INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 07, July 2024, pp: 665-670

e-ISSN : 2583-1062

Impact Factor: 5.725

SAFE AND EFFECTIVE DATA REDUNDANCY IN COMBINED CLOUD STORAGE

U. Vara Prasad[1], T. Lakshman Rao[2], R. Samule[3], J. Udayasri[4], N. Saiteja[5],

M. Prathusha[6], I. Manisha[6]

[1,2]Assistant Professor Computer science and Engineering Bomma institute of technology and science Khammam, Telangana, India.

[3,4,5,6,7]B. Tech degree in Computer and science Engineering University of JNTUH, Bomma institute of technology and science, Khammam, Telangana, India.

ABSTRACT

In this Users' required bandwidth may be decreased and data redundancies in cloud storage can be effectively eliminated using data deduplication. However, because to their high computational overhead, low attack resistance, information leakage, and other issues, the majority of earlier methods that relied on a trusted key server (KS) are constrained and susceptible. Specifically, a single-point-of-failure occurs when the trusted KS fails, causing the entire system to malfunction. In this work, we present a Secure and Efficient data Deduplication strategy (called SED) for a Joint Cloud storage system that collaborates with many clouds to deliver worldwide services. SED facilitates the exchange and updating of dynamic data without reliance on the reliable KS. Furthermore, SED has the ability to circumvent the single-point-of-failure that is frequently present in traditional cloud storage systems. As stated by According to theoretical assessments, our SED has significant anti-attack capabilities, including resistance to collusion and brute-force attacks, and guarantees semantic security in the random oracle model. Furthermore, SED may efficiently remove data redundancies with no expense in terms of processing, connection, or storage. SED's effectiveness and features enhance client-side usage. Lastly, the comparison findings demonstrate that our approach performs better than the current techniques.

Keywords: Drowsiness detection, deep learning, smart edge, convolutional neural network (CNN).

1. INTRODUCTION

One of the primary methods for ensuring data security in deduplication is convergent encryption, which helps shield outsourced data from rogue and unreliable CSPs. A basic known as the message-locked encryption (MLE) method was developed by Bellare et al. in [10]. Then, on the basis of Bellare's study, a few variations [12], [13] were suggested. But because the encryption keys for these MLE-based schemes come from the files themselves, there were a lot of potential hazards associated with them. For limited message distributions, Abadi et al. [11] developed a deterministic encrypted method and a complete randomized system based on the non-degenerate efficiently computable bilinear map. A method for achieving dependable key management in deduplication was given by Li et al. in [15]. Next, Jiang et al. [14] presented a safe plan for Using the random tag for deduplication. They did not, however, address the necessity of updating the data.

Hur et al. later examined dynamic ownership management for safe deduplication and in order to accomplish deduplication, each client must store the keys that are situated along the route of a binary tree that contains all of the keys. Secure deduplication with key management based on secret sharing strategies was proposed. Following that, a decentralized server-aided encryption for deduplication was created. However, it required several exchanges between users and KS, which provided the attackers with chances to extract valuable information from the exchange. n this paper, we propose a novel scheme, which aims at efficiently solving the problem of deduplication with frequent cloud user revocation and new cloud user joining in cloud computing. In particular, different from existing data deduplication methods, which employ either trusted/semi-trusted third party to do proxy re-encryption work, our proposed scheme designs a hybrid cloud architecture, which includes a public cloud and further introduces a private cloud. In implementations, the introduced private cloud in our scheme is involved as a data owner and a proxy at the same time to 1) control access to outsourced data by performing re-encryption techniques and 2) manage the dynamic ownership when real data owner is offline or revoke his/her ownership. exchanges between users and KS, which provided the attackers with chances to extract valuable information from the exchange. Miao et al. presented a secure deduplication method for multi-server-aided. In order to accomplish detailed data deduplication. In this research, we present an effective and safe data deduplication technique (SED) for the Joint Cloud storage system that does not rely on a trusted key. Our SED's sub-algorithms draw inspiration from the completely randomized tag generation algorithm

[11], which aids in the identification of duplication and safeguards the outsourced data against collusion attempts. In contrast to earlier deduplication techniques, our SED guarantees semantic security for both the tag and the ciphertext. The ciphertext and tag together prevent any attacker from learning any valuable information. Furthermore, our SED is the only system that securely allows for data sharing and updates.

TABLE 1. Literature Review Summary.

	Benchmark	Features used	Embedded Device Deployment	Architecture	Compressed	Compression Algorithm
[7]	NTHU	RGB videos, Optical Flow Features	No	Centralized	No	-
[8]	Custom	Facial Features, Upper body Position	No	Centralized	No	-
[9]	NTHU	Region of Interest, Global Facial Features	No	Centralized	No	-
[13]	Custom	Region of Interest	Yes	Centralized	No	-
[14]	DROZY	Region of Interest	Yes	Centralized	Yes	Teacher-Student
[6]	NTHU-Expanded	Region of Interest	Yes	Centralized	Yes	Network Pruning
[10]	YawDD	Region of Interest	Yes	Distributed	No	-

We create an encryption technique in our SED that facilitates sharing, updating, and data deduplication. To the best of our knowledge, SED is the first program that takes into account the scenario in which the data owner gives authorized people access to their outsourced data. In particular, the cooperation of the involved CSPs generates a master encryption key. It guarantees key generation's adaptability and security. The deployment of data updating and sharing procedures is aided by the data access control system that is based on SED authentication. Next, in order to increase the efficiency of data deduplication, SED combines the intra- and inter-deduplication techniques to remove duplicates from the Joint Cloud system. Following that, theoretical evaluations show that the SED performs better in terms of data integrity, data secrecy, robust assaults, and functionalities and resistances to collusion. SED is developed and simulated using Ubuntu's Crypto++ [23], GNU [24], and PBC [25] libraries in order to assess the complexity empirically. The evaluation's findings demonstrate SED's minimal computational overhead and efficiency.

2. SYSTEM MODEL

In this section, we describe the data deduplication system and define the adversary model. Only the file-level deduplication is considered in this paper. A. Hybrid Architecture for Secure Deduplication Figure 1 shows the architecture of the data deduplication system, which consists of three entities stated as follows. • Data users (DU). This is an entity that wants to outsource data to Pub-CSP and access data later. In the authorized deduplication system, each user is issued a pair of secret key and public key (sku, pku) about Proxy Re-Encryption (PRE) and (SKu, PKu) for signature. Moreover, if a data user is the first one to upload file FA, she/he will be defined as the owner u1, A; if the file FA already exist, the user will be defined as the holder Ui, A. • Public Cloud (Pub-CSP). The entity that provides a data storage service in the public cloud. In this paper, Pub-CSP is assumed to be always online and capable to provide abundant storage. • Private Cloud (Pri-CSP). The entity that provides an execution environment and infrastructure for data users as an interface between DU and Pub-CSP, since the computing resources of DU are limited and Pub-CSP is not fully trusted in practice. Pri-CSP maintains an ownership list for storing data, which is composed of a hash code set for the stored data, identities, and reencrypted keys belonging to the owners including collusion, manipulation, and brute-force attacks.

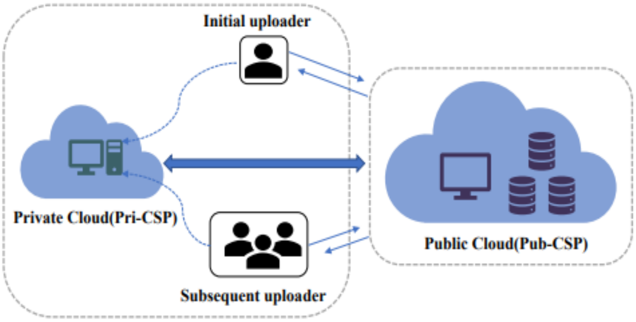


Fig. 1. Architecture of a data deduplication system.

Secure deduplication is implemented by the SED independently of the trusted key server. Additionally, it facilitates cross-cloud data exchange and upgrading. Moreover, SED increases the scalability of the traditional deduplication approach and resolves the single-point-of-failure problem.

Using cryptographic tools, we do experiments using the SED technique to confirm its efficiency and minimal computing cost. Our unique simulations of intra- and inter-deduplication show that these techniques can enhance the effectiveness of duplicates detection.

3. PROPOSED DEDUPLICATION SCHEME

Data Deduplication Fig.2 shows the procedure of initial upload by owner u1. Fig.3 shows the procedure of deduplication when subsequent uploaders u2 uploads the same data with u1. We assume that Pri-CSP plays a role as owner u0 to control deduplication for owner u1. • Step 1 - Key Generation: After system parameter generation, each DU asks KeyGen1(uid) to generate key pair (sku, pku) for PRE and (SKu, PKu) for signature. Meanwhile, Pri-CSP gets (sku0, pku0) by KeyGen1(u0). • Step 2 - Duplication Check: DU u1 stores data F at PubCSP. He/She calculates H(F), signs it with SKu1 and sends data package dp = {H(F), sign(H(F), SKu1)} to Pub-CSP. The duplication check will be performed by Pub-CSP to verify if the same data has been stored already after verifying the signature. If the check is positive, go to Step 5. Otherwise, go to Step 3 to request key for encryption. • Step 3 - Data Storage: When Pub-CSP defines u1 is the first uploader of data F, it generates a unique Fid for F and contacts Pri-CSP to get key for encryption.

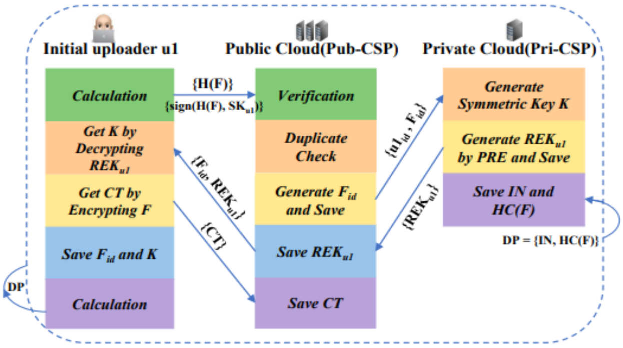


Fig. 2. Procedure of initial upload by owner u1.

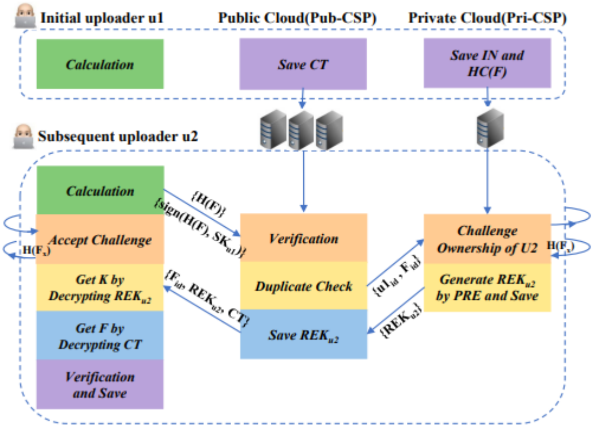


Fig. 3. Procedure of deduplication.

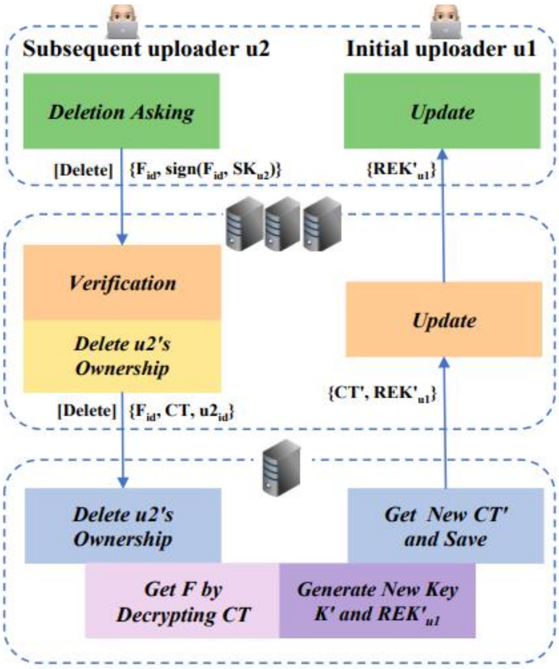
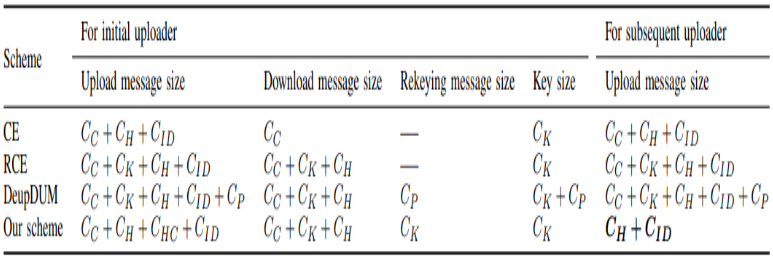


Fig. 4. Procedure of ownership revocation of data holder u2 when owner u1 is offline.

The performance of many sensors, including the accelerometer sensor, using data from as input to produce the categorization is the first tested method. The eye and mouth regions of interest (ROIs) are used in the second tested method. This method builds two classifiers, the outputs of which are combined to determine whether or not the driver is sleepy. The mouth ROI is classified into the normal or yawing condition by the second classifier, while the eye ROIs are classified into open and closed classes by the first classifier. When to determine whether or not the driver is sleepy. The mouth ROI is classified into the normal or yawing condition by the second classifier, while the eye ROIs are classified into open and closed classes by the first classifier. A motorist is deemed tired if their mouth is yawning and their eye is closed, but they appear normal elsewhere utilizes Five frames per second (FPS) of picture frames taken from movies are put into the model to train it in two different scenarios. employing raw frame photos from a single scenario, the model is trained without employing face identification scenarios. The face region is found using the built-in dlib face detector, which is then cropped from the picture and sent into the model.

TABLE.2

COMMUNICATION OVERHEAD.



Scheme	For initial uploader				For subsequent uploader
	Upload message size	Download message size	Rekeying message size	Key size	Upload message size
CE	$C_C+C_H+C_{ID}$	C_C	—	C_K	$C_C+C_H+C_{ID}$
RCE	$C_C+C_K+C_H+C_{ID}$	$C_C+C_K+C_H$	—	C_K	$C_C+C_K+C_H+C_{ID}$
DeupDUM	$C_C+C_K+C_H+C_{ID}+C_P$	$C_C+C_K+C_H$	C_P	C_K+C_P	$C_C+C_K+C_H+C_{ID}+C_P$
Our scheme	$C_C+C_H+C_{HC}+C_{ID}$	$C_C+C_K+C_H$	C_K	C_K	$\mathbf{C_H+C_{ID}}$

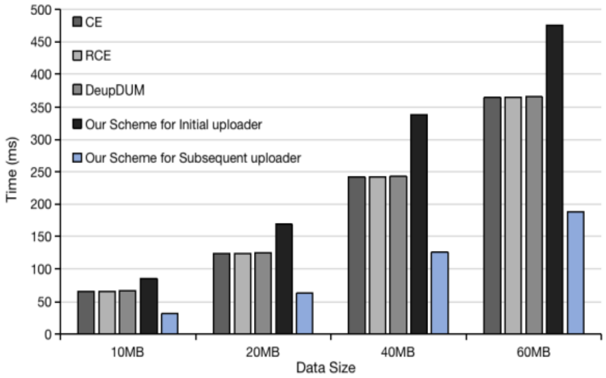


Fig. 5. Computation time for upload.

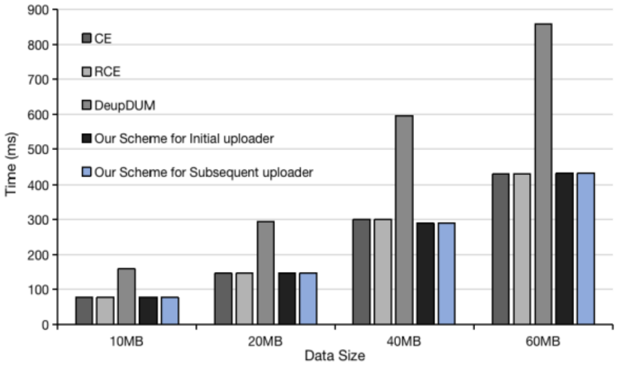


Fig. 6. Computation time for download.

Concerning the rekeying message size, the DedupDUM and our scheme increase the size of the re-encryption key while scheme CE and scheme RCE do not consider key updating. However, even though the group key is used to manage ownership revocation, the encryption key K determined by data F still never changes in DedupDUM once it is settled, which is unsecured since the withdrawn owners can collude with Pri-CSP for getting ciphertext before rekeying. Our scheme updated K as long as there is ownership revocation, which is more secure and computation overhead is accepted since it can be performed by Pri-CSP instead of DU. C. Performance Evaluation In this section, we compare our scheme with previous works. We follow common practice [18], [21], [25] for fair comparison, each cryptographic operation is implemented by using the umbral library ver. 0.3.0 and Crypto library ver. 1.4.1. We perform data encryption and decryption algorithm with AES where the key is 128-bit. The data size ranges from 10MB to 60MB. The testing environment is Intel(R) Core (TM) i5-7300HQ CPU 3.1 GHz 16.0GB RAM.

A. **Data Privacy-** In terms of data privacy, the raw data should be prevented from Pub-CSP (honest but curious) and unauthorized data users. Therefore, there are normally two kinds of attacks, separately from Pub-CSP and invalid data users. Firstly, as an attack from Pub-CSP is concerned, what is saved on Pub-CSP is the authorized DU's re-encrypted key that is encrypted through PRE by Pri-CSP and only can be decrypted by DU's private key, since Pri-CSP and authorized users will not collude with Pub-CSP considering their profits, it is impossible for the Pub-CSP to get plaintext by cipher key. Second, supposing an unauthorized user u2 requests data F with (Fid, u1, id) (u1 is valid and using u2, id will simply not pass the ownership check), Pub-CSP searches the ownership list P2 = {Fid, CT, H(F), (u1, id, REKu1)}, and return {Fid, CT, REKu1 } to user u2 based on (Fid, u1,id). Since REKu1 can only be decrypted by the private key of user u1, it is computationally infeasible for user u2 to obtain plaintext of F by decrypting CT with REKu1. Therefore, data privacy against the honest-but-curious Pub-CSP and unauthorized users is guaranteed.

B. **Data Consistency-** In the data deduplication scheme, data integrity may be threatened by a poison attack on tag consistency, which could be identified during the decryption process by the data holders. Assuming that attacker u2 owns the same data FA with authorized user u3, u2 uses FB 6= FA to generate a fake ciphertext CTB, then uploads (H(FA), CTB) to Pub-CSP to pretend to be CTA. When the authorized user Fig. 8. Computation time for each process with a duplicate ratio grows. u3 wants to upload FA, u3 sends H(FA) to Pub-CSP to check duplication. Since H(FA) exists, Pub-CSP asks PriCSP performing deduplication. Pub-CSP sends (CTB, REKu3) back to user u3 after deduplication, and u3 checks whether H (Decrypt (De (sku3, REKu3), CTB)) = H(FA) holds or not, if this is not consistent then u3 drops the messages and reports this to Pub-CSP. Therefore, our scheme guarantees the data integrity.

C. **Data Ownership Verification-** In our scheme, the data ownership is verified by challenging DU with random H(Fx) in hash code set (randomly select specific parts of the data, e.g., the hash code of 10.5-14.3% of F). Fx is randomly selected and function H (∗) is noninvertible, therefore it is impossible to calculate H(Fx) without the original plaintext.

D. **Ownership Revocation-** Access to the data should be restricted for the user whose ownership has been removed. Our technique uses Pri-CSP to act as owner u0, guaranteeing ownership revocation. The owner, u0, removes the requestor's ownership information from the ownership list whenever the data owner, u1, withdraws ownership or requests that other holders delete or modify their data. If u1 is offline, revokes ownership, or requests that u0 update the data on their behalf, u0 re-encrypts the plaintext using the new symmetric key before re-uploading it to Pub-CSP and updates the re-encrypted keys of the remaining users. As a result, the owner of the withdrawn data will be unable to pass the access check and use the outdated cipher-key to decode the most recent ciphertext.

4. CONCLUSION

In this paper, we proposed a secure and practical scheme that managed the encrypted data with deduplication, based on ownership challenge under a hybrid cloud architecture, where Pub-CSP manages the storage and Pri-CSP plays a role as owner u0 and proxy at the same time to perform deduplication and dynamic ownership management. Further, our scheme proves that the owner holds the real data alone pass the data ownership, and encrypted data can be securely accessed because only authorized data holders can obtain the symmetric keys used for data decryption. Security analysis, comparison with existing work, and implementation-based performance evaluation show that our scheme is secure and efficient, and resists collusion attacks and duplicate faking attacks.

5. REFERENCES

[1] Ibrahim, Abaker, Targio, Hashem, Ibrar, Yaqoob, Nor, Badrul, Anuar, and Salimah, “The rise of” big data” on cloud computing: Review and open research issues,” Information Systems, vol. 47, no. Jan., pp. 98– 115, 2015.

[2] F. M. Awaysheh, M. N. Aladwan, S. Alawadi, J. C. Cabaleiro, and T. F. Pena, “Security by design for big data frameworks over cloud computing,” IEEE Transactions on Engineering Management, vol. PP, no. 99, 2021.

[3] Duan and Qiang, “Cloud service performance evaluation: status, challenges, and opportunities-a survey from the system modeling perspective,” Digital Communications & Networks, pp. 101–111, 2016.

[4] Z. Yan, L. Zhang, W. Ding, and Q. Zheng, “Heterogeneous data storage management with deduplication in cloud computing,” IEEE Transactions on Big Data, pp. 1–1, 2017.

[5] S. Quinlan and S. Dorward, “Venti: A new approach to archival storage,” proc. usenixconf.on. on file & storage tech, 2002.

[6] G. R. Blakley and C. Meadows, “Security of ramp schemes,” in Advances in Cryptology, Crypto 84, Santa Barbara, California, Usa, August, 1984.

[7] M. Bellare, S. Keelveedhi, and T. Ristenpart, “Dupless: Server-aided encryption for deduplicated storage,” in Use nix Conference on Security, 2013.

[8] X. Jin, L. Wei, M. Yu, N. Yu, and J. Sun, “Anonymous deduplication of encrypted data with proof of ownership in cloud storage,” IEEE/CIC International Conference on Communications in China, 2013.

[9] J. Li, X. Chen, M. Li, J. Li, P. P. Lee, and W. Lou, “Secure deduplication with efficient and reliable convergent key management,” IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 2014.

[10] J. Li, Y. K. Li, X. Chen, P. Lee, and W. Lou, “A hybrid cloud approach for secure authorized deduplication,” Parallel & Distributed Systems IEEE Transactions on, vol. 26, no. 5, pp. 1206–1216, 2015.

[11] M. Bellare, S. Keelveedhi, and T. Ristenpart, “Interactive message locked encryption and secure deduplication,” in Springer, Berlin, Heidelberg, 2015.

[12] H. Cui, R. H. Deng, Y. Li, and G. Wu, “Attribute-based storage supporting secure deduplication of encrypted data in cloud,” IEEE Transactions on Big Data, pp. 1–1, 2017.

[13] W. Shen, Y. Su, and R. Hao, “Lightweight cloud storage auditing with deduplication supporting strong privacy protection,” IEEE Access, vol. 8, pp. 44 359–44 372, 2020.

[14] N. Almrezeq, M. Humayun, A. El-Aziz, and N. Z. Jhanjhi, “An enhanced approach to improve the security and performance for deduplication,” Turkish Journal of Computer and Mathematics Education (TURCOMAT), vol. 12, no. 6, pp. 2866–2882, 2021.

[15] Hur, Junbeom, Koo, Dong young, Shin, Young Joo, Kang, and Kyung Tae., “Secure data deduplication with dynamic ownership management in cloud storage.” IEEE Transactions on Knowledge & Data Engineering, vol. 28, no. 11, pp. 3113–3125, 2016.