Secure and Efficient Third-party Auditing Scheme for Cloud Storage

Ghassan Sabeeh Mahmood[1, *], Dong Jun Huang[2], Baidaa Abdulrahman Jaleel[3]
[1,2] School of Information Science and Engineering, Central South University
Changsha 410083, China
{ghassan.programer@gmail.com}
[1,3]Computer Science Department, College of Science
University of Diyala, Iraq

ABSTRACT: *In the cloud environment, pay-per-use is used to provide services for the end-users. With all the benefits available, a threat is present in the security of sensitive data. Users of the cloud cannot rely on cloud service providers for the security of data. In this work we present a model of cloud system that supports privacy-preserving and public auditing. The design introduced now has three components such as data owner, cloud server, and third-party auditor (TPA). In this proposed model the data owner first encrypts the data using the Advanced Encryption Standard (AES) algorithm and applies a Steganography technique by combining Discrete Wavelet Transform (DWT) and Singular Value Decomposition (SVD) in order to hide the encrypted data and subsequently save them on a cloud server. In the next phase, the SHA-2 value is developed for the hidden file and creates a signature on it. The Third Party Auditor then next authorize the data onto the cloud instead of data owners by generating the value of the SHA-2 and create a signature on it, eventually, compares both signatures to verify the data integrity. We thus intend to produce an efficient and secure system that is capable of privacy-preserving public auditing maintaining the data confidentiality and integrity. In the experimentation process, it is found that the security and performance have recorded a good level of security and found to me more efficient.*

Keywords: Cloud Computing, Privacy-preserving Public Auditing, Data Integrity, Data Confidentiality, Third-party Auditor

Received: 10 February 2020, Revised 19 May 2020, Accepted 3 June 2020

DOI: 10.6025/jisr/2020/11/3/67-74

1. Introduction

Cloud computing is an emerging computing model that enables users to remotely store their data onto a cloud, as well as to enjoy services on demand. The process of migrating data from the user side to the cloud computing offers considerable convenience to users because they can access data anytime and anywhere in the cloud computing. Moreover, they can use any device without the need for capital investment to deploy hardware infrastructures. Small- and medium-sized enterprises with a limited budget can particularly save costs, as well as achieve the flexibility to scale investments on- demand, by using cloud

services, thereby enabling them to achieve projects and enterprise-wide contacts and schedules [1].Cloud computing allowing network access to a set of resources based on-demand (e.g., storage, networks, servers, services, and applications). Cloud has five important features, three service models, and four deployment models. The important features are self-service on-demand, location-independent resource combining, wide network access, fast resource elasticity, and measured service. And the three service models include platform as a service (PAAS), software as a service (SAAS), and infrastructure as a service (IAAS). And the deployment models are public cloud, private cloud, hybrid cloud, and community cloud [2].

The capability to store data in the cloud offers to users the suitability of access without needing direct knowledge of the management of the infrastructure or hardware. Although the cloud is significantly more powerful than personal computing, this paradigm provides new privacy and security challenges as well. As users renounce control for their data, they no longer have for such data physical possession.

Complete access to cloud services exposes users' data to a variety of threats and malicious attacks, as well as frequent cases of security. Although the cloud is economically good-looking to clients and enterprises due to offers of data sharing, the privacy and data security of users are guaranteed [3]. Therefore, should be considered the security of the stored data in the cloud, integrity, privacy, and confidentiality and are significant requirements from the user's perspective. Accordingly, new methods should be established to realize these requirements.

To ensure data integrity should be enabled public auditing to compel users to resort to an independent (TPA) to audit data when necessary. The abilities of a TPA will enable him to periodically assess the integrity of data stored in the cloud computing instead of the users. Thus, TPA provides users with an easy and affordable method that ensures the correctness of their storage in the cloud [4]. Furthermore, the auditor requires a strong Cryptographic hash function to assess the integrity of the cloud data. For example, the SHA-2 family of hash functions, which was developed by the National Institute of Standards and Technology (NIST) [5].

As well as, to ensure data confidentiality the cryptography algorithms are a good solution for the stored data on the cloud computing in addition to Steganography techniques. The more secure approaches in Steganography are used that are the Discrete Wavelet Transform (DWT) with Singular Value Decomposition (SVD) to gather and make its robustness against numerous attacks. DWT is a frequency domain technique in which the cover image is initially altered into the frequency domain. Thereafter, frequency coefficients are altered according to the transformed coefficients of the secret file. Subsequently, the result file is obtained, which is substantially robust [6]. One of the most significant analytical tools in linear algebra is SVD and is particularly used for the analysis of matrices. SVD is related to the theory of diagonal a symmetric matrix: a matrix can be decomposed into three sub-matrices, namely, U, S, and VT. U and V are orthogonal square matrices, while S is a rectangular diagonal matrix [7].

The current paper presents a novel and secure cloud storage system that ensures high-level information confidentiality, availability, and integrity. Moreover, the proposed scheme aims to protect information from providers of cloud, TPA, and unauthorized users. The remainder of the paper is organized as follows. Section 2 introduces the problem statement. The related works indicate in section 3. Section 4 provides the complete description of our proposed. Section 5 establishes the implementation and evaluation of our scheme. Lastly, Section 6 provides the conclusions of this paper.

2. Problem Statement

2.1 System and Threat Model

The proposed scheme includes three different entities with well-defined interactions among one another (see figure 1).

- A cloud server is owned by Cloud Service Providers (CSP) and has the infrastructure and skill to host outsourced storage, as well as offers effective mechanisms for its users to create, store and update.
- User (client) has the data that be stored in the cloud.
- TPA is trusted entity to measure the authenticity of the cloud storage instead of users when necessary.

Users can store data on the cloud computing to free themselves from the burden of maintenance and storage. Similar to [8], we assume that CSP is semi-trusted, that is, it follows the normal movement of the protocol in the scheme. However, the CSP

may not be trusted with the real data contents and its integrity, that is, it could behave unfaithfully toward others in terms of their data.

Therefore, integrity mechanisms are required for the storage and maintenance of users' data. Users can mandate a trusted TPA to achieve security tasks because handling such undertakings on their own is not economically practicable for them.

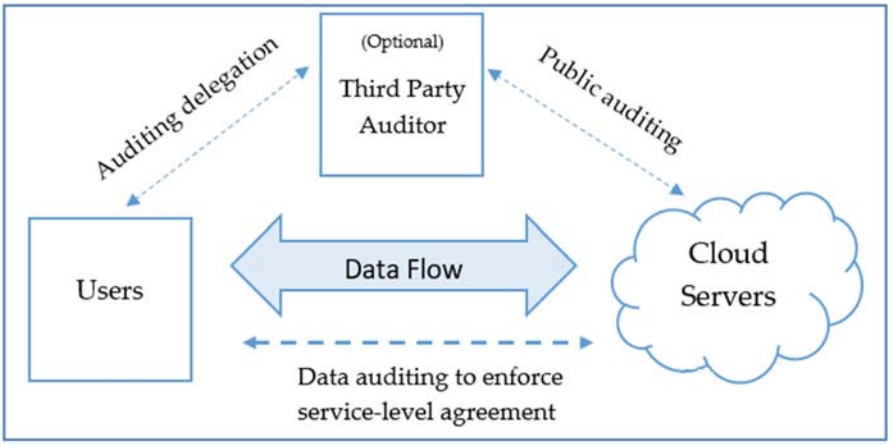

Figure 1. Cloud storage service architecture

2.2 Design Goals

To allow public auditing for data integrity under the proposed model, we designed a scheme to achieve the following goals:

• **Public Auditing:** Enables TPA to authenticate the correctness of information based demand without introducing extra online load to the users of cloud or without retrieving the complete information.

• **Privacy Preserving:** Ensures that TPA during the auditing process cannot obtain the data privacy of users.

• **Data integrity and Confidentiality:** Develops a secure and efficient scheme that can achieve the integrity and confidentiality of data.

• **Efficiency:** The preceding goals should be achieved with low computation, storage, and communication costs.

3. Related Work

Cong Wang et al. Proposed distributed system using dynamic data support. In order to provide redundancy vectors and ensure the reliability of data, they rely on erasure correction code in the file distribution planning. The aforementioned researchers also support third-party auditing, where users can carefully mandate the integrity-checking tasks to TPAs and be worry-free in using the services of the cloud [9]. Loheswaran et al. proposeda renaissance system. It is a semi-trusted proxy agent to return the data blocks during the period of reparation. The reposed renaissance system model is implemented. The results have compared the proposed method with security and without security level. Accordingly, traffic load has reduced, whereas the average net profit, response rate, and utilization have improved [10].

Qian Wang et al. discovered the problem of simultaneous public auditing and data dynamic for data integrity check on the cloud. Their construction is designed to meet these two significant goals while ensuring efficiency. To realize data dynamics, the aforementioned researchers enhanced the current storage models by using the classic Merkle Hash Tree structure for block tag verification. To support the handling of multiple auditing tasks, they additionally explored the method of bilinear aggregate signature to extend their main result into a multi-user setting, where TPA can simultaneously perform many auditing tasks [11].

Solomon et al. proposeda scheme that solved the problem of data integrity by maintaining its privacy using the blinding method. This technique is a public auditing. For improved efficiency, this system provides a privilege for TPA to concurrently

handle auditing from multiple users with dissimilar messages. They improved the efficiency of their system by minimizing computationally intensive operations, such as bilinear mapping and by entirely avoiding such computations from the user side. Through a detailed security and performance analysis, the system is proven secure and efficient. Even though this system is not completely dynamic. This system is not completely dynamic since data block insertion makes it considerably inefficient [8].

The existing systems achieved privacy-preservation public auditing, however they unsuccessful to preserve the data confidentiality. Therefore, a secure system should be developed to effectively perform public auditing by maintaining confidentiality along with integrity of information.

4. Proposed Scheme

To improve privacy and security of data storing in the cloud, we propose a scheme that comprises three components namely, owner of data, TPA, and cloud server.

The owner of data is a significant entity of our proposed scheme because it implements most of the responsibilities associated with the information. At first the owner of the data login to the server and TPA. The new user must initially register by filling in a login form and become an active user of the scheme. A message that reaffirms the successful registration will be provided. The user can perform a login process if becomes a member of the system that's mean if his name and password appear in the database, then he will login successfully as a valid user. Figure 2 shows the architecture of the proposed model.

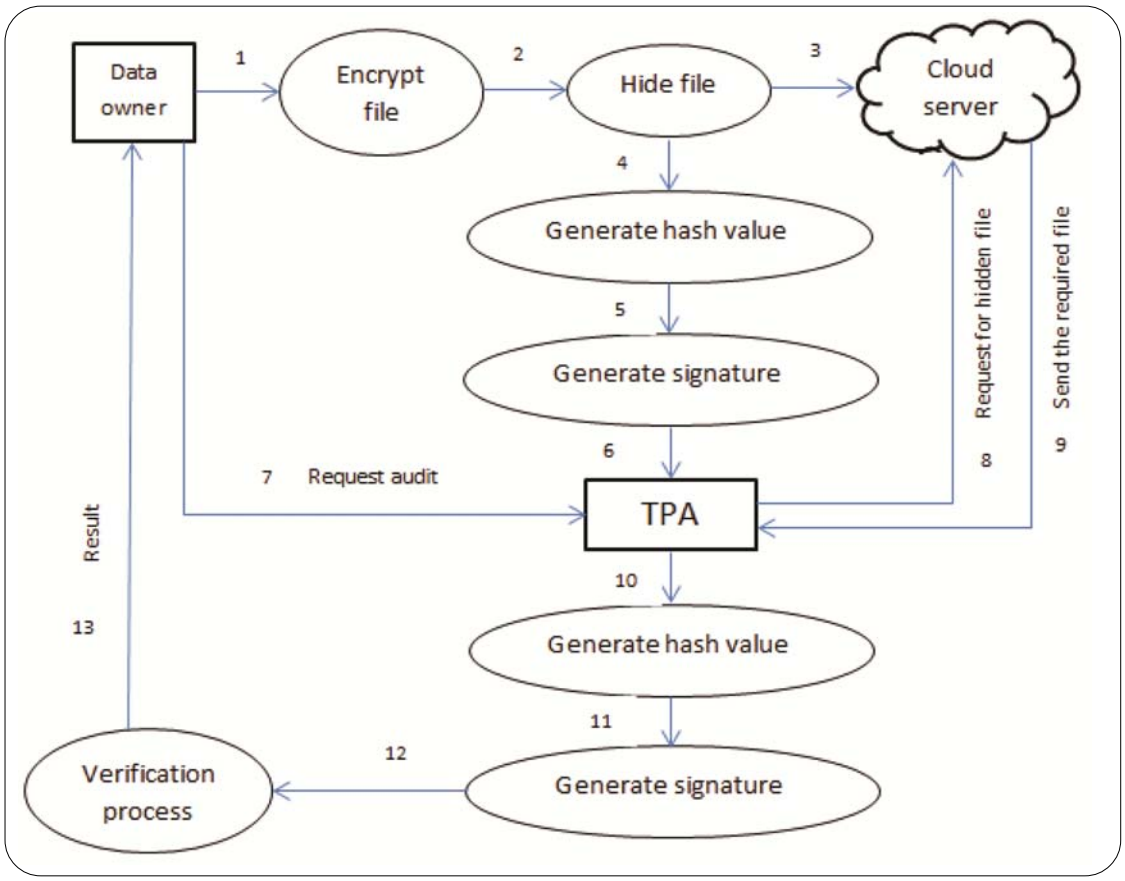

Figure 2. The architecture of the proposed model

4.1 Maintaining Confidentiality of Data

If the login is successful, the owner of data selects the file he wants to store in the cloud. Selected file will be encrypted

using the AES algorithm to provide confidentiality of the data. Subsequently, the encrypted file is embedded in the cover file using a steganography technique by combining DWT and SVD. The algorithms work for embedding and extraction as follows:

Embedding Algorithm

Step 1: The cover and encrypted file are transformed into sub-bands using DWT.

Step 2: SVD is performed on the HH sub-band of the decomposed cover and encrypted file.

Step 3: After performing SVD on the cover and encrypted file, the resultant encrypted file is embedded with the cover.

Step 4: Performed the inverse (SVD) to the embedded file.

Step 5: Lastly, performed the inverse (DWT) to make the stego file.

Extraction Algorithm

Step 1: The cover and encrypted file are transformed into sub-bands using DWT.

Step 2: SVD is performed on the HH sub-band of the decomposed cover and encrypted file.

Step 3: The stego file is transformed into sub-bands using DWT.

Step 4: SVD is performed on the HH sub-band of the decomposed stego file.

Step 5: Thereafter, the extraction is applied to the resultant SVD file.

Step 6: Inverse SVD is applied on the resultant file after extraction.

Step 7: Lastly, inverse DWT is performed to obtain the extracted encrypted file.

4.2 Maintaining the Integrity of Data

To achieve the data integrity, we use a hash algorithm and signature. By creating a hash value for the hidden file. Nobody can obtain two different input values that result in the same hash output, thereby suggesting that the hash functions are collision-resistant. After generating the hash function in the hidden file, we generate a signature on it. Eventually, send this signature to the TPA, which uses to evaluate the integrity of the data (whether it is maintained). The data owner has the authority to request a TPA for a data integrity check.

The data owner utilizes the cloud to store the hidden form of the information. Given that the data are stored in hidden form, the cloud server consequently has zero knowledge regarding the data. Moreover, if the cloud server converts to a malicious server or is attacked by any outside attacker, the data will not be retrieved easily because it is in the hidden form.

4.3 Supporting Privacy-preserving Public Auditing

Our proposed system achieves the privacy-preserving public auditing for computing using TPA, who ensures the auditing without recovering the information copy. Hence, privacy is preserved.

In the proposed scheme, TPA performs auditing of data either occasionally or on request by the client. Upon getting, the request of auditing from the user or owner of data the TPA starts data auditing process.TPA stores the signature, which has been generated by the data owner. And TPA follows the same procedure achieved by the owner of data, such as generating a hash value for a hidden file and generating a signature on it. Thereafter, it compares the two signatures during the verification process. If the signatures match, then the data integrity is maintained otherwise, it is not maintained. This result means that data has not been tampered with or changed. Finally, the TPA provides the results to the data owner.

5. System Implementation and Evaluation

5.1 System Implementation

The proposed system is implemented using Matlab. In the proposed system, the data owner is responsible for encrypting the file using the AES algorithm, hiding it using DWT–SVD, generating an SHA-2 value for the hidden file, and creatinga signature for it. The cloud server stores the hidden file. When the data owner requests a TPA for auditing of data, The TPA immediately requests the hidden file from the cloud server. Afterward getting the data, TPA generates the SHA-2 value for the hidden file. And is usage the same SHA-2 algorithm that was used by the data owner. Later the TPA creates a signature on that

file. In the verification process, the signature generated by TPA and the other one stored in the TPA (submitted by the owner of data) are compared, if the signatures match, then the data are intact, and any outsider or attacker is not tampering with the data. If the signatures do not match, then the data integrity has tampered. The result of check the data integrity submits to the owner of data.

5.2 System Evaluation

The proposed system is developed to enhance the level of information confidentiality, availability, and integrity. We evaluated our system by considering the different file sizes. Table 1 highlights the response time of the Cryptographic performance in terms of encryption and decryption on different sizes.

Size (KB)	Response time (s)	
	Encryption performance	Decryption performance
32	0.2112	0.2223
64	0.4766	0.5143

Table 1. Cryptographic performance

The preceding experiment indicated that the Cryptographic performance is extremely rapid, depending on the response time of encryption and decryption. Moreover, Steganography is applied to add security and confidentiality in our model. Tables 2 and 3 show the experiments. Table 2 shows the performance of embedding and extracting in terms of the mean square error (*MSE*), peak signal-to-noise ratio (*PSNR*), and normalized correlation (*NC*) against various attacks (Salt-paper [density 0.01], Gaussian noise $m = 0$, $v = 0.001$, Poisson Noise, Compression Q.F.60%, Rotation by 10 [Clockwise], and Shifting Attack Translation [5 5]). MSE and PSNR are calculated between the original and stego files. However, *NC* is measured between the original and extracted files.

Size	Parameter	Without attack	Salt-paper (density 0.01)	Gaussian noise $m = 0$, $v = 0.001$	Poisson Noise	Compression Q.F.60%	Rotation by 10 (Clock -wise)	Shifting Attack Trans- lation [5 5]
64 KB	MSE	0.9604	1.9902	3.8339	4.0478	1.8989	68.9448	15.2577
	PSNR	59.3203	55.1418	49.1910	48.5654	55.3458	38.639	44.1069
	Retrieved Secret file NC	0.1917	0.9881	0.9948	0.9928	0.9950	0.2733	0.9010

Table 2. Extraction of secret file

Size	Embedding time in unit seconds	Extraction time in unit seconds
32 KB	0.987032	1.295450
64 KB	1.190910	1.542602

Table 3. Embedding and extracting time in unit seconds

Our model shows a strong anti-interference performance and high stability when facing various malicious attacks. While table 3 shows the efficiency in terms of computation time for embedding and extracting in unit seconds.

The proposed system provides the wanted data integrity guarantee with minimal computation and communication overhead. We mainly focus on the overhead incurred by privacy preservation. Figure 2 presents the outputs of the proposed scheme of auditing of data with respect to the size of the files.

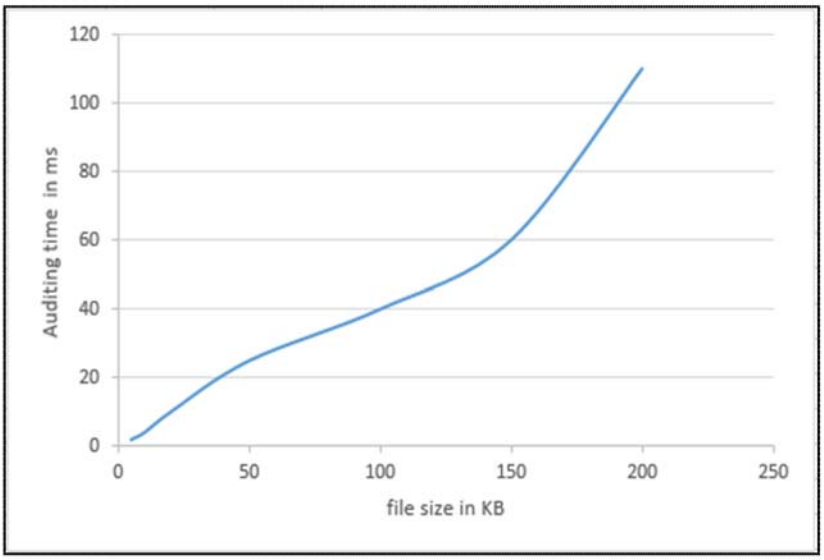

Figure 2. Auditing time with respect to the size of the files

6. Conclusion

This paper discusses the privacy-preserving public auditing. A secure cloud computing storage system is proposed to develop a secure and efficient system that is capable of privacy preserving, public auditing, and maintaining the integrity and confidentiality of data. The data are encrypted and then stored in the hidden formatting in the cloud storage, thereby maintaining the data confidentiality. The integrity of data is verified by TPA upon client request through signature verification. TPA verifies whether the stored data is tamper or not, and informs the result to the client. Reveal the results the efficiency and effectiveness of the proposed scheme when auditing integrity of data. The proposed scheme can be extended to achieve data dynamics for privacy-preserving public auditing.

References

[1] Wang, G., Liu, Q., Wu, J., Guo, M. (2011). Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers. *Computers& security*, 30 (5) 320-331.

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

[3] Dong, X., Yu, J., Luo, Y., Chen, Y., Xue, G., Li, M. (2014). Achieving an effective, scalable and privacy-preserving data sharing service in cloud computing. *Computers & Security*, 42, 151-164.

[4] Wang, C., Chow, S. S., Wang, Q., Ren, K., Lou, W. (2013). Privacy-preserving public auditing for secure cloud storage. *IEEE transactions on computers*, 62 (2) 362-375.

[5] Glabb, R., Imbert, L., Jullien, G., Tisserand, A., Veyrat-Charvillon, N. (2007). Multi-mode operator for SHA-2 hash functions. *journal of systems architecture*, 53 (2) 127-138.

[6] Narula, N., Sethi, D., Bhattacharya, P. P. (2015). Comparative Analysis of DWT and DWT-SVD Watermarking Techniques in RGB Images. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 8 (4) 339-348.

[7] Subhedar, M. S., Mankar, V. H. (2014 November). High Capacity Image Steganography based on Discrete Wavelet Transform and Singular Value Decomposition. *In*: Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies (p. 63). ACM.

[8] Worku, S. G., Xu, C., Zhao, J., He, X. (2014). Secure and efficient privacy-preserving public auditing scheme for cloud storage. *Computers & Electrical Engineering*, 40 (5) 1703-1713.

[9] Wang, C., Wang, Q., Ren, K., Cao, N., Lou, W. (2012). Toward secure and dependable storage services in cloud computing. *IEEE transactions on Services Computing*, 5 (2) 220-232.

[10] Loheswaran, K., Premalatha, J. (2016). Renaissance System Model Improving Security and Third Party Auditing in Cloud Computing. *Wireless Personal Communications*, 90 (2) 1051-1066.

[11] Wang, Q., Wang, C., Ren, K., Lou, W., Li, J. (2011). Enabling public auditability and data dynamics for storage security in cloud computing. *IEEE transactions on parallel and distributed systems*, 22 (5) 847-859.