CLOUD DATA DEDUPLICATION: ENSURING INTEGRITY AND ELIMINATING DUPLICATION

[#1]VADUKAPURAM ROSHINI,
[#2]P.SATHISH, *Assistant Professor,*
[#3]Dr.V.BAPUJI, *Associate Professor& HOD,*
Department of Master of Computer Applications,
VAAGESWARI COLLEGE OF ENGINEERING, KARIMNAGAR, TELANGANA.

ABSTRACT : Deduplication is a technique for reducing storage costs by deleting redundant copies of data. Data deduplication is becoming an increasingly significant requirement for cloud storage providers due to the ongoing and exponential development in both user numbers and data volume. Their cloud providers may save money on storage and data transfer by keeping a unique copy of duplicate data. Cloud computing provides a new method to service delivery by rearranging various resources across the Internet. The most well-known, significant, and commonly utilized cloud service is data storage. Because data owners' privacy is more important, data is usually kept in the cloud in encrypted form to safeguard their private. Encrypted data will complicate cloud data deduplication, which is critical for data processing and storage. Some traditional deduplication techniques are incompatible with encrypted data. Some current methods for encrypted data deduplication include security weaknesses, such as brute force attacks, that preclude them from implementing data access control and revocation in an adaptable manner. As a result, only a few of them may be employed in practice without trouble. This article describes a method for deduplicating encrypted data stored in the cloud. The results show a flow diagram of the deduplication process as well as test scenarios for login purposes, particularly for data deduplication in cloud storage. For encryption and hash code generation, we also use the RAS and AES algorithms.

*Keywords:*Deduplication, Encryption Technique, Decryption Technique.

1. INTRODUCTION

Providers of cloud computing services typically provide consumers virtually unlimited capacity for storing data, so they naturally seek ways to reduce redundant information. The compression technique is used to get rid of redundant information. Cross-user compression is typically utilized. Deduplication is predicated on the principle that duplicate information need only be kept once. Deduplication is disabled by default in cloud services, therefore users will receive an error message if they try to upload duplicate files. Deduplication can reduce backup space requirements by up to 90-95% and standard file system space requirements by up to 68%.Encryption is essential for user privacy and data security, as well as portability and low total cost of ownership. Inconveniently, encryption and deduplication are incompatible technologies. If two pieces of information are encrypted and look identical to one other, you won't be able to tell them apart. Conversely, deduplication seeks duplicates and attempts to store them only once. Deduplication cannot be used by the cloud storage service if users encrypt their data using the prevalent industry standard. However, customers cannot have assurance that their data will remain private if it is stored in the cloud without encryption. Establishing a safe environment. The key is typically generated from a replica of the data in convergent encryption. This strategy is proposed to accommodate these two distinct requirements. Although convergent encryption appears to be a promising approach to privacy protection and redundant data elimination, it is plagued by a number of well-documented issues. The topics of data compression and cloud storage take up the bulk of this piece. The delivery of IT services via the cloud is a relatively new concept.

It does this by redistributing resources like storage and processing to users according to their need.

Cloud computing is able to create a large pool of resources by combining many networked computers. It's advantageous in several ways, including that it's adaptable, error-tolerant, scalable, and cost-effective. This means it can now be employed as a service infrastructure. Data storage in the cloud is the most crucial and often utilized cloud service. Users of the cloud entrust their sensitive information to the cloud provider's data center. Users of the cloud should be aware that they cannot always rely on their CSP to keep their private data safe from intrusion and assault. The likelihood of privacy breaches and other major data security problems increases when an individual gives up control over their personal data. The rapid development of data mining and other analysis techniques has resulted in a serious issue: the protection of individuals' personal information. If you care about your users' security and privacy, you should only upload encrypted data to the cloud. However, CSP may receive duplicate protected data from the same user or from several users. When a large number of people share information, this is especially possible. Data duplication is a problem in the cloud because it consumes unnecessary amounts of energy, creates unnecessary network traffic, and makes data management more difficult. The importance of efficient resource management will grow as more services are introduced. How to effectively store encrypted data using deduplication is a challenge. However, conventional de-duplication processes in industry cannot handle encrypted data. The current state of deduplication is vulnerable to brute force attacks. Access control and revocation cannot be provided for dynamic data simultaneously. None of our current solutions are reliable enough to guarantee confidentiality, safety, or durability. Many factors make it problematic to provide data owners control over deduplication in practice. As a precaution, initial storage delays are implemented in case data consumers aren't always online or available to participate in this kind of management. The second issue is that if the deduplication process requires too many dialogues and calculations, the data owners may not want to take part in it. Third, searching for duplicate information could compromise personal privacy. Fourth, due to data super distribution, the data's owner might not be able to provide a user with access to the data or a compression key. This prevents CSPs and data owners from cooperating on deduplication of data storage. The findings demonstrate the system's efficacy and practicality, especially with regards to data deduplication in cloud storage.

2. RELATED WORK

Without DOM, deduplication would fail to function. Convergent encryption was proposed as a means of dealing with the issue of data redundancy. To obtain the key for a given piece of data F, a user must first determine its hash code, K = H(F). Data F will be encrypted using key K, thus anyone with access to both sets of data can generate identical encrypted data. Bellare devised DupLESS, a server-assisted encryption method, to compress data. Calculating DupLESS block-level deduplication is time-consuming due to the need for a separate key server to generate the necessary keys. Liu proposed an infeasible client-side encryption approach that requires the data's owner to verify ownership and delete any duplicates. Cui developed a deduplication technique that is compatible with hybrid cloud infrastructure and makes use of attribute-based access control. In order to locate backups, a short hash was recently implemented. Because distinct data can share the same short hash, it is resistant to offline brute-force attacks yet has a high collision rate. After that, the intricate scheme relied on straightforward tactics. This strategy is ideal for private users who rely on cloud storage because it does not function for content that is duplicated excessively. These techniques don't account for users' shifting attitudes toward data ownership over time.

B. Duplicate content and ownership changes on the fly The use of a semitrusted third party or a trusted third party to perform proxy re-encryption is one of the proposed solutions for dealing with dynamic control, such as AP and Pub-CSP. Wen

devised a method of distributing convergent keys. Data users do not have access to sufficient computing resources to encrypt and decrypt convergent keys or recover them from concealed shares. Due to the constant quantity of data consumers in the cloud, Hur demonstrated how to eliminate duplicate data on the server side. In subsequent iterations, Yan enhanced the strategy for handling disparate data storage systems. However, the owner can't possibly provide remote access and be available at all times. Additionally, when the owner is unavailable, only the AP will be responsible for granting access. Using trusted entities and attribute-based access control via the group key, Premkamal improved upon previous methods. While these techniques do address the issue of managing dynamic ownership, they do so at the expense of security. Most proxy re-encryption solutions rely on known or semi-trusted third parties, which makes some data owners wary about providing access to a stranger. Additionally, the act of eating

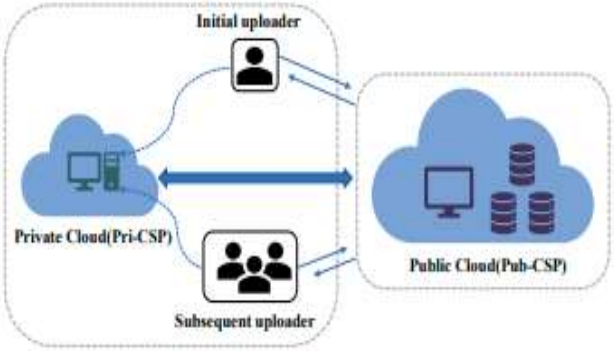


Fig. 1. Architecture of a data deduplication system.

Unauthorized users who normally can only access file A will suddenly get access to file B whenever a malicious user collaborates with a third party (such as a proxy service). This is the central research question that will be addressed.

3.SYSTEM MODEL

In this section, we will discuss the nature of an adversary and the operation of the data deduplication system. In this research, we focus solely on deduplication at the file level.

A. Hybrid Architecture for Secure Deduplication

The data compression system has the three main components shown in Figure 1.

. • **Data users (DU).** The goal of this group is to save data in Pub-CSP for later retrieval. Secret and public signing and proxy re-encryption (PRE) keys (sku, pku) are distributed to each user of the deduplication system. When a user uploads a file for the first time, they are considered the creator (u1,A) of that file. They become the holder (ui,A) if file FA already exists.

• **Public Cloud (Pub-CSP).** The corporation in charge of a shared cloud storage system. This research assumes that Pub-CSP is available and has plenty of room.

. • **Private Cloud (Pri-CSP).**Providing an execution environment and infrastructure for data users works as a bridge between DU and Pub-CSP due to the former's low computing power and the latter's unreliability in practice. Pri-CSP stores a hash value, a list of data owners, and re-encrypted keys. The information is catalogued in this list.

PriCSP is assumed to be used by a single large organization for the purposes of this system. The entire corporation owns, operates, and controls it. Therefore, Pri-CSP can be relied upon by any and all parties. Pub-CSP may be curious about raw user data, but it will still adhere to the system's specifications. Due to their divergent business aims, we also believe that Pri-CSP and DU will never collaborate with Pub-CSP.

B. Security Requirements

Data privacy, data consistency, proof of ownership, reclaiming ownership, and resistance to collusion are some of the other security features.

Data privacy. protecting sensitive information from prying eyes, including the Pub-CSP server.

Data consistency. The tags are immune to harm. Those with access to the information can detect when the ciphertext has been altered.

Ownership verification. Data, including the ciphertext and any accompanying messages, should not be accessible to anybody whose ownership cannot be verified.

Ownership revocation If an authorized user requests that their data be deleted or altered in the

cloud, that user's access to the data and their name from the list of lawful owners must be revoked.

Collusion resistance. Even if they team up with other unauthorized data users or the Pub-CSP, a user who does not legally possess the data should not be able to examine the raw data.

4.PERFORMANCE EVALUATION

A. Take stock of the present circumstance Convergent encryption (CE), randomized convergent encryption (RCE), data deduplication with dynamic user management (DedupDUM), and our own approach are all compared and contrasted in this article. The core features of this system are its ability to restrict access, manage titles in real time, ensure tags are consistent, eliminate duplicate data via encryption, and verify ownership. All of the choices employ encrypted storage, making it simpler to maintain confidentiality. However, Scheme CE is vulnerable to the tag consistency attack. DU can verify the consistency of the tags in the received data using several means while still ensuring the data's integrity. Using the group key generated by DU's public key, DedupDUM addresses the issue of dynamic ownership management. This allows existing users to be removed and new ones to be added. They don't verify that the user actually possesses the complete original file, rather than merely a tag, phony ciphertext, and ID, and they

TABLE 1 COMMUNICATION OVERHEAD.

Scheme	For initial uploader				For subsequent uploader
	Upload message size	Download message size	Relaying 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	$C_C+C_K+C_H+C_{ID}$
DedupDUM	$C_C+C_K+C_H+C_{ID}+C_P$	$C_C+C_K+C_H$	C_P	C_K+C_P	$C_C+C_K+C_H+C_{ID}+C_P$
Our scheme	$C_C+C_H+C_{RC}+C_{ID}$	$C_C+C_K+C_H$	C_K	C_K	C_H+C_{ID}

don't account for attacks in which a dishonest cloud server and attackers work together. Unlike other methods, our technique dynamically manages who owns what data F by maintaining two independent lists, one at the Pub-CSP and one at the Pri-CSP. Our approach enables for the removal and addition of cloud users because pkui is utilized to generate the re-encrypted key REKui. After verifying that DU is not the legitimate owner of file F, deduplication is performed.This means significant savings in transportation costs. Table 1 displays the relative expenses of the four available channels of communication throughout Part B of the evaluation. Cloud user id (CID), hash code (CH), hash code group (CHC), and encrypted data volume (CHC) are represented by the letters CC, CID, CH, and CHC, respectively. F, CK represents the size of a key, and Cp represents the size of a public key. All three methods (CE, RCE, and DedupDUM) provide identically sized upload messages for the initial transfer of data F. Our method enlarges the hash code set HC(F), which is utilized to determine who owns DU0, prior to deduplication. As can be seen in Table 1, our solution just uploads H(F) to get ready for the future upload of F before confirming ownership or validating access, which is in contrast to the other techniques.

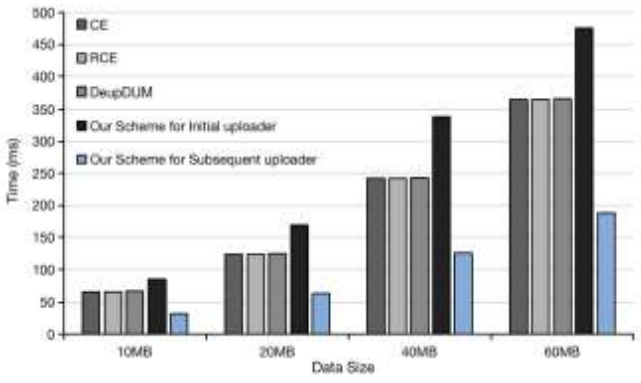


Fig. 2. Computation time for upload.

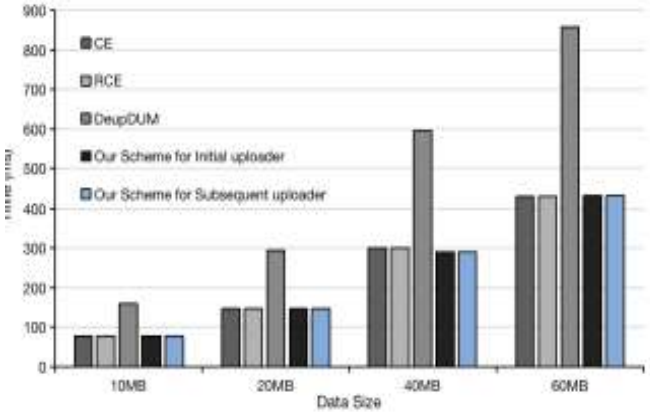


Fig. 3. Computation time for download.

The re-encryption key grows with both DedupDUM and our method. However, while calculating the size of the rekeying message, CE and RCE don't factor in changes to the keys. The data F that defines the encryption key K in DedupDUM does not change after it has been discovered, despite the fact that the group key manages ownership revocation. Owners who choose to leave the system can employ Pri-CSP to obtain ciphertext prior to rekeying, making the system insecure. Our method was safer, and we didn't mind the extra effort involved, because it could be constructed using Pri-CSP instead of DU as long as ownership was revoked. Analysis of results This section contrasts our strategy with those of other authors. We ran each cryptographic operation through versions 0.3.0 of the umbral library and 1.4.1 of the Crypto library to ensure a level playing field. The 128-bit key used for encryption and decryption is generated using the AES method. Storage capacities range from 10MB to 60MB. The 3.1GHz Intel(R) Core(TM) i5-7300HQ CPU with 16.0GB of RAM in the testing system should perform admirably.

We put various file-sharing systems to the test, including 1) the upload processing time. The number of calculations required by the DedupDUM scheme is equivalent to those of the CE and RCE schemes. Data F's hash code and hash code set are calculated, the signature is signed and confirmed, the re-encrypted key is decrypted, and finally, data F is encrypted using AES, as illustrated in Figure 2. This strategy only produces a marginal improvement when compared to others. It's clear that our strategy for simultaneously uploading numerous files has many benefits. In contrast to previous systems, ours just requires reuploading H(F) prior to title and access verification. Using our method, we can cut down on the incredibly high cost of communication.

2) How long it takes to complete the download: In contrast to traditional approaches, our ownership verification technique tests DU's access to the full data set by challenging it with a random H(Fx) from the hash code set. When DU doesn't have to pay to use a dataset, this reduces the price of communication significantly. Download ciphertext computation times are compared between methods and depicted in Figure 3.

Time required for both encrypting and decrypting a message Our technique, like the DedupDUM, addresses the issue of dynamic ownership management by re-encrypting the data during the encryption process. DedupDUM slows down Pub-CSP's processes since it has to decode and re-encrypt the ciphertext for each new user. However, the difficulty of solving the calculation increases with the number of holders. Here are the precise timeframes needed to encrypt and decrypt files ranging in size from 10MB to 60MB: Time required for deduplication processing: Since each DU has its own 2MB file, while the DR is 0, 50MB of information is stored in the cloud. If DR is 20%, for instance, only 5 DUs will share a single file while the remaining 20 DUs will share a wide range of files. Time required for each procedure as DR increases from 0% to 100% is depicted in Figure 8. This is, of course, a fantastic approach.

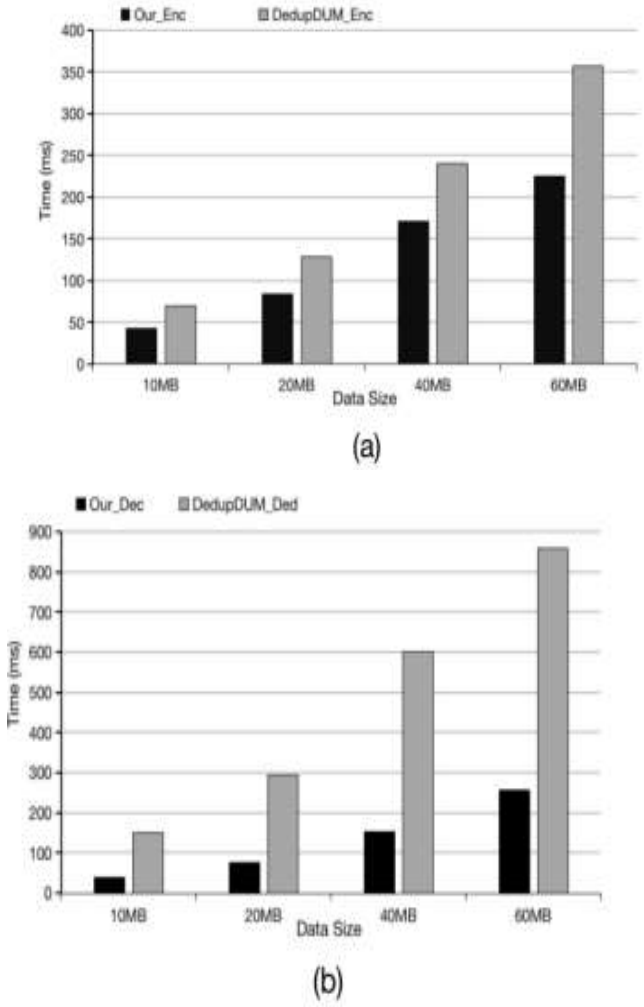


Fig. 4. Computation time for (a) encryption and (b) decryption.

When it's at its peak, the transmission time is 0.373 seconds and the time to remove duplicates is 0.251 seconds. Therefore, the time it takes to upload data can be drastically impacted by the suggested deduplication technique.

5. SECURITY ANALYSIS

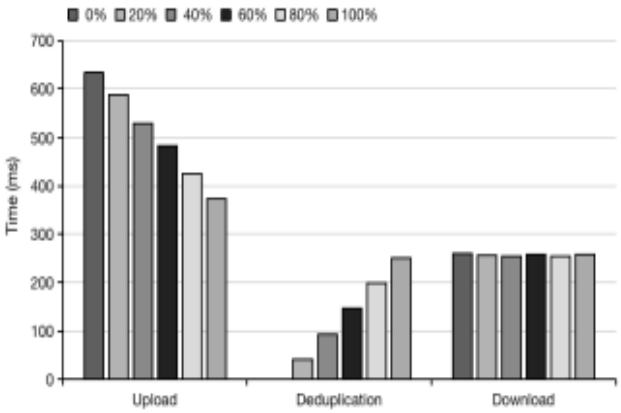
The security of our method is evaluated according to how well it safeguards confidentiality, consistency, ownership, revocation, and resistance to collusion in data.

A.Data Privacy

Raw data should be hidden from the Pub-CSP (which is trustworthy but nosy) and anyone else who has no business seeing it. Therefore, in addition to Pub-CSP and users who acquire false information, there are typically two other forms of dangers. The only thing that remains on Pub-CSP during an assault is the re-encrypted key of the authorised DU. Only the DU's private key can decrypt this key after it has been encrypted using PRE by Pri-CSP. Pub-CSP won't be able to decrypt plaintext using the cipher key if it has to rely on Pri-CSP and authorised users to generate money. Pub-CSP returns "Fid, CT, H(F),(u1,id, REKu1)" to an unauthorized user u2 who requests data F using (Fid, u1,id) (u1 is valid and using u2,id will fail the ownership check). There is no method for user u2 to obtain the data of F by decrypting CT using REKu1, as REKu1 can only be decrypted using user u1's private key. So both the well-intentioned but nosy Pub-CSP and users who shouldn't have access to the information are kept in the dark.

B.Data Consistency

Data deduplication methods are vulnerable to toxic attacks on tag consistency, which may be revealed to data owners upon decryption. Assuming that both u2 and u3 have access to the identical data FA, u2 can create a bogus ciphertext CTB using FB 6= FA and then upload it to Pub-CSP as CTA. When the actual individual has finished



The time it takes to calculate each procedure using a duplicate ratio is displayed in Figure 8. U3 wishes to upload FA as it expands, so it can check for duplicates by sending H(FA) to Pub-CSP. Pub-CSP requests that PriCSP remove any instances of H(FA) that already exist. After Pub-CSP does data deduplication, it provides (CTB, REKu3) to user u3, who then verifies that H(Decrypt(De(sku3, REKu3, CTB)) equals

H(FA). If it doesn't, u3's leaks to Pub-CSP will reveal what's going on. Our system ensures hence that all data is accurate.

C.Data Ownership Verification

By selecting a hash code, say the hash code of 10.5%-14.3% of F, at random from the collection, our method verifies who owns the data. Since Fx is chosen at random and the function H() cannot be inverted, calculating H(Fx) without access to the original plaintext is challenging.

D.Ownership Revocation

Only the users to whom ownership of Data F has been transferred should be able to view it. Using Pri-CSP as u0's owner, we ensure that ownership is revoked. The owner u0 re-encrypts the plaintext with the new symmetric key before re-uploading it to Pub-CSP, and updates the re-encrypted keys of the rest users when the data's original owner u1 revokes ownership or when other holders request to delete or modify their data. Therefore, the data's original owner will fail the access check and be unable to decrypt the most recent ciphertext using the original cipher-key.

E. Collusion Resistance

Given the reliability of Pri-CSP, we will discuss the dangers of Pub-CSP in collusion assaults and identify those responsible for them. If an unauthorized user, u1, uses a dishonest PubCSP to obtain the plaintext of data F, then Pub-CSP will request that Pri-CSP execute deduplication for u1 using spoofed data. PriCSP will verify u1's ownership of the data F before releasing the re-encrypted key REKu1. Since u1 does not have access to the plaintext, the ownership check will fail and the cipher-key will remain in Pub-CSP's possession. Second, even if criminals band together, they won't be able to crack the cipher-keys because each one is unique to its owner. Our approach does not allow for cooperation.

6.CONCLUSION

We solved the problem of how to safely and efficiently handle encrypted data with deduplication in a hybrid cloud architecture by posing the problem as a matter of ownership. While Pub-CSP handles storage, Pri-CSP acts as a proxy and a proprietor u0, handling dynamic ownership and deduplication. Our solution also demonstrates that only the data's rightful owner has access to the original, unencrypted data, and that only authorized parties can have access to the encrypted data. We know that our solution is effective, secure, and resistant to collusion and duplicate forging attacks because of our security analysis, comparison to prior work, and implementation-based performance evaluation.

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.usenix conf.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 Usenix 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 messagelocked 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.