Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2019) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

Third International Conference on Computing and Network Communications (CoCoNet'19)

SDVADC: Secure Deduplication and Virtual Auditing of Data in Cloud

Geeta C M[a,*], Shreyas Raju R G[a], Raghavendra S[a], Rajkumar Buyya[b], Venugopal K R[c], S S Iyengar[d], L M Patnaik[e]

[a]*Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University,Bengaluru*
[b]*Cloud Computing and Distributed Systems (CLOUDS) Lab, The University of Melbourne, Australia*
[c]*Vice Chancellor, Bangalore University, Bengaluru*
[d]*Department of Computer Science and Engineering, Florida International University, USA*
[e]*INSA, National Institute of Advanced Studies, Indian Institute of Science Campus, Bangalore, India*

Abstract

Over the last few years, deploying data to cloud service for repository is an appealing passion that avoids efforts on significant information sustenance and administration. In distributed repository utilities, deduplication technique is often exploited to minimize the capacity and bandwidth necesseties of amenities by erasing repetitive data and caching only a solitary duplicate of them. Proof-of-Ownership mechanism authorize any possessor of the identical information to approve to the distributed repository server that he possess the information in a dynamic way. In repository utilities with enormous information, the repository servers may intend to minimize the capacity of cached information, and the customers may want to examine the integrity of their information with a reasonable cost. We propose Secure Deduplication and Virtual Auditing of Data in Cloud ($SDVADC$) mechanism that realizes integrity auditing and deduplication of information in cloud. The mechanism supports secure deduplication of information and effective virtual auditing of the documents during the download process. In addition, the proposed mechanism lowers the burden of dataowner to audit documents by himself and there is no need to delegate auditing to the Third Party Auditor (TPA). Experimental results demonstrate that the virtual auditing has low auditing time cost relative to the existing public auditing schemes.

Keywords: Cloud Computing; Deduplication; Virtual Auditing; Public Auditing; Proof of Retrievability; Proof of Ownership.

* Corresponding author. Tel.: +91-948-108-2405 ; fax: +0-000-000-0000.
E-mail address: geetacmara@gmail.com (Geeta C M).

E-mail address: geetacmara@gmail.com

1. Introduction

Distributed computing delivers extensible, inexpensive, and locality-independent online facilities extending from simple backup facilities to distributed repository frameworks. Presently, optical networks [16], [17] have been deployed all over the globe for efficient information communication. The rapid expansion of information capacity stockpiled in the distributed repository has motivated to an expanded need for methods for conserving disk capacity and network bandwidth. To minimize resource utilization, various distributed repository utilities, such as Dropbox [2], Google Drive [3], make use of deduplication procedure, where the distributed repository saves one solitary duplicate of repetitious information and furnishes links to the document rather than of saving other genuine copies of that information, irrespective of the number of customers request to save the information.

In order to secure their personal information from unapproved external attackers and from the Cloud Service Provider (CSP), customers encode their documents before deploying to the distant server. While, traditional encryption cannot be used to carry out deduplication due to the following reasons; Deduplication method takes benefit of information equivalence to recognize the identical information and decrease the repository capacity. On the contrary, the cryptographic algorithms shuffle the encoded documents in order to make ciphertext equivalent from hypothetically arbitrary information. Encryption of identical documents by distinct customers with distinct encode keys outcomes in distinct ciphertexts, that renders it critical for the distributed server to decide if the plaintext are identical and deduplicate them.

Concurrent encryption [5] solves this issue successfully. The concurrent encryption algorithm encodes an input record with the hash value of the input document as an encode key. The ciphertext is transmitted to the server and the customer keeps the encode key. As the concurrent encode procedure is imperative, alike records are encoded into equivalent ciphertext, despite of who encodes them. As a result, the distributed repository server can carry out deduplication over the ciphertext and every proprietors of the document be able to retrieve the ciphertext and decode it subsequently as they possess the identical encode key for the document.

When customers utilize cloud repository facilities, the integrity of the stockpiled information is the most important requirement. Hence, it is an essential prerequisite of customers to regularly audit the present state of their information. Clients can download all the information for its integrity confirmation also called as private auditing, however it is not a reasonable arrangement in light of the fact that the I/O and communication cost over the system is extremely expensive. To guarantee the information uprightness and safeguard the cloud clients reckoning gadgets, it is necessary to empower public reviewing administration for cloud information repository, so that clients may trust public verifier to review the deployed information (public auditing) when it is required. The TPA has ability and capacities that clients do not, can intermittently verify the honesty of all the information gathered in the cloud. In the recent past, various remote integrity verifying conventions were recommended to permit the TPA to examine the information uprightness on the distant server. Few of the existing public auditing mechanisms are as follows:

A public reviewing framework [19] is designed for the reliability of transferred information with adequate client repudiation. The public examiner audits the sincerity of the deployed information without fetching the complete information from the cloud. The limitation is that the scheme is not collusion resistant. Yang *et al.,* [24] have designed a framework for public examining for collaborative information in distributed repository that accomplishes identity secrecy and trackability. The mechanism achieves data privacy by utilizing blind signature method. The limitation is that the mechanism incurs little overhead to accomplish the identity trackability. From the above discussion it is observed that the dataowner can examine the correctness of the deployed information also called as private auditing or the dataowner hires the TPA to examine the integrity of information (public auditing) deployed in the cloud.

There is an urgent need to design efficient auditing schemes that reduces the burden of the dataowner to examine the integrity of the data without TPA. Already, innumerable mechanisms have been advocated containing proof of retrievability and confirmable information possession mechanisms. Protected deduplication and integrity verification are the fundamental functions needed in distributed repository utilities. However, nearly few studies [25] have been carried out for implementing combined mechanisms that can bolster these two functions simultaneously. In this paper, we suggest Secure Deduplication and Virtual Auditing of Data in Cloud ($SDVADC$) mechanism that supports virtual auditing and secure deduplication of encrypted data.

1.1. Motivation

Currently, we observe a noticeable growth in the size of information cached in repository utilities, including considerable development of networking methods. In repository utilities with very large information, the repository servers aim to decrease the capacity of stockpiled information, and the customers desires to check the honesty of their information with a reasonable cost. Motivated by this factor, we propose a new Secure Deduplication and Virtual Auditing of Data in Cloud ($SDVADC$) mechanism that bolsters safe and effective auditing of the data owner's documents by the virtual auditing entity and secure deduplication of encrypted information. In our proposed scheme, the information proprietor has been relieved from the responsibility of auditing the documents by himself and there is no necessity to transfer the auditing job to the TPA. Further, the scheme provides efficient data deduplication over encrypted information.

1.2. Contributions

In this paper, we present a new Secure Deduplication and Virtual Auditing of Data in Cloud ($SDVADC$) mechanism that supports secure and effective virtual auditing and deduplication of encrypted information. Particularly, our contributions can be summarized as follows:

(i) Secure deduplication of encrypted information.

(ii) Efficient auditing of data owners' file virtually by the Virtual Auditing Entity (VAE) during download process and hence eliminating the Third Party Auditor (TPA).

1.3. Organisation

The paper is formulated as follows: Related works that give the pros and cons on existing integrity auditing and deduplication schemes are outlined in Section 2. Background work that gives earlier models and their drawbacks are discussed in Section 3. Preliminaries used in the scheme are discussed in Section 4. Problem statement and System framework illustrates the working of the system are discussed in Section 5. In Section 6, scheme details of Secure Deduplication and Virtual Auditing of Data in Cloud ($SDVADC$) has been proposed. Security analysis is performed in Section 7. Experimental results are analyzed in Section 8. Conclusions are presented in Section 9.

2. Related Works

A brief survey of both integrity auditing and secure deduplication, is presented in this section.

2.1. Integrity Auditing

Zhu *et al.,* [27] developed the collaborative PDP in collective distributed repository and Proof of Retrievability (PoR) [10] mechanisms that supports integrity verification. In contrast with PDP, PoR not only satisfies the distributed servers retain the destined documents, but also affirms their complete reconstruction. Wang *et al.,* [20] suggested an identification-based information deploying scheme with extensive verification in clouds. The mechanism permits delegated intermediary to process and deploy the document in favor of the document proprietor. Both the document creation and document integrity can be examined by the TPA. The limitation is that the time cost at the auditor side is more.

Shen *et al.,* [11] proposed a public verifying mechanism with an innovative compelling framework. The scheme supports universal and sampling blockless authentication as well as cluster auditing. The disadvantage of the mechanism is that the transmission cost is more in verify phase. Jin and Zhou [8] proposed a public reviewing convention with public provability, adequate information dynamics and candid controversy negotiation. The limitation is that the

scheme introduces overhead for dynamic update and controversy negotiation. Simulations are carried out in C++ [15]. Hequn *et al.,* [9] introduced a public reviewing scheme for collaborative information utilizing backups with customer repudiation in the cloud. The advantages of the scheme is that it efficiently recovers the documents that can resist the conspiracy assault among the cloud and repudiated clients. The limitation is that the scheme takes more time for re-signing of the blocks.

Shen *et al.,* [12] designed a distant information integrity examining procedure that deals with information distribution with delicate information concealing for cloud storage. The mechanism utilizes identification-based cryptography, that streamline the sophisticated certificate administration. Limitation of the scheme is that the TPA has more computation overhead. Tang *et al.,* [13] proposed an adequate real-time integrity verification method with privacy-conserving agreement for images in distributed repository framework. The advantages of the scheme is that it achieves replay assault protection and privacy-conserving legitimate agreement. Geeta *et al.,* [6] have presented extensive review on the latest methods in information auditing and security in cloud computing.

2.2. Secure Deduplication

Data deduplication is a distinct information confining approach for deleting identical documents of repetitious information in repository. The method is utilized to increase repository usage and decrease bandwidth utilization.

Wu *et al.,* [21] proposed a discerned stockpiling mechanism to allow primary repository deduplication in clouds. The scheme achieves inflated inline backup adeptness and decreased the deduplication workload. The scheme has more computation overhead. Zheng *et al.,* [26] outlined and achieved an encoded cloud media center accomodating encoded Scalable Video Coding (SVC) videos. The advantage of the scheme is that it enables safe deduplication and preserves the video privacy. It is immune to the attackers instigating brute-force attacks over predictable videos. The disadvantage is that it incurs little storage overhead. Venugopal *et al.,* [18] use soft computing methods for data mining cloud applications.

Yan *et al.,* [23] suggested a miscellaneous information repository administration mechanism. The advantages of the scheme is that it supports data privacy and identity privacy. The limitation is that the scheme takes more time to calculate hash code set of a file. Xiong *et al.,* [22] proposed an original secure role re-encode framework that is constructed utilizing concurrent encode technique and the role re-encode function to avoid the information exposure in the cloud. The scheme realizes the dynamic updation and repudiation. The limitation is that the reckoning cost of creating document label is more.

3. Background Work

Hur *et al.,* [7] designed an original server-side deduplication mechanism for encoded data. It permits the cloud server to supervise access to the outsourced data while the proprietorship transforms effectively by using an arbitrary concurrent encryption and reliable proprietorship cluster key administration. This avoids data leakage to repudiated customers and distributed repository server. The disadvantage of the mechanism is that it has added computational overhead.

4. PRELIMINARIES AND DEFINITION

The preliminaries and definitions used in this paper are described below:

4.1. Randomized Convergent Encryption

Bellare *et al.,* [4] presents a randomized convergent encode mechanism. In randomized convergent encryption, an original customer encodes the document and constructs $C_1 \leftarrow \mathcal{E}(\mathcal{L}, \mathcal{M})$, where $\mathcal{L}$ is an arbitrarily selected key, and then encodes the document encode key $\mathcal{L}$ and constructs $C_2 \leftarrow \mathcal{L} \oplus \mathcal{K}$, where $\mathcal{K}$ is a Key-Encoding Key (KEK) that is obtained from the document $\mathcal{K} \leftarrow \mathcal{H}(\mathcal{M})$. Further, the document label $\mathcal{T}$ is constructed from the KEK. When any

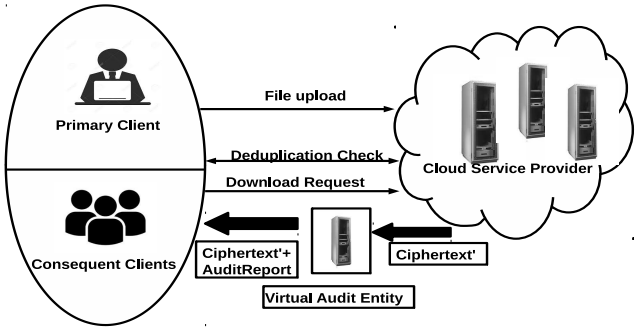

Fig. 1: Data Deduplication with Virtual Auditing System Model

authorized proprietor accepts $C_1, C_2, \mathcal{T}$ from the server subsequently, he estimates $\mathcal{L} \leftarrow C_2 \oplus \mathcal{K}$, decodes C_1 with $\mathcal{L}$, and acquires $\mathcal{M}$. Next, he constructs a label $\mathcal{T}' \leftarrow \mathcal{H}(\mathcal{P}, \mathcal{H}(\mathcal{P}), \mathcal{M})$ and verifies if $\mathcal{T}' = \mathcal{T}$. If $\mathcal{T}' = \mathcal{T}$, he receives it; otherwise, declines it, as the information is impaired.

The concurrent encryption algorithm encodes an input record with the hash value of the input document as an encode key. The ciphertext is transmitted to the server and the customer keeps the encode key. As the concurrent encode procedure is imperative, alike records are encoded into equivalent ciphertext, despite of who encodes them. As a result, the distributed repository server can carry out deduplication over the ciphertext and every proprietors of the document be able to retrieve the ciphertext and decode it subsequently as they possess the identical encode key for the document. Still, the concurrent encryption undergoes security defects with respect to label consistency and proprietorship repudiation. Hence, in the proposed scheme we have utilized the randomized convergent encode mechanism.

5. Problem Definition and System Model

5.1. Problem Definition

Given that the primary information proprietor uploads the document to the cloud, the main objectives are:

i Secure deduplication of encrypted data.

ii Efficient auditing of the dataowners file virtually by the Virtual Auditing Entity (VAE) during download process and hence eliminating the TPA.

5.2. System Model

We propose Secure Deduplication and Virtual Auditing of Data in Cloud ($SDVADC$) mechanism. The distributed repository framework (as shown in Fig. 1.) comprises of three objects:

i *Information proprietor*: The information proprietor is a customer who possess information, and outsources it to the distributed repository to save costs. The information proprietor encodes the information using randomized convergent encryption [7], [4] and deploys it to the cloud repository with its label information. Original uploader is an information proprietor who deploys the document that do not formerly exists in the distributed repository.

Other proprietors might have uploaded the same document earlier, they are called as consequent uploader. Proprietorship cluster is a set of information proprietors who distribute the identical information in the distributed repository.

ii *Cloud Service Provider*: This is an object that offers distributed repository utilities. It comprises of a distributed server and distributed repository. The distributed server deduplicates the deployed information from customers if required and saves the deduplicated information in the distributed repository. The distributed server stores proprietorship lists for deployed information, that consists of label for the cached information and the $id's$ of its proprietors. The distributed server manages access to the stockpiled information. Upon each membership change in the proprietorship list (dynamic proprietorship administration), CSP permits access to the equivalent information to the proprietors only for the time interval within which the proprietors preserve valid proprietorship of the information by re-encoding it utilizing an updated proprietorship cluster key and selectively allocating it. Since the distributed server is a semi-trusted entity, it should be prevented from retrieving the plaintext of the encoded information.

iii *Integrity Auditing by the Virtual Auditing Entity (VAE)*: This is an inbuilt auditing framework constructed by the dataowner. Every time when the dataowner uploads the document, automatically the metadata information of the document is sent to the virtual auditing framework i.e., the Virtual Auditing Entity (VAE) that consists of the metadata information of all the documents uploaded to the distributed server. When the customer sends the download request for his document, the CSP sends the requested document to the VAE. The virtual auditing framework performs auditing of this document and sends the document attached with the auditing report to the information proprietor. The information proprietor receives the requested encrypted document along with auditing report. Hence, in the proposed scheme, the information proprietor has been relieved from the burden of verifying the document and there is also no need for the information proprietor to hire the TPA.

6. The Algorithm

The algorithm has two functions: (i) File uploading (ii) Virtual Auditing

i *File uploading:* Consider an information proprietor encrypts the file F_1 using the randomized convergent encryption [7] and outsource the ciphertext to the distributed server [See Algorithm 1, Phase I]. The CSP accepts the encoded record, and checks for the deduplication. If the file F_1 is an original file, then the CSP saves the file in the server. This uploader is called as a primary client. If the file is a duplicate, then the CSP runs PoW convention with the information proprietor. When information proprietor proves that he is an authorized person, then the CSP provides a link for the file existing in its server. Next, the uploader is added to the consequent clients group. If any one of the client wants to update or delete their respective files, the client sends this request to the CSP. The CSP recomputes the $proprietorshipclusterkey$ and re-encrypts the ciphertext. Now the CSP revokes this client. Next, the CSP shares the updated $proprietorshipclusterkey$ to all the genuine clients in the cluster.

ii *Virtual Auditing:* Information proprietor transmits a *document* request query to the CSP. Upon accepting the query request for the document from the information proprietor, CSP sends *re-encrypted ciphertext* and the tag to the VAE. The VAE possesses the metadata information of the information cached in the distributed server and the public keys of the customers. The VAE performs auditing, if $(Tagofthedocument)' \neq Tagofthedocument$, then VAE transmits auditing report that the document has been modified else the document is correct. VAE can audit efficiently any number of documents at the same time. Then the VAE sends the $ciphertext'$ appended with the auditing report VAR to the dataowner. The dataowner receives the $ciphertext'$ appended with the auditing report VAR from the VAE. Next, the information proprietor decrypts the $ciphertext'$ and also possess the Virtual Auditing Report (VAR) [See Algorithm 1, Phase II].

Algorithm 1: *SDVADC*: Secure Deduplication and Virtual Auditing of Data in Cloud

Input: F_1
Output: $ciphertext'$, VAR
Phase 1: Document Level Deduplication
For every outsourcing document F_1 by information proprietor the following tasks are implemented:
Information proprietor encrypts the file F_1 and outsource the ciphertext to the cloud server.
Upon acquiring the ciphertext of the file, the CSP checks for the deduplication.
If F_1 is the original file, then the CSP saves the file in the server. The uploader is called as a primary client.
Otherwise, the CSP executes PoW convention with the information proprietor. When information proprietor proves that he is an authorized person, then the CSP provides a link for the file existing in its server. The uploader is added to the consequent clients group.
Phase II: Virtual Auditing:
Information proprietor transmits a *document* request query to the CSP.
CSP sends *re-encrypted ciphertext* of the requested document to the VAE. The VAE possesses the metadata information of the data cached in the distributed server and the public keys of the customers.
The VAE performs auditing, if $(Tagofthedocument)' \neq Tagofthedocument$, then VAE transmits auditing report that the document has been modified else the document is correct.
VAE can audit efficiently any number of documents at the same time.
VAE sends the $ciphertext'$ appended with the auditing report VAR to the information proprietor.
The information proprietor receives the $ciphertext'$ appended with the auditing report VAR from the VAE.
Next, the information proprietor decrypts the $ciphertext'$ and also possess the VAR.

7. Security Analysis

In this section, we analyze the security of the proposed mechanism.

7.1. Information Integrity

In the proposed mechanism, the integrity of the outsourced information is securely and efficiently verified by the VAE. The information proprietor constructs and deploys the VAE, such that the VAE securely and effectively audits the integrity of the documents during the download process. When the information proprietor sends a document request query $(\mathcal{T}_i, ID_j)$ to the CSP, then the CSP sends a result C_i' to the inbuilt framework, the Virtual Auditing Entity (VAE). The VAE possesses the metadata information and public keys of the clients, and performs auditing, if $(\mathcal{T}_i' \neq \mathcal{T}_i)$, then VAE sends auditing report that the document has been modified else the document is correct. VAE sends the ciphertext C_i' appended with the auditing report $\{C_i', VAR\}$ to the data owner c_t. Thus the information integrity has been achieved where the information proprietor's worrying factor about the correctness of their outsourced documents has been eliminated and there is no need for the information proprietor to hire a Third Party Auditor (TPA).

8. Performance Analysis

We present the experimental analysis of our mechanism in this section. We have used Intel(R) Core(TM) $i5$-5200U, CPU @2.20GHz, 8GB RAM. Every cryptographic operation was implemented utilizing the Crypto++ library ver. 5.6.2 [1] on a 3.4 GHz processor PC. The key criteria were identified to provide a 128-bit security level. The implementation utilizes an $MD5$ as a cryptographic hash function to create a 128-bit key and label, and an AES with Electronic Code Book mode as an encode/decode function. In this simulation, we set the size of the document as 10MB. To accomplish a 128-bit security level, we set S_K = 128 bits, S_T = 128 bits. Let n be the number of customers in the framework and u be the number of proprietors in the proprietorship list for the document.

Table 1: Comparison of secure deduplication schemes

Scheme	Encrypted deduplication	Tag consistency	Ownership management	Virtual auditing
CE [5]	Yes	No	No	No
RCE [4]	Yes	Yes	No	No
Hur scheme [7]	Yes	Yes	Yes	No
SDVADC	Yes	Yes	Yes	Yes

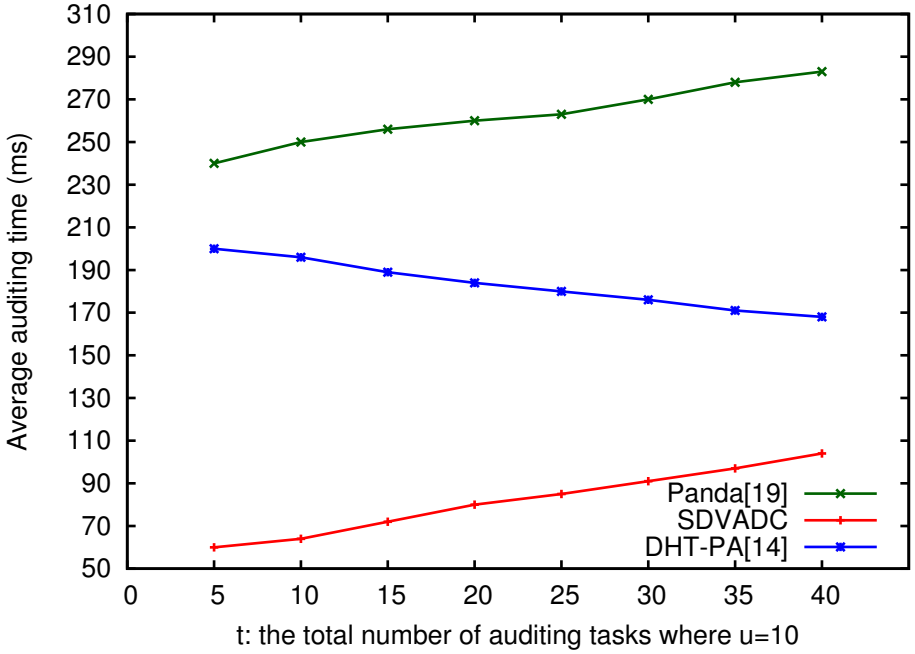

Fig. 2: Comparison of virtual auditing with batch auditing performed by *TPA*

8.1. Comparison of secure deduplication schemes

Table 1 shows the comparison results of the secure data deduplication mechanisms, that is Convergent Encryption (*CE*) [5], Randomized Convergent Encryption (*RCE*) [4], *Hur* scheme [7] and *SDVADC* schemes with regard to the information deduplication over encrypted data, label consistency, dynamic proprietorship management and virtual auditing. As each of the mechanisms permit information proprietors to encrypt their documents and allow deduplication over them, they can assure the information privacy against the distributed server and unapproved external attackers. Regarding information integrity, convergent encryption is not able to assure the integrity of deduplicated information whereas the other mechanisms conserve it by using an auxiliary procedure that allows the information proprietors to validate the label consistency of the accepted information.

In the *Hur* scheme [7] and *SDVADC* upon each membership change in the proprietorship list access to the equivalent information is allowed to proprietors only for the time interval within which the proprietors preserve valid proprietorship of the information by re-encoding it utilizing an updated proprietorship cluster key and selectively allocating it. This resolves the dynamic proprietorship management issue as opposed to the other mechanisms. The proposed mechanism supports virtual auditing while other schemes does not support virtual auditing.

Fig. 2, shows comparison of virtual auditing with batch auditing performed by *TPA*. When a number of auditing requests arrives in a very short period, our scheme supports batch auditing where the *VAE* performs auditing of the

number of auditing requests simultaneously and transmits the auditing report to the information proprietors. When a number of dataowners requests for the file download, CSP sends the respective ciphertext to the VAE. Since, the VAE is an inbuilt framework by the information proprietor, VAE efficiently and securely performs auditing of all the requested files simultaneously. Hence, as illustrated in Fig. 2, the time taken by virtual auditing is less compared to the existing public auditing mechanisms [19] [14]. Thus, in our proposed scheme the information proprietor is totally relieved from the worrying factor of the sincerity of their outsourced information and relieved from the burden of hiring TPA to audit the sincerity of their information.

9. Conclusions

In this study, we introduce Secure Deduplication and Virtual Auditing of Data in Cloud ($SDVADC$) mechanism. The proposed mechanism presents efficient data deduplication of encoded information and supports efficient auditing of dataowners file virtually by the Virtual Auditing Entity (VAE) during download process. The proposed mechanism has an inbuilt Virtual Auditing Entity (VAE) deployed by the information proprietor, that performs secure and efficient auditing of the requested documents efficiently during the download process. The information proprietor has been relieved from the burden of auditing the document and also there is no need for the information proprietor to hire the TPA. The performance analysis results demonstrate that in the proposed mechanism the virtual auditing is more efficient compared to the public auditing.

References

[1] , . Crypto++ library 5.6.2, http://www.cryptopp.com/.

[2] , . Dropbox, http://www.dropbox.com/.

[3] , . Google drive, http://drive.google.com.

[4] Bellare, M., Keelveedhi, S., Ristenpart, T., 2013. Message-Locked Encryption and Secure Deduplication, in: Annual international conference on the theory and applications of cryptographic techniques, Springer. pp. 296–312.

[5] Douceur, J.R., Adya, A., Bolosky, W.J., Simon, P., Theimer, M., 2002. Reclaiming Space from Duplicate Files in a Serverless Distributed File System, in: Proceedings 22nd International Conference on Distributed Computing Systems, IEEE. pp. 617–624.

[6] Geeta, C.M., Raghavendra, S., Buyya, R., Venugopal, K.R., Iyengar, S.S., Patnaik, L.M., 2018. Data Auditing and Security in Cloud Computing: Issues, Challenges and Future Directions. International Journal of Computer (IJC) 28, 8–57.

[7] Hur, J., Koo, D., Shin, Y., Kang, K., 2016. Secure Data Deduplication with Dynamic Ownership Management in Cloud Storage. IEEE Transactions on Knowledge and Data Engineering 28, 3113–3125.

[8] Jin, H., Jiang, H., Zhou, K., 2018. Dynamic and Public Auditing with Fair Arbitration for Cloud Data. IEEE Transactions on Cloud Computing 6, 680–693.

[9] Liu, H., Wang, B., Lu, K., Gao, Z., Zhan, Y., 2018. Public Auditing for Shared Data Utilizing Backups with User Revocation in the Cloud. Wuhan University Journal of Natural Sciences 23, 129–138.

[10] Shacham, H., Waters, B., 2013. Compact Proofs of Retrievability. Journal of Cryptology 26, 442–483.

[11] Shen, J., Shen, J., Chen, X., Huang, X., Susilo, W., 2017. An Efficient Public Auditing Protocol with Novel Dynamic Structure for Cloud Data. IEEE Transactions on Information Forensics and Security 12, 2402–2415.

[12] Shen, W., Qin, J., Yu, J., Hao, R., Hu, J., 2018. Enabling Identity-based Integrity Auditing and Data Sharing with Sensitive Information Hiding for Secure Cloud Storage. IEEE Transactions on Information Forensics and Security 14, 331–346.

[13] Tang, X., Huang, Y., Chang, C.C., Zhou, L., 2019. Efficient Real-Time Integrity Auditing with Privacy-Preserving Arbitration for Images in Cloud Storage System. IEEE Access 7, 33009–33023.

[14] Tian, H., Chen, Y., Chang, C.C., Jiang, H., Huang, Y., Chen, Y., Liu, J., . Dynamic-Hash-Table Based Public Auditing for Secure Cloud Storage. IEEE Transactions on Services Computing 10, 701–714, 2017.

[15] Venugopal, K.R., Buyya, R., Tata McGraw-Hill Education, 2013. Mastering C++.

[16] Venugopal, K.R., Rajan, E.E., Kumar, P.S., 1998. Performance Analysis of Wavelength Converters in WDM Wavelength Routed Optical Networks, in: Proceedings. Fifth International Conference on High Performance Computing (Cat. No. 98EX238), IEEE. pp. 239–246.

[17] Venugopal, K.R., Rajan, E.E., Kumar, P.S., 1999. Impact of Wavelength Converters in Wavelength Routed All-Optical Networks. Computer Communications 22, 244–257.

[18] Venugopal, K.R., Srinivasa, K.G., Patnaik, L.M., Springer, 2009. Soft Computing for Data Mining Applications.

[19] Wang, B., Li, B., Li, H., 2015. Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud. IEEE Transactions on Services Computing, 8, 92–106.

[20] Wang, Y., Wu, Q., Qin, B., Shi, W., Deng, R.H., Hu, J., 2017. Identity-Based Data Outsourcing with Comprehensive Auditing in Clouds. IEEE Transactions on Information Forensics and Security 12, 940–952.

[21] Wu, H., Wang, C., Fu, Y., Sakr, S., Lu, K., Zhu, L., 2018. A Differentiated Caching Mechanism to Enable Primary Storage Deduplication in Clouds. IEEE Transactions on Parallel and Distributed Systems 29, 1202–1216.

[22] Xiong, J., Zhang, Y., Tang, S., Liu, X., Yao, Z., 2019. Secure Encrypted Data with Authorized Deduplication in Cloud. IEEE Access 7, 75090–75104.

[23] Yan, Z., Zhang, L., Ding, W., Zheng, Q., 2017. Heterogeneous Data Storage Management with Deduplication in Cloud Computing. IEEE Transactions on Big Data .

[24] Yang, G., Yu, J., Shen, W., Su, Q., Fu, Z., Hao, R., . Enabling Public Auditing for Shared Data in Cloud Storage Supporting Identity Privacy and Traceability. Journal of Systems and Software 113, 130–139, 2016.

[25] Youn, T.Y., Chang, K.Y., Rhee, K.H., Shin, S.U., 2018. Efficient Client-Side Deduplication of Encrypted Data with Public Auditing in Cloud Storage. IEEE Access 6, 26578–26587.

[26] Zheng, Y., Yuan, X., Wang, X., Jiang, J., Wang, C., Gui, X., 2017. Toward Encrypted Cloud Media Center with Secure Deduplication. IEEE Transactions on Multimedia 19, 251–265.

[27] Zhu, Y., Hu, H., Ahn, G.J., Yu, M., 2012. Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage. IEEE Transactions on Parallel and Distributed Systems 23, 2231–2244.