IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AVOIDANCE OF DUPLICACY AND COMPELLING CLOUD SECURITY IN DIFFERENT CLOUD SITUATIONS

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, School of Computing Science and Engineering (SCSE), VIT Bhopal University, Bhopal, Madhya Pradesh, India

ABSTRACT:

Data deduplication is necessary for making data smaller and preventing duplication when transferring it. It is often used in cloud computing to increase the amount of data that can be transferred and reduce the amount of memory used. During the deduplication handle, delicate information integrity is safeguarded by means of an encryption approach, sometimes recently being redistributed. The SHA calculation is broadly utilized to store content information. The content is padded to make the security bits. During the deduplication handle, it computes the hash, which consists of hexadecimal, string, and integer information. The term "Hash-based deduplication" technique called hashing used to identify and remove duplicate records. The hash values of content information are important characteristics. Customers who share data with the cloud verify that copies of the data are stored in the cloud, unlike traditional methods of removing duplicate data. Strong limitations on virtualization include restricting the capacity of essential memory and preventing memory hindrance. Memory deduplication finds pages with the same content and combines them into a single information record to move forward performance whereas utilizing less memory. The MPT is used in cloud storage to remove duplicate information and store only one copy for multiple users. To keep cloud information safe, data is mixed up before and during deduplication.

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

I. INTRODUCTION

SaaS, IaaS, and PaaS, which are cloud storage and preparation administrations, adaptively increase the cap and enhance capabilities without requiring new frameworks or allowing new programming. The cloud adaptably offers storage and preparation administrations like SaaS, IaaS, and PaaS that greatly increase the limit and extend capabilities without spending money on new frameworks or enabling new programming. Distributed computing expands the current capabilities of Information Technology (IT) because the cloud adaptively provides storage and preparation administrations like SaaS, IaaS, and PaaS that significantly enhance the limit and add capacities without spending resources in new frameworks. Deduplication, which lowers storage costs by enabling us to maintain only one identical duplicate of data with minute changes, has grown in significance as the world's data reservoir has increased significantly. Before being reappropriated, files are typically protected to protect their privacy. Traditional security will always lead to distinct cypher messages being produced from the same plaintext by various families. The riddle will take time, and it will hinder data deduplication. On the data target, working assets can be successfully filtered out and put into a hypercube structure. The hypercube scales uniformly as assets are added or removed in response to changes in the quantity of a given VM sample. Each process hub is self-contained and handles its remaining tasks using various distributed load adjustment criteria and algorithms, with no oversight from focus segments. In a cloud data centre, servers are always over-provisioned to satisfy the highest demand for requests, which wastes a lot of energy.

II. REVIEW

According to a Research study [1], one of the main challenges in virtualization settings is limited principle memory size. CBPS is responsible for detecting and sharing duplicate pages, whereas KSM stores memory pages in to multiple correlation trees, one stable and one risky. CBPS is a good way to reduce the amount of memory needed by servers by removing duplicate information.

In order to determine if they can be shared, it is necessary to compare the engaging pages with pages from two large international trees. However, because unique information appears on a huge number of pages, this will result in a considerable amount of overhead from pointless page correlations. Instead of effectively identifying open doors for page sharing, the authors of this study offer a lightweight page Classification based Memory Deduplication method known as CMD to avoid unnecessary page correlation overhead.

The fundamental concept behind CMD is that sites are categorised according to how accessible they are. Because it is assumed that pages with comparable access features are more likely to contain the same information, they are grouped together. With dedicated local ones in each page order, the enormous global correlation trees in CMD are separated into an immense number of small trees. CMD categorizes its pages into various setups depending on their unique page characteristics. Many different types of trees are made from the big correlation trees around the world. Each page grouping has its own specific trees. Pages from other orders are never compared and examined since there is a good chance that the relationships are meaningless. Page examinations are conducted in a similar order. The authors' usage of basic tools allowed them to employ a memory follow-checking methodology to capture fine-grained page get-to attributes. Based on a research paper [2], virtual machine monitors (VMMs) are a well-known platform used for managing processes in the cloud and providing internet hosting services. Virtual machine managers (VMMs) decrease the expenses and administrative workload of hosting centers by allowing equipment resources to be shared among virtual machines operating on different systems. By using the right layout and movement techniques, factual multiplexing can be used to make the best use of easily accessible processors. However, the fundamental impediment to obtaining higher levels of combination is that this multiplexing is incompatible with principle memory. Previous studies have found that the memory experience of virtual machines running the same OS and applications is slightly decreased due to content-based page sharing. According to research paper [40], adding in-centre memory pressure and sub-page level sharing (via page fixing) may greatly improve performance. One of the main challenges to more advanced implementations of virtual device multiplexing is crucial memory space. The primary objective is that gathering identical Web Pages over multiple virtual devices can provide an adequate storage reserve when conducting identical exceptional actions. Finding and resolving comparable pages and in-memory page pressure releases significantly more memory reserves, necessitating this work. The authors created a Difference Engine to show how memory savings can be achieved by using techniques like page fixing, page sharing, and compression. They also evaluated the effectiveness of these methods. 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.

As per the findings of a research paper, the existing memory system efficiently utilizes the spatial domain to swiftly transmit abundant data, while simultaneously maintaining low power consumption and cost of memory devices. However, because of access streams from autonomous strings are mixed up, the pattern of growing the number of memory-providing centres causes a decline in the perceived size of spatial regions. Memory access planning can recover a tiny portion of the initial region due to buffering constraints. The writers use a method called OS-controlled coloring to allocate banks for storing data. They assign multiple separate banks to each string that might collide with others. In order to make up for the limited bank parallelism in each string, the authors of this research utilize DRAM sub-positioning. The framework allows for an increase in the number of banks without raising its expenses. We can increase production and execution while putting a priority on reasonableness with the help of this integrated bank division and sub-positioning strategy. Our investigation demonstrates that this tactic can increase the territory, which fosters advancements in proficiency and execution. The authors further demonstrate how, when considering framework setups that typically include a small number of banks for each string, our technique becomes more and more applicable. Future frameworks should include equivalent bank-obligated designs, claim the authors, given how expensive it can be to expand memory banks compared to raising the number of simultaneous strings in the CPU. Bank parceling performs effectively when each application in a multi-modified remaining burden has a broad geometric region.

IMPLEMENTATION

This section provides an overview of some data deduplication research studies that are pertinent to the goals of this work. The protected data deduplication strategies proposed in this work are discussed along with concepts and types of algorithms that are pertinent to them.

SDD Framework (Secure Data Deduplication Framework for Cloud Environments)

Any data deduplication technique must first partition the data (in this case, a file) into more manageable pieces. Chunking is the name of this procedure, and the generated units are known as chunks. In the literature, a number of chunking methods have been presented [6]. We suggest using the TTTD (Two Thresholds Two Divisors) method to divide files in our SDD framework. This method is good at managing small changes in a file. According to Kave and Tang in their research, using this strategy would only have an impact on nearby parts if the file's content was modified. To our knowledge, the TTTD approach (two thresholds two divisors) was initially implemented in data backup servers and has yet to be implemented in cloud environments. [8] The BSW procedure is a way to slide through a document used by the CDC algorithm. The CDC algorithm is a specialized version of the TTTD algorithm. A chunk boundary is created when the distance to a certain point in the data, like the previous boundary or the start of the file, meets a set limit called threshold t. Allowing algorithms to search for parts of the data that haven't changed would make the deduplication process faster. However, using these methods might give us numbers that are either too big or too small. This issue is solved in the TTTD algorithm by removing parts that are too small and dividing large parts into smaller pieces. TTTD has been shown to be one of the most effective ways to divide data into smaller parts. It works well with both computer models and real large data files. [6].

To keep the featured data private, the user makes sure each part of the file is safe in our SDD framework. To complete this assignment, we recommend using the convergent encryption method [9]. Instead of employing a user-selected random cryptographic key, this convergent encryption technique employs the actual content of the data to produce the same cypher text at every iteration. Due to this characteristic, this particular encryption technology is an ideal choice for data deduplication strategies that must adhere to user privacy standards. It is important to mention that there are other ways to encrypt information, such as using special codes or keys that only certain people have. This is done so that a message can be turned into a secret code, but different people may create different codes for the same message. However, these methods are not effective in verifying the presence of identical data among various users.

The encryption method created in study [9] was designed for data deduplication systems that utilize disk storage, not for cloud storage scenarios that require fast data access and retrieval. According to research [10], if both the encryption algorithm and data deduplication scheme are merged, a storage service that never exhausts space could be established. However, this is dependent on cloud storage providers being dependable and granting users unrestricted data access. But there was no evidence to prove this statement.

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 may not always work with only one person writing and reading data, but symmetric searchable encryption algorithms do support this configuration. When multiple users need access to data but only one person is responsible for modifying it, multi-user symmetric searchable encryption approaches prove to be highly efficient. This is similar to data deduplication in cloud computing. The CSP is the only person who can read in the cloud; so this setup may not work well for data deduplication. Many researchers have studied how to search for specific information and perform complex searches in a security system. They have looked at different ways to encrypt the searches and protect the privacy of the responses. They have also explored the challenges of using these encryption methods in real-world systems. The final part of our architecture looks at how we can make sure that the information stored in the cloud is true by using proof of storage. The proof that shows how to find big files is explained in [20]. A way to store data that can be checked for accuracy on servers that cannot be trusted, as explained in a study by Ateniese and others. Their plan decreases the amount of data sent over the network and interactions with the server because the user doesn't have to get the information to check if it's authentic. In the end, they only take a small portion of the data sets instead of the whole set. As far as we know, these methods haven't been used to show the cloud computing environment or the context of data deduplication.

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

During the primary phase of the Secure Enterprise Data Deduplication (SEDD) Scheme, a file chunking element separates a data file into multiple smaller segments having varied sizes. [22] To make sure a list is always up to date; we suggest using B trees. A type of B trees called B+ trees is commonly used in disk-based systems.

Multiple techniques for preserving the indexing procedure have been documented in written materials. B+ trees have been identified as reliable data structures suitable for creating secure indexes in databases, as stated in [23]. To show how something works, we use a method to group the nodes and hide the way they are accessed from anyone trying to hack into them. However, this method does not use homomorphic encryption in the database. The significance of homomorphic encryption in our method lies in its ability to produce identical ciphertext for identical plaintext, imperative for data deduplication. Only the ways to access and search the CSP are protected. In the study, the data is split into smaller pieces to form the nodes of the trees using a method called data chunking. The data stored in the cloud is organized using a special type of structure called B+ trees. To simplify, the structured overlay helps create a local and wider index using B+ trees. This reduces the data sent to the cloud and speeds up the development of database applications. Once again, this strategy doesn't consider the security issues or the use of data deduplication in the database.

Authors[25] proposes a method called searchable encryption, which allows for keyword-based searching of encrypted data without revealing the keywords to the service provider. A suggestion was made in [26] for an alternative method called a private keyword search, in simpler terms. In a study conducted by Boneh and his colleagues, referred to as [27], The text says that a new way to encrypt information has been created. In this new method, people who want to see the encrypted information can use special codes (public and private keys) to search for it. This means that the person who owns the information doesn't have to do the searching themselves. Database queries that utilize identity-based encryption and hash chains to ensure the integrity are referred to as having an audit log in reference [28].

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

To protect data in the cloud storage, there are three types of storage protocols used to make sure the data is safe. One of these protocols is called a proof of data possession protocol, and it is used by the person who owns the data to check that it is stored correctly and securely in the cloud storage. The data owner uses a protocol called POR (proof of data retrievability) to check if their information has been changed. If any changes are found, the owner can fix them and get the correct data. The cloud service provider (CSP) uses a protocol called proof of ownership (POW) to make sure they only give data to authorized users. Lastly, there is another proof of ownership protocol run by the CSP. The safety of cloud storage depends on three main ideas: 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].

[84] is one of the earliest studies to examine POR mechanisms in connection with POW and POR procedures. In this research, a distributed setup is explained where a file was broken down into portions and assigned to various servers. The servers check the MACs of the assigned blocks to confirm if the data is accurate. The work did not provide clear explanations of the suggested designs and did not consider any security analysis.. [30] presents a rather helpful piece of work on PDP, a proof-of-data-possession system integrating blocks of files and homomorphic verifiable tags, in conjunction with a security proof based on the idea of game theory. These schemes have two drawbacks. They are, first of all, impractical for real-world applications due to their requirements. They don't allow getting information if it is corrupted. In a later work, a new version of these techniques is introduced. This version can be checked by the public and allows for an unlimited number of exchanges between the user and the server.

These identical approaches could be enhanced further, leading to a small overhead that is almost constant regardless of the size of the server-side files, as proposed in [32]. They are still being evaluated in terms of their feasibility for a majority of applications. Furthermore, these approaches limit data retrievability in the event of data corruption. Another piece of work by Shachams [35] proposed the use of homomorphic authenticator tags in blocks of files. The tag values are averaged over more blocks in this strategy, which lowers the bandwidth requirement. Their approach also provides a limitless number of trials. Above all, none of the aforementioned techniques address data deduplication.

Data sentinels were first considered in [36], which established a POR protocol using error-correcting coding methods. Code checks known as sentinels are implemented for the purpose of rectifying mistakes.They are not made specifically for certain groups of numbers or letters, and they are placed randomly in the stored data. We can reduce the amount of storage needed by using a method called twofold encoding of data. It is an improvement to this idea that includes a complete approach to protect against Byzantine adversaries [29].Sentinels are put into the original data blocks; hence these methods are inappropriate for data deduplication. The main reason is that markers were placed randomly in the original data, which made it hard for the CSP to find and reject similar data.

The author[37] suggests a new way to store data that makes it simpler to remove duplicate information. The information is separated into separate parts, or "chunks," and special tags are created for each part of the file to remove any duplicates. Next, to confirm the authenticity, information and these markers are sent to the cloud. In this method, the CSP is completely trusted because it can see the information without any encryption. Additionally, a system called Random Oracle Model that relies on the computational Diffie-Hellman assumptions demonstrates the security of this method.

When using the POR protocol, data activities at the block level are taken into account in [38] and [39]. The Sobol random sequencing approach [38] is employed, but it is not appropriate for data deduplication since the random arrangement of blocks will make it difficult for the CSP to detect duplicate data. A POR technique using homomorphic tags and Merkle Hash Trees is also put out in [39] for POR for dynamic data. As noted in [32], the tags for the chunks (of the file) are calculated and uploaded, which reduces their usefulness.

In the context of data deduplication, proof of ownership approaches has been put out in [33], [40]. These methods, presuming a reliable CSP, use the Merkle Hash Tree method to establish proof of ownership. This paper improves the POR method and the POW approach to fit the needs of data deduplication (text, images, and videos), assuming the cloud service provider is somewhat trustworthy. To the best of our knowledge, only our POR and POW approaches transform data compression, deduplication of data and proof of storage algorithms in instances where CSP is considered to be semi-honest.

RESEARCH QUESTION

To enable clients or data owners to safely execute duplicate confirmation with various benefits, the private/open cloud is utilised as an intermediary. This approach is workable and has generated a lot of debate among specialists. Every cloud client has a cloud record that is kept in their individual storage area. For instance, the cloud must save all of the papers if numerous cloud users share the same record of saving. The assumption that each cloud client's needs for a similar document they want to keep in the cloud are the same leads to a lot of distributed storage being wasted in this situation.

Restrictions:

- Currently records put away by the information proprietors are scrambled dependent on individual clients, which prompts the following issues.
- More computational overhead.
- More extra room utilization.
- In Cloud, servers go about as noxious suppliers; the substance put away in the server may get connived, which needs to counteract to guarantee security.
- Recovery of substance fallen in the cloud server is a troublesome undertaking that needs to finish with more worry to maintain a strategic distance from the substance misfortune.
- Conventional encryption furnishes information honesty however flops in keeping away from the duplication of a record.
- Similar information duplicates of different clients will wind up with various cypher texts, making deduplication troublesome.

III. CONCLUSION AND FUTURE WORK

Cloud storage services have gained significant popularity for their ability to easily store and retrieve digital data from any place and at any given time. The advent of cloud computing and its corresponding storage services has led to an overwhelming volume of digital data, presenting difficulties in its management for individuals and enterprises alike.Cloud technology can meet the needs for storing, analyzing, and accessing big amounts of data in an accessible way. People are worried about their privacy when it comes to using cloud storage systems. Because of this, not many people are comfortable with using cloud storage. Even if information is encrypted when it is outsourced, there is still no guarantee that it will remain private if a trustworthy but observant service provider has control over the sensitive data. Simultaneously, it is saved in the cloud. Cloud service providers (CSPs) can increase the amount of data they can store by using data deduplication in cloud storage. Data deduplication is a way to get rid of duplicate and repeated data by keeping only one copy. However, this process also brings up important concerns about user safety and privacy.

It presented a two-level data deduplication framework that businesses using the same CSP's data storage services can use. By first deploying cross-user level deduplication and eventually cross-enterprise level deduplication, the CSP may optimise digital space savings and save expenses. First, we create a way to organize data securely using B trees. By utilizing this method, we can effectively handle data deduplication needs for the entire organization as well as individual users. Encrypting the index using convergent encryption enhances its security measures. We use the private keyword search method to allow multiple users to search for encrypted data in an organization. It also allows for sharing files within a company in a safe and practical way, which is necessary for the company to run smoothly. In simple terms, a method has been developed to make sure that data remains safe and accurate in the cloud for small and medium organizations. This method protects privacy and guarantees that data can be retrieved and owned without any problems.

By conducting a thorough security analysis, we demonstrated that the suggested system is secure under particular conditions, impervious to attacks from individuals both inside and outside the system. Examples of such attacks include those that identify files, find out what's within, and use the spell. Our analysis also shows that the proposed POR and POW protocols enable sufficient queries in a single session to allow a user to probe from the CSP while still being secure enough to stop a malicious user. Our performance analysis has demonstrated that our framework evenly distributes the computational load across users and servers while maintaining a low operational cost by using data that has previously been generated during various operations.

In order to strengthen the security of our recommended deduplication strategies, we plan to develop a compressed sensing (CS)-based technique in the future. Utilising various measurement matrices and sampling techniques for multiple sorts of data would assist in reducing the computational cost for the security of the suggested systems. In this regard, taking into account the three multiple forms of data, namely text, video, and image, we intend to examine the Non-Deterministic and Non-Adaptive Measurement matrices / Encodings along with the Non-Deterministic and Adaptive Measurement matrices / Encodings for deduplication purposesOur priority is to ensure that the content of the data stored in the cloud remains confidential and inaccessible to the external auditor. We want to allow the TPA to only check if the data is accurate in the semi-honest CSP. By using the homomorphic linear authenticator (HLA) approach, the auditor can check the data in the cloud without having to download it all.

Additionally, to complement the HLA solutions, we will incorporate some fundamental MAC solutions for this problem. We are also seeking to use artificial intelligence (AI) approaches, which include neural networks, to effectively and quickly identify duplicates. This will enable the development of more sophisticated and affordable techniques for identifying duplication. We are preparing for a scenario where data is stored in the cloud, and a cloud service provider aims to eliminate duplicate information in order to conserve storage capacity.

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. M.Lillibridge, S.Elnikety, A.Birrell, M.Burrows, and M.Isard, "A cooperative internet backup scheme," in Proceedings of the annual conference on USENIX Annual Technical Conference, pp. 3–3, USENIX Association, 2003.
35. H.Shacham and B.Waters, "Compact proofs of retrievability," in Advances in Cryptology-ASIACRYPT 2008, pp. 90–107, Springer, 2008.
36. A.Juels, Jr.Kaliski, and S.Burton, "Pors: proofs of retrievability for large files," in Proceedings of the 14th ACM conference on Computer and communications security, CCS '07, pp. 584–597, 2007.
37. Q.Zheng and S.Xu, "Secure and efficient proof of storage with deduplication," in Pro- ceedings of the second ACM conference on Data and Application Security and Privacy, CODASPY '12, pp. 1–12, 2012.
38. S.Kumar and R.Subramanian, "An efficient and secure protocol for ensuring data stor- age security in cloud computing," International Journal of Computer Science, vol. 8, 2011.

39. Q.Wang, C.Wang, K.Ren, W.Lou, and J.Li, “Enabling public auditability and data dynamics for storage security in cloud computing,” Parallel and Distributed Systems, IEEE Transactions on, vol. 22, no. 5, pp. 847–859, 2011.
40. 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
41. Memory Latency Reduction via Thread Throttling by H. Cheng, C. Lin, J. Li, and C. Yang 2010, IEEE, DOI 10.1109/MICRO.2010.39)
42. 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)
43. Singleton: System-wide Page Deduplication in Virtual Environments Prateek Sharma Purushottam Kulkarni (2012, ResearchGate, 10.1145/2287076.2287081)
44. Enhancing Operating System Support for Multicore Processors by Using Hardware Performance Monitoring (2009, ResearchGate, DOI:10.1145/1531793.1531803)
45. 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)
46. Enhanced Cloud Data Security Using AES Algorithm (2014, IEEE, DOI: 10.1109/JPROC.2013.2287711)
47. A Framework Based on RSA and AES Encryption Algorithms for Cloud Computing Services (2014, IEEE, DOI: 10.1109/I2C2.2017.8321820)
48. Enhanced RSA Algorithm with varying Key Sizes for Data Security in Cloud (2017, IEEE, DOI: 10.1109/WCCCT.2016.50)
49. DROPS: Division and Replication of Data in Cloud for Optimal Performance and Security (2015, IEEE, DOI: 10.1109/TCC.2015.2400460)
50. Applying Encryption Algorithm for Data Security in Cloud Storage (2016, ResearchGate, Springer, DOI: 10.1007/978-981-287-990-5_12)