Privacy protection domain-user integra tag deduplication in cloud data server

Mohanaprakash Thottipalayam Andavan[1], Nirmalrani Vairaperumal[2]
[1]Department of Computer Science and Engineering, School of Computing, Sathyabama Institute of Science and Technology, Chennai, India
[2]Department of Information Technology, School of Computing, Sathyabama Institute of Science and Technology, Chennai, India

Article Info

Article history:

Received May 15, 2021
Revised Mar 30, 2022
Accepted Apr 18, 2022

Keywords:

Cloud computing
Data deduplication
Data protection
Encryption
Security

ABSTRACT

The cloud with strong storage management has recently developed in the big data world which can confirm the data integrity and keep just a single data duplicate. Many cloud auditing storage techniques have been developed to overcome the data deduplication (DD) problem, but they are vulnerable and can't resist brute force attacks (BFA). There is some privacy leakage problem that occurred in the present method. In this article, an original strategy called domain-user integra tag (DUIT) has been presented which comprises inter and intra deduplication with file tag and symmetric encryption key. The DUIT has two phases, the first one is random tag generation for Intra deduplication and the other is random ciphertext (CT) generation for encryption. The benefit of the DUIT is the security of individual user's files would not reveal to people in general, hence we proved that the DUIT is protected from the BFA. Finally, an experiment has conducted in Linux processor and C program software. The outcome of DUIT demonstrates that our method has reduced the computation cost (CC) by 27% and 35% and searching complexity (SC) by 10% and 26% related with the previous methods. It is decided that the DUIT achieves the low CC and SC.

Corresponding Author:

Mohanaprakash Thottipalayam Andavan
Research Scholar, Department of Computer Science and Engineering, Sathyabama Insitute of Science and Technology
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600 119, Tamilnadu, India
Email: tmohanaprakash@gmail.com

1. INTRODUCTION

Recently, cloud computing has achieved great features in the cloud architecture field with the service of cloud storage (CS) systems which have been broadly acknowledged by people and enterprises. It is necessary to do data deduplication (DD), where a single duplicate of data is stored and duplicate copies are rejected. For that, users outsourcing encryption methods have developed and they might not want to expose their sensitive data to the cloud or different parties. To encode the data, concurrent encryption (CE) plan is projected to recognize the novel data [1]–[4]. The use of proxy is achieving server, or a standalone device, [5]. One of the protection techniques is Data splitting in which the complex data is allocated into segmented data and stored in various locations [6]. Fragmentation occurs and produces fragment information, for instance, a single fragment or re-identify the concept to whom it compares nor reveals the private data. In [7], [8]–[10], the data is fully stored in a cloud server hence, the user loses their right to control the data package and privacy leakages. Existing privacy techniques are completely founded on the encoded information, but this may be the deficiency of the unique privacy information and its loss the control of resisting the attacks.

In [11], three layer-based storage schemes were introduced with the guide of fog computing. In [12] introduced a group of vehicles that are topographically near to one another to make a virtual cloud (VC) safely, namelessly, and powerfully. The utilization of VC is to present a secure resource allocation with a secured client to convey a message. The cloud-based street condition monitoring scheme was introduced to screen the road conditions with the guide of a cloud server [13], [14]. Assuming a section involves the upsides of an attribute, then clearly fair knowing a rundown of analysis is futile to an interloper since he cannot relate them with the comparing subjects [15]–[18]. In [19] projected an investigation of different perspectives in distributed computing security dangers. This work gives a wide assortment of choices and moment organization of chose administrations. A convergent encryption calculation scrambles information with a main deterministically resulting from the data. To diminish the DD decentralization issue, [20] introduced, which provides decentralization of blockchain and no need to get the authority from a third authority, and encryption DD using a convergent key (CE). In [21] proposed an outfit dynamic enhancement-based converse versatile heuristic pundit. The proposed stratergy becomes seen from expert observation and gives a surmised arrangement when different work processes show up online at different window time (WT). The user keeps the similar file until the similar ciphertext (CT) storage and CE message encryption key (EK) management (MLE) to expand the privacy protection (PP). Distributed DD systems presented with high dependability accomplishing more security over information [22]–[24]. To confine the side channel data trickle, [25] designed a role symmetric encryption-proof of ownership (PoW) scheme. In [26] proposed a strong key-exposure strong auditing for protected CS which gives the thorough plan. In this plan, a reviewer is permitted to check the honesty of cloud information without downloading the whole information from the cloud. Data protection has been achieved for the local proxy which is a coherent substance that can be found on the client side. To confirm the honesty of data stored in the cloud, many CS auditing plans [27], [28] were proposed. One private key generator (PKG) was proposed to confirm the self and produce m for all clients, and one third party auditor (TPA) is utilized by clients to check the honesty of cloud data [29]. This methodology is unfortunate for huge scope clients since the PKG and the TPA probably will not have the choice to manage the heavy workload. Later, the knowledge of integrating linear error revising codes and linear homomorphic validation schemes combinedly projected to certify the safety [30]. This integration utilized just a single extra block to accomplish error tolerance and validation simultaneously. In this work, brute force attacks (BFA) and how to oppose the BFA and acknowledge DD has been researched with strong security insurance in CS auditing.

2. RESEARCH METHOD

The domain-user integra tag (DUIT) has three phases as data upload (DU), integra deduplication (ID), and data download which are displayed in Figure 1. The major aim of the DUIT is PP and DD between different domains. To figure it out DD with strong PP, a novel DUIT method was proposed to produce the ω1, and employ a new strategy to produce the key for file encryption. In the first phase, the trusted key data structure (KDS) generates Ksym (m) for clients, a public key (n) for the cloud server provider (CSP), and the corresponding system public parameters (PS). The symbol and notation in the work have been listed in Table 1.

2.1. Data upload

For DU into the cloud, intra tag generation, inter and intra DD have been achieved in the upload procedure. For DD, each client from the distinct domain (D=1, 2, 3, …, n) needs to generate a distinct Intra tag. The DU phase mainly includes four parts: Intra tag generation, Intra and Inter DD, and data encryption/key recovery. For each client U in Di, where i=1, 2, . . ., n, when U needs to upload the data length of file (FS), the user first generates an Intra-tag for DD. Then, the agent Ai performs the Intra-deduplication to prove the duplicate in the identical domain Di. If the duplicate does not happen, then the CSP needs to further conduct the inter-deduplication among different domains. Finally, if the duplicate is found, then U recovers the CE made by the first uploader.

The initial client directs a file FS “upload|| ($Length_{FS}$, α_{d_i}) to the agent. The $Length_{FS}$ denotes the $Length_{FS}$. Check whether a duplicated copy of the file (FS) exists by comparing α_{d_i} to the previously stored tag value from Di. When an initial client U$_1$ from D$_i$ needs to upload file FS, U$_1$ chooses a random number $r_m \in W_P$ and generates a random Intra-file-tag α with the EK$_{\mathrm{sym}}$ and K$_{\mathrm{sym}}$ (pk=K$_{sym}$, MAC$_{sym}$).

$$\alpha_{\mathrm{d}_i} = \left(d_i^{hh_1} g^{r_m}\right) \tag{1}$$

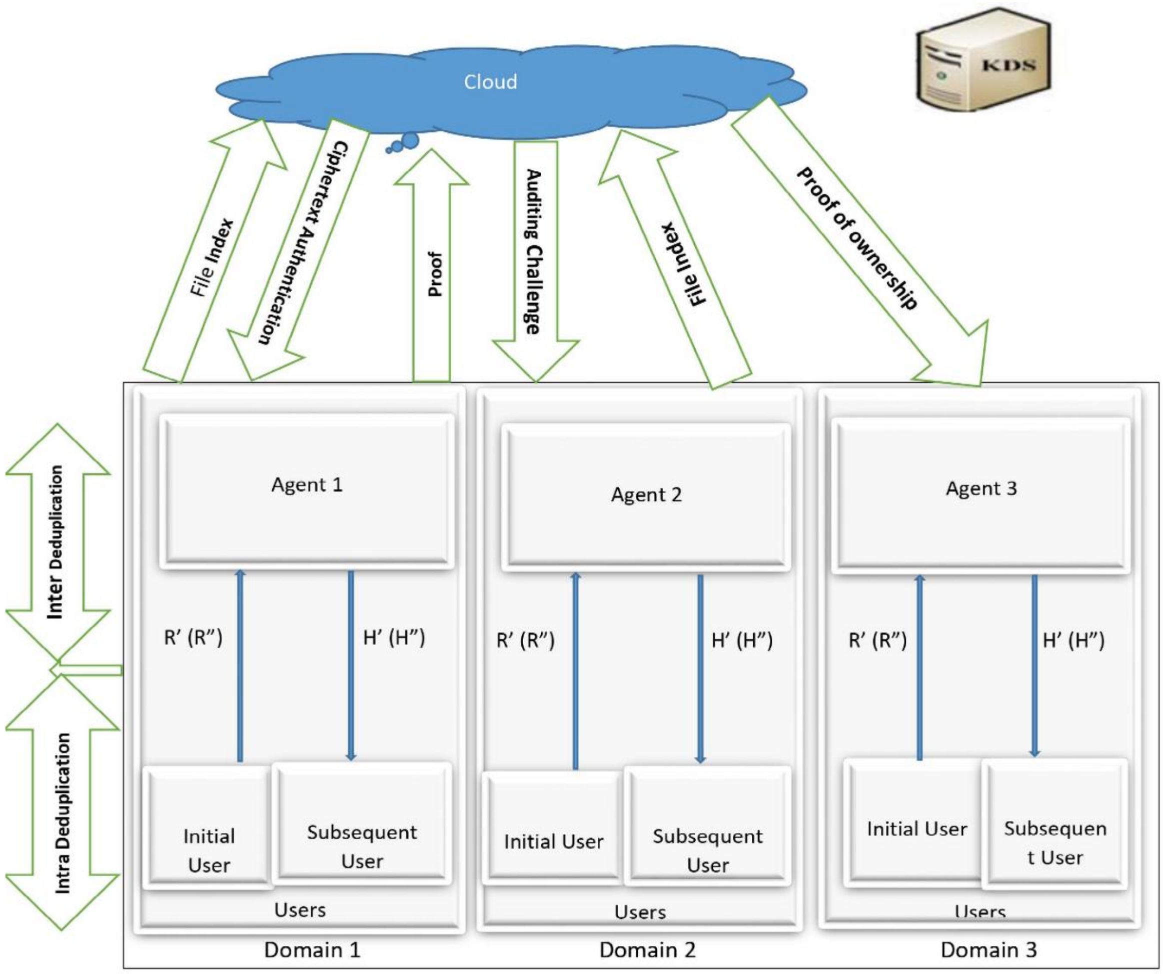


Figure 1. The DUIT model

Table 1. Symbol and notation

Sl. No	Symbol	Notation
1	ω_1, ω_2	ω1, file label
2	m, n	private-n pair
3	EK$_{sym}$, FD	Symmetric Encryption key, File
4	μ, α, k$_{fpk}$, CF	authenticator set, Intra-tag, FPK Key, Cipher text
5	HH1, HH2, hh1, hh2	Hash functions
6	W_p^*, ρ, Di, K$_{sym}$	Pseudo Random function, Security parameter, i-th domain, Private key
7	$Length_{FS}$	Length of File
8	α_{d_i}, ϖ_k	i-th Domain Intra tag, Length of keyword
9	$PO_0, PO_1, \dots, PO_{n-1}$	Pointer
10	$\boldsymbol{c}$, R', δ, φ	Cipher text small box, Blinded hash function, Measuring parameter, random value

2.2. Intra and inter DD

If the similar Intra-file-tag value has now been stored, then the agent returns "duplication||CF" to the initial client. Note that the CT is used to encapsulate some information of the EK$_{sym}$. Otherwise, the agent uses the value to yield a random inter-tag α_{d_i} based on the EK. Then, the Agent stores the Intra-file-tag table and sends the message "upload $(i, Length_{FS}, \alpha_{\mathrm{d}_i})$" to the cloud for the inter-deduplication, where i is the identifier of Di. After getting the message "upload $(i, Length_{FS}, \alpha_{\mathrm{d}_i})$" from Di, the cloud does the Inter-deduplication to additionally take out the repetition of information. In our Integra search mechanism, the DD tree search approach has been proposed to search the DD files. Initially in algorithm 1, the input data has given with length and the resultant node. The searching activity for copied data is examined by the length of the word and node. Each node has searched the defendant on the length of the data. If the leaf node is a duplicate word, it will choose the node. Otherwise, pointers $(PO_0, \varpi_1, PO_1, \varpi_2, PO_2, \dots, \varpi_{n-1}, PO_{n-1})$ are employed to point the data in the root nodes. Note that in the intra-deduplication, Man-made intelligence judges whether the copy exists by contrasting R' which is displayed in algorithm 2. However, in the Inter-deduplication, similar information will relate to various inter-tags. Accordingly, the CS cannot compare the R' of inter-tags to confirm the copy.

Algorithm 1: deduplication tree search

```
1   Input: $(length_{FS}, node)$
2   Search the input (based on data length)
3   The searching node selection process by root node.
    If (node=root node)
    The leaf node search
4   If (node=leaf node)
    Return node
5   Else
    Case 1: $length_{FS} < \varpi_1$
    Return search $(length_{FS}, Po_0)$
6   Case 2:$\varpi_k \leq length_{FS} < \varpi_{k+1}$
    Return search $(length_{FS}, Po_k)$
7   Case 3:$\varpi_{n-1} \leq length_{FS}$
    Return search $(length_{FS}, Po_{n-1})$
    End if
```

Note that $1 \leq k < n-1$, the keyword ϖ_k is the value of the data length $Length_{FS}$ Each non-leaf node contains n-1 keywords and n pointers: (PO_0, ϖ_1, PO_1, ϖ_2, PO_2, . . ., ϖ_{n-1}, PO_{n-1}). Pointers have used to point the stored data that contains the keyword $Length_{FS}$.

Algorithm 2: Inter deduplication

```
1   Initially, the message $(i, length_{FS}, \alpha_{D_i})$ from different
    domain users has given to the cloud server.
2   Decision tree approach: After the reception of
    $(i, length_{FS}, \alpha_{D_i})$, Cloud server calls the function
    $(length_{FS}, node)$ to search whether the $length_{FS}$ has been
    already stored or not.
3   If (the value is not stored in the node)
    return "DU"
    else
    "No need to upload"
4   If (same value found, check i=j)
    Return ("DU")
    Else
    Verify
    $e(\alpha_{D_i}, g_i)^p = e(\alpha_{D_i^*}, g_i)^p$        (2)
5   If equation (2) holds then,
    Return (duplication||link)
    Else
    Return ("DU")
    End if
    End if
    End if
```

2.3. Algorithm set up (initial, subsequent user, and agent)

The main use of Agent is to generate ω_1 and (ω_2) for users presented in the individual domain (D1, D2, D3…). The ω1 is answerable for testing the duplicate file present in the cloud. Just as file label has been utilized to encode and validate key generation. Each domain has different and resulting clients, where the initial client sends the ω1 to the cloud through an agent as the file upload request. The agent has stored one duplicate of the ω1 table, when the files have been guided to the cloud through an agent, the agent acknowledges is there any past file that has a similar ω1. In case, there is not stored in the CS, the initial client encrypts the file FS with the EK$_{sym}$. At first, let accept a random number $c \in W_p^*$ to compute the blinded hash function R', the expression used to calculate the R is addressed in (3).

The calculated R has been directed to AS, after getting the R' from the initial user, the agent calculates the $H' = R'^m$ value with a K$_{sym}$ (m). Finally, H' value will be led to the initial client. The initial client calculated $\delta = H'n^{-c}$ for Agent public key n, then generate ω_1 and file label (ω_1). $\omega_1 = HH_2(\delta||2)$ represents the ω1 for data duplicate checks. $\omega_2 = HH_2(\delta||2)$ represents the file label for EK$_{sym}$, K$_{sym}$, MAC$_{sym}$.

After the generation of ω_1, the ω1 has been directed to the agent, where the agent stores the ω1 value. Assuming there is a current document that has something similar ω1, then it remembers the initial user not to send the file to the cloud. If there is nothing similar to the ω1 that was stored in the agent, then this ω1 was directed to the cloud. The cloud again re-check the ω1 value. If the cloud keeps the ω1 value, the next step will be processed initial user computes the EK_sym value $\mathrm{EK}_{sym} = \mathrm{hh}_1(\omega_2||FS)$, encrypts the file as $C_{Fs} = Enc(FS,\ \mathrm{EK}_{sym})$. The CT has been divided into small blocks $(\mathrm{c}_1, \mathrm{c}_2, \mathrm{c}_3, \ldots \mathrm{c}_n)$.

$$R' = HH_1(FS)r^c \tag{3}$$

2.4. Recovery of data (RD-algorithm)

In this stage, the DD has been performed by request from the user who has the key to open the file. The client, who needs to utilize his file, presents a demand for the cloud foundry (CF) to the cloud. After getting the request from the client, the cloud initially confirms whether the client is the data owner of the ciphertext CF. If he is, the cloud sends the ciphertext CF, its relating authenticators μ, and the file tag α to the client. Something, the cloud dismisses the client's request. After getting the messages from the cloud, the client firstly confirms the validity of media access control (MAC) on file tag α with his K_{sym} K_{mac}. If the MAC is valid, the client parses α, then decodes the encoded portion by utilizing his reserved key and recovers the pseudo-random function (PRF) key k_{prf} and the random value φ. The client checks whether the succeeding authentication equation holds or not. $\sum_{l\in[1,n]} \aleph_i = \sum_{l\in[1,n]} f_{k_{prf}}(i) + \varphi \sum_{i\in[1,n]} c_i$. If the equation holds, the client trusts the ciphertext CF stored in the cloud is intact, then uses the EK_{sym} to decode the ciphertext CF, and improves the file FS: FS=Dec (CF, EK_{sym}).

3. RESULTS AND DISCUSSION

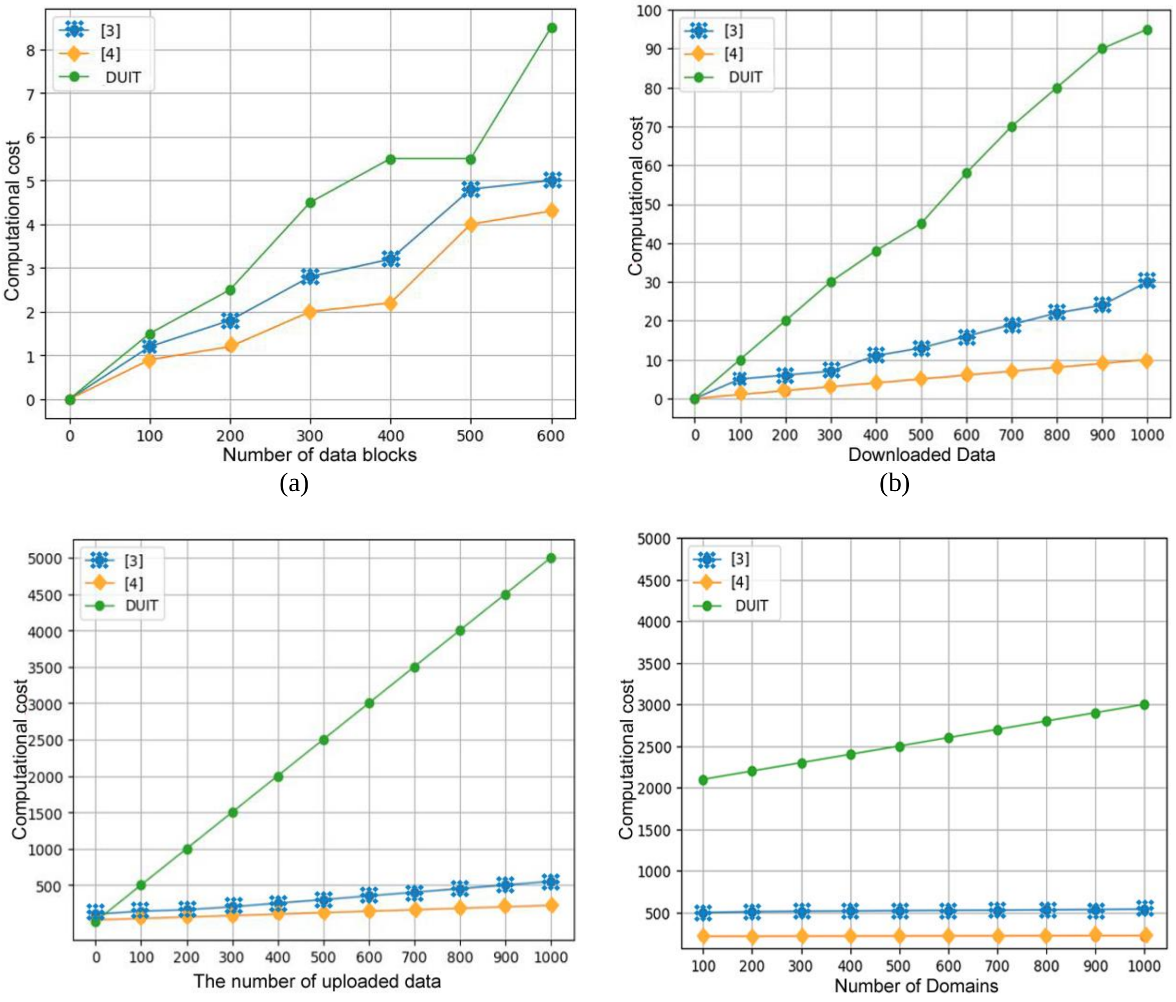
The experiment has been done in Intel i5 processor with RAM (8 GB) and Hard disk size (500 GB), 2 TB. The Ubuntu (64 bit) operating system is utilized for implementing the proposed DUIT and the data set is determined from [3] and some virtual machine (VM) images in the cloudsim platform. We set the base field size to be 512 bits, the size of an element in W_p^* to be |p|=160 bits, the size of an information file to be 20 MB collected by 1,000,000 blocks. The strong security assurance couldn't accomplish by [11]–[13] in which information privacy will have seeped to the key server. Likewise, [12]–[14] cannot accomplish validation DD which brings about substantial storage on the cloud side. In [12] method released the information wherein information stored in the cloud may be defiled or lost. Table 2 shows the comparison of various schemes alongwith parametes. When compared with novel scheme, the DUIT scheme has better results in all the parametes.

Table 2. Data integrity, protection comparison

Schemes	Data integrity auditing	Strong PP	Lightweight computation on the user side	DD	Authenticator deduplication
[11]	Yes	No	No	Yes	Yes
[12]	Yes	Yes	Yes	No	No
[13]	Yes	No	No	Yes	No
[14]	No	No	No	Yes	No
Proposed Model	Yes	Yes	Yes	Yes	Yes

3.1. Computational overhead (CO)

CO was analyzed by client and cloud. Initial and subsequent client want to cost $2(Mul_{F1} + 2Exp_{F1} + 2hash_{F1})$ and generate the ω1, file label, and cost $c\left(PRF_f + Add_{W_p^*}\right) + (c+1)Mul_{W_p^*}$ to confirm the honour of cloud information. The initial client desires to cost $\mathrm{n}(PRF_f + Mul_{W_p^*} + Add_{W_p^*})$ to produce information authenticators. The subsequent client needs to cost $\mathrm{EK}_{sym} + cMul_{W_p^*} + Add_{W_p^*} + (c-1)Add_{W_p^*}$ to create the cloud and demonstrate that he precisely claims the record. Figures 2(a), 2(b), 2(c) and 2(d). In the plan [16], it directly uses the R' of the data as the foremost key to achieving the $\mathrm{EK}_{\mathrm{sym}}$. As such, as long as the client has h(m), the $\mathrm{EK}_{\mathrm{sym}}$ selected by the initial uploader can be attained properly. Thus, this scheme would not introduce countless of additional CTs as in the plan [3]. However, this strategy is not resistant to BFA.



(a) (b)

(c) (d)

Figure 2. Performance outcomes of the DUIT with novel strategy (a) computation cost authenticator generation, (b) downloaded data vs computation cost, (c) the number of uploaded data vs computation cost, and (d) the number of domains vs computational cost

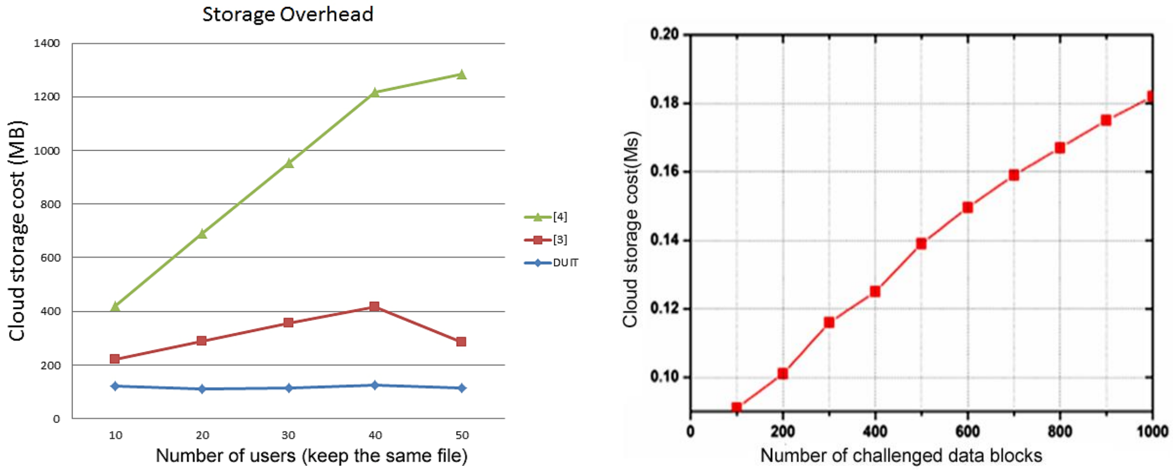
3.2. Storage overhead (SO)

To analyze the SO, we used [1], [3] benchmarks to care for the DD as a benchmark to CS. Figures 3(a) and 3(b) shows the SO of the DUIT, with various novel techniques [1], [3]. It clearly explains that the DUIT achieves, low storage cost (MB) related to the existing scheme. Subsequently, the DUIT is more proficient in cloud information storage. When relating with present procedures, for example, [1], [3] the DUIT attains 27% and 35% advance in CS cost. Just as, computation cost (CC) also linearly increased in the DUIT relating with present [1], [3] technique.

The computation overhead (CO) for CT verification when various numbers of data blocks are tested. When 100 blocks are tested, the running time of CT verification takes 0.091031ms. The running time increases to 0.182273ms when 600 blocks are tested Computation and SO is shown in Table 3.

3.3. Searching complexity vs stored data

Searching complexity (SC) vs the number of stored data graphs is shown in Figures 4(a) and 4(b). When the number of stored data increased, the SC will also increase linearly. But relating to the novel [3] procedure, the SC is very low. Because the proposed strategy uses a decision tree approach for searching the duplicate data. In Intra deduplication, the SO has constant when the numbers of data blocks increased; whereas, in inter deduplication; the SC of the DUIT technique has improved by 15% [30]. The complexity has increased linearly when the number of stored data increased in [3]. The DUIT reduces the SC and SO (1) by constructing the hash table to search the duplicate files.

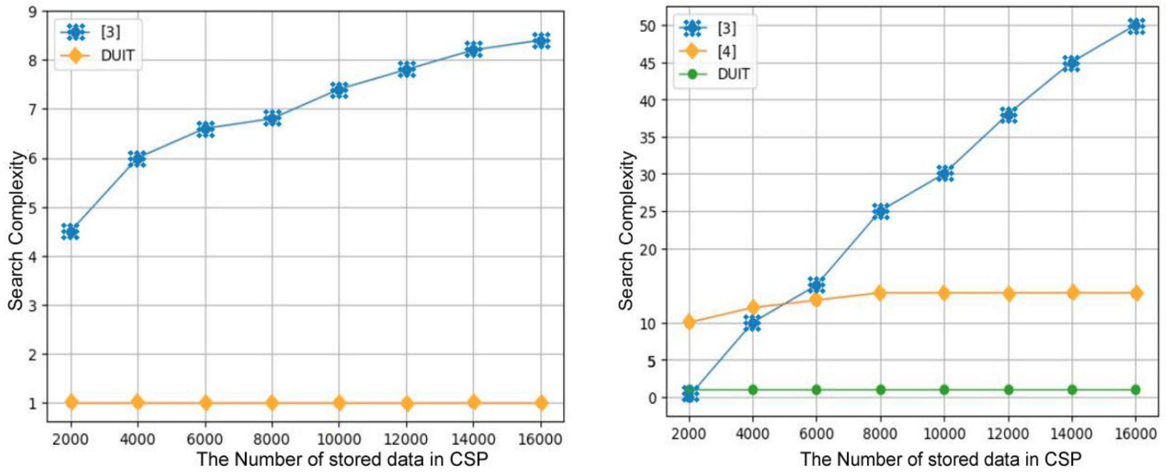


(a) (b)

Figure 3. Performance outcomes of the DUIT with novel strategy, (a) users vs Cloud storage cost and (b) cost vs challenge blocks

Table 3. CO, SO vs number of data blocks

Number of data blocks	Computation overhead	Storage overhead
200	25%	7.4%
400	36.5%	34.5%
600	57.6%	43.6%
800	75%	68.7%



(a) (b)

Figure 4. Performance outcomes of the DUIT with novel strategy (a) intra deduplication and (b) inter deduplication

4. CONCLUSION

In this article, novel DUIT deduplication has been proposed to determine the user privacy leakage and duplication issue when attacks are floated. The DUIT technique has an alternate domain with the initial client and successive client for a trivial CS auditing scheme. Moreover, decision tree-based deduplication with strong PP was likewise carried out to accomplish data integrity. The result displays that our DUIT technique provides 27% and 35% improvement in CS cost when associated with outsourcing and a lightweight computation scheme. The SC of the DUIT strategy increases 10% and 26% when associated with outsourcing and a lightweight computation scheme. Hence, the novel DUIT method achieves the computational cost, lower searching complexity in the deduplication verification and auditing phase.

REFERENCES

[1] W. Guo *et al.*, "Outsourced dynamic provable data possession with batch update for secure cloud storage," *Future Generation Computer Systems*, vol. 95, pp. 309–322, Jun. 2019, doi: 10.1016/j.future.2019.01.009.

[2] H. Hou, J. Yu, and R. Hao, "Cloud storage auditing with deduplication supporting different security levels according to data popularity," *Journal of Network and Computer Applications*, vol. 134, pp. 26–39, May 2019, doi: 10.1016/j.jnca.2019.02.015.

[3] W. Shen, Y. Su, and R. Hao, "Lightweight cloud storage auditing with deduplication supporting strong privacy protection," *IEEE Access*, vol. 8, pp. 44359–44372, 2020, doi: 10.1109/ACCESS.2020.2977721.

[4] X. Yang, R. Lu, J. Shao, X. Tang, and A. A. Ghorbani, "Achieving efficient and privacy-preserving multi-domain big data deduplication in cloud," *IEEE Transactions on Services Computing*, vol. 14, no. 5, pp. 1292–1305, Sep. 2021, doi: 10.1109/TSC.2018.2881147.

[5] M. Bi, Y. Wang, Z. Cai, and X. Tong, "A privacy-preserving mechanism based on local differential privacy in edge computing," *China Communications*, vol. 17, no. 9, pp. 50–65, Sep. 2020, doi: 10.23919/JCC.2020.09.005.

[6] J. Domingo-Ferrer, O. Farràs, J. Ribes-González, and D. Sánchez, "Privacy-preserving cloud computing on sensitive data: A survey of methods, products and challenges," *Computer Communications*, vol. 140–141, pp. 38–60, May 2019, doi: 10.1016/j.comcom.2019.04.011.

[7] T. Wang, J. Zhou, X. Chen, G. Wang, A. Liu, and Y. Liu, "A three-layer privacy preserving cloud storage scheme based on computational intelligence in fog computing," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 3–12, Feb. 2018, doi: 10.1109/TETCI.2017.2764109.

[8] L. Zhang, X. Men, K.-K. R. Choo, Y. Zhang, and F. Dai, "Privacy-preserving cloud establishment and data dissemination scheme for vehicular cloud," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2018, doi: 10.1109/TDSC.2018.2797190.

[9] Y. Wang, Y. Ding, Q. Wu, Y. Wei, B. Qin, and H. Wang, "Privacy-preserving cloud-based road condition monitoring with source authentication in VANETs," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1779–1790, Jul. 2019, doi: 10.1109/TIFS.2018.2885277.

[10] P. Li, J. Li, Z. Huang, C.-Z. Gao, W.-B. Chen, and K. Chen, "Privacy-preserving outsourced classification in cloud computing," *Cluster Computing*, vol. 21, no. 1, pp. 277–286, Mar. 2018, doi: 10.1007/s10586-017-0849-9.

[11] J. Li, J. Li, D. Xie, and Z. Cai, "Secure auditing and deduplicating data in cloud," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2386–2396, Aug. 2016, doi: 10.1109/TC.2015.2389960.

[12] R. Ding, H. Zhong, J. Ma, X. Liu, and J. Ning, "Lightweight privacy-preserving identity-based verifiable IoT-based health storage system," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8393–8405, Oct. 2019, doi: 10.1109/JIOT.2019.2917546.

[13] G. S. Kumar and D. M. P. Chitra, "Finite horizon markov decision process based fuzzy optimization for resource allocation in sdn enabled virtual networks in IAAS cloud environment," *Journal of Theoretical and Applied Information Technology*, vol. 98, no. 13, pp. 2595–2605, 2020.

[14] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in *Proceedings of the 18th ACM conference on Computer and communications security-CCS '11*, 2011, pp. 491–500, doi: 10.1145/2046707.2046765.

[15] J. Zhang, B. Wang, X. A. Wang, H. Wang, and S. Xiao, "New group user based privacy preserving cloud auditing protocol," *Future Generation Computer Systems*, vol. 106, pp. 585–594, May 2020, doi: 10.1016/j.future.2020.01.029.

[16] M. Ilokah and J. M. Eklund, "A secure privacy preserving cloud-based framework for sharing electronic health data," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jul. 2020, pp. 5592–5597, doi: 10.1109/EMBC44109.2020.9175792.

[17] M. Jiang and H. Yang, "Secure outsourcing algorithm of BTC feature extraction in cloud computing," *IEEE Access*, vol. 8, pp. 106958–106967, 2020, doi: 10.1109/ACCESS.2020.3000683.

[18] T. A. Mohanaprakash and D. V. Nirmalrani, "Exploration of various viewpoints in cloud computing security threats," *Journal of Theoretical and Applied Information Technology*, vol. 99, no. 5, pp. 1172–1183, 2021.

[19] S. Wang, Y. Wang, and Y. Zhang, "Blockchain-based fair payment protocol for deduplication cloud storage system," *IEEE Access*, vol. 7, pp. 127652–127668, 2019, doi: 10.1109/ACCESS.2019.2939492.

[20] N. Patil and B. K. Sarkar, "Secure convergent key and deduplication using distributed convergent key management," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 3, no. 1, pp. 567–571, 2017.

[21] G. Senthilkumar and M. P. Chitra, "An ensemble dynamic optimization based inverse adaptive heuristic critic in IaaS cloud computing for resource allocation," *Journal of Intelligent and Fuzzy Systems*, vol. 39, no. 5, pp. 7521–7535, Nov. 2020, doi: 10.3233/JIFS-200823.

[22] H. Qi, Y. Han, X. Di, and F. Sun, "Secure data deduplication scheme based on distributed random key in integrated networks," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, Dec. 2017, pp. 1308–1312, doi: 10.1109/CompComm.2017.8322754.

[23] J. Li *et al.*, "Secure distributed deduplication systems with improved reliability," *IEEE Transactions on Computers*, vol. 64, no. 12, pp. 3569–3579, Dec. 2015, doi: 10.1109/TC.2015.2401017.

[24] J. Xiong, Y. Zhang, X. Li, M. Lin, Z. Yao, and G. Liu, "RSE-PoW: a role symmetric encryption PoW scheme with authorized deduplication for multimedia data," *Mobile Networks and Applications*, vol. 23, no. 3, pp. 650–663, Jun. 2018, doi: 10.1007/s11036-017-0975-x.

[25] J. Yu and H. Wang, "Strong key-exposure resilient auditing for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1931–1940, Aug. 2017, doi: 10.1109/TIFS.2017.2695449.

[26] H. Wang, H. Qin, M. Zhao, X. Wei, H. Shen, and W. Susilo, "Blockchain-based fair payment smart contract for public cloud storage auditing," *Information Sciences*, vol. 519, pp. 348–362, May 2020, doi: 10.1016/j.ins.2020.01.051.

[27] Y. Zhang, J. Yu, R. Hao, C. Wang, and K. Ren, "Enabling efficient user revocation in identity-based cloud storage auditing for shared big data," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2018, doi: 10.1109/TDSC.2018.2829880.

[28] Y. Zhang, H. Zhang, R. Hao, and J. Yu, "Authorized identity-based public cloud storage auditing scheme with hierarchical structure for large-scale user groups," *China Communications*, vol. 15, no. 11, pp. 111–121, Nov. 2018, doi: 10.1109/CC.2018.8543053.

[29] F. Chen, F. Meng, T. Xiang, H. Dai, J. Li, and J. Qin, "Towards usable cloud storage auditing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 11, pp. 2605–2617, Nov. 2020, doi: 10.1109/TPDS.2020.2998462.

[30] S. Jiang, T. Jiang, and L. Wang, "Secure and efficient cloud data deduplication with ownership management," *IEEE Transactions on Services Computing*, pp. 1–1, 2017, doi: 10.1109/TSC.2017.2771280.

BIOGRAPHIES OF AUTHORS

Mohanaprakash Thottipalayam Andavan is a Research Scholar in Sathyabama Institute of Science and Technology. He received her M. Tech degree in Information Technology from Sathyabama University, Chennai, India in 2009. He has 16 years of teaching experience. He is currently working in Panimalar institute of Technology, India. He has published more than 20+ research papers in referred international and national journals. His area of research interest includes cloud computing, Network security and Web Technology. He can be contacted at email: tmohanaprakash@gmail.com.

Nirmalrani Vairaperumal received M.C.A (Computer Application) from Bharathidasan University, Trichy, India in 2000 and M. Tech (Information Technology) from Sathyabama University, Chennai, India in 2007. She has received Ph.D. degree from Sathyabama Institute of Science and Technology, Chennai, India in 2018 and 17 Years of Teaching experience in engineering college. Her area of interest includes Data Science, Machine Learning, Big Data, Data Analytics, Data Mining, and Network Security. She can be contacted at email: vnirmalraniphd@gmail.com.