International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 1s, 2025
https://theaspd.com/index.php

Study On Mitigating Duplication Risks And Enhancing Security Measures In Different Cloud Architectures

JIBIN JOY[1], DR. S. DEVARAJU[2]
[1] Research Scholar (Ph.D.), Department of Computer Science, Sri Krishna Arts and Science College, Coimbatore, Tamil Nadu, India
[2]Senior Assistant Professor, VIT Bhopal University, Bhopal, Madhya Pradesh, India
jibinjoysamuel@gmail.com[1], devamcet@gmail.com[2]

ABSTRACT:
The data has to undergo data deduplication to make it smaller and avoid duplication during its transfer. The term is also frequently applicable to the cloud computing to transfer a greater amount of data and have a diminished use of the memory. An encryption method will be used to safeguard the integrity of sensitive information during the deduplication process and in some cases, the encryption method is recreating the information. The SHA computation is widely used to retrieve the information of content. It is the padding of the content to form the security bits. In the deduplication handle, it calculates the hash that is composed of hexadecimal, string and integer data. Hashing used to justify duplicate records was termed as the Hash-based deduplication technique. The hash values of carriers of content information are significant attributes. Customers sharing data with the cloud confirm that copies of the data are hosted in the cloud as opposed to conventional ways of eliminating duplicated data. Tight constraints on virtualization involve limiting capacity of much required memory and averting memory impediments. The memory deduplication searches the similar content pages and incorporates it as a unit of information to advance performance by using a reduced memory. The MPT allows in cloud storage to eliminate redundant information and save a single copy of such information to be used by several users. To make cloud information safe information is randomized before and during deduplication.

Keywords: In-line duplication check algorithm (HIDC), Mapping Technique (MPT), Virtual Machine (VM), Content-Based Page Sharing (CBPS), Content-Based Page Sharing (CBPS).

INTRODUCTION
Cloud computing services, i.e. Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) services are capable of delivering scalable storage and processing capabilities without involving installation of new systems infrastructure and fresh software installations. These services facilitate the dynamic growth of IT capacity through which the organizations can improve their performance and functionality at minimal cost of the resources.
Deduplication of data has in recent times gained significance due to the fast increasing volumes of information in the world. The approach reduces the storage cost as a copy is made of the information that slightly differ. Prior to outsourcing, files are encrypted so that they can be privatized. However, Pattern of the conventional approaches to encryption may yield distinct ciphertexts using the same elements of plaintext and thus makes it more difficult to deduplicate the ciphertexts and process them longer.
Hypercube structures are introduced at the data target which is made in the provisions of resources optimality. They guarantee equal escalation of these structures with regard to inclusion or removal of assets to scale with the fluctuation of the compatible assets provided across a sample of virtual machine (VMs). Processing nodes are autonomous and perform the processing tasks independently based on the distributed loading algorithms and have a decentralized control.
Even the efficiencies will see cloud data centers over-provision servers to handle peak demand resulting in large quantities of wasteful energy. It is a challenge that should be met with intelligent management of the resources as well as flexibility of scaling up and down so as to facilitate a trade-off type of performance againstustainability.

LITERATURE REVIEW
Subject to research study [1], one of the main challenges in virtual environment is the limited size of key memory. The CBS is responsible for identifying and sharing the duplicate pages, while the KSM memory leaves are stored in several association trees, one stable and one dangerous. The CBPS is an excellent method for eliminating duplicate information, which requires servers.Text: To determine whether the

pages can be shared, it is necessary to engage to compare the pages with the pages of two huge international chip sets. But because the specific information comes on a large number of pages, it will generate an enormous overhead with unnecessary page inspections. In order to identify opportunities for effective identification of page sharing, researchers presented a lightweight page -classed CMD -to - exclusion from unnecessary page inspection overhead.Basic idea of CMD is that websites are classified according to how much they have come. Since it is believed that the pages have the same access characteristics, they are more likely to contain the same information, so they are glued. With local in each page order, huge global relevance is divided into an enormous amount of small trees in CMD. CMD classifies its pages in different settings depending on their specific page features. Many different types of trees are made from large correlation forests around the world. Each page group has a specific tree. The pages of other orders are never compared and examined, as there may be a good chance that the relationships are meaningless. The examination of the pages is done in a similar way. Authors used basic equipment to use a memory monitoring technique to capture fine -grained page access attributes. Based on research paper [2], Virtual Machine Monitor (VMM) is a well -known platform for cloud and Internet hosting services to control processes. Virtual machine management (VMM) reduces the cost and administrative load of hosting facilities by allowing resource sharing between different systems using virtual machines. So here is the thing, when you get the layout right and some decent movement tricks down, factual multiplexing really does allow you to milk the processors around you dry. Things like that sound good, right? With the exception of--and here is a big fly in the ointment--it does not get along at all well with main memory. It is the wall that you know you run into when you become tempted to merge things on a higher level. It does have a little bit of an impact on the memory performance of running a bunch of virtual machines on the same operating system and applications, and this is in large part due to a process known as content-based page sharing. And, miracle of all miracles, it turns out to be effective.
Authors look at our experience dealing with a variety of specialised issues, such as

i. calculations to quickly identify incoming pages for fixing.
ii. request paging to help with over-membership of absolute designated physical memory, and
iii. a clock system to recognise appropriate objective machine pages for sharing, fixing, pressure, and paging.

Recent advancements in memory system design have enabled efficient data transmission across expansive spatial domains, achieving high throughput with relatively low power and cost overheads. Yet, complications arise when multiple autonomous threads simultaneously access memory, leading to significant overlap and contention within memory banks. This phenomenon effectively reduces the exploitable spatial region for each thread, and due to inherent buffering constraints, traditional memory access scheduling can only reclaim a fraction of the original region. To mitigate these challenges, the authors introduce an operating system-managed coloring strategy. This approach assigns distinct memory banks to threads that might otherwise compete for the same resources, thereby reducing contention and improving parallelism. Recognizing that individual threads may still face limited bank parallelism, the framework incorporates DRAM sub-positioning. This technique increases the number of accessible banks per thread without necessitating additional hardware investment.
By integrating bank partitioning with sub-positioning, the proposed methodology substantially enhances throughput and execution efficiency while maintaining cost-effectiveness. Empirical results demonstrate a marked expansion of the usable spatial region, translating to improved overall system performance. Notably, the authors highlight the significance of this solution in scenarios where augmenting the number of banks per thread is prohibitively expensive compared to scaling thread concurrency. Bank partitioning proves particularly advantageous for applications with large and dispersed memory access patterns, suggesting strong potential for shaping future memory architecture designs.

Implementation

This section reviews several research studies on data deduplication that directly inform the objectives of this work. The proposed approaches for protected data deduplication are examined in detail, alongside relevant concepts and algorithmic types related to these strategies.

SDD Framework (Secure Data Deduplication Framework for Cloud Environments)

Prior to data deduplication, file chunking or the process of breaking it into small chunks is needed. Rooting these segments, (also known as chunks), creates the basis of further deduplication phases. There are diverse chunking methods highlighted in the literature [6]. In our SDD model, we suggest usage of Two Thresholds Two Divisors (TTTD) method. This method is particularly good at encompassing small

changes into a file since it makes local the effect of changes made to the neighboring chunks. To illustrate the effectiveness of TTTD usage, which is employed by Kave and Tang, it is suffice to mention that, when the content is changed, only immediate locations will be controlled. It can be noted that TTTD method has been mostly used on data backup servers and, to the best of our knowledge, not yet been extrapolated to cloud-based environments. Further, the BSW process is part of Content-Defined Chunking(CDC) algorithm, where it slides through a document and determines chunking boundaries when a threshold is reached. A specific variant of TTTD can be viewed as CDC algorithm. These techniques optimize the amount of deduplication processes, allowing algorithms to pay attention to unaltered data groups.

Chunk algorithms sometimes simply fail to get it right-they may run up unmanageably large chunks, or irritatingly tiny ones-which is, of course, not what one wants. This is handled by TTTD method which eliminates small segments, and subdivides large ones. It has been found that TTTD works relatively well in controlled simulation environments, in addition to working well with large real world data [6]. The user keeps each fragment separately encrypted inside the SDD framework to have privacy of information. In the suggested method of assignment, convergent encryption [9] is more desirable. Contrary to traditional encryption based on a cryptographic key selected by chance, convergent encryption exploits the content to create cypher text that is deterministic–in other words, encrypting the same data twice will never yield a different result. This characteristic renders it particularly apt in deduplication, in which case it is of utmost importance to be aware of duplicate files without infringing on the anonymity of the user. There are other forms of encryption that include secret codes and individual key selection. These may come up with alternative forms of the same encrypted messages and this makes it hard to determine the multiples between users. Consequently, the techniques lack good compositions in deduplication.

The encryption protocol established in [9] was used in disk based deduplication but not in the cloud area where rapid access and retrieval is key. Experimental work in [10] theorizes that encryption and deduplication combined could theoretically result in storage services that never exhaust their storage capacity-however, this is under idealized scenarios, including not having any problems accessing a limitless supply of data in a cloud provider. It can not be justified with any concrete evidence at the moment.Alternative ways of organizing and finding information use machine and learning-based techniques. A method using Support Vector Machine (SVM) was suggested in research paper [11] to create deduplication rules for each individual. However, this suggested method is not suitable for a regular cloud storage setup because it heavily relies on specific criteria and gathering of data. Authors [12] suggested using a Bloom filter to check if a data is new to the system. Authors [13] also assume that the streamed data is localized.

One of the essential components in our suggested approach is to encrypt the data index of the duplicate data through the application of an asymmetrical searchable encryption technique. The concept of using special encryption techniques to protect a cloud user's data from untrustworthy cloud providers was first proposed in [14]. Many people use cloud storage to write and find data. Also, there could be one person who makes the data but many people who read it. With these limitations, our suggested approach allows the user to store the data on the internet and the service provider to search the data using a special encryption technique.

Data deduplication will not necessarily operate efficiently when there is only one individual involved in writing and reading data, however symmetric searchable encryption (SSE) algorithms are definitely compatible with this configuration. In cases where multiple users require access to data and only one person is tasked with modifying it, multi-user SSE methods turn out to be very effective. This is exactly like data deduplication in cloud computing. The CSP is the only person who can read in the cloud; hence this setup may not be very suitable for data deduplication. The researchers have investigated thoroughly the methods of how they can make highly specific searches and complicated ones within the security system. They have sought various means of encrypting the searches so that the privacy of the responses is protected. Besides this, they have also gotten into the difficult issues that come up when trying to use these encryption methods in actual systems.

The last section of our architecture is about how we can verify that the data kept in the cloud is correct by applying proof of storage. The method that shows how to locate large files is given in [20]. A method of keeping information that can later be verified as true on machines that are not trusted is the idea of a research paper by Ateniese and others. Their plan is more energy-efficient than sending the whole data over the network and doing server interactions because the user is only getting the part that is needed to check the authenticity of the information. Finally, they only take a small sample of the data sets instead

of the entire one. To the best of our knowledge, these techniques have not been demonstrated in the cloud computing environment or the data deduplication context.

SEDD Scheme (Secure Enterprise Data-Deduplication in the Cloud Environment)

In the Secure Enterprise Data Deduplication (SEDD) Scheme primary stage a file chunking unit breaks a data file into numerous smaller parts with different sizes. In order to keep a list current we propose employing B trees. B+ trees, which are a form of B trees, are most often used in disk-based systems.

Various approaches for ensuring the indexing process has been well-preserved during stages of documentation are now available. Reliable B+ trees are the data structures that are identified as suitable for constructing secure indexes in databases in the research work [23]. To demonstrate the principle of operation of a certain technique, we employ a method of grouping the nodes, and this way, we hide from the intruder the manner of them being accessed. Although this approach is not based on homomorphic encryption, it is still implemented in the database. The role of homomorphic encryption in our solution is derived from the fact that it allows us to get the encryption of a message equal to the encryption of the same message, which is extremely necessary for the elimination of duplicated data. Only the access as well as the search of the CSP are covered. To conduct this research, the information is first fragmented into many small parts which constitute the tree nodes through a technique called data chunking. The information available in the cloud has been organized into B+ trees. The structured overlay, in simple words, helps to build a local and extended index using B+ trees. This not only minimizes the data dispatched to the cloud but also keeps up with the fast development of database applications. This approach, however, does not consider the limitations in security or the issues of deduplication in the database once again.

Authors[25] have announced a new concept known as searchable encryption, that is a method allowing a keyword-based search in the encrypted data without the disclosure of keywords to the service provider. A proposal for a different method termed a private keyword search was made in [26], in more colloquial terms. It is mentioned in a work by Boneh and his coauthors that are referred to as [27] that a new method of information encryption has been found. In this method, those who want to see the encrypted information can use special codes (public and private keys) to look for it. Hence the information owner doesn't need to do the search himself/herself. The reference [28] names database queries using identity-based encryption and hash chains as a technology to guarantee the integrity here examples such as audit log are mentioned.

POR-POW Scheme (Proof of Retrieval and Proof of Ownership Protocols for Data Deduplication)

There are three types of storage protocols used to protect data in the cloud storage and to ensure that the data is safe. One of these protocols is the proof of data possession protocol. The user who owns the data confirms that it is stored correctly and securely in the cloud by using the proof of data possession protocol. The data owner is utilizing a protocol named POR (proof of data retrievability) to detect if any changes in the information have occurred. If the owner finds out that the information is not the same as expected, they can solve the problem and get the correct data. A CSP uses a protocol called POW (proof of ownership) to make sure that they only give the data to those who have the right to use it. Thus, the CSP is running the proof of ownership protocol as well. The safety of cloud storage is mainly based on three concepts: PDP, POR, and POW.A POR is a quick message that a storage provider sends to a client to confirm that a specific file is undamaged and can be fully recovered by the customer. The concept was first introduced by Juels and Kaliski in an article called [29]. Using PDP addresses helps users make sure that the data they store in the cloud is trustworthy. However, it doesn't guarantee that the file can be recovered. The idea was first suggested by Ateniese in a publication in [30]. It was then further worked on in publications in [31] and [32]. The term POW means that the CSP can be confident that unauthorized people haven't accessed or tampered with the data's safety. This was mentioned in [33].

The authors of [84] are pioneers in investigating the POR mechanisms in case of a POW and POR procedures. Their study distributed a file among different servers and each server was responsible for a particular portion of the file. The servers perform the MACs of the assigned blocks that confirm if the data is correct. The work, however, lacked proper exposition of the proposed designs and did not conduct any security analysis. A quite interesting study is found in [30] that suggests the PDP scheme for proof-of-possession of data space along with the hash function-generated file's blocks and the homomorphic signatures. The paper also shows a security proof which is based on ideas of game theory. These schemes have two main flaws. The first is that, conditions are so rigorous that they cannot be realized in the real world. They do not provide the possibility of accessing corrupted information. Subsequently, this paper

presents a new version of the latter techniques. The public can check this incarnation and it is also an infinite number of interactions with the user server.
Identical approaches could be developed further to the extent of a very small overhead that is almost always constant independently of the server-side file size, as it is suggested in [32]. The authors of these works are still checking their reality in most of the applications. Besides that, they make it possible to recover less data if there is any damage to the storage. Shachams [35] came up with the idea to use homomorphic authenticator tags in the files' pieces that are similar to the present work. In the latter scheme, the more blocks you take, the less bandwidth you need since the tag values are averaged over them. At the same time, their approach is unlimited in the number of trials. Also, no one of the mentioned techniques is dealing with the problem of data deduplication.
In referencing [36], sentinels were first developed in conjunction with a POR protocol which utilized error-correcting coding techniques. Sentinels function as checks within codes to fix errors. They are not tailored to specific groups of symbols, and instead, are placed randomly throughout the stored information. A method of twofold encoding, which utilizes partial information, improves coding efficiency. It is an improvement to this idea that includes a complete approach to protect against Byzantine adversaries [29]. Sentinels are incorporated into the data blocks, making these techniques unsuitable for data deduplication. The primary cause is that the random placement of markers within the data hindered the cloud service provider (CSP) from identifying and excluding duplicate data.The author [37] proposes a different approach to the unstructured dataset that allows easier deduplication which is to divide the dataset into smaller pieces, or "chunks," and assign unique markers to each "chunk" to facilitate the removal of duplicates. Then, for verification purposes, the data, along with the sentinels, are sent to the cloud. The CSP is fully trusted from this viewpoint since they can access the data without restrictions.Additionally, a system called Random Oracle Model that relies on the computational Diffie-Hellman assumptions demonstrates the security of this method.
References [38] and [39] take into consideration data activities at the block level for the POR protocol. Sobol's random sequencing approach is used in [38], but does not work for data deduplication purposes, as the random ordering of blocks would impede the CSP from identifying redundant data. Moreover, [39] proposes a POR utilizing homomorphic tags combined with Merkle Hash Trees for dynamic data. As pointed out in [32], the tags for the chunks (of the file) are computed and sent, which diminishes their utility. Proof of ownership techniques are proposed in [33], and [40] with regard to data deduplication. These methods, based on the assumption of a trustworthy CSP, apply the Merkle Hash Tree technique for proof of ownership. This paper incorporates enhancements to the POR and POW methods to address text, images, and videos data deduplication needs, assuming the cloud service provider is, to some extent, trustworthy. As far as we know, only our POR and POW methods radically alter data compression, data deduplication, and proof of storage algorithms where the CSP is deemed semi-honest.

RESEARCH QUESTION

In order for clients or data owners to securely perform duplicate confirmation while offering multiple advantages, the intermediate private/open cloud is used. This is a viable approach and it has sparked much discussion among professionals. Each cloud user has a cloud account containing their own documents. For example, the cloud has to store all documents if multiple cloud users participate in a shared document saving paradigm. The assumption that all cloud users sharing a document are in agreement regarding a document they wish to store is fundamentally flawed and results in a considerable amount of distributed storage being wasted.

Restrictions:

- The existing way of arranging records that are stored in the owners of the information is according to individual users.
- This causes the following complications.
- More computation cost.
- Greater use of storage space.
- The servers serve as malicious service providers because it is in the Cloud. The stored information can be hacked and this must be fought off to ensure safety of the data.
- Information that has been lost in the cloud server is a complex process that must be done with much care so as to lose no data.

- The information integrity of a record, however, is ensured by the conventional encryption failure information duplication.
- Deduplication becomes complicated because there are many users and yet each may have a different cypher text with the same data.

CONCLUSION AND FUTURE WORK

The popularity of cloud storage services is attributed to the fact that it is easy to create a copy of digital data and retrieve it at any given time, and space. With the advent of cloud computing and related storage, the big amount of digital information created becomes problematic in terms of management, and both people and companies wish to see the solution in cloud technology as the means to organize its storage, analysis, and retrieval in a convenient manner. The individuals have quit concerns about their privacy in the context of cloud storage systems usage. Thanks to this, a smaller number of people feel comfortable in using cloud storage. Although it is possible to encrypt the information at the time of outsourcing, it cannot be guaranteed to remain confidential in the situation where a service provider who is both trustworthy and observant can control what could be sensitive data. At the same time, it is stored in the cloud. Data deduplication in cloud storage enables cloud service providers (CSPs) to expand the storage capacity of the data stored with them. Data deduplication is being used to eliminate duplicate and repetitive data leaving only one copy to remain. Nonetheless, user safety and privacy are other compelling issues that accompany this procedure.

It proposed a two-level data deduplication framework that may be applied by businesses with the help of using the same services of CSP in terms of data storing. The CSP can also save costs in the following way: first, by implementing cross-user level deduplication, and ultimately by making use of cross-enterprise level deduplication. We first design a method of succinctly secure data organization with B trees. Using this approach, it is viable to manage data deduplication requirements at both organizational-levels and at an individual user-level. The index is encrypted with the help of convergent encryption to increase the level of its security. We apply the use of the keyword search where the organization uses only the private keyword search method to be able to have multiple people using it in the search of encrypted data. It also enables sharing and transferring files within the firm in a secure and convenient manner without which the company could not effectively run. In other words, a novel technique is created that ensures the safety and precision of information in the cloud of small and medium organizations. Such an approach safeguards privacy and ensures that data are retrievable and can be owned, without any issues.

We proved that the suggested system is secure in specific conditions through the comprehensive analysis of its security revealing that it could not withstand attacks carried out by both internal and external users of the system. Some instances of such attacks are the attack when the files are identified, what is inside it is determined then the spell is applied. The comparison of the suggested POR and POW performs as well in that the suggested protocols allow a user to probe using extended queries in a single session to the extent that a CSP user may be deceived well in a single session, yet they are safe enough to prevent malicious user. Through our performance analysis we have found that our framework balances the computational load equally between the users and the servers and also it is a low-cost operation in view of the fact that it only uses data that was generated in the past under different operations. We consider to enhance the security of our suggested deduplication strategies and fortify them in the future by creating a special method based on compressed sensing (CS). Applying multiple general types of measurement matrices and sampling procedures to measure different types of data would have helped in mitigating the calculation expenses that were necessitated to ensure the safety of the proposed systems. In this sense, considering the three multiple forms of data, i.e. text, video and image, we are going to analyze the Non-Deterministic and Non-Adaptive Measurement matrices / Encodings and the Non-Deterministic and Adaptive Measurement matrices / Encodings with the intention to possibly perform the deduplication processOur guide is to make certain that the contents of the data placed in the cloud will be kept secret and incompetent to the external auditor. In the semi-honest CSP we would like to have a TPA check that the information is correct, and nothing more. The data in the cloud does not have to be downloaded so that the auditor will be able to verify the information using the homomorphic linear authenticator (HLA) method. Besides, we shall also add some of the basic MAC solutions to this problem to supplement the HLA solutions. We are also trying artificial intelligence (AI) methods, neural networks included, to effectively and swiftly find duplicates. This will facilitate the creation of more elaborate yet inexpensive

methodologies of detecting duplication. What we have in mind is a situation where the data will exist in the cloud and a cloud service provider will want to remove redundant data so as to save on storage space.

REFERENCES

1. CMD: Classification-based Memory Deduplication through Page Access Characteristics by L. Chen, Z. Wei, Z. Cui, M. Chen, H. Pan, Y. Bao.
2. Difference Engine: Harnessing Memory Redundancy in Virtual Machines by D. Gupta, S. Lee, M. Vrable, S. Savage, A. C. Snoeren, G. Varghese, G. M. Voelker, and A. Vahdat
3. Data Deduplication with Encrypted Big Data Management in Cloud Computing, Nahlah Aslam K.P., Dr. Swaraj K.P. (2019-IEEE Xplore ISBN: 978-1-7281-1261-9)
4. An Effective Data Storage Model for Cloud Databases using Temporal Data De-duplication, S. Muthurajkumar, M. Vijayalakshmi, A. Kannan (2016 IEEE Eighth International Conference on Advanced Computing (ICoAC), 978-1-5090-5888-4/16/$31.00@2016 IEEE)
5. Enhanced Storage Optimization System (SoS) for IaaS Cloud StorageS. Muthurajkumar, M. Augustus Devarajan A, SudalaiMuthu T (2020, Proceedings of the Fourth International Conference on Inventive Systems and Control (ICISC 2020) .IEEE Xplore Part Number: CFP20J06-ART; ISBN: 978-1-7281-2813-9)
6. C. Bo, Z. F. Li, and W. Can, \Research on chunking algorithm of data deduplication,"American Journal of Engineering and Technology Research, vol. 11, no. 9, pp. 1353 {1358, 2011.
7. E. Kave and H. K. Tang, \A framework for analyzing and improving content based chunking algorithms," tech. rep., International Enterprise Technologies Laboratory, HP Laboratories Palo Alto, Sept 2005.
8. A. Muthitacharoen, B. Chen, and D. Mazieres, \A low-bandwidth network le system," in Proceedings of the eighteenth ACM symposium on Operating systems principles, SOSP '01, (New York, NY, USA), pp. 174{187, ACM, 2001.
9. C. Wang, Z. guang Qin, J. Peng, and J. Wang, \A novel encryption scheme for data deduplication system," in Communications, Circuits and Systems (ICCCAS), 2010
10. International Conference on, pp. 265 {269, july 2010. [20] S. Parez, \Bitcasa: Innite cloud storage." http://techcrunch.com/2011/09/18/bitcasa- explains-encryption/, Sept 2011. Online.
11. J. Dinerstein, S. Dinerstein, P. Egbert, and S. Clyde, "Learning-based fusion for data deduplication," in Machine Learning and Applications, 2008. ICMLA '08. Seventh International Conference on, pp. 66 -71, dec. 2008.
12. B. Zhu, K. Li, and H. Patterson, "Avoiding the disk bottleneck in the data domain deduplication file system," in Proceedings of the 6th USENIX Conference on File and Storage Technologies, FAST'08, (Berkeley, CA, USA), pp. 18:1-18:14, USENIX Asso- ciation, 2008.
13. M. Lillibridge, K. Eshghi, D. Bhagwat, V. Deolalikar, G. Trezise, and P. Camble, "Sparse indexing: large scale, inline deduplication using sampling and locality," in Proccedings of the 7th conference on File and storage technologies, FAST '09, (Berkeley, CA, USA), pp. 111-123, USENIX Association, 2009.
14. S. Kamara and K. Lauter, "Cryptographic cloud storage," in Proceedings of the 14th international conference on Financial cryptograpy and data security, FC'10, (Berlin, Heidelberg), pp. 136-149, Springer-Verlag, 2010.
15. D. J. Park, K. Kim, and P. J. Lee, "Public key encryption with conjunctive field key- word search," in Proceedings of the 5th international conference on Information Secu- rity Applications, WISA'04, (Berlin, Heidelberg), pp. 73-86, Springer-Verlag, 2005.
16. D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in Proceedings of the 4th conference on Theory of cryptography, TCC'07, (Berlin, Hei- delberg), pp. 535-554, Springer-Verlag, 2007.
17. J. Baek, R. Safavi-Naini, and W. Susilo, "On the integration of public key data encryp- tion and public key encryption with keyword search," in Proceedings of the 9th interna- tional conference on Information Security, ISC'06, (Berlin, Heidelberg), pp. 217-232, Springer-Verlag, 2006.
18. J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," in Proceeding sof the international conference on Computational Science and Its Applications, Part I, ICCSA '08, (Berlin, Heidelberg), pp. 1249-1259, Springer- Verlag, 2008.
19. T. Fuhr and P. Paillier, "Decryptable searchable encryption," in Proceedings of the 1st international conference on Provable security, ProvSec'07, (Berlin, Heidelberg), pp. 228-236, Springer-Verlag, 2007.
20. A. Juels and J. Burton S. Kaliski, "Pors: proofs of retrievability for large files," in Proceedings of the 14th ACM conference on Computer and communications security, CCS '07, (New York, NY, USA), pp. 584-597, ACM, 2007.
21. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proceedings of the 14th ACM con- ference on Computer and communications security, CCS '07, (New York, NY, USA), pp. 598-609, ACM, 2007.
22. T. Thwel and N. Thein, "An efficient indexing mechanism for data deduplication," in Current Trends in Information Technology (CTIT), 2009 International Conference on the, pp. 1 -5, dec. 2009.
23. H. Pang, J. Zhang, and K. Mouratidis, "Enhancing access privacy of range retrievals over B+trees," Knowledge and Data Engineering, IEEE, pp. 99-99, 2012.
24. S. Wu, D. Jiang, B. Ooi, and K. Wu, "Efficient B+tree based indexing for cloud data processing," The Proceedings of the VLDB Endowment (PVLDB), vol. 3, pp. 1207- 1218, Sep 2010.
25. R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryp- tion: improved definitions and efficient constructions," in Proceedings of the 13th ACM conference on Computer and communications security, pp. 79-88, ACM, 2006.
26. Y. Yang, H. Lu, and J. Weng, "Multi-user private keyword search for cloud com- puting," in Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on, pp. 264 -271, 29 2011-dec. 1 2011.
27. D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryp- tion with keyword search," in Advances in Cryptology-Eurocrypt 2004, pp. 506-522, Springer, 2004.

28. B. Waters, D. Balfanz, G. Durfee, and D. Smetters, "Building an encrypted and search- able audit log," in Proceedings of 11th Annual Network and Distributed System Security Symposium (NDSS 2004), vol. 6, 2004.
29. K.D.Bowers, A.Juels, and A.Oprea, "Proofs of retrievability: theory and implementa- tion," in Proceedings of the 2009 ACM workshop on Cloud computing security, CCSW '09, pp. 43–54, 2009.
30. G.Ateniese, R.Burns, R.Curtmola, J.Herring, L.Kissner, Z.Peterson, and D.Song, "Provable data possession at untrusted stores," in Proceedings of the 14th ACM con- ference on Computer and communications security, CCS '07, pp. 598–609, 2007.
31. G.Ateniese, S.Kamara, and J.Katz, "Proofs of storage from homomorphic identifi- cation protocols," in Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, ASI- ACRYPT '09, pp. 319–333, 2009.
32. G.Ateniese, R.Burns, R.Curtmola, J.Herring, O.Khan, L.Kissner, P.Zachary, and D.Song, "Remote data checking using provable data possession," ACM Trans. Inf. Syst. Secur., pp. 12:1–12:34, June 2011.
33. S.Halevi, D.Harnik, B.Pinkas, and S.Alexandra, "Proofs of ownership in remote storage systems," in Proceedings of the 18th ACM conference on Computer and communica- tions security, CCS '11, pp. 491–500, 2011.
34. Balancing DRAM Locality and Parallelism in Shared Memory CMP Systems, Min Kyu Jeong, Doe Hyun Yoony, Dam Sunwooz, Michael Sullivan, Ikhwan Lee, and Mattan Erez
35. Jibin Joy, S. Devaraju (2024) Novelic Approach For Enhancing Storage Efficiency WithBlock Size Memory Deduplication. Frontiers in Health Informatics, 13 (3), 9714-9727Memory Latency Reduction via Thread Throttling by H. Cheng, C. Lin, J. Li, and C. Yang 2010, IEEE, DOI 10.1109/MICRO.2010.39)
36. Utility-Based Cache Partitioning: A Low-Overhead, High-Performance, Runtime Mechanism to Partition Shared Caches Moinuddin K. Qureshi Yale N. Patt(2006, IEEE, 10.1109/MICRO.2006.49)
37. Singleton: System-wide Page Deduplication in Virtual Environments Prateek Sharma Purushottam Kulkarni (2012, ResearchGate, 10.1145/2287076.2287081)
38. Enhancing Operating System Support for Multicore Processors by Using Hardware Performance Monitoring (2009, ResearchGate, DOI:10.1145/1531793.1531803)
39. Managing Performance Overhead of Virtual Machines in Cloud Computing: A Survey, State of the Art,and Future Directions (2014, IEEE, DOI: 10.1109/JPROC.2013.2287711)
40. Enhanced Cloud Data Security Using AES Algorithm (2014, IEEE, DOI: 10.1109/JPROC.2013.2287711)
41. Jibin Joy, Dr. S. Devaraju, (2024), Securing Cloud Memory Through Efficient Deduplication Using Ecc Algorithm, Educational Administration: Theory and Practice, 30(5), 9421-9429 DOI: 10.53555/kuey.v30i5.4583
42. Jibin Joy, S. Devaraju (2024) Ensuring Secure Cloud Data Sharing Through Blockchain- Based Auditing For Authentication And Fuzzy Identity-Based Proxy Re-Encryption For Access Control. Library Progress International, 44(1s), 134-146.