International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 6 (2017) pp. 1093-1101
© Research India Publications. http://www.ripublication.com

An Efficient Privacy Preserving Public Auditing Mechanism for Secure Cloud Storage

Mohaned Zkaria Salem[1], Sahar F. Sabbeh[2] and Tarek EL-Shishtawy[3]

[1,2,3] Information System department,
Faculty of Computer and Information Benha University Qalyubia, Egypt.

Abstract

one of the services provided by the cloud environment is cloud storage which allows users to store, remotely manage and access their data over the internet. However, the limited control granted to users over remote cloud nodes makes users worry about the integrity of their data as the user' data can be accessed or modified by external attackers. Therefore, a new terminology called data auditing has been evolved to allow users to use cloud storage as they use their local storage without worrying about the integrity of their data. In order to achieve the auditing task, users can resort to an independent entity called Third Party Auditor (TPA) rather than putting the burden of the auditing task on the users themselves. The job of the TPA is to monitor the modifications, deletions, and insertions made on the user's data. However, using a TPA can endanger the data privacy and hence a privacy preserving TPA is required. In this paper, we have designed and implemented a cloud storage framework which includes an efficient TPA with main objective is to provide the public data auditing service while preserving the data privacy. Also, it can act as an intermediary between the data owner and the cloud service provider through which the users can upload, access, modify, delete, and audit their data while maintaining its privacy. The proposed auditing protocol is based on using two well-known encryption techniques, namely, message digest algorithm 5 (MD5) and advanced encryption standard (AES). Also, the TPA allows the users to store their data on multiple clouds to enhance the availability of data. A set of experiments have been conducted to evaluate the performance of the proposed public auditing mechanism.

Keywords: Cloud Storage, TPA, Data Privacy, Data Integrity, Encryption.

INTRODUCTION

Cloud computing is a distributed computing paradigm which is able to host and provide a variety of internet based services for the customers. In recent years, the cloud computing usage rate has increased rapidly because of the different advantages provided by the cloud. These advantages include adopting the pay-as-you-go principle as a pricing model, providing as much resources as needed which achieves a high degree of scalability, saving the efforts and money needed to maintain the IT infrastructure, and decreasing the costs and risks of launching an investment where the user is not obliged to spend a lot of money to build the required infrastructure. All of these advantages attracted many enterprises to depend on the cloud infrastructure rather than the in-house infrastructure [1, 2].

Among the services provided by the cloud is storage as a service which allows the users to store, to remotely manage and to access their data over the internet. From the users' point of view, including IT enterprises and individual users, cloud storage service has achieved a set of appealing advantages which includes removing the burden of data storage management from the users' shoulders, easy data access through the internet regardless the current geographical locations, saving the money needed for the software, hardware, and maintenance, etc [3]. However, there is a set of challenges and research problems which came with cloud storage service and need to be alleviated to increase the user's contentment for this service. One of these problems is the users' concern about the integrity and availability of their data and their feelings that the data can be accessed or modified by external intruders because of the limited control granted to the users over the remote cloud nodes. This worry from the users can be justified when we know that security threats and service outages are occurring for the cloud services from time to time [4]. Also, there are many reasons which can motivate the cloud service provider to behave inappropriately toward the outsourced data such discarding the data of a user for monetary reasons, hiding the security breaches which affected the data to save their reputation [5, 6]. Hence, there is no guarantee on the availability and integrity of data [4].

Therefore, data auditing has been evolved to allow the users to use the cloud storage as they use their local storage without worrying about the integrity of their data. Users can perform data auditing mission by themselves (private auditing) or they can exploit the expertise, knowledge, capabilities, and professional skills of an independent entity called Third Party Auditor (TPA) (public auditing) rather than putting the burden of the auditing task on the users' shoulders [4, 7]. It is necessary that the TPA should be able to audit the data without knowing its contents to the preserve the data privacy. Also, the data availability can be achieved through storing the data of users on multi-cloud instead of a single cloud.

In this paper, we have designed and implemented a complete framework by which the data owners can use the cloud storage in a more secure and reliable manner. In the proposed work, we consider a cloud data storage service which involves three entities: cloud service provider (CSP), Client, and a third party auditor (TPA) as shown in figure.1. CSP is the entity which owns the cloud servers on which the data would be stored. The client is the owner of the data who wants to exploit the cloud storage service provided by CSP. TPA is the entity which is responsible for providing the public data auditing service while preserving the data privacy. Also, if the user wants, it can act as an intermediary between the data owner and the cloud service provider through which the users can upload, access, modify, and delete their data. Also, the TPA allows the users to store their data on multiple clouds to enhance the availability of data.

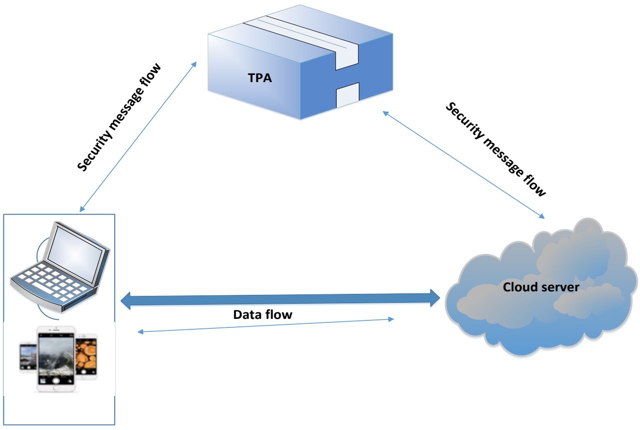


Figure 1: The architecture of the cloud storage service .

The rest of paper is organized as follows: The related work is overviewed in section 2. Then, the basic concepts needed to understand the proposed work are described briefly in section 3. In section 4, the proposed mechanism is presented with a detailed description of each component. In section 5, a set of experiments has been executed to evaluate the performance of the proposed mechanism. Finally, the paper is concluded and the proposed future work is provided in section 6.

RELATED WORK

It is obvious that the cloud storage service faces a set of challenges concerning the security and integrity of users' stored data. Therefore, there are a lot of efforts that has been done to make the user more comfortable during the use of cloud storage service. In [4], they proposed a public auditing method which can be performed by a TPA while preserving the privacy of data which stored on the cloud nodes. Their proposed method is based on random masking and homomorphic linear authenticator to make the TPA know nothing about the content of data during the auditing process. Also, they designed the TPA in such a way that makes it able to handle many auditing requests from multiple users on the same time in a batch manner.

In [8], the researchers have developed a mechanism for the public auditing service through a TPA while preserving the privacy of data. Their proposed mechanism has utilized ring signature for computing the verification information required to verify the integrity of the shared data. TPA cannot recognize the identity of the person who signed on each data block.

Also, the authors in [9] proposed another public auditing scheme to verify the integrity of the shared data which reside on the cloud. In their proposed method they support dynamic operation on data and group dynamic. Also, they decreased the computation burden put on the users through employing a proxy signature. On the same time, by adopting a Lagrange interpolating polynomial, the proposed method achieves the identity's privacy-preserving while keeping the communication overhead and computation cost small as possible.

Toward achieving the same goals, in [7] a public auditing scheme has been developed with design goals to verify the integrity of data along with confidentially, be privacy preserving, be efficient, and be secure. Their proposed auditing scheme employed AES cipher for encryption, SHA-2 for checking the data integrity, and RSA for calculating the digital signature.

Another public auditing scheme is suggested in [10] for monitoring data insertions, modifications, and deletions. The suggested scheme supports data dynamics and performs the auditing process by using multiple TPA. In addition to the ability of the suggested scheme to handle many auditing requests from multiple users at the same time through batch auditing, they used the concept of ring signatures besides improving block level authentication by using Merkle Hash Tree.

In [11], a public auditing scheme has been devised which performed by a TPA while preserving data privacy with efficient computation. Their solution is based on Chinese Theorem Remainder preceded by a cryptographic hash function with an attempt to optimize the computation at cloud server, TPA, and the owner of data. Also, another public auditing mechanism has been proposed in [12] for educational multimedia data stored in the cloud storage which allows fully dynamic auditing. They guarantee the privacy of data through utilizing a homomorphic hash function and random values beside its immunity against temper attack and lose attack.

In [13], a privacy preserving public auditing mechanism called Knox has been proposed for data maintained in the cloud storage in which there a large number of users who can access the data. In order to build homomorphic authenticators, Knox adopts group signatures which allow the TPA to perform the auditing process without the need for retrieving the whole data and without revealing the identity of the signer. The results have been shown that the amount of time and information required for the verification process are not influenced by the number of users.

At last but not least, the researchers in [14] have combined public verification with ID-based aggregate signature to build data integrity checker protocol. Besides performing the auditing process on behalf of the data owners, the proposed mechanism can reduce the overhead of checking tasks based on the identities of users.

BACKGROUND

In order to understand the proposed work, brief descriptions for the relevant concepts is provided in this section. We begin with demonstrating the concept of encryption as a tool for preserving the confidentiality of data then we focus on two types, namely, MD5 and AES as they are the encryption techniques employed in our work. After that a simple description about the concept of multi-cloud and why the owners of data are shifting toward it.

Encryption

In our modern society, the concept of information security became extremely important because of the huge amount of information which needs to be stored or transmitted in an electronic manner through the different networks, especially, the internet [15]. Generally, information security means the different processes designed to preserve the confidentiality, integrity, and availability of information [16]. Cryptography is one of the main tools for maintaining the security of the sensitive data, especially, the confidentiality issue [17]. Simply, Encryption turns a plain text into a cipher text to be stored or transmitted across a network in such a way that allows only the intended recipient to retrieve the plain text as shown in Figure 2 [18].

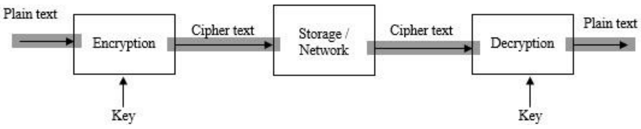


Figure 2: General data encryption scheme

On the basis of the key, encryption algorithms are classified into two main categories: *Symmetric* and *Asymmetric key* which also are known as secret-key and public-key, respectively. In Symmetric-key encryption algorithms, the same key (secret key) is used for both the encryption and decryption processes while in Asymmetric-key encryption algorithms, one key is used for the encryption process called a public key and another key called a private key is used for the decryption process [19]. The key is a sequence of numbers, letters, and/or symbols. Choosing the key is an important issue because the strength of the encryption algorithm depends directly on the chosen key [20]. On the basis of the input data, encryption algorithms are classified as block encryption algorithm or stream encryption algorithm. In block encryption, the size of input data (input block) is fixed. On the other side, in stream encryption algorithm, a continuous stream is used for encryption and decryption [21]. In the following subsections, brief descriptions are provided about two well-known encryption algorithms, namely, MD5 and AES.

Message Digest algorithm 5 (MD5) :

It is a well-known and widely adopted cryptographic hash function which designed in 1991 by Ronald Rivest to take the place of MD4. It takes a message of arbitrary length as input and generates 128 bits hash value. Initially, MD5 was designed to be utilized as a cryptographic hash function but a large number of vulnerabilities have been discovered. However, MD5 is still used for two main reasons. The first purpose, it can work as a checksum for data integrity verification against unintentional corruption. Secondly, it can be used when a large file needs to be compressed securely before being encrypted [22, 23, 24].

Advanced Encryption Standard (AES) :

AES is a symmetric block encryption standard recommended by The National Institute of Standards and Technology (NIST) to replace DES in 1997. This recommendation came after a competition to choose the best encryption algorithm. AES was designed to accomplish a set of characteristics such as its immunity against all known attacks, fast, and simple. Actually, until now, there is no a known attack that is able to unlock AES except brute force attack in which the attacker has to try all the character combination. Unlike DES which states that the size of the block and the key can be 64 and 56 bits, AES allows different sizes for the block and key which includes 128, 160, 192, 224, 256 bits. Furthermore, the sizes of the block and the key need not to be the same. However, AES standard is stating that AES algorithm can only handle a block of size 128 bit and a key of size 128, 192, or 256 bit [25].

Multi-cloud Storage

Cloud data storage or is one of the prominent services offered by the cloud service provider. Cloud storage service allows the customers to store their data on the cloud storage nodes instead of storing it on their own servers. Despite those many advantages which came with this service such as the flexibility, scalability, etc, there are some security issues which need to be addressed to make the customer of the service more comfortable [26]. One of the security issues that need to be considered is the availability of data, especially, when we know that security threats and service outages are occurring for the cloud services from time to time [4]. So, one of the methods which can handle the availability issue is using multi-cloud storage, also known as inter-cloud or cloud of clouds, instead of a single cloud storage because when data is stored on a single cloud that makes the system contains a single point of failure [27]. Away from the preventive measures taken by the cloud provider to ensure the availability of data, the dependency on multi-cloud can remove the fears of the data owners about the availability of their data. Also, there are many real cases where the cloud providers suffered from the service outage.

Therefore, it predicted that using single cloud storage becomes less popular because of the risks related to the service availability failure [28]. In addition to enhancing the availability of data, there some other reasons which can make the customers, whether individual or enterprises, more likely to use multi-cloud storage rather than the single cloud. One reason may be the desire of customers to be able to isolate private and public data through using a hybrid cloud which may consist of a private cloud and a public cloud. Another reason may be the need of customers for some secondary services which exist on other clouds [29].

THE PROPOSED FRAMEWORK

As demonstrated in the previous sections, there is a need for a public auditing mechanism which is able to conceal the fears of the users who have data stored on the cloud storage. In this section, a public auditing protocol has been proposed with main objective is to adopt a third party for performing the auditing process on demand on behalf of the data owners while preserving the data privacy. In the proposed protocol as shown in Figure 1, there are three main entities, namely, cloud service provider (CSP), client, and a third party auditor (TPA). CSP is the entity which owns the cloud servers on which the data would be stored. The client is the owner of the data who wants to exploit the cloud storage service which provided by CSP. TPA is the entity which is responsible for providing the public data auditing service. The proposed auditing protocol has been designed to achieve a set of objectives which can be summarized in the following points:

1. **Public audibility:** A Third Party Auditor (TPA) is employed in order to verify the integrity of data stored on the cloud storage without putting an additional online burden of the clients.
2. **Storage Correctness:** TPA is able to figure out if there is any corrupted blocks in the stored data or not.
3. **Improving data availability:** this means enhancing the resistance against the denial of service (DOS) attacks or whatever attacks which can prevent authorized data access.
4. **Preserving data confidentiality:** this means maintain the secrecy of data whether during its residence on the cloud storage or during the auditing process performed by the TPA.
5. **Efficiency:** minimizing the time required whether for storing the data on the cloud or for performing the auditing process.

In the proposed framework, there are two modes for TPA operations: ***Semi-Trusted TPA*** and ***Fully-Trusted TPA***. In Semi-Trusted mode, the role of TPA is limited to performing the auditing process on demand on the behalf of the user. On the other side, in Fully Trusted mode, the TPA acts as intermediary between the client or the data owner and the cloud service provider. In other words, there is no direct interaction between the client and the cloud service provider.

As shown in Figure 3, in both modes the client is responsible for selecting the file that needs to be stored remotely on the cloud storage. Then, the file is divided into blocks of certain size. After that, the MD5 is applied on the blocks to generate 128 bits hash value for each block. Then, the hashes are concatenated into one string to be the input for the AES cipher algorithm. Also, it is the responsibility of the client to store the keys used in the encryption process locally. In order to store the data in the cloud storage, there are two scenarios. The first scenario occurs when the TPA works in Semi-Trusted mode; it is the responsibility of the client to upload the file into the cloud storage directly without TPA intervention. On the contrary, in the second scenario when the TPA works in Fully Trusted mode, the client directs the encrypted data to the TPA and selects the different cloud providers which will store the data then it is the responsibility of the TPA to upload the data into the cloud storage.

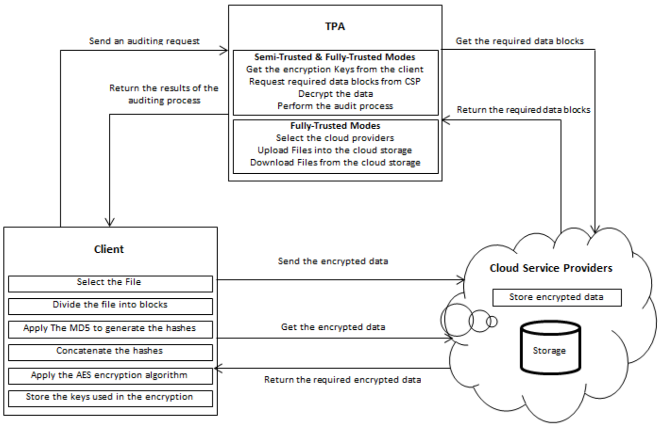
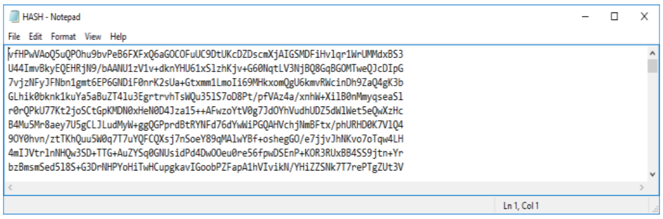

TPA
Send an auditing request
Get the required data blocks
Semi-Trusted & Fully-Trusted Modes
Get the encryption Keys from the client
Request required data blocks from CSP
Decrypt the data
Perform the audit process
Return the results of the auditing process
Return the required data blocks
Fully-Trusted Modes
Select the cloud providers
Upload Files into the cloud storage
Download Files from the cloud storage
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 3: The architecture of the proposed auditing mechanism

In the client side, the MD5 is used for two main purposes. The first purpose is to encrypt the data as MD5 is a well-known encryption algorithm but the essence of its use in the proposed method is to compress the data to minimize the encryption time for the AES algorithm as well as the time required to transfer the data for the client to the cloud service provider and vice versa. Figures 4a and 4b provide an example for using the MD5 algorithm.

(a)



```
vfHPwVAoQ5uQPOhu9bvPeB6FXFxQ6aGOCOFuUC9DtUKcDZDscmXjAIGSMDFiHvlqr1WrUMMdxBS3
U44ImvBkyEQEHRjN9/bAANU1zV1v+dknYHU61xSlzhKjv+G60NqtLV3NjBQ8GqBGOMTweQJcDIpG
7vjzNFyJFNbn1gmt6EP6GNDiF0nrK2sUa+Gtxmm1LmoIi69MHkxomQgU6kmvRWcinDh9ZaQ4gK3b
GLhik0bknk1kuYa5aBuZT4lu3EgrtrvhTsWQu351S7oD8Pt/pfVAz4a/xnhW+XilB0nMmyqseaSl
r0rQPkU77Kt2joSCtGpKMDN0xHeN0D4Jza15++AFwzoYtV0g7JdOYhVudhUDZ5dWlWet5eQwXzHc
B4Mu5Mr8aey7U5gCLJLudMyW+ggQGPprdBtRYNFd76dYwWiPGQAHVchjNmBFtx/phURHD0K7V1Q4
9OY0hvn/ztTKhQuu5W0q7T7uYQFCQXsj7nSoeY89qMAlwYBf+oshegGO/e7jjvJhNKvo7oTqw4LH
4mIJVtrlnNHQw3SD+TTG+AuZYSq0GNUsidPd4DwOOeu0re56fpwDSEnP+KOR3RUxBB4SS9jtn+Yr
bzBmsmSed518S+G3DrNHPYoHiTwHCupgkavIGoobPZFapA1hVIvikN/YHiZZSNk7T7rePTgZUt3V
```

(b)

Figure 4a is a jpg color image with a resolution of 280 * 147 and total size of 22.4 KB. **Figure 4b** is the concatenated hashes of the image shown in Figure 4a with size of 7 KB. AES has been selected as the main encryption algorithm in the proposed auditing mechanism because of its immunity against all known attacks, speed, and simplicity. The doubly encryption process is adopted to make the task of the intruder harder if he wants to read the content of the data and hence the confidentiality of the data is preserved.

After the completion of this doubly encryption process, the client stores the encryption keys locally on a private cloud and transfer the encrypted data to the selected cloud storages by himself if a Semi-trusted TPA is adopted or only directs the data to the TPA which transfers the encrypted to the selected storage on the behalf of the client if a Fully-trusted TPA is adopted. The flow chart for storing the data in the cloud storage is shown in Figure 5. In the proposed framework, the multi-cloud model is adopted where there are at least two clouds: private cloud in which the encryption keys are stored and a public cloud in which the encrypted data blocks are stored. Also, more than one public cloud (multi-cloud) can be used to store the encrypted data blocks to enhance the availability of data in case of service outage in a cloud service provider where the data can be accessible from the other cloud service provider. Once the encrypted data get stored on the cloud storage, the client can ask the TPA to perform an auditing process given the file id, cloud provider id, concatenated hashes, and the encryption key used for the AES algorithm.

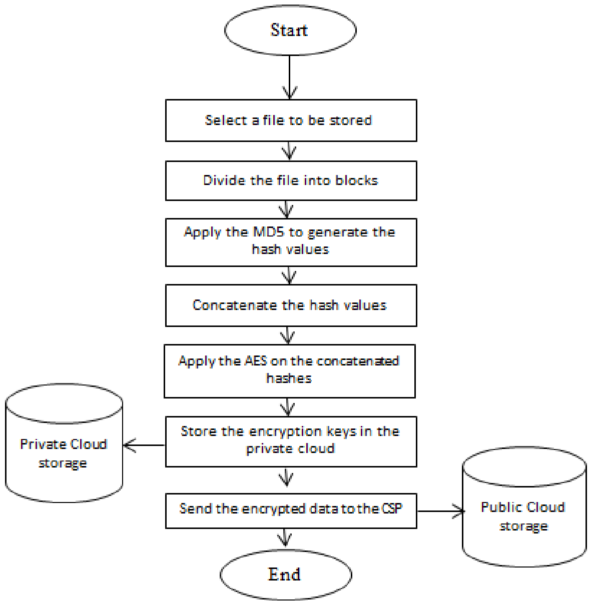

Start
Select a file to be stored
Divide the file into blocks
Apply the MD5 to generate the hash values
Concatenate the hash values
Apply the AES on the concatenated hashes
Store the encryption keys in the private cloud
Private Cloud storage
Send the encrypted data to the CSP
Public Cloud storage
End

Figure 5: Flow chart of the *store data* process

In a Semi-Trusted TPA, the sign up, login and audit processes are the only allowed processes for the clients. In order to use the TPA services, the client has to sign up and create an account then the login process must be executed successfully. After the success of the sing up and login process, the client is allowed to send an auditing request to the TPA. Upon a request from the client, the TPA asks the cloud provider to return the required data. After the arrival of the encrypted data from the cloud provider, the TPA decrypts it using the encryption key provided by the client then the hashes are compared and the results are returned to the client. It is obvious that the TPA does not need to know the content of the data to perform the auditing process as the comparison is performed between the hash values. The sequence diagram of the auditing process is shown in Figure 6.

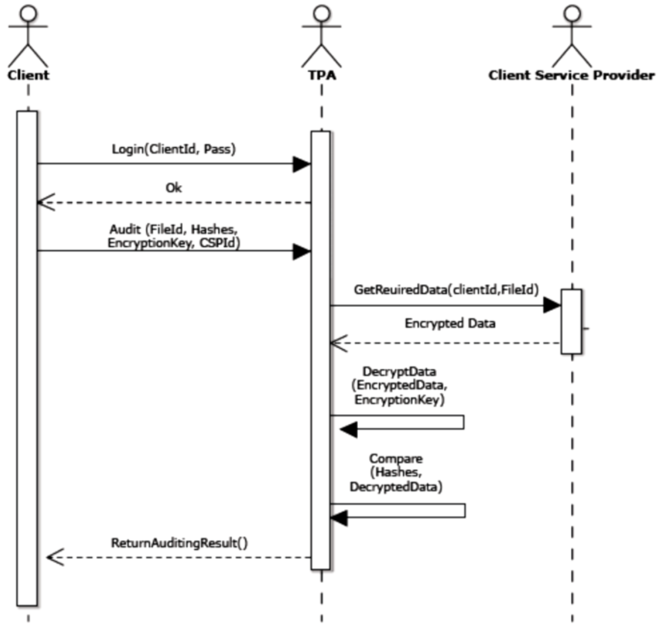

Client
TPA
Client Service Provider
Login(ClientId, Pass)
Ok
Audit (FileId, Hashes, EncryptionKey, CSPId)
GetReuiredData(clientId,FileId)
Encrypted Data
DecryptData (EncryptedData, EncryptionKey)
Compare (Hashes, DecryptedData)
ReturnAuditingResult()

Figure 6: Sequence diagram for the *audit data* process

In the Fully-Trusted TPA, the client is not forced to have a direct interaction with the CSP because the TPA is providing a GUI through which the user can select the intended CSPs to store the encrypted data. Also, through the GUI the client can ask the TPA to download the needed data from the cloud storage.

EXPERIMENTS AND RESULTS

In the proposed work, the TPA has been implemented as a web service available for use by every registered user after completing the sign up process and passing the login process using a web site designed and implemented for this purpose. In the proposed framework, the data owner can access the TPA through two ways. In the first way, the data owner or the client can access the TPA using any web browser such as Google chrome, Firefox, or Internet Explorer, etc. in the second way, the data owner or the client can access the TPA using the mobile application which we have developed in order to make him able to access the TPA easily through any android phone. The technologies used for building the proposed system includes: Java, JSP, Java servlet, HTML, Java Script, MySQL, and Jason for message exchange besides the Netbeans as the used integrated development environment. Also, the client environment which used in the experiments with operating system Windows 7 and with hardware characteristics as follow:

- Processor: Pentium iv 2.6 Ghz
- RAM: 512 mb dd ram

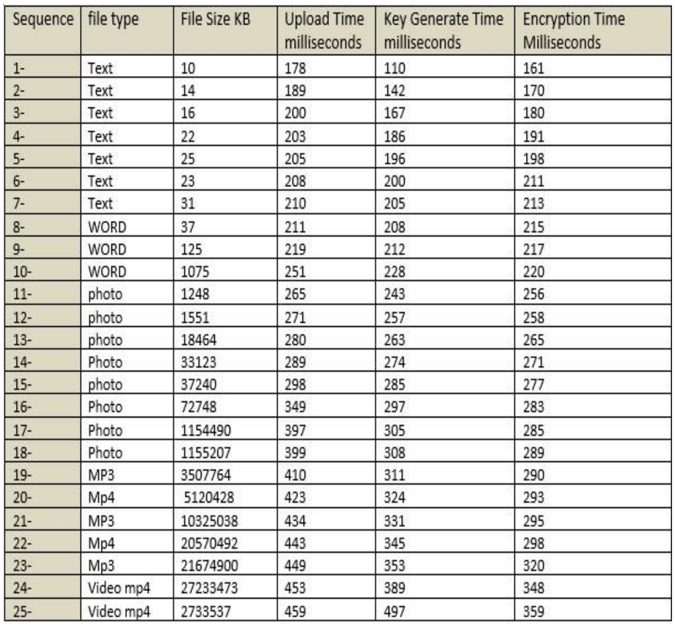
The dataset used in the following experiments consists of 150 different files. The types, numbers, and sources of the files are shown in detail in table 1. The files are of different sizes with total size up to 1.5 Gigabyte.

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

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

In this section, a set of experiments have been executed to evaluate the performance of the proposed mechanism. The first three experiments have been done on the data set described in Table2. As shown in Table 2, the data set consists of 25 files with different sizes and different types. The first three experiments are conducted to measure the upload time, key generation time, and encryption time sequentially.

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



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

As mentioned previously, the first experiment has been done to compute the time (in msec) needed to upload a set of files which have different sizes (in KB). Upload time is the time required to upload a specific file to the cloud storage. I.e. it is the difference between the start upload time and finish upload time. In addition to Table 2, the results of the first experiment are shown in Figure 7.

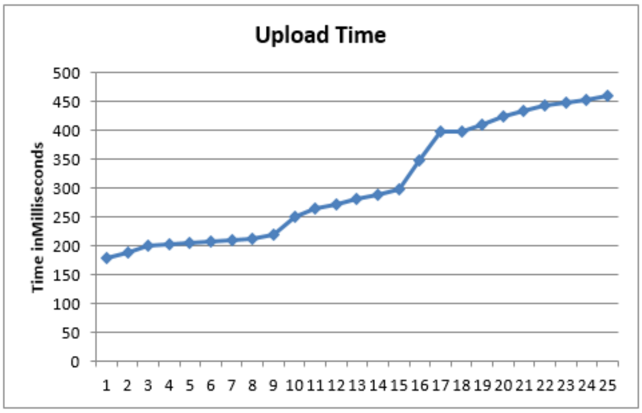


Figure 7: The upload time for a set of files with different sizes.

The second experiment has been done to compute the time required to generate the encryption keys for each file to be encrypted using the AES encryption algorithm. In addition to Table 2, the results of the second experiments are shown in Figure 8. The third experiment computes the time of the encryption for each file in the set and the results are shown in Figure 9. Both of the experiments are applied on 25 files from the data set.

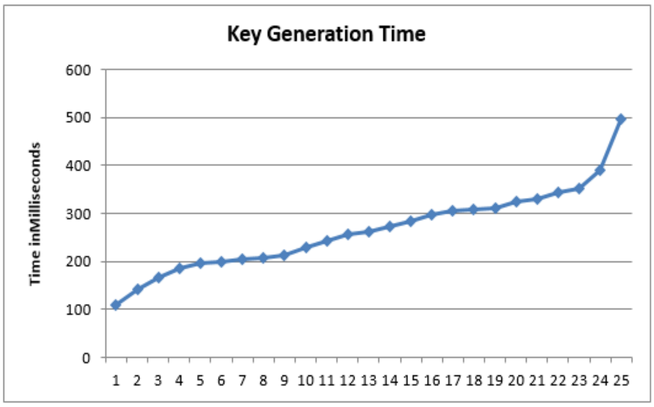


Figure 8: The time of generating the encryption keys for a set of files with different sizes

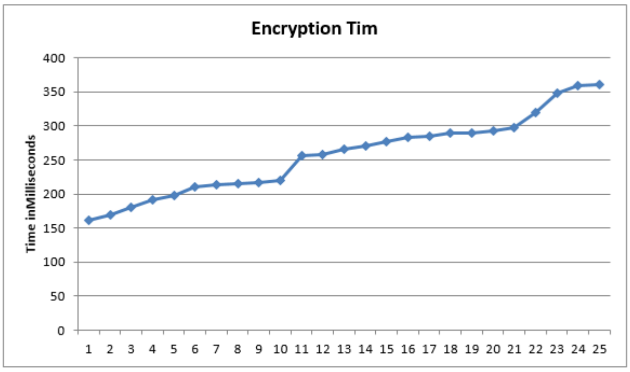


Figure 9: The time of encryption using the AES for a set of files with different sizes.

From the previous figures, it can be seen that the upload time, key generation time, and encryption time are directly proportional to the file size. The fourth experiment has been done to compare the proposed method with another public auditing scheme which mentioned in [7] in terms of the time required to perform the auditing process on a set of files with different sizes. The results in millisecond are shown in Table 3 and Figure 10. Generally, from the results we can see that the auditing time is directly proportional to the file size.

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

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

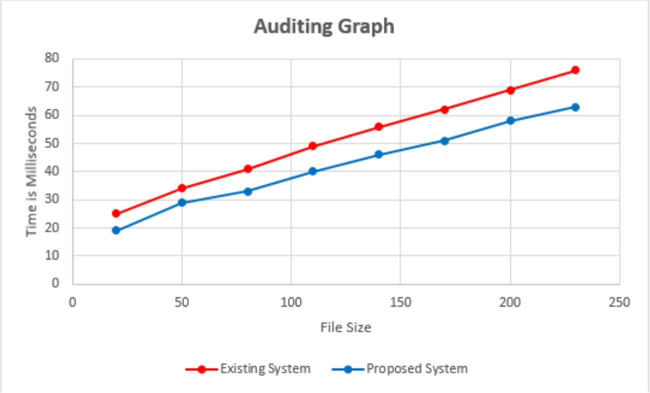


Figure 10: The time of auditing using the existing and the proposed methods for a set of files with different sizes.

Figure 10 shows that the proposed method is faster than the method mentioned in [7]. Therefore, the proposed auditing method is more efficient in terms of the required time for performing the auditing process.

The last experiment is conducted to measure the accuracy of the proposed TPA in terms of two criteria, namely false positive rate and false negative rate. In this experiment we have uploaded 150 file of different size with total size nearly 1.5 Gigabyte. In order to measure the accuracy of the TPA, the half of the files is modified while the other half remained without modifications. Then the TPA is asked to check the integrity of the whole set. The False Positive Rate (FPR) represents the proportion of false alarms rate in which the TPA raises an alarm to indicate a modified file wrongly. On the other side, the False Negative Rate (FNR) represents the proportion of false alarms rate in which the TPA raises an alarm to indicate to an unmodified file wrongly. Of course, the false positive rate is computed during the auditing process for the unmodified half of the files set while the false negative rate is used during the auditing process for the modified half of the files set. FPR and FNR are calculated according to equations 1and 2, respectively.

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

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

Where, FP is the number of false positives, TN is the number of true negatives and N=FP+TN is the total number of negatives, FN is the number of false negatives, and TP is the number of the number of true positives

Figure 11 shows the results of the auditing accuracy for the proposed method and for the method mentioned in [7]. Also, the results are shown in table 4. From the results, we can see that our method is more accurate in terms of FPR and FNR.

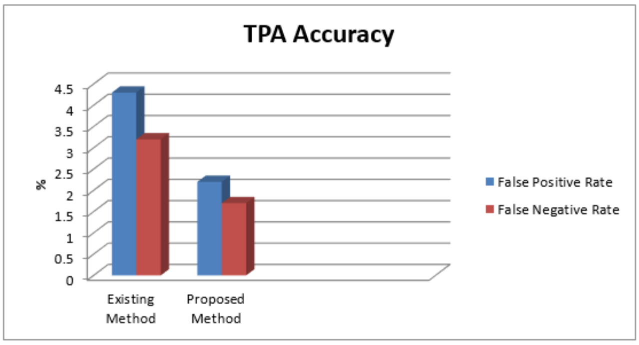


Figure 11: The auditing accuracy using the existing and the proposed methods for a set of files with different sizes.

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

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

From the previous results, we can see that the proposed public auditing mechanism is better than the work proposed in [7] in terms of the time required to perform the auditing process and in terms of the accuracy using FPR and FNR.

CONCLUSION AND FUTURE WORK

Among the services provided by the cloud storage service which allows the users to store, to manage remotely and to access their data over the internet. However, there is a set of challenges and research problems which came with the cloud storage service and need to be alleviated to increase the user's contentment for this service. One of these problems is the users' anxiety about the integrity, availability, and confidentiality of their data and their feelings that the data can be accessed or modified by external intruders because of the limited control granted to the users over the remote cloud nodes. Therefore, there is a need for a public auditing mechanism which able to conceal the justified fears of the users who have data stored on the cloud storage. In this paper, a public auditing protocol has been proposed with based on employing a third party for performing the auditing process on demand on behalf of the data owners while preserving the data privacy. The design objectives of the proposed protocol are public audibility, storage correctness, improving data availability, preserving data confidentiality, and efficiency. Public audibility and storage correctness are achieved by using a TPA and granting him the necessary privileges for performing the auditing process and discovering any corrupted or tampered data. Also, improving data availability is achieved by adopting the replication principle through employing multi-cloud model while preserving the data confidentiality is achieved by storing the data encrypted in cloud storage and performing the auditing process by the TPA without the need to know the content of data on the hash values. Finally, efficiency is achieved by compressing the data using the MD5 algorithm which minimizes the time required for the encryption by the AES algorithm and the time required for transferring the data between the client and CSP. A set of experiments have been done to evaluate the performance of our mechanism on a set of files with different sizes in terms of time and accuracy. The results show that the proposed TPA has 2.2% and 1.7% for the false positive rate and false negative rate respectively when used for verifying the integrity of a set of files which consists of 150 files with different sizes with total size nearly 1.5 Gigabyte. For the future work, the proposed the mechanism can be modified to allow the data operations dynamically.

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: privacy-preserving 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. Privacy-preserving 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. Privacy-preserving public auditing for secure cloud storage. IOSR Journal of Computer Engineering (IOSR-JCE), 16(4), pp.33-38.
11. Dang, H.V., Tran, T.S., Nguyen, D.T., Bui, T.V. and Nguyen, D.T., 2015. Efficient privacy preserving data audit in cloud. In Advanced Computational Methods for Knowledge Engineering (pp. 185-196). Springer International Publishing.
12. Kim, D., Kwon, H., Hahn, C. and Hur, J., 2015. Privacy-preserving public auditing for educational multimedia data in cloud computing. Multimedia Tools and Applications, pp.1-15.
13. Wang, B., Li, B. and Li, H., 2012, June. Knox: privacy-preserving auditing for shared data with large groups in the cloud. In International Conference on Applied Cryptography and Network Security (pp. 507-525). Springer Berlin Heidelberg.
14. Tan, S. and Jia, Y., 2014. NaEPASC: a novel and efficient public auditing scheme for cloud data. Journal of Zhejiang University SCIENCE C, 15(9), pp.794-804.
15. Singh, G., 2013. A study of encryption algorithms (RSA, DES, 3DES and AES) for information

security. International Journal of Computer Applications, 67(19).

16. https://en.wikipedia.org/wiki/Information_security, Last Access: Nov -2016
17. Forouzan, A.B., 2006. Data communications & networking (sie). Tata McGraw-Hill Education.
18. Bhanot, R. and Hans, R., 2015. A Review and Comparative Analysis of Various Encryption Algorithms. International Journal of Security and Its Applications, 9(4), pp.289-306.
19. Stallings, W., 2006. Cryptography and network security: principles and practices. Pearson Education India.
20. Thambiraja, E., Ramesh, G. and Umarani, D.R., 2012. A survey on various most common encryption techniques. International journal of advanced research in computer science and software engineering, 2(7).
21. Masram, R., Shahare, V., Abraham, J. and Moona, R., 2014. Analysis and comparison of symmetric key cryptographic algorithms based on various file features. International Journal of Network Security & Its Applications, 6(4), p.43.
22. Chan, X. and Liu, G., 2007. Discussion of One Improved Hash Algorithm Based on MD5 and SHA1. World Cngress on Engineering and Computer Science (WCECS), San Francisco, USA.
23. Roshdy, R., Fouad, M. and Aboul-Dahab, M., 2013. Design and Implementation of a new Security Hash algorithm based on MD5 and SHA-256. International Journal of Engineering Sciences and Emerging Technologies, pp.29-36.
24. https://en.wikipedia.org/wiki/MD5, Last Access: Nov -2016
25. Daemen, J. and Rijmen, V., 2013. The design of Rijndael: AES-the advanced encryption standard. Springer Science & Business Media.
26. Shrivastava, N. and Kumar, G., 2013. A survey on cost effective multi-cloud storage in cloud computing. Int. J. Adv. Res. Comput. Eng. Technol, 2, pp.1405-1409.
27. Kamble, S., Kumbhojkar, P., Gulgonda, S., Waghmare, T. and Wahul, R., 2016. Design and Implementation of Secured Multicloud Storage System. International Journal of Engineering Science, 4393.
28. AlZain, M.A., Pardede, E., Soh, B. and Thom, J.A., 2012, January. Cloud computing security: from single to multi-clouds. In System Science (HICSS), 2012 45th Hawaii International Conference on (pp. 5490-5499). IEEE.
29. Mohta, A. and Awasti, L.K., 2012. Cloud data security while using third party auditor. International Journal of Scientific & Engineering Research, 3(6), p.1.