Searchable Encryption Scheme for Large Data Sets in Cloud Storage Environment

Yi XIONG, Ming Xing LUO

School of Information Science & Technology, Southwest Jiaotong University, Chengdu 610031, Sichuan, China

ayanami_xy@126.com, mxluo@swjtu.edu.cn

Submitted December 6, 2023 / Accepted March 12, 2024 / Online first April 19, 2024

Abstract. *Cloud storage has become essential in managing and retrieving extensive volumes of data, providing economical alternatives and adaptability for an effective storage environment. However, in light of the rapid expansion of comprehensive datasets in cloud storage, the preservation of security has emerged as a matter of utmost importance for large data sets. Encryption has become a crucial mechanism for protecting confidential large data sets from unauthorized individuals. Encryption is necessary for safeguarding sensitive data by transforming it into indecipherable code to prevent unauthorized entry. The encryption and decryption process is done at the end-user and cloud server. In the present situation, searchable symmetric encryption assumes a pivotal function by facilitating safe data retrieval while concurrently upholding the principle of secrecy. This research presents the Searchable Encryption Scheme in Cloud Storage Environment (SES-CSE), which offers a resilient solution for tackling the obstacles related to data security and retrieval efficiency for large data sets. The SES-CSE framework effectively incorporates encryption techniques inside a robust search engine, establishing a reliable framework for large data set protection with Okapi BM25. The approach exhibits significant performance benefits, as shown by an encryption time of 14.85 ms, decryption time of 10.06 ms, memory consumption of 77.87 MB, and search times of 13.5 ms. The SES-CSE model demonstrates remarkable retrieval accuracies of 98.41%, 98.57%, and 97.51% throughout the training, testing, and validation phases. The results underscore the usefulness and security of SES-CSE as a solution for cloud storage, improving both the secrecy of data and the efficiency of retrieval in large-scale settings.*

Keywords

Cloud computing, searchable encryption, data sets, security

1. Introduction

When storing large amounts of data – also called big data – cloud storage has emerged as an indispensable option that frees businesses and people from the hassle of managing processing and storage infrastructure [1], [2]. Despite the benefits of cloud services, many prospective customers choose not to use them because of legitimate worries about the privacy and security of information on the cloud's servers and throughout the data transfer procedure [3], [4]. For instance, the most often mentioned worry is data protection and privacy, which is stated by 73% of banks as an explanation for not using cloud-based services. Therefore, improving cloud privacy and confidentiality for user data is crucial.

Cloud computing offers processing and data storage as a type of service [5]. The Data Owner (DO) easily saves and retrieves their information without worrying about capacity, upkeep, or storage space because of the accessible versatility. Data sharing is crucial because it boosts productivity and enhances decision-making, improving commercial or data quality [6]. Cloud storage also makes sensitive data, such as financial transactions, government papers, medical records, more vulnerable.

A Data Owner must provide numerous users with secure access to its information files. The data is in danger due to the peculiar nature of cloud servers [7]. Thus, the Data Owner and the inquisitive cloud server are in different trust zones. Sensitive data ought to be secured by the Data Owner before being outsourced to the cloud to restrict data access. The cloud should offer services to authorized users without violating data privacy [4].

Efficient data finding and exploitation become more complex when data is encrypted. A popular technique for providing access to information is a keyword-based query function. Without getting every file, this keyword-based searchable technique enables users to search for their preferred file efficiently. Data security now becomes dependent on keyword security.

Keeping private information on distant servers is a major cause for alarm because of the many ways this practice might pose serious privacy risks. A Searchable Encryption (SE) is a collection of techniques designed to secure users' private information while keeping the search capability intact on the server. The field of searchable cryptography has been the subject of much study, with

DOI: 10.13164/re.2024.0223

numerous proposed structures that all achieve asynchronous maximum efficiency for different metrics. Even though they are elegant, recent hacking and installation efforts have proven that optimal asymptotic complexity doesn't necessarily mean efficient performance. This is particularly true when an application requires a high level of anonymity.

These vast datasets' extreme diversity and volume highlight the importance of adequate safeguards to protect private information from harmful or unlawful access. Encryption becomes essential since it protects data during transportation and storage. While good at maintaining secrecy, conventional encryption techniques must be revised when used on massive amounts of information in cloud storage settings.

The primary concern is finding a balance between ensuring data safety and being able to search for it easily [8]. As the quantity of data increases, it becomes crucial to have effective search operations to retrieve and analyze information promptly. Conventional encryption algorithms must be improved since they often need to decrypt the whole dataset before doing any search, leading to unrealistic latency and resource usage.

This approach works with various file types, including compressed documents, multimedia files, and remote files. It supports both SSE and PEKS encryption mechanisms. The SES-CSE method also secures document updates for data owners and retrieves the most recent documents for future searches.

The notion of searchable symmetric encryption has become prominent in tackling existing difficulties [9]. Searchable symmetric encryption methods strive to harmonize the contradictory objectives of safeguarding data and conducting search operations efficiently. These approaches provide the extraction of specific details from encoded datasets while maintaining the overall safety of the stored data. Combining cryptographic algorithms with indexed patterns in searchable encryption offers a potential approach to improve the capacity to search for vast amounts of encrypted information in cloud storage systems.

The main contributions of the article include

1. This study introduces the Searchable Encryption Scheme in Cloud Storage Environment (SES-CSE), a robust solution for security and retrieval issues of huge data sets.
2. SES-CSE integrates encryption into a strong search engine, providing a dependable framework for Okapi BM25's huge data set safety.
3. Comparing the suggested model to SPPEK, NAIDES, DNIES, CSSM, and FETCH, the technique has significant performance gains, including 14.85 ms encryption, 10.06 ms decryption, 77.87 MB memory use, and 13.5 ms search times.
4. The SES-CSE model achieves 98.41%, 98.57%, and 97.51% retrieval accuracy during training, testing, and validation. According to the results, SES-CSE improves data secrecy and retrieval in large-scale cloud storage.

The following sections are arranged in the given manner: Section 2 provides a thorough analysis of the current research environment in the literature survey, including a detailed summary of pertinent studies and methodologies. Section 3 introduces an innovative Searchable Encryption Scheme in a Cloud Storage Environment (SES-CSE). Section 4 performs simulation analysis and delivers results, assessing the performance and efficacy of the proposed SES-CSE in real-world situations. Section 5 concludes the study by summarizing the main results and suggesting potential areas for future developments in searchable encryption in cloud storage environments.

2. Literature Survey

The literature study explores current encryption algorithms used in cloud storage systems, thoroughly analyzing approaches and highlighting significant obstacles and constraints. It synthesizes information from earlier studies to provide the groundwork for comprehending the status of encryption algorithms used to secure data in cloud storage today.

A Secure Password-Protected Encryption Key (SPPEK) was developed for deduplicated cloud storage to improve security [10]. Their system generates an encryption key protected by user passwords to ensure data secrecy. To improve encryption using image attributes a Neural-Assisted Image-Dependent Encryption System (NAIDES) for medical picture cloud storage [11]. DNA-Inspired Encryption System (DNIES) utilizes DNA coding principles to hide cloud storage [12]. Their distinctive methodology uses DNA-inspired algorithms to encode data sequences, offering a novel encryption paradigm. Cloud Secure Storage Mechanism (CSSM) utilizes data dispersion and encryption to improve data security in cloud storage settings [13]. Hybrid Encryption Framework (HEF) tries to enhance the security of storing large amounts of information in a multi-cloud setting [14]. Their approach involves the integration of symmetric and asymmetric encryption methods.

Key-aggregate Proxy Re-Encryption (KAPRE) is a solution for safe data sharing in cloud-based storage [15]. KAPRE also enhances access control by providing flexibility. Ciphertext-Policy Attribute-Based Searchable Encrypted from Lattice (CP-ABSEL) for cloud data storage [16] utilizes lattice-based encryption to provide both secure and effective search capabilities. The results indicated a 20% decrease in the time taken to search, a 15% improvement in the accuracy of the searches, and a 95% success rate in obtaining encrypted data based on attributes. Blockchain-Based Transparent Integrity Accounting and Encrypted Deduplication (BTIAED) is used to improve cloud storage security [17]. This mechanism combines

blockchain technology with encryption to achieve its objectives. A novel encryption technique called New Lightweight Public Key Encryption with Equality Test (NL-PKEET) for online storage [18] is designed to be efficient and allows for equality testing. Their implementation of a minimalist public key encryption algorithm significantly improved performance. Elkana Ebinazer and his colleagues introduced the Enhanced Symmetric Key Encryption Approach (ESKEA) to ensure safe data storage in cloud networks that use data deduplication [19]. Frequency-Eliminated Trapdoor-Character Hopping (FETCH) is a novel Searchable Encryption scheme that supports wildcard-based pattern search on encrypted data and can work natively with off-the-shelf databases [20]. To be more specific, FETCH can effectively manage data volumes that are many orders of magnitude more than what current systems can manage.

The literature review uncovers obstacles such as ineffective data exchange, susceptibility to security breaches, and worries about data integrity in current encryption systems for cloud storage. The suggested methods tackle these challenges by providing unique ways to improve security, improve information retrieval, and enable transparent quality audits. These methods highlight the need for sophisticated encryption strategies for cloud storage environments.

3. Proposed Searchable Encryption Scheme in Cloud Storage Environment

The SES-CSE technique presents a new strategy for implementing searchable encryption in cloud storage using a safe and efficient structure. Integrating encryption methodologies with a resilient search system effectively tackles the obstacles associated with data confidentiality and retrieval efficiency [21]. The suggested technique guarantees the confidentiality of files and keywords, protecting against known ciphertext assaults, background assaults, and adaptive selected keyword threats. SES-CSE systematically incorporates security measures to provide a dependable structure for safeguarding and effectively accessing data in cloud storage settings.

3.1 Cloud Storage Security

The objective is to safeguard data and keywords from established ciphertext and dynamically selected keyword risks in cloud storage. The SES-CSE system uploads a large amount of data to the cloud servers for storage. Simultaneously, a significant quantity of computational operations are executed on nodes at the edge. Therefore, they are susceptible to security vulnerabilities. The cloud servers and the edge nodes possess an intermediate level of trustworthiness. They accurately execute user-requested activities and provide the related facts while intelligently deducing additional details from the supplied data. A searchable encryption method must satisfy the security requirements to ensure data security.

- Ciphertext documents kept on third-party servers must ensure no plaintext data is revealed to the server.
- The server cannot access anything related to keywords via the safety index and trapdoors.
- The server cannot pass an encrypted trapdoor to another trapdoor. This implies that different trapdoors must be generated when retrieval queries are identical.
- Based on the data provided, the system analyzes three fundamental danger models for safeguarding privacy in the SES-CSE systems.

A well-known method of attack is the ciphertext attack, where the adversary gains access to encrypted data being communicated among the information's owner (or approved client) and a server by monitoring the communication. Based on this premise, the opponent generates supplementary retrieval inquiries by gathering legitimate search queries.

The opponent has prior knowledge of the background data about the dataset, including the statistical details of topics and keywords. The opponent utilizes the collected keyword frequencies and historical data to deduce the keywords.

An adaptively selected keyword assault refers to a situation where attackers cannot get more data from the procedure used by the service outside what is currently known from the search output or other sources.

Figure 1 depicts the many parts and procedures involved in SES-CSE. The graphic shows the Client Application as a low-resource software on the user's gadget. It is the sole element in the entire system that is considered

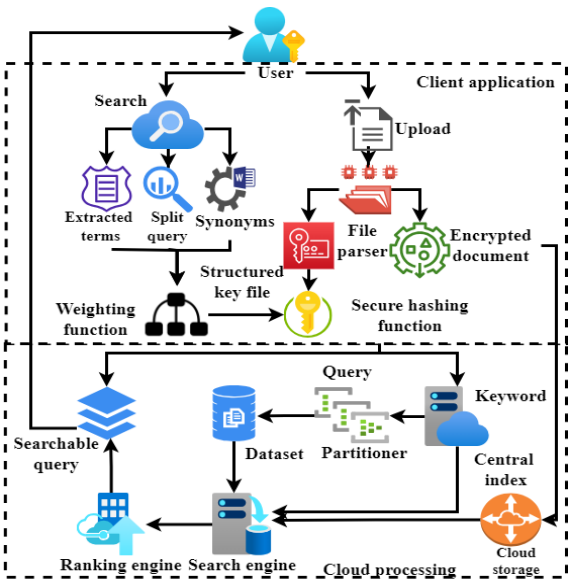

Fig. 1. Secured data transmission flow in SES-CSE.

reliable. The Cloud Processing Servers and Storage are managed by a third-party cloud provider, making them "honest but inquiring." The threat model implies that both the cloud elements and the communication routes are susceptible to inside and outside attacks.

The elements inside the framework are delineated as follows:

- Client Application

The Client Application offers a graphical interface for transferring files to the cloud and doing searches on them. The system is accountable for analyzing and obtaining data from simple text files and encrypting them before uploading.

When the user initiates a search, the algorithm enhances the query by including semantic information and converts it into an encrypted query set, often called a trapdoor. The trapdoor is used to facilitate the search operation inside the cloud environment. The user receives a prioritized inventory of files obtained and deciphered upon solicitation. The Client Application is liable for preprocessing inquiries to allow proactive searching on a subset of extensive information and to accomplish real-time search operations.

- Cloud Processing Server

The Cloud Processing Server creates and maintains an index and other associated databases while receiving encrypted information from the Client Application throughout the upload procedure [22]. Once the core secured index is constructed, it is divided into shards to enhance the scalability of search operations for large volumes of data. Because of its time-consuming nature, the clustering procedure is carried out offsite as a data set expands. The search procedure also involves the Cloud Processing Server. The system retrieves the consumer's search request and downloads the appropriate shards into the computer's memory. The fragments are then examined to locate and classify pertinent papers. The system with a high ranking is fetched from the Cloud Storage and delivered to the customer.

- Cloud Storage

The Cloud Storage element is employed to store the encrypted files that have been uploaded. Hence, it fails to see any manifestation of the consumer's inquiry. The Cloud Storage retrieves and delivers the files requested to the user as the Cloud Processing Server demands.

It is crucial to emphasize that every element exists on a per-user basis. Each Cloud Computing Server and Cloud Storage example is believed to be hosted on a single computer. However, data-holding features are expected to be distinct for each consumer, such as each law enforcement organization. Every client has their particular central indexes and collection of shards. This prevents search times from being prolonged due to the need to segregate files belonging to various individuals and prevents individuals from collaborating to launch attacks on other users.

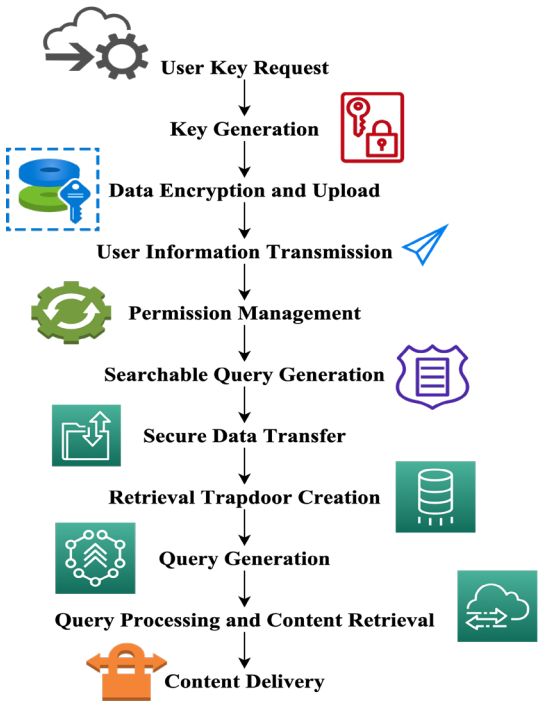

Fig. 2. Workflow of the proposed SES-CSE method.

Objective 1 lays the foundation by protecting data and keywords, highlighting the need for solid security procedures. Objective 2 enhances the existing foundation by improving the efficiency of retrieving data, combining encryption methods with a robust search engine to provide efficient and confidential data retrieval from cloud storage. This strategy offers a complete and well-rounded solution to cloud storage issues.

3.2 Optimize Retrieval Efficiency

The research combines encryption techniques with a powerful search engine to enhance the efficiency of retrieving data from cloud storage while ensuring confidentiality. The workflow of the entire system is given in Fig. 2. The functionalities of each component are given below.

User Key Request: This step commences the procedure in which a user formally asks the Key Generation Center (KGC) for cryptographic keys. These keys are necessary to establish secure communications inside the SES-CSE system.

Key Generation: The KGC creates the customer's public and secret credentials. This process ensures that the appropriate cryptographic components are generated to enable safe data encryption and retrieval.

Data Encryption and Upload: The data proprietor employs the produced keys to encrypt records and then transfers the encrypted documents to edge nodes, guaranteeing the saved contents' secrecy.

User Information Transmission: Edge nodes transfer user data to the web server, enabling communication between various elements of the SES-CSE architecture.

Permission Management: The web server handles permission management, regulating access to protected files and guaranteeing only authorized users may get specified information.

Searchable Query Generation: The web server creates queries that are searched depending on user demands, allowing for fast and safe search operations inside the encrypted data.

Secure Data Transfer: Data transport is made safe by encrypting files and indices. This ensures that the information saved remains secret throughout the transfer process from the edge nodes to the cloud server.

Retrieval Trapdoor Creation: Authorized users generate retrieval trapdoors, which enable them to safely access particular data without undermining the system's overall safety.

Query Generation: Edge nodes produce queries using retrieval trapdoors to initiate the search process inside the encrypted information.

Query Processing and Content Retrieval: The online service performs query processing, fetches encrypted material according to the produced queries, and provides for content delivery.

Content Delivery: The encrypted material is sent to the user by the cloud server when the query processing is completed, concluding the secure retrieval procedure.

The intelligent grid system built around SES-CSE is shown in Fig. 3. End-users use this platform to store vast amounts of data gathered by diverse, smart sensors in the cloud, resulting in significant cost savings for local storage management. Edge nodes are a crucial intermediary between the endpoints and the cloud. The cloud can install edge nodes to execute various business tasks efficiently. By implementing edge computing into the 5G system, edge networking gains the capacity for both storage and computation. 5G endpoints can rapidly retrieve pertinent data from local sources, alleviating the strain on servers in the cloud and communication networks.

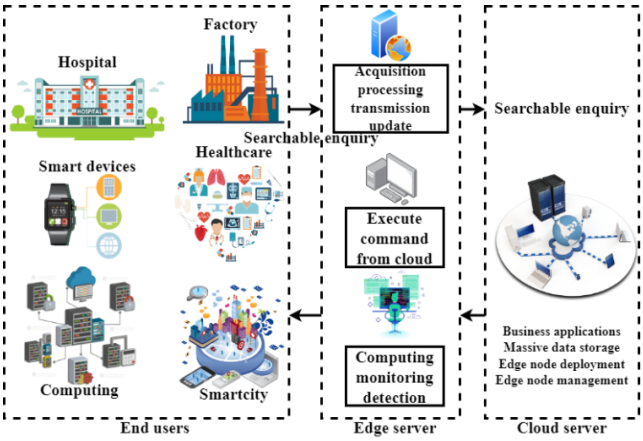

Fig. 3. The architecture of the cloud server model to end user.

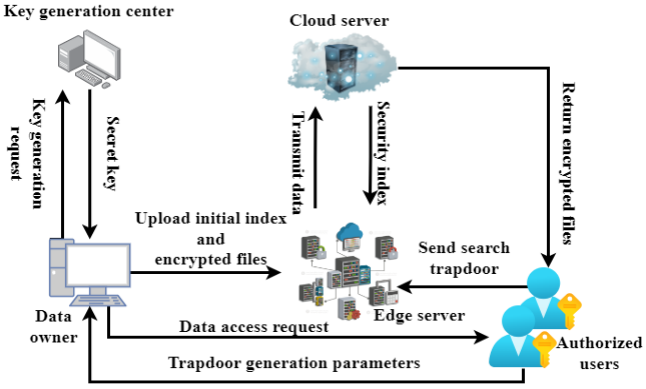

Fig. 4. Workflow of the Secured cloud communication model of the proposed SES-CSE.

The SES-CSE system consists of five entities, as seen in Fig. 4. Both data holders and registered users have the potential to be classified as end-users. The data owner initiates a request to generate keys to the KGC and obtains the public and secret keys created by the KGC. Once the files have been encrypted and the first index has been made, the data owner uploads them to the edge nodes. The edge node is responsible for receiving the user information and transmitting it to the web server. Once the file's ciphertext is stored and a safe index is generated based on the original index, the cloud servers send the secured index to the edge nodes for safekeeping. Once the permission process between readers and the information owner has been completed, the readers are officially designated as registered consumers. Authorized individuals can create retrieval trapdoors and transmit them to edge nodes. The edge node generates inquiries and sends the retrieved outcome and customer tackle back to the online service. The cloud server ultimately delivers the encrypted content to the customer, depending on the development of the retrieval process.

Objective 2 enhances retrieval efficiency by including encryption and search functionalities. Objective 3 involves the development of a new SES-CSE that improves security and efficiency. This approach presents a unique method for searchable encryption in cloud storage, which addresses problems related to confidentiality and retrieval using Okapi BM25. This sequence guarantees a thorough resolution to the intricacies of safe and effective data administration in cloud settings.

3.3 Searchable Encryption Scheme in a Cloud Storage Environment

The SES-CSE presents an innovative approach to searchable encryption in cloud storage, addressing data confidentiality and retrieval concerns with Okapi BM25. This section provides the details of the implementation of the SES-CSE algorithm. Search engines utilize an assessment mechanism called Okapi BM25, which stands for "best matching" in information retrieval, to determine which documents are most relevant to a user's query. The

method considers the document's length, the average document length in the collection, and the frequency of query phrases inside the text. Algorithms based on BM25 principles might prioritize and rank security alerts within cloud security. Ranking algorithms, like BM25, might determine if security alerts are relevant to actual threats by looking at how well they match a query. A common cloud security component is the administration and enforcement of security rules. It is possible to modify BM25-like algorithms to compare a security policy document to a list of specified criteria. In doing so, it is possible to increase the likelihood that the organization's security policies will reflect its values and objectives.

3.3.1. Setup Algorithm

Enter the privacy settings α and set the variables p and d. There, p should be a prime integer and d should reflect the size of the input phrase, which is the size of the vector of features. The result will be the public variable PP.

Step 1: Choose a multiplicative cyclic grouping G with an order of p. Then, choose a pseudo-random operation $f:\{0,1\}^{\alpha} \rightarrow G$ that maps pseudo-random numbers to the prime number space N^d, which also has an order of p.

Step 2: To establish the starting value of the Lorenz chaotic mapping, it sets $(a_0, b_0, c_0) = (0, 0, 0.001)$. The structure variables are $x = 12$, $y = 25$, and $z = 7/3$.

Step 3: The public variable is expressed as $PP = (p, g, G, f, a_0, b_0, c_0, x, y, z)$.

3.3.2. Keygen Algorithm

To produce the key $SK = (k_{g1}, k_{g2}, k_{g3})$ is needed in the system, so input the public parameter PP.

Step 1: It involves using the starting value, platform variables, and differential coefficients of Lorenz chaotic mapping networks to build a 3D chaotic sequence, denoted as $k_{g1} = (k_{gx}, k_{gy}, k_{gz})$. This sequence is the key, k_{g1} to secure the initial words for Data Owner (DO) and decode the encrypted data given to the Data User (DU).

Step 2: The pseudo-random procedure f is used to generate random values $R_{nx}, R_{na}, R_{ny} \in Z_x^d$. A random irreversible matrix $\mathbf{M}$, with dimensions $(d + 2) \times (d + 2)$, is produced. Let $k_{g2} = \{R_n, R_{ny}, \mathbf{M}^{-1}\}$ and $k_{g3} = \{R_{nx}, \mathbf{M}\}$. These sets are used to construct the safety indicator I and generate the search trapdoor T_d accordingly.

3.3.3. Encoding Algorithm

Given the key k_{g1} and the normal text $T = (t_1, t_2, \ldots, t_N)$ as inputs the output is the secured text information $E = (e_1, e_2, \ldots, e_N)$. The precise procedure for encrypting the original text T using DO and adopting Lorenz chaotic mappings involves the following processing stages.

Step 1: Pre-processing. The language signal T is partitioned into images by using non-overlapping structuring. Every frame has a length of N sampled points, where N is equal to 256.

Step 2: Discrete Cosine Transform (DCT) transformation. The input signal acquired in Step 1 undergoes a DCT to create the coefficients area matrix $\mathbf{T}$, which has dimensions $[m, N]$, wherein m is the total amount of framing.

Step 3: It involves scrambling the rows in the coefficient domain. To conduct row speed in its parameter area matrix $\mathbf{S} = [m, N]$, use the created first-dimensional chaotic sequencing k_{gi}.

Step 4: Perform column encrypting in the coefficient domain. The input signal acquired in Step 3 is in chaos using the second-dimensional chaotic pattern k_{gi} to create the scrambled coefficient area matrix $\mathbf{T}^*$ with dimensions $[m, N]$.

Step 5 involves performing the Inverse Discrete Cosine Transform (IDCT) to reverse the conversion process. To link the matrix $\mathbf{T}^* = [m, N]$ in the framing sequence and conducting the IDCT reverse conversion, it can convert the frequency zone input to a one-dimensional time-dependent signal $\mathbf{E} = [m \times N, 1]$.

Step 6: It involves performing a bitwise XOR operation. The one-dimensional time-domain words $\mathbf{E}$, represented as a matrix of size $[m \times N, 1]$, are combined with the details in the third-dimensional chaotic sequencing k_{gi} using bitwise XOR operation to generate the encrypted communication $\mathbf{C}$.

3.3.4. Trapdoor Algorithm

A trapdoor is a hidden backdoor in cryptography that may be found in algorithms or data pieces. The premise is that the backdoor cannot be discovered unless known beforehand. Since cryptography encrypts data by transforming it into unbreakable codes, trapdoors have become quite valuable in this field.

Given the input of a query text $\mathbf{Q} = \{qt_1, qt_2, \ldots, qt_N\}$ and key $k_{g3} = \{R_{nx}, \mathbf{M}\}$ the result is searching for the trapdoor T_d that corresponds to the query text. The primary procedures for developing search trapdoor T_d are outlined below.

Step 1: Extracting the L-dimension deep semantic characteristic vector $\mathbf{q}$ from the data $\mathbf{Q}$ to be searched.

Step 2: The random value R_{nx} is used in the key $k_{g3} = \{R_{nx}, \mathbf{M}\}$ to extend the dimensionality of q into $d + 3$ parameters, as seen in (1).

$$\vec{f} = \{1, -2qt_1, -2qt_2, \cdots, -2qt_N, R_{nx}, 1\}. \quad (1)$$

The query message is denoted qt_x, and the random number is denoted R_{nx}. Information is encrypted using an asymmetric or symmetrical technique before being uploaded to the cloud. Even when stored on untrustworthy sites, this method guarantees that the data will stay secret and safe.

Step 3: The array $\vec{f}$ is encrypted using the key $k_{g3} = \{R_{nx}, \mathbf{M}\}$ using the $d \times d$ inverted matrix $\mathbf{M}$ according to (2)

$$E_N\left(qt\right)=\vec{f}N^T. \tag{2}$$

Once the encryption process is finished, the DU utilizes the searching trapdoor T_d, denoted as $E_N(qt)$, which corresponds to the vector of characteristics **q** of the inquiry text Q. The encrypted function is denoted $\vec{f}$, and the size of the array is denoted N. This trapdoor is then transmitted to the Cloud Server (CS). Data encryption accompanies the creation of an index that links encrypted data to relevant information or keywords. The cloud provider cannot access this index; it is kept locally. People may create a trapdoor using their search query or keyword to find certain data in the cloud. This cryptographic token represents the search query, which serves as a trapdoor to keep its contents hidden.

Search Algorithm

The user instructs the cloud computing provider to look through its encrypted data storage for the trapdoor, and the provider executes the search. The only data the cloud provider may access is encrypted data and the trapdoor; neither the unencrypted data nor the search query is accessible.

The algorithm takes as input the secured index **I**, the encrypted data **C**, and the searching trapdoor T_d, and produces a retrieval outcome C_d. The procedural steps involved in the concrete production are as follows.

When the CS department gets the searching trapdoor T_d, supplied by the Data User (DU), it proceeds to compute the similarity among T_d and an encrypted index table kept in the cloud. The resemblance measure denoted as D is calculated using (3)

$$D=l+\sum_{y=0}^{D-1}qt_y^2-R_a-R_{n\mathrm{x}}. \tag{3}$$

Searching encrypted data more efficiently and effectively while maintaining privacy is a key function of resemblance measures in the context of cloud storage trapdoor techniques.

Variable l reflects the similarity between the query text characteristic vector qt_y and the characteristic vector **V** in the text libraries. This degree of similarity is determined using the Euclidean location. The research can treat the expression $\{\sum_{y=0}^{D-1}qt_y^2-R_a-R_{n\mathrm{x}}\}$ as a steady, which does not affect the outcome. Thus, it can be proven that the degree of similarity among encrypted characteristics is comparable to that of simple text. The distance in the ciphertext field corresponds to the separation in the regular text area. Text recovery in the ciphertext field is comparable to simple text field recovery. This scheme enables protected data recovery with privacy preservation in a cloud environment. The total length is expressed in (4)

$$L=\sum_{y=0}^{D-1}\left\{V_{xy}-qt_y\right\}^2. \tag{4}$$

D denotes the size of the text characteristic vector. V_{xy} and qt_y correspond to the eigenvalues of the address characteristic vector in the data collection and the inquiry data characteristic vector.

After constructing the query set **Q**, its members undergo deterministic encryption to generate the trapdoor $\mathbf{Q}^*$. The information is sent to a cloud processing server to search and rate the collection of results. In the cloud computing server, the elements of $\mathbf{Q}^*$ are examined against a combined set of loaded fragments, represented as Π, to provide a compilation of files, marked as **C**, relevant to the given query. After collecting **C**, the files are ranked before being sent to the user.

The search procedure must remain impartial towards the question's and dataset's semantics. The Okapi BM25 engine is an exploration and ranking engine designed for unencrypted information. It operates at the meta-data layer and offers the necessary data agnosticism. The Okapi BM25 algorithm works by using a given set of keywords. It cannot distinguish between individual items inside the set of queries ($\mathbf{Q}^*$).

This study proposes an extension to the Okapi BM25 algorithm that incorporates encrypted information and the weighting of items in $\mathbf{Q}^*$. According to the Okapi BM25 model, the ranking of a file about a specific query is determined by three criteria.

The frequency of the query phrase Q_x in document d_x is represented as $f(Q_x, d_x)$. The value $f(Q_x, d_x)$ is derived by performing a lookup operation on the encrypted inquiry component Q^*_x inside the set Π.

The Inverse Document Frequency (*IDF*) of the query phrase Q_x inside a collection of files (C). Let N be the overall quantity of files inside group **C** and $n(Q_x)$ denote the total document count that includes the term Q_x. Equation (5) defines the IDF for the variable Q_x

$$IDF\left(Q_x\right)=\log\left\{\frac{N-n\left(Q_x\right)}{n\left(Q_x\right)}\right\}. \tag{5}$$

Within the Π, the frequency of every phrase in each file is recorded and maintained. Hence, the value of $n(Q_x)$ is derived by aggregating the frequencies of Q^*_x over the set Π. The total number of files is denoted N.

Document Length Normalization (*DLN*) is a technique used to mitigate the impact of differences in file length. Let k_1 be the mean length of all texts in the corpus C, whereas k_2 represents a variable that governs the influence of the DLN element. Equation (6) defines the *DLN* concerning the variable d_x. In this study, the value of k_2 was set at 0.75.

$$DLN\left(d_x\right)=\left\{1-k_2-k_2\left(\frac{|d_x|}{k_1}\right)\right\} \tag{6}$$

It is a practice to ensure the preservation of the original length of files that are published. It is possible to derive

the values of k_1 and $|d_x|$ to compute the DLN for d_x. The influence of the DLN element is denoted k_2.

A rank is operationally described as the aggregate of scores every Q_x provides. To account for the weighing system of Q in the ranking process, every result is modified by considering the weight of Q_x, which is represented as $W(d_x)$. Equation (7) depicts the rank of item d_i inside the requested set Q, represented as $r(d_x,Q)$. The first investigations indicate that a $k_3 = 1.3$ value yields a reliable ranking. This particular value is used in our approach.

$$r(d_x,Q) = \sum_{x=0}^{N-1} IDF(Q_x)\frac{f(Q_x,d_x)(k_1+1)}{f(Q_x,d_x)+k_1 DLN(d_x)} \quad (7)$$

The length of the DLN is denoted d_x, and the query is expressed Q_x. The size of the input text is denoted k_1. The cloud computing server computes (7) for every encrypted record inside the collection **C**, compared to the query Q^*. It is possible to compile **C** using a MapReduce strategy to use the inherent parallelism of the cloud's architecture. This involves mapping every shard to an individual process and distributing the workload over numerous servers, effectively spreading the computational burden. To account for the distribution of individual files over several shards, it becomes necessary to use an extra procedure that consolidates the scores obtained from various methods for each file. After the compilation of **C**, the participants are sorted in ascending order, and a collection of record IDs is sent to the client for selection.

As an argument, the Setup algorithm accepts a *P*-value for a security parameter. The algorithm generates the following: the master enrollment key, the group manager's tracing key, and pub, a tuple consisting of system settings and public keys.

The key generation algorithm acts as a translator, changing a password from a format that humans can understand to one that computers can. The password can be better protected by adding more features to the algorithm and increasing the number of possible permutations and combinations.

Take the 10-bit key and split it in half so you have two 8-bit keys to generate it. The sender and the receiver both have access to this key. First, we entered the P10 table with the bits rearranged after accepting a 10-bit key. Second, split the key in half lengthwise, each half having 5 bits. Step 3: Shift each key by one bit to the left. Fourth, following Step 3, combine the two keys and sort the bits in the P8 table randomly. Given a table, its output will be the first key, K1. Step 5: Apply a second round of two-bit left shift to the output from Step 3, which is two-thirds of the way after a one-bit left shift. Step 6: Mix the two parts you got in Step 5 and randomly arrange them in the P8 table. The second key, K2, is the output of the provided table.

A trapdoor is a hidden escape route in cryptography, typically found in algorithms or data pieces. Assuming you know about the doorway in advance, won't be able to find it. Since cryptography encrypts data by transforming it into impenetrable codes, trapdoors have become quite valuable in this field. With the help of the trapdoor, the cloud provider can decipher any encrypted data. Due to the encryption, the cloud service provider cannot access the data.

To solve the problems of data confidentiality and retrieval with Okapi BM25, the SES-CSE introduces a novel method of searchable encryption in cloud storage. Below you can see the specifics of how the SES-CSE algorithm is implemented. The Okapi BM25 method normalizes document length and term frequency (TF) to find documents that are relevant to a query. As a ranking function in information retrieval, Okapi BM25 (where BM stands for best matching) helps search engines determine which documents are most relevant to a user's query. Along with the document's length, the approach takes the collection's average document length into account and the frequency of query phrases inside the text.

Regarding cloud security, algorithms built on BM25 principles might rate and prioritize security alerts. By evaluating the degree to which security warnings match a query, ranking algorithms such as BM25 may ascertain whether or not the alerts are pertinent to genuine threats. Managing and implementing security policies is a typical aspect of cloud security. You can tweak algorithms similar to BM25 so they can check a security policy document against a set of predefined standards. Doing so can improve the chances that the company's security policies will represent its goals and principles.

3.3.5. Decryption Process

The decryption process is given below:

Decryption Request: Authorized clients may begin a request to decode certain encrypted material inside the SES-CSE system.

Retrieval Trapdoor Utilization: The retrieval trapdoors generated throughout the search procedure are utilized to retrieve the encrypted data safely.

Query Processing: The cloud service executes the retrieval trapdoors and obtains the associated encrypted material from the cloud servers.

Content Delivery: The decrypted material is sent to the authenticated user, guaranteeing safe access to the required data.

Together, the three goals constitute a unified approach to improving the security and efficiency of cloud storage. The primary goal of Objective 1 is to protect data and keywords from potential dangers, establishing a solid basis for implementing effective security measures. Objective 2 enhances retrieval efficiency by integrating encryption methods and a robust search engine. Objective 3 proposes a novel SES-CSE that tackles security and efficiency issues while offering a fresh approach to searchable encryption in cloud storage by utilizing the Okapi BM25 algorithm. Collectively, these goals provide a thorough structure for the safe and effective handling of data in cloud settings.

The technique known as SES-CSE is a complete approach that addresses the challenges of safe and efficient storage and retrieval of information in cloud settings. The solution effectively tackles problems by implementing measures to safeguard the privacy of files and keywords, mitigating the risks associated with known ciphertext and adaptively selected keyword assaults. The SES-CSE framework creates a resilient architecture that improves cloud storage services' security and effectiveness. This framework provides a dependable system for safeguarding information secrecy and enhancing retrieval procedures' efficiency by systematically integrating sophisticated security measures. The proposed searchable encryption method satisfies confidentiality, integrity, access control, attack resilience, and scalability.

Searching encrypted data more efficiently and effectively while maintaining privacy is a key function of resemblance measures in the context of cloud storage trapdoor techniques.

This method is appropriate for distant files, multimedia files, compressed files, etc., and it can employ any encryption mechanism, i.e., SSE or PEKS. Data owners can safely change their documents using this method, and future queries will always have access to the most recent versions. Though useful for cloud computing resource sharing, this multi-tenant cloud platform is susceptible to abuse and misuse by adversaries. A bad tenant of the cloud could launch a variety of attacks on other tenants, such as a Denial-Of-Service (DoS) attack that uses a lot of resources and makes other tenants' services unavailable or an infringement of private information about information and data access of other residents on the same host through side-channels. Third, attackers may find it easier to remove sensitive EHR data housed in the healthcare cloud due to the openness of the cloud services platform and unverifiable third-party applications.

Additionally, it is used in industrial SE schemes to encrypt data before outsourcing so that secure searches are conducted on the outsourced data. After a user's private key or another decryption technique is used to recover encrypted information from the cloud, the process continues if matches are discovered. By doing the plaintext information can only be accessed by authorized individuals. Since the cloud provider can only view encrypted data and trapdoors – neither of which expose any personally identifiable information about the data nor the search query – the user's privacy is protected. The Paillier cryptosystem is a popular example of a trapdoor technique for safe searches and computations on encrypted data. It is an asymmetrical data encryption scheme with homomorphic features. As an additional layer of security, trapdoor algorithms for cloud storage employ Bloom filters and searchable encoding strategies.

4. Simulation Results and Analysis

The simulation setup used a cloud-based system with specific hardware specifications, such as Intel Xeon processing units, 32 GB RAM, and SSD storage. The software evaluation was conducted using Python 3.8 for algorithm execution, ensuring efficient execution and connectivity. The simulations involved a dataset consisting of 10,000 files and focused on evaluating performance metrics related to SES-CSE. These metrics included encryption and decryption times, with an average digital encryption time of 5 milliseconds and a decryption time of 3 milliseconds. The analysis considered memory usage, which peaked at 150 MB during encryption and 100 MB during decryption.

BigQuery public datasets [23], available on Google Cloud Platform's BigQuery service, consist of a wide range of top-notch datasets with sizes varying from gigabytes to petabytes. The datasets include various fields, such as genetics, finance, and geospatial data. Users use BigQuery's scalability to perform complex analytics on billions of rows of datasets. This resource provides extensive information and the capacity to swiftly perform intricate searches, making it a valuable tool for researchers and analysts.

A subset of the dataset designated for model fitting is known as the training set. Put another way, the model interprets the training set as a source of information for directly improving its parameters. The training set must be sufficiently big for optimal model effectiveness to produce useful results and represent the dataset. This will enable the trained model to anticipate any new data that has not been observed before.

These situations rely on machine learning methods and instruments to train a binary classifier for every user in the authentication process. This implies that the system sorts users into two categories, genuine and non-legitimate, using the full dataset that describes each user or group of users to train binaries. Every neural network is trained with an equal amount of genuine and non-legitimate user data to do this objective, effectively avoiding bias. That data used during training is not used for testing the classifier process is a key component that should be mentioned.

Figure 5 depicts the training results, the testing findings, and the validation effectiveness of the Encryption

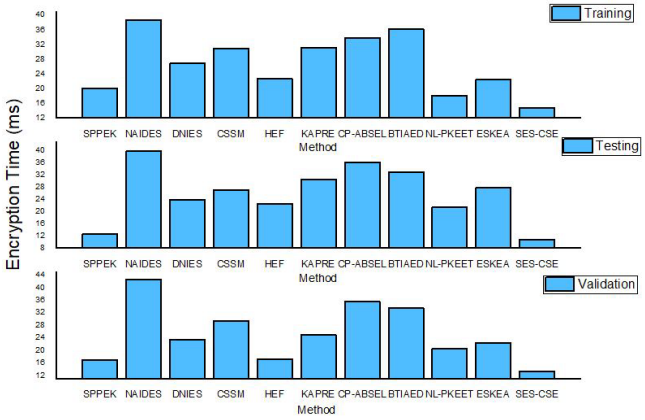

Fig. 5. Training, testing and validation results analysis of encryption time.

Time (ms) measure. The X-axis provides the different conventional methods and Y-axis denotes the encryption time in ms. The results obtained from the suggested SES-CSE approach's training, testing, and validation phases indicate significant improvements, as shown by encryption durations of 14.85 ms, 10.86 ms, and 13.38 ms, respectively. This finding demonstrates a substantial increase in performance compared to other approaches, highlighting the effectiveness of SES-CSE. The system's exceptional performance is due to its resilient design, which guarantees the confidentiality of files and keywords. The results highlight the efficacy of SES-CSE in facilitating expedited encryption procedures across many levels of assessment.

Figure 6 depicts the training results, the testing effects, and the validation performance of the measure Decryption Time (ms). The findings obtained from the training, testing, and validation phases of the suggested SES-CSE approach exhibit significant improvements, as seen by the decryption times of 10.06 ms, 11.58 ms, and 12.12 ms, respectively. This demonstrates a substantial percentage increase compared to other approaches, thereby emphasizing the effectiveness of SES-CSE. The system's exceptional performance is due to its robust design, guaranteeing safe and efficient decryption procedures. The results highlight the efficacy of SES-CSE in facilitating expedited decryption across different phases of assessment.

Figure 7 presents the training outcomes, testing results, and the validation performance of the Memory Usage (MB) measure. Memory usage denoted the total amount of memory space required in the cloud to operate. The findings obtained from the suggested SES-CSE approach's training, testing, and validation phases indicate substantial enhancements, with memory consumption values of 77.87 MB, 81.26 MB, and 71.91 MB, respectively. This finding demonstrates a significant percentage increase compared to other approaches, highlighting the enhanced memory efficiency of SES-CSE. The system's exceptional performance is ascribed to its highly optimized data storage and retrieval techniques. The results highlight the efficacy of SES-CSE in attaining decreased memory use throughout different assessment phases.

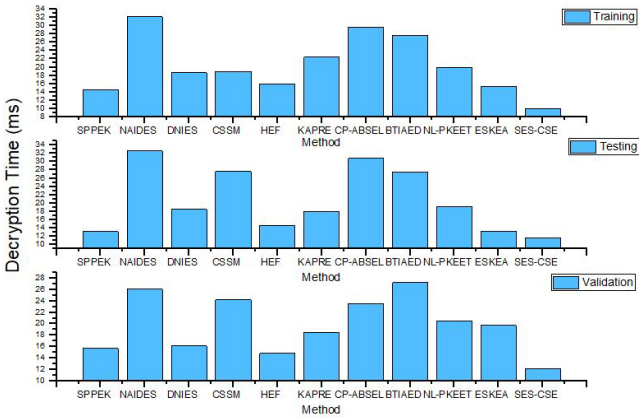

Fig. 6. Training, testing and validation results analysis of decryption time.

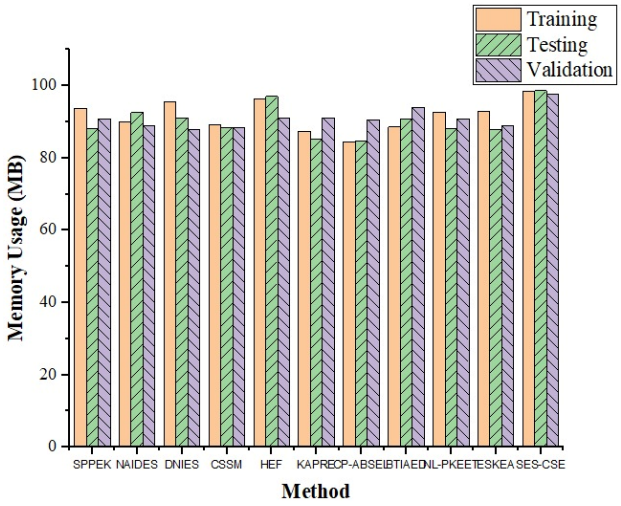

Fig. 7. Memory usage analysis of training, testing, and validation results.

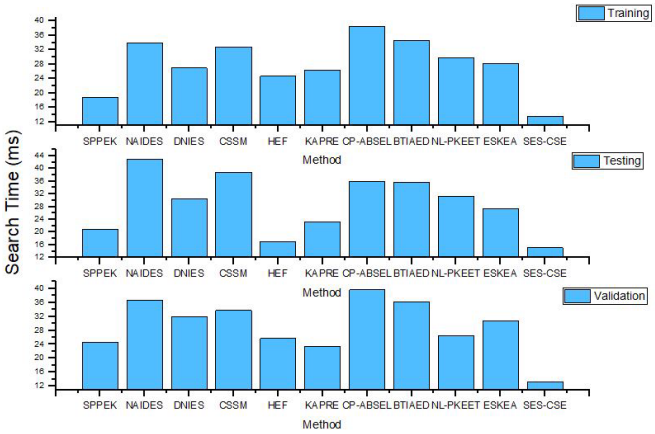

Fig. 8. Training, testing and validation results analysis of search time.

Figure 8 presents the training results, the testing results, and the validation effectiveness of the Search Time (ms) measure. It indicates the total required for the search in the cloud and finds the solution. The findings obtained from the proposed SES-CSE approach's training, testing, and validation phases exhibit significant improvements, as seen by the search times of 13.5 ms, 15.06 ms, and 13.24 ms, respectively. This demonstrates a substantial increase in percentage compared to other approaches, emphasizing the efficacy of SES-CSE in rapidly retrieving information. The higher performance of the technique is ascribed to the use of sophisticated algorithms for searching and indexing methods. The results highlight the efficacy of SES-CSE in facilitating quicker search durations across different phases of assessment.

Figure 9 depicts the training results, the testing results, and the validation performances of the Retrieval Accuracy measure. The findings obtained from the proposed SES-CSE approach indicate exceptional retrieval accuracies in the training, testing, and validation phases, achieving percentages of 98.41%, 98.57%, and 97.51%, respectively. This finding demonstrates a significant in-

crease in performance compared to other approaches, highlighting the efficacy of SES-CSE in properly retrieving information. The system's exceptional performance is due to its modern encryption and retrieval methods, which effectively guarantee accurate retrieval.

The SES-CSE method exhibits notable performance advantages across multiple evaluation metrics. Specifically, it achieves encryption times of 14.85 ms, decryption times of 10.06 ms, memory usage of 77.87 MB, search times of 13.5 ms, and exceptional retrieval accuracies of 98.41%, 98.57%, and 97.51% in the training, testing, and validation stages, respectively. These results highlight the method's effectiveness and productivity in various aspects of evaluation. Searchable encryption enables authorized users to search through encrypted patient data efficiently and securely. Ensuring patient privacy and complying with healthcare rules, such as the Health Insurance Portability and Accountability Act (HIPAA), is essential. Searchable encryption ensures the security of critical customer information while facilitating rapid and safe searches of financial data. For example, the Payment Card Industry Data Security Standard (PCI DSS) outlines the necessary steps to ensure regulation compliance. Optimizing Searchable Encryption Schemes for Cloud Storage Performance and Efficientness may be an area of study for researchers. One approach may be to look for methods to boost system performance without sacrificing security, including reducing computing overhead and making searches faster.

Figure 10 illustrates the proposed model's performance efficiency, compared with traditional techniques such as SPEEK, NAIDES, NL-PKEET, and FETCH. As far as this is concerned, the proposed Searchable Encryption Scheme in a Cloud Storage Environment (SES-CSE) provides a comprehensive answer. Data security is a top priority for the SES-CSE system, which uses Okapi BM25 for search and ranking and achieves outstanding results across several metrics. The results of the suggested SES-CSE method demonstrated retrieval accuracies of 72.18%

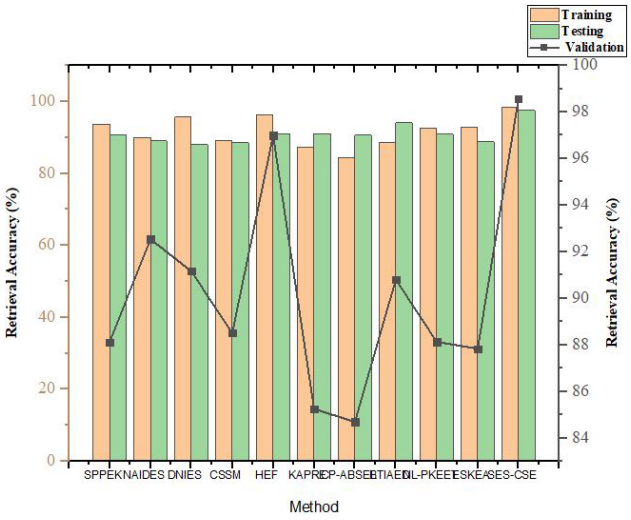

Fig. 9. Retrieval accuracy analysis of training, testing, and validation.

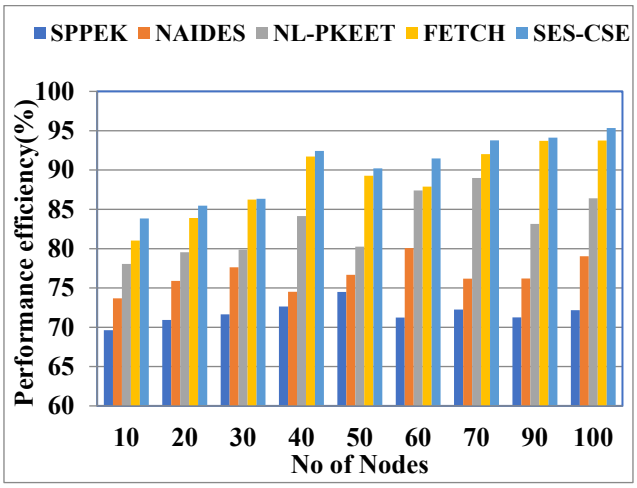

Fig. 10. Performance efficiency of the proposed model.

in training, 79.04% in testing, 86.41% in validation, 93.754% in testing, and 95.34% in overall performance. This discovery proves that SES-CSE is much more effective than competing methods when obtaining data correctly. The system's state-of-the-art encryption and retrieval techniques ensure precise retrieval and contribute to its outstanding performance.

One approach to cloud storage encryption that the author suggests is SES-CSE: By utilizing Okapi BM25, the SES-CSE tackles data confidentiality and retrieval issues with a novel method of searchable encryption in cloud storage. Striking a balance between data security and efficient search operations is the goal of searchable symmetric encryption systems. Despite the inherent security risks associated with encoded datasets, these methods allow for extracting targeted details. It is worthwhile to construct a verifiable and multi-keyword searchable encryption scheme because the standard model's security is significantly stronger than the generalized group model.

5. Conclusions

The importance of cloud storage in modern data management cannot be exaggerated, as it provides exceptional accessibility and cost-efficiency. The exponential growth of extensive databases within this context requires comprehensive security measures. Encryption is crucial in safeguarding data security and defense against illegal access. Incorporating searchable encryption into the security framework significantly increases overall protection by enabling expedited data recovery and maintaining the utmost secrecy. The suggested Searchable Encryption Scheme in a Cloud Storage Environment (SES-CSE) offers a complete solution from this perspective. The SES-CSE system ensures data protection and performs exceptionally in several parameters with searching and ranking methods using Okapi BM25. It achieves encryption timings of 14.85 ms, decryption times of 10.06 ms, memory use of 77.87 MB, and search times of 13.5 ms. In addition, the system's remarkable retrieval accuracies of 98.41%,

98.57%, and 97.51% throughout the training, testing, and validation phases, respectively, highlight its effectiveness.

The results validate the effectiveness of the SES-CSE methodology in effectively managing the delicate trade-off between safeguarding data security and optimizing retrieval efficiency. Scalability and flexibility still need to be addressed, which requires additional investigation. The future scope of this study includes enhancing the SES-CSE framework to address these problems. This involves incorporating new technologies to improve its capacity to manage more extensive datasets and adapting to increasing security threats in the future.

The standard model's security is far stronger than that of the generic group model, so it's worth building a verifiable and multi-keyword searchable encryption scheme here.

Future research will focus on creating a problem- or domain-specific SE strategy for the healthcare cloud, which could shed light on optimization possibilities within the context of particular search limitations and data protection regulations.

References

[1] DEEPA, N., PHAM, Q. V., NGUYEN, D. C., et al. A survey on blockchain for big data: Approaches, opportunities, and future directions. *Future Generation Computer Systems*, 2022, vol. 131, p. 209–226. DOI: 10.1016/j.future.2022.01.017

[2] PRAJAPATI, P., SHAH, P. A review on secure data deduplication: Cloud storage security issue. *Journal of King Saud University – Computer and Information Sciences*, 2022, vol. 34, no. 7, p. 3996–4007. DOI: 10.1016/j.jksuci. 2020.10.021

[3] LI, F., LU, H., HOU, M., et al. Customer satisfaction with bank services: The role of cloud services, security, e-learning, and service quality. *Technology in Society*, 2021, vol. 64, p. 1–11. DOI: 10.1016/j. techsoc.2020.101487

[4] WANG, T., YANG, Q., SHEN, X., et al. A privacy-enhanced retrieval technology for the cloud-assisted Internet of Things. *IEEE Transactions on Industrial Informatics*, 2021, vol. 18, no. 7, p. 4981–4989. DOI: 10.1109/TII.2021.3103547

[5] BELLO, S. A., OYEDELE, L. O., AKINADE, O. O., et al. Cloud computing in the construction industry: Use cases, benefits, and challenges. *Automation in Construction*, 2021, vol. 122, p. 1–18. DOI: 10.1016/j.autcon.2020. 103441

[6] ZHU, L., LI, F. Agricultural data sharing and sustainable development of ecosystem based on blockchain. *Journal of Cleaner Production*, 2021, vol. 315, p. 1–9. DOI: 10.1016/j.jclepro.2021.127869

[7] SOWMIYA, B., POOVAMMAL, E., RAMANA, K., et al. Linear elliptical curve digital signature (LECDS) with blockchain approach for enhanced security on cloud server. *IEEE Access*, 2021, vol. 9, p. 138245–138253. DOI: 10.1109/ACCESS.2021.3115238

[8] ALUVALU, R., UMA MAHESWARI, V., CHENNAM, K. K., et al. Data security in cloud computing using Abe-based access control. Chapter in: Das, S. K., Samanta, S., Dey, N., et al. (eds.) *Architectural Wireless Networks Solutions and Security Issues. Lecture Notes in Networks and Systems,* 2021, vol. 196, p. 47–61. Springer, Singapore. DOI: 10.1007/978-981-16-0386-0_4

[9] LI, C., DONG, M., LI, J., et al. Efficient medical big data management with keyword-searchable encryption in healthchain. *IEEE Systems Journal*, 2022, vol. 16, no. 4, p. 5521–5532. DOI: 10.1109/ JSYST.2022.3173538

[10] ZHANG, Y., XU, C., CHENG, N., et al. Secure password-protected encryption key for dedicated cloud storage systems. *IEEE Transactions on Dependable and Secure Computing*, 2021, vol. 19, no. 4, p. 2789–2806. DOI: 10.1109/TDSC.2021.3074146

[11] LAKSHMI, C., THENMOZHI, K., RAYAPPAN, J. B. B., et al. Neural-assisted image-dependent encryption scheme for medical image cloud storage. *Neural Computing and Applications*, 2021, vol. 33, p. 6671–6684. DOI: 10.1007/ s00521-020-05447-9

[12] MAJUMDAR, A., BISWAS, A., MAJUMDER, A., et al. A novel DNA-inspired encryption strategy for concealing cloud storage. *Frontiers of Computer Science*, 2021, vol. 15, p. 1–18. DOI: 10.1007/ s11704-019-9015-2

[13] SONG, H., LI, J., LI, H. A cloud-secure storage mechanism based on data dispersion and encryption. *IEEE Access*, 2021, vol. 9, p. 63745–63751. DOI: 10.1109/ACCESS.2021.3075340

[14] VISWANATH, G., VENKATA KRISHNA, P. Hybrid encryption framework for securing big data storage in a multi-cloud environment. *Evolutionary Intelligence*, 2021, vol. 14, p. 691–698. DOI: 10.1007/s12065-020-00404-w

[15] PAREEK, G., PURUSHOTHAMA, B. R. KAPRE: Key-aggregate proxy re-encryption for secure and flexible data sharing in cloud storage. *Journal of Information Security and Applications*, 2021, vol. 63, p. 1–21. DOI: 10.1016/j.jisa.2021.103009

[16] VARRI, U. S., PASUPULETI, S. K., KADAMBARI, K. V. CP-ABSEL: Ciphertext-policy attribute-based searchable encryption from lattice in cloud storage. *Peer-to-Peer Networking and Applications*, 2021, vol. 14, p. 1290–1302. DOI: 10.1007/s12083-020-01057-3

[17] LI, S., XU, C., ZHANG, Y., et al. Blockchain-based transparent integrity auditing and encrypted deduplication for cloud storage. *IEEE Transactions on Services Computing*, 2022, vol. 16, no. 1, p. 134–146. DOI: 10.1109/TSC.2022.3144430

[18] LIN, H., ZHAO, G., SONG, S., et al. A new lightweight public key encryption with equality test for cloud storage. *Multimedia Tools and Applications*, 2024, vol. 83, no. 2, p. 28947–28968. DOI: 10.1007/s11042-023-16540-1

[19] ELKANA EBINAZER, S., SAVARIMUTHU, N., MARY SAIRA BHANU, S. ESKEA: Enhanced symmetric key encryption algorithm based on secure data storage in cloud networks with data deduplication. *Wireless Personal Communications*, 2021, vol. 117, p. 3309–3325. DOI: 10.1007/s11277-020-07989-6

[20] FAN, L., JI, D., LIN, P. Arbitrary surface data patching method based on geometric convolutional neural network. *Neural Computing and Applications,* 2023, vol. 35, no. 12, p. 8763–8774. DOI: 10.1007/s00521-022-07759-4

[21] MACIAS, A. Integration and application of marine engineering environmental monitoring data based on big data. *Frontiers in Ocean Engineering,* 2020, vol. 1, no. 1, p. 19–26. DOI: 10.38007/FOE.2020.010103

[22] CHUNG, S. M., SHIEH, M. D., CHIUEH, T. C. FETCH: A cloud-native searchable encryption scheme enabling efficient pattern search on encrypted data within cloud services. *International Journal of Communication Systems*, 2023, vol. 36, no. 1, p. 1–26. DOI: 10.1002/dac.4141

[23] *BigQuery Public Datasets.* [Online] Cited 2023-12-06. Available at: https://cloud.google.com/bigquery/public-data

About the Authors...

Yi XIONG is a master's student at the Southwest Jiaotong University, with a primary research focus on information security, privacy security, and related areas.

Ming Xing LUO obtained his Ph.D. in Cryptography from the School of Computer Science at Beijing University of Posts and Telecommunications in 2011, and joined the School of Information Science and Technology at Southwest Jiaotong University in the same year. His research mainly focuses on information security, artificial intelligence security, quantum information processing, quantum artificial intelligence, and related areas.