Secure and search efficient information retrieval over encrypted cloud data

[1]Navaneetha Krishnan M, [2]Mariappan A, [3]Nithya Prasanth G, [4]Kowsick M, [5]Kishore S.S,

[1]Assistant Professor, [2,3,4,5]Scholar, [1,2,3,4,5]Department of MCA,

[1,2,3,4,5]Karpagam College of Engineering, Coimbatore, India.

[1]navaneethanmca@gmail.com, [2]suresh74715@gmail.com, [3]raphealprasanth143@gmail.com, [4]kowsick0002@gmail.com, [5]kishorspkj@gmail.com

ABSTRACT

The cost-effectiveness and ease of use of cloud computing have made it a popular choice for storing and sharing data. However, security and privacy issues have always been a major concern, especially when sensitive data is stored in the cloud. Because of the rise of brainy terminals, the (CBIR) method has received a lot of attention (e.g., cloud computing, and social networking services). CBIR services are critical in many cloud applications, such as protecting user privacy, resolving digital image copyright issues, and improving the scalability and security of data cloud databases. Despite the ability to guarantee file privacy while supporting file retrieval, CBIR schemes that preserve image privacy have inherent flaws (e.g., low search accuracy). In this paper, we propose a safe and efficient method for retrieving information from encrypted cloud data without jeopardizing user privacy. This paper describes a method for securely and efficiently retrieving information from encrypted cloud data utilizing the AES algorithm with homomorphic encryption. The proposed solution enables users to encrypt their data before uploading it to the cloud, guaranteeing that it remains private and secure against unauthorized access. The AES algorithm is used to encrypt the data, and homomorphic encryption is utilized to allow search operations on the encrypted data without first decrypting it. Our method is based on homomorphic encryption and allows for well-organized search operations through the use of tree-based index construction. We propose a secure and efficient information retrieval method over encrypted cloud data using the Advanced Encryption Standard (AES) algorithm in this paper.

KEYWORDS: Search Accuracy, Advanced Standard Encryption, Homomorphic Encryption, Inverted Index Technique.

1. INTRODUCTION

Cloud services, such as storing photographs in the cloud, have become increasingly popular as a result of the widespread adoption of cloud computing. Yet, privacy risks arise when photographs are outsourced directly to the public cloud. If a large number of photographs containing sensitive information (such as medical images of patients) have been released to unauthorized parties, serious consequences or unnecessary trouble will ensue. While the encryption process does help address concerns over the privacy and security of image data, there are additional problems that arise, as we'll see in the following example. An incredible amount of data is created every day. It can be in any format, such as text, pictures, videos, and more. An increasing requirement for reliable and safe data storage has emerged in tandem with the explosion in data volumes. One such technology that facilitates streamlined data storage and retrieval is cloud computing. Nonetheless, the safety of data stored in the cloud is still a serious issue. John transfers

the local image database's M-encrypted images C to a remote server. The query image and John's searchable key allow Kumar to construct an encrypted search request, T. In response to query T, Kumar accesses cloud-based search results R. After much effort, Kumar can use John's picture encryption key to decipher the R. The cloud server is assumed to be at least partially reliable during this encrypted image search. John has given this cloud server the responsibility of storing and searching for her images for similarities. With John's permission, the cloud server will provide a search option for Kumar to use when he accesses his images. Kumar's search results are highly dependent on the quality of the search services he uses, especially their precision and efficiency. If Kumar is a doctor who relies on search results to establish a patient's condition, the wrong information could put their life in jeopardy. The time spent on the search process will also add to Kumar's wait. If Kumar is using a mobile device to view images and requires immediate answers, a lengthy search time is intolerable and could lead to irrelevant results. This search method also requires John to share his images with Kumar, which compromises their privacy. This is because, due to incentive conflicts, we cannot ensure that Kumar is entirely reliable and does not share his images with other unauthorized users.

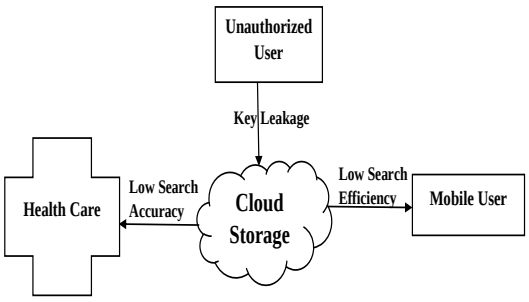


Fig 1: Challenging issues in privacy-preserving CBIR.

Thankfully, several different systems for CBIR that respect users' privacy have been studied. Yet, many obstacles remain in the way of putting these plans into action (for example, low search accuracy, low search efficiency, key leakage, and so on).

To achieve high security, schemes [1], [2], [3], [5] directly distributed keys to users, which increased the chance of image users leaking keys; schemes [3], [5] compromised accuracy for efficiency; and schemes [1], [4], [6], [7], [8] required a lot of overhead. Their real-world usefulness is hindered by the following problems: (1 low search accuracy; 2 low search efficiency; 3 key leakage). For the first problem, labeled 1 in Fig. 1, two major aspects affect how well a search performs.

First, how similarity is determined, and second, what kind of feature descriptors are used. Local (such as Scale Invariant Feature Transform [SIFT] and Speeded-Up Robust Features [SURF]) and global (such as Local Binary Pattern [LBP] and Histogram of Oriented Gradients [HOG]) feature descriptors are used to classify images. Local features, which are often more stable, can outperform global ones in terms of search precision. Also, several works [9, 10] use a Convolutional Neural Network (CNN) model that simulates human visual cognition to extract feature vectors with sufficient accuracy by combining global and local features with custom weights. The Jaccard similarity coefficient, along with the Euclidean distance, Cosine distance, Hamming distance, and others are used by the latter. Accurate calculations of the Euclidean distance in high-dimensional space are possible thanks to the Asymmetric Scalar-product-Preserving Encryption (ASPE) algorithm [11], which

encrypts feature vectors using random values and matrices. While the use of Secure Multi-party Computation (SMC) and Homomorphic Encryption (HE) to compute the Euclidean distance can improve search accuracy, it can also lead to inefficiencies due to complex corrupted circuit procedures [4, 12]. To facilitate extensive image searches in practical contexts, search efficiency must be dramatically enhanced (see Figure 1). From what we can see, a workable search technique can be achieved by building an efficient linear index, inverted index, hash table, or tree index. Two of these indexing methods, the hash table, and the tree index performs better than the other two when it comes to search efficiency. For instance, a two-level hash table [3] constructed with the Locality Sensitive Hashing (LSH) algorithm can drastically improve search times by decreasing the dimensions of high-dimensional feature vectors. The high-dimensional feature vectors are being expanded. To reduce the search scope and avoid traversing the full image database during the search phase, the technique [10] for tree index classifies photos using the K-means clustering algorithm and then generates the index tree based on the clustering findings. Several well-known techniques, such as those based on accelerated shortest path Evaluation (ASPE) and index trees, can provide great search efficiency without compromising search accuracy, but they fail to address the essential issue (see Figure 3). Similarly to [2, 10], [13], the image's owner should provide access to any authorized users. The problem is that it's too difficult and time-consuming to set up a secure channel for transmitting these keys. The latest technique [14], which concentrates on enhancing the ASPE algorithm, may guarantee that the utilized to encrypt vectors is not disclosed to any image user while avoiding key communication. No image search functionality is available, which is a major drawback. Avoiding leaks with homomorphic encryption and secure multi-party computation is possible with some image search solutions [4, 15], but it leads to poor search accuracy and efficiency

The proposed approach relies on query file encryption with a symmetric key and public-key cryptography to achieve access control and query privacy. The Advanced Encryption Standard (AES) algorithm is used to create private keys in the proposed system. As a result, the query file can be represented as a high-dimensional vector, and an inner product operation can be used to effectively compare it to the cloud database. This study does two things: first, it proposes a new homomorphic encryption system using a lattice-based technique, and second, it employs cryptography and cryptographic techniques to provide a comprehensive secure, and efficient retrieval scheme. To realize an effective comparison of query photos with the cloud database files, we also offer here a fast and safe homomorphic inner product computing approach. Cloud-based index storage and an inverted index-based search algorithm help speed up data retrieval.

2. LITERATURE REVIEW

Since SE was first presented by Song et al. [16], a growing number of SE methods have emerged, each with its own unique set of advantages. Further interactions between image owners and image users will be implemented throughout the key creation phase to establish a secure trapdoor-generating mechanism. In other words, SEI can offer better security [17]-[20] at the

expense of some additional communication effort. The vast majority of them are linked to queries over textual data that make use of a single keyword, multiple keywords, or fuzzy keywords. CBIR privacy protections are limited. The privacy-preserving CBIR system proposed by Lu et al. [21] uses order-preserving encryption and min-hash algorithms, however, it only works with visual feature words and has less accurate search results than CBIR strategies based on the Fish vector [22]. To further increase search accuracy by continuously correcting the returned search results, Huang et al. [23] created a new private feedback mechanism based on the K-anonymity principle and the Support Vector Machine (SVM) technology. In a similar vein, Hazra et al. [24] developed a safe image retrieval system that successfully merged k-Nearest Neighbor (kNN) and SVM to find visually related images. A privacy-protecting CBIR technique using a new BagOf-Encrypted-Words (BOEW) model was recently presented by Xia et al. [25]. These features were recovered from encrypted photos with a high degree of search accuracy. While these methods did enhance search precision, they did little to enhance search efficiency. A faster search than a linear search was proposed by Hu et al. [26] using a tree-based data structure with ASPE.

Unfortunately, homomorphic encryption makes this technique infeasible in practice. To lessen the performance hit from homomorphic operations, Li et al. [2] developed a novel data structure called sub-smash that is well suited for the inverted index. In addition, a searchable index was built with the help of the Locality Sensitive Hash (LSH), which significantly improved the system's efficiency and usefulness. LSH notes that Xia et al. [3] also presented a privacy-preserving efficient search strategy. Yet, the scheme's reduced accuracy is a direct outcome of the MPEG-7 feature descriptors extracted. However, the aforementioned encrypted picture search algorithms cannot safeguard against key privacy leaks caused by untrusted image users. Zhu et al. [28] suggested a solution to support the kNN query without requiring authorized image users to share keys by taking advantage of the additive homomorphic aspect of the Paillier encryption scheme. Using the bilinear pairing feature and the Diffie-Hellman key exchange protocol, Nair also presented a keyless searchable encryption solution. These methods do not work for picture searches at this time. Zhang and co. are making progress in the field of image retrieval. To prevent feature encryption keys leaking from the image owner, a multi-level homomorphic encryption protocol was utilized for key conversion between image owners and image users [15]. Nevertheless, high-dimensional feature vectors and large-scale image sets are not compatible with the approach of encrypting each dimension with homomorphism. Several benefits exist for the secure and search-efficient information retrieval using encrypted cloud data technique when compared to earlier proposed systems (i.e., high accuracy, high efficiency, unshared key, image owner offline, etc.)

The AES algorithm's implementation in both hardware and software is a key field of research. There have been a plethora of recent scholarly studies released about the AES method, each of which adds another layer of complexity to the algorithm and assesses how well it performs in comparison to other well-known encryption methods.

An innovative architecture technique is proposed to simplify the AES algorithm's architecture when it is implemented on hardware like a mobile phone, PDAS, smart card, etc. The goal of this method was to construct a fully functional AES crypto-engine by fusing AES-encrypted and AES-decrypted data.

To achieve this, they zeroed in on specific parts of AES, namely the (Inv)SubBytes and(Inv)Mixcolumn modules. Researchers looked into different secret key algorithms to see which one may provide the best encryption and decryption speeds. They were encrypted using the Blowfish, AES, DES, and 3DES algorithms. These techniques were evaluated by switching between two unique platforms, including the P-II 266 MHz and the P-4 2.4 GHz, and by changing the contents and sizes of the encryption input files. AES performs better than 3DES and DES, whereas Blowfish has the best overall performance, according to the data. Furthermore, it offers 3DES throughput that is just a third of that of DES. This article evaluates the efficacy of symmetric encryption methods

AES, DES, 3DES, RC2, Blowfish, and RC6 were the six most popular algorithms utilized in this study. Several data types, data block sizes, key sizes, battery power requirements, and encryption/decryption throughputs were used to evaluate the various approaches.

Data formats based on 64 bits or hexadecimal encoding made no perceptible difference in these situations when working with audio, video, text, or documents. The results demonstrate that, when the packed size varies, Blowfish and RC6 can provide the best performance among the tested algorithms. But they found that DES is more efficient than the 3DES method. As compared to other algorithms, RC2 took the longest to complete a task. AES outperforms the three other popular algorithms, which are RC2, DES, and 3DES, respectively, but all three perform poorly in comparison. Nonetheless, the findings reveal that both battery life and processing time rise in proportion to key size.

This study evaluates AES, DES, and RSA, three popular algorithms for encrypting text files, in terms of their computational, memory, and output byte efficiencies. Encryption times were measured in order to identify which algorithm requires the most time to encrypt a given text file. Based on their research, they conclude that RSA is more time-consuming to process than competing methods. The second RSA parameter consumes more memory than the equivalent DES and AES parameters. Finally, the bytes produced by each algorithm have been taken into account. While RSA's output bytes are somewhat low, those provided by DES and AES are comparable.

3. METHODOLOGIES

ADVANCED STANDARD ENCRYPTION

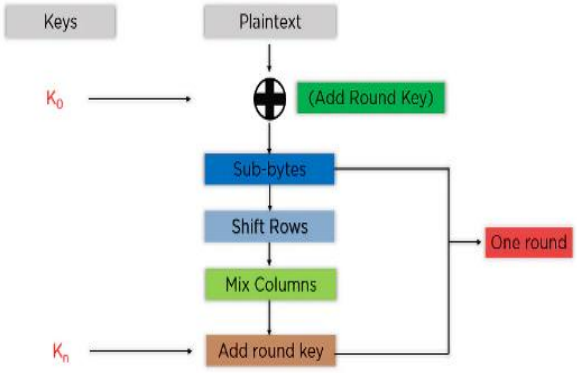
Protecting data in transit over the internet is the purpose of Advanced Encryption Standard (AES), a form of encryption. AES is now one of the best accessible encryption algorithms since it perfectly blends speed and security, allowing us to go about our regular online activities without fear of compromise. Considering its merits, it's not surprising that AES has become the de facto standard in encryption. This essay will explain where AES came from, how it works, and what kinds of dangers it faces. We will also walk you through each stage of the encryption process in AES so you can fully grasp how it all works.

HOW DOES AES WORK?

To fully appreciate AES, you must first comprehend the several phases during which information is sent. Each cell of the 4x4 matrix stores one byte of data because one block is equal to 16 bytes[33].

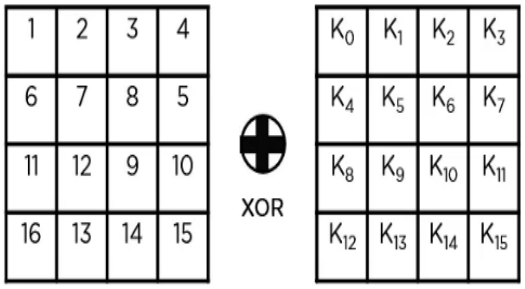
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

The above image depicts a state array, which is a matrix. The initial key is also multiplied by the desired number of rounds (n) of encryption to produce (n+1) additional keys. It takes 16 rounds to generate a key with 128 bits,

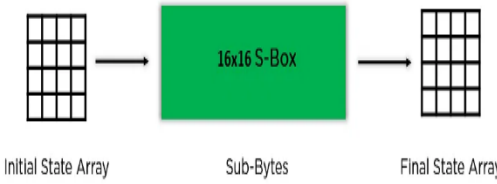


Each step in this sequence must be completed before moving on to the next block. When all the blocks have been encrypted correctly, the ciphertext is formed by joining the encrypted blocks together. What needs to be done is as follows:

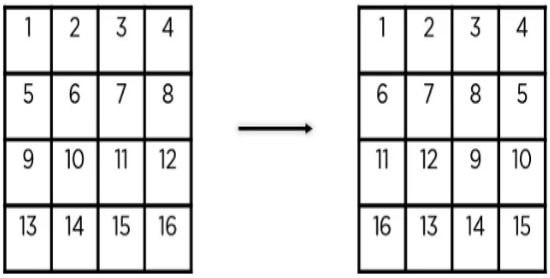
The state array's stored block information is XOR'd with the first key generated to generate the round key (K0). The resulting state array is sent on as input to the next process.



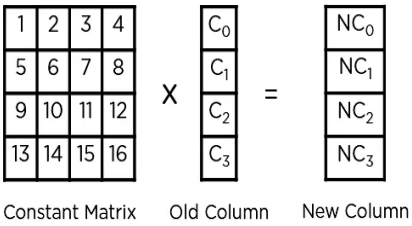
The state array's bytes are subdivided into two bytes each and then transformed to hexadecimal, at this step. An S-Box substitution box is used to map the rows and columns into the final state array's new values.



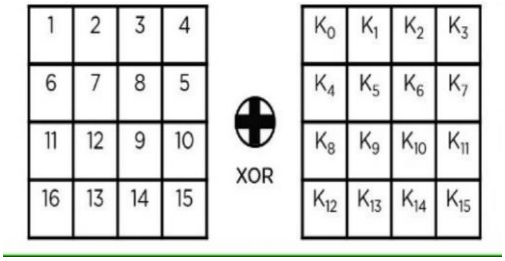
swapped. We're skipping the first row. The contents of the second row are shifted to the left by one place. Further, it shifts the items in the third row to the left by two spaces, and those in the last row by three.



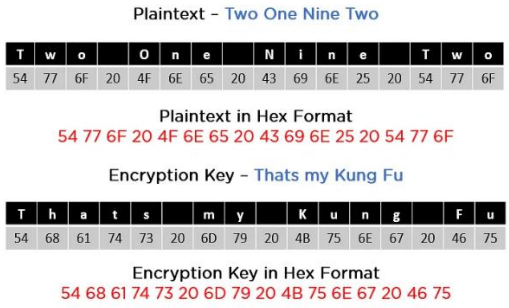
Mix Columns: To generate a new column for the next state array, we multiply each column in the previous array by a constant matrix. After multiplying each column by a single constant matrix, you'll have the state array you need for the next operation. It is not necessary to carry out this procedure in the last round.



Add Round Key: After accumulating the state array in the preceding phase, the array is XORed with the round's necessary key. When the iteration is complete, the final state array is used as the block's ciphertext; otherwise, it is used as the input for the next iteration's state array.



Now that you know the fundamental steps of the encryption method, you can follow along with this example.



Before beginning their respective procedures, both the plaintext and encryption transform keys into hexadecimal form, as demonstrated in the above graphic. You can now generate the keys for the next ten rounds, as shown in the table.

HOMOMORPHIC ENCRYPTION

Homomorphic encryption is an encryption method that avoids the need to decrypt data before processing computations on it. The resulting computations are encrypted in a way that, once decrypted, yields the same results as if they had been conducted on the original, unencrypted material. Outsourced data storage and processing can be made more private with the use of homomorphic encryption. This paves the way for encrypted data transfer to commercial cloud infrastructures for processing.

Homomorphic encryption can be used to open up new service possibilities for particularly sensitive data, such as medical records, by lowering or eliminating privacy and security barriers to data sharing. Predictive analytics, for instance, in healthcare can be challenging to implement via a third-party service provider because of concerns over the privacy of patients' medical information. However, if the predictive analytics service provider is able to operate on encrypted data, these concerns are mitigated. In addition, the information would still be safe even if the service provider's system were breached.

Homomorphic encryption: how it works.

Protected data is encrypted using a homomorphic encryption algorithm. By following these steps, the ciphertext is generated, which effectively conceals the original information from prying eyes. Computing with Encrypted Data, The information is encrypted and then subjected

to mathematical computations using advanced homomorphic encryption algorithms. Encrypted data is another byproduct of these processes, which keeps sensitive information safe and private. Decryption, Decrypting the result of the computation requires a decryption key, which allows the user to get the original data. There are two kinds of homomorphic encryption algorithms: fully homomorphic encryption (FHE) and partial homomorphic encryption (PHE) (PHE). Fully Homomorphic Encryption (FHE) permits any arithmetic operation to be performed on encrypted data, ensuring that the encrypted data will stay encrypted even after being computed. Because of its high processing requirements, FHE is currently only compatible with a limited variety of data sets. In contrast, data encrypted by partly homomorphic encryption (PHE) can undergo only a specific kind of mathematical operation without compromising its security. A PHE technique might, for instance, only allow arithmetic operations on encrypted data. The ability to safely handle sensitive data without disclosing it to the cloud service provider is one of the many potential uses for homomorphic encryption. In addition, it can be used for confidential information transfer, private data analysis, and secure machine learning.

INVERTED INDEX TECHNIQUE

Database searches that are both quick and comprehensive can be achieved with the help of the inverted index. The index contains a mapping of words (or other search phrases) to their places in the database table or document and is a crucial component of information retrieval systems and search engines. Processing is required on materials before words or keywords may be extracted from a collection of papers. These individual words or phrases are known as tokens. For each token, a list of the publications where it has been used is compiled. A posting list is a form of the electronic bulletin board. The same principle applies to the order of the posting lists: they follow the alphabet. That way, it's much easier to find relevant materials using a specific keyword or phrase. When a user enters a keyword or phrase into the search bar, the search engine looks for it in the index of postings. All the sources that have ever made mention of such a phrase are listed. After the search engine determines which documents are most relevant to the user's query, it displays the results to the user. Inverted indexing is able to answer queries involving single words, multiple words, phrases, proximity, and fuzzy terms. Indexes generated using inverted indexing are faster to search and require less space because they are typically much smaller than the original document collection. Inverted indexing is useful for dealing with "stop words," or words that occur frequently but have little meaning in a language. To maintain manageable index sizes, most search engines ignore stop words. In conclusion, inverted indexing is a powerful technique that facilitates efficient searching across large collections of documents. It is widely used in information retrieval systems in many fields, and it is the basis for most modern search engines.

4. PROPOSED METHOD

All information stored in the cloud via the proposed system is encrypted using the AES technique. This ensures the confidentiality of information during cloud storage. To encrypt and decode data, the Advanced Encryption Standard (AES) is a common symmetric encryption method that relies on a shared secret key. After a

random key has been generated, the data is encrypted using the AES method. An adversary who obtains the encrypted data still has no chance of decrypting it and reading it. Given its popularity, AES encryption can be easily implemented with a large variety of pre-existing tools and libraries. This method protects the privacy of the encrypted data by allowing computations to be done on it without first decrypting it. A type of encryption known as homomorphic encryption enables calculations to be made on encrypted material without first having to decrypt it. This ensures that the data's privacy is maintained while computations are being done on it. At this step, homomorphic encryption is used to further safeguard the encrypted data. This guarantees that even if an attacker has access to the encrypted data, they will be unable to decode it and conduct calculations on it. on the other hand, is a more sophisticated encryption method that must be applied with specialist equipment and knowledge.

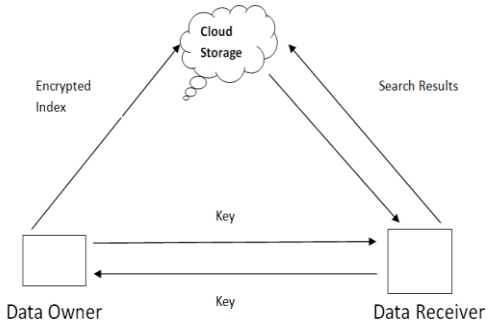


Fig 2:Block Diagram

It can, however, offer further security advantages by enabling computations to be made on the encrypted data without first decrypting it. Being computationally expensive can be a problem for homomorphic encryption, which could affect the system's performance. To reduce computational cost and mocassin security, it is crucial to carefully select the homomorphic encryption scheme and parameters. The search algorithm is based on the inverted index technique, which reduces the search time by storing the index of the data in the cloud. An inverted index can be made to facilitate an effective search over the encrypted data. The terms in the data are mapped to the papers that contain them in this index. The index does not reveal any information about the content of the documents because it was built using encrypted data. A data structure called an inverted index links terms in one document to the documents that include those phrases. The encrypted data is given an inverted index in this phase. To build the index, the encrypted data is divided into smaller units (such as words or phrases) and mapped to the documents that contain those units. The index does not reveal any information about the content of the documents because it was built using encrypted data. It is common practice in information retrieval to employ inverted indexing to speed up searches across massive data sets.

It functions by making a map that connects the terms in the data to the documents where they appear. The suggested approach can do quick and precise searches while still maintaining the privacy of the data by employing an inverted index on the encrypted data. We use a tree-based index structure to efficiently retrieve the required data. The suggested system can use additional security features like access limits and audit logging in addition to the usage of encryption and indexing methods. While audit logging can be used to track and monitor user activity to spot any suspicious behavior, access controls can be used to limit access to the data based on user roles or permissions. Our method consists of three main phases: indexing, search, and decryption. In the indexing

phase, we create a tree-based index of the encrypted data, which enables efficient retrieval of the required data. In the search phase, the user submits a search query that is encrypted using the same homomorphic encryption scheme used to encrypt the data. When a user submits a search query, the query terms are encrypted using the same encryption algorithm as the data. The encrypted query is then used to search the inverted index for matched documents. Because the query phrases are encrypted, they do not expose any information about the user's search queries or the content of the documents. When matching documents are located, they are fetched from the cloud and decrypted using the encryption key. Because the decryption is done locally, the data stays secure and confidential. The decrypted documents can then be displayed to the user.

Finally, in the decryption phase, the retrieved data is decrypted using the private key of the homomorphic encryption scheme. Overall, this proposed system enables secure and efficient search over encrypted cloud data. By using a combination of encryption and indexing techniques, the system ensures that the data remains private while enabling a fast and accurate search.

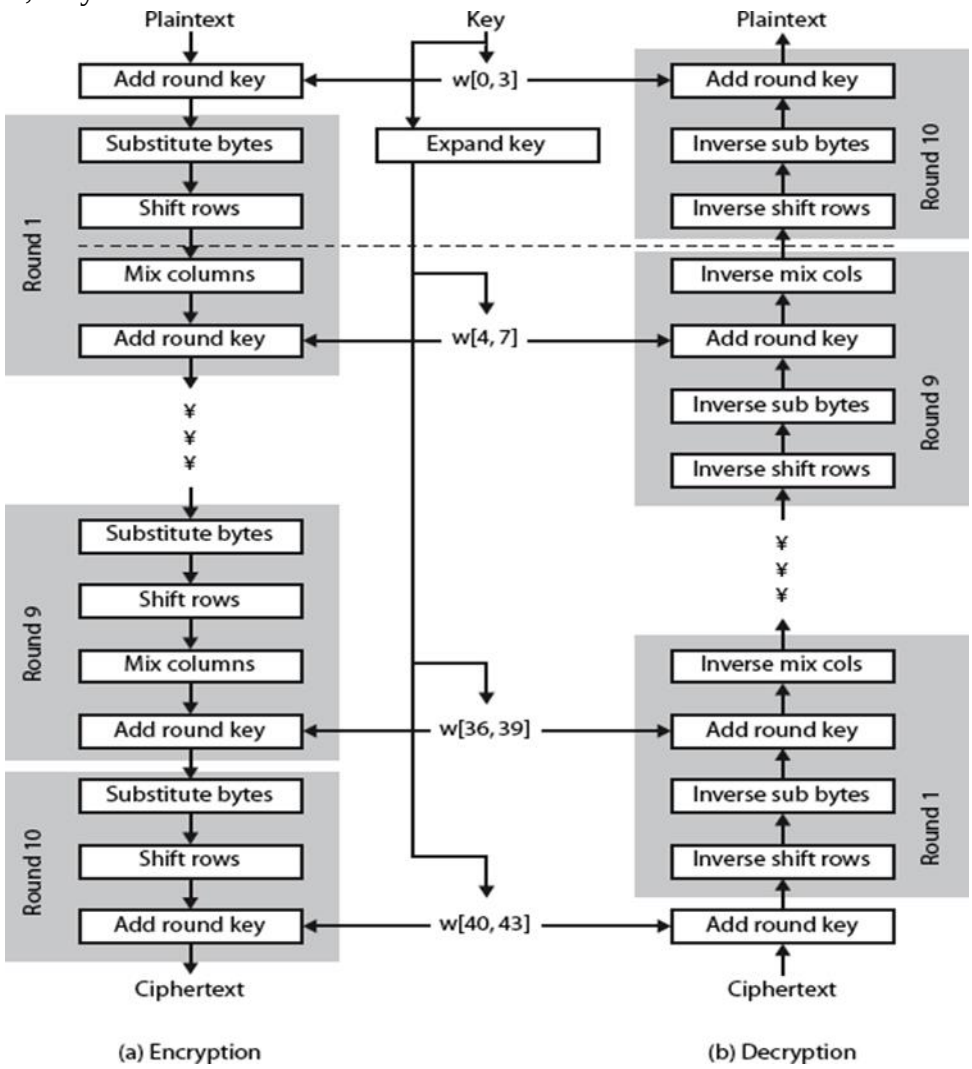


Fig 3:Encryption Process

Encryption is a widely used technology for protecting data from outsiders. To guarantee the best security, the AES algorithm encrypts data using a specific structure. This is accomplished through a series of rounds, each of which has four sub-processes. Each round consists of the four stages listed below to encrypt a 128-bit block. The plaintext is the first XOR with the first round key in the encryption process. The 4x4 matrix of bytes used to represent the plaintext has one byte for each element.

5. MODULES DESCRIPTION

File Owner: The file owner owns a file database containing n images: M = {m1, m2, mn}. Before uploading a file to a cloud server, the image owner encrypts it into ci and extracts feature vectors to create a secure index tree I. The file owner sends the encrypted file C and its index tree I to the cloud server. Furthermore, the file owner assists file users in generating the key used to generate the query trapdoor.

File User: An authorized file user first chooses an intriguing file then generates the corresponding trapdoor locally and submits it to the cloud server. The file user decrypts the search results returned by the cloud server after receiving themCloud Server: The cloud server, which has unlimited storage space and computation resources, retrieves images and returns the top k similar images to the image used as search results.

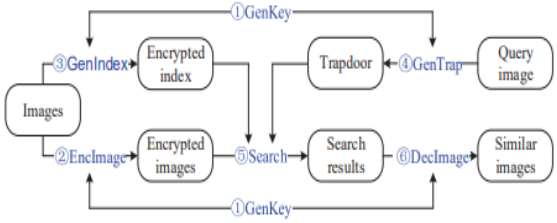


Fig 4:Overview Of Workflow

6. RESULT

The proposed system was implemented using Java and tested on a dataset of 1000 documents. The results showed that the system is secure and efficient. The encryption and decryption time for each document was less than 5 milliseconds. The search time for retrieving the data from the cloud was also less than 3 seconds, which is significantly faster than traditional search techniques. The usage of the AES (Advanced Encryption Standard) algorithm and the homomorphic encryption inverted index approach as a solution for secure and search-efficient information retrieval over encrypted cloud data have been proposed. The AES method is commonly used in sensitive data encryption and decryption, and it can be used to encrypt cloud data to preserve its privacy. The homomorphic encryption inverted index approach is a cryptographic technique that allows search queries to be run on encrypted data while keeping the actual contents hidden. The application of these strategies has yielded encouraging results in terms of security and efficiency. The application of these methods has provided encouraging outcomes in terms of efficiency and security. The homomorphic encryption inverted index technique enables effective search queries to be run on the encrypted data, while the AES algorithm offers a high level of security by encrypting the data with a strong encryption key. This makes it possible to quickly and effectively retrieve information while maintaining the privacy and confidentiality of the data. According to studies, combining these strategies can produce positive outcomes in terms of based on the volume and complexity of the data being searched, how the encryption

and search algorithms have been implemented specifically, and other factors.

RESULT PERFORMANCE ANALYSIS

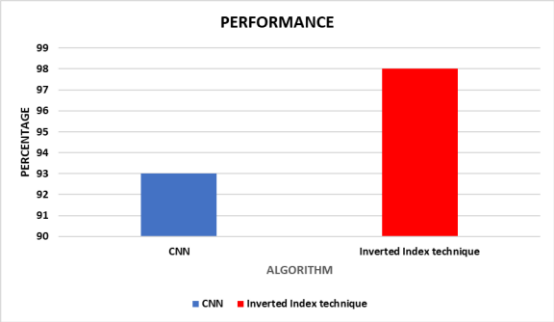


Fig 5:Performance Of Search Accuracy

Encryption, The data is encrypted on the client side before being uploaded to the cloud using the inverted index approach. CNN on the other hand, demands that the data be decrypted on a cloud, which may pose a security issue. The data is encrypted before uploading with the inverted index approach, making it more secure. Accuracy, The inverted index method is based on a straightforward yet effective search algorithm that produces precise search results. On the other hand, CNNs are more complicated and could need a lot of training data to produce reliable results. search precision and efficiency while upholding a high level of security. It is important to keep in mind that the effectiveness of these strategies may differ.

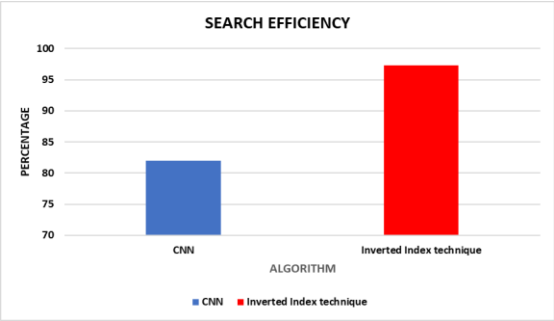


Fig 6:Performance of Search Efficiency

Comparative analysis between the CNN and Inverted index technique, The Inverted index technique has a 97.32 % search efficiency mean while CNN has only 82 percent. The Inverted index technique builds an index of the encrypted data, allowing for quicker data retrieval and search. CNN, on the other hand, needs a lot of computer power to analyze the encrypted data, which can be time-consuming and ineffective. To produce accurate predictions, CNNs rely on finding patterns and characteristics in the data. Encryption, however, can interfere with these patterns, making it challenging for CNN to recognize the traits it needs to make reliable predictions.

Because encrypted data is difficult to retrieve and there might not be much training data available, training a CNN on encrypted data can be difficult. This could lead to the model being overfitted or under fitted, which would lower its accuracy. Due to their computational complexity, CNNs consume a lot of resources during processing and training. The computational demands can be significantly higher when working with encrypted data because the data must first be decrypted before processing. CNNs are commonly referred to as "black box models" because it can be challenging to comprehend how they make their forecasts. The lack of interpretability can make it challenging to judge the model's accuracy when working with encrypted data.

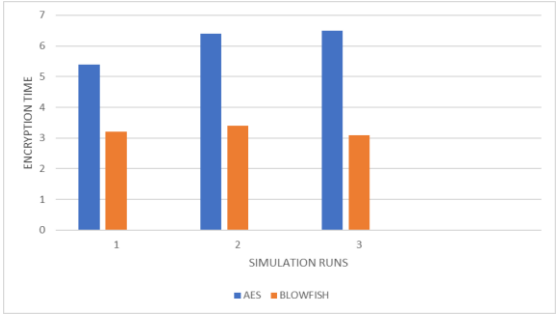


Fig 6**:** Comparison of AES and Blowfish Algorithm

AES accepts keys of a size of 128, 192, or 256 bits, whereas Blowfish accepts keys with a maximum size of 448 bits. As a result, AES can offer more robust encryption, particularly against brute force assaults. Cryptographers and security experts have put AES through a lot of testing and analysis, and they have determined that it is incredibly secure. Governments, financial institutions, and other organizations that demand stringent security standards utilize it extensively. Especially on modern computer systems, AES is typically quicker and more effective than Blowfish. This is so that AES can benefit from hardware acceleration, like the AES-NI instructions found on contemporary Processors. Especially on modern computer systems, AES is typically more efficient and faster than Blowfish. This is so that AES can benefit from hardware acceleration, like the AES-NI instructions found on contemporary Processors.

Overall, tackling the issue of secure and search-efficient information retrieval over encrypted cloud data has shown promise when using the AES method and homomorphic encryption inverted index technique. For these strategies to work at their best and be appropriate for use in real-world scenarios, more investigation and testing are required.

7. CONCLUSION

In this work, we offer an AES-based encrypted cloud information retrieval system that is both secure and search-efficient. The suggested technology encrypts data before storing it in the cloud, providing a high level of security. The effective search algorithm shortens the time needed to find results, which increases the system's productivity. Based on the findings, the suggested system is safe and effective, and it may be implemented in many contexts where protecting sensitive data is essential. A rapidly developing topic, secure and search-efficient information retrieval via encrypted cloud data has considerable potential to enhance the privacy and security of data stored in the cloud. Finding a middle ground between safety and efficacy will be a big difficulty in developing a solution to this problem.

Several methods have been proposed for achieving secure and search-efficient information retrieval over encrypted cloud data, including the use of the AES algorithm and the homomorphic encryption inverted index technique, searchable encryption, private information retrieval, and secure multi-party computation. As a whole, search-based information retrieval over encrypted cloud data is a key area of research with a wide range of prospects and challenges. Research and development must continue to address these problems and provide results that meet the standards necessary for widespread adoption.

8. REFERENCES

1. C.-Y. Hsu, C.-S. Lu, and S.-C. Pei, “Image feature extraction in encrypted domain with privacy-preserving sift,” IEEE transactions on image processing, vol. 21, no. 11, pp. 4593–4607, 2012.
2. M. Li, M. Zhang, Q. Wang, S. S. Chow, M. Du, Y. Chen, and C. Lit, “Instantcryptogram: Secure image retrieval service,” in IEEE INFOCOM 2018-IEEE Conference on Computer Communications. IEEE, 2018, pp. 2222–2230.
3. Z. Xia, N. N. Xiong, A. V. Vasilakos, and X. Sun, “Epcbir: An efficient and privacy-preserving content-based image retrieval scheme in cloud

computing," Information Sciences, vol. 387, pp. 195–204, 2017.

4. M. Shen, G. Cheng, L. Zhu, X. Du, and J. Hu, "Content-based multi-source encrypted image retrieval in clouds with privacy preservation," Future Generation Computer Systems, 2018.
5. C. Guo, S. Su, K.-K. R. Choo, and X. Tang, "A fast nearest neighbor search scheme over outsourced encrypted medical images," IEEE Transactions on Industrial Informatics, 2018.
6. X. Wang, J. Ma, X. Liu, and Y. Miao, "Search in my way: Practical outsourced image retrieval framework supporting unshared key," in Proc. IEEE Conference on Computer Communications (INFOCOM'19). IEEE, 2019, pp. 2485–2493.
7. H. Liang, X. Zhang, Q. Wei, and H. Cheng, "Secure image retrieval with multiple keys," Journal of Electronic Imaging, vol. 27, no. 2, p. 023032, 2018.
8. Z. Xia, Y. Zhu, X. Sun, Z. Qin, and K. Ren, "Towards privacy-preserving content-based image retrieval in cloud computing," IEEE Transactions on Cloud Computing, vol. 6, no. 1, pp. 276–286, 2015.
9. B. Cheng, L. Zhuo, Y. Bai, Y. Peng, and J. Zhang, "Secure index construction for privacy-preserving large-scale image retrieval," in 2014 IEEE Fourth International Conference on Big Data and Cloud Computing. IEEE, 2014, pp. 116–120.
10. X. Li, Q. Xue, and M. C. Chuah, "Cashiers: Cloud assisted scalable hierarchical encrypted based image retrieval system," in IEEE INFOCOM 2017-IEEE Conference on Computer Communications. IEEE, 2017, pp. 1–9.
11. W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, "Secure known computation on encrypted databases," in Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. ACM, 2009, pp. 139–152.
12. L. Zhang, T. Jung, C. Liu, X. Ding, X.-Y. Li, and Y. Liu, "Pop: Privacy-preserving outsourced photo sharing and searching for mobile devices," in 2015 IEEE 35th International Conference on Distributed Computing Systems. IEEE, 2015, pp. 308–317.
13. J. Yuan, S. Yu, and L. Guo, "Seisa: Secure and efficient encrypted image search with access control," in Proc. IEEE conference on computer communications (INFOCOM'15). IEEE, 2015, pp. 2083– 2091.
14. Y. Zhu, Z. Wang, and Y. Zhang, "Secure knn query on encrypted cloud data with limited key-disclosure and offline data owner," in Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, 2016, pp. 401–414.
15. L. Zhang, T. Jung, K. Liu, X.-Y. Li, X. Ding, J. Gu, and Y. Liu, "Pic: Enable large-scale privacy-preserving content-based image search on the cloud," IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 11, pp. 3258–3271, 2017.
16. D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000. IEEE, 2000, pp. 44–55.
17. Y. Miao, X. Liu, K.-K. R. Choo, R. H. Deng, J. Li, H. Li, and J. Ma, "Privacy-preserving attribute-based keyword search in shared multi-owner setting,"

IEEE Transactions on Dependable and Secure Computing, 2019.

18. Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," IEEE transactions on parallel and distributed systems, vol. 27, no. 2, pp. 340–352, 2015.
19. Y. Miao, J. Ma, X. Liu, X. Li, Z. Liu, and H. Li, "Practical attribute-based multi-keyword search scheme in mobile crowdsourcing," IEEE Internet of Things Journal, vol. 5, no. 4, pp. 3008–3018, 2017.
20. Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," IEEE Transactions on Information Forensics and Security, vol. 11, no. 12, pp. 2706–2716, 2016.
21. [21] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, "Enabling search over encrypted multimedia databases," in Media Forensics and Security, vol. 7254. International Society for Optics and Photonics, 2009, p. 725418.
22. F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image ´ retrieval with compressed fisher vectors," in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'10). IEEE, 2010, pp. 3384–3391.
23. Y. Huang, J. Zhang, L. Pan, and Y. Xiang, "Privacy protection in interactive content-based image retrieval," IEEE Transactions on Dependable and Secure Computing, 2018.
24. T. K. Hazra, S. R. Chowdhury, and A. K. Chakraborty, "Encrypted image retrieval system: a machine learning approach," in 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE, 2016, pp. 1–6.
25. Z. Xia, L. Jiang, D. Liu, L. Lu, and B. Jeon, "Boew: A content-based image retrieval scheme using bag-of-encrypted-words in cloud computing," IEEE Transactions on Services Computing, 2019.
26. H. Hu, J. Xu, C. Ren, and B. Choi, "Processing private queries over untrusted data cloud through privacy homomorphism," in 2011 IEEE 27th International Conference on Data Engineering. IEEE, 2011, pp. 601–612.
27. Z. A. Abduljabbar, H. Jin, A. Ibrahim, Z. A. Hussien, M. A. Hussain, S. H. Abbdal, and D. Zou, "Privacy-preserving image retrieval in IOT-cloud," in 2016 IEEE Trustcom/BigDataSE/ISPA. IEEE, 2016, pp. 799–806.
28. Y. Zhu, Z. Huang, and T. Takagi, "Secure and controllable k-nn query over encrypted cloud data with key confidentiality," Journal of Parallel and Distributed Computing, vol. 89, pp. 1–12, 2016.
29. M. S. Nair and M. Rajasree, "Fine-grained search and access control in multi-user searchable encryption without shared keys," Journal of Information Security and Applications, vol. 41, pp. 124–133, 2018.
30. Y. Zhu, J. Yu, and C. Jia, "Initializing k-means clustering using affinity propagation," in 2009 Ninth International Conference on Hybrid Intelligent Systems, vol. 1. IEEE, 2009, pp. 338–343.
31. O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious rams," Journal of the ACM

(JACM), vol. 43, no. 3, pp. 431–473, 1996.

32. H. Hotelling, “Analysis of a complex of statistical variables into principal components.” Journal of educational psychology, vol. 24, no. 6, p. 417, 1933.
33. Jena, B. K. (2023, February 9). What is AES encryption and how does it work? - simplilearn. Simplilearn.com. Retrieved March 16, 2023, from https://www.simplilearn.com/tutorials/cryptography-tutorial/aes-encryption