International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

A Survey of Various Encryption Techniques in Multi Cloud to Reduce Information Leakage

Geetinder Saini[1], Dr. Navdeep Kaur [2]

Submitted:12/03/2024 **Revised**: 27/04/2024 **Accepted**: 04/05/2024

Abstract: Unintentionally revealing private data is known as data leakage. In the context of cloud data warehousing, a malicious insider with data access can utilize cross-database correlations and repetitive database queries to deduce sensitive information. This particular security threat has not received sufficient attention in the past. As the volume of data in public clouds continues to grow, the impact of information leaks is anticipated to escalate. The data leakage can be reduced using encryption techniques in multi cloud. In this paper, various encryption techniques for data leakage reduction are reviewed and examined in light of specific criteria.

Keywords: *Multi Cloud, Data Leakage, Encryption Techniques, Data security*

1. Introduction

The growing volume of digital data necessitates large-scale, easily accessible network storage, given the widespread usage of devices like laptops, smartphones, and tablets. Affordable and user-friendly cloud-based applications such as Dropbox, Google Drive, and Amazon S3 have gained popularity. However, these centralized cloud storage services have faced criticism for potentially exerting control over user data and utilizing it for marketing research. Concerns also arise regarding data leaks stemming from corrupt practices, intimidation, backdoors, and malicious insiders. One potential answer to mitigate information leakage risk is employing multi-cloud storage systems, where data is distributed across various cloud service providers, making it harder for a single attack to compromise all data.

By adopting multi-cloud strategy, data is distributed across individual clouds, reducing the risk of permanent failure and maintaining data integrity even if some clouds experience issues. Simultaneously using multiple clouds can also enhance data and application security in a public cloud environment. However, standard cloud adoption challenges, such as security, reliability, cost, and loss of control, persist. The implementation of multi-cloud environments offers organizations greater flexibility in determining where to run specific workloads and increased control over the services they utilize. Three dispersed entities collaborate in a multi-cloud storage structure to synchronize user data from distant clients to the cloud. The client is responsible for optimizing data through various processes like chunking (dividing files into smaller chunks), deduplication (avoiding duplicate content storage), delta encoding (transmitting only modified parts of a file), bundling, and encryption as well as decryption [2].

Metadata servers play a crucial role in storing structured data, comprising information about files, cloud service providers (CSPs), and users, representing the entire cloud file system. On the other hand, storage servers are responsible for storing both structured and unstructured raw data blocks.

1.1. Data Security Requirements in Cloud Storage

Safety of data in cloud storage encompasses various aspects, including:

1.1.1. Data Confidentiality:

It involves safeguarding user information from malicious use by unidentified individuals, ensuring that the information shared between the sender and the recipient remains unchanged. In simpler terms, only authorized individuals should have access to and be able to obtain the data. This concept is similar to how you expect to access your bank account, and while employees at the bank assisting youwith transactions can access it, no one else should be able to do so. If accessed by unauthorized individuals, data confidentiality is compromised, and this breach is difficult to reverse [3].

1.1.2. Data Integrity

This factor pertains to the reliability and integrity of the data, ensuring that it cannot be tampered with or arbitrarily altered. For instance, when shopping online on platforms like Amazon, the integrity of your data ensures that Nobody else without permission may alter the goods in the shopping cart. The lack of this factor can lead to grave security threats as it undermines the accuracy and trustworthiness of the data.

Department of Computer Science, Sri Guru Granth Sahib World University, Fatehgarh Sahib, Punjab, India[1],[2]
* *[1]geetsaini2024@gmail.com, [2]drnavdeep.iitr@gmail.com*

1.1.3. Data Availability

This aspect ensures the accessibility of data according to the requirement without any hindrance. Users should be able to access, download, and make modifications to their data in the cloudpromptly whenever they require it.

1.1.4. Fine-Grained Access Control

Fine-grained access control involves implementing precise and detailed control over data accesspermissions. It allows for granular restrictions,specifying access rights at a very specific levelfor different users or groups.

1.1.5. Leakage-Resistant

Leakage resistance refers to the ability of the system to prevent data leaks or unauthorized access to sensitive information. Robust security measures are put in place to safeguard against malicious activities, insider threats, or accidental disclosures of data to unauthorized parties.

1.1.6. Completely Data Deletion

Users can securely delete their data from cloud storage when they no longer need it, ensuring it is permanently removed and preventing unauthorized access by unscrupulous cloud service companies.

1.1.7. Privacy Protection

Privacy security mechanisms are employed to safeguard users' private information, including personal identities, locations, and sensitive data, from enquiring opponents and malevolent workers of cloud hosting companies. These measures ensure the confidentiality of users' data [5].

1.2. Data Encryption

Data security in the cloud is vulnerable when data is outsourced. To address this, encryption serves as a successful means of safeguarding privacy of information. The fundamental concept of data encryption involves transforming the initial plaintext file or data into an indecipherable code, known as ciphertext, using specific algorithms. Even if someone intercepts the encrypted data, they cannot decipher it without the proper decryption key, ensuring the confidentiality of the data and preventing unauthorized tampering. The data can be accessed and modified by authorized users who possess the corresponding private key for decoding the ciphertext.

Symmetric and asymmetric encryption are the two primary categories of encryption techniques. The decryption and encryption steps of symmetric encryption share the same secret key. However, it requires establishing a consensus key beforehand [6], which can be inconvenient, especially in scenarios involving multi-user file sharing. On the contrary, public key encryption, or asymmetric encryption, offers greater convenience. It makes use of two keys: a private key that is needed to decrypt the ciphertext and a public key that can be freely shared for file encryption.

In the context of cloud storage systems, various encryption technologies are widely applied to enhance data security and ensure the protection of sensitive information.

1.2.1. IBE: Identity-Based Encryption

To ensure alignment between identification details and the public key used for encryption, it is essential for the sender to validate the recipient's identity through a trustworthy third-party Certificate Authority (CA) in the traditional Public Key Infrastructure (PKI) before encrypting a file with the public key. However, when sharing data with multiple recipients, this process can pose a significantly higher burden for the sender.

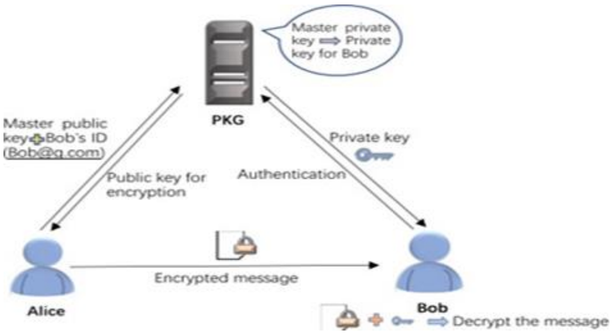
Master private key ⟹ Private key for Bob
PKG
Master public key + Bob's ID (Bob@g.com)
Public key for encryption
Authentication
Private key
Encrypted message
Alice
Bob
+ ⟹ Decrypt the message

Fig 1. Identity-based encryption (IBE).

Fig. 1 demonstrates the process of identity-based encryption. In this system, Alice wants to send Bob an encrypted message. The necessary public and private keys are generated by a reliable third party known as the Private Key Generator (PKG). Alice obtains the public key from the PKG by using Bob's distinctive identity information, such as his email address (Bob@g.com). She then sends Bob the encrypted message. After obtaining the encrypted communication, Bob verifies his identity with the PKG and acquires the relevant private key to unlock the message. [7]. Identity-based cryptography has emerged as a resolution to this challenge, eliminating the need to authenticate the recipient's certificate before encrypting data by directly associating the user's identification information with the public key.

1.2.2. ABE: Attribute-Based Encryption

A distinct and significant string represents each user's identity in the IBE system. However, the flexibility of this scheme becomes limited when multiple users need legal access to the same ciphertext. In 2005, fuzzy identity-based encryption was developed as a solution to this problem, and it served as the basis for ABE (Attribute-Based Encryption). In ABE, instead of using individual identities, a set of attributes replaces the identity. Unlike with standard IBE, access to encrypted data is restricted to users whose attribute set fits the preset access policy. This allows for greater versatility and extremely fine access control.

The ABE algorithm typically comprises four main phases:

1. Setup Phase: This face is also named as the system initialization phase, this step involves providing relevant security parameters as input to the system. Consequently, the system produces the matching master key (MK) and public parameters (PK). [8].

2. KeyGen Phase: The owner of the information provides the platform with their attributes at this step, which is referred to as the key generation phase, in order to receive the private key linked to those attributes.

3. Encryption Phase: This stage creates ciphertext (CT), which is the encrypted data that the data holder has access to using their public key. The ciphertext can then be sent to the intended receiver or stored in a public cloud.

4. Decryption Phase: After receiving the ciphertext, users of decryption are able to access the original information by decrypting the data using a private key they created (SK).

Data proprietors can designate who has access to encrypted data using ABE (Attribute-Based Encryption), which provides the possibility of restricted access in data sharing apps [9]. Key-Policy Attribute-Based Encryption (KP-ABE) and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) are the two categories into which it is divided.

1.2.2.1. KP- ABE (Key-Policy Attribute-Based Encryption)

Each ciphertext in this type of encryption is linked to a collection of attributes, and a rule of access for these attributes is connected to a user's private key. As an illustration, Fig. 2 displays C1 as a ciphertext encrypted with qualities such as “Applied Mathematics” and “Student”. User 1's access policy is defined as “(‘Department of Mathematics’) OR (‘Student’ AND ‘Applied Mathematics’)’’. User 1 is authorized to decrypt ciphertext C1 since its attributes meet User 1's access policy. On the other hand, user 2's access policy allows them to decrypt ciphertexts with attributes (‘Department of Mathematics’, ‘Student’) OR (‘Department of Mathematics’, ‘Basic Mathematics’), but not C1 [10]. Similarly, user 3 cannot decrypt C1, as their access policy does not match the attributes in the ciphertext.

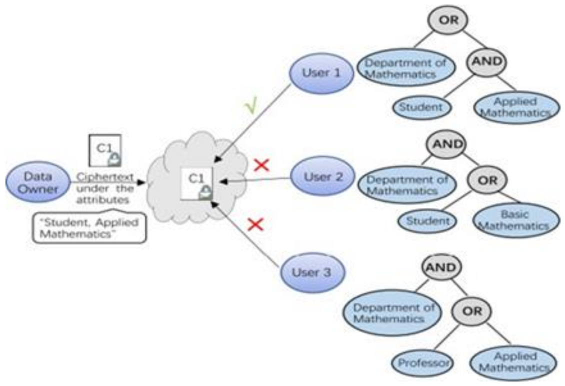
OR
Department of Mathematics
AND
Student
Applied Mathematics
User 1
√
C1
Data Owner
Ciphertext under the attributes
"Student, Applied Mathematics"
C1
✕
User 2
AND
Department of Mathematics
OR
Student
Basic Mathematics
✕
User 3
AND
Department of Mathematics
OR
Professor
Applied Mathematics

Fig 2. KP-ABE in cloud.

1.2.2.2. CP-ABE (Ciphertext-Policy Attribute-Based Encryption)

The access policy in CP-ABE is part of the ciphertext, the data owner can decide which qualities are required in order for an individual to be able to gain access to the ciphertext. A collection of matching attributes are linked to each user's private key. Access patterns can be mathematically described as a monotonic “access tree”, in which characteristics are represented by the leaves and threshold gates are the nodes.

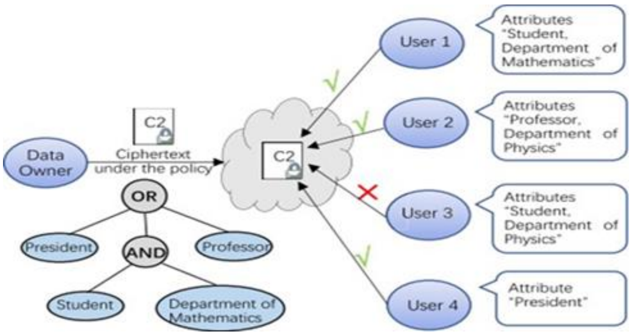
User 1
Attributes "Student, Department of Mathematics"
√
C2
√
User 2
Attributes "Professor, Department of Physics"
Data Owner
Ciphertext under the policy
C2
✕
User 3
Attributes "Student, Department of Physics"
OR
President
AND
Professor
√
Student
Department of Mathematics
User 4
Attribute "President"

Fig 3. CP-ABE in Cloud

An instance of a sensitive file is encrypted with the access policy “(‘President’) OR (‘Student’ AND ‘Department of mathematics’) OR (‘Professor’)’’. This infers that individual with attributes (“President”) or (‘Student’, ‘Department of Mathematics’) or (‘Professor’) can access the file, as depicted in Figure 3. Users' attributes may vary due to several reasons, such as job transfers. Attribute changes can result in users becoming ineligible to access data they were previously authorized to access. Furthermore, some permitted users' bad behavior [11], including their cooperation with hackers, might jeopardize data privacy and confidentiality, resulting in damages for the data proprietor. Hence, a secure revocation mechanism in ABE is essential to address these issues.

1.2.3. Homomorphic Encryption (HE)

Even though IBE and ABE provide some degree of data secrecy in the cloud, they have certain shortcomings. There are two ways for a user to make changes to their encrypted files that are kept on the cloud. Making changes to the ciphertext immediately in the cloud is one method. However, data corruption frequently occurs from decrypting the altered ciphertext, which produces useless jumbled code. The alternative technique entails uploading the freshly encrypted file to the cloud after locally updating the decrypted file. This is a laborious and complicated operation, particularly when handling big data sets. It takes a long time and uses a lot of the user's local device's processing power. Moreover, there is a chance of data leakage while moving data from local to cloud storage [12].

To address these challenges, homomorphic encryption (HE) emerges as a superior solution. After decryption, the results match those obtained from performing the same operations on plaintext.

A comprehensive explanation of Homomorphic encryption's cloud-based functionality may be found in Figure 4. In this procedure, the file is sent to the cloud server encrypted by the data owner using homomorphic encryption.

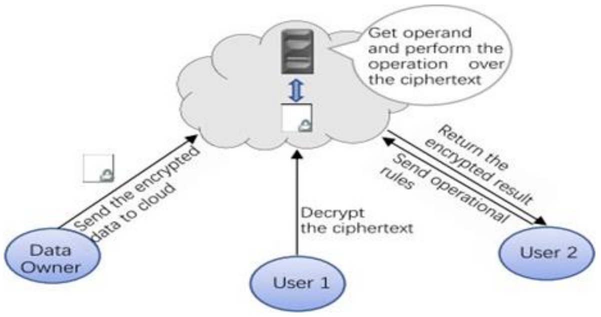
Get operand and perform the operation over the ciphertext
Send the encrypted data to cloud
Decrypt the ciphertext
Return the encrypted result
Send operational rules
Data Owner
User 1
User 2

Fig. 4. Homomorphic encryption in cloud

The ciphertext can be decrypted by authorized users by using the matching private keys. User 2 only needs to submit the appropriate functions or operations to the cloud server in order to carry out certain actions on the ciphertext. User 2 receives the encrypted results from the server once it has completed the actions without first decrypting the ciphertext. This way, homomorphic encryption ensures that data security is effectively protected while enabling secure computation on encrypted data in the cloud [13].

Partial homomorphic encryption (PHE), Somewhat homomorphic encryption (SHE), and Full homomorphic encryption (FHE) are the three variants of homomorphic encryption that can be distinguished according to the ciphertext's processing capability.

a) **Partial Homomorphic Encryption (PHE):** PHE permits addition homomorphism or multiplication homomorphism, but not both, to be performed on the ciphertext. The Paillier system, for instance, is a traditional additive homomorphic encryption technique that allows the use of the private key for decryption and the public key for addition operations on encrypted data.

b) **Somewhat Homomorphic Encryption (SHE):** With some restrictions on the total number of multiplications that can be carried out, SHE permits addition as well as multiplication operations on the ciphertext. With a restricted amount of permitted multiplications, the majority of SHE schemes allow combined multiplication and addition on data encrypted with a single public key.

c) **Full Homomorphic Encryption (FHE):** Any kind of function can be calculated on encrypted data thanks to FHE's ability to perform any amount of additions and multiplications. FHE is capable of executing various operations on encrypted data without requiring decryption. It is a crucial technology for secure cloud computing, as computations can be outsourced to the cloud server using the secret key for decryption. FHE algorithms have additive and multiplicative homomorphism features and can execute multiple addition and multiplication operations [14].

1.2.4. Searchable Encryption

Cloud storage is popular due to its limitless space and flexible services. Users frequently encrypt their data before transferring it to the cloud, guaranteeing secrecy and protecting the safety of their information. However, searching for specific encrypted files in the cloud can be challenging, as the data is encrypted and not directly searchable. Two solutions are proposed to address this issue. The first solution involves downloading the encrypted files to the local device, decrypting the ciphertext, and then searching for keywords in the plaintext. Although secure, this approach can be resource- and time-consuming, particularly when dealing with big data sets. The second solution is to perform the search in the cloud after decrypting the ciphertext. However, this approach poses a serious risk to data security and user privacy since it exposes the plaintext content to the cloud server [15].

To tackle this concern, researchers are working on a cryptographic primitive known as searchable encryption. With searchable encryption, authorized users can retrieve ciphertext in the cloud through keyword queries or similar methods. The key feature of searchable encryption is that it enables the cloud server to return encrypted data files without being aware of the content of the ciphertext. Searchable Symmetric Encryption (SSE) and Public Key Encryption with Keyword Search (PEKS) are the two categories under which searchable encryption falls within the framework of encryption techniques. Whereas PEKS is a searchable encryption method based on public key cryptography, SSE is a type of searchable encryption relying on symmetric cryptography. A Public Key Encryption with Keyword Search (PEKS) technique is shown in Figure 5 and is used to enable searchable encryption on emails that have been encrypted employing a public key.

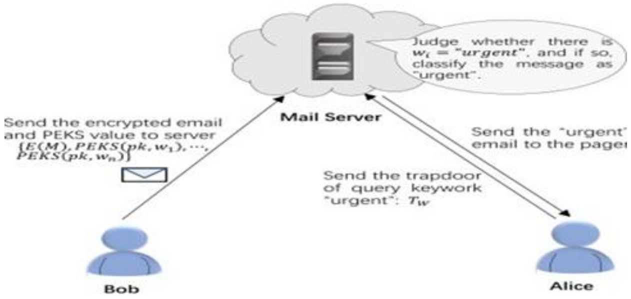
Judge whether there is w_i = "urgent", and if so, classify the message as "urgent".
Mail Server
Send the encrypted email and PEKS value to server {E(M), PEKS(pk, w_1), ⋯, PEKS(pk, w_n)}
Send the "urgent" email to the pager
Send the trapdoor of query keywork "urgent": T_W
Bob
Alice

Fig. 5. Public Key Encryption with keyword Search (PEKS)

PEKS value will not disclose any email content other than the designated keywords throughout the entire process [16]. In this approach, Bob delivers the encrypted message E(M) and PEKS values associated with the keywords in the message M, denoted as $PKES$ (pk, wi), i = 1,2, … , n to the email server. Meanwhile, Alice forwards the trapdoor Tw for the specific keyword (e.g., 'urgent') to the server. The server then verifies if there exists an $i \in$

{1,2, … , n} such that wi = w. Throughout this process, the PEKS values will not disclose any email content except for the specific keywords. This ensures that the server can perform keyword searches without accessing the actual message content, maintaining data confidentiality while enabling efficient and secure keyword-based retrieval.

2. LITERATURE REVIEW

Liu, Y. et. al [1] paper provides an in-depth analysis of various cryptographic approaches for secure cloud data storage. It encompasses a variety of encryption methods, such as proxy re-encryption, attribute-based encryption, and homomorphic encryption. The article talks about their strengths and limitations concerning data leakage prevention in multi-cloud environments.

Chang, Y. C. et. al [2] proposes a hybrid encryption scheme integrating symmetric and asymmetric encryption to make cloud host applications more secure. The study evaluates the scheme's effectiveness in reducing data leakage risks in multi- cloud scenarios and assesses its computational overhead

Martínez, S. G. et. al [3] presents a taxonomy of cloud computing systems, including various encryption mechanisms used for data protection. It discusses how encryption techniques can be employed in multi-cloud settings to minimize data leakage and improve data security.

J. Fan, et.al [4] proposes a privacy- preserving data analysis framework using homomorphic encryption in a multi-cloud setting. The study demonstrates how multiple cloud providers can collaborate on data analysis tasks while keeping the data encrypted throughout the process. The authors highlight the security and performance trade-offs of the approach.

Jin, H. et. al.[5] proposes a multi-cloud data sharing scheme using homomorphic encryption to ensure data privacy during sharing and reduce data leakage risks. The authors demonstrate how the proposed scheme achieves secure and efficient data sharing among multiple clouds, maintaining the confidentiality of the shared data.

Chen, X. et. al[6] presents a safe, multi-keyword search algorithm for cloud data that is secured. The authors investigate methods for enabling effective and private keyword searches over encrypted information while reducing the possibility of data leaks in multi-cloud settings.

Y. Zhang, et.al. [7] explores the implementation of a multi-cloud computing model with secure data sharing capabilities. The study focuses on practical aspects of using homomorphic encryption to enable secure data sharing across multiple clouds. The researchers address the scalability of the suggested architecture and assess its efficacy.

R. Duan, et. al [8] work addresses the challenges of efficient data sharing in multi-cloud environments while preserving data privacy. The researchers propose an optimization technique to improve the efficiency of data sharing operations using homomorphic encryption. The study includes a performance evaluation of the proposed optimization.

A. Tchernykh, et.al [9] discussed that the multi-cloud data was suffered from the issue of data breaching, and confidentiality [22]. Thus, a configurable, and protected dispersed method for storing data was established to make the data reliable, eliminate the redundant data and enhance the encoding/decoding speed. A new approach has been implemented for a Polynomial Residue Number System (PRNS), which integrates secret sharing techniques with error-correcting codes. Additionally, the concept of approximate rank (AR) for a polynomial has been introduced to streamline the computational processes of data encoding and decoding while also reducing the size of PRNS coefficients. The AR-PRNS technique leverages the approximate value alongside the benefits of PRNS to identify and correct errors, manage computational outcomes, and support scalable parallel computing. A theoretical framework has been developed to configure and optimize the redundancy of stored data and the speed of encoding and decoding, addressing diverse objectives, workloads, and storage characteristics. The findings indicate that the proposed technique enhances data security and reliability while reducing data storage overhead.

Zhang, X. et. al.[10] propose a method to secure data processing in multi-cloud environments using homomorphic encryption. They evaluate the performance and security of the approach, highlighting its potential to mitigate data leakage concerns.

M. S. Rane,[11] provides an overview of various homomorphic encryption schemes and their applications in multi-cloud scenarios. The authors review the state-of- the-art homomorphic encryption techniques and discuss their strengths and limitations. The paper serves as a comprehensive guide for researchers and practitioners interested in homomorphic encryption in multi-cloud environments.

S. Zhou, et.al [12] suggested a method to extend ciphertext policy attribute-based encryption (CP-ABE), and a multi-authority CP-ABE access control algorithm was presented to fulfill the requirements for multi-cloud storage access control (MSAC) [26]. A tree structure of this method was adopted in a mapping mechanism. This method focused on supporting the kinds of attribute values. A simple prototype system was developed to compute the suggested method in analysis. The outcomes demonstrated the practicality and efficacy of the suggested method for control research in multi-cloud storage systems (MCSS). Additionally, this method offered lower computing time and overhead in comparison with others.

K. V. Nil, et.al [13] recommended a data leakage aware storage system in the multi- cloud [17]. In order to detect the data leakage, two algorithms known as SHA-1 and Advanced Encryption Standard (AES) were adopted in this system for placing the data at little data leakage on the same cloud that was based on similarity. The evaluation outcomes indicated the effectiveness and robustness of the recommended system to alleviate the data leakage in multicloud storage system. This system had performed well and effective for lessening the information leakage. In the end, this system was quantified further. According to experimental results, the recommended system was capable of securing the data from leakage as well as making the data more complicated to prevent it from assaults.

W. Shi, et.al [14] emphasized on dealing with the issue of securing data in multi-cloud and developing a multi-cloud storage system relied on the traditional single cloud storage system . An Erasure Code was implemented for blocking the original data and Advanced Encryption Standard (AES) approach was adopted for encrypting the data blocks. After that, this system concentrated on storing the encrypted data blocks in dissimilar cloud storages terminals. This process resulted in eliminating the hidden danger of data loss or leakage because of relying on a cloud service provider, enhancing the existing security issues of cloud storage system. Based on experiments, the developed system was secure and reliable against the data leakage.

N. A. Tolasa, et.al. [15] constructed a StoreSim system which provided leakage awareness regarding the information from the storage system in multiple clouds [21]. This system was deployed for storing the similar data on the same cloud in syntactic way for mitigating the information leakage. Subsequently, a MinHash algorithm was suggested for generating similarity- preserving signatures for data chunks and a function was built for evaluating the information leakage on the basis of these signatures which were sued to hash the data. The fingerprinting algorithms, called SHA-1 and MD 5 were adopted for this purpose. Eventually, an optimal system was generated when the data chunks were distributed across multiple clouds. Furthermore, the optimal multicloud storage providers were also presented for alleviating the information leakage in an effective way. The results revealed that the constructed system optimized the cost, made the data reliable and available.

L. Megouache, et.al. [16] projected an innovative framework for making the data authentic and reliable in a distributed and interoperable scenario . Primarily, an analysis was carried on some security technique in a huge and distributed environment. Subsequently, a novel mechanism was put forward for tackling the security issues. The initial stage suggested a private virtual network for securing the data during its broadcasting. The second stage made the deployment of an authentication technique on the basis of encrypting data for protecting the identity of the user and its data. The last stage focused on verifying the integrity of data whose distribution was done over multi-cloud using an algorithm. the projected framework was applicable for authenticating the identity and inter-operating among procedures executed on dissimilar cloud's provider. Furthermore, a data integrity algorithm was presented. The simulation outcomes confirmed that the projected framework was effective and secure to generate a reliable and stable mechanism in the multi-cloud environment.

F. Shahid, et.al [17] focused on dealing with the issues related to data security on multi cloud (MC) and suggesting a Proficient Security over Distributed Storage (PSDS) technique . The initial task of this technique was to split the data into two classes, namely normal and sensitive and further partitioning the sensitive data into 2 sections. This technique was implemented for encrypting and distributing every portion over multi-cloud and uploading the normal data on a single cloud in encrypted form. The decryption phase was executed for integrating the sensitive data from MC. Various attacks were launched for testing the suggested technique. The results indicated the resistance of suggested technique against key assaults, pollution attack (PA), chosen ciphertext attack, and known plain text attack (KPTA). Moreover, this technique consumed least computation time in contrast to the traditional technique.

W. Shi, et.al[18] projected a multi-cloud storage system on the basis of the classic single cloud storage system . An Erasure Code was executed for blocking the original data, and encrypting the data blocks relied on Advanced

Encryption Standard (AES) algorithm. After that, this system was utilized for storing the encrypted data blocks in dissimilar cloud storages terminals. It resulted in removing the unseen danger of data loss or leakage occurred because of a cloud service provider (CSP); enhancing the security issues and making the multi-cloud secure storage system more secure and reliable. The next intend of this system was to encrypt the data blocks obtained after coding the file segmentation and transmitting them transferred to multiple cloud object storage services. Hence, the projected system had potential for alleviating the storage redundancy, making the system more stable, and making the data more reliable and secure.

J. Yao, et.al [19] formulated a robust method to slice the data blocks, encrypt them at an individual level, and store them on multiple clouds. Furthermore, a multi cloud dynamic storage scheduling (MCDSS) and a local storage optional configuration (LSOC) approaches were presented to make the data storage more reliable. Meanwhile, a micro service model and a decentralization strategy was employed for ensuring the availability of services. The issue of data storage efficacy was resolved using data De duplication method in the multi cloud backup system. The notion related to count and reach the reference was adopted in MCSS. An analysis was conducted on the formulated method. The experimental outcomes revealed the feasibility of the formulated method for meeting the design objective.

S. Zhou, et.al [20] suggested a method to extend ciphertext policy attribute-based encryption (CP-ABE), and a multi-authority CP-ABE access control algorithm was presented to fulfill the requirements for multi-cloud storage access control (MSAC) [26]. A tree structure of this method was adopted in a mapping mechanism. This method focused on supporting the kinds of attribute values. A simple prototype system was developed to compute the suggested method in analysis. The outcomes demonstrated the practicality and efficacy of the suggested method for control research in multi-cloud storage systems (MCSS). Additionally, this method offered lower computing time and overhead in comparison with others.

Dinh, T., et. al.[21] presents a practical implementation of a multi-cloud computation system using homomorphic encryption. The study demonstrates how homomorphic encryption can be utilized for real-world computations while addressing performance challenges. The authors evaluate the system's efficiency and discuss potential use cases.

Zhang, X., et. al. [22] investigates the security aspects of using homomorphic encryption in multi-cloud data processing scenarios. The researchers analyse potential vulnerabilities and propose countermeasures to strengthen the security of homomorphic computation across multiple clouds. The paper also discusses practical deployment considerations.

Zhu, H., et. al.[23] introduces a hybrid approach that combines different homomorphic encryption schemes to optimize data analytics in multi-cloud environments. The authors explore the benefits of this approach in terms of performance and security while ensuring data confidentiality throughout the computation process.

V. Miranda-López, et.al [24] devised a two- level 2Lbp-RRNS technique on the basis of a Redundant Residue Number System (RRNS) with a backpropagation (BP) and hamming distance (HD) techniques to make a configurable and secure multi-cloud data storage more reliable. A new version of fully homomorphic encryption (FHE) was presented and deployed to preserve the privacy, execute the parallel processing and make the data scalable. The major properties of this technique were explained for expanding the current knowledge according to the critical bounding RRNS norms. The devised technique was capable of recognizing and recover more errors. According to analysis, this technique was proved effective for detecting 1.58× and correcting 3.37× more errors in contrast to the traditional methods. The DropBox, GoogleDrive, OneDrive, Sharefile, Box, Egnyte, and Salesforce platforms were applied to compute the devised technique. The results confirmed that the devised technique was performed well on real data with regard to speed to encode and decode the data.

R. Maher, et.al [25] constructed a DropStore model for generating an easy, secured, and consistent backup system in multi-cloud environment. This model concentrated on inserting an abstraction

layer for the end-user so that all the complexities of system were concealed. For this, a locally hosted device called the Droplet was adopted and user was allowed to manage it. Therefore, the user was independent of any unreliable third party. This model was more effective due to the potentials of convergence of Multi-Cloud and Fog Computing. The constructed model was implemented on public platform. The trial results demonstrated the effectiveness of the built model in securing and safeguarding data. Additionally, this architecture maintained a straightforward and user-friendly interface with edge devices while maintaining privacy.

M. Sohal, et.al [26] introduced a multi- cloud security model in which a client-side cryptography was deployed [18]. The BDNA encryption method was exploited in order to encrypt the data of users. Moreover, a hybrid cryptographic approach was presented in which the Identity-based Broadcast Encryption (IBBE) algorithm was employed to manage the keys of the exploited method for which the data was encrypted via a specific edition of IBBE algorithm. This algorithm had effectively secured the encryption keys. An analysis exhibited that the presented approach was secure against Indistinguishable Chosen- Ciphertext Assaults.

Afterward, the security of the introduced model was enhanced using a double encryption procedure against insider assaults and malevolent users. The data was stored in real-time storage clouds. A web application was applied to simulate the introduced model. According to simulation, the introduced model had worked securely while encrypting the data and offered lower overheads.

G. P. Kanna, et.al [27] created a unique RSDM-ACPAR technique that isolated sensitive data in the customer profile and uploaded encrypted data to the multi-server cloud model by using the rule-based statistical disclosure method (RSDM) and access control policy-based access restriction (ACPAR). [19]. First of all, To separate the highly sensitive information from the regular attributes in the customer profile, the normalization process was completed by assigning visibility and hiding metrics to the relevant fields in the dataset. After that, an enhanced ElGamal algorithm was implemented to encrypt the data and store it into the multi-cloud. The initial method was employed as the base to isolate the sensitive data. To further protect the data, the policy on access control was designed to limit who may view a profile. In the end, the developed method was assisted in decrypting the data regarding the de- normalized profile for integrity. The experimental outcomes revealed the supremacy of the developed method against the existing ones and proved effective in complex data-based applications concerning encryption time, policy generation time, execution time and the access time.

Y. Ameur, et.al [28] designed an innovative framework for Electronic Health Records (EHRs) with Multi-Clouds [20]. The purpose of implementing this framework was to guarantee user data security in a multi-cloud and network environment. As a result, this made use of the multi-key homomorphic encryption (MHE) technique. Additionally, this architecture helped the user or a third party, a cloud provider (CP), to deploy functions on encrypted data without having to reveal the data's values. This framework was planned on the basis of OpenFHE which was effective to perform HE and a non-expert user was capable of configuring its structure. The analysis validated that the designed framework was feasible to perform homomorphic operations easily.

Table 1. Various Encryption techniques used to reduce data leakage in multi-cloud environment

Author	*Year*	*Techniques Used*	*Results*
Liu, Y. et al. [1]	2013	Homomorphic, ABE, PRE encryption	Analysis of cryptographic approaches for secure cloud storage in multi-cloud environments.
Chang, Y. C. et al. [2]	2014	Hybrid encryption	Enhanced security in cloud storage, reducing data leakage risks and assessing computational overhead.
Martínez, S. G. et al. [3]	2016	Various encryption mechanisms	Taxonomy of encryption mechanisms minimizing data leakage in multi-cloud environments.
J. Fan et al. [4]	2016	Homomorphic encryption	Privacy-preserving data analysis with security and performance trade-offs in multi-cloud settings.
Jin, H. et al. [5]	2017	Homomorphic encryption	Secure and efficient multi-cloud data sharing, ensuring data privacy and reducing leakage risks.
Chen, X. et al. [6]	2017	Multi-keyword search	Efficient and privacy-preserving keyword searches in multi-cloud environments.
Y. Zhang et al. [7]	2018	Homomorphic encryption	Secure data sharing model with performance and scalability discussions in multi-clouds.
R. Duan et al. [8]	2019	Homomorphic encryption	Optimization technique enhancing data sharing efficiency with performance evaluation.
A. Tchernykh et al. [9]	2019	Polynomial Residue Number System	Secure and reliable multi-cloud data storage, reducing leakage risks and overhead.
Zhang, X. et al. [10]	2019	Homomorphic encryption	Secure data processing with performance and security

			evaluation in multi-cloud environments.
M. S. Rane [11]	2020	Homomorphic encryption	Overview of homomorphic encryption applications in multi-cloud scenarios.
S. Zhou et al. [12]	2020	CP-ABE	Multi-authority CP-ABE method for efficient multi-cloud storage access control.
K. V. Nil et al. [13]	2020	SHA-1 and AES	Effective data leakage detection system in multi-cloud storage.
W. Shi et al. [14]	2020	Erasure Code & AES	Enhanced security and reliability in multi-cloud storage, mitigating data loss/leakage.
N. A. Tolasa et al. [15]	2020	MinHash for similarity-preserving signatures	StoreSim system for mitigating information leakage in multi-cloud storage.
L. Megouache et al. [16]	2020	Private virtual networks, encryption	Data authenticity and reliability framework for distributed multi-clouds.
F. Shahid et al. [17]	2020	Encryption	Proficient Security over Distributed Storage, resistant to attacks, with lower computation time.
J. Yao et al. [19]	2020	Data block slicing, encryption	Robust method for efficient data slicing, encryption, and scheduling in multi-clouds.
S. Zhou et al. [20]	2020	CP-ABE, Tree Structure	Efficient multi-cloud storage access control with lower computing time and overhead.
Dinh, T. et al. [21]	2021	Homomorphic encryption	Implementation of multi-cloud computation addressing performance challenges and efficiency.
Zhang, X. et al. [22]	2021	Homomorphic encryption	Security analysis of homomorphic encryption in multi-cloud data processing.
Zhu, H. et al. [23]	2021	Hybrid homomorphic encryption	Optimized data analytics performance and security with hybrid homomorphic encryption.
V. Miranda-López et al. [24]	2021	2Lbp-RRNS	Secure multi-cloud data storage with effective error detection and correction.
R. Maher et al. [25]	2021	DropStore model	Reliable and secured multi-cloud backup model enhancing data privacy and security.
M. Sohal et al. [26]	2022	Client-side cryptography	Secure multi-cloud storage model with BDNA encryption for real-time storage.
G. P. Kanna et al. [27]	2022	RSDM-ACPAR	Secure multi-cloud data isolation using access control policy, effective against existing methods.
Y. Ameur et al. [28]	2023	Multi-key homomorphic encryption	Secure framework for Electronic Health Records in multi-cloud environments.

From the table, homomorphic encryption stands out for its ability to perform computations on encrypted data without needing decryption, providing strong privacy despite higher computational costs (Liu, Y. et al. [1]; J. Fan et al. [4]; Zhang, X. et al. [22]). Hybrid encryption, combining symmetric and asymmetric methods, offers balanced security and performance, making it effective for reducing data leakage risks (Chang, Y. C. et al. [2]). CP-ABE allows fine-grained access control over encrypted data, ensuring efficient multi-cloud storage access with practical efficacy (S. Zhou et al. [12], [20]).

3. Result & Discussion

Identity-Based Encryption (IBE), Attribute Based Encryption (ABE), Homomorphic Encryption (HE) and Searchable Encryption (SE) are compared in terms of Encryption and Decryption time. The results show an improvement in both encryption and decryption process. Table 2. shows the Encryption and Decryption time for all the four algorithms discussed earlier.

Table 2. Comparison Of Encryption Time Of Various Algorithms With Different File Sizes

File Size (KB)	Encryption Time (ms)			
	Identity-Based Encryption on	Attribute Based Encryption on	Homomorphic Encryption	Searchable Encryption on
4	16	15	10	18
8	31	47	15	50
16	47	62	33	43
32	78	109	69	82
64	218	202	160	187

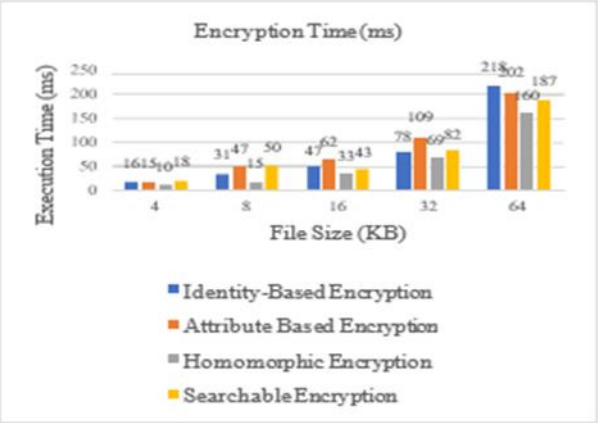

Fig. 6. Comparison of Encryption Time with different file sizes.

Based on Figure 6, it can be seen that the Homomorphic encryption algorithm exhibits the fastest encryption speed compared to Identity-Based Encryption (IBE), Attribute-Based Encryption (ABE), and Searchable Encryption (SE). It performs the encryption in the shortest time frame among the mentioned encryption techniques

Table 3. Comparison Of Decryption Time Of Various Algorithms With Different File Sizes

File Size (KB)	Decryption Time (ms)			
	Identity-Based Encryptionon	Attribute Based Encryptionon	Homomorphic Encryption	Searchable Encryptionon
4	15	16	11	17
8	16	18	13	16
16	17	16	12	15
32	42	43	30	45
64	48	46	39	45

Based on below Figure 7, it is evident that the Homomorphic encryption algorithm surpasses the decryption speed of Identity-Based Encryption (IBE), Attribute-Based Encryption (ABE), and Searchable Encryption (SE). It achieves the shortest decryption time among the mentioned encryption techniques.

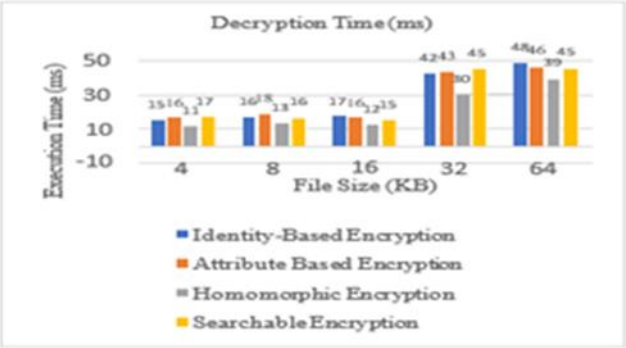

Fig. 7: Comparison of Decryption Time with different file sizes

4. Conclusion

Multi-Cloud is a scalable system in which services from several heterogeneous cloud service providers (CSPs) can be obtained. This system assists the users in acquiring superior better Quality of Service (QoS) reliably and flexibly and securing the data. However, the multi-cloud systems are more prone to data leakage. The issue of data leakage is occurred due to the dishonest actions of malicious users involved in conspiracy. Thus, diverse encryption methods are developed to tackle this issue. This work emphasizes on reviewing four encryption methods, such as Identity-Based Encryption (IBE), Attribute Based Encryption (ABE), Homomorphic Encryption (HE) and Searchable Encryption (SE). The initial one makes the deployment of public identities of entities to achieve cryptographic objectives. This technique prohibits the adversaries from attaining information which is essential to encrypting the data. The second one called ABE is implemented for protecting the security of file data and transmitting the files securely which contain secret data. Homomorphic Encryption (HE) is an encryption method which computes the encrypted data without any decryption. This technique assists in securing the private information even in case of leakage of ciphertext as well as maintaining the data integrity. The last technique is employed for searching any confidential files from database containing an enormous number of encrypted data files securely. A comparative analysis is conducted on these techniques. The results indicate that HE performed more effectively in contrast to others and consumes least time for encryption and decryption.

References

[1] Liu, Y., Lou, W., & Hou, Y. T. (2013). “A Survey of Cryptographic Approaches to Secure Cloud Data Storage” IEEE Communications Surveys & Tutorials, 15(2), 843-859.

[2] Chang, Y. C., & Lin, T. Y. (2014). A Hybrid Encryption Scheme for Secure Cloud Storage Services. Journal of Systems and Software, 92, 25-34.

[3] Martínez, S. G., Choo, K. K. R., & Ashman, H. (2016). A Taxonomy and Survey of Cloud Computing Systems. International Journal of Information Management, 36(4), 605-617.

[4] Fan, J., & Chow, S. S. M. (2016). Secure Multi-Party Computation for Privacy-Preserving Data Analysis in Multi-Cloud Environments. IEEE Transactions on Parallel and Distributed Systems, 27(2), 381-394.

[5] Jin, H., Wang, S., & Wang, Q. (2017). A Secure Multi-Cloud Data Sharing Scheme Based on Homomorphic Encryption. 2017 IEEE Trustcom/BigDataSE/ICESS, 215-222.

[6] Chen, X., Zhang, L., & Xiang, Y. (2017). Secure Multi-Keyword Search with User-Defined Search Results over Encrypted Cloud Data. IEEE Transactions on Cloud Computing, 5(2), 264-278.

[7] Zhang, X., Li, X., Chen, Y., & Zhang, L. (2021). Securing Multi-Cloud Data Processing with Homomorphic Encryption. Future Generation Computer Systems, 117, 67-78.

[8] Duan, R., Yu, Z., Liang, H., & Jia, J. (2019). Efficient Multi-Cloud Data Sharing with Homomorphic Encryption. 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 280- 287.

[9] A. Tchernykh et al., "Data Reliability and Redundancy Optimization of a Secure Multi-cloud Storage Under Uncertainty of Errors and Falsifications," 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Rio de Janeiro, Brazil, 2019, pp. 565-572, doi: 10.1109/IPDPSW.2019.00099.

[10] Zhang, Y., Wu, X., Zhu, X., & Zhang, W. (2018). Multi-Cloud Computing with Secure Data Sharing using Homomorphic Encryption. 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 140- 147.

[11] Rane, M. S., Yavari, A., & Muppala, J. K. (2020). A Survey of Homomorphic Encryption for Secure Computation in Multi-Cloud Environments. Future Generation Computer Systems, 105, 192-209.

[12] S. Zhou, G. Chen, G. Huang, J. Shi and T. Kong, "Research on multi-authority CP-ABE access control model in multicloud," in China Communications, vol. 17, no. 8, pp. 220-233, Aug. 2020, DOI: 10.23919/JCC.2020.08.018

[13] K. V. Nil, “Data Leakage Optimization in Multi-cloud Storage Services”, International Journal of Computer Applications, vol. 175, no. 16, pp. 43- 47, September 2020, DOI: 10.5120/ijca2020920668

[14] W. Shi, H. Wang, L. Wang, and Z. Qin, "Enhancing Security of Multi-Cloud Storage Systems through Erasure Code and AES Encryption," Journal of Cloud Computing: Advances, Systems and Applications, vol. 9, no. 1, p. 55, 2020.

[15] N. A. Tolasa, A. M. Negi, R. Gaur, and S. S. Tyagi, "StoreSim: Mitigating Information Leakage in Multi-Cloud Storage," in Proceedings of the International Conference on Cloud Computing and Big Data Analysis, 2020, pp. 120-126.

[16] L. Megouache, Y. Kadi, and D. Zeghlache, "A Framework for Secure and Authentic Data Distribution in Multi-Cloud Environment," IEEE Transactions on Cloud Computing, 2020.

[17] F. Shahid, Z. Tari, and S. Kanhere, "Proficient Security over Distributed Storage Technique for Multi-Cloud Environment," Journal of Cloud Computing: Advances, Systems and Applications, vol. 9, no. 1, p. 68, 2020.

[18] W. Shi, H. Wang, L. Wang, and Z. Qin, "Enhancing Security and Reliability of Multi-Cloud Storage Systems," IEEE Transactions on Dependable and Secure Computing, 2020.

[19] J. Yao, B. Zhang, and Y. Liu, "Robust Data Storage Scheme for Multi-Cloud Backup System," IEEE Transactions on Services Computing, 2020.

[20] S. Zhou, Y. Wu, and Y. Sun, "CP-ABE Access Control Algorithm for Multi- Cloud Storage," IEEE Transactions on Cloud Computing, 2020.

[21] T. Dinh, C. Liu, and C. Zhang, "Practical Multi-Cloud Computation with Homomorphic Encryption," arXiv preprint arXiv:2106.14040, 2021.

[22] X. Zhang, X. Li, Y. Chen, and L. Zhang, "Securing Multi-Cloud Data Processing with Homomorphic Encryption," Future Generation Computer Systems, vol. 117, pp. 67-78, 2021.

[23] H. Zhu, X. Sun, L. Liu, and X. Han, "Hybrid Homomorphic Encryption for Multi-Cloud Data Analytics," International Journal of Distributed Sensor Networks, vol. 17, no. 4, p. 15501477211011144, 2021.

[24] V. Miranda-López, J. Villalba-Díez, and A. G. López-Herrera, "2Lbp- RRNS: Two-Level Backpropagation Algorithm for Error Detection and Correction in RRNS," IEEE Transactions on Computers, 2021.

[25] R. Maher, S. Shirazipourazad, M. Bakhouya, and A. G. López-Herrera, "DropStore: A Fog Computing-Based Backup Model for Multi-Cloud Environment," IEEE Transactions on Cloud Computing, 2021.

[26] M. Sohal, K. Thapar, and A. Kumar, "Client-Side Cryptography for Secure Multi-Cloud Data Storage," in Proceedings of the International

Conference on Cloud Computing and Big Data Analysis, 2022.

[27] G. P. Kanna, R. Dhatchayani, and S. Ramathilagam, "RSDM-ACPAR: Rule-Based Statistical Disclosure Method for Access Restriction in Multi- Server Cloud Model," IEEE Transactions on Cloud Computing, 2022.

[28] Y. Ameur, K. Salah, and L. Khoukhi, "Innovative Framework for Secure Electronic Health Records in Multi- Cloud Environment," IEEE Transactions on Cloud Computing, 2023