Convergent Encryption Enabled Secure Data Deduplication Algorithm for Cloud Environment

Shahnawaz Ahmad
Jamia Millia Islamia (A Central University)
Shabana Mehfuz (smehfuz@jmi.ac.in)
Jamia Millia Islamia (A Central University)
Iman Shakeel
Jamia Millia Islamia (A Central University)

Research Article

Keywords: Cloud Computing, Deduplication, Approaches, servers, UML diagrams, convergent encryption algorithm

Posted Date: December 9th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2347062/v1

Abstract

The exponential growth of data management nowadays is quite a tedious and critical issue. It is also evident that methods employed for collecting data for cloud storage exert additional load on different cloud servers operated by many enterprises. Various approaches are used these days to reduce the burden on computer servers. One such approach is de-duplication, which has gained much attention due to its efficient, extensive storage system. In this approach, redundant data is removed, which improves storage utilization and reduces the cost of secure storage. International Data Corporation (IDC) reported 33 Zettabytes in 2018 to 175 ZB by 2025, putting cumbersome loads on present servers. Due to this enormous amount of data, it is challenging for the local and small servers, usually used in various enterprises, to handle it. It has also been observed that most data are generally duplicated in terms of space; therefore, data transmission places extra effort on small servers. This study provides a more comprehensive analysis of the literature on safe data duplication. Furthermore, it classifies the various secure data storage techniques applied at different levels of encrypted data collecting storage.

Furthermore, this article looks into the classification of the de-duplication procedures as per literature and other Unified Modeling Language (UML) activity diagrams, exhibiting both their classification and detection difficulties. Moreover, current duplication techniques suffer from a couple of security challenges. Therefore, a convergent encryption algorithm has been proposed and implemented along with the de-duplication techniques, and the different UML diagrams and comparative analysis have illustrated the proposal's viability.

1. Introduction

Web clients have expanded significantly in recent times. Individuals and organizations are increasingly switching to online modes of working. It has become a crucial part of their way of life and is also related to their social lives. Also, since the COVID-19 pandemic struck us, our association with the internet has changed at a fundamental level. We have been utilizing the gift of innovation to accomplish things we have never done before. Many employment sectors, industries, and teaching professionals benefit from innovation to simplify work in various domains. The advantages witnessed are here to stay, but such developments render it vulnerable to many veritable security dangers. This makes cyber dangers one of the most prominent threats. In recent times, online threats have risen six times more than ever. Attackers can seize control of the users' framework and attempt to steal or manipulate their data.

Due to the significance of cloud storage and its process models, the cloud user base is growing every day. Besides its advantages of sharing knowledge and reducing the storage price and alternative resources, it is prone to security threats while accessing the information. The time and effort involved in sifting through vast amounts of explicit data is a new drawback of cloud services. The resources must be used carefully to save data. Various strategies like Proof of ownership, Multiple Line Encryption (MLE), homomorphism encryption, and oblivious pseudo-random function are used in the de-duplication process (Huynh Thu et al. 2008). All these techniques have been implemented in block-level or file-level de-

duplication. Secret sharing protocols and the HMAC SHA algorithm have also been implemented to increase storage capacity and decrease upload bandwidth. To detect duplication in cloud storage, the convergent key management scheme was created (Umberto Martinez-Penas, 2018).

Using cryptography, the distributed convergent key provides secure and reliable networking (Ade Monika et al. 2016). Various analyses and research are carried out to enhance data protection and retrieve information from the cloud. In general, encrypting the data before storing it on the server increases data security in the cloud, and conjointly, cryptography occurs during information retrieval. However, cryptography alone cannot guarantee the server's data preservation needs. When obtaining encrypted cloud data, information leakage is possible. As a result, numerous types of research have been carried out to maintain the confidentiality of encrypted cloud data.

There have been colossal development in the utilization of "computers," "information," "online applications," and "versatile computing" recently. As a result, the ever-growing information and capacity space required to store that information has become a prime concern.

Recent developments in cloud computing have led to several reports of vendor lock-in solutions. Smartphone, multimedia, and social networking platform usage on the internet is rising, which has significantly increased the amount of data stored in clouds. This also encourages enterprises to embrace cloud-based solutions. Although cloud computing-based services (IaaS, SaaS, and PaaS) are incredibly affordable these days, they have a few drawbacks like "reliability," "security," "accessibility," and "confidentiality" of the data that is being shared to the CSPs (Ali et al. 2015), (Singh, P et al. 2017). Global statistics organization (IDC-International Data Corporation) reported 33 ZB by 2018 to 175 ZB by 2025. Size information may even be created and transferred globally. The utilization of cloud capacity of different cloud computing applications can be visualized in Fig. 1 and Fig. 2. Due to the limitless possibilities open to them, data de-duplication is currently attracting considerable attention as everyone has become dependent on cloud services.

Data de-duplication removes duplicate copies of the data from the servers, leaving only one distinct copy. However, encryption seeks to randomize content so that an adversary cannot read it, whereas de-duplication aims to detect similarities in data. Combining the two principles is a complex undertaking. A convergent key is used for secured analysis in the encryption and decryption mechanisms for duplicate-less key encryption and key retrieval. It has been suggested to manage, store, and safeguard cloud data from unreliable buyers using a dekey-protected de-duplication system. The encryption key is acquired from the content itself in this method. One of the disadvantages of a convergent encryption algorithm is that data becomes subject to security breaches if the key is compromised. As a result, key management is one of the critical factors that must be addressed. These problems prompted a search for solutions that combine reliability, accessibility, confidentiality, integrity, and effective key management into a single framework. An extensive understanding of the state-of-the-art and a novel taxonomy to define CASBs are the results of the survey Ahmad S et al. 2022 suggested, which uses a systematic examination of the literature to uncover and categories strategies for achieving CASB.

The following are the article's primary contributions:

1. In addition, a thorough comparative analysis of various CSPs concerning the security controls incorporated has been taken up.
2. Various data de-duplication techniques have been compared based on their different levels of security.
3. A secure convergent encryption algorithm-based block-level de-duplication technique has been proposed and implemented, enhancing the entire process's security and catering to repeated cycle problems. Finally, we discuss the future direction of this research field.

This research work references illustrious journals, court cases from various gatherings, and data from multiple research offices. The manuscript has been divided into six sections. Cloud Computing based data de-duplication security is described in section 2. Section 3 portrays the foundation, development of de-duplication, various methods used for de-duplication, and their advantages. Section 4 presents details of secure data de-duplication approaches, considering all the latest proposals in this area. A detailed presentation of their features has been provided in this section. In section 5, convergent encryption for secure data de-duplication has been proposed. This section also contains the experiments and analysis of the proposed scheme, along with the security analysis and comparison with other counterparts. Finally, section 6 ends the article with concluding remarks and provides scope for potential future research.

2. Cloud Computing-based Data De-duplication Security: A Conundrum

Cloud Computing facilitates remote access to available resources by controlling and providing software and hardware resources. Cloud Computing is an independent platform. There is no requirement to install any software on the Personal Computer (PC) to access the resources remotely. "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." This definition has been given by Mell and Grance (2011). The use of the cloud and its growth in recent years has been a major benefit and convenience for society as cloud computing helps to lessen the need for resources for data storage. It is simple to access the data stored in the cloud from any location at any time. IT resources, including a physical server, virtual server, software, service, storage device, and network device, are all included in the cloud. The term "cloud service" refers to the software API used by remote clients (such as medical and paramedical professionals) to access resources.

There are merits to cloud computing for all types of end users. For instance, a company could share a sensitive file for business objectives, or a regular person could save their information there. A de-duplication mechanism has been devised for significant data utilization in the cloud context (Zheng Yan et al. 2016). The number of users of cloud services is increasing daily due to the advantages of storage and process models. Due to the inflated number of users, the amount of data stored in the Cloud will continue to increase, and a third party may be able to access the Cloud. Despite its benefits in terms of

quick information exchange, lower storage costs, and need for more resources, it has security access issues. Finding a chunk of explicit information from the large pool is a new drawback of cloud services.

There are two primary drawbacks of cloud computing i.e., security and reliability. This is due to the ease with which any other client can access the user's data. For instance, hackers might try to access client data using valid user names and passwords, changing the data, and so forth. Security in the cloud can be ensured via several methods, including encryption, authorization, and authentication. Both users and cloud service providers expose the cloud to security risks. Some risks related to cloud security include data loss, hacking, denial of service, malicious insider assaults, and issues with shared technology. Authentication, authorization, data protection, and other security considerations are just a few things cloud service providers need to take into account. As data goes to the cloud, these fundamental security objectives are more important than ever. If a customer has confidence in the cloud service provider (CSP) and the services it provides, that confidence will likely play a significant role in whether they decide to use a cloud platform or continue with the legacy framework. Trust is reliant on the CSP's obligation to protect data from all dangers, VM security, and other legal factors. During the cloud system security review (CIA), "Confidentiality, Integrity, and Availability" are the three factors that are taken into account (R. D. Labati et al. 2020). The cost of each attack in 2019 was 3,83,365 US Dollars, as per the Cyber Security Breaches Report. It also states that 12.3 billion additional records were compromised in 2019. This is shown in Table 1.

Table 1
Data Breaches in the years 2019 and 2020 with different attacks

References	Records	Data Breaches involved in various activities (%)
Cybersecurity Breach Report: Black Hat Ethical Hacking (2019)	12.3 Billon	● "48% of businesses identified at least one attack per month." ● "62% of businesses can respond to a breach immediately." ● "67% of organizations are reported being breached at some of the points in the past systems." ● "31% of organizations experienced cyber-attack on their operation infrastructure." ● “80% attacks from phishing attacks” ● "28% come from spoofed mail attacks." ● "27% of attacks came from malware, ransomware, and spyware-related attacks."
Report on Verizon Data Breach Investigations (2020)	3,950	● "70% involved in external actor's breaches." ● "55% involved in organized criminal groups." ● "30% involved internal actors." ● “1% involved partner actors” ● "1% featured multiple parties." ● "72% of breaches involved large business victims." ● "58% of the victim had personal data compromised." ● "28% of breaches involved small business victims." ● "86% of breaches involved in financially motivated." ● “43% of breaches involved in web application” ● "37% of breaches stole or used credentials." ● "27% of malware incidents were ransomware." ● "22% of breaches involved phishing."

Users can quickly move to meet their requirements by sharing a vast pool of shared assets offered by cloud data storage. To offer consumers storage services, cloud storage merges numerous storage devices using "application cluster," "web technology," and "distributed file system." Many cloud service

providers exist in the literature like "Amazon Drive," "Apple iCloud," "BigMIND," "Certain Safe Digital Safety Deposit Box," "Cloud Me," "Crash Plan," "DriveBox," "Dropbox," "Google Drive," "IBackup," "IDrive," "Microsoft OneDrive," "Mega," "Mozy," "NextCloud," "pCloud," "SpiderOak," "SugarSync," "Sync.com," "Team Drive," "Ubuntu One," "Wuala" are a few of Cloud capacity suppliers expressed over offer cloud capacity with distinctive sorts of security like "Confidentiality," "Integrity" and "Availability." According to Bai. et al. (Bai. et al. 2020), cloud security controls provided in different CSPs (Like AWS, Azure, Cloud, Oracle, IBM, and Alibaba) are presented in Table 2. Our observation from Table 2 is that if the objective is to achieve complete security controls, then Microsoft Azure should be prioritized. For instance, if any reputable organization's email protection is necessary, Google Cloud can be given a higher priority than Microsoft Azure. For managing services like "backup and recovery," "vulnerability assessment," "patch management," and "change management" in the application, Microsoft Azure or AWS can be given higher importance. On the other hand, any of the CSPs mentioned above can be used if the requirement is to achieve only "firewall," "log analytics," "key management," "encryption at rest," "DDoS protection," "identity and access management," or "multi-factor authentication."

Table 2
CSPs like AWS, Azure, Google Cloud, Oracle, IBM, and Alibaba offers a variety of security control services.

Control Services involved in cloud service providers	AWS	Azure	Cloud	Oracle	IBM	Alibaba
MFA	YES	YES	YES	YES	YES	YES
Centralized Logging & Auditing	YES	YES	YES	YES	YES	YES
Load Balancer	YES	YES	YES	YES	YES	YES
LAN	YES	YES	YES	YES	YES	YES
VAN	YES	YES	YES	YES	YES	YES
VPN	YES	YES	YES	YES	YES	YES
Governance Risk and Compliance Monitoring	YES	YES	YES	By 3rd Party	By 3rd Party	YES
Backup and Recovery	YES	YES	YES	YES	YES	YES
Vulnerability Assessment	YES	YES	YES	YES	YES	YES
Patch Management	YES	YES	By 3rd Party	YES	By 3rd Party	By 3rd Party
Change Management	YES	YES	By 3rd Party	By 3rd Party	By 3rd Party	YES
Firewall and ACLs	YES	YES	YES	YES	YES	YES
IDS/IPS	By 3rd Party	YES	By 3rd Party	By 3rd Party	By 3rd Party	YES
WAF	YES	YES	YES	YES	YES	YES
SIEM & Log Analytics	YES	YES	YES	YES	YES	YES
Antimalware	By 3rd Party	YES	By 3rd Party	By 3rd Party	By 3rd Party	YES
DLP	YES	YES	YES	By 3rd Party	By 3rd Party	YES
FIM	By 3rd Party	YES	By 3rd Party	By 3rd Party	By 3rd Party	By 3rd Party
Encryption At Rest	YES	YES	YES	YES	YES	YES
Key Management	YES	YES	YES	YES	YES	YES
DDoS Protection	YES	YES	YES	YES	YES	YES
PAM	By 3rd Party	YES	By 3rd Party	By 3rd Party	By 3rd Party	By 3rd Party
Email Protection	By 3rd	YES	YES	By 3rd	By 3rd	By 3rd

	Party			Party	Party	Party
I AM	YES	YES	YES	YES	YES	YES
SSL Decryption & Reverse Proxy	YES	YES	YES	By 3rd Party	YES	YES
Container Security	YES	YES	YES	YES	YES	YES
End Point Protection	By 3rd Party	YES	By 3rd Party	By 3rd Party	By 3rd Party	YES
Certificate Administration	YES	YES	By 3rd Party	By 3rd Party	YES	YES

In Table 3, different evaluation schemes of data de-duplication present in literature are discussed in context with "Confidentiality," "Integrity," "Availability," "De-duplication type," and "De-duplication level." Various proposed techniques can work with existing CSCs. However, the CE algorithm is mainly used for securing data de-duplication over the cloud.

Table 3
Data Deduplication Schemes Comparison with different levels and security

References	De-duplication Type	De-duplication Level	Confidentiality	Integrity	Availability
Bai, J et al., (2020)	SS	BL	AES-128	Yes	Yes
Duan, Y., (2014)	SS	FL & BL	AES	Yes	Yes
He et al. (2020)	CS & SS	FL	Elliptic Curve	Yes	Yes
Keelveedhi S et al. (2013)	SS	FL	AES	SHA-256	Yes
Lee et al., (2020)	SS	FL & BL	Yes	SHA-1	Yes
Li, J et al., (2013)	CS	FL & BL	AES-256 with CBC	SHA-256	Yes
Li. J et al., (2015)	CS & SS	FL & BL	Yes	Yes	Yes
Li. J et al., (2020)	CS	BL	AES-256	SHA-256	Yes
Li, J et al., (2020a)	CS	FL & BL	AES	SHA-1	Yes
Liu, X et al., (2020)	SS	FL	AES	Yes	Yes
Meyer, D.T et al., (2012)	SS	FL & BL	No	MD5	Yes
Nayak, S.K. (2020)	SS	FL	AES	Yes	Yes
Ni. J et al., (2018)	CS	FL	AES	SHA-256	Yes
Prajapati, P. (2014)	CS	FL	Blowfish	SHA-256	Yes
Prajapati, P. (2017)	CS	FL	No	SHA	Yes
Premkamal, P.K et al., (2020)	SS	FL	Yes	Yes	Yes
Puzio, P et al., (2012)	SS	BL	Yes	SHA-256	Yes
Rahumed, A et al., (2011)	CS	FL	AES-128	SHA-1	Yes
Scanlon, M et al., (2016)	SS	FL	No	Yes	Yes
Shen. W et al., (2020)	SS	FL & BL	Yes	MAC	Yes
Shin, Y et al., (2017)	SS	FL	AES-256	SHA-256	Yes

Stanek, J et al., (2014)	CS	FL	AES-256	SHA-256	Yes
Storer, M.W et al., (2008)	CS	BL	Yes	HMAC	Yes
Wang, Y et al., (2020)	CS	FL	Yes	Yes	Yes
Yin. J et al., (2020)	CS & SS	FL & BL	No	SHA-1	Yes
Yuan, H et al., (2020)	SS	BL	AES-128 & AES-256	SHA-128 & SHA-256	Yes
Zhang, Y et al., (2020)	SS	BL	No	SHA-1	Yes
Zheng, Q et al., (2012)	CS	FL	Yes	Yes	Yes

Table 4
Analysis of different Data De-duplication schemes based on simulation

References	Simulation Details	KeyGen Cost	Hash Cost/Encryption	Size of Ciphertext	Size of Tag	Security Analysis
Rahumed, A et al., (2011)	C	H	H + SE	|F|	|SHA|	DI and DP
Meyer, D.T et al., (2012)	C + + and C	-	H	|B| + |F|	|MD5|	-
Storer, M.W et al., (2008)	NA	H	SE + H	|B|	|HMAC|	DI and DP, CompKS
Zheng, Q et al., (2012)	NA	Mul + H + Exp	SE + H + Exp + Mul	|F|	|G|	DI and DP, CompKS
Bai, J et al., (2020)	C	KS: H + Mul + Exp	SE + H + Pair + Mul + Exp	$\|G_T\|$ + |G| + |B|	|G|	OnBF, CompKS, LeHT, DP & DI
Zhang, Y et al., (2020a)	C++	KS: H + Exp	U: ASE + H + Mul + Pair + Exp KS:Exp	$\|G_T\|$ + |G| + |F|	|G|	OnBF, LeHT, DP & DI
Premkamal, P.K et al., (2020)	Python	U: Mul + H + Exp	U: Mul + ASE + Exp + Pair	|F|	|SHA|	DI and DP
Li, J et al., (2013)	Solidity and C++	H	CuE	|F| + |B|	|SHA|	DI and DP
Puzio, P et al., (2012)	HSM	H	H + SE	|F| + |B|	|SHA|	OnBF, CompKS, DP & DI
Li, J et al., (2020)	C++	H	SE	|B|	|SHA|	FA & OfBF
Li, J et al., (2020a)	C++	H	SE + CuE	|F| + |B|	|SHA|	FA, DP & DI
Liu, X et al., (2020)	C	H + Mul + Exp	(2t + 3) SE∎G + Exp + Mul + Pair	3 (t + 3) ∎ $\|G_T\|$ + |F|	|G|	LeHT, DP, DI & CR
Nayak, S.K. (2020)	C	H + 2Mul + 6Exp	SE + 6Exp + 2Mul + Pair	|F| + 2|G| + $\|G_T\|$	|G|	OfBF, OnB, FLeHT & DI
Zhang, Y et al., (2020)	C	-	H	|B|	|SHA|	-

Wang, Y et al., (2020)	Java	H + 6Mul + 6Exp	ASE + 6Exp + 6Mul + Pair	$2\|G\|+2\|G_T$	$\|G_1\|+\|G_2\|$	CompKS, DP & DI
Yuan, H et al., (2020)	Solidity and Java	H + 2Mul + Exp	SE + 2Exp + Mul + Pair	$\|G_1\|+\|G_2\|+\|G_T\|$	|SHA|	DI and DP
Yin, J et al., (2020)	NA	-	H	-	|SHA|	DI
Shen W et al. (2020)	C	H + 2Mul + 2Exp	SE + 2Exp + 2Mul	$\|F\|+\|B\|$	|MAC|	OfBF & DI

The evaluation of different data de-duplication systems concerning parameters like "key generation cost," "encryption/hash cost," and "ciphertext size," along with security analyses for resistance against different attacks like "online and offline brute force attack," "compromise of the key server," "leakage from hash" and "collision resistance" etc. are deliberated in Table 4 for reference. Different notations used in Table 4 are "H: Hash of Data," "Mul: Group Multiplication," "Exp: Group Exponentiation," "U: User," "KS: Key Server," "SE: Symmetric Encryption," "ASE: Asymmetric Encryption," "CuE: Customized Encryption," "Pair: Pairing," "$|F|$: File," "$|B|$: Block," "$|GT|$: Size of group elements," "N: Number of Key Server," "t: Number of key share server (KSS)," "DP: Data Privacy," "DI: Data Integrity," "CR: Collusion Resistance," "FA: Frequency Attack," "OfBF: Offline Brute Force Attack," "OnBF: Online Brute Force Attack," "CompKS: Compromise of KS," and "LeHT: Leakage from Hash or Tag."

3. Data De-duplication

The most practical approach in present times is cloud computing and statistics de-duplication. The demand for digital information has increased, justifying the adoption of record de-duplication in a wide storage network and the cloud. The enormous amount of data being stored on storage systems emphasizes the importance of creating more straightforward methods to get rid of unwanted data. With the development of statistics de-duplication and its multiple ways, cloud computing can eliminate superfluous duplicates. De-duplication reduces bandwidth usage, storage costs, and energy consumption. Data de-duplication can be classified based on location and size: at "Server-Side (SS)" or "Client-Side (CS)" and based on size: "File Level (FL)," "Block Level (BL)," and "Byte Level (ByL)." This classification is shown in Fig. 3. One method for performing de-duplication in an encrypted domain is to use a convergent encryption algorithm.

Differently formatted files (File 1, File 2, …, File 5) are depicted in Fig. 4, which shows the storage of the heterogeneous files in the file storage database using secure data de-duplication schemes. Files (File 1 (a), File 2 (b), File 3 (c), File 4 (d), File 5 (e)) conversion into vectors is shown in Fig. 5. The comparison of data de-duplication between files is depicted in Fig. 6.

Figure 7. outlines the fundamental idea of a multi-cloud atmosphere. Any record will be apportioned and examined for changes, to begin with. The information broadcast avoids all parts that are, as of now, put away within the framework by any client. Henceforth, by using the approach of secured data de-duplication, as where modern substances will be transferred and put away securely.

A flowchart of the concealing computation is shown in more detail in Fig. 8. The process will start once a record transfer asks has been part of the partitioned request. Initially, it checks if the segment is known by looking for its unquestionable locator within the database. On the off chance that it is obscure, an unused storage area reference will be arbitrarily made. Renaming the operation to an unused irregular title will be accomplished by the benefit once the uploading is completed. This step is required to muddle the supplant operation, which takes put afterward within the algorithm.

Data de-duplication, a crucial step in removing duplicate copies of information, is a component of data compression techniques frequently used in cloud storage to reduce storage capacity and implement data-saving procedures. Specific techniques are utilized on different forms of statistics, such as material, pictures, and videos, using distinctive de-duplication procedures. Each of the three record types has unique characteristics and excellent capacity positions. De-duplication systems include explicit methods to locate and delay copy records depending on the data (Maan, AJ 2013, Xia, W et al., 2014). Accordingly, the type of information is vital for enhancing the de-duplication processes. The records arrangement is essential for evaluating, getting, and organizing the points. It takes bit-stage blending to find duplicates in usable records. Because of the diverse configurations of statistics, there are different ways to examine copies in text, picture, and video. A small number of data duplicates, or "replication aspect," are kept in a sizable distributed storage device to achieve high data accessibility. Any reproduction stats above the replication factor are evacuated to reduce the need for a garage, storage cost, calculation, and electricity. De-duplication techniques for large-scale distributed storage architectures have gained popularity in both the academic and commercial worlds due to their numerous benefits to the business.

However, these solutions need to be improved due to the effectiveness and sufficiency of ways for coordinating the facts. Researchers and business analysts are working to develop more practical distributed de-duplication techniques. The entire document is divided into chunks of fixed or varying sizes. According to Venish and Sankar (2015), a de-duplication system saves a single copy of each segment and uses rules for multiplication portions. Suppose that the de-duplication engine of the capacity device learns that this is stored somewhere inside. In that scenario, it stores a pointer in the duplicate information area that points back to the genuine duplicate. It makes it possible for the barriers inside the garage device to be removed, freeing up memory space.

RW Ahmad et al (Ahmad RW et al. 2015, and Barreto J, Ferreira P. 2009), After reviewing the most recent research on de-duplication techniques, it was determined that it was necessary to carefully evaluate the text that was available on the subject. In any case, this stage summarizes the motivations, dedication, and interest in this subject.

1. The functions, necessities, benefits, and drawbacks of a de-duplication approach to improving the overall implementation of an extensive memory system have been discussed.
2. The garage, point of utility, and stage have all been considered when organizing the present de-duplication techniques. The de-duplication strategy has also been described as dependent on videocassette, photo, and literary content. It has been made clear how important content de-duplication techniques are to take into account. There are numerous scientific classifications, picture- and video-based de-duplication, and content evaluation offered. As a result, this literature assesses writing and provides a broad overview of de-duplication techniques (Alvarez C. 2011).
3. Future de-duplication research criteria were highlighted for academic and business experts.

De-duplication of facts is a phrase that was developed in the early 2000s to handle massive storage systems with high granularity (Gu M. et al., 2014, Tian Y et al., 2014, Hovhannisyan H et al., 2016, Mandagere N et al., 2008, and Paulo J, 2014). It opposes conventional data compression methods that ignore repetition over a limited record organization that is heavily dependent on excessive intra-document information. Instead, the cryptographic hashes of each text or mass are used to figure out how to identify the copies. On mixed media substance, these tactics were also revised in 2008. By evaluating the degree of similarity between different photo or video frames, the highlight extraction and hashing techniques analyze the replicated multimedia content. These techniques for reducing record duplication emerged to address the issue of creating data length within storage structures proposed by (Banu AF, 2012, Chandrasekar C. 2012 and Xu J et al., 2016).

Bytes and strings can be compressed using Huffman and dictionary coding, while de-duplication algorithms eliminate log or bite redundancy. In addition, by using delta and loss-related compression techniques, redundant fact-reduction techniques dating back to the 1950s started to appear more frequently in the 1990s. Finally, in 2000, optical de-duplication techniques and computer de-duplication tactics arrived here.

Modern CRM and decision-assistance programs draw heavily on statistics warehouses, which are repositories of data collected from various statistics assets (Client Dating Control). Because choice guide evaluations of data warehouses influence key business decisions, accuracy is essential. Facts sources can be used to observe impartial and almost incompatible standards since they are impartial. Most of the time, confusion affects group sites with diverse items with a similar percentage of nomenclature. In the finest recovery tool, a user can enter an entity or concept call and look for results grouped according to the unique entities/standards that percentage that call. Including more data in the listed files is one way to enhance such devices (Xia, W et al., 2016). Businesses typically become aware of rational-specific discrepancies or inconsistencies while compiling data from several sources to create an information warehouse. Such issues fall under the category of fact heterogeneity. When data from numerous statistics sources are used to keep the information overlapping, the facts are foolishly repeated. However, there are grammatical flaws, conflicting statistical source conventions, omitted fields, and other issues with the data collected for the facts warehouse from external assets.

To provide for the provision of excessive statistical quality, arriving statistics tuples outdoor assets need justification and modification. High-quality statistics show that the data warehouse should operate " error-free " Techniques for record cleansing are essential for enhancing the first class of records. Information mining approaches successfully mined data using software or algorithms to produce artistic and useful analysis. Descriptive and extrapolative facts about fashion are two different categories. Illustrative representations like “clustering”, “summarization”, “affiliation Rule”, “series discovery”, and many others locate the homes of the investigated data using tracing patterns or association records. A group of substances is grouped by clustering at the number of subsets known as clusters, where items from the same cluster are relatively similar to one another and those from different clusters are dissimilar (Mao B et al., 2016, Kim C et al., 2012, Lillibridge M et al., 2009, Zhu, B et al., 2008, and Li, YK et al., 2015). To consolidate and summarize the data, clustering is a fundamental technique that may provide a summary of the preserved information (Venish A, 2015). Predictive versions like classification, regression, time series analysis, prediction, and so on select whether to employ known values for an unknown information type technique. To build a version, the classification rules study the training set.

The above-described classification model is used to classify new things. The k-nearest neighbor approach (k-NN), the Naive Bayes algorithm, the neural network (Wang, J., 2016, Di Pietro R., 2016 and Chen, C.P., 2014), and the auxiliary vector mechanism are examples of strict classification techniques (SVM). The widely used type algorithm, k-NN, exhibits favorable performance uniqueness and is employed in many diverse applications, including text primarily dependent on image recovery data and three-dimensional item interpretation (evaluation of entropies and deviations). The process of "data cleaning," also known as "records cleaning" or "scrubbing," raises the caliber of statistics by identifying and removing data mistakes and anomalies (Witten, I.H et al., 1987). By removing data set modification and reducing fact replication, it inspirationally improves the overall statistics compatibility. Modern information-removing techniques identify record replicas, unrecognized values, record and field resemblances, and replica deletion. The duplicate document detection technique is used to identify additional or multiple characteristics of one exceptional actual global object or item.

3 (A). Repetitive Data Minimising Methods

Repeated statistics reduction approaches have been created to control the expanding number of virtual records and select surplus at the byte, string, and report levels. Similarity exists between the business and its development of redundant statistics reduction approaches. A bit-price discount strategy is used to compress information to express data in a condensed manner. This looks for redundant information and reduces the amount of storage space needed. The lossless compression process serves as the standard definition of statistics compression (Chen CP, Zhang CY. 2014 and Di Pietro R et al. 2016). The distinct, unique information is rebuilt from the compacted data using the lossless compression technique. By highlighting useless data, such as the compression of jpeg photos, lossy compression reduced the details. The first-act prediction is reenacted in this. With lossy compression algorithms, data in movies and music is compressed (Maan 2013). This section contains essential historical background on

redundant information reduction methods, demonstrating the development of each conventional method for lossless data compression, delta compression techniques, and information de-duplication techniques. Additionally, it provides a scientific classification of many approaches and their expansion.

3 (A-i). Lossless Data Compression Methods

The term "information compression" was established due to its wide usage. Word reference encoding, run duration encoding, and entropy encoding are three techniques for data compression that don't sacrifice quality (Ng CH et al., 2011). A few sequences in a little served as symbols for the character series. Vast amounts of superfluous statistics are generated and ejected in these strings as designs of facts.

Byte stage: Entropy encoding is used in the early record compression techniques to detect byte-level redundancy. Two entropy encoding methods are used to represent frequently occurring samples with fewer bits each: Huffman encoding and mathematical coding. The best prefix code was developed by D. A. Huffman using binary trees with frequency considerations. Variable-length codes are used in place of constant-length codes (Shanmugasundaram S. et al., 2011). Short coding is used to present frequently used images. Elias created arithmetic coding in 1960 by Witten et al. (1987). The entire message is stored as a set of floating-point numbers within a predetermined range.

String Level

In a different study, the String Level approach developed by Bhadade US and Trivedi AI (2011) was designed to review and eliminate repeated strings.

3 (A-ii). Mechanisms For The Compression Of The Delta

The delta compression method was developed in the 1990s to compress similar files or chunks. The two most popular applications are backup storage and remote synchronization. It searches similar chunks for matched texts using a sliding window that is byte sensitive. Variations between sequential files and complete records are kept by Xia et al. in the "delts" or "diffs" form (2016). One of the elements of the Delta Compression Strategies is the string step. The delta computer uses a byte-clever sliding window during the X delta and Z delta string phases of delta compression to find redundant strings between the destination chunk and the source chunk (Brereton P et al. 2007).

3 (A-iii). De-duplication Methods For Data

- The records de-duplication strategy was first implemented in 2000 to help with coarse-grained global compression. While statistical de-duplication techniques can be applied at the reporting or sub-reporting level, the methods outlined above need extremely long processing times and are not feasible. By using pieces of fixed or variable size, it compresses statistics. Using cryptographic hash functions, some chunks are assigned hash values, and duplicates are recognized by hash values

that are the same. For record-level de-duplication at the file level, both methods are applied, and reporting is handled separately. This verifies the backup document's index to verify the attributes kept there. If an identical record is available, either an indicator of the current document is given, or the index price is updated and stored. The single instance garage, where it is simple to use the whole-document hashing approach, is the best example of the study and has thus been saved because creating and processing report hash numbers is clean and uses very little energy. However, the paper contains a few one-byte exchanges that lead to novel technology that requires exclusive data. With the development of block de-duplication algorithms, the issue of document-level de-duplication is resolved.

- Block-degree (sub-report-level) de-duplication: These methods include breaking up a file into several compact blocks of either fixed or variable lengths. Comparable blocks are recognized using hash techniques like “MD5”, “SHA-I”, “Rabin fingerprinting”, and “similar hash algorithms”. As a result, a block is updated in its index and, unlike other blocks, gets written to the disc. In every other situation, the block of equivalent truth will be moved back to its original position. As a result, IDs will significantly grow, necessitating more processing power. Constant-duration or variable-length de-duplication and block-stage de-duplication are both types of de-duplication (Meyer D.T, 2012). When analyzing fixed-length computer blocks utilizing de-duplication of the fixed-length block period approaches, the same statistics block is not duplicated. The disadvantage of the fixed-length block technique is overcome by variable-length innovation. There are blocks of statistics with different lengths. One kind of method is used by variable block algorithms to choose the block time. As a result, their chord block boundaries may "drift" within. To keep the borders of several block locations from altering when changes are made to one part of the structure (Witten, I.H et al., 1987). The phase includes bytes in the duration in a content-based and restricted manner. It increases the granularity and flexibility of a block's management.

3 (A-iv). Merits And Demerits Of Data De-duplication

The subsequent deserves of de-duplication are identified as follows:

- **Lesser garbage area**: The amount of storage space needed to store archives, files, or various forms of data is reduced via de-duplication. The most precise reproduction of the statistics is stored, and duplicate copies are removed. Therefore, storing more data results in greater free space (Maan, AJ, 2013).
- **Boost of community bandwidth**: It makes sense to recommend data duplication to avoid sending replication copies over the network as the precise copies are archived on the disc. That is, the amount of bandwidth needed by the community can be decreased through de-duplication.
- **Energy reduction consumed**: By lowering the demands for capacity and electricity, de-duplication is a capacity optimization strategy that minimizes energy usage. In decreased storage, less electricity and coolants are required. It reduces the burden on the tech gear and conserves energy.

- **Lessen usual garage value**: Significant time, space, network bandwidth, human resources, and financial savings are made possible through de-duplication. Additionally, it improves the effectiveness and efficiency of storage systems.

3 (A-v). Demerits Of De-duplication

- **Impact on garage efficiency**: The main garage machine's fixed-size technique stores many chunks at different memory locations. It causes fragmentation issues that hurt the outcomes. For its execution, the de-duplication process needs additional resources like memory, recollection, and bandwidth. Any ineffective de-duplication strategy reduces the effectiveness of a sizable storage network.
- **Information integrity loss**: The data blocks are listed for easier hash value searching. Due to a hash collision, equal hashes can be generated for unusual data blocks, which could jeopardize the records' accuracy. Therefore, hash collisions must be managed correctly to prevent any data loss and keep the data's integrity.
- **Issues with backup devices**: Information relocation and technique may require a new hardware tool for fact de-duplication. According to Borges EN et al., such support equipment may also increase costs and impact storage performance (Borges EN et al. 2011).
- **Respect for privacy and security**: The de-duplication techniques can always be finished.
- **(Storage) Archiving**: It can be used from the repository's entrance. To prevent such security lapses and the loss of sensitive data on devices, the security of the deductibility mechanisms must be properly planned.

3 (A-vi). The Ann Technique

A synthetic neural network is a device that, in other words, mimics organic brain devices and is primarily focused on the operation of biological neural networks. A tool for specific approaches like categorizing, optimizing, etc., is an ANN. A neural network can perform tasks that a linear program cannot complete. The utilization of the neural community's parallel nature can continue without issue, even if a specific component malfunctions. A learning approach is implemented using a neural network, eliminating the need for further programming. Particularly, a synthetic neuronal culture can take the following forms:

- If data from input to output processors must be intentionally sent forward, the neural network is referred to as a feed-forward neural network. For example, records processing may involve several (layers of) devices. Still, no remarks are present— meaning that links running from device outputs to devise inputs in the same layer of lower levels.
- Recurring neural networks contain references to statements. Contrary to feed-ahead networks, the variety of residences within the community is crucial. In other cases, the activation levels of the devices undergo a resting process so that the neuronal population can form a stable nation where such activations no longer alternate. For the complex activity to form the neural network output,

additional packages depend on changing the output neurons' activation values. Therefore, a multi-layered neural network was employed. The facts, hidden, and output layers make up the skeleton of a neural network. The following discernment may show the framework that governs the basic shape of a multi-layered neuronal community.

3. (B). Statistics Of Data De-duplication

These documents are duplicated on distributed storage devices for high dependability, accessibility, and disaster recovery as data in cloud storage expands tremendously (Xia, W, 2014). For the device to be protected against errors and to maintain high availability, the replication factor, or minimum range of data replications, is essential. The replication component must be the only wide variety left in the garage system. In all other cases, duplicate data on the home computer adds to the strain due to the limited bandwidth and additional space. De-duplication techniques are used to increase the storage device's capacity for cost and use terms to lessen or manage this information duplication. The usefulness of de-duplication techniques depends on the nature of the facts, such as whether they are based, unstructured, or semi-structured. Statistics fit into the same categories as text, pictures, and videos. The maximum possible network traffic processing speed, overall storage capacity, and storage device efficiency are all impacted by replication knowledge (Clements AT et al., 2009). Researchers could comprehend how cutting-edge de-duplication methods for garage buildings are being built as a result. This involves sending green information to a garage unit and eliminating duplicate records. De-duplication is a technique for automatically removing reproduction statistics from garage systems.

With Microsoft's assistance, the de-duplication reduction of facts is reported. Over four weeks, 857 desktop Windows machines from NetApp Microsoft assessed the stability of space savings between full-document and sub-record de-duplication (Meyer, D.T, 2012). According to observation, block-level de-duplication only meets 32% of the initial requirements, whereas total record de-duplication has a gap reserve of 75%. Records de-duplication was also employed in a virtual library that used duplicate bibliographic information reports to detect the use of similarity functions on two real datasets (He Q et al., 2010, Zhou R et al., 2013). Datasets include article quotation data for the core collection and metadata information for two free virtual libraries (BDB Comp and DBLP). Think about how the high-quality metadata de-duplication enhances the digital library data set by 2 to 62 percent and by 7 to 188 percent in the item dataset. According to NetApp, de-duplication may remove 95% of replicate information inside storage structures. Pereira and Paulo (2014b). According to experimental data, 95% of regular backup costs are covered by VMware, 30% by email, 35% by records, and 72% by regular backups.

3. (B-i). Classification Of De-duplication

3. (b-i). Classification of De-duplication *techniques*
Based on local and global deductions, the categories of storage, such as primary or secondary de-duplication, source/goal, and handling time, were divided into online and post-process deductions. The

four criteria for de-duplication classification taxonomy are length, form, timing, and degree.

They are classified as follows (Witten, I.H et al., 1987):

- Depending on the storage type, de-duplication

Based on the type of storage, de-duplication classification was carried out. Paulo and Pereira (2014b) and Banu and Chandrasekar apply de-duplication for primary and secondary storage, respectively (2012). Initially storing: The main memory or active storage directly available to the CPU is where the primary storage-based de-duplication is executed. An additional or external storage device without direct access to the CPU is referred to as secondary storage. It backs up data from primary storage. Only historical data is stored in and retrieved from these systems. Storage archives, snapshots, and backup storage are a few examples.

- De-duplication based on the type

The supply side or the appropriate direction is used to carry out the de-duplication process. De-duplication is based largely on source and target, depending on these types. Before being transmitted to the backup objective, the statistics on the supply side are plicated (Meyer DT, 2012). The program remembers moving the data to the backup server and is installed on the server's CPU. Therefore, it also lowers the bandwidth, storage, and time requirements for information backup. On a dedicated garage device on backup servers, de-duplication is typically carried out on the duplicates. In this sense, all de-duplication functionalities—garage utilization with the added benefit of committed use—are addressed by dependable hardware de-duplication machines. Therefore, the statistics for complex garage systems may not have any overhead. However, as discussed later, it requires more resources and is better categorized as a publishing method.

- De-duplication of data based on timing

Timing-based de-duplication only applies at the instant the de-duplication algorithm is running. After that, it sets a deadline for completing the de-duplication process. De-duplication techniques like duplicate searching are the main remedy for timing-based de-duplication. It can be done using either an asynchronous/in-band operation or an asynchronous/out-of-band operation. The timing-based de-duplication was also divided into inline and post-system categories. Before being written to disc or on the source side, the deduplicated data is processed. Therefore, additional disc space is not required to maintain and defend the facts. Since the information is exceeded and processed most effectively when Inline de-duplication requires further processing, it boosts efficiency. In addition to being faster than inline de-duplication, it also goes by offline de-duplication (Gu M. et al. 2014). This enables the backup time to be cut down.

- Data Deduplication based on level

There are two types of data de-duplication: local level-based and global level-based. Only one VM can do local de-duplication, and only one node can identify replicas. Because it cannot eliminate all duplicates, it

impacts overall performance (Xia, W et al., 2014). It has a few facts and nodes. De-duplication, or noneducation of the report, is a process that is carried out in a distributed setting or across many datasets.

- De-duplication of data in the cloud for storage systems

The data de-duplication approach is frequently employed in cloud storage, backup environments, and data storage systems since it lowers the need for storage space and storage costs. De-duplication techniques use only one body replica to save data, which consumes more internet bandwidth than the total amount of data sent to the cloud or the community. It promotes the acceleration of cloud backup (Hu et al. 2016), leading to faster and more environmentally friendly information security operations.

Direct cloud de-duplication, secondary garage copies, and cloud gateway de-duplication can all be deployed for cloud storage de-duplication (Ni J et al., 2018). De-duplication can also be employed in unique garage systems, such as primary, secondary, and cloud storage platforms. Devices for personal, public, and mixed cloud storage all profit from the de-duplication technique.

4. Secure Data De-duplication: State-of-the-art

In the past, numerous scholars have proposed several studies in data de-duplication for cloud storage. One of them was a cloudy system presented by P. Puzio et al. (P. Puzio et al. 2016), which ensured block-level data de-duplication and confidentiality. It is carried out by removing redundant copies of data to limit cloud providers' capacity. De-benefits duplication's come at a cost, however, in the form of increased security and assurance risks. More highlights should be provided to ClouDedup, such as retrievability and data integrity proof.

B. Gupta et al. (B. Gupta et al. 2017, B. Gupta et al.) provided their proposal with the audit of the sentiment analysis concepts on Twitter, illuminating the techniques discovered and models employed along with a condensed python-based methodology. A sack of words model, a form of content unigram model, is created using the NLTK toolkit. Python has been used for logical processing using the Scikit-learn library and NumPy basic module. It does, however, have application problems with the slang used, and the abbreviated forms of many of the phrase analyzers need to perform better as the number of classes is increased.

To ensure the secrecy and security of the information, R. Raghatate, S. et al. (2014) proposed a straightforward information protection paradigm in which data is encrypted using the Advanced Encryption Standard (AES) before being sent into the cloud. Cloud computing is not an exception to the security and protection concerns that are true in the processing world. By shifting the encryption and decryption process from the cloud to the self, they provide the engineering and guidelines to scale up the security and protection of the information owner. This design may make it more difficult for a cloud provider to misuse or mine customer data. There should be tight restrictions on planning and

computation to prevent the misuse of cloud computing's capabilities. When CSP can enable search on encrypted material, this is possible (Y. Peng et al., 2012).

1. K. Akhila et al. (K. Akhila et al. 2016) proposed convergent encryption as an easy way to make de-duplication compactable with encrypted data. De-duplicating encrypted data in the cloud while maintaining security is a difficult problem. They claimed it is difficult to use the cloud to cut down on duplication and compromise security. It can be improved even more with new storage optimization methods.
2. J. Amalraj and J.R. Jose examined the various encryption techniques in 2016 and evaluated the performance of many symmetric algorithms. Among them are DES, 3DES, RSA, AES, and Blowfish. DES are block figures that encrypt and decrypt data using a common secret key. The DES technique starts with a string of a set length in plain text bits and transforms it into a string of ciphertext bits, with each block being 64 bits throughout many operations. The handling process consists of sixteen similar stages called rounds. The initial permutation (IP) and final permutation (FP) are additional starting and finishing permutations (final permutation). 3DES is an improvement of DES, which has a key size of 192 bits and a 64-bit size (N. vurukonda, B.T. Rao, and T Jiang, 2016). Like the first DES, the encryption approach involves increasing encryption and the average safe time.

In cloud computing, de-duplication emerges as an active research topic. This section comprises the research related to the work done in data de-duplication incorporating convergent encryption. The baseline approach and Dekey are the two distinct approaches recommended by Jin Li et al. for CE key management (Jin Li et al. 2014). Two significant deployment concerns plague the default strategy. First off, because it generates several keys, its efficiency is quite high. Because each user is required to safeguard their master key, it is also unreliable. The user also loses the data if the master key is lost.

Dekey, however, makes use of RSSS (Ramp Secret Sharing Scheme). The encryption key is distributed using this method to the various key servers. Instead of distributing the key, the author uses a key generation method in the cloud as a service to empower the key to authorized consumers. Taek-Young et al. suggest a different method for creating the key with access privileges for convergent encryption (Taek-Young et al. 2016). Only individuals with the appropriate authorizations can access the shared data to create a convergent key. This method uses the DupLESS scheme's RSA blind signature-based oblivious PRF protocol to generate a key from the key server (MihirBellare et al., 2013). Privileged information has been used in this technique to restrict access to the data to authorized users only. This system consists of users, cloud storage providers, and an authorization server, among other entities. AS is employed for the creation and administration of private keys. Additionally, based on a user's privileges, it is utilized to calculate a convergent encryption key for a particular file. The authors, however, had to have clarified how someone who has the same data copy might obtain the convergent key.

DICE (Dual Integrity Convergent Encryption), a secure data de-duplication technique, was proposed by Ashish Agarwala et al. (Ashish Agarwala et al. 2017). It is focused on removing threats and preventing duplicate faking and ii) providing integrity checks at both client and server ends. The produced tag is

uploaded to the server, where it is subsequently subjected to an integrity check. Only when the client downloads the tag to access the ciphertext is this check run. As a result, bandwidth usage is decreased because only the tag is sent rather than a lengthy ciphertext, and de-duplication is also accomplished at the same time. The main generating and management method needs to be thoroughly defined in this article.

Nearly 80% of the firms surveyed by DuBois et al. (DuBois et al., 2011) revealed that they were looking into data de-duplication technologies for their storage frameworks to reduce redundant data and thereby increase storage effectiveness and lower storage expenses. Many academics have already suggested many methods for data de-duplication in cloud storage with CE.

Cheng Guo et al. (Cheng Guo et al. 2020) provide a client-side de-duplication system (R-Dedup) that is randomized and secured. Both peer users and reliance on any third party are not prerequisites for this system. For users who have the same file copy, an encryption key is generated. In the data verification stage, R-Dedup also provides user authentication for the cloud server, ensuring data integrity. R-dedup provides a straightforward architecture with increased security.

A Secure De-duplication and Virtual Auditing of Data in the Cloud (SDVADC) mechanism is proposed by Geeta CM et al. (Geeta CM et al. 2020), which efficiently deduplicates the data of encoded information. Furthermore, virtual Auditing Entity (VAE) is inbuilt into this proposed mechanism which virtually supports efficient auditing of the data owner's file during the download process.

Xiang Gao et al. (Xiang Gao et al. 2021) proposed a low-entropy secure data auditing scheme having file and authenticator de-duplication. The computation of authenticators and tagging of a file is designed in a new way. In this scheme, one copy of the data block and authenticators for the duplicated file is stored in the cloud.

Yunling Wang et al. (Yunling Wang et al. 2021) focus on the secure de-duplication scheme and efficient user revocation. A multi-user updatable encryption is proposed first, which helps the data owner to update the ciphertext efficiently for a new group of users. Then by using this technique, a new de-duplication scheme is constructed. While updating the data authority, a token is to be sent to the cloud by the data owner. Only then will the cloud update the ciphertext for a new group.

In this research, Guipeng Zhang et al. (Guipeng Zhang et al. 2021) present a blockchain-based de-duplication technique for the cloud. To achieve the approved de-duplication, a novel hierarchical role hash tree (HRHT) is also created, which maps the relationship between the user's role and the role key.

5. Convergent Encryption For Secure Data De-duplication

Integration of Convergent Encryption (CE) with Data Deduplication plays a vital role in ensuring that the advantages of integrated CE and data de-duplication are retained to solve the issues of CE, i.e., "Dictionary attack," "Confirmation of File (CoF)" and "Learn-the-Remaining information (LRI)" Puzio et al.

(Puzio et al. 2013) proposes ClouDedup. Data confidentiality is offered by CE during de-duplication. Each original data copy is used to generate a Convergent Key (CK), which is then used to encrypt copies of the data by users or data owners. The user provides a tag that will be used to distinguish duplicate copies of the data for each copy of the data. The user sends the tag to the server first to check for duplication to see whether the same copy has already been stored. The CK and tag are both separately obtained, hence maintaining the confidentiality of data cannot be done by using the tag to discover the CK. The server side will store the encrypted data copy and its accompanying tag.

5 (I). Ce Algorithm

Step 1: $KeyGen_{CE}$ (M)

Step 2: Enc_{CE} (K, M)

Step 3: Dec_{CE} (K, C)

Step 4: TagGen (M)

A key generation algorithm K, CK is mapped to a copy of the data M. The inputs CK, K, and the data copy M are all accepted by the symmetric encryption technique C, which then returns a ciphertext. The ciphertext C and the CK, K inputs to the decryption mechanism M, which outputs the original data copy M. The tag-creation algorithm, T(M), translates the original data copy M to the tag T (M).

5 (Ii). Steps For Ce By A User Alice

- Get a file F. Alice derives an encryption key K from the file by applying SHA-256 to file F.
- Then we have to encrypt that file F into ciphertext C with AES under key K.
- Protect the key K by encrypting it into W using its public key.
- Upload or send both C and W, ensuring that C and W are stored together.

5 (iii). Alice can decrypt the encrypted file C into the original file F by

- Receiving the C and W files from the site/storage and downloading them.
- Using a personal decryption key, extract the key K from W.
- To recover the original file F, decrypt the ciphertext C using the AES algorithm and key K.

5 (Iv). Ce Using Sha-256 And Aes

From convergent import CE

C1 = CE (“Secret Code”)

Key, block id, ciphertext = C1■Encrypt (“ShahnawazAhmad”)

If (len (ShahnawazAhmad) = = len (ciphertext))

true

C2 = CE ()

Plain_text = C2■Decrypt (key, ciphertext)

Plain1 text = = ShahnawazAhmad

true

Using the user's unique information, CE creates a key (hash value) and uses it to jumble the data. The identical information will be jumbled using this technique, which can help identify duplicate data. Additionally, a technique for creating the CK by processing the client's data was suggested by this inquiry. The CEKGA is denoted as $CE_K = CE_{keyGen}(C_k)$. CE_K is a CE key, and $CE_{keyGen}(C_k)$ is a primitive function for generating CEKGA. The unique phases in the proposed algorithm are followed by the suggested CEKGA, which is a digested algorithm that reads the client's input and produces the CE_K.

5 (V). Pseudo Code Of Cekga

S CE_{keyGen} (C_K)

1. C_D ← Client's data
2. Ns ← size of (C_D)
3. C_D [S] ← array (C_D)
4. for i ← 1 to Ns
   $B_{SC}C_D$[i] ← ASCII (C_D[i])
   next i
5. j ← 1
6. k ← 1
7. for i ← 1 to Ns
   if (i%2= = 0) then
   E_{BLOCK} [k] ← $B_{SC}C_D$[i]
   k←k+1
   else
   O_{BLOCK} [j] ← $B_{SC}C_D$[i]
   j←j+1
   end if
   next i
8. Middle ←Ns/2
9. for i ← 1 to Middle
   S_{BLOCK} [i] ← E_{BLOCK} [j] + O_{BLOCK} [j]
   next i
10. for i ← 1 to Middle
    $BinU_D$ ← append (binary (S_{BLOCK} [i]))
    next i
11. BinNs ← size of ($BinC_D$)
12. Blk ← BinNs/256
13. n ← 1
14. Bin ← 256 0's block
15. for i ← 1 to Blk
    Binck[i] ← split ($BinC_D$, n, n+255)
    n ← n + 256
    Bin ← 0's & 1's addition (Bin, Binck[i])
    next i

16. $BinC_D$ ← B
17. Blk ← size of ($BinC_D$)/8
18. n ← 1
19. for i ← 1 to Blk
    $DecC_D$ [i] ← deci (split (BinUp, n, n+7))
    $BscC_D$ [i] ← ASCII ($DecC_D$ [i])
    BscBuff ← append ($BscC_D$[i])
    n ← n + 8
    next i ⊕ C_k
20. CE_K ← BscBuff
21. End S

The user can utilize the generated CE_k to use encryption to encrypt data using the pseudo-code. The suggested technique is sometimes called a digest algorithm because it breaks down the user's input into key-based data content. The key generation and management function provide a key utilized to create the algorithm proposed by CE_k (KGMS). The key does not have to be carried about by the user. You can safely remove this key. CE_k is also saved in KGMS and is linked to Ck since the key is checked by KGMS and provided to users once they submit and prove their ownership of the data. Users must verify their data ownership to retrieve these keys from KGMS.

For our, by and large, investigated work, we set up the simulation environment. Amazon Linux AMI 2018.03.0 (HVM) miniaturized scale occurrence with 30 GB storage and 2GB RAM, EC2 instance t2.micro, and RDS database are leased from AWS cloud foundation - a cloud server for cloud storage. The proposed CEKGA for the C_k era is developed using Java 14.0.1, and it is facilitated within the stage given by the AWS platform. A primary cloud service is provided by the cloud server, where this CEKGA is additionally included. All these handles are coded in the primary cloud service, which is sent within the AWS PaaS from AWS. The recreation environment is represented diagrammatically in Fig. 9.

5 (B). Proposed Secure Data De-duplication

The proposed algorithm is shown in Fig. 10. Starting with the SHA-1 hash of the uploaded folder as the document level, Token Req (Tag, UserID) asks the private server (PS) for a token along with File Tag (File), and DupCheckReq (Token) asks the storage server (SS) to check the file for duplicates by sending the file token obtained from the private server. ShareTokenReq (Tag, "Priv.") requires the PS to create the shared file token with the target sharing privilege set and the File Tag (File). FileEncrypt (File) encrypts the file with CE using the 256-bit AES algorithm in Cipher Block Chaining (CBC) mode, and the convergent key is derived from the file's SHA-256 hash. FileUploadReq (FileID, File, Token) uploads the File Data to the SS if the file is distinct and the changes of the File Token (FT) are saved. The PS is implemented with matching request handlers for token creation and supports HashMap key storage. TokenGen (Tag, UserID) loads the user's associated opportunity keys and generates an HMAC-SHA-1 token. Based on the ideas of MD4 and MD5, SHA-1 generates an MD. Only one bitwise rotation in the message scheduling of its compression mechanism distinguishes SHA-1 from SHA-0. MD is a 160-bit hash value generated using SHA-1. This hash value is represented as a 40-digit hexadecimal number.

The suggested methodology is employed at a medical facility so that a doctor can access it remotely. De-duplication is used at a healthcare facility to expand its cloud storage capacity. More storage space is available when data duplication is avoided. In medical facilities, the data will be kept in files with various formats and records, and its report will be frequently updated based on the check-ups. Due to the bulk of medical pictures like MRI, ultrasound scans, and echocardiography reports, data duplication lowers storage capacity in such a case.

Additionally, the security element of this medical data is crucial because any security breach could result in the disclosure of the patient's personal information. Therefore, a data de-duplication method and convergent encryption method are suggested as solutions to this issue to increase information confidentiality and storage capacity. First, redundant data is eliminated to conserve storage space in the cloud. Then, the various file types are subjected to the data de-duplication procedure. The following describes how the suggested methodology operates:

- After the user completes the login process, the upload question will show up. By choosing "yes," the user can upload files containing the data. The upload will stop if the user chooses "No."

- After selecting the image, the cloud administrator will search for duplicate images. If there are no duplicate photos, the image will be posted immediately; otherwise, the ownership will be checked to ensure a successful upload. The original image kept on the local server will be used in its stead if the ownership of the image cannot be confirmed. The following procedure is used to check for image duplication and the uploading process:
- File Encryption
- Generation of Hash key
- Searching for duplicates
- File shared with the user

5 (b-i). Pseudo Code of the Proposed Framework

1. File_Tag ← File
2. Token_Req ← (Tag, UserID)
3. DupCheckReq ← Token
4. ShareTokenReq ← (Tag, {Priv.})
5. FileEncrypt ← File
6. FileUploadReq ← (FileID, File, Token)
7. TokenGen ← (Tag, UserID)

5 (b-ii). System Model of Proposed Scheme

Figure 11. shows the proposed system model of the cloud environment. It contains Client (C), key generation, and Cloud Storage (CS), where the data de-duplication with CE occurs. The process is explained in the given steps below:

- First, a token Tkn is generated from the uploaded client data (C_D) using T_{KN} = Tkn_Gen(C_D) primitive function.
- The token T_{kn} is then forwarded to the key generation to get a key for generating CEk through a secure channel.
- KeyGen has then verified the metadata for the existence of Tkn in the database. In case of the presence of T_{KN} in the database, KeyGen will forward the key to the user corresponding to the same Tkn. In case of its non-existence in the database, a key (GCK) is generated and sent to the user by KeyGen.
- The user then uses this forwarded key GCK by KGMaaS for generating convergent encryption key CEK.

- Clients generate the CEk using the proposed algorithm $CE_K = CE_{keyGen}(C_k)$ with the key GCK received from the KeyGen.

Table 4 shows the data classification based on primary storage and description.

Table 4
Data Classification in the Proposed Framework

Classification	Basic Capacities	Description
Data Classification in the proposed framework (Bellare, Keelveedhi, and Ristenpart, 2013 [64])	KeyGenSE (1)! *K*	"This is an algorithm used to develop *κ* through parameter 1".
	EncSE (*κ, M*)! *C*	"This algorithm is responsible for hiding the secret of *κ* and *M*, respectively. It also outputs the coded text *C"*.
	DecSE (*κ, C*)! *M*	"Unlike the other algorithms, DecSE (*κ, C*)! *M* is used for decryption. It is used to take the secret *κ* and coded text *C"*.
	KeyGenCE (M)! K	"This is a key generation algorithm. It assists in illustrating data copy M that is convergent to K".
	EncCE (K, M)! C	"This is the symmetric encryption system."
	DecCE (K, C)! M	"This is the decryption algorithm."
	TagGen (M)! T (M)	"This is the tag generation algorithm representing the original data copy M."

The various UML diagrams, including the class diagram, use case diagram, activity diagram, and sequence diagram, are shown in Figs. 12, 13, 14, and 15, have been used to demonstrate the effectiveness of the suggested scheme.

5 (C). Experiments And Analysis Of The Suggested Approach

The experimental results and their analysis are reported in this section. Java was used to accomplish the suggested plan on Amazon EC2 servers with Intel i7 preprocessors. Using data frequently received on mobile devices, we chose eight different datasets (DS 1, DS 2, DS 3,..., DS 6). DS 1 consists of text files (.text). DS 2 consists of Java application data (.java), DS 3 consists of CSS application data (.css), DS 4 consists of HTML application data (.html), DS 5 consists of JavaScript Application data (.jss), DS 6 consists of React application data (.jsx). The size of data blocks before de-duplication was taken to be 2.08 KB, 1.80 KB, 1.55 KB, 2.50 KB, 2.66 KB, and 1.66 KB (as shown by Table 5). By applying the proposed approach of data de-duplication on these DS, the size of data blocks becomes 1.90 KB, 1.60

KB, 1.30 KB, 2 KB, 2.20 KB, and 1.20 KB (as shown in Table 6). The files can be broken into size blocks to make share management easier; however, this will reduce the file's de-duplication rate.

Table 5
Size of files before De-duplication

Data Sets	File Name	File Extension	File Size in KB
DS 1	data	.text	2.08
DS 2	abc	.java	1.80
DS 3	boot	.css	1.55
DS 4	abc	.html	2.55
DS 5	script	.js	2.66
DS 6	script	.jsx	1.66

Table 6
Size of files after De-duplication

Data Sets	File Name	File Extension	File Size in KB	Saved Space in KB
DS 1	data	.text	1.90	0.18
DS 2	abc	.java	1.60	0.20
DS 3	boot	.css	1.30	0.25
DS 4	abc	.html	2.00	0.55
DS 5	script	.js	2.20	0.46
DS 6	script	.jsx	1.20	0.46

Table 5 showed the outcomes of de-duplication when different dataset sizes were employed. The size of DS 1 was reduced by around 9%, the size of DS 2 was reduced by about 13%, the size of DS 3 was reduced by about 18%, the size of DS 4 was reduced by about 22%, the size of DS 5 was reduced by about 19%, and the size of DS 6 was reduced by about 29%.

We can save cloud storage space using a de-duplication strategy on these files.

5 (D). Security Analysis Of The Proposed Scheme

This section examines the security of the proposed scheme before comparing its functionality and effectiveness to other related works. The suggested scheme is compared to existing schemes in Table 7

based on the de-duplication difficulties the system provides.

Table 7
Comparative security analysis of the proposed scheme

De-duplication Issues	Bellare et al.	Halevi et al.	Koo, D	Ng et al.	Xu et al.	Proposed Scheme
Encrypted data de-duplication	√	x	√	x	√	√
Tags for preserving consistency	√	x	√	x	√	√
Updating data that has been outsourced	x	x	√	x	x	√
De-duplication on the client side	√	x	√	x	√	√
Repeated heterogenous files	x	x	x	x	x	√

6. Conclusion And Future Research Direction

The cloud uses convergent encryption to erase duplicated user data and copy the backing for data duplication. A CEKGA is proposed, which generates the convergent key by employing a key with the user's information. The AWS cloud platform is used to simulate the proposed work. This proposed scheme effectively manages the generation and maintenance of keys and reduces the burden on the users. CEKGA effectively executes the generation of key CE_K through GCK. The analysis of the proposed scheme will ensure the security of CKk and the de-duplication of stored data in cloud storage. A secure data de-duplication scheme facilitating a thorough comparative analysis of various CSPs concerning the security controls incorporated has been taken up. Various data de-duplication techniques have been compared based on their different levels of security. A secure convergent encryption algorithm-based block-level de-duplication technique has been proposed and implemented, enhancing the entire process's security and catering to repeated cycle problems presented in this paper. This study also comprehensively analyzes and explains the strategies of data de-duplication. We have also reviewed several brand-new surveys exploring related topics in depth. Studies done recently in this area have only looked at storage-based de-duplication methods. The first insight gained from this survey is the discovery of de-duplication methods mostly targeted at text and multimedia. According to the review, de-duplication presents several difficult situations that can be fully resolved using textual and multimedia resources. In the future, the proposed algorithm will be implemented on a cloud platform and compared with other currently available methods. This paper will help researchers and academicians identify de-duplication techniques and propose further improvements to secure data de-duplication.

Declarations

Ethics approval and consent to participate Not applicable.

Consent for publication All authors have read and agreed to the published version of the manuscript.

Availability of data and materials Not applicable

Competing interests The authors declare that they have no competing interests.

Funding Not applicable.

Authors' contributions Conceptualization, SA, and SM; data curation, SA and SM; formal analysis, SM, and IS; investigation, SM, and IS; methodology, SA, and SM.; resources, SA and IS; supervision SM; validation, SM and IS; visualization, SM; writing—original draft, SA; writing—review and editing, SM, and IS.

Acknowledgments The authors acknowledge the financial support received own, for their support and encouragement in carrying out his college work. The authors also would like to acknowledge the administration of Jamia Millia Islamia, which the authors represent.

References

1. Ahmad RW, Gani A, Ab. Hamid SH, M. Shiraz, Feng Xia, & S. Madan (2015). Virtual machine migration in cloud data centers: a review, taxonomy, and open research issue. *Journal of Supercomputing 71 (7):*2473–2515. https://doi.org/10. 1007/s11227-015-1400-5.
2. Ahmad, S., Mehfuz, S., Mebarek-Oudina, F. et al. RSM analysis-based cloud access security broker: a systematic literature review. Cluster Comput 25, 3733–3763 (2022). https://doi.org/10.1007/s10586-022-03598-z
3. Ali, M., Khan, S. U., & Vasilakos, A. V. (2015). Security in cloud computing: Opportunities and challenges. *Information Sciences*, *305*, 357–383. https://doi.org/10.1016/j.ins.2015.01.025.
4. Alvarez C. (2011). NetApp deduplication for FAS and V-Series Deployment and implementation guide. *In: Technical ReportTR-3505*.
5. Ashish Agarwala, Priyanka Singh & Pradeep K. Atrey. (2017, October). *DICE: A Dual Integrity Convergent Encryption Protocol for Client-Side Secure Data Deduplication*. IEEE International Conference on Systems, Man, and Cybernetics (SMC) pp 2176-2181.
6. B. Gupta, M. Negi, K. Vishwakarma, G. Rawat & P. Badhani. (2017). Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python, *International Journal of Computer Applications* 165(9), 29–34. https://dx.doi.org/10.5120/ijca2017914022.
7. Bai, J., Yu, J., Gao, X., (2020). Secure auditing and deduplication for encrypted cloud data supporting ownership modification. *Soft Computing*, 24, 12197–12214 (2020). https://doi.org/10.1007/s00500-019-04661-5.
8. Banu AF, & Chandrasekar C. (2012). A survey on deduplication methods. *Int J Computer Trends Technol* 3(3), 364–368.
9. Barreto J, & Ferreira P. (2009, November). *Efficient locally trackable deduplication in replicated systems*. In: Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware.

Springer-Verlag New York, Inc. USA, p 103-122.

10. Bellare, M., Keelveedhi, S., & Ristenpart, T. (May, 2013). *Message-locked encryption and secure deduplication*. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 296–312, Springer, Berlin, Heidelberg.
11. Bhadade US, & Trivedi AI. (2011). Lossless text compression using dictionaries. *Int J Comput Appl Algorithms* 13(8), 27–34. DOI:10.5120/1799-1767.
12. Borges EN, de Carvalho MG, Galante R, Gonçalves MA, & Laender AH. (2011). An unsupervised heuristic-based approach for bibliographic metadata deduplication. *Information Processing & Management* 47(5), 706– 718. https://doi.org/10.1016/j.ipm.2011.01.009
13. Chen CP, Zhang CY. (2014). Data-intensive applications, challenges, techniques, and technologies: a survey on big data. *Information Sciences* 275, 314–347. https://doi.org/10.1016/j.ins.2014.01.015.
14. Cheng Guo, Xueru Jiang, Kim-Kwang Raymond Choo, & Yingmo Jie. (2020). R-Dedup: Secure client-side deduplication for encrypted data without involving a third-party entity, Journal of Network and Computer Applications,162, 102664. https://doi.org/10.1016/j.jnca.2020.102664.
15. Clements AT, Ahmad I, Vilayannur M, & Li J. (2009, June). *Decentralized Deduplication in SAN Cluster File Systems*. In: USENIX Annual Technical Conference, pp 101–114, San Diego, California, US.
16. Cyber Security Breaches Report of Black Hat Ethical Hacking (2019). https://www.blackhatethicalhacking.com.
17. Di Pietro R, & Sorniotti A. (2016). Proof of ownership for deduplication systems: a secure, scalable, and efficient solution. *Computer Communications 82*, 71–82. https://doi.org/10.1016/j.comcom.2016.01.011.
18. Duan, Y., (2014, November). *Distributed key generation for encrypted deduplication: Achieving the strongest privacy*, CCSW '14: Proceedings of the 6th edition of the ACM Workshop on Cloud Computing Security, pp. 57–68.
19. DuBois, L., Amaldas, M., & Sheppard, E., (2011). Key considerations as deduplication evolve into primary storage. White Paper 223310.
20. Geeta C M, Shreyas Raju R G, Raghavendra S, Rajkumar Buyya, Venugopal K R, S S Iyengar, L M Patnaik. (December, 2020). *SDVADC: Secure Deduplication and Virtual Auditing of Data in the Cloud.* Procedia Computer Science, Third International Conference on Computing and Network Communications (CoCoNet'19), 171, pp- 2225-2234, Trivandrum, Kerala, India.
21. Gu M, Li X, & Cao Y (2014). Optical storage arrays: a perspective for future big data storage. *Light Science & Applications* 3(5), e177. https://doi.org/10.1038/lsa.2014.58.
22. Guipeng Zhang, Zhenguo Yang, Haoran Xie, & Wenyin Liu. (2021). A securely authorized deduplication scheme for cloud data based on blockchain. *Information Processing & Management*, 58, Issue 3, 102510. https://doi.org/10.1016/j.ipm.2021.102510.
23. Halevi, S., Harnik, D., Pinkas, B., and Shulman-Peleg, A. (October, 2011). *Proofs of ownership in remote storage systems*. In Proceedings of the 18th ACM conference on Computer and

communications security, pp. 491–500, ACM.

24. He Q, Li Z, & Zhang X. (2010, September). *Data deduplication techniques*. IEEE Int Conf Future Inf Technol Manag Eng (FITME) 1, 430–433. https://doi.org/10.1109/FITME.2010.5656539.
25. He, Y., Xian, H., Wang, L., & Zhang, S. (2020). Secure encrypted data deduplication based on data popularity. *Mobile Networks and Applications*, 1–10. https://doi.org/10.1007/s11036-019-01504-3.
26. Hovhannisyan H, Qi W, Lu K, Yang R, & Wang J. (2016). Whispers in the cloud storage: a novel cross-user deduplication-based covert channel design. *Peer-to-Peer Networking and Applications*, 11, pages277–286. https://doi.org/10.1007/s12083-016-0483-y.
27. Hu Y, Li C, Liu L, & Li T. (2016, June). *Hope: Enabling Efficient Service Orchestration in Software-Defined Data Centers*. In: Proceedings of the 2016 International Conference on Supercomputing, p. 1-12 ACM, Istanbul, Turkey.
28. Huynh-Thu, Q., & Ghanbari, M. (2008). Scope of Validity of PSNR in Image/Video Quality Assessment. *Electronics Letters*, 44(13), pp. 800-801, IEEE 2008. https://doi.org/10.1049/el:20080522.
29. IDC REPROT ON EXPONENTIAL DATA Gantz J, Reinsel D. (2012). *The digital universe in 2020: big data, bigger digital shadows, and biggest growth in the far east.* In: IDC iView: IDC Analyze the Future, pp. 1–6. http://www.emc.com/collateral/analyst-reports/idc-digital-universe-unitedstates. pdf
30. J. Amalraj & J.R. Jose. (2016). A survey paper on cryptography techniques. *International Journal of Computer Science and Mobile Computing,* 5(8), 55–59.
31. Jin Li, Xiaofeng Chen, Mingqiang Li, Jingwei Li, Patrick P.C. Lee, & Wenjing Lou. (2014). Secure Deduplication with Efficient and Reliable Convergent Key Management. *IEEE Transactions on Parallel and Distributed Systems,* **25**6 pp. 1615-1625. doi:10.1109/TPDS.2013.284.
32. K. Akhila, A. Ganesh & C. Sunitha. (2016). A Study on Deduplication Techniques over Encrypted Data. *Procedia Computer Science*, 87(3), 38–43. doi: 10.1016/j.procs.2016.05.123.
33. Keelveedhi, S., Bellare, M., & Ristenpart, T. (2013, August). *Dupless: server-aided encryption for deduplicated storage*. In: 22nd USENIX Security Symposium Security 13, pp. 179–194, Washington D.C.
34. Kim C, Park KW, & Park KH (2012, February). GHOST: *GPGPU-offloaded high-performance storage I/O deduplication for the primary storage system*. In: Proceedings of the 2012 International Workshop on Programming Models and Applications for Multicores and Manycores, ACM, pp. 17–26. https://doi.org/10.1145/2141702.2141705.
35. Koo, D. & Hur, J. (2018). Privacy-preserving deduplication of encrypted data with dynamic ownership management in fog computing. Future Generation Computer Systems, 78, pp. 739–752. https://doi.org/10.1016/j.future.2017.01.024.
36. Lee, D., & Park, N. (2020). Blockchain-based privacy-preserving multimedia intelligent video surveillance using secure Merkle tree. *Multimedia Tools and Applications*, 1–18. https://doi.org/10.1007/s11042-020-08776-y.

37. Li YK, Xu M, Ng CH, & Lee PP. (2015). Efficient hybrid inline and out-of-line deduplication for backup storage. *ACM Trans Storage (TOS),* 11(1), 1–21. https://doi.org/10.1145/2641572.
38. Li, J., Chen, X., Li, M., Li, J., Lee, P.P., & Lou, W. (2014). Secure deduplication with efficient and reliable convergent key management. *IEEE Transactions on Parallel and Distributed Systems,* 25, pp. 1615–1625. DOI: 10.1109/TPDS.2013.284.
39. Li, J., Chen, X., Xhafa, F., & Barolli, L. (2015). Secure deduplication storage systems supporting keyword search.
Journal of Computer and System Sciences, 81, pp. 1532–1541.
https://doi.org/10.1016/j.jcss.2014.12.026.
40. Li, J., Lee, P.P., Tan, C., Qin, C., & Zhang, X. (2020). Information leakage in encrypted deduplication via frequency analysis: Attacks and defenses. *ACM Trans. Storage (TOS),* 16, 1–30.
https://doi.org/10.1145/3365840.
41. Li, J., Yang, Z., Ren, Y., Lee, P.P., & Zhang, X. (2020, April). *Balancing storage efficiency and data confidentiality with tunable encrypted deduplication*. In: Proceedings of the Fifteenth European Conference on Computer Systems, pp. 1–15, Heraklion, Greece. ACM, New York, NY, USA.
42. Liu, J., Wang, J., Tao, X., and Shen, J. (2017). Secure similarity-based cloud data deduplication in the ubiquitous city. Pervasive and Mobile Computing pages 231–242.
43. Lillibridge M, Eshghi K, Bhagwat D, Deolalikar V, Trezise G, & Camble P. (2009, February). *Sparse Indexing: Large Scale, Inline Deduplication Using Sampling and Locality*. In Proceedings of the 7th USENIX Conference on File and Storage Technologies, vol 9, pp 111–123, San Francisco.
44. Liu, X., Lu, T., He, X., Yang, X., & Niu, S. (2020). Verifiable attribute-based keyword search over encrypted cloud data supporting data deduplication. IEEE Access, 8, 52062–52074. Doi: 10.1109/ACCESS.2020.2980627.
45. Maan AJ. (2013). Analysis and comparison of algorithms for lossless data compression. *Int J Inf Comput Technol*, 3(3), 139–46.
46. Mandagere N, Zhou P, Smith MA, & Uttamchandani S. (2008, December). Demystifying data deduplication. In: Proceedings of the ACM/IFIP/USENIX Middleware'08 Conference Companion, pp. 12–17. https:// doi.org/10.1145/1462735.1462739.
47. Mao B, Jiang H, Wu S, & Tian L. (2016). Leveraging data deduplication to improve the performance of primary storage systems in the cloud. *IEEE Trans Comput.,* 65(6), 1775–1788. https://doi.org/10.1109/TC.2015.2455979.
48. Mao B, Jiang H,Wu S, Fu Y, & Tian L. (2014). Read-performance optimization for deduplication-based storage systems in the cloud. In: *ACM Transactions on Storage (TOS),* vol 10(2). https://doi.org/10.1145/2512348.
49. Mell, P., Grance, T. (2011). *The NIST definition of cloud computing*. NIST SP 800-145, The NIST Definition of Cloud Computing.
50. Meyer, D.T., Bolosky, & W.J. (2012). A study of practical deduplication. *ACM Trans Storage (ToS)*, 7, 1–20. https://doi.org/10.1145/2078861.2078864.

51. N. vurukonda and B.T. Rao. (2016). A Study on Data Storage Security Issues in Cloud Computing, *Procedia Computer Science,* 92, pp. 128–135. https://doi.org/10.1016/j.procs.2016.07.335.
52. Nayak, S.K., & Tripathy, S. (2020). Seds: secure and efficient server-aided data deduplication scheme for cloud storage. *Int. J. Inform. Security,* 19, pp. 229–240. https://doi.org/10.1007/s10207-019-00455-w.
53. Ng CH, MaM, Wong TY, Lee PP, & Lui J. (2011, December). *Live deduplication storage of virtual machine images in an open-source cloud*. In: Proceedings of the 12th International Middleware Conference. International Federation for Information Processing, pp 80–99*,* Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-25821-3_5.
54. Ng, W. K., Wen, Y., & Zhu, H. (March, 2012). Private data deduplication protocols in cloud storage. *In Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pp. 441–446. ACM.
55. Ni, J., Zhang, K., Yu, Y., Lin, X., & Shen, X.S. (2018). Providing task allocation and secure deduplication for mobile crowdsensing via fog computing. *IEEE Trans. Dependable Secure Computing*. DOI: 10.1109/TDSC.2018.2791432.
56. Nyo, M.T., Mebarek-Oudina, F., Hlaing, S.S. et al. Otsu's thresholding technique for MRI image brain tumor segmentation. Multimed Tools Appl (2022). https://doi.org/10.1007/s11042-022-13215-1
57. P. Puzio, R. Molva & M. Onen, & S. Loureiro (2013, December). *ClouDedup: Secure Deduplication with Encrypted Data for Cloud Storage*. 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, pp. 363–370, Bristol, UK.
58. Paulo J, & Pereira J. (2014). A survey and classification of storage deduplication systems. *ACM Computing Surveys,* 47(1), 1–30. https://doi.org/10.1145/2611778.
59. Paulo J, Pereira J. (2014). Distributed Exact Deduplication for Primary Storage Infrastructures. In Magoutis K., Pietzuch P. (eds) *Distributed applications and interoperable systems*, DAIS 2014, vol 8460, LNCS Springer, Heidelberg. https://doi.org/10.1007/978-3-662-43352-2_5.
60. Prajapati, P., & Shah, P. (2014, April). *Efficient cross-user data deduplication in remote data storage*. In: International Conference for Convergence for Technology-2014. IEEE, pp. 1–5, Pune, India.
61. Prajapati, P., Shah, P., Ganatra, A., & Patel, S. (2017). Efficient cross-user client-side data deduplication in Hadoop. *Journal of Computers,* 12, 362–370. DOI: 10.17706/jcp.12.4.362-370.
62. Premkamal, P.K., Pasupuleti, S.K., Singh, & A.K., Alphonse, P. (2021). Enhanced attribute-based access control with secure deduplication for big data storage in the cloud. *Peer-to-Peer Networking and Applications*, 14, pages102–120. https://doi.org/10.1007/s12083-020-00940-3.
63. R. Raghatate, S. Humne, & R. Wadhwe. (2014). A Survey on Secure Cloud Computing using AES Algorithm, *International Journal of Computer Science and Mobile Computing,* 3, 295301–295301.
64. Rahumed, A., Chen, H.C., Tang, Y., Lee, P.P., & Lui, J.C. (2011, September). *A secure cloud backup system with assured deletion and version control*. In: 2011 40th International Conference on Parallel Processing Workshops. IEEE, pp. 160–167.

65. R. D. Labati, A. Genovese, V. Piuri, F. Scotti, & S. Vishwakarma (2020). Computational Intelligence in Cloud Computing. *Recent Advances in Intelligent Engineering, Topics in Intelligent Engineering and Informatics*, vol 14. Springer, Cham. https://doi.org/10.1007/978-3-030-14350-3_6.
66. Scanlon, M. (2016, August). *Battling the digital forensic backlog through data deduplication*. In: 2016 Sixth International Conference on Innovative Computing Technology (INTECH). IEEE, pp. 10–14, Dublin Ireland.
67. Shanmugasundaram S, & Lourdusamy R. (2011). A comparative study of text compression algorithms. *Int J Wisdom Based Computer,* 1(3), 68–76. Doi: 10.21917/ijct.2011.0062.
68. Shen, W., Su, Y., & Hao, R. (2020). Lightweight cloud storage auditing with deduplication supporting strong privacy protection. *IEEE Access,* 8, 44359–44372. DOI: 10.1109/ACCESS.2020.2977721.
69. Shin, Y., Koo, D., Yun, J., & Hur, J. (2017). Decentralized server-aided encryption for secure deduplication in cloud storage. *IEEE Trans. Services Computing*. DOI: 10.1109/TSC.2017.2748594.
70. Singh, P., Agarwal, N., & Raman, B. (2016, December). *Don't see me, just filter me: towards secure cloud-based filtering using Shamir's secret sharing and pob number system*. In Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing, pp. 1-8. ACM.
71. Singh, P., Raman, B., Agarwal, N., & K. Atrey, P. (2017b). Secure cloud-based image tampering detection and localization using a pob number system. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 13:1–23. https://doi.org/10.1145/3077140.
72. Srinivasan K, Bisson T, Goodson GR, & Voruganti K. (2012, December). *iDedup: latency-aware, inline data deduplication for primary storage*. In: Proceedings of the USENIX Conference on File and Storage Technologies, vol 12, pp 24–24, San Jose, CA.
73. Stanek, J., Sorniotti, A., Androulaki, E., & Kencl, L. (2014, November). *A secure data deduplication scheme for cloud storage*. International conference on financial cryptography and data security. Springer, pp. 99–118, Berlin, Heidelberg.
74. Storer, M.W., Greenan, K., Long, D.D., & Miller, E.L. (2008, October). *Secure data deduplication*. In: Proceedings of the 4th ACM international workshop on Storage security and survivability, pp. 1–10.
75. T. Jiang, X. Chen, Q. Wu, J. Ma, W. Susila, & W. Lou (2016). Secure and efficient cloud data deduplication with randomized tags. *IEEE transactions on information forensics and security* 12, pp. 532–543. DOI: 10.1109/TIFS.2016.2622013.
76. Taek-Young Youn, Ku-Young Chang, Kyung Hyune Rhee, & Sang Uk Shin2 (2016, June). Authorized convergent encryption for client-side deduplication. IT CoNvergencePRActice (INPRA)**, 4** 2 pp. 9-17.
77. Tian Y, Khan SM, Jiménez DA, & Loh GH. (2014, June). *Last-level cache deduplication*. In: Proceedings of the 28th ACM International Conference on Supercomputing, pp 53–62. https://doi.org/10.1145/ 2597652.2597655.
78. Umberto Martinez-Penas (2018). Communication Efficient and Strongly Secure Sharing Schemes based on Algebraic Geometry codes. *IEEE Transactions on Information Theory*. 64(6), pp. 4191-4206, April 2018. Doi: 10.1109/TIT.2018.2823326

79. Venish A, & Sankar KS. (2015). The framework of data deduplication: a survey. *Indian J Sci Technol, 8, 26, pp. 1-7*. https:// doi.org/10.17485/ijst/2015/v8i26/80754.
80. Wang J, & Chen X. (2016). Efficient and secure storage for outsourced data: a survey. *Data Sci Eng*, 1(3), pp. 178–188. https://doi.org/10.1007/s41019-016-0018-9.
81. Wang, Y., Cui, Y., Huang, Q., Li, H., Huang, J., & Yang, G. (2020). Attribute-based equality test over encrypted data without random oracles. *IEEE Access,* 8, pp. 32891–32903. DOI: 10.1109/ACCESS.2020.2973459.
82. Witten IH, Neal RM, & Cleary JG. (1987). Arithmetic coding for data compression. *Commun ACM*, 30(6):520–40. https://doi.org/10.1145/214762.214771.
83. Xia W, Jiang H, Feng D, & Hua Y. (2015). Similarity and locality-based indexing for high-performance data deduplication. *IEEE Trans Comput.,* 64(4), 1162–1176. https://doi.org/10.1109/TC.2014.2308181.
84. Xia W, Jiang H, Feng D, Tian L, Fu M, Zhou Y. (2014). Ddelta: a deduplication-inspired fast delta compression approach. *Perform Eval*, 79, pp. 258–272. https://doi.org/10.1016/j.peva.2014.07.016.
85. Xiang Gao, Jia Yu, Wen-Ting Shen, Yan Chang, Shi-Bin Zhang, Ming Yang, & Bin Wu (2021). Achieving low-entropy secure cloud data auditing with file and authenticator deduplication. *Information Sciences*, 546, pp- 177-191. https://doi.org/10.1016/j.ins.2020.08.021.s
86. Xia. W, Jiang H, Feng D, Douglis F, Shilane P, HuaY, Fu M, ZhangY, & ZhouY. (2016). A comprehensive study of the past present and future of data deduplication. *Proc IEEE* 104(9), pp. 1681–1710. https://doi. org/10.1109/JPROC.2016.2571298.
87. Xu, J., Chang, E.-C., & Zhou, J. (May, 2013). Weak leakage-resilient client-side deduplication of encrypted data in cloud storage*. In Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pp. 195–206. ACM.
88. Xu J, ZhangW, Zhang Z, Wang T, & Huang T. (2016). Clustering-based acceleration for virtual machine image deduplication in the cloud environment. *J Syst Softw*, 121, pp. 144–156. https://doi.org/10.1016/j. jss.2016.02.021
89. Y. Peng, W. Zhao, F. Xie, Z.-h. Dai, Y. Gaao, D.-q. Chen. (2012), Secure cloud storage based on cryptographic techniques. *The Journal of China Universities of Posts and Telecommunications,* 19(2), pp. 182–189. doi:10.1016/s1005- 8885(11)60424-x.
90. Yin, J., Tang, Y., Deng, S., Bangpeng, Z., & Zomaya, A., (2020). Muse: A multi tierd and sla-driven deduplication framework for cloud storage systems. *IEEE Trans. Computers*, pp. 759-774. DOI: 10.1109/TC.2020.2996638
91. Yuan, H., Chen, X., Wang, J., Yuan, J., Yan, H., & Susilo, W. (2020). Blockchain-based public auditing and secure deduplication with fair arbitration*. Information Sciences*, 541, pp. 409-425. https://doi.org/10.1016/j.ins.2020.07.005.
92. Yunling Wang, Meixia Miao, Jianfeng Wang, & Xuefeng Zhang (2021). Secure deduplication with efficient user revocation in cloud storage. *Computer Standards and Interfaces*, 78, 103523. https://doi.org/10.1016/j.csi.2021.103523.

93. Zhang, Y., Yuan, Y., Feng, D., Wang, C., Wu, X., Yan, L., Pan, D., & Wang, S. (2020). Improving restore performance for an in-line backup system combining deduplication and delta compression. *IEEE Trans. Parallel Distributed Syst*. 31, 2302–2314. DOI: 10.1109/TPDS.2020.2991030.
94. Zhang, Y., Yuan, Y., Feng, D., Wang, C., Wu, X., Yan, L., Pan, D., & Wang, S. (2020a). Improving restore performance for an in-line backup system combining deduplication and delta compression. *IEEE Trans. Parallel Distributed Syst*. 31, 2302–2314. DOI: 10.1109/TPDS.2020.2991030.
95. Zhao X, Zhang Y, Wu Y, Chen K, Jiang J, & Li K. (2013). Liquid: a scalable deduplication file system for virtual machine images. *IEEE Trans Parallel Distrib Syst*, 25(5), pp. 1257–1266. https://doi.org/10.1109/ TPDS.2013.173.
96. Zheng Yan., Wenxiu Ding., Xixun Yu., & Haiqi Zhu. (2016). Robert H Deng. Deduplication on Encrypted Big Data in Cloud. *IEEE Transactions on Big Data*, 2(2), pp. 138-150. doi:10.1109/TBDATA.2016.2587659.
97. Zheng, Q., & Xu, S. (2012, February). *Secure and efficient proof of storage with deduplication*. Proceeding of the second ACM conference on Data and Application Security and Privacy, pp. 1-12.
98. Zhou R, Liu M, & Li T. (2013, September). *Characterizing the efficiency of data deduplication for big data storage management*. In: IEEE International Symposium on workload Characterization (IISWC), pp 98–108, Portland, OR, USA.
99. Zhu B, Li K, & Patterson RH. (2008). Avoiding the disk bottleneck in the data domain deduplication file system*. Proc USENIX Conf File Storage Technol*, 8, pp. 1–14. DOI/10.5555/1364813.1364831.

Figures

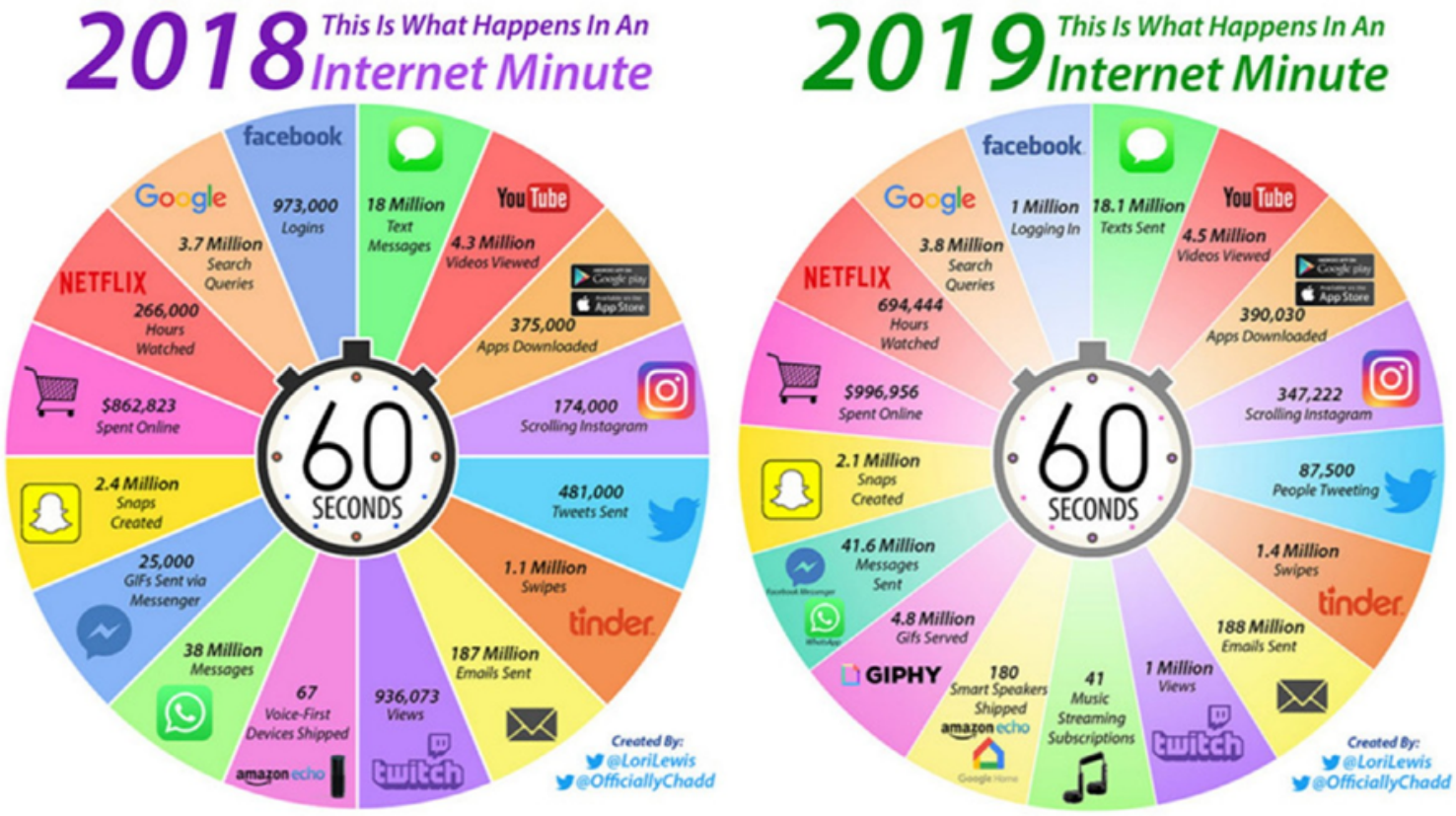

Figure 1

Infographic of a Minute on the Internet (2018, 2019)

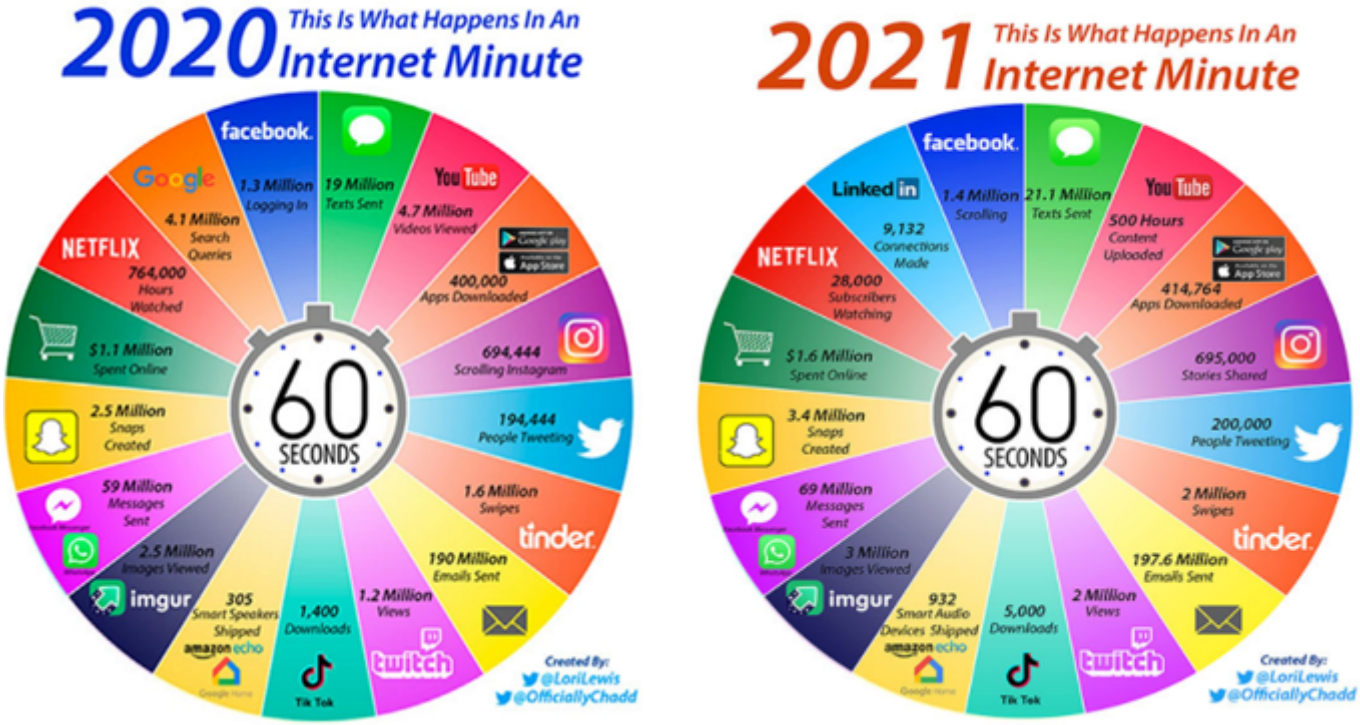

Figure 2

Infographic of a Minute on the Internet (2020, 2021)

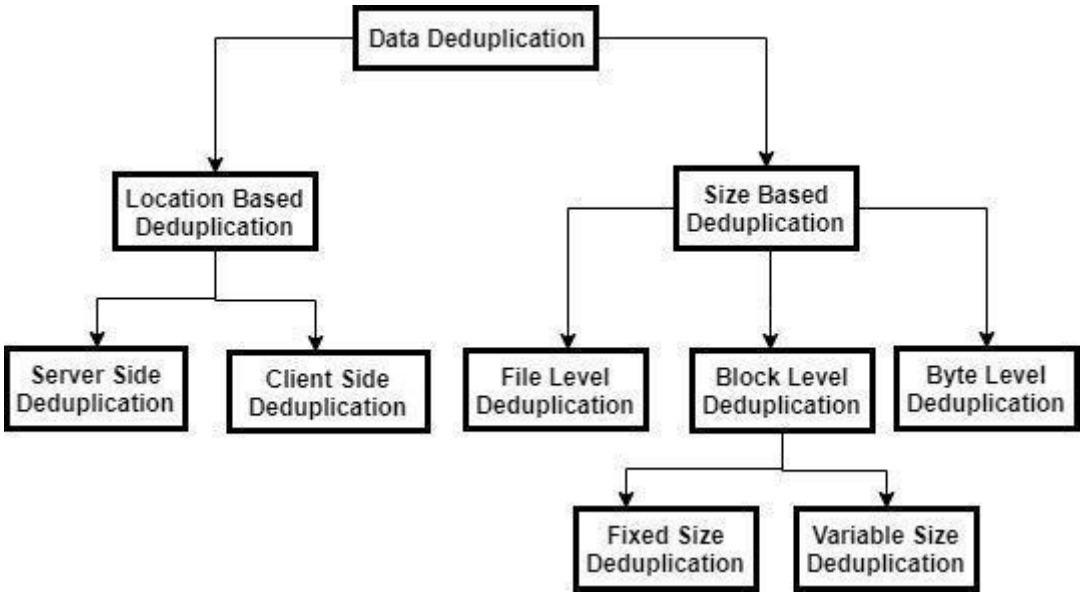

Figure 3

Classification of Data Deduplication

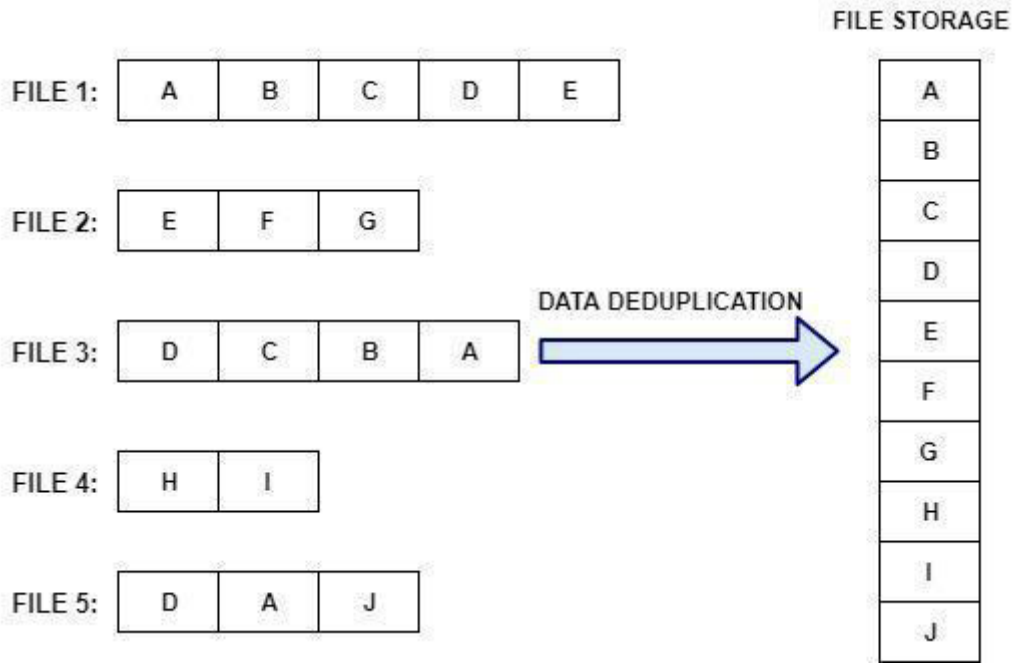

Figure 4

Data De-duplication for File Storage

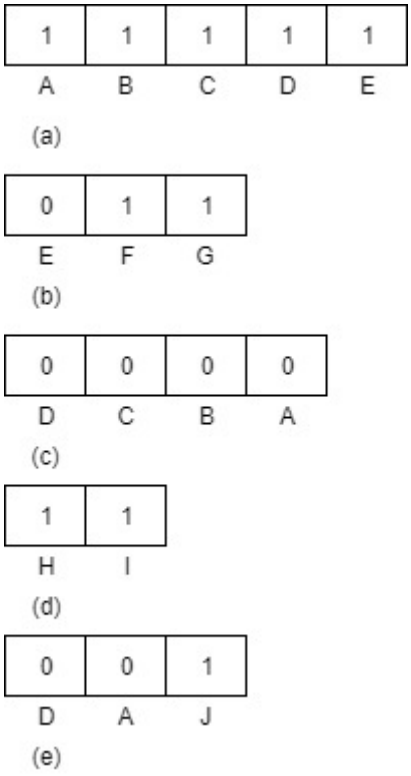

Figure 5

Conversions into vectors: Files 1 (a), 2 (b), 3 (c), 4 (d), and File 5 (e).

	A	B	C	D	E	F	G	H	I	J
A	1	0	0	0	0	0	0	0	0	0
B		1	0	0	0	0	0	0	0	0
C			1	0	0	0	0	0	0	0
D				1	0	0	0	0	0	0
E					1	0	0	0	0	0
F						1	0.3	0.2	0.1	0
G							1	0.6	0.5	0.4
H								1	0.8	0.7
I									1	0.9
J										1

Figure 6

Data Deduplication Comparison between Files

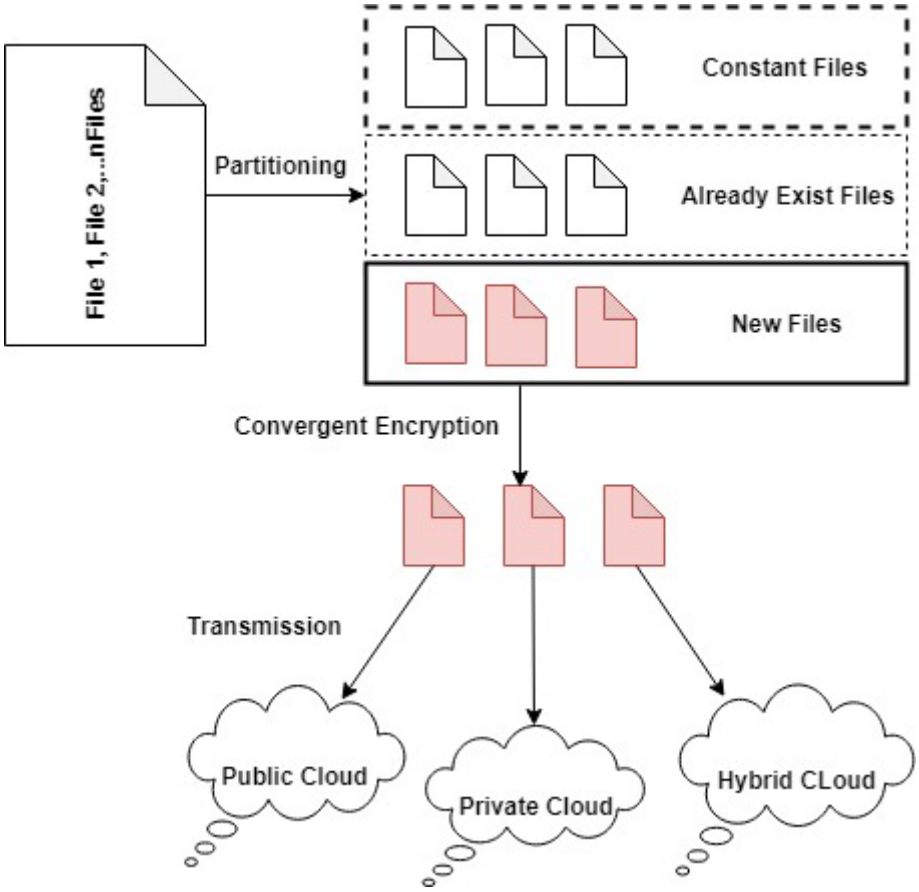

Figure 7

Secured Cloud Data Deduplication Overview

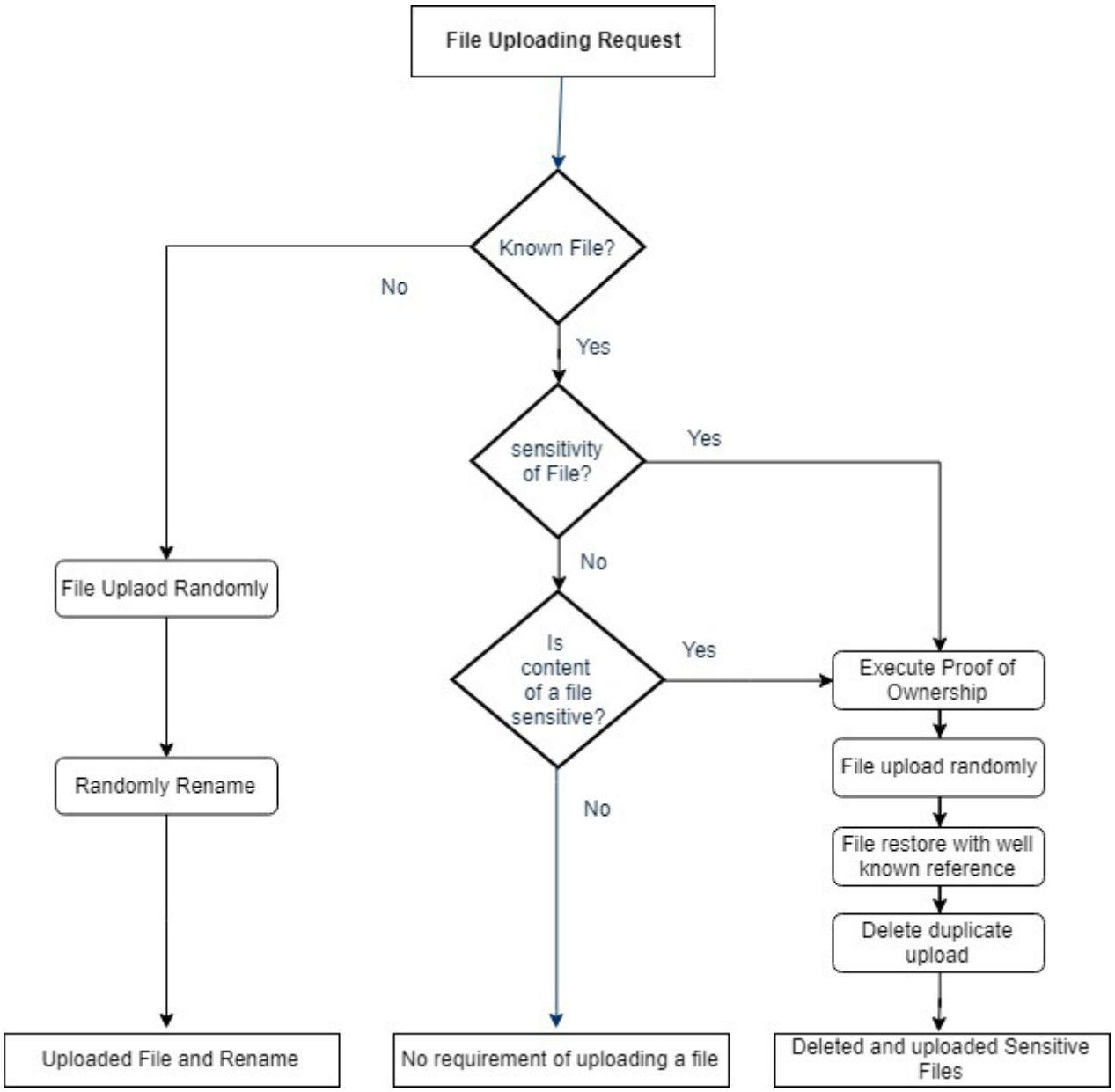

Figure 8

Flowchart of DAIC

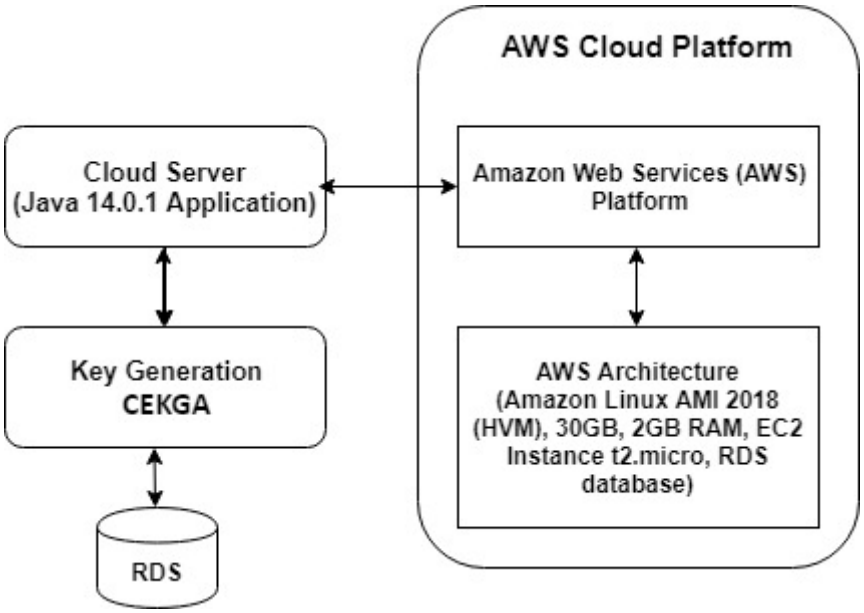

Figure 9

Block Diagram of Simulation Cloud Environment of CEKGA

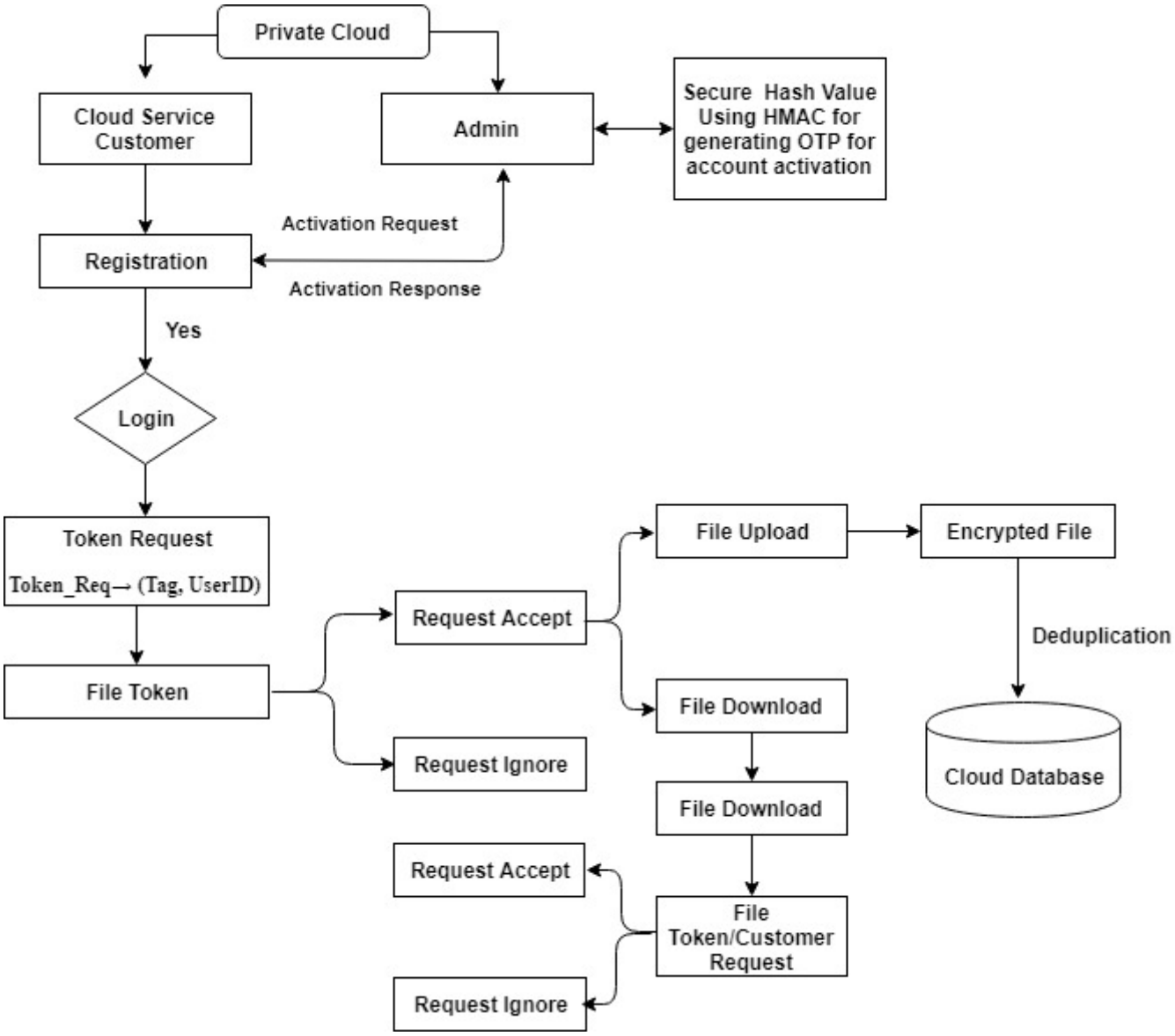

Figure 10

Block Diagram of the Suggested Framework

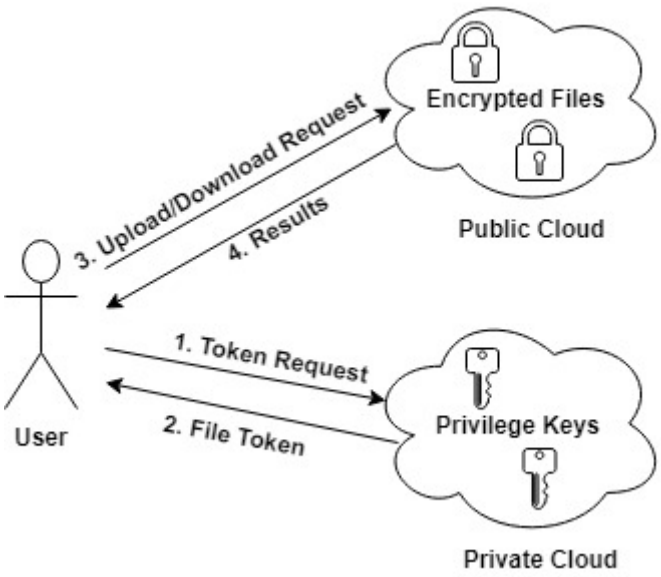

Figure 11

Block Diagram of de-duplication based on the cloud environment

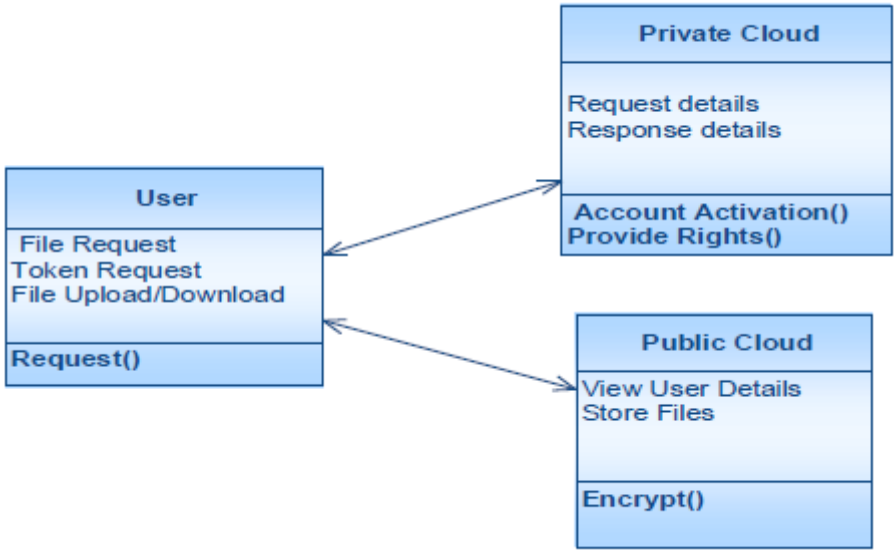

Figure 12

Class Diagram of cloud environment

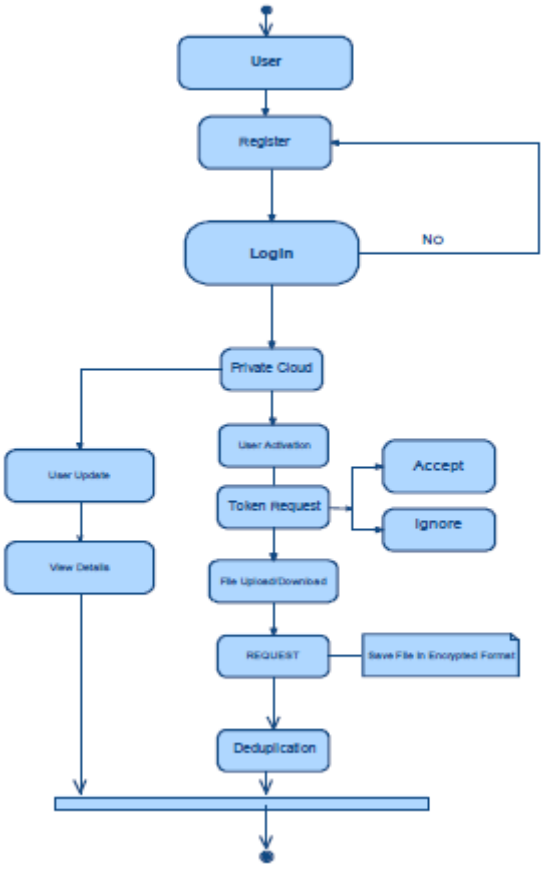

Figure 13

Activity Diagram of data de-duplication

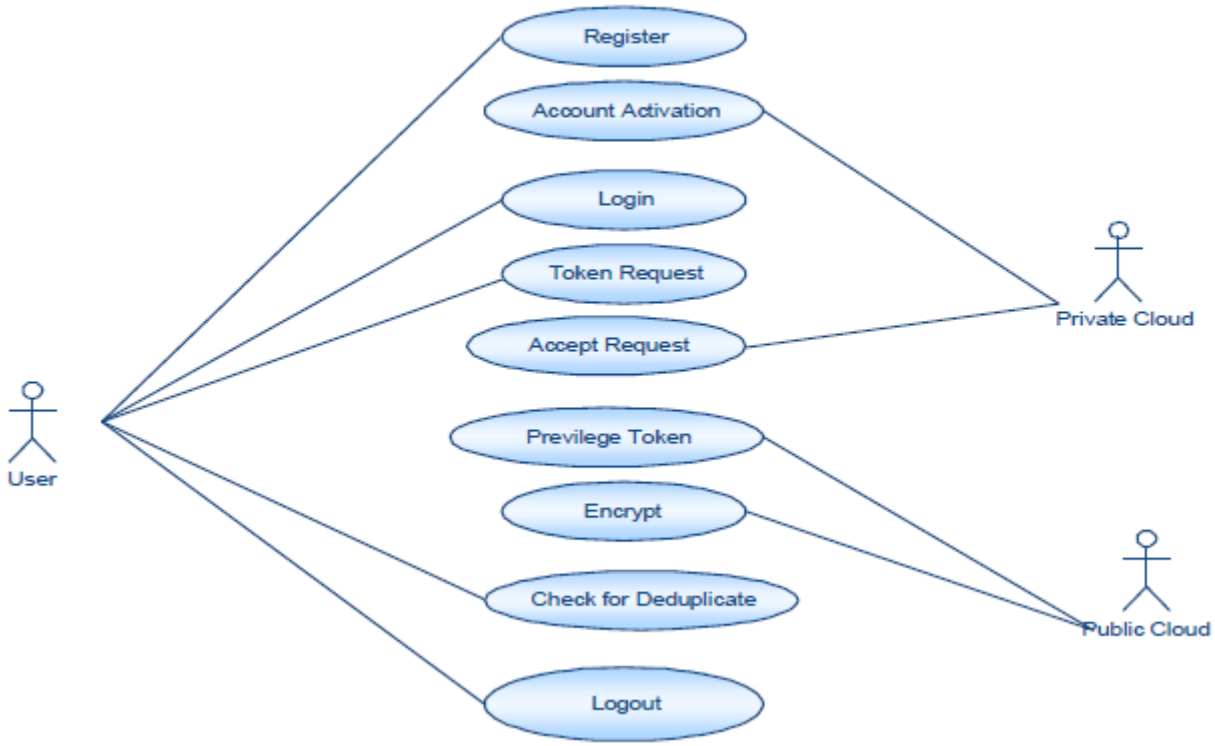

Figure 14

Use Case Diagram of data de-duplication

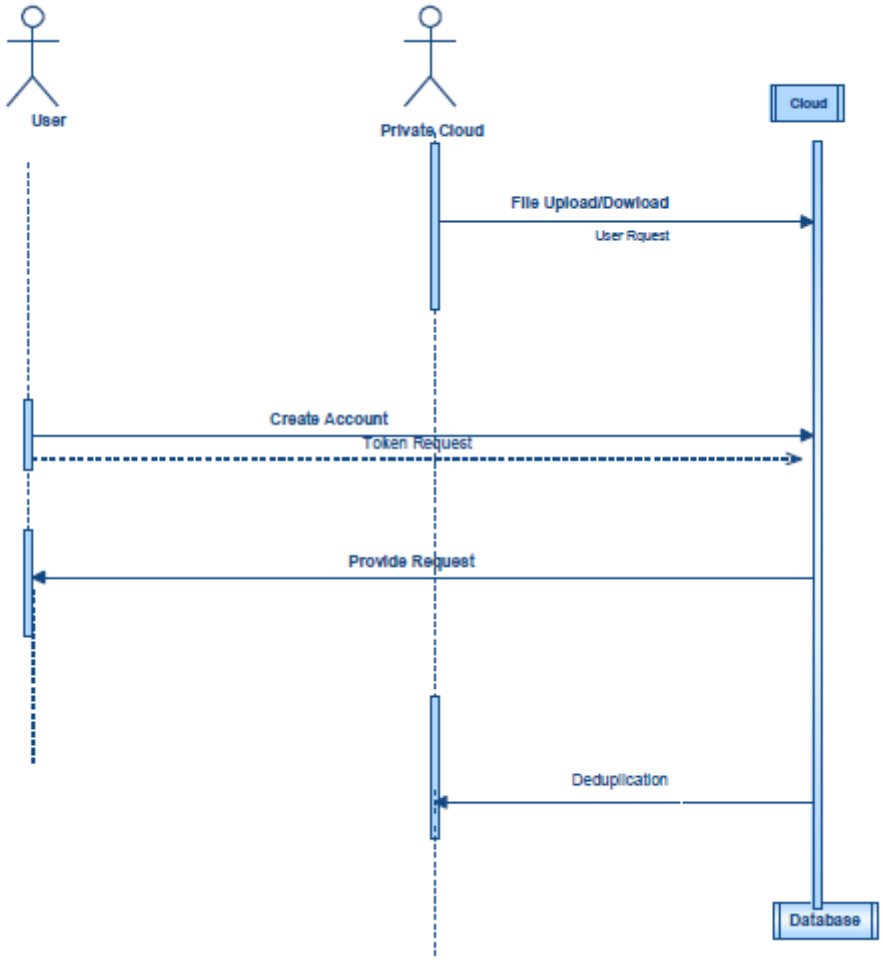

Figure 15

Sequence Diagram of data de-duplication

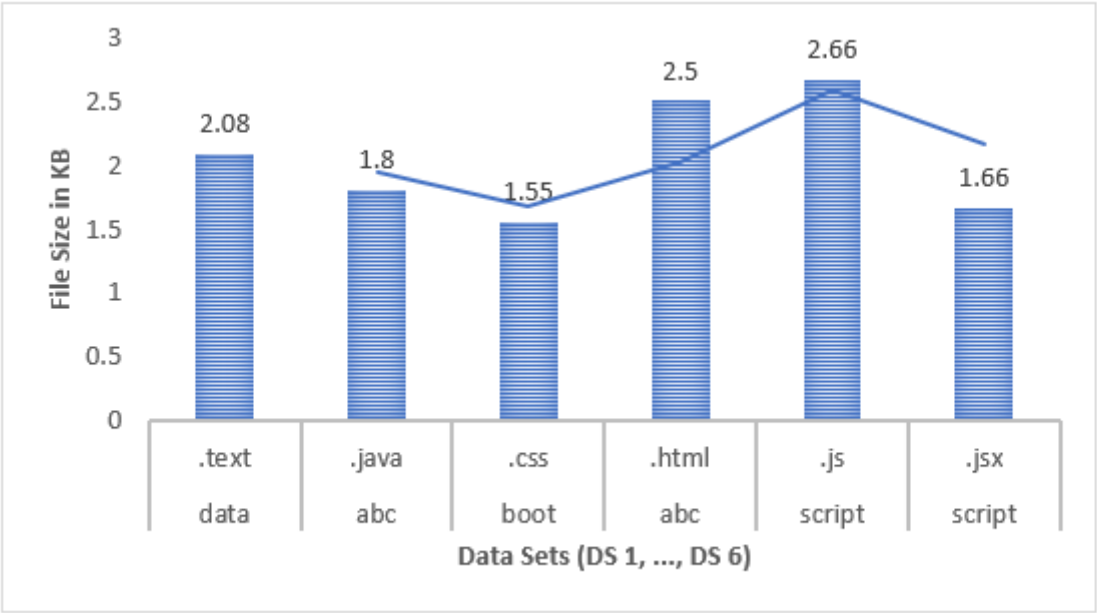

Figure 16

Size of files Before De-duplication

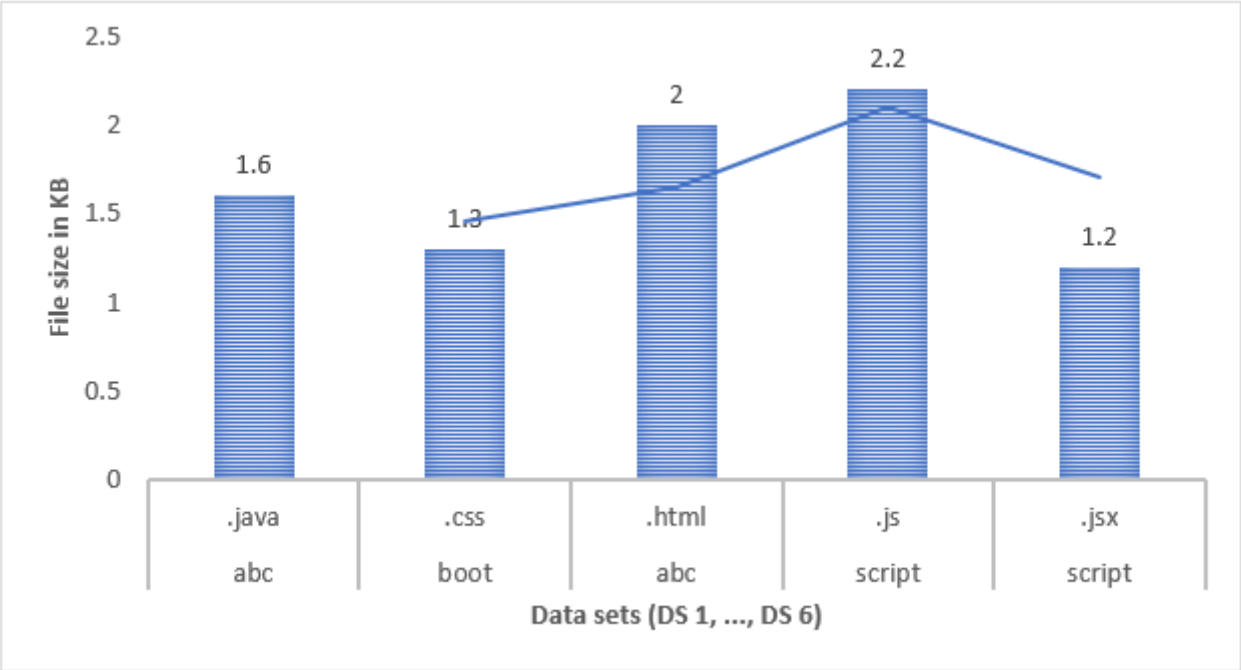

Figure 17

Size of files after De-duplication