International Journal of Technology, Leadership and Sciences (IJTLS)
ISSN: 3107-9083 www.ijtls.com Paper ID: TLS-26/V2/I2-006
DOI: https://doi.org/10.5281/zenodo.19123519

OPTIMIZING CLOUD STORAGE OPERATIONS WITH COUNTING BLOOM FILTER-BASED DATA MANAGEMENT

[#1]Dr. KISHOR KUMAR GAJULA, *Associate Professor, Dept of CSE,*
Mother Theressa College of Engineering & Technology, Peddapalli, Telangana.
[#2]Dr. KOTHAM SRIDHAR, *Associate Professor, Dept of CSE,*
Mother Theressa College of Engineering & Technology, Peddapalli, Telangana.
RISHI KRISHNA THODUPUNURI, *Software Developer*

ABSTRACT: The trend toward storing data on remote servers, rather than individual computers, is growing among data owners. The reason behind this is the rapid expansion of cloud storage, which could result in significant cost savings compared to local storage options. Since various providers provide varied degrees of data storage service—in terms of security, dependability, access speed, and cost—cloud data transfer is now a necessity for data owners who wish to switch cloud service providers. Finding a secure way to transfer data to another cloud while simultaneously removing it from the first cloud is, thus, the largest challenge for data owners. In this research, we tackle this problem by introducing a new counting Bloom filter-based method. Secure data delivery and permanent data deletion are both possible with the suggested strategy. It is possible that the suggested approach satisfies the public verifiability requirement even in the absence of a reliable third party. Lastly, by executing a virtual implementation of our concept, we demonstrate its potential functionality.

***Keywords**:* Cloud storage, Data deletion, Data transfer, Counting Bloom filter, Public verifiability.

1. INTRODUCTION

The ever-evolving concepts of grid, distributed, and parallel computing form the basis of cloud computing. One of the most prevalent applications of cloud computing nowadays is cloud storage. It can be made easier for individuals and organizations to store and retrieve data by connecting numerous storage devices to a network. Customers can significantly reduce costs on on-premises hardware, software, and labor by migrating data to the cloud.

The convenience and adaptability of cloud storage have made it a popular choice for businesses and individuals alike. This is why cloud storage is becoming increasingly popular among individuals and businesses with limited resources. The separation of data ownership and administration occurs when data is outsourced. Because of this, cloud storage security concerns like data erasure, data availability, data integrity, and privacy are exacerbated. Cloud storage might not gain much traction unless these issues, particularly those pertaining to data deletion, are adequately addressed. Destroying data when it's no longer needed impacts the efficiency of the data life cycle as a whole. For the sake of data privacy and anonymity, this is crucial. Encrypting and safeguarding data is far more important than erasing it. Though there are many time-tested methods for erasing data stored in the cloud, there are still certain issues and hurdles that must be addressed without delay.

International Journal of Technology, Leadership and Sciences (IJTLS)
ISSN: 3107-9083 www.ijtls.com Paper ID: TLS-26/V2/I2-006
DOI: https://doi.org/10.5281/zenodo.19123519

A new approach to data processing, "the cloud" incorporates elements of global computing, parallel computing, and grid computing. Cloud storage is a great feature of cloud computing. It facilitates data storage and retrieval for organizations by linking several storage devices across a network. By transferring their data to a remote cloud server, customers can significantly reduce their spending on on-premises hardware, software, and personnel. Cloud storage has become ubiquitous in both individual and corporate settings due to its user-friendliness and adaptability. This is why cloud storage is becoming increasingly popular among individuals and businesses with limited resources.

Cloud storage raises many security problems, such as those pertaining to the confidentiality, integrity, availability, and erasure of stored data. This is due to the fact that data ownership and the management are distinct entities. Until these issues, particularly the one about the removal of outdated data, are resolved, many users may refrain from utilizing cloud storage. Ensuring the correct end of the data lifecycle is crucial for data security and privacy. This is particularly the case when it comes to deletion, the last stage of data lifecycle. Encrypting and safeguarding data is far more important than erasing it. Although there are a number of options, some issues and considerations must be given serious consideration when deleting rented data in a cloud computing setting.

The majority of systems do not yet provide incremental data deletion. Data encryption using a data key is a must before storing sensitive information on a cloud server. The data might be lost indefinitely if the key required to decipher the pertinent information and the ciphertext were to be taken away.

The entire rented file cannot be accessed without the decryption key. Customers typically wish to erase data in the real world. It takes more effort than necessary for the user and the cloud server to remove a portion of an outsourced file. In order to remove unnecessary data, users require granular, third-party solutions.

2. RELATED WORK

The topic of safely erasing data has been extensively covered in literature thanks to years of research. There are now primarily three methods for erasing information. Data erasure begins with unlinking it, the first and foremost stage.

The file system's operation renders the file's link useless. The user is subsequently informed of the data deletion's success or failure by seeing a binary result. When you unlink a file from a disc, the connection between the two ends, but the data within the file remains intact. Therefore, all it takes for an evildoer to recover data that has been erased is the correct software on the correct disc. This suggests the proposed method for erasing the file might not work. Another method of erasing data is overwriting it.

The data in a file can be entirely erased using that. Substituting random data for a solid medium is the central principle. In 2010 (PoSE-s), Perito and Tsudikdeveloped the Proofs of Safe Erasing method. They exhibit a great deal of bizarre pattern recognition while exchanging disc pairings. The same pattern is sent back once the item has been erased to confirm its deletion. A novel approach to outsourcing data deletion was developed by Luo, which achieves the following objectives. The Attribution 4.0 license from Creative Commons governs this work. Visit

International Journal of Technology, Leadership and Sciences (IJTLS)
ISSN: 3107-9083 www.ijtls.com Paper ID: TLS-26/V2/I2-006
DOI: https://doi.org/10.5281/zenodo.19123519

Editing is under underway for this piece of work, which has been approved for publication in an upcoming edition of the journal. Prior to its release to the public, the data may undergo editing. "A Scheme for Efficient and Publicly Verifiable Fine-Grained Data Deletion in the Cloud"

Yang et al., is a publically verifiable and effective approach for erasing fine-grained data in the cloud. Repetition is a Breeze for Yang's Team, "Yang et al." Simply substituting it with random data will remove the data. As a result, information was altered while feigning to replace it. Concurrently, a challenge-response system was used to guarantee data destruction. However, if the network is too sluggish, the proof can fail. Deletion overwriting has been complicated, but many standards have simplified it, most notably.

The third method for erasing data is to destroy the decryption key. The primary objective is to eliminate any potential means of decryption by destroying the encryption keys. Several studies have been conducted in this field since Boneh and Lipton introduced the initial data erasing method based on cryptography in 1996. Using a TPM, Hao and his colleagues devised a method to securely erase data.

3.SYSTEM DESIGN

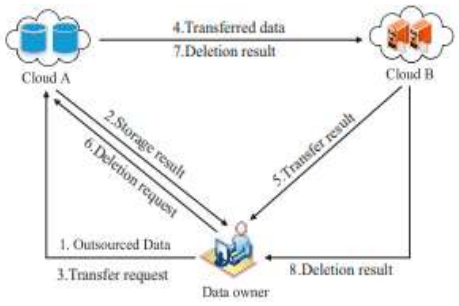
The details of our new approach are laid forth in this section. The data owner's identity must be verified by the cloud provider. Assume for the sake of argument that the data owner has successfully completed the necessary steps to verify their identity and is thus a valid tenant of clouds A and B.

Data transmission and deletion must be verifiable if we are to accomplish our goals. Data is encrypted before being transferred to cloud A by the proprietor. After the results from the storage examination are reviewed, the local copy is deleted. The data owner has the option to move all or part of their data from cloud A to cloud B when they switch cloud storage providers. The prior data owner wants to know where the transfer stands. Once the transfer is successful, the data owner can confirm the deletion of their data from cloud A.We stress reality as a second point. Our innovative method is based on the six algorithms listed above.

Create ECDSA public-private key pairs for A and B's data using the following formats: (PKO, SKO), (PKA, KA), and (PKB, SKB). Every integer from 1 to N is transformed into a separate cell in the Cipher Block Chain (gi: [1, N] [1, m]) using one of the k secure hash algorithms chosen by the data owner. For the purpose of uploading it to cloud A, the data owner gives the file a unique name.

Data encryption

Prior to transmission, the data owner ensures the file's privacy by using a robust encryption mechanism.



International Journal of Technology, Leadership and Sciences (IJTLS)
ISSN: 3107-9083 www.ijtls.com Paper ID: TLS-26/V2/I2-006
DOI: https://doi.org/10.5281/zenodo.19123519

Fig. 1. The main processes of our scheme

An encrypted file C = Enck(F) is created by the data owner using an encryption key k = H(tagf ||SKO). The IND-CPA secure method is then used to decode the file. After dividing the ciphertext C into n′ blocks, the data owner can insert n n′ random blocks anywhere within the n′ blocks. After moving or erasing data, the CBF must be completed. As a next step, the knowledgeable person enters these random integers into table P F.

The data owner then uses a random index number ai to calculate a hash value Hi = H(tagf ||ai ||Ci for each data block Ci. Lastly, the data owner uploads D to cloud A using the file's identified tag.

Data outsourcing

The data owner deletes the local copy after examining the backup's results, and D is sent to the cloud as soon as A generates storage proof.

Cloud A builds a counting Bloom filter CBFs using the indices (a1, a2, an), where i = 1, 2,..., n, after receiving file tag tagf and data set D. Cloud A also keeps data set D. Cloud A now stores the tagf index. Here, Cloud A uses an ECDSA signature method and a timestamp to create evidence that is identical to (CBFs, Ts, sigs). The data owner is thereafter sent the evidence. To create a signature before sending the message, cloud A uses the following formula: Sigs = SignSKA (storage||tagf ||CBFs||Ts).

The data owner verifies the information's accuracy upon getting the store evidence. Before anything else, the data owner makes sure the signatures are legitimate. In the event that the CBF tests do not succeed, the data owner will choose two indices at random from the set [a1, a2, and an] and send out a "failure" notice. In the event that the CBFs are incorrect, the data owner will announce their resignation and admit defeat. The owner of the data would be within their rights to remove the local copy.

Data transfer

Transferring the entire file or specific data blocks from cloud A to cloud B is an option for the data proprietor when they decide to switch service providers.

The data owner needs to gather a list of block identifiers before they can figure out which data segments need to be transmitted. The data owner generates a signature using the formula sigt = SignSKO (transfer||tagf ||||Tt), where Tt is a timestamp. After then, the data owner makes a request to move it to cloud A. For example, you can write Rt as (transfer, tagf, Tt, sigt). "Hi" and "i" are the hash values that the data owner transmits to cloud B.

Upon receiving the relocation request, Cloud A checks its legitimacy. When data blocks (ai, Ci)i are sent to cloud B, cloud A creates a signature called sigta = SignSKA (Rt||Tt). The invalidity of Rt causes Cloud A to return an error. Cloud A will keep moving upwards assumi ng that Rt is correct.

Transfer check

The success of the transfer will be confirmed by Cloud B before notifying the rightful owner of the data.

Cloud B must first confirm the authenticity of the transfer request Rt and the signature siga. Cloud B stays put if and only if Hi = H(tagf || ai || mi) is true; if not, it leaves and sends out an error message. For cloud A to retransmit (ai, Ci), cloud B will order it to do so if Hi = H(tagf

International Journal of Technology, Leadership and Sciences (IJTLS)
ISSN: 3107-9083 www.ijtls.com Paper ID: TLS-26/V2/I2-006
DOI: https://doi.org/10.5281/zenodo.19123519

|| ai || Ci). Step ii can only be executed by Cloud B if and only if Hi is equal to H(tagf || ai || Ci).

Using the indices (ai, i), Cloud B generates a new counting Bloom filter (CBFb) and keeps the blocks (ai, Ci)i. After considering all of the variables, Cloud B has concluded that Sigtb= Success + tagf + Tt + CBFb. After that, the rightful data owner receives the proof of transfer from Cloud B. The notations "sigta, sigtb, CBFb." serve as proof of this.

The data's rightful owner checks the transfer's success upon receiving the notification. The data owner checks if the signature is genuine. The data owner chooses half of the numbers at random to test how well the counting Bloom filter CBFb works. Cloud B will obtain and store the contentious data if everything goes according to plan. The data's rightful owner will double-check the authenticity of the transfer paperwork.

Data deletion

Data owners have the option to ask for the removal of blocks from cloud A once they've been transferred to cloud B. Sigd = SignSKA (delete||tagf ||||Td), where Td is a timestamp, is the formula that the data owner uses to produce data signatures initially. The data owner then sends a deletion request to cloud A with the parameters Rd = (delete, tagf, Td, sigd).

After receiving Rd, Cloud A checks it. Cloud A will stop working and show an error message if Rd is wrong. Next, Cloud A will change and remove the "(ai, Ci)" data blocks. It updates the counting Bloom filter (CBFd) and gets rid of the old CBF indexes (aqq). Next, Cloud A uses the formula sigda = Sign(delete||Rd||CBFd) to create a signature, and then it sends the deletion proof = (sigda, CBFd) to the data owner.

The signature is verified by the data owner upon receipt. An error warning will be sent and the data owner will resign if sigda is false. In order to ensure that aq is present in CBFd and that CBF(aq) equals 0, the data owner will choose half of the indexes. The data owner considers it accurate if the numbers match up. For every given value of q, there is an equation that can be expressed as a perfect square. That is proven by the fact that CBF(aq) = 0. No aq is present in the counting Bloom filter CBFd.

4. SECURITY ANALYSIS

1. Data confidentiality

Data in plaintext is insecure because an attacker needs the decryption key to access it. The data owner encrypts the file on our platform using the IND-CPA secure AES technique. For data decryption, the solution to the equation k = H(tagf || SKO) is given. A secure hash function is denoted by H, and a confidential private key is represented by SKO. Because of this, the enemy can't generate a key that can access the data. The data decryption key is also the owner's responsibility. This makes it such that an intruder can't get the decryption key and read the data in its raw form.

2. Data integrity

Data cannot be received until it is fully complete due to the strict integrity restrictions enforced by Cloud B. Cloud B checks the data provided by Cloud A and the hash values from the data owner to ensure that Hi = H(tagf || ai || Ci) is correct. No one can create a new data block (ai, C′ i) that meets the condition Hi = H(tagf || ai || C′ i) because the data owner uses a secure hashing algorithm to identify "Hii." Cloud A and its enemies are secure. Cloud

International Journal of Technology, Leadership and Sciences (IJTLS)
ISSN: 3107-9083 www.ijtls.com Paper ID: TLS-26/V2/I2-006
DOI: https://doi.org/10.5281/zenodo.19123519

B can reject data if it determines that the transfer was unfair or if it finds evidence that hostile actors modified the transmitted data blocks. This safeguards the data while it is being transmitted.

3. Public verifiability

The accuracy of the deletion and relocation outcomes is assessed. It is possible for the verifier to confirm the accuracy of the transfer outcome by validating the transfer evidence and the transfer request RT. Before doing anything else, the validator checks that Rt is correct. The data owner specifically asked for its transfer to cloud B, if Rt is correct. The validator then checks sigta and sigtb to make sure they are legitimate. Keep in mind that the data owner is being tricked by cloud B's purposeful incompatibility with cloud A. The verifier has to confirm that both signatures are genuine before they may accept the result. Cloud B's assertion of data destruction prompts the validator to conduct a more thorough inspection of the counting Bloom filter CBFb.

The result of the deletion can also be determined by a verifier who has deletion proof and a deletion request Rd. Finding out if Rd exists is the validator's job. If Rd is wrong, the data owner is under no obligation to delete any material. In that case, the verifier checks the sigda signature and the CBFd counting Bloom filter to see how well they work.

The verifier will consider the proof of deletion legitimate if all checks are successful. No personally identifiable information is needed by the verifier in order to validate the successful completion of a transfer or deletion. What this means is that our plan is up to scratch when it comes to public review. While it is possible to reduce, it is not possible to totally eliminate, the possibility of failure during the deletion phase verification due to a counting Bloom filter false positive. Pf = (1 e kn/m) k is the formula for the false-positive rate, which can be found in Reference.

The likelihood of getting a false positive is decreased by setting k = ln2(m/n), which is close to (0.6185)m/n. Our method requires that k be equal to 20 and that m/n be equal to 29. So, Pf is about 2, which is the same as 0 from a statistical standpoint, which is quite low. In our opinion, the verifier is capable of accurately validating the deletion outcome.

5. EFFICIENCY EVALUATION

Our approach is compared to two others. From Table 1 we can deduce the following observations. Data can be verifiably eliminated using any of the three approaches. Second, our methodology allows data transmission and verification across a separate cloud, just like the scheme. Despite their similarities, neither the plan nor the scheme actually carry out the TTP that is specified in them.

Table 2 shows the results of the encryption, generation, verification, G1 exponentiation, hashing, and pairing operations as E, S, V, Exp, H, and P, respectively. l is the number of changed or removed data blocks, and N is the total number of data blocks. For the sake of clarity, we will not be discussing more advanced mathematical concepts like addition and multiplication.

Table 1. Functionality comparison

International Journal of Technology, Leadership and Sciences (IJTLS)
ISSN: 3107-9083 www.ijtls.com Paper ID: TLS-26/V2/I2-006
DOI: https://doi.org/10.5281/zenodo.19123519

	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

A Linux computer with 4GB of RAM and 3.30GHz Intel(R) Core(TM) i5-4590 processors is used to execute our scheme and additional schemes using the OpenSSL and PBC libraries. At 20 and 29, we also check the values of k and m/n.

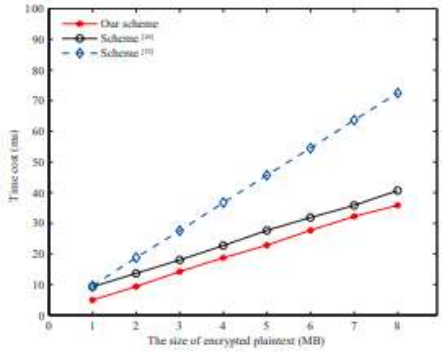


Fig. 2. The time cost of data encryption

On a Linux machine with 4 GB of RAM and an Intel(R) Core(TM) i5-4590 processor running at 3.30GHz, we run our scheme and earlier schemes using the OpenSSL and PBC libraries. At 20 and 29, we also look at the k and m/n values.

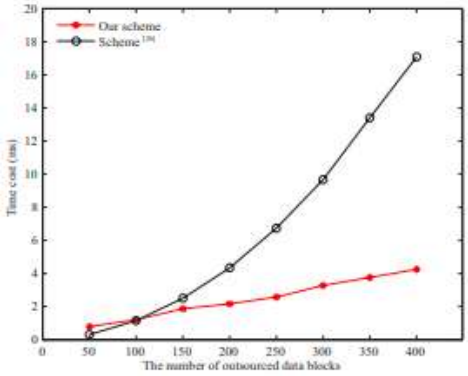


Fig.3. The time of storage proof generation

International Journal of Technology, Leadership and Sciences (IJTLS)
ISSN: 3107-9083 www.ijtls.com Paper ID: TLS-26/V2/I2-006
DOI: https://doi.org/10.5281/zenodo.19123519

The time needed to create storage proofs and validate storage results is what determines the cost of computing the storage phase. Finding storage verification takes time, as seen in Figure 3. Although our methodology is slightly rapid, Yang et al.'s methodology shows a far better growth rate.

One major benefit of our method is that it makes evidence storage easier. Once the data owner has confirmed the data storage technique, they can review the performance comparison shown in Figure 4. Because our methodology just necessitates a system for signature verification and fundamental hash computations, we suffer far lower costs compared to Yang et al. On the other hand, Yang et al.'s method requires a large number of bilinear coupling calculations.

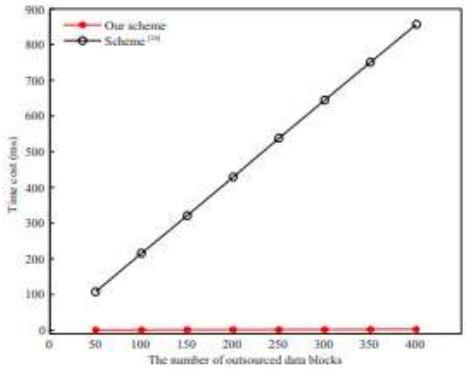


Fig 4. The time of storage result verification

The data transmission is being replicated by sending 80 data segments rather than the usual 10. Figure 5 shows that we can simplify things by ignoring the transmission overhead and setting n to 400. Additional time will be needed for the transfer of data units. In contrast to Yang et al.'s method, which involves several bilinear coupling calculations, our solution only needs a small number of hash value computations to ensure data integrity. In comparison to the hash calculation, the bilinear coupling computation is time-consuming. Our strategy is now more effective as a result of this.

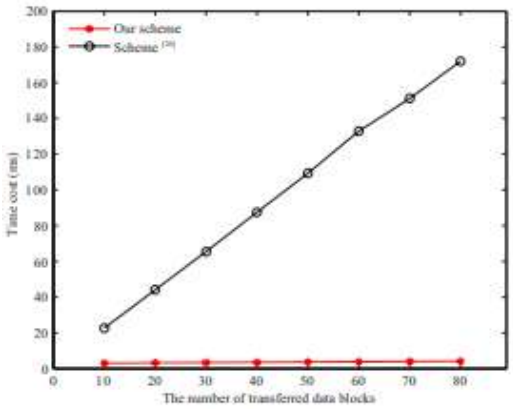


Fig. 5. The time cost of data transfer

The speed test is shown in Figure 3 when n equals 400 and the data owner demands the deletion of data from cloud A. The method developed by Hao et al. has a uniform time cost that is maintained throughout. While both our approach and Yang et al.'s system undergo lengthier processing times as the number of data blocks eliminated grows, the scalability of Yang et al.'s system is significantly better. When more than 20 data units need to be deleted,

International Journal of Technology, Leadership and Sciences (IJTLS)
ISSN: 3107-9083 www.ijtls.com Paper ID: TLS-26/V2/I2-006
DOI: https://doi.org/10.5281/zenodo.19123519

the procedure by Yang et al. becomes tedious. We claim to have the best method for data extraction.

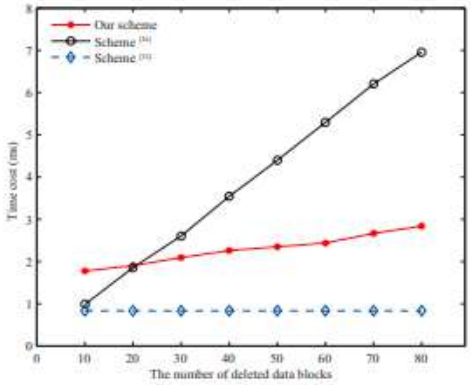


Fig.6.The time cost of data deletion

6. CONCLUSION

Data owners have no guarantee that files stored on a cloud server will be safely moved or deleted. Our solution to this problem is a CBF-based secure data transmission system that allows for the erasure of data with proof. Cloud B will check the data for completeness and accuracy after it is transmitted, following our strategy. Cloud A needs to use CBF to prove that the data was later deleted. This confirmation of deletion will be utilized by the data owner. Cloud A cannot wilfully mislead the data owner because of this. Finally, the results of the security evaluation and the simulation show that our method is safe and works. Future commitments. Our approach can move data between cloud servers just like any other approach. On the other hand, data owners may choose to spread their data across numerous clouds as cloud storage capabilities improve. Deliberately misleading the data owner is possible due to the coordinated maneuvers of the multiple-target clouds. Finding a trustworthy way to move data between three or more cloud environments requires additional research.

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"

[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.