JDEBM
Journal of Digital Engineering and Business Managemet

Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/44
DOI: https://doi.org/10.5281/zenodo.20717802

ENCRYPTED METADATA MANAGEMENT WITH DUPLICATION CONTROL FOR CLOUD STORAGE OPTIMIZATION

Dr. B. Srinivasa Rao, *Assistant Professor,Department of CSE (Data Science)*,
Gurunanak Institutions Technical Campus,Ibrahimpatnam, Hyderabad.

ABSTRACT: Cloud service providers allow users store and move data quickly and easily. Cloud service providers are using data de-duplication tools to conserve bandwidth and cut storage expenses. Customers want to use the cloud safely and privately so that their data is safe. So, they encrypt the data before sending it to the cloud. The encryption aim and the de-duplication function don't work well together, making it hard to employ the data de-duplication functionality. The current methods for de-duplication don't work and aren't safe from a security or efficiency point of view. They either cost a lot of processing power or are easy to break into, which lets the attacker get files back. This is what drives us to come up with a safe and effective way to get rid of extra data. Before we talk about the literature, which talks about a lot of different ways to de-duplicate data and the security and efficiency issues that current systems have, we will first explain how de-duplication techniques work and how they are used. Using hashing techniques and the AES-CBC algorithm, our suggested invention makes data de-duplication safer and faster for users. Users' keys are always made safely and consistently, without the help of anyone else. We prove how useful it is by putting the suggested strategy into practice and comparing it to current ways.

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

1. INTRODUCTION

Technology has advanced significantly over the past decade, providing individuals and companies with a plethora of new options. Businesses have discovered a plethora of fantastic ways to enhance operations, increase profits, and get more done thanks to recent technological advancements. There is no pertinent information in the user's message. Businesses are now able to complete transactions and operations with more efficiency because to technological improvements. Regardless, data protection remains the top priority for system security for organizations of all size.

The possibility that database duplication can lead to mistakes in company information networks is one of the main concerns. Regular data collection makes it more difficult for users to locate the correct record.

The data stored on the server can contain a wide variety of file formats, which could increase the storage requirements and potentially compromise data security. Nevertheless, the data's consistency could be compromised if duplicated. Due to having duplicate data in many locations, some insurance and financial companies have had storage and security issues. Using highly effective strategies to improve the performance of cloud storage systems is essential for modern civilization's needs.

A well-known method to increase storage efficiency, optimize bandwidth utilization, and decrease costs is data deduplication,

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/44
DOI: https://doi.org/10.5281/zenodo.20717802

which entails removing duplicate files .Take a look at and as an example. Data reduction using de-duplication is illustrated in Figure 1. The terms "intelligent compression" and "single-instance storage" can describe this technique.

The issue of data duplication is presently causing significant difficulties for some multinational corporations. Individuals, SMEs, and multinational corporations all face significant storage and security challenges when it comes to data replication. It may be challenging for experts to maintain and locate records. According to the comparison results in reference [9], data redundancy impacts a system's time and cost as shown in Figure 2. Here we can see how the amount of time and money needed to maintain the system operational is impacted by redundant information.Backup activity decreased significantly when data deduplication was implemented.

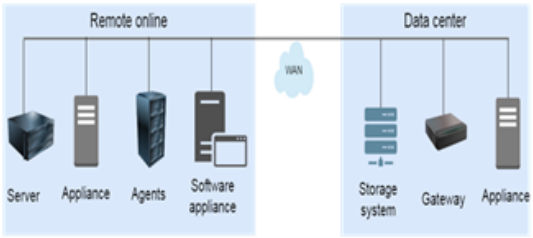

Figure 1: Where de-duplication can applied.

Duplicate data storage poses serious risks to data security, system performance, and data integrity, among other data management issues. The main cause of this is data duplication, which increases complexity, decreases efficiency, and poses a security risk. In this scenario, it complicates things even though it's trivial to remove unencrypted duplicate data .

The security of your data stored on your computer or in the cloud could be compromised if you copy it, as this practice can lead to numerous cybersecurity issues. From a cyber security perspective, encrypted data stored in the cloud is essential (11). It becomes more challenging to remove duplicate data when encrypted data is stored in the cloud. Dedu isn't compatible with encrypted files, even though it can delete duplicate data well. Because they are incompatible with data deduplication techniques, current encryption methods make it difficult to secure data stored in the cloud. The fact that numerous cipher texts may include duplicate data for several users makes de-duplication even more challenging. Without providing any background or explanation, the user only provided a reference number.

Data owners should securely encrypt their data before transferring it to a cloud storage system. Exactly how a cloud service determines whether a given string of text has numerous encrypted letters is an intriguing question. Working with a trustworthy third party was recommended by the researchers as a means to remove extraneous encrypted data. The difficulty in obtaining an entirely reliable third party is the main reason why this method is inadequate. While convergent encryption is more vulnerable to brute force attacks, it is nevertheless a helpful method for dealing with the encryption problem and removing unnecessary data. Because it combines encryption with de-duplication, encrypted de-duplication makes data storage more secure and dependable.

The aforementioned objectives, however, are fundamentally incompatible with one another. We are currently developing a new method to de-duplicate encrypted data in order to address this issue. The focus of this piece is convergent encryption, a method that can securely decrypt data

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/44
DOI: https://doi.org/10.5281/zenodo.20717802

without going into debt. For a more reliable integration, we'll employ AES-CBC encryption. Because it is both inexpensive and resource-efficient, the AES method is also considered a novel approach of eliminating duplication. In addition to accomplishing its primary goal of de-duplication, the strategy enhances performance. To make the encryption key less predictable, the method will employ the hash derivation approach plus a lengthy random string salt.

Even if the criminal succeeds in transmitting the file's hash, they will still be unable to supply the additional salt. If you need more room on the server, you can decrypt the file and delete any duplicates. The signature of the encrypted file can be used as the label value for this purpose. By enhancing present security measures and enabling quick and easy access to critical information, our research primarily aims to improve the quality of life for data consumers. To determine how effective the methods and strategies used to accomplish this goal are, this research reviews and synthesizes all available information. Given that the present approaches are not without their flaws, a practical remedy is proposed. A suite of algorithms that enable you to remove superfluous data while simultaneously enhancing data security is the basis of the proposed solution.

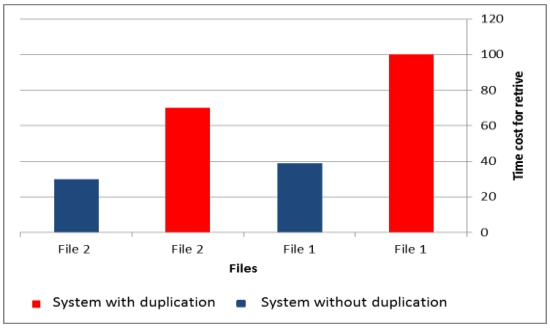

Figure 2: A Comparison results of data with de-duplication

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.

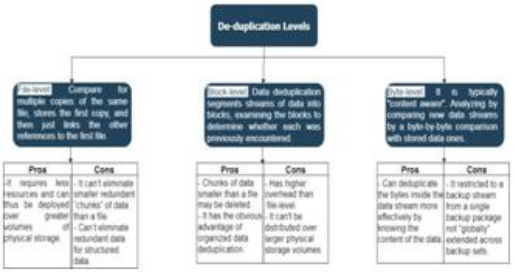

Figure 3: De-duplication Levels.

File-level de-duplication

File-level de-duplication is a data reduction technique that operates at the file level, as its name implies. The storage in issue is commonly referred to as single instance storage (SIS). A unique identification number is assigned to each file, which is determined by its unique attributes. The identifier facilitates the exclusion of files that share similarities but have distinct names by preserving the entire dataset associated with the file.

Block -level de-duplication

The block-level data de-duplication technology involves the division of a data stream into blocks. Subsequently, each block is scrutinized to ascertain whether the data is consistent with that of previous blocks. This is frequently achieved by employing a hash algorithm to generate a digital signature or unique identifier for

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/44
DOI: https://doi.org/10.5281/zenodo.20717802

each data block . Furthermore, the block's identifier is included in the index if it has been successfully stored on the disk and possesses unique properties. The only deposit indicator that maintains the initial location of the same data block is if it is absent.

Byte -level de-duplication

The fact that it uses content information to differentiate files from other data formats is what makes bytes-level de-duplication a type of file-level de-duplication. Deduplication at the byte level is one subset of deduplication at the block level.the data de-duplication approach that considers the data's semantics or substance. These systems are sometimes referred to as content-aware systems (CAS). Data deduplication is an additional step that follows byte stream processing as an alternative method.

De-duplication Applications

Data de-duplication is a way to cut down on unnecessary or excessive data. It organises the data into files or blocks according to predetermined criteria, finds segments that are duplicated, and stores a little reference to the duplicated bits rather than the original information.

The goal of most conventional data compression algorithms is to reduce file sizes by eliminating unnecessary data duplication. When data is present in many files, de-duplication can be applied to eliminate duplicate entries using various data processing or indexing procedures. Archive backup systems benefit greatly from de-duplication as it maximizes storage capacity utilization compared to conventional data compression approaches.

Backup storage De-duplication

Applying de-duplication in cyclical complete backup operations is beneficial for backup systems that mostly contain data that is widely copied. In order to find duplicate data blocks, the de-duplication process looks at the source data, pulls one copy of a data block from storage, and then uses a block index to find other copies. Users can enhance the storage capacity of their existing devices by drastically decreasing the space needed for the source data using this strategy. Increasing the amount of data kept on individual machines allows backups to last longer. Data center de-duplication reduces the need for storage devices, which improves operational and budgetary efficiency.

Cloud storage De-duplication

Users can access and recover data from several platforms whenever it's convenient for them thanks to cloud storage, a type of offsite data storage. Since the amount of data required can be changed to accommodate different demand levels, consumers may be able to save money by controlling it well.

Customers frequently hold similar data spread across multiple accounts in modern cloud storage systems. As a result, de-duplication becomes essential to achieving cost savings. De-duplication reduces the amount of network bandwidth and storage space used by allowing a cloud storage provider to efficiently manage and keep only one physical copy of identical data in a designated location.

Virtual machines storage De-duplication

Data duplication is a common problem for host file systems and primary storage systems. Similar operating systems have probably been preinstalled on many electronic gadgets. A number of sources state that virtual machines can share configurations, programs, and storage device settings with one another .

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/44
DOI: https://doi.org/10.5281/zenodo.20717802

2. LITERATURE REVIEW

An all-encompassing review of data de-duplication, including its core idea and particular approaches, is the goal of this section. Readers will gain a modern understanding of the topic as a whole.

Data De-duplication

Efficacious compression methods are used to lessen the quantity of duplicate data and successfully remove unnecessary information. In addition, it has seen heavy use in cloud storage to reduce bandwidth consumption and storage capacity requirements. According to the results of the research, the procedure for locating duplicate data or documents also takes the permission levels of the users into account.

The value of data is well recognized by businesses. Businesses will rely on continuous and dependable big data to help them make informed decisions. Using high-quality data and an effective approach to reduce duplicate data are crucial for drawing solid findings, according to the research.

Data security and integrity have become major issues due to the prevalence of duplicate records in cloud-based storage systems. One of the trickiest aspects of cloud computing is dealing with data duplication. A strategy to reduce data duplication has been put out by the researchers. Using video and picture data, this research demonstrates the creation and implementation of tools and processes. Data duplication reduction technologies are classified into two groups in this research : those that use identical data detection approaches and those that use comparable data encoding and detection methods. A survey was used by researchers to gather data on how common data duplication concerns are in businesses. Discovering the problems caused by duplicate records and finding solutions to those problems is the main goal of this inquiry.

Researchers in the research came up with a strategy to improve security measures and get rid of duplicate data in cloud storage systems. Additionally, a safe central storage system for eradicating duplicates has been created by the researchers.

The results of this research lend credence to the idea that data duplication should be minimized on both the block and file levels. The exponential growth of data, which includes several forms like audio, video, and text files, poses many problems for storage and retrieval procedures, according to the research's results (5). A large portion of a company's budget goes into purchasing and maintaining data storage infrastructure.

As a result, a very efficient technique must be put into place for the management of giant data sets. Since the researchers are now aware of the difficulties connected with these processes, they are focusing on ways for removing redundant data. Additionally, problems emerge when available bandwidth is limited or diminished, which lowers the data's value. A thorough survey is created to detect and evaluate any possible risks and weaknesses, in addition to ranking optimization tactics.

Implementing a method to eliminate duplicate data is highly recommended for risk mitigation. Copies of data stored in the cloud quickly exhaust available storage space. Several approaches to creating a system that successfully decreases levels of cloud storage redundancy are detailed in this research. Actually collected data also

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/44
DOI: https://doi.org/10.5281/zenodo.20717802

shows that this method has cut down on duplicate data in the cloud by about 20-30% . It is critical to optimize the removal of duplicate data since it has a major impact on the outcome of a procedure. A biometric method and an enhanced version of Rabin's algorithm were proposed by the researchers. The problem of duplication will be efficiently handled by fully implementing the procedure.

Each sector of the cloud storage system is given a unique block number that corresponds to the data entries, and this system is used to store the data file that is part of this framework. According to the research, the suggested solution outperforms competing approaches when it comes to eliminating data duplicates.

A research report referenced in reference indicates that cloud computing is a technological solution that is both highly effective and extremely valuable. A wide variety of data storage choices are provided. Data saved in the cloud may be accessed from any place, which is quite convenient. Using expensive gear, software, and specialized facilities further lengthens the time it takes to access replicated data. The difficult and complicated task of meeting growing storage demands also necessitates improvements to the organization's infrastructure. Many problems can arise for businesses as a result of duplicate papers. After applying the checksum algorithm, the problem was fixed.

As network technology improve at a dizzying rate, data center volumes are also growing at a dizzying rate. When looking for a storage solution, many companies are considering "green stores," or storage choices that are good for the environment. Optimizing storage systems and reducing the requirement for substantial data storage are both possible outcomes of data duplication reduction technology. By decreasing the number of drives needed for a given operation, data compression techniques can help lower the expenses associated with disk energy use. The data duplication elimination technique, including all relevant phases and implementation, will form the foundation of this inquiry.

In order to get rid of unnecessary details, the research used the duplicate and multiple representations technique. Subsequently, the team revealed their two new data duplication methods, which would improve the system's performance by identifying duplicate information. Within the allotted period, the algorithms may considerably improve the process's overall efficiency. By applying these algorithms to data that had already been collected and saved in a repository, the researchers not only found unanticipated results, but also came to the conclusion that the procedure is rapid.

Nearly 80% of firms used deduplication technologies to reduce storage system redundancy, according to a survey by the IDC analytics group . Research in this area mainly aimed to improve storage efficiency and decision-making by creating efficient data deduplication methods. The goals were to improve the data's quality and efficacy while reducing the frequency range. There is no denying the need of preserving storage efficacy. However, a faster method that optimizes and balances storage efficiency and security must be developed when a more comprehensive assessment of cloud computing storage security is undertaken. As a result, we must recognize that data privacy and

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/44
DOI: https://doi.org/10.5281/zenodo.20717802

security are major issues with data duplication technologies, and we must keep trying to find solutions to these problems.

Considering that data owners may choose to transform external multimedia information into more secure forms, this research suggests a way to resolve the conflict between the watermarking technique and the deletion of duplicate data [40]. It may be difficult to delete duplicate data due to the fact that it can be transmitted in multiple formats and by different data owners.

The suggested watermarking method is all-inclusive, and it doesn't rely on a third party or require data owners to connect with one another. The solution is improved with the implementation of a protocol that ensures all users are supplied with the same data in a standard and safe fashion. Using a watermark is one way to stop copying.

The article cited as discusses index management for techniques that reduce duplicate data. In order to maximize storage and memory efficiency and reduce input/output procedures, the mechanism for erasing duplicate data is activated when this happens. As a flash caching strategy, this inquiry suggests using Austerecache to optimize memory indexing. In addition, this idea streamlines crucial activities for cache management, constructs appropriate data structures, and removes a large quantity of data for indexing. Research has shown that this method can significantly reduce memory requirements by 97% while keeping the reduction ratios for reading and writing the same.

In relational databases, data de-duplication is the process of finding duplicate tuples. Since a similarity value must be determined for each pair of tuples, the compute cost of the operation is always quadratic with respect to the number of tuples. By separating the tuples into separate blocks, blocking approaches limit the comparison to tuples within the same block and avoid comparing tuple pairings that are clearly not duplicates.

Data de-duplication for big datasets is still a costly ordeal, even with blocking mechanisms in place. The purpose of this research is to demonstrate how to use parallelism in a shared-nothing computing environment to improve data de-duplication efficiency, as described in reference . An solution that reduces the burden on worker nodes and provides strong theoretical guarantees is the Dis-Dedup delivery mechanism.

The Bloom filter is used in this inquiry to remove any unnecessary information (Smith, 2019). There are essentially three parts to the suggested approach. Removing a duplicate was one possible option. The allowed requests are overseen by the Administration Center.

The next step is to build a radix trie to show how roles and alternatives are related. When measuring efficiency and refreshing data, a BLOM filter is your best bet. The model has a considerable repetitive data cancellation rate of up to 25% and efficiently computes the ciphertexts, according to the results of the simulation tests. The content of the message might be deduced by the adversary if one of the generated ciphertexts matches the target message's ciphertext.

Single-Server Cross-User De-duplication

Implementing client-side encryption is made possible by this system. Through the

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/44
DOI: https://doi.org/10.5281/zenodo.20717802

PAKE protocol, one client can borrow an encryption key from another client that has already sent the same data. It is advised to use a randomly generated encryption key to guarantee data security during the initial file transmission. But this approach opens the door to a side-channel attack, which lets users figure out that a particular file isn't there in the cloud storage. It is still possible to alter a session key before sending it to the cloud storage provider so that it has a fake value. With the goal of preserving bandwidth efficiency, cloud storage providers provide client-side encryption and single-server cross-user de-duplication algorithms to prevent server-side assaults. These attacks aim to determine which users submitted a certain item.

Symmetric or asymmetric encryption

Prior to uploading the data to the CSP, it is encrypted using a symmetric or asymmetric encryption mechanism. The data is encrypted in accordance with the ciphertext policy before being stored in the CSP. The encryption keys are generated by the data owner.

The cryptographic keys are not suitable for the Cloud Service Provider (CSP) to acquire. Because the CSP's information sharing gives no clue to a hostile user about the existence of data storage, this technique effectively eliminates online brute-force assaults.

The suggested method employs bilinear mappings and features to improve processing performance during encryption and decryption . Therefore, cutting down on computing expenses becomes the main goal . This issue is solved by relying on a decryption mechanism that is provided by an outside party, which is not recommended in real-world scenarios.

3. PROPOSED APPROACH

Specification requirements

The suggested approach is expected to meet the following capabilities requirements:

Efficiency: The suggested approach efficiently minimizes data duplication, allowing for better utilization of disc space, cost savings, and less network traffic. Anyone with permission to access the data can efficiently and effectively handle data stored in the cloud. The provider of cloud services can effortlessly and rapidly handle encrypted data duplication.

Security: Everyone concerned can rest easy with the proposed approach. The elimination of redundant data is possible for the service provider without jeopardizing the privacy of the plain text. To further ensure the security of transferred files to the cloud storage platform, the data controller employs encryption. An advantage for the data user is that it is secure from unauthorized access; only authorized users can view the data in its unencrypted form.

Computation: Opting for a compact and protected storage solution can help keep computing costs and maintenance to a minimum. The third-party administrator must also grant you authorization.

Auditing: For the purpose of ensuring that the user's validation procedure is unsuccessful due to the fact that the user's data is lost by the cloud.

Data owner: Once data is uploaded to the cloud, users can utilize cloud-based services to ensure the security of encrypted files. If you have permission to access the cloud, you are responsible for handling the decrypted file.

Data users: Now that they've made the

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/44
DOI: https://doi.org/10.5281/zenodo.20717802

file upload feature, the creator is curious if they can retrieve files saved in the cloud. If the data owner has granted them access to see the file, it means their user key is compliant with their policy. Users have the option to utilize a cloud storage auditing tool to verify the correctness of their cloud data.

Cloud service provider: Controls the storage and administration of encrypted files in the cloud. A person or organization also has an obligation to make appropriate use of any information to which they have access. Following the generation of data authenticators, we verify that the data hasn't been altered by checking the veracity of the search results. in addition to the requirement to eliminate duplicate data from files.

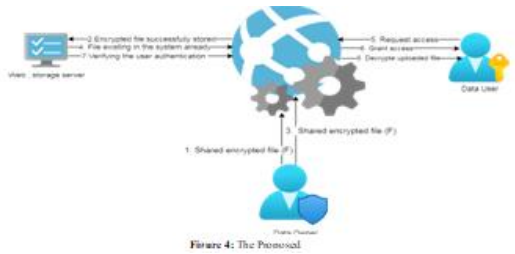

Figure 4 illustrates the presence of three entities, namely the cloud server provider, the data owner, and the cloud server user.

Security model

Adhere to these strategies to eliminate superfluous data from the cloud while simultaneously safeguarding critical information.

Setup algorithms(S): The system's setup is governed by setup algorithms (S). Currently, the safety setting is not enabled.

Key Generation (Key Gen): The data, along with a powerful string salt, must be used to generate an MD5 hash in order to obtain the key. The AES approach can also be used with an intravenous line. A trustworthy cryptographic key, finalized, will be the result.

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

Integrity Check(i): The security measures implemented by the cloud service guarantee the confidentiality and integrity of all sent data.

Access structure schema (AS): The permission structure is connected to the encrypted text. Complementing one another are the hidden and distinctive attributes. To rephrase, decryption of data can only take place if the attributes of the set of attributes are compatible with the access structure."Decrypt" really means "crack a code" here. The input consists of the shared key (K) and the ciphertext (CT). In the end, all text output will be in plain text (PT) format.

Dedup Check (Dedup): The data label for data duplication is created using the hash value of the encrypted data. The output, when applied to encrypted data, can be either 0 (storage) or 1 (de-duplication-induced destruction).

```
Setup algorithm:

Key generation:
Input: Security parameterOutput: shared key(K)
1- seed(time(0))
2- for i ∈ [1..32]
Calculate Xi=MD5(MD5(PT)+Salt)) , (32,127) //printabl
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
```

Figure 5: Security model.

The server configures the AES cipher parameters when the program is being set up and stores them in a file that can be retrieved from any location. No amount of damage can ever reach the root. Whatever the case may be, the server-side random number generator will always begin with the current timestamp, which is an integer.

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/44
DOI: https://doi.org/10.5281/zenodo.20717802

A random string of 32 bits for the key and 16 bits for the IV is then generated.

Use the web interface to access the shared F file. Data encryption is the responsibility of the user. The AES CBC algorithm is used, and the key length is 256 bits. The server's AES cipher settings are stored in a file on the file system that can only be accessed by root during setup. In order to access the secret, a suid-wrapped executable file will be created. To initiate it, we will make use of Python's subprocess module. The Popen algorithm, which is defined as F + Key + IV + E = C, will be used by the server. The computer will then decipher the Sig, or signature, of the ciphertext. You can use the md5 and sha512 algorithms to get the formula Sig = S1(C) + S2(C).The newly generated signature will be used in a secure SELECT query to search the database for a matching entry.

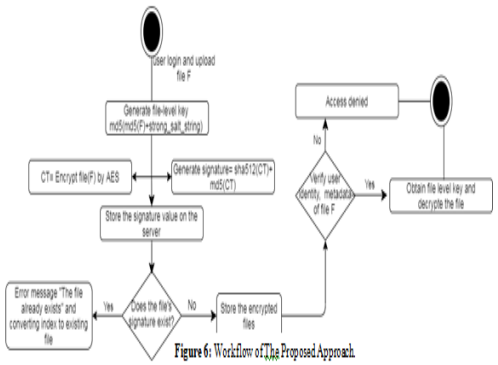

Figure 6: Workflow of The Proposed Approach

4. SECURITY ANALYSIS

Encryption values in De-duplication

The first part of this section gives real-world proof that our method for encrypting and decrypting data works. Before being uploaded to the cloud, the initial dataset is encrypted with an asymmetric encryption method, like AES-256. When a data owner uploads files, encryption keys are made and encrypted. After that, the cloud safely stores these keys. It is easier to trust important data when hash methods are used to check the integrity of both the plaintext and the symmetric key. Moreover, unlike most other methods, our method only uses encryption and decryption to verify user identity and data integrity. The Consumer Safety Program (CSP) is in charge of making product labels easier to read and producing complete label indexes. The cloud infrastructure will employ a label index [10] to find out if users have shared files that are the same.

Security using AES with salted and MD5 as key

The AES algorithm key is constructed by a crucial generation function using data extracted from files stored in the cloud. Such an approach is often called "data label-based encryption." To create an encryption key that is compliant with the Advanced Encryption Standard (AES), a key derivation technique is used. The input file needs to contain a salt value and an MD5 hash in order for this to work. Obtain the files' MD5 hash, make a salt out of a random string, then figure out the right encryption key. After that, the data is encrypted in the correct block cipher mode using the key and AES. All that is kept are the encrypted data, the salt, and the initialization vector (IV) if any is required for the cipher mode. To gain access to the files, you must first determine the key that was used to encrypt them.

The main goal of the salt is to make precomputation upgrades useless, which will make dictionary and brute force assaults less effective. After finding the right salt, a brute force dictionary assault can be launched. The encryption keys are protected by the dynamic environment, on the other hand. This ensures that no one, not even the service provider, is aware of

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/44
DOI: https://doi.org/10.5281/zenodo.20717802

them, making them impossible to track.

The protected files that were stored with the data cannot be labelled by the service provider. Generated using encrypted information, inferring this data label is no longer a practical option. This renders the previously used brute force attack against convergent encryption completely ineffective.

Hash Values in De-duplication

Our approach makes use of the hash number to detect file duplication. This hash value is calculated using a cryptographic method for the encrypted file that is being saved. To find discrepancies, we compare the original hash value to the updated hash value after the modification. We will remove files that have the same hash value. In case of inconsistency, only the revised part is sent to the storage medium.

People frequently assume that hashes are all the same. Every time you edit a file, the hash value changes. This makes it possible for service providers to apply changes. Every one of the hashes has been properly archived and catalogued. When you edit a file, it goes through a mathematical hashing process and then it's compared to the original. By comparing the hashes, we may find duplicates and remove them together with their hashes.

Data restoration, optimizing bandwidth consumption, time conservation, and related issue solving are all made easier with versioning approaches. There will be fewer problems because all it takes to achieve this is to send the revised file over the internet. Because more space is being used, the de-duplication ratio is better overall. In order to determine the file number, our system employs two separate hash functions. Maximizing speed and security are the goals of this approach. Both types of hashes involve keeping an account that is aggregated. In brute force attacks, an attacker tries to obtain secret data by abusing files submitted by authorized users; this method shows that our system is effective in protecting against such attacks.

5. PERFORMANCE EVALUATION

Our web application now uses the suggested approach. The web application is used by the server. Python was used to create the web app. Web application development requires a 64-bit Linux PC with a 1Mbps network connection, 17GB of RAM, and a dual-core CPU. Each user-uploaded file to the server is duplicated and encrypted once using the proposed method. Because the storage system in question is a cloud storage solution without de-duplication, we compare our method to the industry-standard methodology described in reference.

The growing practice of sharing files, as shown in Figure 7, improves the efficiency of our approach and reduces storage requirements. Storage capacity changes before and after the approach's implementation, as well as changes seen without duplication, are shown in Figure 8. Processing efficiency and bandwidth utilization are both affected by the distribution of storage capacity as a result of redundant data. For the convenience of our many clients, we store roughly twenty different file kinds, which allows for real-time analysis. We measured how long each step of our process took before sending out a file. Finding out how much time people typically spend on expenses is crucial.

Figure 9 shows the results of the experiment. To encrypt, generate keys, and

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/44
DOI: https://doi.org/10.5281/zenodo.20717802

decode on the client side, very little processing power is needed. Compared to other studies that used bilinear over encryption and third-party entities, our method is less complex and easier to understand. When a third party is involved, the task takes longer to complete. This is how "overhead" is defined when talking about planning.

The term for this is "overhead" in planning.

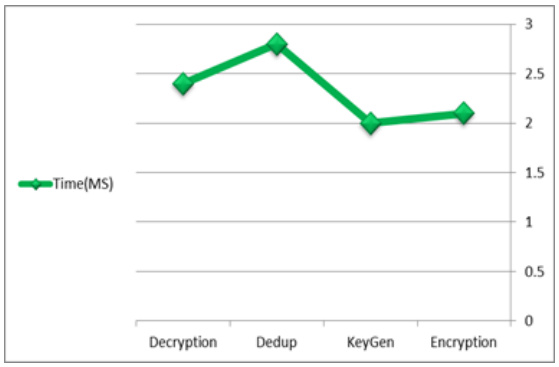

Figure 7: Execution Time of Different Operations.

6. CONCLUSION

The necessity to investigate and resolve issues like de-duplication emerges in response to the increasing demand for data storage. In this research, we rely on academic literature that discuss several methods for erasing duplicate data. Much has been made of the de-duplication tool's security flaws and other difficulties. A novel and widely-recognized method for efficiently combining deduplication with safe encryption is presented in this research. Going this route is prudent and risk-free. Anyone with access to the same data can, directly from the Internet, obtain the same encryption key, eliminating the need for a middleman. The key is created using the file's hash. The next step is to add a random salt to the key, which will make it harder to predict and defend it from attacks. These keys are also not used by the service because they are composed of variables. An easy technique to save space and transmission speeds on the cloud server is to check the data mark numbers of each file to make sure there are no duplicates. This value is concealed by the ciphertext policy attribute to prevent data leakage. If you need to encrypt sensitive data so no one can access it, you should utilize the AES-CBC approach. Compared to other forms of self-defense, it is really simple. This strategy's efficacy and safety were assessed using the realistic proposal method. In order to remove unnecessary data while maintaining a good security profile, the results show how well the proposed strategy works.

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,

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/44
DOI: https://doi.org/10.5281/zenodo.20717802

doi: 10.1007/s12083- 020-00940-3.

[5] E. Manogar and S. Abirami, "A research 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.