ZKG INTERNATIONAL ISSN: 2366-1313

Effective Recovery on Documents Encrypted by Features in Cloud Computing

[1]MUGI RAKESH REDDY, [2] A. CHENNAKESAVA REDDY

[1]PG Scholar, Dept. of MCA, Newton's Institute of Engineering, Guntur, (A.P)

[2]Assistant Professor, Dept. of CSE, Newton's Institute of Engineering, Guntur, (A.P)

Abstract: *One of the most active areas of study in cloud computing is the safe and efficient management of sensitive documents. While there have been numerous proposals for searchable encryption techniques, very few allow for fast recovery of documents that have been encrypted based on properties. In this research, we first develop a hierarchical attribute-based encryption technique for a data set consisting of documents. If many documents have the same access requirements, they may be encrypted as a group. Both the ciphertext storage space and the time costs of encryption/decryption are reduced as compared to the ciphertext-policy attribute-based encryption (CP-ABE) techniques. Next, using the TF-IDF model and the documents' attributes, an index structure called an attribute-based retrieval features (ARF) tree is built for the document collection. The search efficiency for the ARF tree may be enhanced by the use of parallel computing and a depth-first search technique. By making certain adjustments to the ARF tree, our method may be used with datasets other than document collections. The safety and effectiveness of the suggested method are shown by an in-depth study and a battery of trials.*

Keywords: *Cloud computing, document retrieval, file hierarchy, attribute-based encryption.*

I. INTRODUCTION

Cloud-based document management is a potential information technology (IT) for handling the exponentially growing data needs of businesses and individuals alike [1]. With cloud computing, a large number of IT resources may be pooled and reorganised, and it would seem that cloud servers can provide safer, more adaptable, more varied, cheaper, and more individualised services than their on-premises counterparts. While there are many benefits to using cloud services, there is also a significant risk that private information, bank records, or even government papers might be leaked. Cloud data consumers also want versatile and effective access to these data in order to realise their full potential. Therefore, keeping data private while yet making it

ISSN: 2366-1313

searchable is a significant difficulty when moving it to the cloud.

Using encryption to secure files before sending them to the cloud is a natural course of action. Extremely sizable

The National Natural Science Foundation of China (Grant No.11261060) has funded this study.

Both N. Wang and J.W. Zeng may be reached through email at (wangna@stu.xmu.edu.cn, jwzeng@xmu.edu.cn) from the School of Mathematical Science at Xiamen University in Xiamen, China 361005.

Email (12120067@bjtu.edu.cn) may be used to contact J.S. Fu of the School of Electronic and Information Engineering at Beijing Jiao tong University in Beijing, China.

B.K. Bhargava can be reached at (bbshail@purdue.edu), and he works in the Computer Science Department at Purdue University in West Lafayette, Indiana.

Various searchable document encryption schemes have been proposed in the literature, such as single-keyword Boolean search schemes [2]–[6], single-keyword ranked search schemes [7–9], and multi-keyword Boolean search schemes [10–14]. Simple features of these schemes prevent them from supporting effective, flexible, and efficient document search. More effective and realistic are privacy-preserving multi-keyword ranked document search systems [15]-[18]. However, each authorised data user has access to all encrypted documents since these methods employ a coarse-grained access control mechanism. The data owners and consumers will likely not be satisfied in the future if, for instance, the whole IEEE Xplore Digital Library is available to all the authorised organisations (such as the universities).

In this work, we explore an original scenario. It seems sense that a data user who just needs access to a subset of the library's resources (say, the computers and papers relating to data) would want to pay less than a user who needs access to the whole library's resources. Each document in the collection is restricted to a select group of authorised data consumers. When compared to the current way, a more realistic alternative would be to build a fine-grained access control mechanism for the documents.

One potential solution to this problem is to encrypt the documents using attribute-based encryption (ABE) techniques [19], [20] before sending them to the cloud, where they will be accessible to data consumers on demand. Meanwhile, a set of characteristics is ascribed to the authorised data consumers. Only if the data user's characteristics match those of the file may

ISSN: 2366-1313

the file be decrypted. Access control that is fine-grained, one-to-many, and adaptable may be achieved via ciphertext-policy attribute-based encryption (CP-ABE) [21]-[27]. Using hierarchical attribute-based encryption methods [28]-[31] may increase the efficiency of these systems, in which each document is encrypted separately. Unfortunately, we can't just apply these methods to our situation and expect them to work. To begin, the majority of currently available techniques just encrypt a single access tree. The papers in IEEE Xplore Digital Library cannot possibly have a single access tree, and it is a significant issue to figure out how to build a set of optimised access trees for the document collection. Second, as will be discussed in Section IV.B, when the documents are mapped to a set of shared access trees, the data users are required to hold a high number of secret keys. For a really big document collection, this seems to be a significant strain on the data users, and figuring out how to reduce the number of secret keys for the data consumers is still another difficulty. A huge document collection has several challenges, one of which is access control.

In our experience, most current methods do not let attribute-based access control mechanism-organized documents to be retrieved in a timely manner.

We begin by developing an algorithm to construct hierarchical access trees for the document collection, which will be used to provide the service we've been discussing. The suggested technique uses a greedy approach to create the access trees progressively, with each tree expanding by repeatedly dividing its nodes. Then, we create CP-ABHE, a hierarchical document collection encryption method based on attributes in the ciphertext policy. Instead of encrypting each document separately, the suggested system allows them to share a single integrated access tree. This reduces the need for extra space for ciphertext and the time spent encrypting and decrypting. The suggested scheme's security is shown theoretically, and its efficacy is measured by simulation.

A complex index structure is then built for the document collection to provide accurate and efficient document search across the encrypted documents. First, we use the TF-IDF model to convert the documents into document vectors; document properties are also considered. Document vectors are organised in the ARF tree according to their resemblance to one another, which is determined by a properly crafted similarity function.

To be more precise, groups of related vectors (micro clusters) are aggregated into larger groups (macro clusters) until

ISSN: 2366-1313

every vector is part of only one macro cluster. Node ARF vectors are used to characterise the underlying characteristics of the clusters represented by the tree's nodes. Finally, an efficient and accurate depth-first search method for the ARF tree has been developed.

The following is a brief overview of the paper's most important findings:

• A feasible attribute-based hierarchical document collection encryption technique is given, whereby documents are classified and managed in a tree structure. The suggested method may significantly lessen the requirements for data storage and processing power.

We take into account both the keywords and the accompanying properties when mapping the documents to vectors. It is suggested that the ARF tree be used to store the document vectors and facilitate fast document retrieval. There is also the development of a depth-first search algorithm.

• Our scheme's safety, efficiency, and efficacy are shown by extensive simulation. The suggested encryption system, in particular, has excellent efficiency in both time and storage. Furthermore, our technique offers a reliable and fast means of document retrieval.

Here's how the remainder of the paper is laid out: Section II has the corresponding content. In Section III, we define the issue and provide some first solutions. In Section IV, we develop the hierarchical attribute-based document encryption system, and in Section V, we introduce the ARF tree-based, time-efficient document retrieval method. In Section VI, we detail our scheme's security and efficiency analyses, and in Section VII, we conduct experimental evaluations of the suggested method's efficacy. The last section of this work is Section VIII.

II. LITERATURE SURVEY

For online infrastructures, Kan Yang et al. (2014) proposed a PDPS service, which stands for Privacy-preserving Data Publish-Subscribe. The challenge of privacy-preserving data publish/subscribe services arose as a result of first efforts to address the security requirements of data publish/subscribe services in cloud settings. To bridge this gap, we invented Bi-Policy Attribute-Based Encryption (BP-ABE) to allow the decrypt or to provide filtering restrictions and the encryptor to describe access policies. The authors developed a PDPS strategy in cloud settings that relies on the BP-ABE. Using the PDPS method, the cloud server may safely evaluate both the subscription policy and the access policy. The PDPS strategy's strengths lay in the fact that it is both time-saving and risk-free when applied to typical models.

ISSN: 2366-1313

The matching was done before the decryption was carried out in the match-then-decrypt approach, which was presented by Yinghui Zhang et al. (2017). This procedure determined the out-of-the-ordinary parts of cypher texts and checked for a successful decoding only when the cypher texts' concealed access rules matched those of the attribute private key. For speedy decryption, we designed the remarkable attribute secret key components that aggregate pairings at decryption time. The authors introduced the Anonymous ABE and relied on a strong one-time signature to achieve the secure extension. Here, the minimum cost to run the attribute matching test was less than the minimum cost to complete the decryption. The number of pairings used in the attribute matching test has to be steady and low. The measurement of performance proved that this approach ensured attribute privacy and improved decryption efficiency. The slowness of the decryption procedure is a major downside of this technology.

Using Identity-Based Encryption (IBE) and CP-ABE, Xin Dong et al.(2014) presented a pliable, scalable, and efficient privacy-preserving data policy with semantic security. This technique ensured the anonymity of cloud users, provided robust data sharing security, and facilitated safe and effective dynamic activities like user attribute changes, file generation, and the like. In the random oracle model, the security analysis proved that this technique was safe since it enforced backward secrecy, complete collusion resistance, and granular access control in the generic bilinear group model. Expenditures were shown to be low using this strategy in both experimental and performance analyses. Size of cypher text, cost of transmission, and computational complexity were taken into account as assessment measures for this approach. This approach was more effective while using less resources. However, on the actual platform, the strategy did not succeed in protecting the privacy.

Ying-Tsung Lee et al. (2017) presented a smart home system that protects users' anonymity. This setup used community-based connections to link many data-concealment devices to a central house controller. The information was then included into the cloud's hierarchical design for data analytics access control. This technique used a large pool of smart house data apps to collect information from a smart community setting. From the first stages of system design all the way through the management of the data lifecycle, the protection and monitoring methodology was used to integrate the

ISSN: 2366-1313

privacy-enhancing approaches with the privacy-preserving approach. Different types of data from smart homes were collected and organised according to a community's hierarchy, including non-sensitive data, sensitive data, and identifying values. The benefits of this approach were improved data privacy and accessibility. This approach has the drawback that the hourly, daily, and weekly scales were split apart from the normal monthly, seasonal, and yearly statistical studies. The communities were divided according to administrative areas, which also meant that the distribution of socioeconomic levels was more uneven.

To ensure user anonymity in the cloud, Liu Wu et al. (2016) created the Reputation and Attribute-Based Dynamic Access Control (RA-DAC) system. This strategy was used to limit the malicious users' access to the cloud storage and services. The RA-DAC was then used as the foundation for the RA-DAC system (RADACS). By identifying the malicious acts of dishonest users and limiting their access to the cloud, RA-DACS ensured the safety of any data stored there. The study of the method's performance revealed that it provided a high level of security for the cloud.

The cloud storage method and the privacy protection of two kinds of characteristics were defined and addressed by Xia long Xu et al. (2016). The first category is "data attributes," which include such things as the transmission's route information, the data's size, the frequency with which it is sent, and the data's time stamps. The algorithmic description of the route data. The Tor method is widely used and is the most popular option. Additional improvements were needed to keep all data properties intact. The access frequency of the data was hidden and the frequency of the fake information's access was raised by the use of random false information and deterministic message production. Data packet size privacy preservation facilitated the transmission of data packets of uniform size. This technique was quite demanding on computer power.

III. RELATED WORK

Our method has close ties to two areas of cloud computing study: encrypted document retrieval and ciphertext-policy attribute-based document encryption. What follows is a compilation of works in these two areas that are connected to one another. Since Sahai et al. developed the identity-based encryption (IBE) method [19], numerous ABE systems have been suggested [20], [32]-[34], among which CP-ABE schemes [21–26], [35] are particularly promising because to their

ISSN: 2366-1313

adaptability and scalability. For these CP-ABE systems to work, each document requiring encryption must have a unique access structure. There has been much study into hierarchical attribute-based encryption [28]- [31], whereby a group of documents that share an access structure are encrypted as a unit. This has the potential to increase the efficiency and scalability of the encryption/decryption process. Theoretically proving its security, Wang et al. [28] offer a hierarchical attribute-based encryption system called FHCP-ABE. One benefit of the approach is that authorised users only need to calculate the secret key once in order to decode all encrypted official papers.

As a result, both the encryption and decryption processes may be completed in less time. High performance, fine-grained access control, scalability, and complete delegation are all features of the HABE method [29] designed by Wang et al. Hierarchical identity-based encryption and cypher text analysis-based encryption (HABE): a formidable duo. As an extension of ciphertext-policy attribute-set based encryption (ASBE), hierarchical attribute-set based encryption (HASBE) [30] is proposed by Wan et al. By adding a delegation mechanism to ASBE, the HASBE scheme may be easily integrated with a hierarchical structure of system users. In order to facilitate the hierarchical distribution and delegation of secret keys in big organisations, Deng et al. [31] expand ABE to CP-HABE. To counter the auxiliary input leakage attack, Guo et al. [36] offer a resilient-leakage hierarchical attribute-based encryption system, and they conduct a thorough analysis of the method's security.

We not only encrypt the files, but we also make an effort to do a thorough search inside the encrypted file.

As a result, our method has close ties to the problem of retrieving documents from encrypted collections using multiple keywords. Cao et al. [17] first offer a secure k-nearest neighbours (kNN)-based multi-keyword ranked search technique that protects user privacy. After establishing a stringent set of privacy standards, two approaches are offered to enhance both security and the search process. One obvious flaw with this approach is that it cannot be utilised to handle excessively large document databases since the search efficiency is linear with the cardinality of the document collection. Xia et al. [18] offer a "Greedy Depth-First Search" technique to increase search performance, and they use a keyword balanced binary (KBB) tree to structure the document vectors. Furthermore, the index tree may be

ISSN: 2366-1313

dynamically updated with a manageable amount of extra communication overhead. However, the document vectors are unsystematically arranged in the tree, thus there is room for improvement in terms of search efficiency. The index structure developed by Chen et al. [15] takes into account the connections between texts and is based on hierarchical clustering to boost search performance. Their method further incorporates a verification process to ensure accurate outcomes. While the index structure does allow for sub-linear search performance, it does not provide precise search results. In order to facilitate personalised search and enhance the user experience, Fu et al. [16] describe a personalised multikey word ranked search scheme in which an interest model of the data users is included into the document retrieval system. In order to accurately portray a user's actions, an interest model is constructed from her search history with the aid of WordNet [38]. However, because the document vectors are built using the statistical information of all the documents in the collection, this technique cannot enable dynamic update activities. Even if an MDB-tree is used to increase search efficiency, the tree's performance is difficult to foresee. An innovative attribute-based encryption technique (KSF-OABE) that allows for keyword search has been proposed by Li et al. [39]. Unlike our approach, KSFOABE cannot hierarchically encrypt a document collection nor does it provide efficient multi-keyword document retrieval, while sharing some of the same design goals.

IV. PERFORMANCE EVALUATION

Here, we assess the effectiveness of the ARF tree as a search tool and the hierarchical document encryption system. We begin with a theoretical efficiency study and then test our hypothesis experimentally.

A. Conceptual Dissection

We define certain terms before comparing our hierarchical encryption system to the CPABE scheme in [21] and the FH-CP-ABE scheme in [28]. In this context, the time required to perform a group operation, such exponentiation or multiplication, is assumed to be CGi (i = 0, 1). Take the cost of the bilinear map operation e to be Ce, and let Zp be the group [0, 1..., p 1]. Let N denote the total number of documents in the set, signify a parameter unique to Algorithm 1, and N denote the total number of nodes in all the access trees. Since many file IDs might share the same leaf in the access tree, must be less than 1. Let A be the attribute dictionary, Ui the data user, and Ci the document's attributes. Let ti denote the depth of the file Fi's

ISSN: 2366-1313

access tree in nodes. In addition, the number of items in is denoted by | |, and the length of each element in is denoted by L.

Assuming the data owner possesses N document files encrypted using CPABE, FH-CP-ABE, and our technique, we do an analysis. It is important to remember that our attention is directed towards the content keys themselves and not the documents that are symmetrically encrypted using those keys. The analytical result is shown in Table 1, and we further assume that a data user is in charge of decrypting all the documents. We have |A| (|AC1 |+ +|ACN |) and N N |t1|+ +|tN | for a large document set.

Therefore, our technique outperforms CPABE in terms of the sizes of PK, SK, and CT, as well as the time required for encryption and decryption. In terms of MSK size, the two systems perform similarly. In conclusion, as compared to CPABE, our method is faster and uses less space. Our technique has linearly growing encryption time, decryption time, and CT size for a fixed attribute set A and parameter. There is no correlation between the number of documents and the key sizes. In addition, our method performs better than FH-CP-ABE in terms of MSK, and it is on par with FH-CP-ABE in terms of PK and SK size. Since FH-CP-ABE was developed to encrypt a collection of documents with incremental attribute sets (AC1 > AC2 >... > > > ACN), it is impossible to predict the time cost of encryption and decryption as well as the size of CT for a collection of documents with randomly assigned attribute sets. Therefore, in Section VII, we will use simulations to evaluate our method against FH-CP-ABE.B. The search efficiency of a document collection is heavily influenced by its structure. The KBB tree [18] is a balanced binary tree that is optimised for searching certain keywords.

However, the document vectors are chaotically organised and put at random into the tree. It's possible for vectors that are identical to be located quite far apart in the tree, while vectors that are completely unlike might be neighbours. As a result, the tree's inner nodes don't have much to offer in terms of guiding a query vector to the region containing a cluster of highly relevant document vectors. In contrast, ARF trees tightly organise vectors according to their similarities, such that vectors with comparable features may always construct a cluster independently of the input order. A cluster containing relevant document vectors may be quickly located using the query vector. In a search procedure, the search percentage is the ratio of the number of nodes that were

ISSN: 2366-1313

actually searched to the total number of nodes in the search tree. For a quick overview of the differences between the two trees, see Fig. 7. Both the 2D and 3D document vectors are created at random. Since the KBB tree does not provide attribute limited search, we must disregard data user and document attributes. In both 2D and 3D settings, the ARF tree clearly outperforms the KBB tree. To be more precise, the ARF tree has a search percentage of roughly 5% to 10% compared to the KBB tree.

Simulation Experiments, Type B

On a real-world data set, the, we perform extensive experimental assessment of the proposed document retrieval technique.

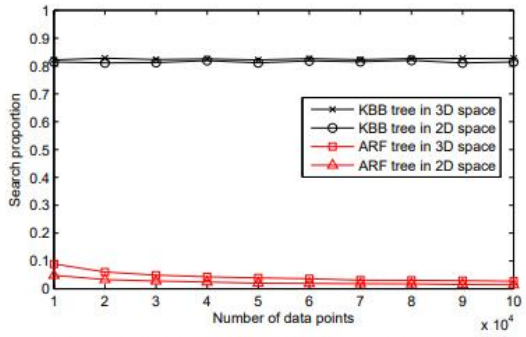

Fig. 7. Search proportion of KBB tree and ARF tree

Algorithm 3 AttributeGeneration.

Input: $\mathcal{A} = \{C_1, C_2, C_3, C_4\}, \mathcal{F}, p_r (0.25 \leq p_r \leq 1)$
Output: The attributes of each document

1: **for** each document $F_i \in \mathcal{F}$ **do**
2: $Att = \emptyset$;
3: Randomly select a number m from $\{1, 2, 3, 4, 5\}$;
4: Randomly select an attribute A_n from $\mathcal{A}$ and we assume that $A_n \in C_k, k = 1, 2, 3, 4$;
5: Insert A_n to Att;
6: **for** $i = 2 : m$ **do**
7: Randomly generate a number $p_r'(0 \leq p_r' \leq 1)$ and if $p_r' \leq p_r$, randomly select an attribute A_q from C_k; otherwise, uniformly randomly select an attribute A_q from $\mathcal{A}$;
8: Insert A_q to Att;
9: **end for**
10: The attributes in Att is defined as the attributes of document F_i;
11: **end for**

the building procedure. Therefore, for the sake of simplicity, we will define = 1 while building the ARF tree in the following. In addition, k is fixed at 10 (i.e., a query returns 10 encrypted documents). However, Fig.13 shows the simulated outcome after using characteristics to search for documents. The search time in MRSE seems to grow linearly with the number of files, given that the document vectors are organised randomly and all the document vectors need to be scanned at once. The ARF tree, on the other hand, sorts files according to their commonalities, making searches far more effective. In particular, the search process involves pruning a large number of search pathways, and ARF tree's time complexity grows logarithmically with the number of files.

V. CONCLUSION

In this research, we focus on a novel encrypted document retrieval situation where the data owner desires granular control over the documents. We begin by developing a unique hierarchical attribute-based document encryption strategy to encrypt a group of documents that have a common access structure and so need encryption. In addition, we suggest using the ARF tree to hierarchically organise the document vectors according to their commonalities. Finally, a depth-first search method is developed to enhance the search

ISSN: 2366-1313

efficiency of the data consumers; this is crucial for huge document collections. The method's efficacy is measured in several ways, both theoretically and experimentally.

Several enhancements may be made to the suggested scheme: In this study, we suppose that each node in the access trees represents an "AND" gate, which restricts how freely the characteristics may be assigned to the documents. We want to eventually include "OR" gates into the access trees. Second, we will examine if the greedily-generated access structure of the document collection may be further optimised to reduce the number of access trees. In addition, a procedure for revoking the characteristics of data users must be developed. The third step is to suggest an update strategy for the ARF tree. The ARF tree has inherent support for adding new nodes to the tree, but it lacks a mechanism for removing existing nodes. To examine the impact of parameter on the strategy, it is necessary to create a new document collection in which each file is tagged with the appropriate characteristics and then perform extensive experiments on the collection.

REFERENCES

[1] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," IEEE Internet Computing, vol. 16, pp. 69–73, Jan. 2012.

[2] D. X. Song, D. Wagner, and A. Perrigo, "Practical techniques for searches on encrypted data," in Security and Privacy, 2000. SandP 2000. Proceedings. 2000 IEEE Symposium on, pp. 0–44, 2002.

[3] E. J. Goh, "Secure indexes," Cryptology ePrint Archive, http:// eprint.iacr.org/2003/216., 2003.

[4] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in ACM Conference on Computer and Communications Security, pp. 79– 88, 2006.

[5] J. Li, Y. Shi, and Y. Zhang, "Searchable ciphertext-policy attribute-based encryption with revocation in cloud storage," International Journal of Communication Systems, vol. 30, no. 1, 2017.

[6] Y. Miao, J. Ma, X. Liu, X. Li, Q. Jiang, and J. Zhang, "Attribute based keyword search over hierarchical data in cloud computing," IEEE Transactions on Services Computing, vol. PP, no. 99, pp. 1–1, 2017.

[7] A. Swaminathan, Y. Mao, G. M. Su, H. Gou, A. L. Varna, S. He, M. Wu, and D. W.

ISSN: 2366-1313

Oard, "Confidentiality-preserving rank-ordered search," in ACM Workshop on Storage Security and Survivability, Storagess 2007, Alexandria, Va, Usa, October, pp. 7–12, 2007.

[8] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," IEEE Transactions on Parallel and Distributed Systems, vol. 23, pp. 1467–1479, Aug. 2012.

[9] S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber +r : top k retrieval from a confidential index," in International Conference on Extending Database Technology: Advances in Database Technology, pp. 439–449, 2009.

[10] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," Lecture Notes in Computer Science, vol. 3089, pp. 31–45, 2004.

[11] B. Dan and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in Theory of Cryptography Conference, pp. 535–554, 2007.

[12] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption," in International Conference on Theory and Applications of Cryptographic Techniques, pp. 62–91, 2010.

[13] 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. PP, no. 99, pp. 1–1, 2017.

[14] Y. Miao, J. Ma, X. Liu, Q. Jiang, J. Zhang, L. Shen, and Z. Liu, "Vcksm: Verifiable conjunctive keyword search over mobile e-health cloud in shared multi-owner settings," Pervasive and Mobile Computing, vol. 40, pp. 205–219, 2017.

[15] C. Chen, X. Zhu, P. Shen, J. Hu, S. Guo, Z. Tari, and A. Zomaya, "An efficient privacy-preserving ranked keyword search method," IEEE Transactions on Parallel and Distributed Systems, vol. 27, pp. 951–963, Apr. 2016.

[16] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," IEEE Transactions on Parallel and Distributed Systems, vol. 27, pp. 2546–2559, Sep. 2016.

[17] N. Srivani, Dr Prasadu Peddi, "Efficient Fr a Geometrical-Model-Based Face Segmentation and Identification in

ISSN: 2366-1313

Terms of Identification the Face ", *JFCR*, pp. 1283-1295, Jun. 2022.