International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I10P224

Volume 70 Issue 10, 256-261, October 2022

Original Article

An Efficient Data Deduplication with Block Sorting Encoding Technique for Cloud Storage Environment

N. Mageshkumar[1], L. Lakshmanan[2]

[1,2]*Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India.*

[1]*Corresponding Author : mageshkumarns19@gmail.com*

Received: 28 July 2022 Revised: 22 September 2022 Accepted: 12 October 2022 Published: 20 October 2022

Abstract *- Due to the advances in cloud computing (CC) applications, Cloud storage providers continuously require new techniques for minimising redundant data and increasing space spacing. By saving an individual copy of replica data, cloud service providers (CSP) significantly minimize the saving and communication cost. But, the use of deduplication poses several security issues. This paper employs a new scheme of secure data deduplication using the Perfect Hashing (PH) and block sorting encoding (BSE) technique called PH-BSE. The presented PH-BSE technique assumes the popularity of the data blocks and controls the characteristics of PH for ensuring block-level deduplication and data confidentiality. Afterwards, a compression technique named the BSE technique was used to compress the data segments to handle the storage efficiency effectively. The BSE algorithm establishes the relation to other compression techniques and attempts to enhance the compression efficiency. It considers the documents in blocks, which may be as large as the entire file and reorders the document. A detailed experimental analysis takes place to verify the superior nature of the proposed methodology.*

Keywords *- Block sorting encoder, Cloud computing, Deduplication, Encryption, Perfect Hashing.*

1. Introduction

The network and multi-user storage modules have developed rapidly through extensive data generation. By considering data security, the migration of users to remote locations can be eliminated. Some of the traditional solutions undergo encryption before leaving the administrator's location. From the security point of view, this method prevents the storage provider from using storage efficiency performance, like compression and deduplication, enabling better resource application and minimum cost. Client-side data deduplication approves that similar content initialization applies network bandwidth and memory space of individual uploads. Deduplication is employed by cloud backup providers (Bitcasa) and various cloud services (e.g. Dropbox). Inopportunely, encrypted data is pseudorandom and cannot be deduplicated. Consequently, existing methods have to detriment security or storage efficiency.

The function of storage efficiency is compression and deduplication, which provides storage providers with the best application of the storage backend and the capability to facilitate massive users with similar architecture. Data deduplication is the function where a storage provider records an individual copy of a file applied by various users. It is operated in 4 diverse deduplication stages according to the presence of deduplication at the client side (before upload) or server side, and deduplication occurs at file or block levels. It is one of the remarkable modules accelerated at the client side and stores the initialized bandwidth.

Client-side deduplication reduces the need for server-side processing, reducing the cost of providing remote storage to the general public. This is why services like Dropbox and Memopal have become so popular. On the contrary, data deduplication is a primary factor in abandoning cloud storage and cloud backup. End-to-end encryption may be secure, but it comes at a performance cost. Data is encrypted at both the source and the destination. It is called end-to-end encryption. The security measures taken in response to the presence of leftover unencrypted data [1] and the expansion of industry-specific legislation and regulations have significantly impacted its development. Since the authentic decryption key determines whether or not two cipher-texts correspond to similar plaintext, file deduplication is not applicable when using semantically secure encryption. The distribution of encryption keys is a common feature of trivial solutions, which raises the bar for security. Thus, end-to-end encryption should be implemented in reformed storage systems to accommodate the current disk or user ratio. Implementing storage efficiency features like deduplication that doesn't compromise end-to-end security is difficult.

The study group offered many deduplication models [2, 3, 4], alluding to a method of allowing deduplication by sanctioned cuts to the use of storage resources [5, 6]. Harnik et al. [7] predicted several methods that result in data leverage at client-side deduplication, and massive approaches

do not reveal these attacks. The assaults described in [8, 9] can be defended against using the permission of ownership technique. Unfortunately, it is a realistic means of protecting user privacy when dealing with a criminal or honest but nosy cloud service provider. Douceur et al. [10, 11] used a cryptographic primitive called convergent encryption, protecting sensitive information while eliminating duplicates. This method uses a symmetric encryption model to encrypt plaintext using derived solutions. Semantic security cannot be provided via convergent encryption because of the risk of content-guessing attacks.

Convergent encryption was then formally described by Bellare et al. [18] under the label message-locked encryption. Marks offers a privacy analysis of message-locked encryption, which protects private information from being intercepted but falls short of achieving semantic security. Client-side deduplication in a bounded leakage situation is made possible by the PoW method presented by Xu et al. [13]. However, the issue of minimal entropy files is not reported, even though it gives security approval in an arbitrary oracle technique for the derived solution. Lately, Bellare et al. [20] designated DupLESS server-assisted encryption for deduplicated storage. Likewise, the solution exploits a protected component for a key generation as part of an enhanced convergent encryption method. The recently introduced models necessitate protected client-side deduplication, while DupLESS only offers the possibility of secure server-side deduplication.

This work offers an effective, secure compressive data deduplication utilizing the Perfect Hashing (PH) and block sorting encoding (BSE) approach named PH-BSE. To protect the privacy and provide deduplication at the block level, the given PH-BSE method regulates PH's features while assuming the popularity of data blocks. In addition to deduplication, we further compress the data segments through the BSE approach using burrows wheeler transform (BWT) for available storage. Extensive experimental analysis is performed to prove the superiority of the suggested methodology.

2. The Proposed Model

Data deduplication and compression are two main stages that operate on the proposed model. The PH technique is applied at the initial stage of the data deduplication process. Followed by, BSE based compression process takes place.

2.1. PH-based Data Deduplication

The inherent incompatibility between encryption and deduplication and previous solutions is composed of massive limitations. CE was assumed to be highly consistent for secured deduplication; however, it is clearly susceptible to diverse attacks. Therefore, CE does not apply to data confidentiality and requires a robust encryption mechanism. Also, it is pointed out that the data might require diverse protection levels based on popularity, and the data segment is developed as “popular” when it is higher than t users (where t denotes the popularity threshold). The “popularity” of a block is defined as the simulation of deduplication. In line with this, a data segment is unfamiliar when it comes to t users. A simple distinction is applied and monitored by well-known data, which does not necessitate a similar level of defense as unknown data and projects various encryption models for familiar and detested data. Besides, a secret file has sensitive data, like the list of usernames and passwords, that requires maximum robustness. Familiar data is secured through CE to activate source-based deduplication, while ostracized data has to be saved with an efficient encryption method. If the unpopular data segment becomes more familiar, then the encrypted data segment has been transmitted into Convergent Encrypted (CE) form to activate deduplication.

In this approach, the encryption of unpopular data blocks and symmetric encryption with the help of a random key offers the maximum level of protection at the time, enhancing processing costs on the client side. When a client desires to upload a data segment, it is presented with a popularity degree to process the encryption task. The client initially seeks a CE model of data saved at a CSP. If these data segments exist, the client finds the familiar data segment and deduplicates it. Alternatively, the client encrypts i symmetric encryption model. It optimizes the cost of encrypting and initializing processes on the user side. Hence, a benchmark lookup solution for the CEdata segment (CE-DS) states that the CEDS-ID, which determines the data, is highly violated. A data segment is a simple issue as the ID is applied as the input for the lookup query, which leads to serious data leakage. Additionally, the client requires a secured “popularity transition”, which is simulated with a data segment.

The challenge of identifying levels of popularity may be described as follows: The client wishes to ascertain, provided a data segment D and an identifier ID, whether or not the ID belongs to a collection P of IDs for popular data segments that have been stored at an untrusted CSP. The client has been supplied with these two pieces of information. The fact that no information needs to be disclosed for CSP while ID/P is performed may be relevant. It is frequently cited as an example of a private-set intersection (PSI) issue. As a direct consequence of this, older methods incur significant costs in terms of both processing and transmission when dealing with larger data sets. The recovery of confidential data in a non-public context, often known as "private information retrieval," is one potential approach to resolving this problem (PIR).

Hence, applying PIR improves 2 major issues: initially, it incurs a prominent communication overhead; then, PIR is

developed for deriving individual elements for each query, while an effective protocol for popularity verification enables the presence of diverse data segment IDs simultaneously. Thus, complex cryptographic primitives such as PSI and PIR recommend a protective approach for popularity detection that depends upon a lightweight building block named PH. It mainly concentrates on resolving the issue by deploying a new secured lookup protocol, according to PH, explained in the upcoming section.

The popularity detection solution uses the PH, which allows for an input dataset and investigates a collision-free hash function [21], which matches the input to a group of integers. This function is named after its collision-free nature. The process of producing PHFs as mapping IDs of CE popular blocks is recorded at the CSP, which also carries out the implementation of the procedure. The completed PHF is then encoded into a file before being transmitted to the user for usage. The user searches for a new lookup solution in the encrypted popular block IDs recorded at the CSP, as shown in Figure 1. The user must encrypt the block to acquire CE(D) for novel block D, and an unkeyed hash algorithm, such as SHA-3, must be used to ascertain the ID. After all, is said and done, the client obtains the search index for the novel block by calculating the PHF across ID. The integer serves as one of the lookup query's inputs when the client is given it. Following the acquisition of the CSP, the lookup query incorporates and provides the CE popular block (CEPB-ID) ID, which is filed away under.

Here, the client simply identifies the popularities of data segments relating to the ID and evaluates by obtained CSP: when the 2 IDs are the same, then D is familiar. It is a significant objective for preventing the CSP from establishing the data of block D in case of an unknown value. It is accomplished and improvised with the secured version of PHF, where CSP is not suitable to get the input of PHF from result$i$. And it defines that the PHF should produce effectively distributed collisions for the unknown block.

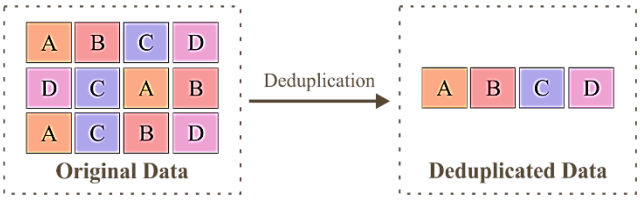

Fig. 1 Deduplication Process

Though the client can predict the popularity of a block securely, it still requires a popularity transition where the block attains a threshold t, and CEPB should be uploaded to the CSP. As the client is unaware of the duplicate copies of similar blocks, a model is used to monitor the unpopular data blocks. Then, the client becomes independent of the CSP due to the untrusted CSP. Finally, the outcome of a popularity check is not positive; after that, the user updates the IS based on transmitting CEPB-ID, which is symmetrically encrypted. When the block is familiar by reaching the threshold t, the popularity transition is simulated, and the user is pointed out of uploading the CEPB, where the deduplication is carried out by the CSP.

According to the popularity transition, IS removing from the storage of recently popular blocks. Upon the popularity threshold, the users are notified with no awareness of a value, as the popularity transition is completely balanced by the IS, to compute the recent value for t. For the sample, the value of t is static or dynamic. Then, the recently developed model is entirely autonomous of strategy applied to equate the value of a popularity threshold.

2.2. BSE technique

In the BSE technique, the text is considered a block. The précised function should be comprised as follows. For every $\theta \in \Lambda$, let $H_\theta(X^n)$ and $H_\theta(\mathcal{X})$ be nth-order entropy and entropy rate, correspondingly, of $P_\theta$which is represented as,

$$H_\theta(X^n) = \sum_{u^n \in \mathcal{X}^n} [-P_\theta(u^n) \log P_\theta(u^n) \tag{1}$$

and

$$H_\theta(\mathcal{X}) = \lim_{n\to\infty} \frac{1}{n} H_\theta(X^n) \tag{2}$$

Let $\ell_n(u^n)$ be the definite length employed in the lossless description of u^n with selected coding principles [15]. Hence, $\delta_n(\theta)$ implies the variations among the target rate per symbol $E_\theta \ell_n(X^n)/n$ with the application of block length-n code as well as the best rate for each symbol $H_\theta(X^n)/n$ for coding $n - vectors$from P_θ; therefore,

$$\delta_n(\theta) = \frac{1}{n} E_\theta l_n(X^n) - \frac{1}{n} H_\theta(X^n) \tag{3}$$

The series of coding strategies are determined as redundancy functions $\{\delta_n(\cdot)\}_{n=1}^{\infty}$. Such deterministic bounds simplify the code functions on X^n for "empirical entropy" of X^n relevant to the distribution model approximates the basic source statistics.

A study of unifiliar, ergodic and finite-state-machine (FSM) sources are helpful in the following. FSM is a definite alphabet $\mathcal{X}$, next-state function $f: \mathcal{S} \times \mathcal{X} \to \mathcal{S}$, a finite set of states $\mathcal{S}$, along with $|\mathcal{S}|$ conditional probability measure $\{p(\cdot|s)\}_{scs}$ as represented by,

$$Pr(u^n|s_0) = \prod_{i=1}^{n} p\,(u_i|s_{i-1}) \tag{4}$$

where $s_i = f(s_{i-1}, u_i)$ for $1 \leq i \leq n$. The FSMX source is termed an FSM source where the integer M exists in which $i \geq M$, the M symbol $u_{i-\Lambda J+\perp}^{i}$ evaluate the state s_i at time i. For the FSMX source, the collection $\mathcal{S}$ is said to be a lower suffix set of strings from $\mathcal{X}^*$ using a feature where $s \in S$ and every $u \in \mathcal{X}$ so that $p(u|s) \neq 0$, the string su has a single suffix in $\mathcal{S}$. For the FSMX source, $f(s_{i-1}, \mathrm{u}_i) = suf(s_{i-1}u_i)\}$ for each i, where suf(su) is the suffix of a string attained through u symbol throughout the string s.

FSMX source is acquired from FSM source with the condition of the current state and existing state ($s_i = f(s_{i-1}, u_i)$. Finally, the limitation has been minimized by providing generalized FSMX sources, named finite-memory sources, has been expressed as,

$$Pr\left(u^n \middle| u_{-(M-1)}^{0}\right) = \prod_{i=1}^{n} p\left(u_i | s_{i-1}\right) \tag{5}$$

and $s_{i-1} = suf(u_{i-M}, u_{i-(M-1)\}} \dots, u_{i-1})$, for all i. For stationary, the $X_{-M+1}, X_{-M+2}, \dots, X_0$ symbols have to be obtained from a stationary distribution on $\mathcal{X}^M$ promoted by the finite-memory source model, represented as

$$Pr(u^n) = p(u^M) \prod_{i=M+\perp}^{n} p\left(u_i | s_{i-\perp}\right) \tag{6}$$

where $p(u^M)$ signifies the stationary distribution on $\mathcal{X}^M$ provided by FSM.

The FSM defines that it does not require the context with length k. The rapid development in $|\mathcal{S}|$ leads to performance alleviation, as the convergence values results, as defined in Section V grow with $|S|$. Here, $\theta = (p(1|s): s\mathcal{C}\mathcal{S})$ defines the distribution P_θ, and hence, $K = |\mathcal{S}|$ and $\Lambda \underline{\mathcal{C}} \mathbb{R}^K$, with well-known $(K/2) \log n/n + O(1/n)$. Commonly, $K = |S|(|\mathcal{X}| - 1)$ shows the parameter count required to define the conditional probabilities $p(u|s)$ for all $u \in \mathcal{X}$ and all values of $s \in \mathcal{S}$.

$$BWT_n: \mathcal{X}^n \to \mathcal{X}^n \times \{1, \dots, n\} \tag{7}$$

Be the n-dimensional BWT function as follows:

$$BWT_n^{(-1)}: \mathcal{X}^n \times \{1, \dots, n\} \to \mathcal{X}^n \tag{8}$$

Which is the inverse of BWT_n. As the sequence length n is acquired from the source argument as

$$(v^n, u) = BWT(u^n) and\ BWT^{(-1)}(v^n, u) = u^n. \tag{9}$$

The functions BWT_u and $BWT_{\mathbb{N}}$ indicates the character and integer portions of BWT. It is done to operate the forward BWT. The findings of the BWT are divided into two parts.

3. Performance Validation

Table 1 provides the details of the dataset. File 1 is a text file which extracts a few files from the Encron emails dataset and packs them into 30MB. Then, file 2 is a document which comprises lab documentation of 16MB. At last, file 3 is also a document with a data size of 22MB.

Table 1. Dataset Description

Files	File Type	File size
File 1	.txt	30 MB
File 2	.doc	16 MB
File 3	.txt	22 MB

Table 2 provides a detailed comparison of the results of the PH-BSE algorithm under popular file, unpopular file and popularity transition in terms of communication overhead. On assessing the results under popular file, the communication overhead for the download is 0.56MB. Besides, the communication overhead incurred by the PH-BSE algorithm for check request and check response is 0.005MB and 0.03MB, respectively. Moreover, the PH-BSE technique offers minimal communication overhead for update requests; update response and upload request are 0.20MB, 0.06MB and 14.56MB, respectively. In the same way, on determining the results under an unpopular file, the communication overhead for the download is 0.54MB. Besides, the communication overhead incurred by the PH-BSE algorithm for check request and check response is 0.004MB and 0.03MB, respectively. Moreover, the PH-BSE technique offers minimal communication overhead for update requests, update responses, and upload requests are 0.20MB, 0.04MB and 15.24MB, respectively.

Finally, determining the results under the popularity transition file, the communication overhead for the download is 0.52MB. Besides, the communication overhead incurred by the PH-BSE algorithm for check request and check response is 0.005MB and 0.03MB, respectively. Moreover, the PH-BSE technique offers minimal communication overhead for update requests, update responses, and upload requests are 0.20MB, 0.06MB and 15.14MB, respectively.

Table 2. Result Analysis of Communication Overhead of Proposed PH-BSE

Comm. Overhead	Popular File	Unpopular File	Popularity Transition
Download_in	0.56	0.54	0.52
Check_Request	0.005	0.004	0.005
Check_Response	0.03	0.03	0.03
Update_Request	0.20	0.20	0.20
Update_Response	0.06	0.04	0.06
Upload_Request	14.56	15.24	15.14

Table 3. Compressed File Size of Proposed PH-BSE

Files	File size (MB)	Compressed size (MB)
File 1	30	19
File 2	16	11
File 3	22	14

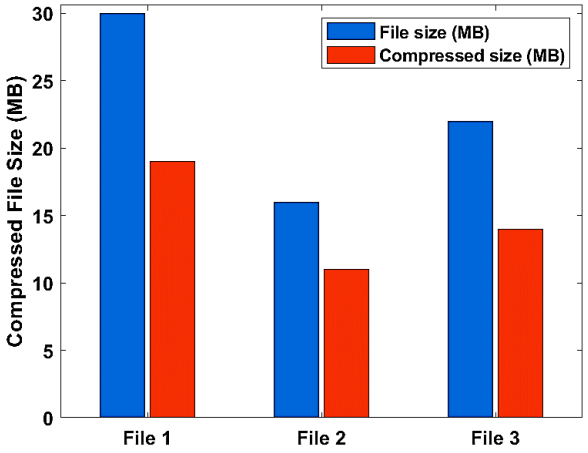

Fig. 2 Compression Efficiency Analysis

Table 3 and Fig. 2 analyse the proposed PH-BSE technique in terms of file size and compressed size. In the applied file 1, the proposed PH-BSE technique has compressed the file size of 30MB into 19MB. Similarly, in the applied file 2, the presented PH-BSE algorithm has compressed the file size of 16MB into 11MB. Likewise, the PH-BSE technique has resulted in a better-compressed size of 14MB out of the original 22MB. These values indicate that the PH-BSE technique has improved compression effectiveness on the applied dataset.

Table 4 and Fig. 3 show the analysis of the results offered by the PH-BSE technique with compared methods in terms of Access Time. The table values indicate that the presented PH-BSE technique has attained minimum access time over the compared methods. Under the existence of 1 user, it is shown that the PH-BSE technique has offered minimal access time over the compared methods. It is noticed that the EEDSP model has achieved insignificant outcomes with a maximum access time of 178s. At the same time, the SADS technique has attained slightly better access time over the EEDSP model, with an access time of 130s. Simultaneously, the presented PH-BSE technique has resulted in an effective outcome with a minimal access time of 123s.

Table 4 Number of Users Permission vs Access Time (sec)

Number of Users	PH-BSE	EEDSP	SADS
User 1	123	178	130
User 2	97	143	100
User 3	86	116	94
User 4	43	86	59
User 5	19	34	26

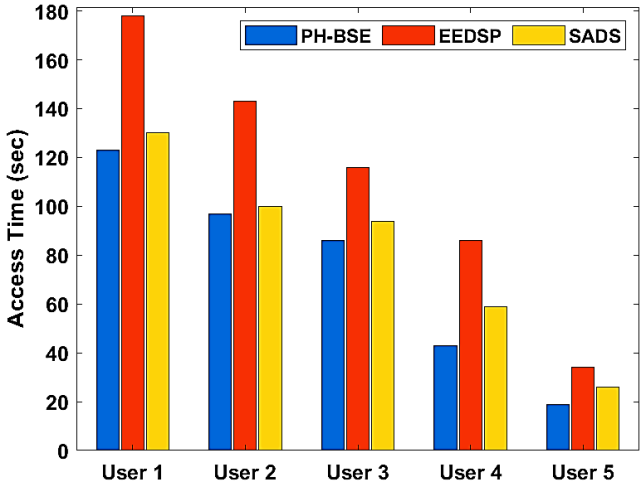

Fig. 3 Access time analysis under different numbers of users

Under the application of 2 users, it is implied that the PH-BSE model has provided minimum access time than other approaches. It is pointed out that the EEDSP technique has accomplished ineffective results with a higher access time of 143s. Meanwhile, the SADS scheme has reached a moderate access time compared with the EEDSP model, with an access time of 100s. At the same time, the proposed PH-BSE method has concluded with an optimal result with a lower access time of 97s. Under the presence of 3 users, it is exhibited that the PH-BSE model has provided minimum access time than the alternate models. It is clear that the EEDSP approach has reached a non-optimal result with a higher access time of 116s. Concurrently, the SADS scheme has achieved gradual access time over the EEDSP model with an access time of 94s. In line with this, the projected PH-BSE approach has implied efficient results with a lower access time of 86s.

With the application of 4 users, it is exhibited that the PH-BSE technique has resulted in the least access time compared with previous models. It is evident that the EEDSP model has attained an irrelevant outcome with a higher access time of 86s. Concurrently, the SADS method has reached a gradual access time than the EEDSP model with an access time of 59s. Meanwhile, the projected PH-BSE approach has shown productive results with a minimal access time of 43s. Under the application of 5 users, it is defined that the PH-BSE method has provided minimal access time over alternate methods. It is pointed out that the EEDSP model has been accomplished with the non-important outcome with the higher access time of 34s. The SADS technique has reached a better access time over the EEDSP model with an access time of 26s. Concurrently, the newly deployed PH-BSE method has offered an efficient outcome with a minimal access time of 19s. The enhanced performance of the proposed model is due to the encryption-based secure data deduplication using PH and BSE techniques.

4. Conclusion

This paper has developed a new scheme of compressive encryption based on secure data deduplication using PH and BSE techniques called PH-BSE. The presented PH-BSE technique assumes the popularity of the data blocks and controls the characteristics of PH for ensuring block-level deduplication and data confidentiality. Followed by deduplication, a compression technique named BSE technique is applied to compress the data segments. The proposed model effectively deduplicates and compresses the data. A detailed experimental analysis takes place, and the experimental outcome of the proposed model ensures that the PH-BSE technique is better than compared methods in several aspects.

References

[1] Shuangquan Li, Tongbin Zhang, Chuandi Pan, Li Cai, " Health Checkup Could Reveal Chronic Disorders With Support From Artificial Intelligence," *International Journal of Engineering Trends and Technology*, vol. 67, no. 11, pp. 8-15.

[2] Dilmurod Nabiev, Khayit Turaev, "Study of Synthesis and Pigment Characteristics of the Composition of Copper Phthalocyanine With Terephthalic Acid," *International Journal of Engineering Trends and Technology*, vol. 70, no. 8, pp. 1-9, 2022.

[3] Mandagere, N., Zhou, P., Smith, M.A., Uttamchandani, S, "Demystifying Data Deduplication," *In: Middleware '08, New York, NY, USA, ACM*, pp. 12–17, 2008.

[4] Aronovich, L., Asher, R., Bachmat, E., Bitner, H., Hirsch, M., Klein, S.T, "The Design of a Similarity Based Deduplication System," *In: SYSTOR '09*. vol. 6, pp. 1–6:14, 2006.

[5] Shwetambari Borade, Dhananjay Kalbande, Kristen Pereira, Rushil Patel, Sudhanshu Kulkarni, "Deep Scattering Convolutional Network for Cosmetic Skin Classification," *International Journal of Engineering Trends and Technology*, vol. 70, no. 7, pp. 10-23, 2022.

[6] Harnik, D., Margalit, O., Naor, D., Sotnikov, D., Vernik, G, "Estimation of Deduplication Ratios in Large Data Sets," *In: IEEE MSST* vol. 12, pp. 1 –11, 2012.

[7] Harnik, D., Pinkas, B., Shulman-Peleg, A, "Side Channels in Cloud Services: Deduplication In Cloud Storage. Security Privacy," *IEEE* vol. 8, no. 6, pp. 40 –47, 2010.

[8] Halevi, S., Harnik, D., Pinkas, B., Shulman-Peleg, A, "Proofs of Ownership in Remote Storage Systems," *CCS '11: Proceedings of the 18th ACM Conference on Computer and Communications Security*, New York, NY, USA, ACM , pp. 491–500, 2011.

[9] Di Pietro, R., Sorniotti, A, "Boosting Efficiency and Security in Proof of Ownership for Deduplication," *In: ASIACCS '12*, New York, NY, USA, ACM, pp. 81–82, 2012.

[10] Douceur, J.R., Adya, A., Bolosky, W.J., Simon, D., Theimer, M, "Reclaiming Space From Duplicate Files in a Serverless Distributed File System," *In: ICDCS '02, Washington, DC, USA, IEEE Computer Society* , pp. 617–632, 2002.

[11] Storer, M.W., Greenan, K., Long, D.D., Miller, E.L, "Secure Data Deduplication," *In: Storagess '08, New York, NY, USA, ACM* , pp. 1–10, 2008.

[12] Naveentha, Sangeetha Priyalakshmi , "An Efficient Data Deduplication Methodology in a Hybrid Cloud," *International Journal of P2P Network Trends and Technology (IJPTT)*, vol. 7, no. 1, pp. 6-9, 2017. ISSN:2249-2615, Www.Ijpttjournal.Org, Published By Seventh Sense Research Group.

[13] Xu, J., Chang, E.C., Zhou, J, "Weak Leakage-Resilient Client-Side Deduplication of Encrypted Data in Cloud Storage," *ASIA CCS '13: Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, pp. 195–206, 2013.

[14] Dr.I.Lakshmi, "A Review on Security in Mobile Cloud Computing," *SSRG International Journal of Mobile Computing and Application*, vol. 6, no. 2, pp. 4-11, 2019. Crossref, https://doi.org/10.14445/23939141/IJMCA-V6I2P102.

[15] Effros, M., Visweswariah, K., Kulkarni, S.R. and Verdú, S, " Universal Lossless Source Coding with the Burrows Wheeler Transform," *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp.1061-1081, 2002.

[16] Sanjana M. Kavatagi, Dr. Rashmi Rachh, "Implementation of Searchable Encryption Using Key Aggregation for Group Data Sharing in Cloud," *SSRG International Journal of Computer Science and Engineering*, vol. 4, no. 8, pp. 11-14, 2017. Crossref, https://doi.org/10.14445/23488387/IJCSE-V4I8P103.

[17] K.Srilakshmi, N.V.Ashokkumar, C.P.V.N.J Mohan Rao "An Empirical Model of Data Deduplication Technique Over Cloud", *International Journal of Engineering Trends and Technology (IJETT)*, vol. 45, no. 9, pp. 461-466 , 2017. ISSN:2231-5381. www.ijettjournal.org. Published By Seventh Sense Research Group

[18] Bellare, M., Keelveedhi, S., Ristenpart, T, "Message-Locked Encryption and Secure Deduplication," *In: Advances In Cryptology–EUROCRYPT* 2013. Springer 296–312 , 2013.

[19] Miss. Jayashri Patil, Dr. Sunita Barve, Mrs. Mayura Kulkarni, "Improving Security and Storage Availability in Deduplication Storage System," *SSRG International Journal of Computer Science and Engineering*, vol. 4, no. 5, pp. 10-13, 2017. Crossref, https://doi.org/10.14445/23488387/IJCSE-V4I5P103.

[20] Bellare, M., Keelveedhi, S., Ristenpart, T, " Dupless: Server-Aided Encryption for Deduplicated Storage," *In: 22nd USENIX Conference on Security*, pp. 179–194, 2013.

[21] Puzio, P., Molva, R., Önen, M. and Loureiro, S., "Perfectdedup: Secure Data Deduplication," *In Data Privacy Management, and Security Assurance* , Springer, Cham, pp. 150-166, 2015.