An Enhanced Approach to Improve the Security and Performance for Deduplication

Nourah Almrezeq [1], Mamoona Humayun[1], A. A. Abd El-Aziz[1,2] and NZ Jhanjhi[3]
[1]College of Computer and Information Sciences, Jouf University, Al-Jouf, Saudi Arabia
[2] Faculty of Graduate Studies for Statistical Research (FGSSR), Cairo University, Egypt
[3] School of Computer Science and Engineering (SCE), Taylor's University, Selangor

Article History: Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; Published online: 5 April 2021

Abstract: Cloud service providers providing users with efficient and effective storage and transmission of data. To reduce storage costs and save bandwidth, cloud service providers are attracted to use data de-duplication feature. Cloud users are interested in using the cloud safely and privately to protect the data they share on the cloud. Therefore, they encrypt the data before uploading it to the cloud. Since the intent of encryption conflicts with the de-duplication function, the data de-duplication feature becomes a hard problem. Existing de-duplication methods are ineffective in terms of both security and efficiency. They are either vulnerable to brute force attacks that enable the attacker to retrieve files, or they are computationally expensive. That is what drives us to suggest a method for removing duplicate data that is both performance and security effective. We'll start with a description of the implementations and functionality of de-duplication strategies, then move on to the literature that proposes various approaches to de-duplication and the security and efficiency problems that existing approaches face. Via the use of the AES-CBC algorithm and hashing functions, we have proposed an enhancement to improves the performance and protection of data de-duplication for users. Without the involvement of a third party, users' keys are created in a consistent and safe manner. We prove the efficacy of the recommended solution by putting it into practice and comparison with the existing techniques.

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

1. Introduction

1.1 Overview

This decade has seen significant advancements in technology, and has provided many opportunities to both businesses and individuals. Consumers can also access high-quality solutions and a variety of ways to speed up their corporate processes and boost revenues and productions due to recent technological advancements. [1]. In the one hand, technology has advanced, allowing businesses to accelerate the tempo of their transactions and handle their tasks more efficiently, while on the other hand, small and large businesses alike are grappling with a variety of system security issues, the most dominant of which is data protection [2]. Database duplication is one of the serious issue for companies' information network because it leads to data inaccuracy. Due to the increased replication of data collection, users are having trouble finding the correct record. The data stored in the server may have many different types of records, which will consume more storage and reduce the data quality. Data duplication affects the consistency of data on one hand. While on the other side, it is also observed that many insurance companies and different types of financial institutes have to face problems of storage enhancement and security due to having multiple copies of the same data at different places. Optimal techniques for effective cloud storage are an indispensable necessity in the era of big data[3]. One of the best techniques is the de-duplication of data, which is a deletion of duplicate and identical copies of the same file to improve storage efficiency, improve bandwidth and reduce cost[4][5][6]. As shown in Figure 1, de-duplication is a data reduction technique applied to many sides[7]. This technology is also called the concept of intelligent compression or single-instance storage[8]. Data duplication has now become one of the serious challenges, with even major corporations grappling with it. Data duplication causes many issues for individuals, small businesses, and large corporations in terms of data storage and protection. Professionals are having problems in searching and updating records as well. Figure 2 depicts the impact of data redundancy on time and expense in

the system, according to the comparison results [9].Data de-duplication eliminates backup needs by a factor of two[9].

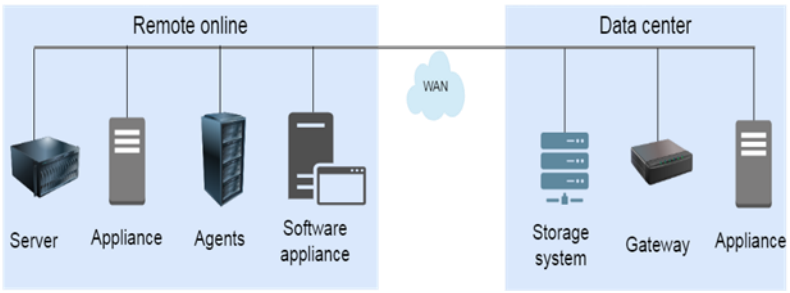

Figure 1:Where de-duplication can applied.

The data duplication in the storage is a very challenging, ineffective, and vulnerable aspect that can affect the data quality, system performance, and data security. Removing unencrypted duplication of data is very simple, but the issue here is removing encrypted data[10].. Due to data duplication, many cybersecurity issues may occur, reducing the security of the sensitive data in local and cloud storage. From a cybersecurity perspective, encryption of the data stored on the cloud is of utmost necessity and of utmost importance to ensure security[11]. But this encrypted data in the cloud makes the application of removing duplicate data a complex and difficult task. Dedu[12] , is an example technology for removing duplicate data but not compatible with encrypted data. We can ensure the confidentiality of data in the cloud by traditional encryption, but it is incompatible with the de-duplication of data. On the other hand, different cipher texts would duplicate data copies to different users, making de-duplication difficult[13][14].To maintain security, data owners encrypt their data before transferring this data to the cloud[15]. But how does a cloud service decide if the same text is the product of multiple encrypted texts[16]. The researchers proposed using a trustworthy third party to help get rid of encrypted duplicates[17][18], [19]. However, it is not advised that it is not advised to provide a totally reliable third party in reality is almost impractical[20]. Convergent encryption is the best practical solution to resolving the conflict between encryption and the removal of redundant data, but it is more vulnerable to brute-force attacks.

Encrypted de-duplication combines encryption and de-duplication to provide data security and storage stability at the same time. However, the two goals mentioned above are diametrically opposed. To deal with the aforementioned conflict, we're working on a new encrypted data de-duplication system. In our article, we will concentrate on taking advantage of the benefits of convergent encryption, which is a more realistic and less expensive approach for removing encrypted data, and we will combine it with the AES-CBC encryption algorithm to improve reliability.. Further, due to lack of resources and low cost, the AES algorithm is a unique approach for de-duplication, achieving the nominal objective of de-duplication and increasing performance. To increase the complexity of the encryption key, it will be developed using the hash derivation function and a long random string salt. And if the attacker is able to propose the hash for the attached file, he would be unable to suggest the additional salt.. After achieving the encryption principle to protect the files, the signature is calculated for the file as a label value for the file to allow its turn to delete duplicated data on the server. The motivation behind conducting this research work is to facilitate data users by providing them easy and fast access to data and enhanced security. A sufficient amount of literature is studied and presented in this paper to evaluate the current methods and techniques used for this purpose, as well as their benefits and drawbacks and based on the shortcomings of these methods, a realistic approach is proposed. The proposed approach is built on a variety of algorithms which allows for the deletion of redundant records and will also aid in data security

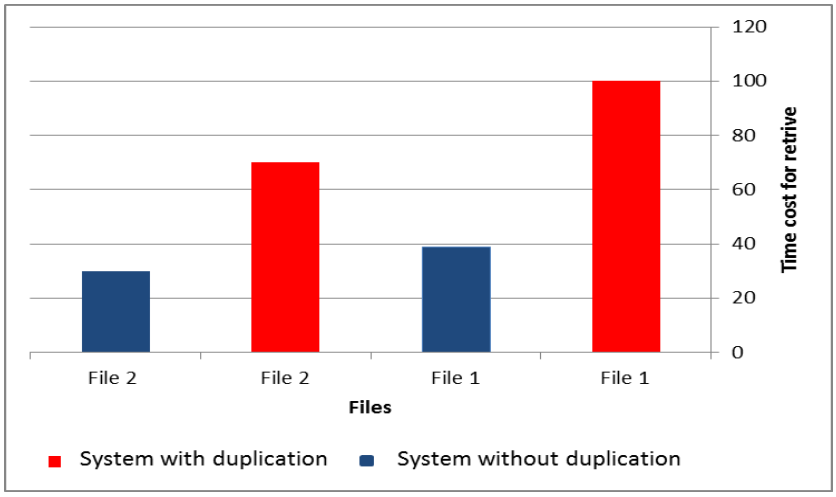

Figure 2: A Comparison results of data with de-duplication[9].

1.2 De-duplication levels

Data de-duplication technology is used to identify duplicate data. Eliminate data redundancy in the overall capability to reduce the need to transfer or store data[21][22]. De-duplication is a technique for detecting duplicate data items; such as judging a file, block or bit, then judging another file. At the present, the strategy of deletion can primarily be used at the file, block, and byte levels, and they can be optimized for storage space. As shown in Figure 3, the de-duplication levels can be divided into three levels: based on file, block and byte[23][24].

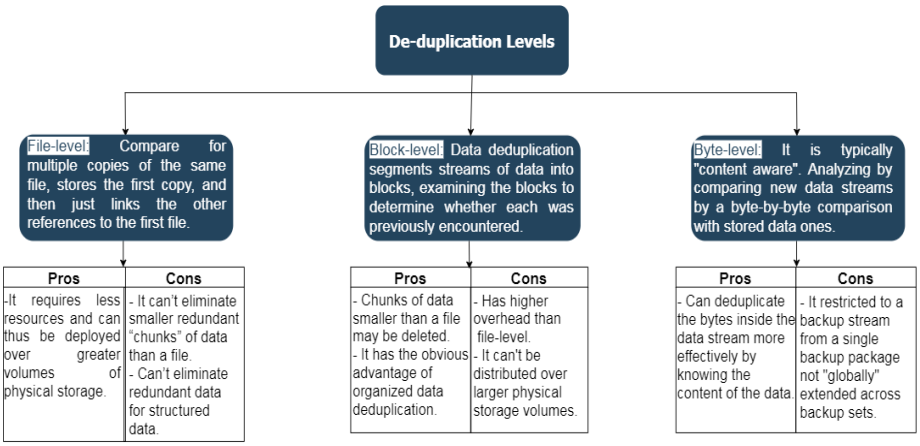

Figure 3: De-duplication Levels.

1.2.1. File-level de-duplication

As the name suggests, file-level de-duplication is based on the file. Single Instance Storage (SIS) is a term used to describe this form of storage[7]. Per file is assigned a unique identifier that is based on the attributes of the file. That identifier is used to store all of the information about that file (as a whole). So that similar files with different names can be removed easily.

1.2.2. Block -level de-duplication

Block-level data de-duplication technology divides a data stream into blocks, reviews each data block, and decides if it met the same data before the block (usually by implementing a hash algorithm for each data block to establish a digital signature or unique identifier)[25]. The block's identifier is also stored in the index if it is special and was written to disk. Otherwise, the only deposit pointer to save the original position of the same data block[26] [27][22].

1.2.3. Byte -level de-duplication

Byte-level de-duplication ironically is a type of file-level de-duplication since detecting a file against another type of data requires some level of content knowledge. Also byte-level de-duplication is a form of block-level

de-duplication that recognizes the data's content, or semantics. Content-Aware-Systems(CAS) is a term used to describe these systems. It is another method is to analyze data at the byte stream stage and then deduplicate it [22].

1.3 De-duplication Applications

Data de-duplication is a specialized data reduction method for removing unnecessary data duplicates. It divides input data into predefined data units such as files or blocks, detects duplicate parts, and keeps a small-size reference to the duplicate parts keeps a small-size reference to duplicate parts instead of storing the original data. Standard data compression techniques tend to reduce the amount of redundant content in a file in order to save space. On the other hand, de-duplication is a technique that uses various data processing algorithms or indexing methods to eliminate redundant data across files. As opposed to traditional data compression, de-duplication is more effective for maximizing storage space, especially for archive backup systems.

1.3.1. Backup storage De-duplication

De-duplication works in tandem with backup systems that are primarily comprised of highly duplicated data, especially for applications that perform cyclical complete backups of a system. By examining the source data, capturing a single copy of a particular data block on the disk, and referencing other copies by creating a block index, de-duplication explicitly distinguishes the duplicate data blocks. This solution dramatically decreases the amount of space needed for source data, allowing users to store more data on their current storage devices. Meanwhile, backups are stored for a longer period of time by storing more data on each computer. As a result of de-duplication, the data center can run more efficiently with less storage devices, lower energy usage, and lower costs[28].

1.3.2. Cloud storage De-duplication

Cloud storage is a category of remote data storage that allows users to access and download data at any time from a variety of platforms. Consumers can also save money by simply accounting for the amount of data they need, which is scalable on demand. . De-duplication must be implemented in today's well-known cloud storage systems to save cost [28][29] because customers frequently have similar files spread across various accounts. In some de-duplication cases, it allows the cloud storage provider to store a single physical copy of identical data in a single venue, reducing both storage capacity and network bandwidth.

1.3.3. Virtual machines storage De-duplication

Host file systems or primary storage systems usually have a lot of duplicate data. Digital machines in such a setting are likely to have similar operating systems mounted. Virtual machines can share storage device settings, applications, and special configurations in the same way[29].

2. Literature Review

This section will provide the detailed overview of data de-duplication and general and its techniques in particular in order to provide the state-of-the art picture of the area under research.

2.1 Data De-duplication

The effective compression technique is an effective data duplication technique that is used for effectively duplicate data elimination. Furthermore, it has also been widely used in cloud storage to reduce the saving bandwidth and the number of the volume in the storage. Furthermore, the study shows that different types of privileges for the system users are also measured in the duplicated record or data checking[30].

Data is very important and it is important for companies as well. To make decisions right, there is a big data that needs to be consistent and accurate to make decisions effective because every company requires this. Moreover, the study emphasizes the need for an effective duplicate data deletion strategy and must have high-quality data for effective decision making [31].

There is a critical issues with the digital data of cloud storage with multiple copies of the records. It is very difficult to handle data duplication in the cloud. The researchers have proposed an approach that helps in reducing data duplication. This study determines tools and tricks for the development based on video and image data [32]. This paper[33] describes that the data duplication removal technologies are divided into two categories: similar data encoding techniques and detection and identical data detection techniques. Researchers have created a survey to collect data on the data duplication problems in organizations. The main aim of this study is to identify challenges and determine the strategies to reduce duplicate records.

As described by[34], they proposed a method that helps in the removal of duplicate data from cloud storage and increases security. Furthermore, a centralized privacy-preserving duplicate removal storage system is built by the researchers. This paper supports block-level and file-level data duplication removal. This paper [5], found that the explosion of data, such as image, audio, or text files, causes numerous problems in both retrieval and storage processes. For storing data, a lot of money is invested by enterprises. Therefore, it needs an efficient technique to handle massive data. They are focused on the techniques of data redundancy elimination strategies

because these are very challenging. The problem is also the case of limited or reduced bandwidth and less data usability. A survey is also created to identify the risks, besides this study is also focusing on optimization techniques.

The duplication of data removal technique is proposed to reduce risks. If the data is stored on the cloud multiple times, it will consume a lot of data storage. The study is providing multiple techniques to develop the system which reduces the duplication in cloud storage. The analysis is also showing that this technique has also removed 20 to 30% of duplicate data from the cloud [35].

Duplicate data removal optimization is very important because the operation will be effective if it is optimized. They proposed a system of a biometric scheme and also proposed an optimized Rabin's algorithm. For the duplication, the effective solution for the problem is to complete the technique application. Furthermore, the file stored in this framework is stored on cloud storage that is separated into various types of block numbers depending on the data record. The proposed solution provides better results in removing duplicates copies of data [36].

Paper[37] described that cloud computing is a very useful and powerful technology providing different types of data storing methods and the data stored in the cloud can easily be accessed from any place. Also, duplicate data need more time to access the data for the expensive hardware, software, and dedicated space. Furthermore, increasing the storage requirements is a time-demanding and challenging task that also needs a larger company infrastructure. Companies also face many problems due to duplicated records. To handle them, they used the checksum algorithm.

With the rapid growth of technology, there is a rapid increase in the data center size and the network technologies are also rising. The green store is being considered by many companies that are hoping to minimize the problems related to storage. The amount of data can greatly be reduced by the data duplication removal technology for the optimization of the storage system. Furthermore, the number of disks that are used in the operation can be reduced by data compression to minimize the cost consumption of the disk energy. The foundation will be laid by studying the data duplication removal strategy, its related processes, and the implementation [1].

In paper [38]the identification of the duplicate and multiple representations process is used for duplicate data removal. Researchers further described that they have introduced two novel data duplication algorithms that will help identify duplicate records from the system and increase the efficiency plus quality of the system. The algorithms can effectively increase the efficiency of the overall process into the available time. Furthermore, by applying these algorithms to the data which is already stored in the storage, they found interesting results and observed that the process is also fast.

According to a study conducted by the IDC analytics group[39], it confirmed that nearly 80% of companies were using techniques to de-duplicate of data in storage systems to get rid of redundant data. Therefore, these researches mentioned above focused on developing effective approaches to de-duplicate data from the perspective of storage efficiency and improving decision-making to achieve the highest quality, effectiveness of data, reducing the frequency range. Despite the importance of maintaining the storage efficiency, the issue of storage security of cloud computing still exists that needs more discussion and in the most urgent need to apply an approach that enhances and balances between storage efficiency and storage security simultaneously. Therefore, we point out that data privacy and security are still considered one of the main concerns facing the data duplication technology, and more efforts are still needed to solve these concerns.

This paper[40], proposes a solution to the conflict between deleting duplicate data and watermarking, given that data owners may choose to convert external multimedia data into highly secure formats. Which makes it difficult to implement the removal of duplicate data because it may be transferred between data owners to different forms. The suggested approach is comprehensive for watermarking and does not require any communication between data owners nor does it depend on a third party. Optimizing the solution using a protocol ensures that similar data for different users will contain the same protected format. This contributes to eliminating duplication even with the watermark.

Paper[41] discusses the problem of managing indexes for duplicate data removal techniques. That is when the duplicate data removal technique seeks to reduce storage and save I/O takes a heavy load of memory. This paper proposes Austerecache for flash caching in order to effectively index memory. Moreover, this suggestion helps to manage Strict caching by basic techniques, to organize data efficiently, and to delete the largest possible number of metadata for indexing. Experiments have shown that the proposal saves 97% of memory with the advantage of maintaining comparable reading ratios and write reduction ratios.

The process of identifying tuples in a relation is known as data de-duplication. Since a similarity value must be computed for any pair of tuples, the problem's complexity is necessarily quadratic with respect to the number of tuples. To avoid comparing tuple pairs that are clearly non-duplicates, blocking strategies are used to split the tuples into blocks, with only tuples from the same block being compared. Data de-duplication remains a costly

issue for large datasets, particularly though blocking is used. In this paper [42], demonstrate how to use parallelism in a shared-nothing computing environment to speed up data de-duplication even further. Dis-Dedup is a delivery technique that reduces the overall workload across all worker nodes while providing good theoretical guarantees.

Finally, this paper[43] develops a solution to remove duplicate data through the use of the Bloom filter. This proposal consists of three main stages: authorized duplicate cancellation. The Administration Center deals with the authorized requests. Then it creates a radix trie and to define the relationship between roles and switches. Moreover to implement BLOM filter for data refresh and efficiency check. Simulation experiments proved that the model provides the maximum repetitive data cancellation rate, up to 25%.

2.2 Data De-duplication Security Issues

With the increase in the importance of using cloud services, cloud providers provide the needs of companies and individuals to store and share their data in the easiest and fastest way. The service provider uses data duplication removal as it allows storing one copy of the file and eliminating duplicates to manage the services provided by the cloud, increase the effectiveness of performance, reduce storage space as much as possible, and reduce bandwidth. Therefore, eliminating duplicate data is an important and indispensable matter [44].

De-duplication of data is useful but it causes problems related to data availability, it is possible data with a unique copy in the cloud may be accidentally deleted, which means causing this unique copy to be lost and inaccessible. Also, from a cybersecurity perspective, the protection and security of data on the cloud are very critical. Although many cloud providers use different encryption techniques, the duplicate data removal technology conflicts with the encrypted content. Therefore, it is not recommended to use traditional encryption while deleting the data because users encrypt the content by themselves, which results in different encryption data for the same data, for this reason it does not work with de-duplication data. Therefore, we can say that deduplicate data opens up new horizons of security problems and challenges that the cloud may face[17] .We will present the literature work about suggested approach for securing the cloud when using de-duplication data.

Cloud service providers use de-duplication technologies to store only a single copy of their content, reduce storage space ,and increase efficiency, but we must consider the security concerns that de-duplicate data creates. This paper[45] suggests Cloudedup is effective and achieves the goal of eliminating duplicate data and provides secure block-level storage. This approach relies on convergent encryption which is a solution to the problem of traditional encryption conflicts with eliminating duplicate of data. The proposed solution relies on securing storage through two-component servers that implement servers that implement an additional encryption procedure and a mechanism for access control and the second component metadata to manage the keys of blocks. The proposed approach protects against offline dictionary attacks but does not provide a guarantee against online dictionary attacks.

The volume of cybercrime is increasing with the development of technology and the Internet revolution. Investigating these crimes, using digital forensic techniques, and analyzing the behavior of criminals must be done to counter these attacks. In this paper[46], a new approach is proposed to de-duplication data that has been processed and consumed by digital forensic, given that the crimes are constantly increasing and the information obtained exceeds acceptable storage. Therefore, law enforcement agencies require fast time and speed in the investigation into requirements. Therefore, there is a need to reduce backlogs of digital forensic analysis that require an enormous amount of effort and time to investigate. The proposed approach to de-duplication data relies on solving a central database problem at the file level. Central databases are used to handle criminal case investigations to reduce costs and provide technologies and share them for digital forensic departments to speed up the investigation process. A hash technique is used for each evidence acquired and compared with the stored values so that these values are not duplicated. This proposal provides a solution to the information overload problems for digital forensic analysts, but this proposal remains vulnerable to attacks because it does not achieve confidentiality.

Another paper discusses the problem of data duplication, but on the Hadoop platform. The Hadoop platform is an open source used to store and process a large amount of data, but in a decentralized manner, by storing data on several devices and then to speed up the processing it is distributed on these devices. Despite the importance of Hadoop, it is still not characterized by effective storage because it is possible that it duplicate storing the same file several times when this file is saved under a different name. This is a waste of storage space and reduced processing efficiency on devices. This paper[47] proposes DeDup approach to eliminating duplication data by calculating the hash value at the file level before uploading it to HDFS and comparing it with existing files. When download the file by any user, check to Hbase through the hash account to see if the files download the same content or not.

Fog-assisted secure data de-duplication scheme (Fo-SDD) [48] is an approach proposed to improve communication efficiency and remove duplication data in a secure manner by fog-assisted mobile

crowdsensing. The proposed system is able to improve the accuracy of the tasks depending on the user's mobility. The proposed system is based on the BLS-oblivious pseudo-random function and it is able to discover duplicate data and delete it easily but without recognizing the content. The proposed approach increases privacy for those who use mobile by masking their identity and achieving anonymity by using the Chameleon hash algorithm. Another study[49] solves the problems related to auditing sensitive data in the untrusted cloud, among these disadvantages is the destruction of data by using the malicious cloud service provider and the demolition of the ability to recover the data that has been destroyed. In addition to another problem, storing data frequently that may lead to duplication data and increased costs. The proposed approach is public auditing and secure de-duplication scheme based on block chain with equal arbitration. It is done through smart contracts that help prevent malicious cloud service providers and recover affected user data. The proposed relies mainly in auditing data and removing random caching on the algorithm of locked-messages. The proposed approach is proven effectiveness by the Ethereum blockchain.

The de-duplication data feature offers many advantages, such as reducing storage cost and many advantages such as reducing storage cost and increasing the effectiveness of cloud services[50]. On the other hand, this feature impacts data fragmentation and processors. This paper[51] proposes MUti-tiered and SLA-drivEn to address these challenges. The proposed approach based on a de-duplication-oriented service level agreement as a protocol to improve the services provided. In addition, de-duplication data developed is considered a dynamic system. This approach is effective for improving performance and eliminating duplication data compared to other approaches. Moreover, this approach works accurately with a number of parameters that critical to eliminate duplication data.

Cloud service providers are interested in using de-duplicate data to manage storage more efficiently and avoid storing duplicate copies of the same file. De-duplicate data with data privacy is very difficult. Due to users resort to encrypting their sensitive data before uploading it to cloud service providers or external sources. Several proposed encryption techniques can be used while eliminating duplicate data. This paper[52] demonstrates the security of the suggested approach. It adopts a server-aided approach to de-duplicated data by creating a fixed size of ciphertext. Using symmetric encryption, bi-linear Re-encryption by mapping and proxy.

One of the leading technologies in the cloud service field is searchable encryption technology, whereby the service provider can search and control the files that the user shares even when they are encrypted. Paper proposes a new attribute-based approach to keywords in order to eliminate duplicate data on encrypted data. Moreover, effective access and data confidentiality is achieved through attribute-based encryption while data integrity achieved by hash function and third-party auditor. This approach[53] enforce fine-grained permission to scan and enforce indistinguishable keywords. According to the experiments conducted on this approach, it is evident that the telecommunication overheads are very low compared to other approaches due to the outsourcing of encryptions to reduce the high cloud load.

Moreover, paper[54] proposes a solution to data leakage problems during the de-duplication of data. They suggested a tunable encrypted de-duplication. One of this approach's advantages is that it strikes a balance between the de-duplication of data and confidentiality. The central premise is that the key derivation is based on the chunk content and number of duplicate chunk copies, but this study still has problems for recovering files. The previous work[54] was developed to increase performance effectiveness and reduce data leakage. This study[55] was applied from both an attack and defense perspective to understand how frequency analysis could affect data leakage when using de-duplication of data . An inference attack was suggested in relation to the attack while MinHash was introduced encryption and scrambling as a defense function. The proposal achieves higher storage efficiency and reduces inference attacks as assessed by the trace-driven.

The de-duplicate of data by relying on external parties is not satisfactory in terms of security and efficiency. This proposal[10] improves the de-duplicate of data through the popularity of the data. The files are classified into two types: Popular and Unpopular. Initially, both files are declared unpopular .Then, these unpopular files are encrypted using two separate encryption layers .The outer protection layer is eliminated later when unpopular files' popularity level is met and the file becomes popular[56]. Data popularity used without the support of any third party, the cloud service will execute de-duplication. Using bilinear mapping, tags are determined to decide whether separate encrypted data comes from the same plain text. Attribute-based encryption of the cipher text policy is used to secure the mark. The efficiency of this proposed approach has been demonstrated through simulation experiments and security analysis. Another study addresses the problem of supporting ownership modification during cloud auditing and de-duplication of data [57].This paper develops the first proposal to audit the integrity of data stored in the cloud and supports ownership modification. This approach provides data integrity through dynamic control. The identity-based broadcast algorithm, is used to prevent access to non-owners or to owners who have been revoked. Moreover, Randomized convergent encryption used to encrypted the original data .Its proved efficiency through analysis and experiments[58][59].

2.3 Security Concerns of De-duplication

Despite how much de-duplication has progressed, there are still various problems and issues associated with redundant data removal strategies, including attacks that threaten the privacy of cloud-based content. The removal of redundant data will result in a variety of attacks on consumer privacy. We look at several well-known attack models that exploit cloud storage providers' data de-duplication.

2.3.1. Expose file content

This type of attack helps the attacker to obtain knowledge of the file contents by uploading them to the cloud in plain text and then moving them to an untrusted channel. The intruder will make several copies of a file prototype with all of the possible file content values exclusive to a specific user, exposing file contents.. The attacker can brute-force fill out the file models and upload each version to the cloud storage service to assume the existence of a file. The attacker gets access to the value entered in a version that has been marked as having been previously submitted after it has been marked as having been previously submitted.. For example, the attacker obtains a copy of an employment contract prototype file and is aware of other users who are likely to store their contracts in the cloud[27]. Furthermore, if the service provider or a third party is not trusted, the original content of the files saved to the cloud by the user may be revealed.

2.3.2. Brute-force attack

There are a few attack options if de-duplication can be identified. Only theoretically is a brute-force attack using has manipulation feasible. During the attack, the attacker must check for hash collisions using a variety of hashes that he will create from a file[60][61]. If a hash collision occurs and a similar hash remains, it is presumed that the same file has already been submitted to the server. The only way to find cryptographic collisions is to use a brute force attack. Data deduplication is vulnerable to brute force attacks because it necessitates hashing of files or chunks. A brute force attack is an adversary's exhaustive attempt to target encrypted information and obtain preimages, which is virtually impossible due to the message digest's duration. As a result, a minimal brute force is used in practice, where the adversary knows a set of preimages and a hash function. The attacker then attempts to create a hash collision by computing all possible hash values of the preimages to match the known hash value.

2.3.3. Birthday attack

Birthday attack is a form of brute force attack that corresponds to the mathematical Birthday paradox problem. It uses probabilistic theory to calculate hash function collisions, which means that the hash values generated for different user files are the same. A hash collision occurs when the hash value generated by a hash function is identical for a variable-sized input[27].

2.3.4. Side channel attack

Identifying a specific file and identifying the file content are also possible using the side channel attack. This means identifying a specific file within a group of files. The attacker must make an assumption about a file that the victim has and complete the rest of the task. De-duplication can be identified by tracking network traffic or the file's upload status, but monitoring network traffic is one of the better ways to see if de-duplication is taking place on that server[62]. As a result, the attacker has uploaded the correct file, which will result in a bandwidth consumption shift in network traffic compared to the previous one. An attacker can perfectly determine whether the file is already uploaded or in the hands of the user after verifying this aspect, if there is de-duplication in the cloud.

2.4 Encrypted De-duplication

2.4.1. Message-Locked Encryption

Cloud users often tend to store their data in encrypted form due to the risk of content leakage while using outsourced data. Traditional encryption is incompatible with deduplication since it assumes that users encrypt messages with their own unique keys, resulting in similar messages being encoded as distinct ciphertexts, preventing deduplication. Define a cryptographic primitive called message locked encryption (MLE)to allow encrypted replication storage , Convergent encryption is a well-known example of MLE, which uses a uniform derivation function to extract the encryption key from the message itself, allowing the same message to symmetrically return the same ciphertext[61]. The derivation function is the cryptographic hash of the message text[63]. This method ensures that identical data produce identical ciphertext after encryption, allowing the cloud storage provider to deduplicate the encrypted user data. Since the encryption key is derived deterministically from the data material, a malicious cloud storage provider may use the same algorithm to classify the owners of specific files. Side-channel attacks are also possible by using deterministic encryption in combination with client-side deduplication. The intruder will obtain the encrypted data's key and use it to decide the presence of the data's original plain text in the cloud. Furthermore, MLE is vulnerable to brute-force cipher-text recovery attacks by default. The adversary will sample all messages, determine each message's MLE key,

and compute the ciphertexts. The adversary may deduce the target message if one of the computed ciphertexts matches the ciphertext of the target message[64].

2.4.2. Single-Server Cross-User De-duplication

Client-side encryption is possible with this scheme. The PAKE protocol, which allows clients to receive the encryption key from another client who has previously uploaded the same file, is at the core of the process. When a file is being presented for the first time, a random key should be used to encrypt it. This method, on the other hand, makes a side-channel attack in which clients can infer the absence of a specific file in the cloud. Any of the session keys could be replaced with false values, and the keys could be sent to the cloud storage provider. The single-server cross-user de-duplication with client-side encryption avoids server-side attacks, in which the cloud storage provider attempts to distinguish clients who have uploaded a specific file without reducing bandwidth usage[65].

2.4.3. Symmetric or asymmetric encryption

Before being uploaded to the CSP, the data is encrypted using a symmetric or asymmetric encryption algorithm. The data owner generates the encryption keys, which are then encrypted using the ciphertext policy before being stored in the CSP [10][53]. Since the keys do not meet the criteria, the CSP is unable to access them. Since a malicious user cannot decide if any data are stored in the CSP based on the information returned by the CSP, he is unable to run online brute-force attacks on encrypted data, this method prevents online brute-force attacks. This method creates a bilinear mapping based on bilinear maps and attributes in order to achieve high computation while encrypting and decrypting data[66]. As a result, the most pressing problem is to lower the cost of computation[67]. The outsourced decryption scheme, which is not recommended in practice, is used to solve this issue[10].

Table 1: Litreture Review.

Paper	Approach	Algorithms	Security Goals		Pros	Limitation
[45]	ClouDedup	Convergent Encryption	Confidentiality Integrity Availability	√ √ √	Security guarantee against offline dictionary.	Vulnerable to online dictionary attack.
[46]	Central Database	Hash	Confidentiality Integrity Availability	× √ √	Improves the effectiveness of digital forensic investigation by deduplication of evidence.	Vulnerable to attacks it does not achieve confidentiality.
[47]	DeDup in Hadoop	Hash	confidentiality Integrity Availability	× √ √	Increase the effectiveness of Hadoop.	Vulnerable to attacks it does not achieve confidentiality.
[48]	Fog-assisted secure data deduplication	AES & Hash	Confidentiality Integrity Availability	√ √ √	Security guarantee against brute-force attacks and "duplicate-replay" attacks.	Limited to users of mobile devices and fog nodes.
[49]	Blockchain-based public auditing and secure deduplicatio	AES & Hash	Confidentiality Integrity Availability	√ √ √	Punish the malicious CSP and reimburse users whose data integrity is lost.	Suggested solution is limited to cloud service providers (server-side).
[51]	MUti-tiered and SLA-drivEn deduplication	Hash	Confidentiality Integrity Availability	× √ √	Accurately with a number of parameters that critical to eliminate duplication data(offline, chunk-level, file-level, dynamic).	Vulnerable to attacks it does not achieve confidentiality.
[52]	Server-aided data deduplication	AES	Confidentiality Integrity Availability	√ √ √	Achieve Privacy of data, Possession proof,Immune to attack and scalability.	Not to have end-to-end data integrity (both client and server) Confirmation.
[53]	Attribute-based encryption	AES & Hash	Confidentiality Integrity Availability	√ √ √	Communication overheads are very low compared to other approaches.	Used third-party auditor and outsourcing decryption.
[54]	Tunable encrypted deduplication	AES & Hash	Confidentiality Integrity Availability	√ √ √	Provide balances between storage efficiency and data confidentiality	No optimizations for restoring files are supported.
[55]	Frequency Analysis	AES & MinHash & scrambling	Confidentiality Integrity √	√ √	Reduces the intensity of inference attacks while retaining a high	The approach is only proven effective against an inference attack.

						performance in storage.	
[10]	Data popularity	Elliptic Curve & Hash	Confidentiality Integrity Availability	√ √ √	Perform deduplication without the assistance of third party.	Security is provided for all the unpopular data whereas for popular data the security is slightly weaker.
[57]	Deduplicated data integrity auditing	AES & ElGamal	Confidentiality Integrity Availability	√ √ √	Supports ownership modification even for owners who have been revoked.	Suggested solution is limited to cloud service providers (server-side).

3. PROPOSED APPROACH

3.1 Specification requirements

The proposed approach will be expected to have the following functionality requirement:

- Efficiency: the proposed approach achieves the ability to effectively remove duplicate data to save more space and reduce cost as well as reduce network frequency. The authorized data user can handle the data on the cloud easily, effectively and quickly. The cloud service provider can also handle encrypted duplicate data.
- Security: the proposed approach achieves a high level of security for all entities. The service provider could remove duplicate data without revealing the plain text. The data owner cloud also upload files to the cloud in an encrypted way. For the data user, it lies in the fact that the authorized user is the one who can access the data and show it as plain text.
- Computation: this proposed approach must be lightweight secure storage to reduces the management and computation cost. Moreover, achieve dispensation from the management of the third party.
- Auditing: to ensure that the user's validation will fail if the cloud destroys the user's data.

3.2 System model

As described in Figure 4, there are three cloud server provider, data owner and data user.

- Data owner: who uploads data to the cloud then used cloud services to store encrypt files. Also responsible for the decrypted file if access structure from the cloud is granted.
- Data users: who uploaded the file and now he to want access to files on the cloud. If the user obtains the file that means the access policy defined by the data owner satisfies the user key. Through carrying out the cloud storage auditing procedure for the cloud, users can check the integrity of the cloud data.
- Cloud service provider: is responsible to handle and storing encrypted files in the cloud, also responsible for auditing the access structure for data usage. Moreover, compute data authenticators, and check data integrity by verifying the accuracy of the findings of the search. Also, responsible for data de-duplication of files.

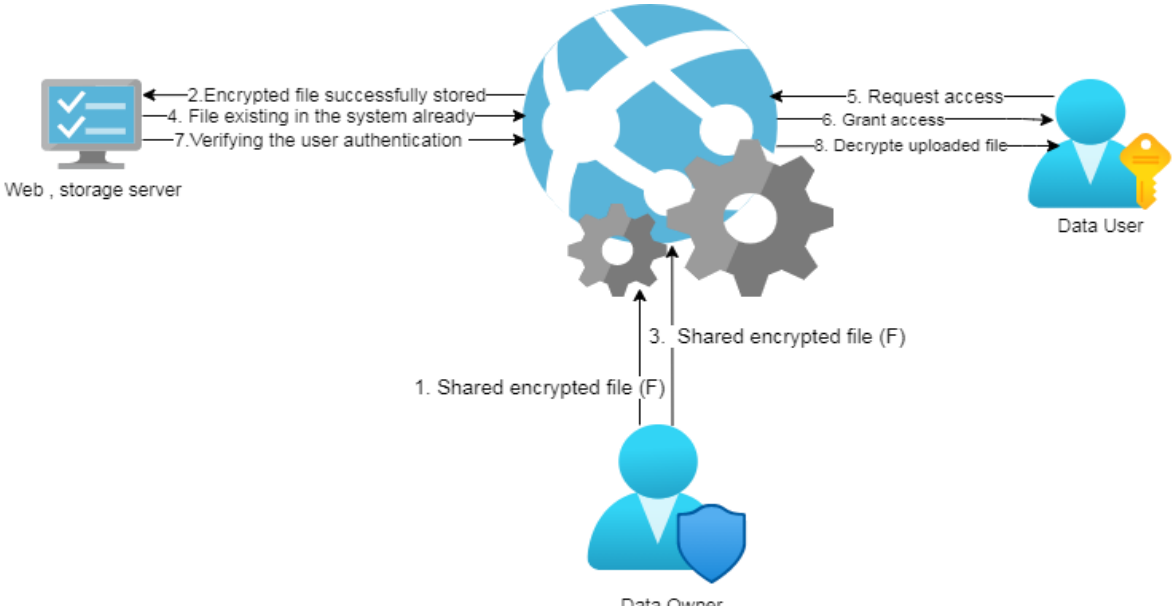

Figure 4: The Proposed Model.

3.3 Security model

The following algorithms are used to de-duplicate data in cloud storage while still supporting good encryption and protection:

- Setup algorithms(S): configuration of a system. It initializes a security parameter.
- KeyGeneration(KeyGen): the key generation by input the MD5 of uploaded data with strong string salt , then concatenation them and again compute the MD5. Moreover, the initialization vector of AES algorithm. Then, the outcomes will be a shared key.
- Encrypt algorithm (E): The input contains the shared key (K), the plaintext (PT). The output will be cipher text (CT).
- Integrity Check(i): integrity check run by cloud service to ensure the integrity by setting limited access permissions on the uploaded data.
- Access structure schema (AS): An access structure corresponds to the cipher text. The attribute is associated to the key. That is, if the attributes in the attribute set meet the access structure, data can be decrypted.
- Decrypt (D). The input includes shared key(K) and CT. The output will be plain text(PT).
- DedupCkeck(Dedup): The data de-duplication algorithm perform by compute the data label based on hashes value of encrypted data. Its input is encrypted data, and the output is 0 (store data) or 1 (not store, de-duplication).

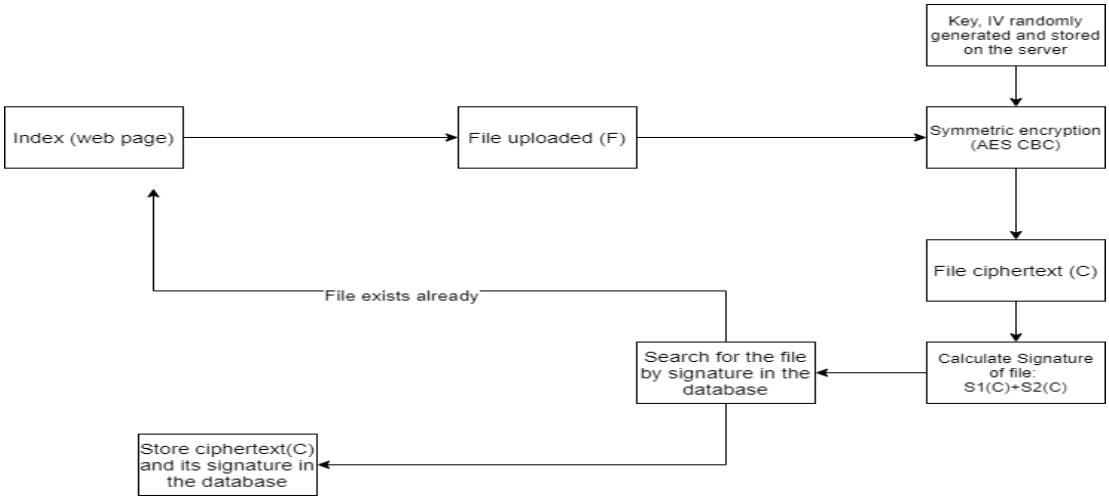

Figure 5: Security model.

3.4 Development

Setup algorithm:

Key generation:
Input: Security parameter
Output: shared key(K)
1- seed(time(0))
2- for i ∈ [1..32]
Calculate Ki=MD5(MD5(PT)+Salt)) , (32,127) //printable char
3- 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 setup of the application, the server will generate the parameters for AES cipher and store them in a local file read-protected only by root user. In both parameters, the randomness of the server will be seeded using current live time as an integer (timestamp). Then it'll generate a random sequence of 32 bytes for the key and 16 bytes for the IV.

Handle Data:

Input: Uploaded File UF
Output: None
1- cont <--- FIleContent(UF)
2- Cipher <--- Encrypt(cont,K,IV)
3- sig = data_label(Cipher)
4- Exists <--- Search for "sig" in database
5- if Exists
    Return "File exists"
  else
    Store_file(Cipher)
    Store_signature(sig)
6-return Success

Data label algorithm:

Input: Ciphertext C
Output: Signature of the ciphertext
1- S1 <--- MD5(C)
2- S2 <--- SHA512(C)
3- result <--- S1+S2
4- return result

Encryption algorithm:

Input: File F
Output: Ciphertext C
1- cont <--- FileContent(F)
2- PaddedCont <--- Pad(cont,AES.BLOCK_SIZE)
3- C <--- EncryptAEScbc(PaddedCont,K,IV)
4- return C

The server will apply the encryption function on the content of every uploaded file using the parameters (K and IV) generated during the setup phase. It'll handle any length of data provided by user using the default pad function.

Decryption algorithm:
Input: File CF
Output: Plaintext Pt
1- cont <--- FileContent(CF)
3- C <--- DecryptAEScbc(cont,K,IV)
4- UnpaddedCipher <--- Unpad(C,AES.BLOCK_SIZE)
5- return UnpaddedCipher

The file F is uploaded through the web interface. The user will proceed with the encryption phase. It uses AES CBC with key size equal to 256 bits. AES cipher parameters are generated during the server setup and stored in a file on the filesystem read-protected only by root and an executable wrapper will be created with suid permissions to read the key. It'll be triggered using python's subprocess. Popen method the server will apply :C = E(F,key,IV)Then, the server will calculate the signature Sig of the ciphertext. It'll apply sha512 function named S1 and md5 function named S2 ,Sig = S1(C) + S2(C).Then it'll search for a match in the database with a safe SELECT query using the calculated signature.

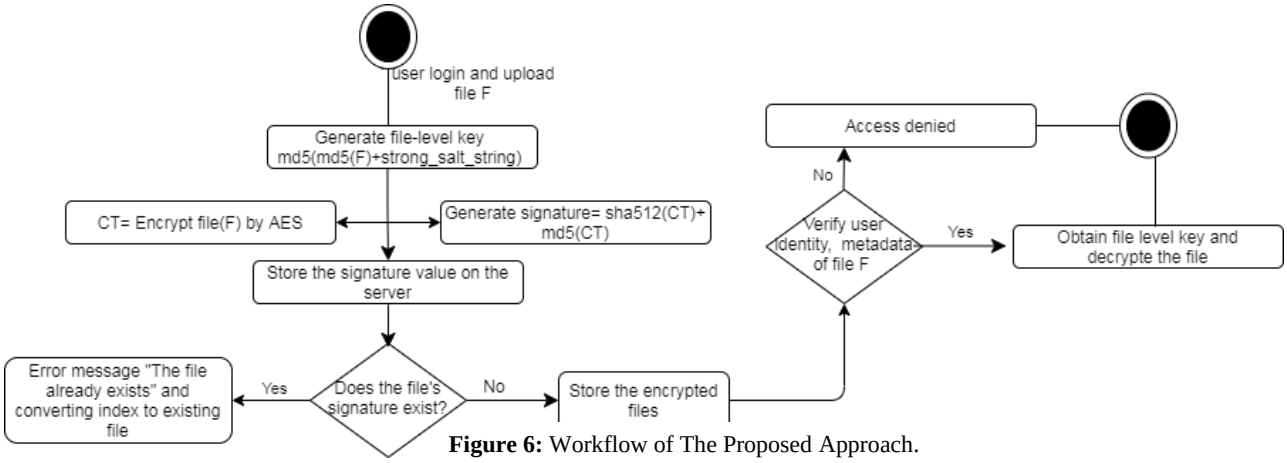

Figure 6: Workflow of The Proposed Approach.

4. Security analysis

4.1 Encryption values in De-duplication

In this section, we first show the correctness of our approach, with respect to the encryption and decryption algorithm. Before being uploaded files to the Cloud, the original data is encrypted using an asymmetric encryption algorithm such as AES-256. When a data owner uploads files, the encryption keys are created and encrypted before being stored in the cloud. In addition, the user checks the symmetric key and plaintext using hash functions to ensure the integrity of interesting data. Furthermore, our solution uses encryption and decryption as authenticators and data integrity verifiers without relying on a third party, which is generally not recommended. The CSP is in charge of assisting consumers with label meaning and label index generation. Using the label index, the cloud will search whether a file submitted by a user is duplicated or not [10].

4.2 Security using AES with salted and MD5 as key

The key to the AES algorithm is generated by deriving the data from files uploaded to the cloud,which is known as data label-based encryption by using a key derivation function. A key derivation function takes salt and an MD5 of the supplied file and produces a key that can be used with AES. To encrypt, you would prompt an MD5 of files, generate a random string salt, and derive the key. Then use that key with AES in a suitable block cipher mode to encrypt the data, and store only the salt and the encrypted data (and whatever IV the cipher mode requires). To decrypt re-derive the key to decrypt the files. The purpose of the salt is to prevent precomputation optimizations from being applied to a dictionary or brute force attack. It is indeed possible to perform a brute force dictionary attack once the salt is known. However, the cryptographic keys will not be disclosed to the service provider or any other party and are saved in an environment for the variables, which makes detection impossible. Also, the service provider cannot identify the data to label the stored encrypted files as well. The brute force attack cannot be applied as is the case in convergent encryption, because This data label was calculated on the encrypted data and cannot be predicted once the hash is calculated for the file.

4.3 Hash Values in De-duplication

In our approach, the hash value is already used in the file-based de-duplication. The stored encrypted file is used to produce the hash value. This hash is compared to the new to see if there is a difference when the data is modified or processed. Files that have the same hash will be removed. Only the newly inserted portion is sent to the storage if there is a discrepancy. Hashes are considered to have one. When one section of the file is modified, the hash is updated as well, allowing service providers to update it. These hashes are stored and indexed properly. The new files are mathematically hashed and compared to the previous ones when a file is modified. As a result, the hashes match; they are not stored and therefore de-duplicated. . Since only the modified file is sent over the internet in this situation, the versioning mechanism aids data recovery, bandwidth usage, time consumption, and so on. It saves more space and improves the overall de-duplication ratio. In our approach, the hash for files will be calculated with two different types of hash to achieve efficiency and increase security. An account is collected for the two hash functions and then stored. Through this method, we confirm the effectiveness of our system against brute force attacks when the opponent wants to violate data privacy through files uploaded by legitimate users.

5. Performance evaluation

We implemented the proposed approach as a web application . The web application is used on the server side.. The web application is created using a python programming language. To implement the web application is implemented on a 64-bit Linux with a 2-Core CPU, 17GB RAM, 1Mbps network connection. In our approach, the server stores only one copy of the encrypted uploaded files even when it is uploaded from different users. We compare our approach with [68] current approach as a standard, because it is a cloud storage and the storage does not support de-duplication. As shown in Figure 7, the number of users who keep similar files is increasing, so our scheme is more efficient in terms of performance and storage. Moreover, Figure 8 shows a comparison between storage capacity with duplication before using the approach and after using the approach without duplication. We also note that the proportion of duplicate data takes up a portion of storage capacity, which in turn affects performance and bandwidth. In order to evaluate the approach in real-time, we store more than 20 different files for different users. First, we uploaded a file and calculated the time period for each stage of our approach. Then, calculate the average time overhead. The experimental results are shown in Figure 9. The overhead of client-side operations (encryption, and key generation and decryption) is relatively low. When we compare our approach with approaches that use third party and bilinear over encryption have been used in research work such as [53][61]. The time overhead is higher due to the third-additional party's activities and the coordination overhead the average is calculated as the result.

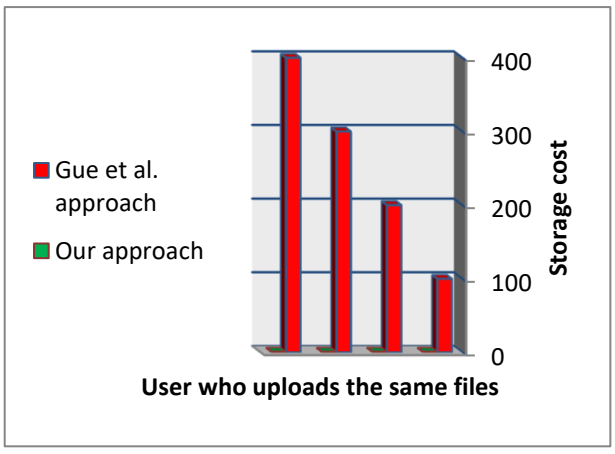

Figure 7: Storage Overhead Comparison**.**

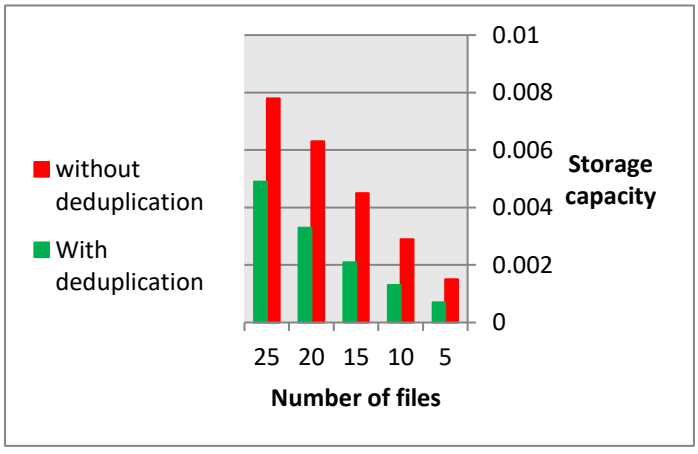

Figure 8: Storage Capacity.

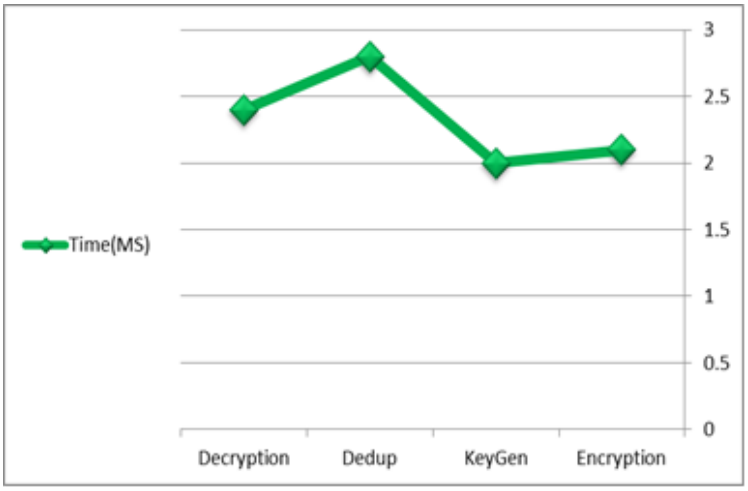

Figure 9: Execution Time of Different Operations.

6. Conclusion

Due to the challenges that increased demand poses to data storage, de-duplication is an issue that can be researched and developed. We provided a literatures in this paper that discussed various approaches to de-duplicate data. We've outlined the problems with the de-duplication application's success as well as security concerns. In this paper, we proposed a secure approach that uses traditional secure encryption and supports the deduplication feature. Users with the same file can obtain the same encryption key without using any third party over the Internet. These keys are created through the function of deriving a key from the hash of the file and add additional random salt to complicate the key and protect it against attacks. Moreover, the server cannot access these keys, because the key creation takes place within the variables environment. The data mark value of each file is often compared to effectively de-duplicate on the cloud server and reduce network bandwidth. Since the ciphertext policy attribute protects this value, no information about the data can be leaked. The encryption is applied through the AES-CBC algorithm to secure the system from attacks and it is lightweight compared to other security algorithms. We did a security and efficiency review and evaluated the scheme using the realistic proposal method. The results reveal that the proposed method performs well at removing redundant data while still being secure against attacks.

Acknowledgements

The authors would like to thank the Deanship of Graduate Studies at Jouf University for funding and supporting this research through the initiative of DGS, Graduate Students Research Support (GSR) at Jouf University, Saudi Arabia.

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.

[18] J. Li, J. Li, D. Xie, and Z. Cai, "Secure Auditing and Deduplicating Data in Cloud," *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2386–2396, 2016, doi: 10.1109/TC.2015.2389960.

[19] M. Bellare, S. Keelveedhi, and T. Ristenpart, "DupLESS: Server-Aided Encryption for Deduplicated Storage," in *Proceedings of the 22nd USENIX Conference on Security*, 2013, pp. 179–194.

[20] H. Cui, R. H. Deng, Y. Li, and G. Wu, "Attribute-Based Storage Supporting Secure Deduplication of Encrypted Data in Cloud," *IEEE Trans. Big Data*, vol. 5, no. 3, pp. 330–342, 2019, doi: 10.1109/TBDATA.2017.2656120.

[21] M. Bhadkamkar *et al.*, "BORG: Block-reORGanization for Self-optimizing Storage Systems," 2009.

[22] A. Clements, I. Ahmad, M. Vilayannur, and J. Li, "Decentralized deduplication in SAN cluster file systems," p. 8, 2009.

[23] S. P. G., N. R. K., V. G. Menon, V. P., M. Abbasi, and M. R. Khosravi, "A secure data deduplication system for integrated cloud-edge networks," *J. Cloud Comput.*, vol. 9, no. 1, p. 61, 2020, doi: 10.1186/s13677-020-00214-6.

[24] Y. Gao, H. Xian, and A. Yu, "Secure data deduplication for Internet-of-things sensor networks based on threshold dynamic adjustment," *Int. J. Distrib. Sens. Networks*, vol. 16, p. 155014772091100, 2020, doi: 10.1177/1550147720911003.

[25] J. Guerra, L. Useche, M. Bhadkamkar, R. Koller, and R. Rangaswami, "The case for active block layer extensions," *Oper. Syst. Rev.*, vol. 42, pp. 3–9, 2008, doi: 10.1145/1453775.1453778.

[26] G. Ali, M. Ilyas Ahmad, and A. Rafi, "Secure Block-level Data Deduplication approach for Cloud Data Centers," in *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2020, pp. 1–6, doi: 10.1109/iCoMET48670.2020.9074109.

[27] A. K. M. B. Haque, "ANALYSIS OF ATTACK TECHNIQUES ON CLOUD BASED DATA DEDUPLICATION TECHNIQUES." 2019, doi: 10.13140/RG.2.2.18801.74089.

[28] K. Rajkumar and V. Dhanakoti, "Methodological Methods to Improve the Efficiency of Cloud Storage by applying De-duplication Techniques in Cloud Computing," in *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 2020, pp. 876–884, doi: 10.1109/ICACCCN51052.2020.9362940.

[29] A. M. and D. B., "Security Analysis and Preserving Block-Level Data DE-duplication in Cloud Storage Services," *J. Trends Comput. Sci. Smart Technol.*, vol. 2, pp. 120–126, 2020, doi: 10.36548/jtcsst.2020.2.006.

[30] G. Prakash, K. Snehal, A. Khiste, and P. Rohini, "A Survey Paper on Removal of Data Duplication in a Hybrid Cloud," *Int. Res. J. Eng. Technol.*, vol. 03, no. 01, p. 6, 2016.

[31] P. Pahwa and R. Chhabra, "BST Algorithm for Duplicate Elimination in Data Warehouse," *Int. J. Manag. Inf. Technol.*, vol. 4, pp. 190–197, 2013.

[32] R. Kaur, I. Chana, and J. Bhattacharya, "Data deduplication techniques for efficient cloud storage management: a systematic review," *J. Supercomput.*, vol. 74, no. 5, pp. 2035–2085, 2018, doi: 10.1007/s11227-017-2210-8.

[33] Li AO, S. Ji-Wu, and L. M.-Q. LI, "Data Deduplication Techniques," *Natl. Lab. Inf. Sci. Technol. (TNList).*, vol. 04, no. 01, pp. 1–11, 2015, doi: DOI: 10.3724/SP.J.1001.2010.03761.

[34] H. Yan, X. Li, Y. Wang, and C. Jia, "Centralized Duplicate Removal Video Storage System with Privacy Preservation in IoT.," *Sensors (Basel).*, vol. 18, no. 6, Jun. 2018, doi: 10.3390/s18061814.

[35] D. Meister, J. Kaiser, A. Brinkmann, T. Cortes, M. Kuhn, and J. Kunkel, "A study on data deduplication in HPC storage systems," in *SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, 2012, pp. 1–11, doi: 10.1109/SC.2012.14.

[36] J. Min, D. Yoon, and Y. Won, "Efficient Deduplication Techniques for Modern Backup Operation," *IEEE Trans. Comput.*, vol. 60, no. 6, pp. 824–840, 2011, doi: 10.1109/TC.2010.263.

[37] A. F. Osuolale, "Data Finding, Sharing and Duplication Removal in the Cloud Using File Checksum Algorithm," *Int. J. Res. Stud. Comput. Sci. Eng.*, vol. 6, no. 1, pp. 23–44, 2019.

[38] T. Papenbrock, A. Heise, and F. Naumann, "Progressive Duplicate Detection," *Knowl. Data Eng. IEEE Trans.*, vol. 27, pp. 1316–1329, 2015, doi: 10.1109/TKDE.2014.2359666.

[39] L. DuBois, A. Marshall, and E. Sheppar, "Key considerations as deduplication evolves into primary storage," *White Pap.*, vol. 223310, 2011.

[40] W. You, B. Chen, L. Liu, and J. Jing, "Deduplication-Friendly Watermarking for Multimedia Data in Public Clouds," in *Computer Security -- ESORICS 2020*, 2020, pp. 67–87.

[41] Q. Wang, J. Li, W. Xia, E. Kruus, B. Debnath, and P. P. C. Lee, "Austere Flash Caching with Deduplication and Compression,"

2020.

[42] X. Chu, I. F. Ilyas, and P. Koutris, "Distributed Data Deduplication," *Proc. VLDB Endow.*, vol. 9, no. 11, pp. 864–875, 2016, doi: 10.14778/2983200.2983203.

[43] S. E. Ebinazer, N. Savarimuthu, and M. S. B. S, "An efficient secure data deduplication method using radix trie with bloom filter (SDD-RT-BF) in cloud environment," *Peer-to-Peer Netw. Appl.*, 2020, doi: 10.1007/s12083-020-00989-0.

[44] P. Singh, N. Agarwal, and B. Raman, "Secure data deduplication using secret sharing schemes over cloud," *Futur. Gener. Comput. Syst.*, vol. 88, pp. 156–167, 2018, doi: https://doi.org/10.1016/j.future.2018.04.097.

[45] P. Puzio, R. Molva, M. Önen, and S. Loureiro, "ClouDedup: Secure Deduplication with Encrypted Data for Cloud Storage," in *Proceedings of the 2013 IEEE International Conference on Cloud Computing Technology and Science - Volume 01*, 2013, pp. 363–370, doi: 10.1109/CloudCom.2013.54.

[46] M. Scanlon, "Battling the digital forensic backlog through data deduplication," in *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, 2016, pp. 10–14, doi: 10.1109/INTECH.2016.7845139.

[47] P. Prajapati, P. Shah, A. Ganatra, and S. Patel, "Efficient Cross User Client Side Data Deduplication in Hadoop," *J. Comput.*, vol. 12, pp. 362–370, 2017, doi: 10.17706/jcp.12.4.362-370.

[48] J. Ni, K. Zhang, Y. Yu, X. Lin, and X. S. Shen, "Providing Task Allocation and Secure Deduplication for Mobile Crowdsensing via Fog Computing," *IEEE Trans. Dependable Secur. Comput.*, vol. 17, no. 3, pp. 581–594, 2020, doi: 10.1109/TDSC.2018.2791432.

[49] H. Yuan, X. Chen, J. Wang, J. Yuan, H. Yan, and W. Susilo, "Blockchain-based public auditing and secure deduplication with fair arbitration," *Inf. Sci. (Ny).*, vol. 541, pp. 409–425, 2020, doi: https://doi.org/10.1016/j.ins.2020.07.005.

[50] M. B., P. K. K, R. S., and S. E., "Review on Data Deduplication Techniques in Cloud.," *Embed. Syst. Artif. Intell. Adv. Intell. Syst. Comput.*, vol. 1076, 2020, doi: https://doi.org/10.1007/978-981-15-0947-6_78.

[51] J. Yin, Y. Tang, S. Deng, Z. Bangpeng, and A. Zomaya, "MUSE: A Multi-tierd and SLA-Driven Deduplication Framework for Cloud Storage Systems," *IEEE Trans. Comput.*, p. 1, 2020, doi: 10.1109/TC.2020.2996638.

[52] S. K. Nayak and S. Tripathy, "SEDS: secure and efficient server-aided data deduplication scheme for cloud storage," *Int. J. Inf. Secur.*, vol. 19, no. 2, pp. 229–240, 2020, doi: 10.1007/s10207-019-00455-w.

[53] X. Liu, T. Lu, X. He, X. Yang, and S. Niu, "Verifiable Attribute-Based Keyword Search Over Encrypted Cloud Data Supporting Data Deduplication," *IEEE Access*, vol. 8, pp. 52062–52074, 2020, doi: 10.1109/ACCESS.2020.2980627.

[54] J. Li, Z. Yang, Y. Ren, P. P. C. Lee, and X. Zhang, "Balancing Storage Efficiency and Data Confidentiality with Tunable Encrypted Deduplication," 2020, doi: 10.1145/3342195.3387531.

[55] J. Li, P. P. C. Lee, C. Tan, C. Qin, and X. Zhang, "Information Leakage in Encrypted Deduplication via Frequency Analysis: Attacks and Defenses," *ACM Trans. Storage*, vol. 16, no. 1, 2020, doi: 10.1145/3365840.

[56] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, "A Secure Data Deduplication Scheme for Cloud Storage," 2014, vol. 8437, pp. 99–118, doi: 10.1007/978-3-662-45472-5_8.

[57] J. Bai, J. Yu, and X. Gao, "Secure auditing and deduplication for encrypted cloud data supporting ownership modification," *Soft Comput.*, vol. 24, pp. 12197–12214, 2020.

[58] C. Wang, Z. Qin, J. Peng, and J. Wang, "A novel encryption scheme for data deduplication system," in *2010 International Conference on Communications, Circuits and Systems (ICCCAS)*, 2010, pp. 265–269, doi: 10.1109/ICCCAS.2010.5581996.

[59] J.-S. Li, I.-H. Liu, C.-Y. Lee, C.-F. Li, and C.-G. Liu, "A Novel Data Deduplication Scheme for Encrypted Cloud Databases," *J. Internet Technol.*, vol. 21, no. 4, pp. 1115--1125, 2020.

[60] Z. Rasjid, B. Soewito, G. Witjaksono, and E. Abdurachman, "A review of collisions in cryptographic hash function used in digital forensic tools," *Procedia Comput. Sci.*, vol. 116, pp. 381–392, 2017, doi: 10.1016/j.procs.2017.10.072.

[61] Y. Zhang, C. Xu, N. Cheng, and X. Shen, "Secure Encrypted Data Deduplication for Cloud Storage against Compromised Key Servers," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6, doi: 10.1109/GLOBECOM38437.2019.9013792.

[62] D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side Channels in Cloud Services: Deduplication in Cloud Storage," *Secur. Privacy, IEEE*, vol. 8, pp. 40–47, 2011, doi: 10.1109/MSP.2010.187.

[63] P. Anderson and L. Zhang, "Fast and secure laptop backups with encrypted de-duplication," *Proc. 24th Int. Conf. Large Install. Syst. Adm.*, 2010.

[64] J. R. Douceur, A. Adya, W. J. Bolosky, P. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," in *Proceedings 22nd International Conference on Distributed Computing Systems*, 2002, pp. 617–624, doi: 10.1109/ICDCS.2002.1022312.

[65] J. Liu, N. Asokan, and B. Pinkas, "Secure Deduplication of Encrypted Data without Additional Independent Servers," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 874–885, doi: 10.1145/2810103.2813623.

[66] H. CUI, Z. WAN, R. H. DENG, G. WANG, and Y. LI., "Efficient and expressive keyword search over encrypted data in the cloud.," *IEEE Trans. Dependable Secur. Comput.*, vol. 15, no. 3, pp. 409–422, 2018.

[67] J. Li, X. Lin, Y. Zhang, and J. Han, "KSF-OABE: Outsourced Attribute-Based Encryption with Keyword Search Function for Cloud Storage," *IEEE Trans. Serv. Comput.*, vol. 10, no. 5, pp. 715–725, 2017, doi: 10.1109/TSC.2016.2542813.

[68] W. Guo *et al.*, "Outsourced dynamic provable data possession with batch update for secure cloud storage," *Futur. Gener. Comput. Syst.*, vol. 95, pp. 309–322, 2019, doi: https://doi.org/10.1016/j.future.2019.01.009.