo	72748	349	297	283
17-	Photo	1154490	397	305	285
18-	Photo	1155207	399	308	289
19-	MP3	3507764	410	311	290
20-	Mp4	5120428	423	324	293
21-	MP3	10325038	434	331	295
22-	Mp4	20570492	443	345	298
23-	Mp3	21674900	449	353	320
24-	Video mp4	27233473	453	389	348
25-	Video mp4	2733537	459	497	359

Table 2. The details of the files set used in the first three experiments and the results of the different experiments

As was noted before, the first experiment was conducted to determine the time required to upload a collection of files ranging in size from several hundred to several thousand bytes. The time it takes for a file to be uploaded to a cloud storage service. In other words, it describes the time lag between starting an upload and having it finished. Figure 5 and Table 2 show the outcomes of the first experiment.

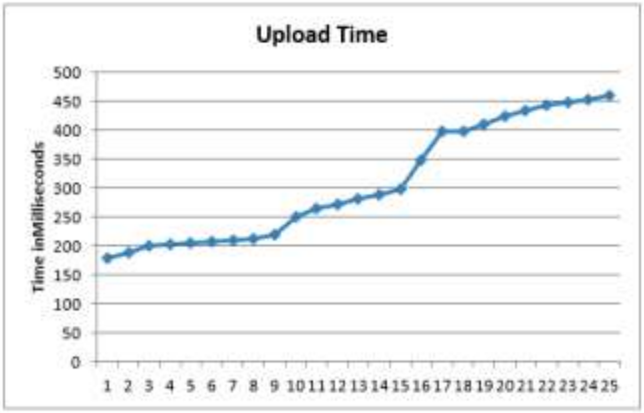


Figure 5: The upload time for a set of files with different sizes.

In the second test, we estimated how long it would take to produce AES encryption keys for all of the files that needed to be encrypted. Figure 6 and Table 2 show the findings of the additional research. The third experiment shows the results in Figure 7 by calculating the time needed to encrypt each file in the dataset. In both experiments, we used 25 different files from the dataset.

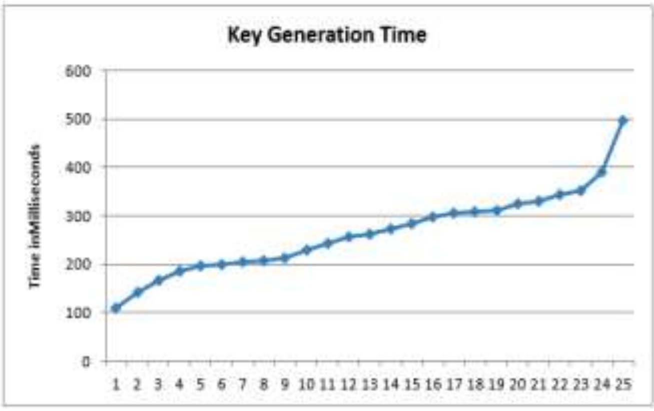


Figure 6: The time of generating the encryption keys for a set of files with different sizes

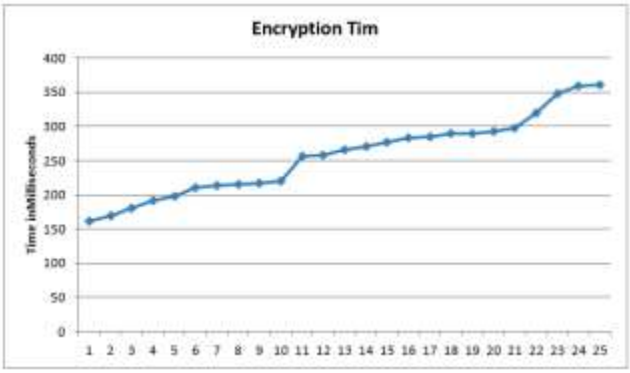


Figure 7: The time of encryption using the AES for a set of files with different sizes.

The available data shows a positive relationship between file size and the times it takes to upload, generate a key, and encrypt a file. The fourth test compared how long it took using the proposed method to perform the auditing procedure on files of varying sizes. The results are shown in Table 3 and Figure 8 in milliseconds. The findings point to a negative relationship between file size and the auditing time required.

sequence	File Size KB	Auditing Time Existing System milliseconds	Auditing Time Proposed System Milliseconds
1-	20	25	19
2-	50	34	29
3-	80	41	33
4-	110	49	40
5-	140	56	46
6-	170	62	51
7-	200	69	58
8-	230	76	63

Table 3. Size of the files used to measure the speed of Auditing process

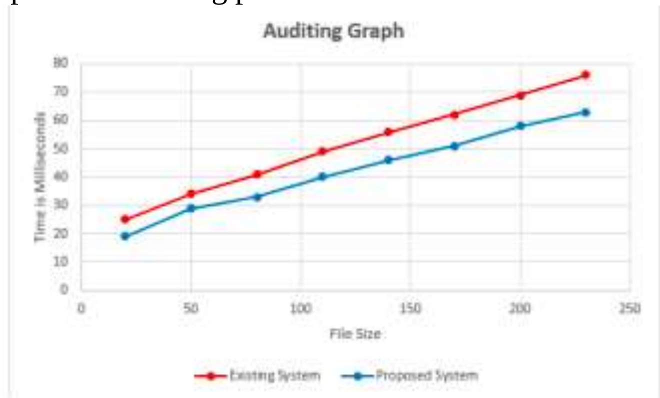


Figure 8: The time of auditing using the existing and the proposed methods for a set of files with different sizes.

In the final experiment, we use two criteria—the false positive rate and the false negative rate—to determine how well the proposed TPA works. A total of 150 files, with a combined size of almost 1.5 GB, were successfully uploaded in the trial. It was decided to modify half of the files and leave the other half unchanged so that the accuracy of the TPA could be evaluated. The TPA is then responsible for determining whether or not the collection is complete and accurate. How often a Third Party Application (TPA) falsely generates an alert that a change has been made to a file is measured by its False Positive Rate (FPR). On the other hand, the False Negative Rate (FNR) measures how often the TPA incorrectly reports an unchanged file. The auditing method uses the false negative rate for the changed files and automatically calculates the false positive rate for the original files. The False Negative Rate (FNR) and False Positive Rate (FPR) are calculated using Equations 1 and 2, respectively.

$$FPR = FP\ N = FP\ FP{+}TN \qquad (1)$$

$$FNR = FN\ N = FN\ FN{+}TP \qquad (2)$$

Let's say the total number of negatives is N, and that FP plus TN adds up to N. Number of false negatives (FN), number of true positives (TP), number of false positives (FP), and number of true negatives (TN). Figure 9 displays the results of an audit to determine the accuracy of the suggested method. Table 4 shows the results as well. Based on the numbers, it's clear that our method achieves a lower FPR and FNR than competing approaches.

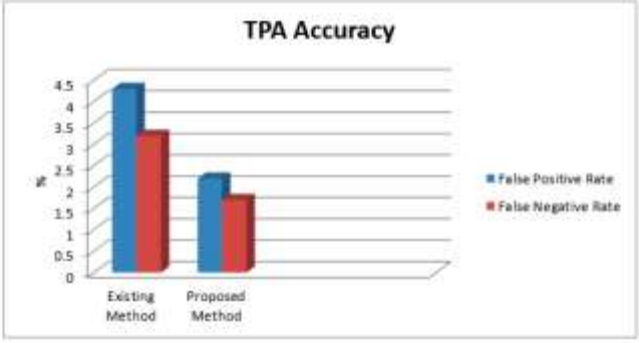


Figure 9: The auditing accuracy using the existing and the proposed methods for a set of files with different sizes

The Auditing Mechanism	False Positive Rate [FPR]	False Negative Rate [FNR]
In [7]	**4.3%**	**3.2%**
The Proposed Work	2.2%	1.7%

Table 4. the results of the auditing accuracy in terms of FPR and FNR

Using metrics like false positive rate (FPR) and false negative rate (FNR), past research shows that the proposed public auditing method is superior to the alternative strategy in terms of both auditing time and accuracy.

5. CONCLUSION

The ability to store, manage, and access data via an internet interface is a key feature of cloud storage services. There are a number of roadblocks and research problems that need to be overcome in the field of cloud storage if we want to see an increase in client satisfaction with this service. Users have voiced worries about the security, privacy, and availability of their information. Because users have so little say over the remote cloud nodes, they worry that their data could be accessed or altered by third parties without their knowledge. Therefore, consumers' legitimate worries about cloud data storage need to be allayed through a public auditing method. To protect personal information, this research proposes a public auditing protocol in which an independent party is hired to do audits on behalf of data owners. Public audibility, proper storage, increased data availability, data secrecy preservation, and improved efficiency are only few of the design goals that the proposed protocol tries to realize. Assuring both public accessibility and storage accuracy requires an auditing process to be conducted by a Trusted Third Party (TPA) with the necessary access privileges to detect any instances of data corruption or tampering. The replication concept and a multi-cloud architecture can improve data accessibility while keeping sensitive information secure. To do this, data can be encrypted before being stored in the cloud, allowing for audits to be conducted by a Trusted Third Party Auditor (TPA) without the TPA needing access to the audited data. The MD5 method of data compression leads to optimal effectiveness. This method efficiently shortens the time needed to encrypt data using the AES algorithm and transport data between the client and CSP. Through a battery of trials, we've evaluated our method's performance on a wide variety of files of varying sizes with respect to both time and accuracy. The results show that the suggested Third-Party Application (TPA) has a false positive rate of 2.2% and a false negative rate of 1.7% when used to evaluate the integrity of a set of 150 files of varied sizes, totaling roughly 1.5 Gigabyte.

REFERENCES

1. Zhang, Q., Cheng, L. and Boutaba, R., 2010. Cloud computing: state-of-the-art and research challenges. Journal of internet services and applications, 1(1), pp.7-18.

2. Malik, S., Huet, F. and Caromel, D., 2012, December. RACS: a framework for resource aware cloud computing. In Internet Technology And Secured Transactions, 2012 International Conference for (pp. 680-687). IEEE.

3. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I. and Zaharia, M., 2009. Above the clouds: A berkeley view of cloud computing.

4. Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z. and Song, D., 2007, October. Provable data possession at untrusted stores. In Proceedings of the 14th ACM conference on Computer and communications security (pp. 598- 609). Acm.

5. Wang, Q., Wang, C., Li, J., Ren, K. and Lou, W., 2009, September. Enabling public verifiability and data dynamics for storage security in cloud computing. In European symposium on research in computer security (pp. 355-370). Springer Berlin Heidelberg.

6. Wang, C., Chow, S.S., Wang, Q., Ren, K. and Lou, W., 2013. Privacy-preserving public auditing for secure cloud storage. IEEE Transactions on computers, 62(2), pp.362-375.

7. More, S. and Chaudhari, S., 2016. Third Party Public Auditing Scheme for Cloud Storage. Procedia Computer Science, 79, pp.69-76.

8. Wang, B., Li, B. and Li, H., 2014. Oruta: privacypreserving public auditing for shared data in the cloud. IEEE transactions on cloud computing, 2(1), pp.43-56.

9. Zhang, J.H. and Zhao, X.B., 2015. Privacypreserving public auditing scheme for shared data with supporting multi-function. J. Commun, 10(7), pp.535-542.

10. Bhagyashri, S. and Gurav, Y.B., 2014. Privacypreserving public auditing for secure cloud storage. IOSR Journal of Computer Engineer ing (IOSR-JCE), 16(4), pp.33-38.