AN EFFICIENT HYBRID SECURED MULTI-KEYWORD RANKED SEARCH (EHMRS) OVER ENCRYPTED CLOUD DATA

[1]Dr. K. Sivakumar, [2]Mr. A. Ashikali, [3]Mrs. V. Pavithra, [4]Ms. A. Lakshmi
[1]Asso. Prof. & Head, PG Department of CS, Kathir College of Arts and Science, Cbe.
[2]Assistant Professor, PG Department of CS, Kathir College of Arts and Science, Cbe.
[3]Assistant Professor, PG Department of CS, Kathir College of Arts and Science, Cbe.
[4]Assistant Professor, PG Department of CS, Kathir College of Arts and Science, Cbe.

Abstract: Individuals and businesses are encouraged to move from in-house storage systems to cloud storage services due to the numerous advantages of cloud storage such as greater flexibility, faster deployment, availability, backup and disaster recovery facility, and lower operating costs. Recent news has shown that by using cloud services, users' privacy is not secure. All the applications are used separate encryption mechanism for before storing the data into the cloud environment. It is mainly used for the purpose of privacy. There have been important and ongoing advances in cloud computing security, but protecting cloud data from unauthorized access outside the boundaries has been a significant challenge. Given the real need for cloud-based file access, a mechanism that can in demand of security and productivity was needed. The task of designing an effective multi-keyword searchable encryption mechanism has piqued the researchers' interest, and this has been the primary focus of this research work. The proposed work addresses the problem of securing data in the cloud at the data owner's end by encrypting it and making the data accessible to any approved data user. The proposed work implements an encrypted hybrid secured multi-keyword ranked search (EHMRS) technique for performing multi-keyword searches in a safe manner and extracting necessary documents containing searched keywords in a shorter amount of time. As a result, the performance of the proposed EHMRS technique on data access on the cloud is authenticated, which improves security.
Keywords: - Cloud Data, Multi-Keyword Search, Encryption, Decryption, Index.

1 INTRODUCTION

The cloud has a vast space for store data, causing businesses to transfer their valuable information or records to a set of connections of distant servers located over the internet. When evaluated to a neighbouring server and PC, various functions like that information storage, maintenance, control, and admittance are conducted in a cloud environment. The NIST detects five critical features in the cloud, including required service, wide set of connections admittance, resource grouping, quick stretch, and calculated facility. In recent days, cloud services have received a lot of attention from organizations and the general public for six reasons: cost, pace, global scope, and efficiency and performance organizations. The cloud architecture includes two basic components: one is called front-end and another one is called

back-end [18]. The front- end consists of client/server, device, and applications that are utilized to process data on the cloud through a UI like that a web browser. The back-end then includes a variety of processors, servers, and data storage devices.

Cloud storage is act as key services to ensure access, power, and on-demand remote backup of data to cloud users via the internet. The main benefit of cloud storage is that it allows for universal document access, improved data security, and easier community collaboration. In manufacturing companies, cloud storage is applied to store personal information, share records between two organizations, and distribute relevant information.

The secure retrieval system is a critical component in providing improved productivity in secure data communication. By preventing unauthorized users in the cloud environment, safe process of information stored in the cloud is achieved. Unauthorized users are detected and prevented on the cloud by maintaining proper account and access control authority and privacy [4]. Due to the lack of user authentication and authorization in the cloud, protecting cloud data from unauthorized users is a difficult challenge. The keyword extraction method is the simplest way to extract the required documents/files related to the searched keywords. Many researchers have concentrated on providing reliable keyword search to access data stored on cloud storage by implementing authentication protocols and cryptography techniques such as encryption and decryption algorithms [21].

Yu, J., et al. use SSE to resolve data privacy concerns. The authors develop a multi TRSE framework that supports top-k multi-keyword retrieval. TRSE employs a vector space model and cryptography. As a result, information leakage can be prevented, and data security is ensured, but file recall cannot be guaranteed. [1]. Rizomiliotis, P., et al, introduces REX, a modern Searchable Symmetric Encryption mechanism that supports range queries. REX is a one-round collaborative and response-hiding mechanism. Its contact and search computation complexity are optimal. [2]. Xiangyang, Z., et al. suggested PPMKTS over encrypted cloud data, which is focused on hierarchical agglomerative clustering. It increases text search efficiency by using hierarchical agglomerative clustering and the HAC tree. However, it evaluates document similarity directly in the word-count space, which can be time-consuming for broad vocabularies. [3].

Among these, multi-keyword ranked search is gaining popularity due to its functional applicability. Recently, several complex mechanisms for supporting adding and deleting operations on document collections have been proposed. These are significant tasks because it is the responsibility of data owners should update their data on the cloud server. However, only a minimum of the dynamic mechanisms support effective multi-keyword ranked search. Furthermore, an adaptively protected Searchable Symmetric Encryption (SSE) mechanism was built in order to achieve a privacy-preserving keyword search, data storage, and retrieval mechanism on the cloud [5]. The time required by an adaptively safe SSE mechanism to perform keyword extraction was excessive.

With the aim of addressing the existing issues, the proposed work introduces the encrypted hybrid protected multi-keyword ranked search (EHMRS) technique for secured multi keyword search on encrypted cloud data platform by ensuring proper authority and authentication. This proposed work focuses on giving the user the ability to selectively scan the relevant files without any security leakage, by effectively searching through the encrypted cloud data and decrypting only at the user's end, and it also makes an effort to propose a searchable encryption

mechanism for safe data outsourcing that is not only powerful and secure, but also dynamic in nature.

The rest of the work is structured as follows. Explain the various models for the problem statement used in the proposed mechanism in section 2. In Section 3 illustrate the overall approach to constructing the appropriate search mechanism. Section 4 compares the performance analysis of the proposed EHMRS methodology to existing methods. Finally, section 5 summarizes the proposed work.

2 PROBLEM FORMULATION

2.1 System and Threat Model

The primary reason for the proposed SE mechanism is to locate relevant information from the cloud by comparing a pre-computed index to a user's search query [6][7]. There are three types of entities in the proposed stable ranked mechanisms' system architecture: Data Vendor (DV), Data User (DU), and Cloud Server (CS). The DV has a file collection DC = {dc1, ...,dcn} and farm out it in an encrypted structure to the CS. therefore, initially the DV take outs a geoup of discrete keywords kw = {kw1, ...,kwm} and significance rank values (to enable ranking) from the DC and constructs an encrypted searchable inverted index, I. Then, the DV creates the encrypted DC = {dc1, ...,dcn} using a symmetric encryption mechanism (e.g., AES), and suggests both the encrypted document collection DC and the index I to the CS.

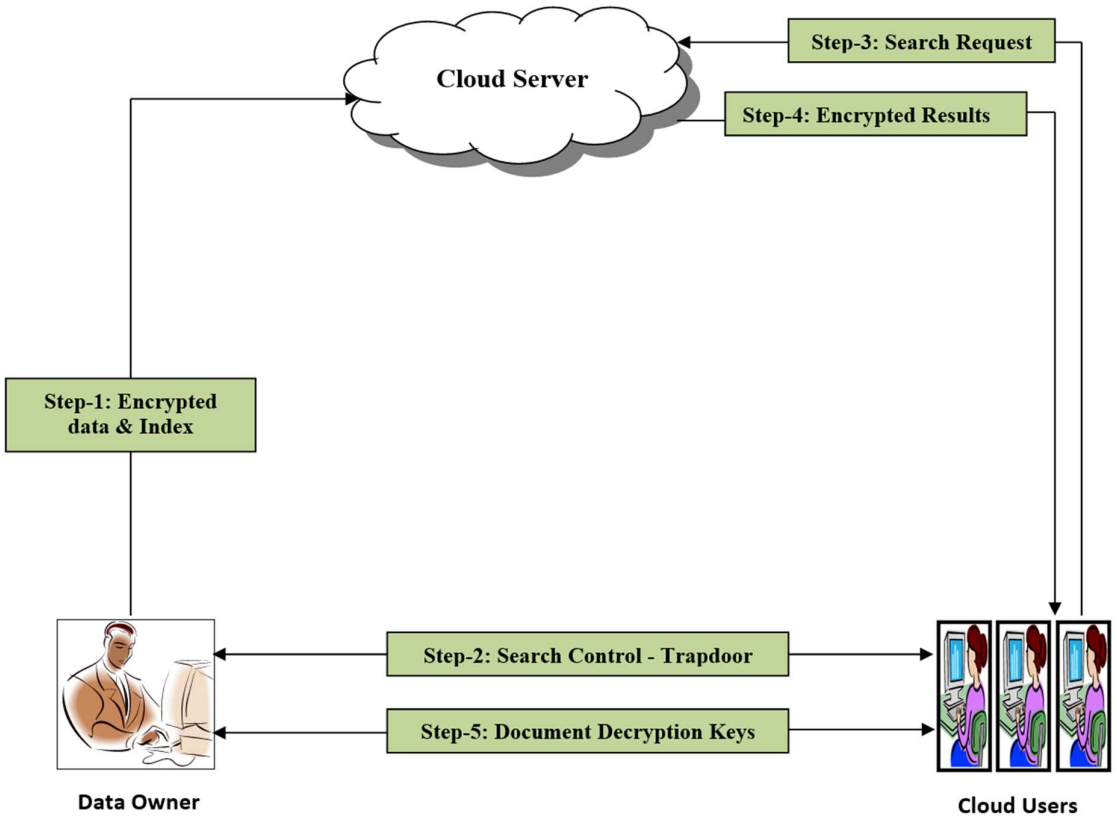

Fig. 1: - System Model

The purposes of the three entities are revealed in Figure 1. An outsider is believed to be aware of the connection among the server and the owner/user [8]. At the same time, it is assumed that the intermediary is intensely involved in intercepting the useful information from the contact. Consider the following two threat models with varying attack capabilities:

Known Ciphertext Model - The cloud server is now only expected to know encrypted dataset C and searchable index I in this design, all of which are exported from the data vendor. [12].

Known Background Model - The cloud server is expected to have more expertise in this better design than what can be obtained in the known ciphertext model. Such details may include the significant correlation of specific search requests (trapdoors), as well as dataset-related statistical data. [13].

2.2 Design Goals

Under the previous methods, the proposed mechanism has the following design strategic goals: to enable secure, efficient, precise, and scalable multi-keyword ranked search over outsourced encrypted cloud data. [9][10].

Searching for Ranked Multiple Keywords- This mechanism allows for the extraction of data using a query of multiple keywords and a similarity rank.

Privacy – This search mechanism should satisfy privacy criteria by preventing the outsider from learning more information from the intercepted data.

Maximization of Efficiency – The proposed design could achieve privacy and functional objectives with low data transmission and computation costs. Furthermore, as file updates occur, the searchable index and encrypted records must be updated quickly and efficiently.

Multi-keyword Ranked Search – Provide information about the privacy enhancements here.

3 THE PROPOSED MECHANISM

3.1 Overview of the Mechanism

Using conceptual extension, define the EHMRS in this section. To begin, use the Enhanced Term Frequency-Inverse Document Frequency analysis to obtain keywords through external provider records and create indexes (E-TFIDF). This is the most efficient method for producing a more accurate index. The data owner encrypts the outsourcing data and uploads it to the CS using the Elliptic Curve Cryptography (ECC) algorithm. [11]. Whenever a server receives a query to the cloud server, the index tree, also known as the iterative deepening search tree, searches and ranks the relevant files in order to obtain a particular document that contains keywords. The authorized user is then shown the highest query results. The procedure is divided into two stages: preparation and retrieval.

3.2 Index Construction

Create a tree node for each document in the set before beginning the index creation process. The index tree's leaf nodes are represented by these nodes. The internal tree nodes are then created using these leaf nodes. Algorithm 1 depicts the index's formal construction process.

Enhanced semantic function extraction (Term Frequency-Inverse Document Frequency) E-TFIDF method was implemented for multi-keyword ranked search on encrypted cloud data

with synonym queries. For extracting the encrypted cloud platform, the E-TFIDF method uses synonyms queries by performing a semantics-based multi keyword ranked search. During the synonym-based multi keyword ranked search, cloud users submit feedback as synonyms of predefined keywords rather than specific or fuzzy matching keywords because they lack precise data details. Improved semantic feature extraction The E-TFIDF method evolved from the TFIDF (term frequency-inverse document frequency) method for feature extraction [14]. The most common keywords were extracted from outsourced documents on an encrypted cloud data platform using the E-TFIDF process. As a result, the accuracy of search results in an encrypted cloud environment has been improved. The E-TFIDF approach was introduced in the cloud environment with two important goals in mind: enhanced keyword extraction method and building keyword collection expanded by synonym. The details for Algorithm 1 are as follows. Besides which, the tree node data structure is identified as (ID; D; Pl; Pr; FID), where the unique identity ID for each tree node is provided using the function GenID ( ).

Algorithm 1: - Indexcons

Step-1 *Scanning file sets FS = $fs_1, fs_2,\ldots,fs_n$.*
Step-2 *End filtering words; remove unnecessary punctuation and detail.*
Step-3 *Extraction of keywords from FS to construct a set WD = $wd_1, wd_2,\ldots, wd_m$ and index I.*
Step-4 *Insert the file's id that corresponds to the keywords into the indexical filelist.*
Step-5 *Weight Matrix (WM) find out IDF*
If IDF = zero then
get rid of the word from the Wordlist.
get rid of the consequent TF from the WM.
Else
Compute TF/IDF and store normalized.
TF/IDF in the consequent factor of the WM.
Step-6 *return*

The proposed algorithm (I-TF/IDF) improves the position weighted to enhance keyword extraction precision and avoid conceptual variance during corresponding conceptual extension. The keyword score is calculated using Equation (1).

$$\boldsymbol{score} = \boldsymbol{N_i} * \boldsymbol{log}\left(\frac{\boldsymbol{n}}{\boldsymbol{n_i}} + \boldsymbol{\beta}\right) * \boldsymbol{m_i} \times \frac{\boldsymbol{l_{wi}}}{\boldsymbol{L}} \qquad \ldots \boldsymbol{equ}(\mathbf{1})$$

The frequency of keywords kwi in a file fcj is indicated by the value Ni. The total No. of files is represented by n. The number ni reflects the total No. of keywords kwi that appear in the file. ß denote an objective fact. lwi is the paragraph number in the file where the keyword appears. L represents the total No. of paragraphs. mi is the file's keyword location weight.

3.3 Encrypt Data

To protect the privacy of the outsourced data, the proposed work incorporates the Elliptic Curve Cryptography (ECC) algorithm [15]. This algorithm takes reduced processing and speeds up encryption and decryption compared to public key cryptography. Through uploading encrypted files and indexes to the cloud server, it will reduce the computational cost of encryption and decryption. It can then authorize users to build trap doors and send them to

the Cloud Service Provider (CSP) to check for matched ciphertext using the hash conflict feature [16]. EP is a real-number field non-singular elliptic curve. This is shown by the equation (2).

$$\boldsymbol{Y^2 = x^3 + ax + b(mod\ p)} \quad \boldsymbol{\dots equ(2)}$$

Here p is a prime number. Kp is a big prime region. Kp's elements are, a, b, and p. The elliptic curve also intersects 4a3 + 27b2! = 0. Ep(a, b) denotes an elliptic curve.

Following the formation of the index and the creation of the key, the groups of documents are authenticated with the ECC to preserve the security of the result found. The authenticated protocol is as follows. Let F stand in for the paper. F = m1, m2,..., mn Where F has a duration of n. And mi is a binary string that also serves as an index. Equation shows how to compute the ciphertext (3).

$$\boldsymbol{C_i = m_i + r * PK_{server}} \quad \boldsymbol{\dots equ(3)}$$

Here Ci denotes the ciphertext obtained by encrypting the point G PKserver is the CS's public key. The algorithm for encryption is as follows.

Algorithm 2:- Encryptdata

Step-1 *Read a character from a register.*

Step-2 *Save the ASCII value of each character into an integer variable.*

Step-3 *The integer-corresponding point on the Elliptic Curve is chosen.*

Step-4 *Encrypts the point with the user's public key B=n*A, and the corresponding encrypted point is E.*

Step-5 *Connect point E' to the database to obtain a new integer. K1 is the new integer that corresponds to E'.*

Step-6 *This K1 is transformed into data with two parameters:*

- ***a.*** *A printable ASCII character ($) that serves as an index.*
- ***b.*** *The page number (for example, N=1) to which the corresponding index belongs.*

Step-7 *Two parameters are sent into two separate files, which are then transmitted.*

The message will be encrypted by the server using the user's public key. Enable the message that needs to be sent to be sent. This message should be depicted by a curve.

3.4 Search Process of EHMRS Mechanism

The proposed EHMRS mechanism's search protocol is the "Iterative Deepening Depth-First Search" (ID-DFS) algorithm, which is a repetitive process on the tree. IDDFS integrates depth-first search's space-efficiency with breadth-first search's speed. IDDFS invokes DFS for different depths beginning with a specified value. DFS is prohibited from going beyond the depth specified in each call. So, essentially, to do DFS in a BFS manner [17]. The algorithm's basic principle is to begin with a start node and then examine the node's first child. It then looks at that node's first child (grandchild of the starting node), and so on, until a node has no more children. If the goal node has not been reached after returning from the last child of the start node, the goal node cannot be found because all nodes have been traversed by that stage.

The R_Score is the document fFID's relevance score to the query in this scenario. The R_List

saves the last k documents obtained with the top relevance scores to the query. The components of the list are ranked descending depending on the R_Score and are real - time basis during the initial search. Other equations are as follows, and Algorithm 3 defines the IDDFS algorithm.

R_Score(Dn;Q) - The feature described in Formula to measure the relevance score for query vector Q and index vector Dn contained in node u. (1).

Kth_score - The lowest significance score in the present R List, which is set to 0.

H_child - A tree node's child node with a higher relevance ranking.

L_child - A tree node's child node with a smaller relevance ranking.

Algorithm 3:- IDDFS(IndexTreeNode n)

```
Step-1   if the node n is not a leaf node then
Step-2       if R_Score(Dn, Q) >
             kth_score then
Step-3           IDDFS(n.h_child)
Step-4           IDDFS(n.l_child)
Step-5       else
Step-6           return
Step-7       end if
Step-8   else
Step-9   if R_Score(Dn, Q) > kth_score then
Step-10  remove the element with the lowest relevance score from R_List;
Step-11  IDDFS(tree)
Step-12      for depth = 0 to infinity
Step-13          if (DLS(tree, depth))
Step-14              return true
Step-15      return false
Step-16  Insert a new element (R_Score(Dn,Q);n.FID) and sort all the elements of R-List;
Step-17      end if
Step-18      return
Step-19      end if
```

3.5 Retrieval Process of EHMRS Mechanism

The key to storing keywords is the cloud storage server, which is denoted by the letter KW. KW = (keyword0, keyword1, keyword2, and keyword3). Analyze the assistance for a specific itemset by navigating each document and determining whether or not it contains an itemset. It is impractical to repeat the above equation for a dataset with 'n' words since these datasets have (2n – 1) * (2n – 1) combinations. The current Apriori algorithm requires the generation of candidate itemsets. If the itemset in the database is high, these itemsets may be numerous.

The proposed EHRMS mechanism uses the FPgrowth algorithm to locate frequent itemsets in a transaction database without generating candidates. The FPgrowth algorithm represents the database as a tree known as a frequent pattern tree or FP tree. The relationship between the itemsets will be maintained by this tree structure. The database is fragmented by a common object. This fragmented portion is referred to as a "pattern fragment." These scattered patterns' itemsets are examined. To handle the file collections as a transaction T. Consider keywords to be a set of objects.

Algorithm 4: - FPgrowth Algorithm

Step-1 Scanning the database for occurrences of the itemsets in the database.
Step-2 Construct the FP tree. Establish the tree's root for this. The root is defined by the letter null.
Step-3 Re-scan the database and review the transactions.
Step-4 The database transaction is checked.
Step-5 Mine the newly formed FP Tree.
Step-6 Create a Conditional FP Tree based on the number of itemsets in the route.
Step-7 The Conditional FP Tree generates regular Patterns.

Finally, the search results will include keywords that are semantically important. The top-k files are returned by the CS. This query's findings would be more relevant to the user's needs.

3.5.1 Ciphertext retrieval

Authorized users are using the trapdoor to access files on the cloud server in the retrieval phase. The keyword's trapdoor Twi is determined using the data owner's private key PRs. Equation depicts the trapdoor calculation process (4).

$$\boldsymbol{T_{w_i} = \sum_{i=1}^{m} H\,(wi)^r} \qquad \boldsymbol{\ldots equ(4)}$$

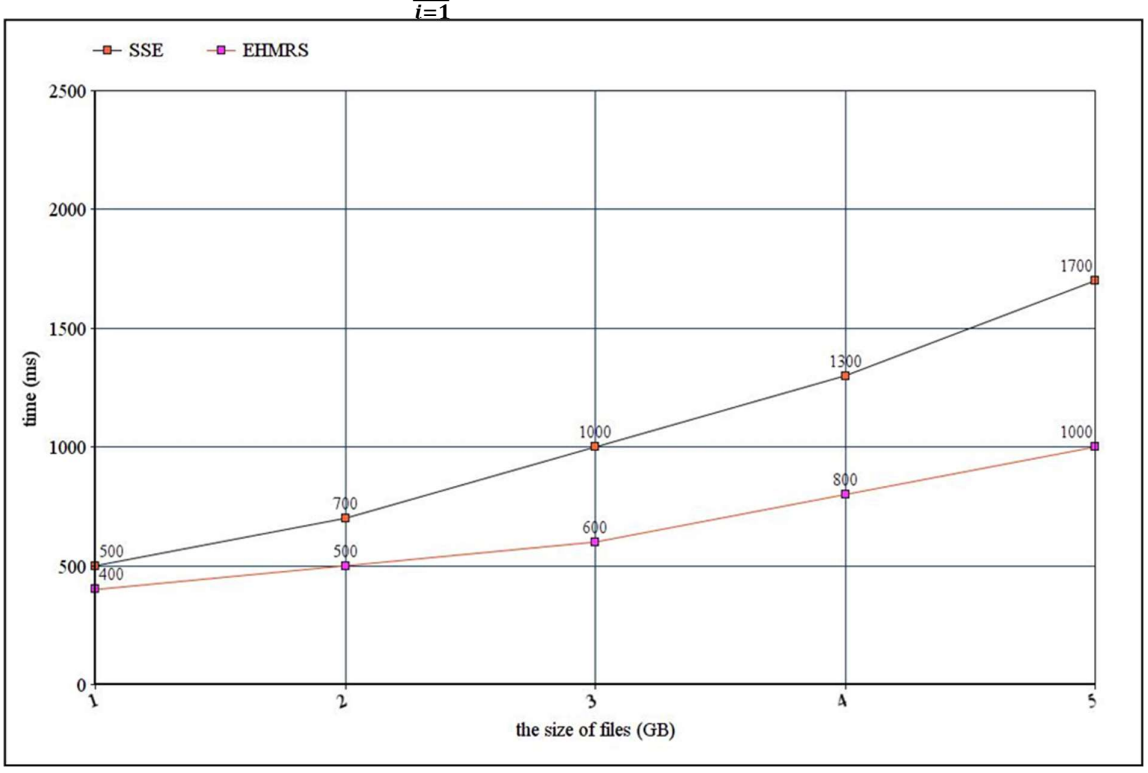

Fig. 2: - Comparison of Encryption Time

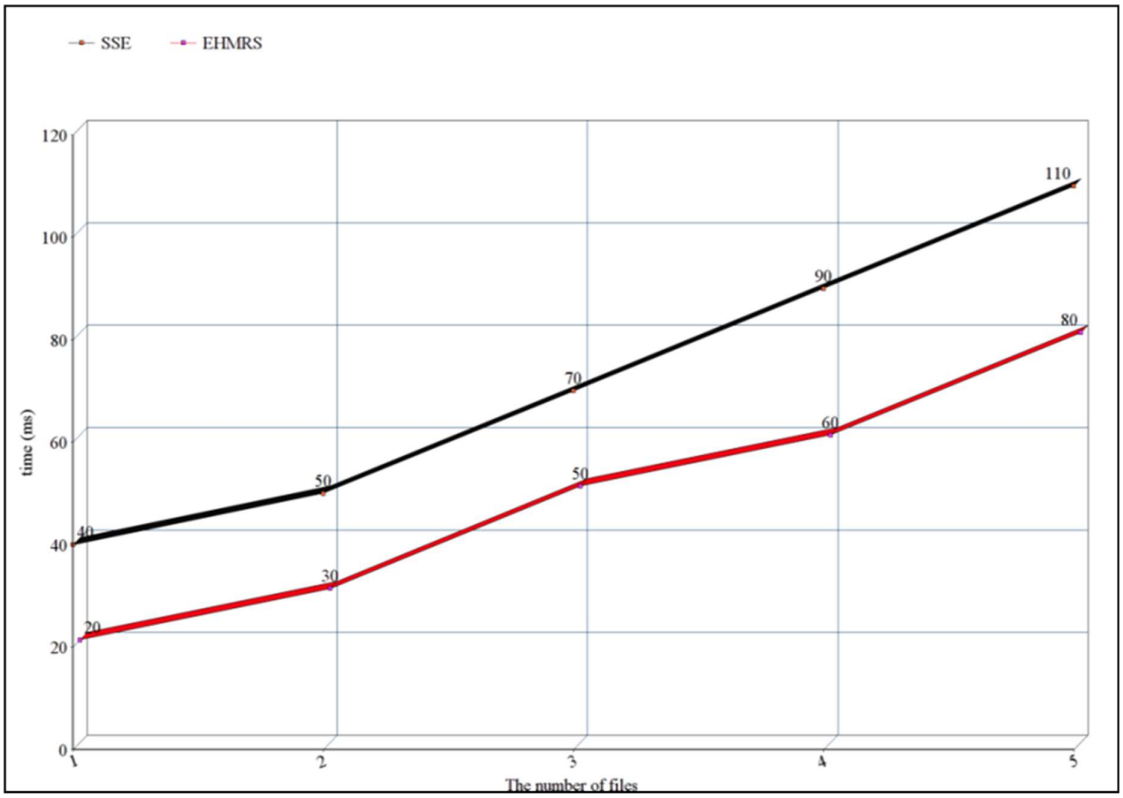

Fig. 3: - Comparison of build Index time

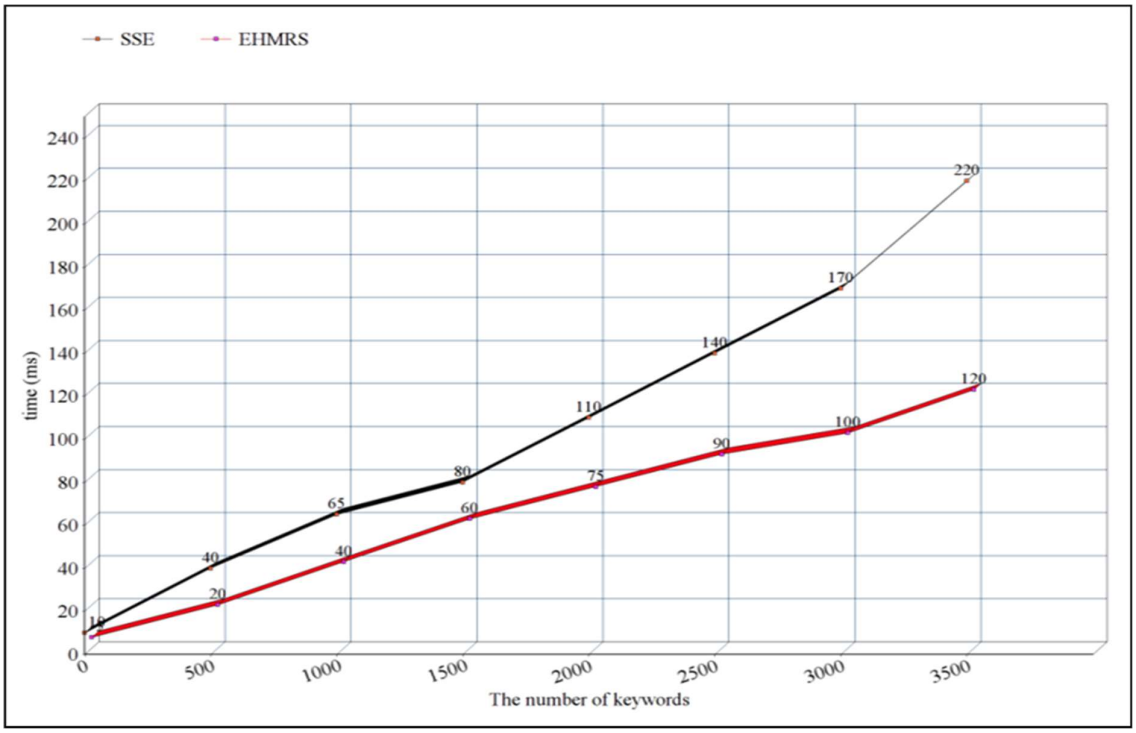

Fig. 4: - Comparison of trapdoor generation among different set of keywords

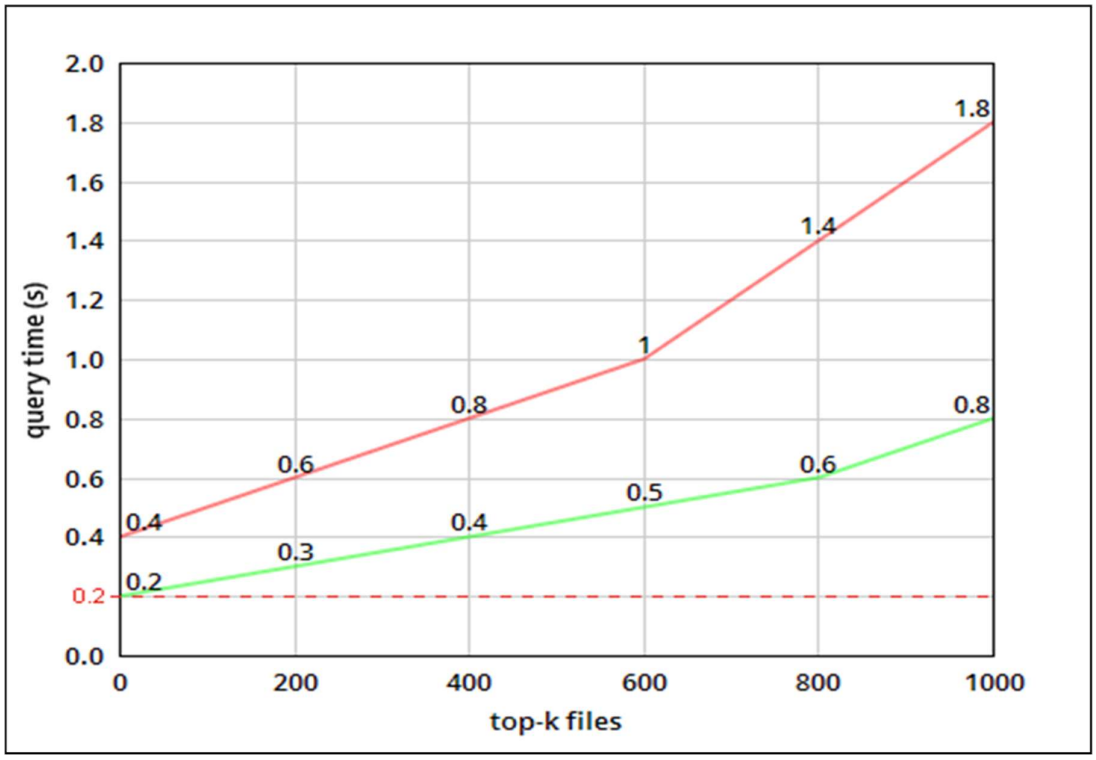

Fig. 5: - The top-k results retrieve

The search request is processed by the cloud in two stages. First, the cloud compares the searched keywords to all keywords stored on it, and a candidate file set is generated. Second, the cloud ranks the candidate file set's files and selects the top-k most relevant files. In order to obtain the dataset Sw, CSP extends the keyword. After trapdoor processing, it is routed to CSP for index matching. The top-k cipher texts that fit the index are then returned to the approved users. Otherwise, the Ciphertext set does not contain the files that approved users will search. PRs are private keys supplied by the data holders. CSP's public key is given by PKserver [20]. Authorized users decode the obtained Ciphertext using PRs and the PKserver. Finally, we obtain the target plaintext set L. Ci - s * r * G = M is the decryption equation. To encrypt data, the ECC algorithm is used.

4. PERFORMANCE EVALUATION

To assess the overall performance of the proposed methodology, JAVA was used on a Linux server with an Intel Core i3 processor running at 3.3 GHz. The Cranfiled dataset is used for the experimental evaluation of the proposed EHMRS methodology and current methods. Following keyword extraction, compute keyword statistics such as keyword frequency in each file, file duration, number of files containing a given keyword, and so on. Based on these figures, measure the relevance score of a keyword to a file. During the experiment, the number of files is considered to be between 50 and 500, and the number of keywords is considered to be between 100 and 1000. Furthermore, the file size is estimated to be between 50 and 230 KiloBytes (KB). The proposed EHMRS technique's efficiency is compared to the current adaptively stable Searchable Symmetric Encryption (SSE) mechanism [19].

4.1 Performance of encryption time and build index time

To allow the cloud server to perform a secure search among data encrypted with different secret keys belonging to multiple owners. ECC is used to encrypt data for its higher reliability, faster processing speed, and reduced storage space. The results of the experiments are now visible. Figure 2 depicts the encrypted time of files. The proposed EHMRS encrypts files in less time than the SSE.

Figure 3 depicts the time it takes to build an index in a different number of files. To obtain the keywords' package, all files must be scanned. And then create an inverted index with the appropriate values.

4.2 Performance of trapdoor generation

The keyword is used by approved users to generate the trapdoor. Experiment results are available. The development of a trapdoor necessitates a vector splitting operation as well as two multiplications of a (m x m) matrix, resulting in a time complexity of O. (m2),

Figure 4 show that when the dictionary size is set, the number of query keywords has little impact on the entropy of trapdoor generation. The EHMRS mechanism has a reduced time cost than the SSH mechanism due to the dimension extension.

The retrieval process involves keyword extensions, the search index, as well as the retrieval of the top-k files relevant to the query request. Figure 5 illustrates the time required to return the top-k file when the size of SW is 6.

5. CONCLUSION

The proposed encrypted hybrid protected multi-keyword ranked search (EHMRS) technique is implemented to perform safe hybrid search on an encrypted cloud platform while ensuring user authentication. The proposed EHMRS technique is used throughout the keyword search by combining the enhanced TF-IDF, Elliptic Curve Cryptography (ECC), Interactive Deepening – Depth First Search (IDDFS), and FPgrowth algorithms. Initially, the data owner stores encrypted files and an index list on a cloud platform. By comparing the search token and index list, the generation of tokens for keywords aids in the extraction of similar documents from the encrypted cloud setting. The cloud server employs an index-based tree traverse search mechanism to ensure quick access to the necessary document while maintaining high privacy. The proposed EHMRS technique uses the FPgrowth algorithm to scan the ranked based retrieval of files to protect cloud data from unauthorized users. As a result, the protection of cloud data is improved. Furthermore, the proposed EHMRS technique's efficiency is compared to that of the current SSE system.

6. REFERENCES

[1] Yu, Jiadi, Peng Lu, Yanmin Zhu, Guangtao Xue, and Minglu Li. "Toward secure multikeyword top-k retrieval over encrypted cloud data." IEEE transactions on dependable and secure computing Volume 10, Issue 4: 239-250, July, 2013.

[2] Rizomiliotis, Panagiotis, Eirini Molla, and Stefanos Gritzalis. "REX: A searchable symmetric encryption scheme supporting range queries." In Proceedings of the 2017 on Cloud Computing Security Workshop, Dallas Texas USA, November, 2017. pp. 29-37.

[3] Xiangyang, Zhu, Dai Hua, Yi Xun, Yang Geng, and Li Xiao. "MUSE: an efficient and

accurate verifiable privacy-preserving multikeyword text search over encrypted cloud data." Security and Communication Networks, Special Issue, July, 2017.

[4] Zhang, Bo, and Fangguo Zhang. "An efficient public key encryption with conjunctive-subset keywords search." Journal of Network and Computer Applications, Volume 34, Issue 1: 262-267, January, 2011.

[5] Li, Jiguo, Yuerong Shi, and Yichen Zhang. "Searchable ciphertext-policy attribute-based encryption with revocation in cloud storage." International Journal of Communication Systems, Volume 30, issue 1, January, 2017.

[6] Leontiadis, Iraklis, and Ming Li. "Storage efficient substring searchable symmetric encryption." In Proceedings of the 6th International Workshop on Security in Cloud Computing, Incheon Republic of Korea, May, 2018. pp. 3-13.

[7] Pasupuleti, Syam Kumar, Subramanian Ramalingam, and Rajkumar Buyya. "An efficient and secure privacy-preserving approach for outsourced data of resource constrained mobile devices in cloud computing." Journal of Network and Computer Applications, Volume 64, issue C, pp 12 – 22, April, 2016.

[8] Yang, Kan, Kuan Zhang, Xiaohua Jia, M. Anwar Hasan, and Xuemin Sherman Shen. "Privacy-preserving attribute-keyword based data publish-subscribe service on cloud platforms." Information Sciences, Volume 387 issue C, 116-131.: May, 2017. doi:10.1016/2016.09.020.

[9] Wang, Tianhao, and Yunlei Zhao. "Secure dynamic SSE via access indistinguishable storage." In Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, Xi'an China, May, 2016. pp. 535-546.

[10] Wang, Wei, Peng Xu, Hui Li, and Laurence Tianruo Yang. "Secure hybrid-indexed search for high efficiency over keyword searchable ciphertexts." Future Generation Computer Systems, Volume 55, Issue C: 353-361, February, 2016.

[11] Kamara, Seny, Charalampos Papamanthou, and Tom Roeder. "Dynamic searchable symmetric encryption." In Proceedings of the 2012 ACM conference on Computer and communications security, Raleigh North Carolina USA, October, 2012. pp. 965-976.

[12] Curtmola, Reza, Juan Garay, Seny Kamara, and Rafail Ostrovsky. "Searchable symmetric encryption: improved definitions and efficient constructions." Journal of Computer Security, Volume 19, Issue 5: 895-934, September, 2011.

[13] Wang, Cong, Ning Cao, Kui Ren, and Wenjing Lou. "Enabling secure and efficient ranked keyword search over outsourced cloud data." IEEE Transactions on parallel and distributed systems, Volume 23, Issue 8: 1467-1479. August, 2012.

[14] Li, Jiguo, Yuerong Shi, and Yichen Zhang. "Searchable ciphertext-policy attribute-based encryption with revocation in cloud storage." International Journal of Communication Systems, Volume 30, Issue 1: January, 2017.

[15] Armbrust, Michael, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee et al. "A view of cloud computing." Communications of the ACM, Volume 53, Issue 4: 50-58. April, 2010.

[16] Lu, Yanbin. "Privacy-preserving Logarithmic-time Search on Encrypted Data in Cloud." In NDSS. February, 2012.

[17] Miao, Yinbin, Jianfeng Ma, Fushan Wei, Zhiquan Liu, Xu An Wang, and Cunbo Lu. "VCSE: Verifiable conjunctive keywords search over encrypted data without secure-channel."

Peer-to-Peer Networking and Applications, Volume 10, Issue 4: 995-1007. July, 2017.
[18] Khan, Neelam S., C. Rama Krishna, and Anu Khurana. "Secure ranked fuzzy multi-keyword search over outsourced encrypted cloud data." In 2014 International Conference on Computer and Communication Technology (ICCCT), IEEE, India, June, 2014. pp. 241-249.
[19] Rane, Deepali D., and V. R. Ghorpade. "Multi-user multi-keyword privacy preserving ranked based search over encrypted cloud data." In 2015 International Conference on Pervasive Computing (ICPC), IEEE, India, March, 2015. pp. 1-4.
[20] Hu, Chengyu, and Pengtao Liu. "Public key encryption with ranked multi-keyword search." In 2013 5th International Conference on Intelligent Networking and Collaborative Systems, Wasington, United States, IEEE, September, 2013. pp. 109-113.
[21] Zittrower, Steven, and Cliff C. Zou. "Encrypted phrase searching in the cloud." In 2012 IEEE Global Communications Conference (GLOBECOM), California, USA, IEEE, December, 2012. pp. 764-770.