ENCRYPTED DUPLICATION: SAFE AND SPACE-SAVING METADATA MANAGEMENT

[#1]PINIKASI SAICHANDANA,

[#2]B.ANVESH KUMAR, ***Assistant Professor,***

[#3]Dr.V.BAPUJI, ***Associate Professor& HOD,***

Department of Master of Computer Applications,

VAAGESWARI COLLEGE OF ENGINEERING, KARIMNAGAR, TELANGANA

ABSTRACT: Cloud service providers provide users efficient and effective data storage and transfer. To reduce storage costs and conserve bandwidth, cloud service providers are attracted to use data de-duplication solutions. Cloud users want to use the cloud securely and privately in order to protect the data they keep there. As a result, they encrypt data before uploading it to the cloud. The data de-duplication feature becomes difficult to utilize since the encryption intent competes with the de-duplication function. Existing de-duplication approaches are unsecure and inefficient in terms of both security and efficiency. They are either brute force assault vulnerable, allowing the attacker to retrieve files, or computationally expensive. This is what drives us to develop a quick and secure way for deleting duplicate data. We'll start with a description of de-duplication technique implementations and functionality before going on to the literature, which discusses alternative approaches to de-duplication as well as the security and efficiency issues that existing systems face. We proposed an invention that, by leveraging the AES-CBC algorithm and hashing approaches, increases the performance and security of data de-duplication for users. Without the involvement of a third party, users' keys are created in a consistent and secure manner. We demonstrate the usefulness of the recommended strategy by putting it to use and comparing it to existing techniques.

Keywords: *Data duplication, De-duplication, Cloud computing, Security, Encryption.*

1. INTRODUCTION

There have been significant technological advancements in the current decade, resulting in a plethora of opportunities for organizations and individuals. Recent technological advancements have provided consumers with a plethora of high-quality solutions and a variety of ways they can use to better their businesses' operations, increase their profits, and boost their productivity. The user's message does not provide any relevant details. Business processes and transactions have been more streamlined as a result of technological advancements. However, businesses of all sizes are concerned about various aspects of system security, with data protection taking center stage [2]. Data in corporate information networks is vulnerable to inaccuracies caused by database duplication, which is a major concern. Users have a harder time finding the right record when data collection becomes more routine.

The server's data could include many different types of files, necessitating more space for storage and even jeopardizing data security. However, data consistency can be affected by data duplication. The existence of duplicate data in many locations has raised security and storage problems for many insurance and financial institutions. Modern society's requirements necessitate the application of extremely efficient methods for maximizing cloud storage systems' efficacy[3]. Data deduplication, or the elimination of identical duplicates of a file, is generally accepted as a powerful method of boosting storage efficiency, optimizing bandwidth consumption, and decreasing expenses [4].See, for instance, [5] and [6]. De-duplication is a method of data reduction that can be seen in action in Figure 1[7]. One can also refer to this method as "intelligent

compression" or "single-instance storage" [8]. The problem of data duplication is one that many large companies are now facing. The storage and security implications of data replication are significant for individuals, SMEs, and multinational businesses alike. The chore of keeping and finding records can be difficult for professionals. Based on the comparison results reported in reference [9], the effects of data redundancy on system time and cost are depicted in Figure 2. In this diagram, we can see how redundant information affects the time needed to run the system and the money spent on maintaining it.There was a significant reduction in backup activity after using data deduplication [9].

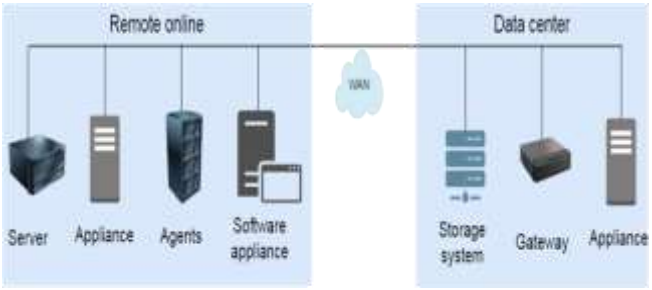


Figure 1:Where de-duplication can applied.

Data integrity, system performance, and data security are just some of the data management concerns that could be compromised by the existence of duplicate data in storage. Data duplication is mostly to blame for this because of the complications, inefficiencies, and security risks that come along with it. While it's a simple effort to delete duplicate data that hasn't been encrypted, doing so in the current case adds another layer of complication [10]. The security of sensitive data kept locally or in the cloud may be jeopardized if it is duplicated, as this practice can give rise to a number of cybersecurity problems. Encryption of cloud-stored data is crucial for ensuring its safety from a cyber security perspective (11). The existence of encrypted data in the cloud increases the difficulty and difficulties of deleting duplicate data. Dedu is a program that can efficiently delete duplicate data, but it cannot work with encrypted files. Using traditional encryption methods to protect the privacy of cloud-based data is complicated by their inconsistency with data deduplication strategies. However, de-duplication might be difficult because many cipher texts may contain duplicate data copied with different users in mind [13].

The user only offered a number as a reference, with no explanation or background.Before transferring information to a cloud-based storage system, data owners should encrypt the information using a secure method [15]. It is interesting to see how a cloud service decides [16] whether or not a given text is the product of mixing many encrypted texts. The researchers suggested using a reliable third party to help delete unnecessary encrypted data (17), (18), and 19). The inherent difficulty in setting up a completely trustworthy third party makes this approach unfavorable [20]. In spite of its increased susceptibility to brute-force attacks, convergent encryption is a pragmatic method to resolve the discrepancy between encryption and the reduction of superfluous data.

Encrypted de-duplication improves data storage security and reliability by combining encryption and de-duplication methods. The aforementioned goals, however, are inherently incompatible. To address this matter, a new encrypted data de-duplication method is now under development. This article focuses on convergent encryption, a mechanism that can securely delete encrypted data without breaking the bank. To make the integration more trustworthy, we will use AES-CBC encryption. Due to its minimal cost and resource requirements, the AES method is also regarded as a ground-breaking solution to de-duplication. The strategy not only improves performance but also accomplishes its primary purpose of de-duplication. The encryption process will use a hash derivation method and a long random string salt to increase the difficulty of the encryption key. In addition, even if the offender successfully relays the hash for the associated file, they will be unable to pass on the supplemental salt.

By using the encrypted file's signature as a label value, the server's storage space can be freed up

by removing any copies of the file. The fundamental goal of this research is to improve the lives of data users by making it possible for them to retrieve the information they need quickly and easily while also increasing the level of security that is in place. This study explores and presents a complete body of knowledge to assess the efficacy of current methods and tactics used to achieve this goal. In light of the drawbacks of existing techniques, a workable solution is offered. The suggested approach is based on a set of algorithms that permit the discarding of unnecessary data while simultaneously enhancing data security.

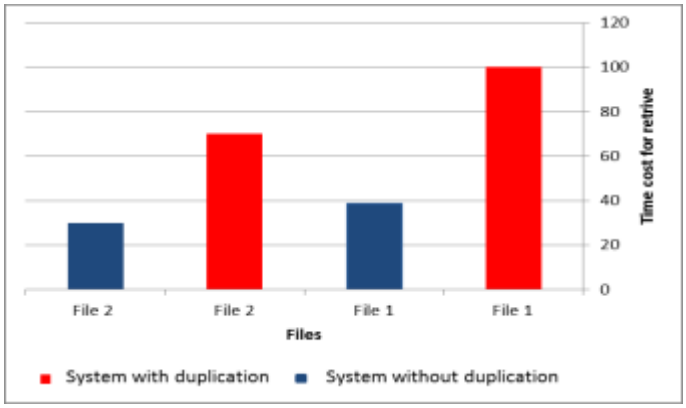


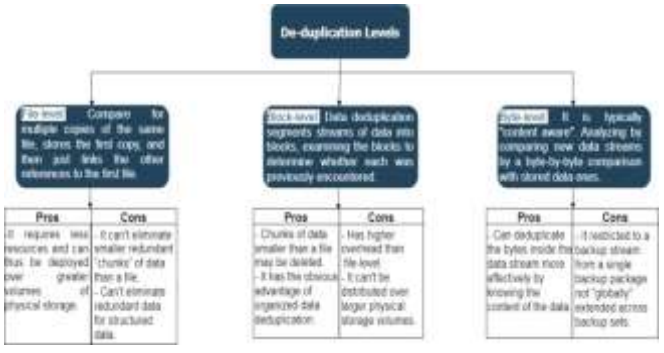
Figure 2: A Comparison results of data with de-duplication[9].

De-duplication levels

Data deduplication is a method for eliminating data repetition. If you want to limit the amount of data being transferred and stored, you should get rid of duplicates[21][22]. By examining data sequentially, file by file, block by block, and byte by byte, de-duplication is able to identify duplicates. Deletion methods are currently most effective when applied to data at the file, block, and byte levels. But there's room for improvement in terms of storage. The three categories of de-duplication levels are depicted in Figure 3[23, 24] as file, block, and byte.

Figure 3: De-duplication Levels.

File-level de-duplication



File-level de-duplication is a data reduction technique that operates at the level of individual files, as implied by its name. The storage referred to in this context is commonly referred to as single instance storage (SIS) [7]. Every individual file is allocated a distinct identification number that is derived from its specific characteristics. The identifier retains the complete dataset associated with the file, facilitating the removal of files that exhibit similarities but possess distinct names.

Block -level de-duplication

The technology of block-level data de-duplication involves the segmentation of a data stream into blocks. Each block is then analyzed to see if it contains identical data to previous blocks. This is commonly achieved by utilizing a hash algorithm to establish a digital signature or unique identifier for each data block [25]. In the event that the block possesses distinct characteristics and has been successfully stored on the disk, its identifier is also recorded within the index. In the event of its nonexistence, the sole deposit indicator that maintains the original position of the identical data block is [26] [27][22].

Byte -level de-duplication

Byte-level de-duplication can be classified as a form of file-level de-duplication due to its reliance on content information for identifying a file in relation to other data formats. Byte-level deduplication is a form of block-level deduplication.

The process of data de-duplication, which takes into account the semantics or substance of the data. The aforementioned systems are commonly referred to as content-aware systems (CAS). An alternative approach involves doing data deduplication subsequent to processing the data at the byte stream level [22].

De-duplication Applications

Data de-duplication is a data reduction technique that aims to reduce redundant or unneeded data. Instead of preserving the original dataset, it partitions it into predetermined data entities such as files or blocks, identifies redundant segments, and maintains a compact reference to the duplicated portions. Common data compression

techniques often aim to minimize the presence of repetitive information within a file, hence optimizing storage capacity. On the contrary, de-duplication is a method that use diverse data processing techniques or indexing strategies to minimize redundant data present in several files. De-duplication is a very efficient approach for optimizing storage capacity, particularly in the context of archive backup systems, as compared to conventional data compression methods.

Backup storage De-duplication

The utilization of de-duplication in cyclical complete backup processes is advantageous when backup systems consist predominantly of extensively replicated data. The process of de-duplication involves identifying duplicate data blocks by examining the original data, retrieving a single instance of a certain data block on the storage medium, and creating references to other copies using a block index. This approach enables users to enhance the storage capacity of their current storage devices by significantly reducing the space needed for the source data. In order to extend the duration for which backups are retained, a larger volume of data is stored on individual computers. The implementation of de-duplication in data centers enables enhanced cost-effectiveness and operational efficiency by reducing the number of storage devices required [28].

Cloud storage De-duplication

Cloud storage, a form of offsite data storage, enables users to access and retrieve data from several platforms at their convenience. Consumers have the potential to achieve cost savings by effectively managing the quantity of data being requested, as it may be adjusted to meet varying levels of demand. In contemporary cloud storage systems, it is common for customers to own similar data distributed across several accounts. Consequently, the process of de-duplication becomes necessary in order to achieve cost savings [28][29]. The process of de-duplication enables a cloud storage provider to effectively manage and store only a single physical copy of identical data in a specific place, resulting in reduced utilization of network bandwidth and storage capacity.

Virtual machines storage De-duplication

Data duplication is a prevalent occurrence in host file systems and primary storage systems. In this scenario, it is highly probable that comparable operating systems have been installed on digital devices. According to the literature, virtual machines have the capability to share applications, custom configurations, and storage device settings [29].

2. LITERATURE REVIEW

This section aims to offer a comprehensive overview of data de-duplication, encompassing its fundamental concept and specific methodologies. The objective is to provide an up-to-date understanding of the subject matter.

Data De-duplication

Efficient compression techniques are employed to reduce the presence of duplicate data and effectively eliminate superfluous information. Furthermore, it has been extensively employed in the realm of cloud storage with the purpose of conserving bandwidth and minimizing the amount of storage capacity required. The research findings also indicate that the consideration of system users' access levels is a factor in the process of identifying duplicate records or data [30].

The value of data is widely acknowledged by enterprises. The expectation of dependability and consistency in big data is a requisite for companies to make informed judgments. The study underscores the need of utilizing high-quality data and implementing an effective duplicate data reduction strategy in order to draw sound conclusions [31].

The proliferation of duplicate records in cloud-based storage systems has raised significant concerns over the security and integrity of digital data. Managing data duplication in cloud computing can present significant challenges. The researchers have presented a methodology for mitigating data redundancy. This study presents the design and implementation of development tools and methods that are founded on video and

picture data [32]. This paper[33] identifies two distinct categories of data duplication reduction technologies: identical data detection approaches and comparable data encoding and detection techniques. A survey was conducted by researchers with the aim of gathering data on the prevalence of data duplication issues inside organizations. The main objective of this study is to ascertain the challenges associated with duplicate records and determine appropriate strategies for their minimization.

In their study, the authors [34] proposed a methodology that effectively addresses the issue of eliminating redundant data from cloud storage systems while simultaneously enhancing security measures. The researchers have moreover developed a safe central duplicate elimination storage system. This study provides evidence in favor of reducing data duplication at both the block and file levels. Based on the findings of this study (5), the exponential growth of data, including various forms such as images, audio, and text files, presents several difficulties in terms of storage and retrieval methodologies. Business enterprises allocate significant resources towards the acquisition and maintenance of data storage infrastructure. Consequently, the management of substantial volumes of data requires the implementation of a highly effective methodology. The researchers are currently directing their attention towards strategies for data redundancy elimination, as they are faced with the challenges associated with these procedures. The issue also arises in situations when there is a constraint on bandwidth or a reduction in available bandwidth, resulting in a decrease in the utility of data. In addition to prioritizing optimization methodologies, a comprehensive survey is devised to identify and assess potential risks and vulnerabilities.

The implementation of a data eradication procedure duplicate is highly recommended as a means to mitigate hazards. The rapid depletion of storage occurs when data is replicated on cloud-based platforms. This paper outlines many tactics aimed at the development of a system that effectively reduces redundancy in cloud storage. In addition, empirical evidence indicates that the use of this particular approach has resulted in a reduction of redundant data in cloud storage by around 20 to 30% [35].

The optimization of duplicate data removal is of utmost importance due to the significant impact it has on the outcome of a procedure. The researchers put forth a biometric methodology in addition to an improved iteration of Rabin's algorithm. The comprehensive implementation of the technique will successfully address the issue of duplication. Additionally, the data file included inside this framework is saved within a cloud storage system that is divided into distinct partitions, each assigned with specific block numbers corresponding to the data records. According to the literature, the proposed technique demonstrates superior efficacy in the eradication of redundant data copies [36].

Cloud computing, as stated in the research paper by reference [37], is a very practical and efficient technological solution that offers a wide range of data storage options. The accessibility of data saved in the cloud is highly convenient, as it can be accessed from any location. In addition, the utilization of costly equipment, software, and dedicated facilities contributes to the prolonged duration required for accessing replicated data. Moreover, the necessity to enhance the organization's infrastructure arises in order to address the demanding and complex endeavor of augmenting storage needs. Duplicate records can lead to a multitude of issues for organizations. The checksum algorithm was employed to address the issue.

The dimensions of data centers are seeing a swift expansion in tandem with technological advancements, while network technologies are concurrently undergoing rapid evolution. Numerous organizations that aim to mitigate storage challenges are contemplating the adoption of environmentally sustainable storage solutions, sometimes referred to as green stores. The implementation of data duplication reduction technology has significant promise in terms of

mitigating the need for extensive data storage in optimization of storage systems. The utilization of data compression techniques can result in a reduction in the number of drives required for a given activity, hence mitigating the costs associated with disk energy consumption. The foundation of this study will be established by an examination of the data duplication elimination technique, including its associated procedures and execution [1].

The paper [38] employs the duplicate and multiple representations approach for the purpose of eliminating redundant data. The researchers subsequently reported that they had developed two novel data duplication algorithms, which would assist the system in identifying duplicate entries and enhancing system effectiveness and efficiency. The algorithms have the capability to significantly enhance the overall efficiency of the process within the given time limitations. Moreover, by the implementation of these algorithms on pre-existing data stored in a repository, the researchers found unexpected outcomes and subsequently deduced that the approach exhibits expeditiousness.

Based on research conducted by the IDC analytics group (39), it was found that around 80% of organizations employed de-duplication technologies in order to mitigate redundant data inside their storage systems. The primary objective of these studies was to develop effective methodologies for data deduplication, with a specific focus on enhancing storage efficiency and optimizing decision-making processes. The aim was to achieve superior data quality and effectiveness, while simultaneously reducing the frequency range. The significance of maintaining storage efficiency is undeniable. However, it is imperative to further explore the matter of cloud computing storage security and promptly devise a strategy that optimizes and harmonizes both storage efficiency and storage security. Consequently, it is imperative to acknowledge that data security and privacy remain significant challenges associated with data duplication technologies, necessitating more endeavors to effectively tackle these concerns.

This study presents a proposed strategy to address the conflict between the removal of duplicate data and the process of watermarking, considering the possibility that data owners may opt to transform external multimedia content into more secure formats [40]. Deleting duplicate data can be challenging due to the potential transmission of such data between different data owners and in various forms. The suggested method for watermarking is extensive and does not necessitate any form of communication between data proprietors or dependence on an external entity. The implementation of a protocol is employed to enhance the solution, so guaranteeing that comparable data for distinct users will possess a consistent secured format. The utilization of a watermark serves as a means to prevent replication.

The topic of index management for duplicate data reduction processes is discussed in the paper referenced as [41]. When such an occurrence takes place, the duplicate data elimination process is activated, with the objective of conserving memory and storage resources by reducing input/output operations. In this study, Austerecache is suggested as a flash caching technique for the purpose of effectively organizing memory indexing. Moreover, this idea facilitates the implementation of suitable data structures, the removal of a significant portion of information for indexing purposes, and the effective management of caching through important procedures. Based on empirical research, the utilization of this technique presents the benefit of preserving similar ratios of reading and write reduction, while achieving a substantial memory savings of 97%.

The identification of tuples within a relation is commonly referred to as data de-duplication. The computational complexity of the operation is inherently quadratic in relation to the number of tuples, as it necessitates the determination of a similarity value for each pair of tuples. In order to prevent the comparison of tuple pairings that are clearly not duplicates, blocking techniques are employed to partition the tuples into distinct

blocks, hence restricting the comparison to tuples within the same block. Despite the implementation of blocking techniques, the process of data de-duplication for extensive datasets continues to be a financially burdensome undertaking. This study aims to illustrate the utilization of parallelism inside a shared-nothing computing environment as a means to enhance the efficiency of data de-duplication, as outlined in reference [42]. The Dis-Dedup delivery technique is an effective approach that minimizes the workload on worker nodes while offering strong theoretical assurances.

In this study (Smith, 2019), the utilization of the Bloom filter is employed as a means to develop a technique for the removal of redundant data. The proposed strategy consists of three primary segments. The cancellation of a duplicate was a viable option. The Administration Center is responsible for managing requests that have been granted authorization. The development of a radix trie subsequently delineates the correlation between roles and switches. It is recommended to incorporate a BLOM filter for the purposes of data refreshing and efficiency testing. Based on the results of simulation tests, the model effectively calculates the ciphertexts and exhibits a notable repeated data cancellation rate of up to 25%. If one of the generated ciphertexts coincides with the ciphertext of the target message, it is possible for the adversary to infer the content of the message [64].

Single-Server Cross-User De-duplication

This system enables the implementation of client-side encryption. The PAKE protocol facilitates the retrieval of an encryption key by clients from another client who has previously transmitted the same content. In the initial delivery of files, it is recommended to employ a randomly generated encryption key for the purpose of securing the data. However, this particular approach gives rise to a side-channel attack whereby clients can potentially deduce the absence of a specific file within the cloud storage system. Prior to being transmitted to the cloud storage provider, it is possible for any session key to be modified to incorporate a deceptive value. To prevent server-side attacks that include the cloud storage provider attempting to identify users who have uploaded a certain item without compromising bandwidth efficiency, the utilization of client-side encryption and single-server cross-user de-duplication techniques is employed [65].

Symmetric or asymmetric encryption

Prior to being uploaded to the Cloud Service Provider (CSP), the data undergoes encryption using either a symmetric or asymmetric encryption scheme. The encryption keys are generated by the data owner and then encrypted using the ciphertext policy before being stored in the CSP [10][53]. The Cloud Service Provider (CSP) is unable to obtain the cryptographic keys due to their failure to meet the specified requirements. The utilization of this method effectively mitigates online brute-force attacks, as the CSP's information disclosure does not provide any indication to a hostile user regarding the presence or absence of data storage. The proposed methodology involves the construction of a bilinear mapping through the utilization of bilinear mappings and features, with the aim of optimizing computational efficiency during the encryption and decryption processes [66]. Therefore, reducing computational costs emerges as the foremost concern [67]. The resolution of this matter is accomplished through the utilization of an externally sourced decryption mechanism, an approach that is not advised in practical applications [10].

3. PROPOSED APPROACH

Specification requirements

The predicted functionality criteria for the proposed technique are as follows:

Efficiency: The proposed methodology effectively mitigates the presence of redundant data in order to optimize disk space use, save expenses, and minimize network traffic. A data user with proper authorization have the ability to effectively and efficiently manage cloud-based data in a manner that is both expedient and convenient. The cloud service provider possesses

the capability to effectively manage duplicate encrypted data.

Security: The proposed methodology demonstrates a significant degree of security for all entities. The service provider has the capability to remove duplicate data without compromising the confidentiality of the plain text. Furthermore, the entity responsible for the data use encryption techniques to securely transmit files to the cloud storage platform. One benefit for the data user is the exclusive access granted solely to authorized individuals, ensuring that only those with proper authorization can view and present the data in its unencrypted form.

Computation: In order to reduce expenses associated with maintenance and computation, it is recommended to implement a secure storage solution that is lightweight in nature. Additionally, it is necessary to acquire permission from the administration of the third party.

Auditing: In order to ensure that the user's data loss in the cloud results in a failure of the user's validation process.

Data owner: An individual use cloud-based services to securely keep encrypted files subsequent to putting data onto the cloud platform. In the event that authorization to access the cloud is granted, the responsibility for managing the decrypted file falls upon you.

Data users: The individual responsible for creating the file upload feature now seeks authorization to access files stored in a cloud-based environment. If the individual is granted permission to retrieve the file, their user key aligns with the access policy established by the data owner. The accuracy of cloud data can be verified by the implementation of a cloud storage auditing technique by users.

Cloud service provider: is responsible to handle and storing encrypted files in the cloud, also If an individual or organization has access to certain information, it is their responsibility to use that information correctly. Data authenticators are computed, and then the correctness of search results is verified to ensure that the data has not been tampered with. In addition to the requirement of de-duplicating data in files.

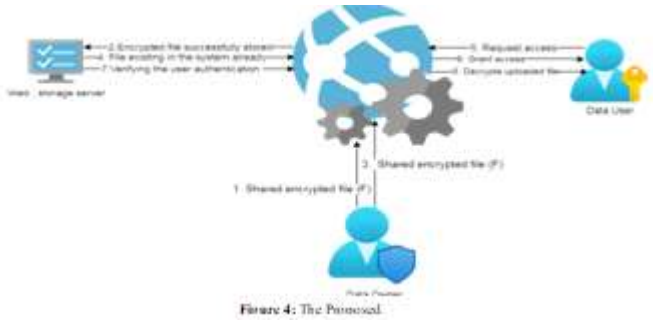


Figure 4: The Proposed

Figure 4 illustrates the presence of three entities, namely the cloud server provider, the data owner, and the cloud server user.

Security model

The following procedures are used to purge unnecessary information from cloud storage while keeping sensitive data safe.

Setup algorithms(S): How the system is set up. The safety setting is currently disabled.

Key Generation (Key Gen): Calculating an MD5 hash using the supplied data and a robust string salt yields the key. Additionally, the IV is provided for use with the AES algorithm. The end result will be a secure cryptographic key that both parties can trust.

Encrypt algorithm (E): The plaintext and the shared key are both included in the input. The final product will be ciphertext, abbreviated CT.

Integrity Check(i): The cloud service takes precautions to ensure that all uploaded data remains private and secure.

Access structure schema (AS): The encrypted text is linked to the authorization framework. The hidden quality and the characteristic go hand in hand. To put it another way, data decryption is possible if and only if the attributes contained within the attribute set match the access structure. In this context, "decrypt" means to unravel a code. The input is made up of the ciphertext (CT) and the shared key (K). The final text output format will be PT, which stands for plain text.

Dedup Check (Dedup): The encrypted data's hash value is used to generate the data label during the de-duplication process. When encrypted data is used as input, the resulting output can be either 0 (indicating storage) or 1 (representing deletion due to de-duplication).

Figure 5: Security model.

```
Setup algorithm:

Key generation:
Input: Security parameterOutput: shared key(K)
1- seed(time(0))
2- for i ∈ [1..32]
Calculate Ki=MD5(MD5(PT)+Salt)) , (32,127) //printable
char3- K <--- {K1, K2, K3, ... X32}
4- return K
IV generation: Input: None Output: key IV
1- seed(time(0))
2- for i ∈ [1..16]
Calculate Xi=random(32,127) //printable char 3- IV <--
{X1, X2, X3, ... X16}
4- return IV
```

The server generates the AES cipher parameters and stores them in a locally accessible, read-only file during application setup. Only root is immune to harm. Either way, the server's random number generator will be seeded with the current time as an integer (timestamp). Then, we generate a random string of 32 bytes for the key and 16 bytes for the IV.

Use the web interface to access the shared F file. The onus of data encryption rests squarely on the user. It employs AES CBC with a 256-bit key length. The AES cipher settings are configured and stored in a file on the file system that only root may access during the server setup process. A suid-wrapped executable will be created to gain access to the secret. It will be launched using Python's subprocess module. The server will employ the Popen technique:F + Key + IV + E = CThe computer will then determine the Sig, or signature, of the ciphertext. An example formula involving md5 and sha512 functions is Sig = S1(C) + S2(C).The created signature will be used as part of a secure SELECT query to look for a matching record in the database.

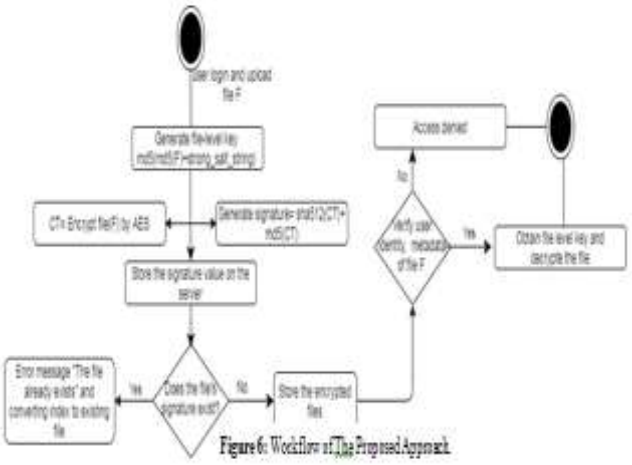


Figure 6: Workflow of The Proposed Approach

4. SECURITY ANALYSIS

Encryption values in De-duplication

In the first part of this section, we show proof from the real world to show that our method for encrypting and decrypting data works. Before the first dataset is sent to the cloud, it is encrypted using an asymmetric encryption method like AES-256. When a data owner uploads files, encryption keys are created and encrypted. These keys are then safely saved in the cloud. Hash methods are used to check the integrity of both the symmetric key and the plaintext, which makes it easier to trust important data. Also, unlike most current solutions, our method only uses encryption and decryption to verify user identity and data integrity. The Consumer Safety Program (CSP) is in charge of making it easier for people to understand product labels and making complete label indices. A label index [10] will be used by the cloud infrastructure to check if users have shared duplicate files.

Security using AES with salted and MD5 as key

The AES algorithm key is made with the help of a key creation function, which pulls information from files that are uploaded to the cloud. People often call this process "data label-based encryption." The Advanced Encryption Standard (AES) uses a key derivation function to create an encryption key that can be used with the AES. For this process to work, the input file must have both a salt number and an MD5 hash. The process involves asking for an MD5 hash of the files, making a random string of text to use as a salt, and

then figuring out the encryption key to use. The data is then encrypted using the given key and AES in a block cipher mode that works for it. Only the salt, the data that has been encrypted, and any initialization vector (IV) that the cipher mode requires are kept. In order to get to the files, you need to figure out the key that was used to protect them. The main goal of the salt is to make precomputation improvements useless, which makes dictionary or brute force attacks less effective. Once the exact salt has been found, a brute force dictionary attack becomes possible. The encryption keys, on the other hand, are stored safely within the variable environment. This makes sure that neither the service provider nor any other entity knows about them, which makes it impossible to find them. The protected files that have been saved with the data cannot be labeled by the service provider. Since encrypted data was used to figure out this data label, it is no longer possible to guess it once the hash of the file was calculated. Because of this, a brute force attack, which is often used in convergent encryption, is no longer possible.

Hash Values in De-duplication

Using the hash number for file-based de-duplication is the method we use in our approach. A cryptographic algorithm is used on the encrypted file that is being saved to figure out the hash value. After the data has been changed or manipulated, the hash value is compared to the new hash value to see if there are any differences. If two files have the same hash number, the files will be deleted. If there is a difference, only the part that was just added is sent to the store medium. People usually think that hashes have one thing in common. The hash number changes whenever any part of the file is changed. This makes it possible for service providers to do updates. The hashes have been indexed and saved in a way that is correct. When a file is updated, the new files are put through a mathematical hashing process and then compared to the files that already exist. The hashes are compared, which leads to their removal and removal of duplicates. Using versioning methods makes it easier to do things like recover data, use bandwidth, save time, and deal with other similar issues. This is done by sending only the new file over the internet, which reduces the chance of problems. There is more use of space, which makes the total de-duplication ratio better. In our method, we use two different kinds of hashes to figure out the file number. The goal of this approach is to make sure that both security and performance are at their best. In both kinds of hashes, an account is gathered and then saved. This approach is used to show how well our system protects against brute force attacks, in which an attacker tries to get into private data by using files uploaded by authorized users.

5. PERFORMANCE EVALUATION

We implemented the suggested approach into our web app. The web application is used by the server. Python was used in the development of the web application. Web application development calls for a 64-bit Linux machine with a 2-Core CPU, 17GB of RAM, and a 1Mbps network link. According to the proposed system, the server stores just one copy of any file uploaded by an encrypted user. We compare our approach to the industry-standard approach stated in reference [68] because the storage in question is a cloud storage system that does not do de-duplication. Figure 7 illustrates how the growing trend of users sharing the same files contributes to our method's efficiency and space savings. A comparison of the storage capacity before and after applying the method to the storage capacity without duplication is also shown in Figure 8. The allocation of storage space due to duplicate data has an effect on both bandwidth utilization and processing performance. To conduct real-time analysis, we keep more than twenty distinct types of files for numerous users. Before submitting a file, we did the math to determine how long each stage of our process would take. The average amount of time spent in overhead needs to be calculated. The results of the experiment are depicted in Figure 9. Client-side encryption, key generation, and decryption only necessitates a marginal increase in processing power. Our approach is more basic and

less complex than that used in previous research investigations, such as those that employed bilinear over encryption and third-party organizations [53][61]. It takes more time to complete a task when a third party is involved. The term for this is "overhead" in planning.

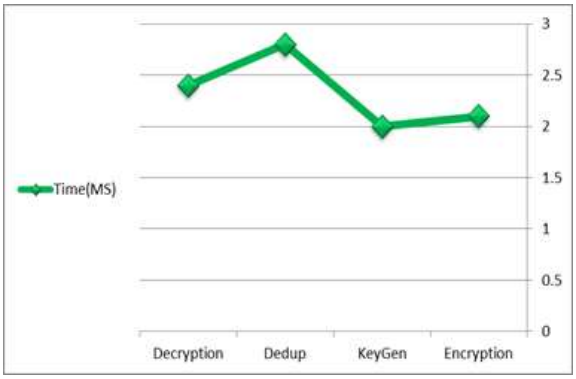


Figure 7: Execution Time of Different Operations.

6. CONCLUSION

Because there is a growing demand for data storage, issues like de-duplication arise and must be investigated and addressed. In this research, we draw from scholarly articles detailing the various techniques for eliminating data duplication. There has been extensive discussion about security worries and issues with the de-duplication tool. This research demonstrates an innovative approach to effectively combining deduplication with industry-standard safe encryption. This is a solid and secure choice. Without going through a third party, users who have access to the same data can obtain the same encryption key from the Internet. The hash of the file is used in a process that generates the key. The key is then complex and protected from attacks by adding a random salt. The service cannot utilize these keys either since they are generated by variables. In order to avoid wasting storage space and transfer rates, it is common practice to check the data mark numbers of all files to ensure that there are no duplicates on the cloud server. Data leakage is prevented thanks to the use of the ciphertext policy attribute, which conceals this value. When necessary, sensitive data is encrypted using the AES-CBC technique to prevent unauthorized access. It's not as complicated as some other forms of security. The effectiveness and security of the strategy were evaluated using the realistic proposal approach. The outcomes demonstrate the efficacy of the proposed strategy in removing superfluous information while maintaining a robust security profile.

REFERENCES

[1] Q. He, Z. Li, and X. Zhang, "Data deduplication techniques," *2010 Int. Conf. Futur. Inf. Technol. Manag. Eng.*, vol. 1, pp. 430– 433, 2010.

[2] K. Hashizume, D. G. Rosado, E. Fernández-Medina, and E. B. Fernandez, "An analysis of security issues for cloud computing," *J. Internet Serv. Appl.*, vol. 4, no. 1, p. 5, 2013, doi: 10.1186/1869-0238-4-5.

[3] S. Hema and A. Kangaiammal, "A SECURE METHOD FOR MANAGING DATA IN CLOUD STORAGE USING DEDUPLICATION AND ENHANCED FUZZY BASED INTRUSION DETECTION FRAMEWOR," *Elem. Educ. Online*, vol. 20, no. 5, pp. 24–36, 2021.

[4] P. K. Premkamal, S. K. Pasupuleti, A. K. Singh, and P. J. A. Alphonse, "Enhanced attribute based access control with secure deduplication for big data storage in cloud," *Peer-to-Peer Netw. Appl.*, vol. 14, no. 1, pp. 102–120, 2021, doi: 10.1007/s12083- 020-00940-3.

[5] E. Manogar and S. Abirami, "A study on data deduplication techniques for optimized storage," in *2014 Sixth International Conference on Advanced Computing (ICoAC)*, 2014, pp. 161–166, doi: 10.1109/ICoAC.2014.7229702.

[6] A. Arya, V. Kuchhal, and K. Gulati, "Survey on Data Deduplication Techniques for Securing Data in Cloud Computing Environment," in *Smart and Sustainable Intelligent Systems*, John Wiley & Sons, Ltd, 2021, pp. 443–459.

[7] Qinlu He, Zhanhuai Li, and Xiao Zhang, "Data deduplication techniques," in *2010 International Conference on Future Information Technology and Management Engineering*, 2010, vol. 1, pp. 430–433, doi: 10.1109/FITME.2010.5656539.

[8] J. Paulo and J. Pereira, "A Survey and

Classification of Storage Deduplication Systems," *ACM Comput. Surv.*, vol. 47, no. 1, 2014, doi: 10.1145/2611778.

[9] Y. Zhang, J. Yu, R. Hao, C. Wang, and K. Ren, "Enabling Efficient User Revocation in Identity-Based Cloud Storage Auditing for Shared Big Data," *IEEE Trans. Dependable Secur. Comput.*, vol. 17, no. 3, pp. 608–619, 2020, doi: 10.1109/TDSC.2018.2829880.

[10] Y. He, H. Xian, L. Wang, and S. Zhang, "Secure Encrypted Data Deduplication Based on Data Popularity," *Mob. Networks Appl.*, 2020, doi: 10.1007/s11036-019-01504-3.

[11] N. Indira and R. Devi, "Cloud Secure Distributed Storage Deduplication Scheme for Encrypted Data," 2018, doi: 10.2991/pecteam-18.2018.26.

[12] Z. Sun, J. Shen, and J. Yong, "DeDu: Building a deduplication storage system over cloud computing," in *Proceedings of the 2011 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2011, pp. 348–355, doi: 10.1109/CSCWD.2011.5960097.

[13] J. Li, Y. Li, X. Chen, P. Lee, and W. Lou, "A Hybrid Cloud Approach for Secure Authorized Deduplication," *Parallel Distrib. Syst. IEEE Trans.*, vol. 26, pp. 1206–1216, 2015, doi: 10.1109/TPDS.2014.2318320.

[14] W. Meng, J. Ge, and T. Jiang, "Secure Data Deduplication with Reliable Data Deletion in Cloud," *Int. J. Found. Comput. Sci.*, vol. 30, no. 04, pp. 551–570, 2019, doi: 10.1142/S0129054119400124.

[15] B. T. Reddy, P. S. Kiran, T. Priyanandan, C. V Chowdary, and B. J. Aditya, "Block Level Data-Deduplication and Security Using Convergent Encryption to Offer Proof of Verification," in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, 2020, pp. 428–434, doi: 10.1109/ICOEI48184.2020.9143055.

[16] W. Ding and R. Deng, "Secure Encrypted Data Deduplication with Ownership Proof and User Revocation," 2017, pp. 297–312, doi: 10.1007/978-3-319-65482-9_20.

[17] P. Puzio, R. Molva, M. Önen, and S. Loureiro, "PerfectDedup: Secure Data Deduplication," in *Data Privacy Management, and Security Assurance*, 2016, pp. 150–166.