Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 8, August : 2023

CLOUD-BASED UPLOADING AND DELETION OF COUNTING BLOOM FILTER DATA

#1**RAGI SARITHA,** ***Department of MCA,***
#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: Because of the rapid expansion of cloud storage, which may substantially reduce local storage overhead, an increasing number of data owners are preferring to outsource their data to a cloud server. Data owners who want to move cloud service providers must now use cloud data transfer since different cloud service providers provide varying levels of data storage service, including security, reliability, access speed, and pricing. As a result, the key problem that data owners confront is figuring out how to safely migrate data from one cloud to another while also removing the data from the original cloud. To address this issue, we present a novel counting Bloom filter-based technique in this paper. In addition to safe data transport, the proposed approach can provide permanent data deletion. Furthermore, the proposed system may be able to meet the requirements for public verifiability without the involvement of a trustworthy third party. Finally, we provide a simulated implementation to demonstrate the viability and effectiveness of our proposal.

Key words *— Cloud storage, Data deletion, Data transfer, Counting Bloom filter, Public verifiability.*

1. INTRODUCTION

The cloud computing model includes the ever-evolving concepts of parallel computing, distributed computing, and grid computing. Cloud storage has become one of the most widely used applications of cloud computing. A vast number of storage devices can be linked together in a network, simplifying data storage and access for both individuals and organizations. Customers can save significantly on on-premises hardware, software, and man-hours by storing their data in the cloud. The convenience and versatility of cloud storage have led to its widespread adoption in both private and professional contexts. This is why a growing number of individuals and organizations with restricted means are opting for cloud storage solutions. Data privacy, data integrity, data availability, and data erasure are just some of the additional security challenges that cloud storage must address because of the division of data custody and management that occurs when data is outsourced. The widespread adoption of cloud storage could be slowed if these concerns, especially those related to data deletion, are not adequately addressed. Deleting data at the end of its useful life affects how successfully the entire data life cycle may conclude. This is crucial for protecting confidentiality and anonymity in stored information. There has been a lot of focus on data encryption and protection, but far less on data deletion. There are a few challenges and bottlenecks that need to be fixed immediately, despite the fact that there have been a number of established ways to remove outsourced data in the cloud computing environment. Combining elements of parallel computing, global computing, and grid computing, "the cloud" is a novel approach to data processing. One of the most useful aspects of cloud computing is cloud storage, which streamlines the process of storing and retrieving data for commercial enterprises by connecting several storage devices across a network. By transferring data to a remote server in the cloud, customers can significantly reduce their expenditures on on-premises resources including computers, software, and human labor. The convenience and versatility of cloud storage have led to its widespread adoption in both private and professional contexts. This is why a growing number of individuals and organizations with restricted means are opting for cloud storage solutions. There are a number of security concerns unique to cloud storage, including data privacy, data accuracy, data availability, and data deletion. This is due to the fact that there is a separation between the data's owner and its manager. Many people may not adopt cloud storage until these issues, notably the disposal of obsolete data, are resolved. Ending the data lifecycle properly is crucial for maintaining data security and privacy, and this is especially true for the final phase in the data lifecycle, deletion. There has been a lot of focus on data encryption and protection, but far less on data deletion. While several approaches have been demonstrated for erasing rented data in the cloud computing setting, there are still some challenges and considerations that must be handled carefully.

Granular data deletion is currently unavailable in most systems. Data should be encrypted using a data key before being stored on a cloud server. If the key to decode the relevant data were removed, the matching ciphertext would also be lost, and the data might be permanently deleted. However, the entire leased file cannot be accessed without the data-decryption key. Most of the time in real life, the user wants to delete data. Both the user and the cloud server have to put in more effort than necessary because the user must update the entire outsourced file in order to delete a portion of it. Users necessitate granular, third-party data deletion mechanisms to rid themselves of irrelevant information.

2. RELATED WORK

Years of research have resulted in a wealth of literature on the subject of secure data deletion. There are essentially three types of data deletion techniques now in use. As the first and most efficient phase in the data deletion process, delinking is essential. Because of how the file system works, the link to the file is removed. The user is subsequently given a binary result (Success or Failure) that indicates whether or not the data deletion was successful. The data in a file remains unchanged even after its relationship to a disc is severed by unlinking. Therefore, a malicious actor need only employ the proper software on the proper disc to recover seemingly "lost" information. This means that the suggested solution about the removal of files may be incorrect. Overwriting information is another option for deletion. That can completely wipe out a file's contents. The core concept is to substitute random data for a solid medium. In 2010, Perito and Tsudik developed the Proofs of Safe Erasing technique. (PoSE-s). They demonstrate a selection of unpredictable patterns in their procedure for exchanging pairs of discs. After the data has been deleted, the same pattern is sent back to verify the deletion. Luo proposed a novel method of outsourcing data deletion that accomplishes the following. This piece is covered by the Creative Commons Attribution 4.0 license. To learn more about this license, please visit https://creativecommons.org/licenses/by/4.0/.
This manuscript has been accepted for publication in a subsequent issue of this journal and is currently undergoing the editorial process. Prior to being released, the data may undergo revisions. A

publically verifiable and effective fine-grained data eradication strategy for cloud computing has been released by Yang et al. IEEE AccessC, DOI:10.1080/ACCESS.2020.2997351, "A Scheme for Efficient and Publicly Verifiable Fine-Grained Data Deletion in the Cloud." "Yang et al." Yang's group Easy to Perform Once More You can delete the data by substituting random data for it. What this means is that overwriting was disguised as a means of data modification. The data deletion was verified using a challenge-response method in the meantime. However, if the network is too slow, the proof could fail. Several specifications, in particular [24], have also simplified the concept of deletion overwriting. Destroying the decryption key is the third option for erasing information. The primary goal is to destroy the encryption keys so that the corresponding ciphertext is no longer accessible. Since Boneh and Lipton's 1996 publication of the first cryptography-based data erasure solution, a lot of research has been done in this area. Using a trusted platform module (TPM), Hao et al. devised a method for securely erasing data.

3.SYSTEM DESIGN

This section elaborates on the specifics of our new approach. Keep in mind that the cloud service must verify the identity of the data owner. Let's say, for the sake of argument, that the data owner has completed the necessary identification steps and is now a legitimate tenant of clouds A and B.

For our purposes, it is important that data destruction and transfer be auditable. Before uploading information to cloud A, the data's owner encrypts it. The local copy is deleted after the storage results have been reviewed. The data's owner may decide to switch cloud storage providers and migrate some or all of their data from cloud A to cloud B in the future. The original data owner is curious as to how the transfer went. The data's owner requests deletion of the transferred data from cloud A and verifies that it was removed once the transfer completes successfully.

Second, a focus on realism Our new strategy is comprised of the aforementioned six algorithms.

Make (PKO, SKO), (PKA, KA), and (PKB, SKB) ECDSA public-private key pairs for the data owner, clouds A and B. The data owner then picks k secure hash algorithms, each of which converts each integer between 1 and N to a distinct cell in the Cipher Block Chain (gi: [1, N] [1, m]). In addition, the data's owner specifies a unique label for the file that will be uploaded to cloud A.

Data encryption

The data owner employs a robust encryption method before transferring the outsourced file to ensure its confidentiality.

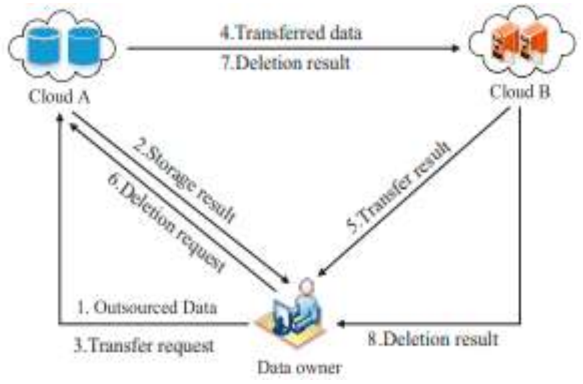

Fig. 1. The main processes of our scheme

i) The data owner generates an encrypted file C = Enck(F) using an encryption key k = H(tagf ||SKO), and then uses the IND-CPA safe encryption method to decrypt the file. After transferring and erasing data, the CBF must be filled, so the owner of the data separates the ciphertext C into n ′ blocks and inserts n n ′ random blocks into the n ′ blocks at random places. The information owner then records these arbitrary numbers in table P F. ii)) The data owner chooses a random index number ai and uses it to compute a hash value Hi = H(tagf ||ai ||Ci for each data block Ci. The transmitted data set, D = ((a1, C1),,(an, Cn),) was constructed as follows. The data owner then uploads D to cloud A with the identifier tagf for the file.

Data outsourcing

D is sent to the cloud once A creates a storage proof. The data's owner reviews the backup's output and then deletes the local copy.

Cloud A maintains data set D and generates a counting Bloom filter CBFs with the indexes (a1, a2, an), where i = 1, 2, and n, when it receives file tag tagf and data set D. D's index, Tagf, is currently stored on cloud A. In this case, Cloud A uses an ECDSA signature method and a timestamp to create evidence equal to (CBFs, Ts, sigs), which is then sent to the data owner. Sigs = SignSKA (storage||tagf ||CBFs||Ts) is the formula used to generate a signature in cloud A before the message is sent.

After receiving the proof of store, the data owner verifies the authenticity of the information. The data's owner verifies first that the signatures are authentic. If the CBF checks fail, the data owner will send out a "failure" message and choose two random indices from the range [a1, a2, and an]. The data owner will quit and declare failure if the CBFs are incorrect. In that case, the data's owner would be correct in erasing the local copy.

Data transfer

The data owner can transfer individual data blocks or the entire file from cloud A to cloud B when he decides to switch service providers.

To determine which chunks of data need to be transferred, the data's owner must first compile an index of block IDs. After that, the data's owner creates a signature using the formula sigt = SignSKO (transfer||tagf ||||Tt), where Tt is a timestamp. The data's owner then submits a request to move it to cloud A. You can express this as Rt = (transfer, tagf, Tt, sigt). Moreover, the data owner transfers the hash values "Hi" and "i" to cloud B.

Upon receipt, cloud A verifies the authenticity of the relocation request. When sending data blocks (ai, Ci)i to cloud B, cloud A generates a signature, denoted by sigta = SignSKA (Rt||Tt), and provides an error if Rt is invalid. So long as Rt is correct, cloud A will continue onward and upward.

Transfer check

Before notifying the data's rightful owner, cloud B will verify the transfer's success.

Cloud B initially verifies the transfer request Rt and the signature siga. If the condition Hi = H(tagf || ai || mi) holds, cloud B stays put; otherwise, it exits and broadcasts the error. If Hi = H(tagf || ai || Ci), then cloud B will send another request to cloud A to resend (ai, Ci). Step ii) is reached for cloud B if and only if Hi = H(tagf || ai || Ci).

Using the indexes (ai, i), Cloud B stores the blocks (ai, Ci)i and creates a new counting Bloom filter (CBFb). Success + tagf + Tt + CBFb = Sigtb, as cloud B figures out. Cloud B then provides the data's rightful owner with the proof of transfer. The notation for this is "(sigta, sigtb, CBFb)".

Upon receiving the alert, the data's rightful owner verifies the transfer's success. The data owner makes sure the signature sigtb is legitimate. The data owner picks fifty percent of the numbers at random to evaluate the performance of the counting Bloom filter CBFb. If everything checks out, the data's rightful owner will have assurance that the evidence of transfer is accurate, and cloud B will receive and store the data in question.

Data deletion

The data's owner can request deletion of blocks from cloud A once they've been copied to cloud B. Data signatures are initially generated by the data owner using the formula sigd = SignSKA (delete||tagf ||||Td), where Td is a timestamp. Next, the data's owner sends cloud A request to delete the information, formatted as Rd = (delete, tagf, Td, sigd).

Cloud A verifies Rd after receiving it. Cloud A will stop functioning and indicate failure if Rd is incorrect. In that case, cloud A will replace and delete the blocks of information "(ai, Ci)".A new counting Bloom filter (CBFd) is installed in cloud

A, and the old CBF indexes (aqq) are removed. After sending the deletion proof = (sigda, CBFd) to the owner of the data, cloud A generates a signature using the formula sigda = Sign(delete||Rd||CBFd).

The data owner verifies the signature sigda after receiving it. If sigda is not true, the data owner will exit with an error message printed. If this is the case, the data owner will select 50 percent of the indexes and verify that CBF(aq) = 0 and that aq belongs to CBFd. If the figures add up, the data's owner believes it's accurate. 1st Instance The fact that CBF(aq) = 0 indicates that q can be represented by at least one equation with the value 0. The counting Bloom filter CBFd does not contain aq.

4. SECURITY ANALYSIS

1. Data confidentiality

For security reasons, information in plaintext cannot be viewed by an adversary without the corresponding decryption key. On our platform, the data's owner uses the IND-CPA secure AES technique to encrypt the file. Meanwhile, the data decryption key is determined by the formula k = H(tagf || SKO), where H is a secure hash function and SKO is a hidden private key. This prevents the opponent from creating a key to decode the data. The decryption key is likewise under the owner's care. This prevents a future adversary from gaining access to the raw data by obtaining the decryption key.

2. Data integrity

Cloud B must be complete in order to accept the data due to the strict integrity requirements placed on it. Cloud B verifies the equation Hi = H(tagf || ai || Ci), where i, after receiving the transmitted data (ai, Ci) from cloud A and the hash values (Hi) from the data owner. Be aware that "Hii" is calculated by the data owner using a secure hashing technique. Since Hi = H(tagf || ai || C′ i), cloud A and the other bad guys can't generate a new data block (ai, C′ i) that satisfies this equation. In other words, cloud B can figure out what's going on and refuse to accept the data if it suspects that the data wasn't moved from cloud A to cloud B honestly or if hackers change the transmitted data blocks while the data is being moved. This ensures the data's safety throughout the transmission process.

3. Public verifiability

The accuracy of the deletion and relocation findings is investigated. The verifier can use the transfer proof and the transfer request RT to ensure the transfer outcome is correct. The validator first verifies the veracity of Rt. If Rt is correct, the data's owner specifically requested that it be transferred to cloud B. The validity of sigta and sigtb are then verified by the validator. Keep in mind that cloud B is deliberately incompatible with cloud A in order to deceive the data's owner. Therefore, the verifier must verify the authenticity of both signatures before accepting the sent result. If cloud B claims to have deleted the sent data, the verifier further examines the counting Bloom filter CBFb.

A verifier with deletion proof and a deletion request Rd can also validate the deletion's outcome. The validator's initial step is to determine if Rd. The data owner is under no obligation to delete any information if Rd is incorrect. If not, the verifier checks the signature sigda and the functionality of the counting Bloom filter CBFd. If all of the tests pass, the verifier accepts the evidence of deletion as valid. To verify the success of a deletion or transfer, the verifier does not require any personally identifying information. That is to say, our strategy satisfies the criteria for being subject to public scrutiny. The possibility of the deletion phase verification failing due to a counting Bloom filter false positive cannot be eliminated, although it can be reduced. The formula for the false-positive rate, Pf = (1 e kn/m) k, can be found in Ref. [31]. The likelihood of a false-positive result is minimized at k = ln2(m/n), or roughly

(0.6185)m/n. Our strategy calls for k to be 20, and for m/n to be 29. Since this is the case, Pf is extremely low, being close to 2, which is statistically indistinguishable from zero. Therefore, we believe the verifier can accurately verify the deletion outcome.

5. EFFICIENCY EVALUATION

Quite the opposite Here, we evaluate our strategy beside two alternatives. The following observations are derived from Table 1. All three options allow for verifiable data deletion. Second, our method, like the scheme, allows for proved data transmission and verification of the transferred data on a different cloud. Our plan and the scheme both lack a TTP, yet the scheme insist that one be implemented anyhow.

Calculations for encryption, generation, verification, G1 exponentiation, hashing, and paring are indicated in Table 2 by the corresponding letters E, S, V, Exp, H, and P. N is the total number of data blocks, and l is the number of data blocks that have been modified or removed. We skip complex calculations like multiplication and addition to keep things simple

Table 1. Functionality comparison

	Scheme[32]	Scheme[26]	Our scheme
TTP	√	x	x
Data integrity	x	√	√
Provable transfer	x	√	√
Verifiable deletion	√	√	√

Table 2. Complexity comparison

	Scheme[32]	Scheme[26]	Our scheme
Encrypt	$2\mathcal{E} + 4\mathcal{H}$	$n\mathcal{H} + 1\mathcal{E}$	$1\mathcal{H} + 1\mathcal{E}$
Storage	-	$n^2E + 2n\mathcal{P}$	$1\mathcal{S} + 1\mathcal{V} + 30n\mathcal{H}$
Transfer	-	$2\mathcal{S} + 2\mathcal{V} + 2l\mathcal{P} + (l+n)Exp$	$3\mathcal{S} + 3\mathcal{V} + 31l\mathcal{H}$
Deletion	$1\mathcal{S} + 1\mathcal{V}$	$(l+1)(\mathcal{S} + \mathcal{V}) + (l+n)Exp$	$2\mathcal{S} + 2\mathcal{V} + 30l\mathcal{H}$

Simulation results

On the same Linux box with 4GB of main memory and 3.30GHz Intel(R) Core(TM) i5-4590 processors, we utilize the OpenSSL library and the PBC library to mimic our scheme and schemes that came before it. Other values of k and m/n that we investigate include 20 and 29.

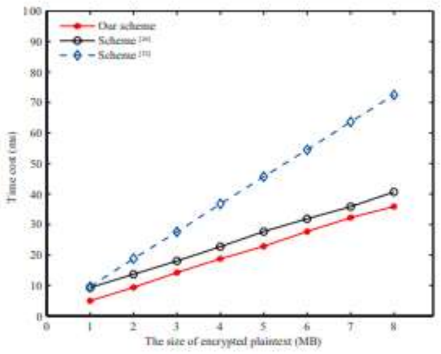

Fig. 2. The time cost of data encryption

On the same Linux box with 4GB of main memory and 3.30GHz Intel(R) Core(TM) i5-4590 processors, we utilize the OpenSSL library and the PBC library to mimic our scheme and schemes that came before it. Other values of k and m/n that we investigate include 20 and 29.

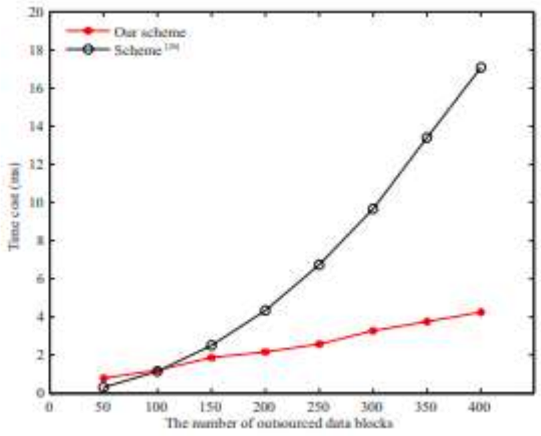

Fig. 3. The time of storage proof generation

The storage phase computing cost is a function of the time required to generate storage proofs and confirm storage results. The time required to locate storage verification is depicted in Figure 3. The growth rate of Yang et al.'s method is substantially higher than that of our own, but our method is much faster. This facilitates proof storage, which is a key benefit of our approach. Figure 4 depicts a comparison of performance for the data owner to review once they have verified the data storage method. Since our solution just requires simple hash calculations and a signature verification mechanism, we incur much less overhead than Yang et al. The method proposed by Yang et al., on the other hand, requires extensive calculations using bilinear coupling.

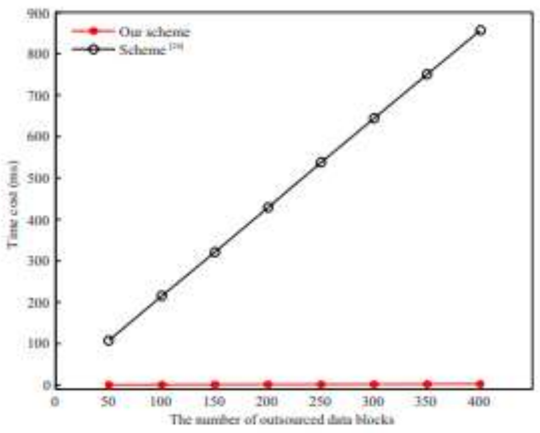

Fig. 4. The time of storage result verification

We are duplicating the data transfer by sending 80 data blocks rather than the usual 10. As may be seen in Figure 5, we simplify matters by fixing n at 400 and disregarding the transmission overhead. As more data units are transmitted, the time required to do so will increase. Moreover, ensuring the accuracy of the data using Yang et al.'s method requires several bilinear coupling calculations, whereas our method just requires a minimal number of hash value calculations. The time required to calculate the bilinear coupling is larger than that required to calculate the hash. This has improved the efficacy of our strategy.

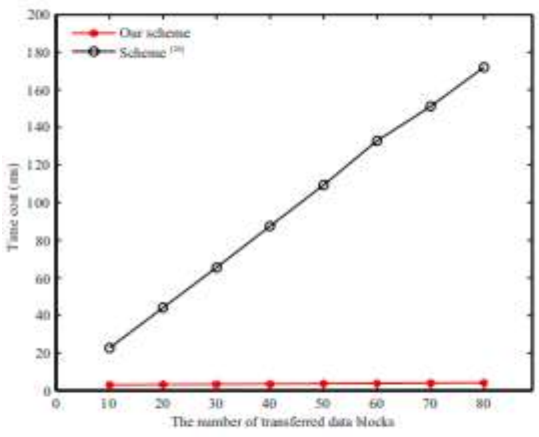

Fig. 5. The time cost of data transfer

When n is 400 and the data owner requests that the data be removed from cloud A, the evaluation of speed is depicted in Figure 3. Hao et al. proposed an approach that has a nearly constant time cost. Both our method and Yang et al.'s take more time when more data blocks are deleted, but Yang et al.'s system scales significantly more quickly. When there are more than 20 eliminated data units, Yang et al.'s procedure becomes extremely time-consuming. Therefore, we believe that our approach to erasing chunks of sent data is superior.

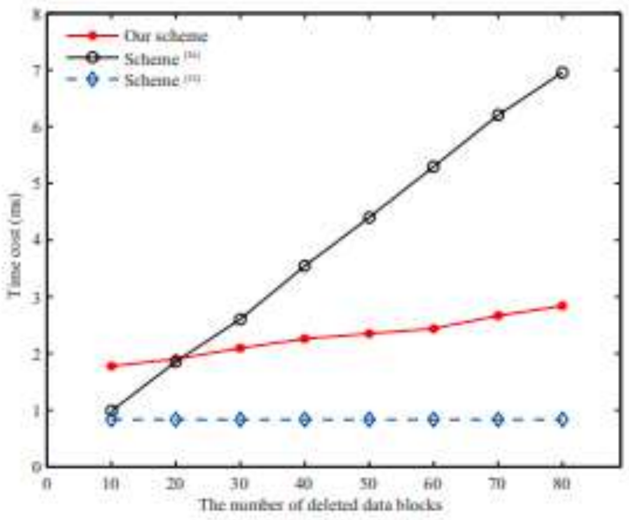

Fig. 6. The time cost of data deletion

6.CONCLUSION

The data's owner can't be confident the cloud server will relocate or delete files in good faith. To address this issue, we propose a CBF-based secure data transmission system that allows for auditable data deletion. Our strategy calls for cloud B to double-check the data for accuracy and completeness after transmission. In addition, cloud A needs to employ CBF to prove that data was destroyed after the fact. This verification that the data has been removed will be used by the data owner. This prevents cloud A from intentionally deceiving the data owner. In conclusion, our solution is both practical and secure, as evidenced by the simulation and security analysis results. upcoming obligations Our plan, like all other plans, has the ability to move data between two separate cloud computers. As cloud storage evolves, however, the data's owner may want to split leased information over multiple clouds. The data owner may be intentionally misled by the coordinated efforts of the multiple-target clouds. Therefore, more research is required to determine how data can be transferred between three or more clouds in a verifiable manner.

REFERENCES

[1] C. Yang and J. Ye, "Secure and efficient fine-grained data access control scheme in cloud computing", Journal of High Speed Networks, Vol.21, No.4, pp.259–271, 2015.

[2] X. Chen, J. Li, J. Ma, et al., "New algorithms for secure outsourcing of modular exponentiations", IEEE Transactions on Parallel and Distributed Systems, Vol.25, No.9, pp.2386–2396, 2014

[3] P. Li, J. Li, Z. Huang, et al., "Privacy-preserving outsourced classification in cloud computing", Cluster Computing, Vol.21, No.1, pp.277–286, 2018.

[4] B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions", Future Generation Computer Systems, Vol.79, pp.849–861, 2018.

[5] W. Shen, J. Qin, J. Yu, et al., "Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage", IEEE Transactions on Information Forensics and Security, Vol.14, No.2, pp.331–346, 2019.

[6] R. Kaur, I. Chana and J. Bhattacharya J, "Data deduplication techniques for efficient cloud storage management: A systematic review", The Journal of Supercomputing, Vol.74, No.5, pp.2035–2085, 2018.

[7] Cisco, "Cisco global cloud index: Forecast and methodology, 2014–2019", available at: https://www.cisco.com/c/en/us-/solutions/collateral/service-provider/global-cloud-index-gci/ white-paper-c11-738085.pdf, 2019-5-5.

[8] Cloudsfer, "Migrate & backup your files from any cloud to any cloud", available at: https://www.cloudsfer.com/, 2019-5-5.

[9] Y. Liu, S. Xiao, H. Wang, et al., "New provable data transfer from provable data possession and deletion for secure cloud storage", International Journal of Distributed Sensor Networks, Vol.15, No.4, pp.1–12, 2019.

[10] Y. Wang, X. Tao, J. Ni, et al., "Data integrity checking with reliable data transfer for secure cloud storage", International Journal of Web and Grid Services, Vol.14, No.1, pp.106–121, 2018.

[11] Y. Luo, M. Xu, S. Fu, et al., "Enabling assured deletion in the cloud storage by overwriting", Proc. of the 4th ACM International Workshop on Security in Cloud Computing.