Available online at: http://www.ijmtst.com/vol3issue11.html

International Journal for Modern Trends in Science and Technology

ISSN: 2455-3778 :: Volume: 03, Issue No: 11, November 2017

Multi-keyword Ranked Search Scheme and Fuzzy Search over Encrypted Cloud Data

P.Kavya[1] | Dr.H.Venkateswara Reddy[2] | Vivek Kulkarni[3]

[1]PG Scholar, Department of CSE, Vardhaman Engineering College, Hyderabad, Telangana, India.
[2]Associate Professor, Department of CSE, Vardhaman Engineering College, Hyderabad, Telangana, India.
[3]Professor & Head, Department of CSE, Vardhaman Engineering College, Hyderabad, Telangana, India.

To Cite this Article
P.Kavya, Dr.H.Venkateswara Reddy and Vivek Kulkarni, "Multi-keyword Ranked Search Scheme and Fuzzy Search over Encrypted Cloud Data", *International Journal for Modern Trends in Science and Technology*, Vol. 03, Issue 11, November, 2017, pp.-30-35.

ABSTRACT

With the extended rate of improvement and change of distributed computing each day, more delicate information is being united onto the cloud. For the security of productive restrictive information, the data must be encoded before outsourcing. There is no resilience for grammatical errors and configuration irregularities which are typical client conduct. This makes successful information storage and usage an extremely difficult errand, rendering client seeking exceptionally disappointing and wasteful. In this paper, we concentrate on secure capacity utilizing Advanced Encryption Standard (AES) and data recovery by performing fuzzy keyword seek on this encrypted information. We are proposing the usage of a progressed fuzzy catchphrase look system based strategy which restores the coordinating records when clients' seeking inputs precisely coordinate the predefined keywords or the nearest conceivable coordinating documents of comparability keyword semantics, when correct match falls flat. In the proposed arrangement, we abuse alter separation to measure catchphrases comparability and build up a productive method for developing fuzzy keyword sets, which concentrate on lessening the capacity and portrayal overheads.

KEYWORDS: *Fuzzy search, cloud computing, encryption on cloud, AES*

I. INTRODUCTION

Because of the adaptability and financial reserve funds offered by the cloud server, the clients have been roused to outsource the administration of their information to the cloud. In light of security concerns, information proprietors encode touchy information before outsourcing, which thusly makes information usage a testing issue. Along these lines, improvement of an effective Secure and Dynamic Multi-keyword Ranked Search Scheme and Fuzzy Search over encrypted cloud information is of awesome significance. The most widely recognized hunt strategies recover records utilizing catchphrases as opposed to recovering all the encrypted documents back. To safely seeking over encoded information, the information proprietor as a rule manufactures an encrypted record structure utilizing the extricated catchphrases from the information documents and a comparing list based keyword coordinating calculation and therefore outsources both the encrypted information and this built file structure to the cloud. While looking through the records, the cloud server coordinates the trapdoors of the keywords with the file data and after that profits the relating documents to the information clients. In addition, the information proprietor can impart their information to countless which requires the

cloud server to be able to meet a lot of solicitations with successful information recovery administrations. One viable technique for taking care of this issue we proposed Advanced Encryption Standard (AES) and data recovery by performing fuzzy catchphrase look. This paper proposes a protected tree-based hunt plot over the encrypted cloud information, which bolsters multikeyword positioned inquiry and dynamic operation on the report accumulation. In particular, the vector space display and the generally utilized "term recurrence (TF) × converse archive recurrence (IDF)" show are joined in the list development and question era to give multikeyword positioned seek. Keeping in mind the end goal to acquire high hunt effectiveness, we build a tree-based file structure and propose a "Ravenous Depth-first Search" calculation in light of this file tree. Because of the uncommon structure of our tree-based list, the proposed look plan can adaptably accomplish sub-direct pursuit time and manage the erasure and inclusion of reports. The safe k-NN calculation is used to encode the file and question vectors, and in the mean time guarantee precise pertinence score count between encrypted file and inquiry vectors. In this manner, we concentrate on empowering powerful proficient Secure and Dynamic Multi-keyword Ranked Search for data put away in cloud situations. Fuzzy catchphrase increases framework ease of use by restoring the coordinating records when client looking information sources precisely the predefined keywords or the nearest conceivable coordinating documents in light of keywords closeness semantics, when correct match comes up short. Alter separate is utilized to measure catchphrase likeness and for the improvement of a novel strategy i.e. a special case based strategy, for developing fuzzy keyword sets. This strategy wipes out the requirement for tallying all the fuzzy catchphrases and the aggregate size of the fuzzy keywords sets is essentially diminishes.

II. Related Work

Tune et al. proposed the principal symmetric accessible encryption (SSE) plot, and the inquiry time of their plan is direct to the extent of the information accumulation. Goh proposed formal security definitions for SSE and planned a plan in view of Bloom channel. The hunt time of Goh's plan is O (n), where n is the cardinality of the archive gathering. Curtmola et al. Proposed two plans (SSE-1 and SSE-2) which accomplish the ideal inquiry time. Their SSE-1 conspire is secure against picked catchphrase assaults (CKA1) and SSE-2 is secure against versatile picked keyword assaults (CKA2). These early works are single keyword Boolean hunt plans, which are extremely basic regarding usefulness. A while later, rich works have been proposed under various danger models to accomplish different inquiry usefulness. Cao et al. understood the principal security safeguarding multi-keyword positioned look conspire, in which reports and inquiries are spoken to as vectors of word reference estimate. With the "arrange coordinating", the archives are positioned by the quantity of coordinated inquiry keywords. Be that as it may, Cao et al's. plot does not consider the significance of the diverse keywords, and in this way is not sufficiently precise. Furthermore, the inquiry proficiency of the plan is direct with the cardinality of report gathering. Kamara et al. proposed another pursuit plot in light of tree-based file, which can deal with dynamic refresh on report information put away in leaf hubs. Be that as it may, their plan is outlined just for single catchphrase Boolean inquiry. Money et al. introduced an information structure for keyword/character tuple named "TSet". At that point, a record can be spoken to by a progression of free T-Sets. In view of this structure, Cash et al. Proposed a dynamic accessible encryption conspire. In their development, recently included tuples are put away in another database in the cloud, and erased tuples are recorded in a disavowal list. The last query output is accomplished through barring tuples in the repudiation list from the ones recovered from unique and recently included tuples. However, Cash et al's. dynamic pursuit conspire doesn't understand the multi-catchphrase positioned look usefulness. Zhang et al. proposed a plan to manage secure multi-keyword positioned look in a multi-proprietor display. In this plan, distinctive information proprietors utilize diverse mystery keys to encode their archives and keywords while approved information clients can inquiry without knowing keys of these distinctive information proprietors. The creators proposed an "Added substance Order Preserving Function" to recover the most applicable indexed lists. Be that as it may, these works don't bolster dynamic operations.

III. Issue Formulation

In this paper, we consider a cloud information framework comprising of cloud server, information proprietor and information client. With the commonness of cloud administrations, more

touchy data are being incorporated into the cloud servers, for example, messages, individual wellbeing records, private recordings and photographs, organization fund information, government reports, and so forth. To ensure information protection and battle spontaneous gets to, touchy information must be encoded before outsourcing in order to give end-to-end information privacy confirmation in the cloud and past. Be that as it may, information encryption makes viable information use an exceptionally difficult undertaking given that there could be a lot of outsourced information documents. Plus, in Cloud Computing, information proprietors may share their outsourced information with countless, who may need to just recover certain particular information documents they are occupied with amid a given session. A standout amongst the most mainstream approaches to do as such is through catchphrase based pursuit. Such keyword look strategy enables clients to specifically recover documents of intrigue and has been broadly connected in plaintext seek situations. Tragically, information encryption, which confines client's capacity to perform catchphrase look and further requests the assurance of keyword security, makes the customary plaintext scan techniques come up short for encrypted cloud information.

Disadvantages:

1. For each hunt ask for, clients without pre-learning of the encrypted cloud information need to experience each recovered document keeping in mind the end goal to discover ones most coordinating their advantage, which requests potentially expansive measure of post handling overhead.
2. Constantly sending back all documents exclusively in light of quality/nonattendance of the catchphrase additionally causes expansive superfluous system activity, which is completely undesirable in the present pay-as-you-utilize cloud worldview.

IV. Proposed Work

This paper proposes a safe tree-based pursuit conspire over the encrypted cloud information, which bolsters multi catchphrase positioned inquiry and dynamic operation on the report accumulation. In particular, the vector space display and the broadly utilized "term recurrence (TF) × backwards report recurrence (IDF)" demonstrate are consolidated in the record development and question era to give multi catchphrase positioned look. Keeping in mind the end goal to acquire high inquiry effectiveness, we build a tree-based list structure and propose a "Voracious Depth-first Search" calculation in light of this list tree and to secure encoded and decoded information give an AES calculation. Because of the exceptional structure of our tree-based list, the proposed look plan can adaptably accomplish sub-straight pursuit time and manage the erasure and inclusion of archives. The safe k-NN calculation is used to encode the list and question vectors, and in the interim guarantee exact importance score count between encrypted record and inquiry vectors.

k-NN Algorithm: The preparation illustrations are vectors in a multidimensional element space, each with a class mark. The preparation period of the calculation comprises just of putting away the component vectors and class names of the preparation tests. In the grouping stage, k is a client characterized consistent, and an unlabeled vector (an inquiry or test point) is ordered by doling out the mark which is most incessant among the k preparing tests closest to that question point. A normally utilized separation metric for nonstop factors is Euclidean separation. For discrete factors, for example, for content order, another metric can be utilized, for example, the cover metric (or Hamming separation). With regards to quality articulation microarray information, for instance, k-NN has additionally been utilized with connection coefficients, for example, Pearson and Spearman Often, the arrangement precision of k-NN can be enhanced essentially if the separation metric is found out with particular calculations, for example, Large Margin Nearest Neighbor or Neighborhood segments examination.

AES Encryption: AES involves three square figures: AES-128, AES-192 and AES-256. Each figure encodes and unscrambles information in squares of 128 bits utilizing cryptographic keys of 128-, 192-and 256-bits, separately. Symmetric (otherwise called mystery key) figures utilize a similar key for encoding and decoding, so the sender and the beneficiary must both know - and utilize - a similar mystery key. Every single key length are considered adequate to secure grouped data up to the "Mystery" level with "Top Secret" data requiring either 192-or 256-piece key lengths. There are 10 rounds for 128-piece keys, 12 rounds for 192-piece keys and 14 rounds for 256-piece keys - a round comprises of a few preparing steps that incorporate substitution, transposition and blending of the information plaintext and change it

into the last yield of figure content. The AES encryption calculation characterizes various changes that are to be performed on information put away in an exhibit. The initial step of the figure is to put the information into a cluster; after which the figure changes are rehashed over various encryption rounds. The quantity of rounds is controlled by the key length, with 10 rounds for 128-piece keys, 12 rounds for 192-piece keys and 14 rounds for 256-piece keys. The main change in the AES encryption figure is substitution of information utilizing a substitution table; the second change shifts information pushes, the third blends segments. The last change is a basic elite or (XOR) operation performed on every section utilizing an alternate piece of the encryption key - longer keys require more adjusts to finish.

Architecture of Secure Search Scheme

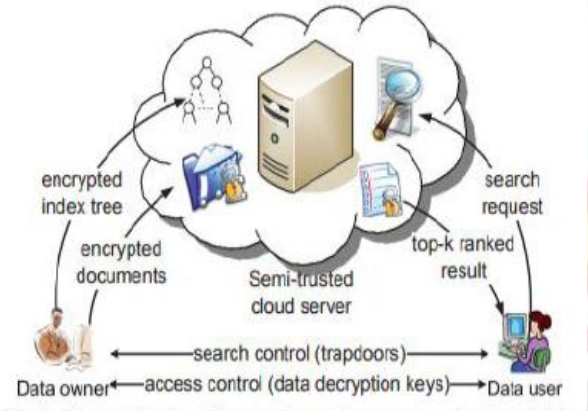


Fig. 1. The architecture of ranked search over encrypted cloud data

Modules:

1. Data Owner module
2. Data User module
3. Semi-Trusted Cloud Server module

Module Description:

Data Owner: The data owner is responsible for the update operation of his documents stored in the cloud server. While updating, the data owner generates the update information locally and sends it to the server.

Data User: Data users are authorized ones to access the documents of data owner. He fetches encrypted documents from cloud server, and then he can decrypt the documents with the shared secret key.

Semi-Trusted Cloud Server: Cloud server stores the encrypted document collection and the encrypted searchable tree index for data owner.

Architectural Design

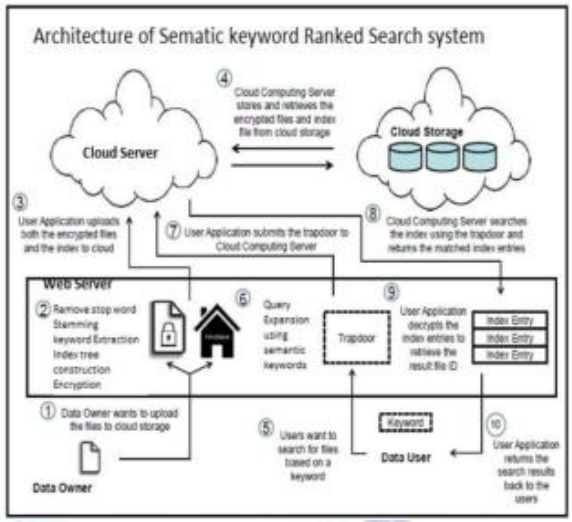


Figure 2: Proposed Architectural Design

Mathematical Model

We consider the set theory to deal with the system S System

S = {s, e, X, Y,Fs ,SS,FS| Φs }

Notation

s= Start state.

Preprocessing of entire document collection. DC- Set of n original text files,

DC={d| d1,d2,d3….dn) fdp={fl ,fs ,fst}

fdp is a document collection preprocessing function which consist of various functions, lexical analysis (f l), stop word removal (fs) and stemming (fst)

fdp={DC}→ {T}

{T} is the list of terms or keywords generated after preprocessing. The weight of each term is calculated using following equation.

Wieght=tf*idf

Where tf is a term frequency of term and idf is the inverse document frequency of term W- Set of m keyword dictionary, W= {w| w1, w2, w3…,wm} after preprocessing

I - Searchable Keyword balanced binary(KBB) Index tree built form all files DC.

I - Encrypted index tree generated from I.

C- Set of n chipper text files after encryption,

C={c| c1, c2, …,cn)

X = Input of the system

Here X is the search request or query entered by the user.

X=Q;

k- No. of expected relevant documents. For the query same document preprocessing functions are applied.

Y = Output of the system

Y ⊆ DC={ d1, d2,…, dk}

SS= Success state

Set of relevant documents such that

Relvd1,Q>= , Relvd1,Q>= , >=, Relvdk,Q

FS= Failure state

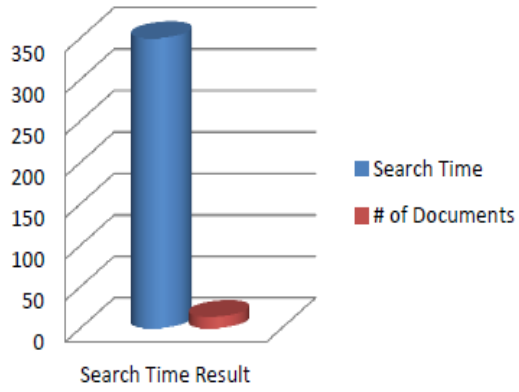
Failure state occurs when there are no documents presents on cloud storage for the search request or query.

Fs= Functions of the system

={data_Preproecssing(), generate_Searchable_Index_Tree(),encryption(), upload_doc_indextree_to_cloudserver(), genQuery(),search()}

V. Performance Analysis

Amid the hunt procedure, if the pertinence score at hub u is bigger than the base significance score in result list RList, the cloud server looks at the offspring of the hub; else it returns. Along these lines, bunches of hubs are not gotten to amid a genuine pursuit. We mean the quantity of leaf hubs that contain at least one catchphrase in the inquiry as by and large is bigger than the quantity of required archives k, however far not as much as the cardinality of the record gathering n. As an adjusted double tree, the stature of the record is kept up to be log n, and the intricacy of importance score count is O(m). In this way, the time many-sided quality of pursuit is O (_mlog n). Note that the genuine pursuit time is not exactly _mlog n. It is on the grounds that 1) many leaf hubs that contain the questioned catchphrases are not gone to as per our inquiry calculation, and 2) the getting to ways of some unique leaf hubs share the common navigated parts. Likewise, the parallel execution of inquiry process can build the effectiveness a great deal



We test the hunt effectiveness of the proposed conspire on a server which underpins 24 parallel strings. The hunt execution is tried individually by beginning 1, 4, 8 and 16 strings. We analyze the pursuit proficiency of our plan with that of Sun et al. In the execution of Sun's code, we isolate 4000 keywords into 50 levels. In this way, each level contains 80 keywords. The more elevated amount the inquiry catchphrases live, the higher the hunt proficiency is In our analysis, we pick keywords from the client transfers the documents alongside the comparing set of catchphrases that are utilized later for perform fuzzy catchphrase look, with the keyword score of k-NN which is closest component and afterward client needs to list every one of the records that has discovered the catchphrases and can be download from the cloud.

VI. Conclusion

In this paper, a protected, proficient and dynamic pursuit plot is proposed, which bolsters the exact multikeyword positioned look as well as the dynamic erasure and inclusion of archives. We plan a propelled look component for developing storage productive fuzzy keyword sets in view of the similitude metric.

References

[1] C. Wang, K. Ren, S. Yu, and K. M. R. Urs, "Achieving usable and privacy-assured similarity search over outsourced cloud data," in Proc. IEEE INFOCOM, 2012, pp. 451–459.

[2] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud," in Proc. IEEE INFOCOM, 2014, pp. 2112–2120.

[3] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in Proc. Appl. Cryptography Netw. Secur., 2004, pp. 31–45.

[4] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in Proc. 1st Int. Conf. Pairing-Based Cryptography, 2007, pp. 2–22.

[5] L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," in Proc. 7th Int. Conf. Inf. Commun. Secur., 2005, pp. 414–426.

[6] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in Proc. 4th Conf. Theory Cryptography, 2007, pp. 535–554.

[7] B. Zhang and F. Zhang, "An efficient public key encryption with conjunctive-subset keywords search," J. Netw. Comput. Appl., vol. 34, no. 1, pp. 262–267, 2011.

[8] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in Proc. Adv. Cryptol., 2008, pp. 146–162.

[9] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on. IEEE, 2000, pp. 44– 55.

[10] E.-J. Goh et al., "Secure indexes." IACR Cryptology ePrint Archive,vol. 2003, p. 216, 2003.

[11] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions,"in Proceedings of the 13th ACM conference on Computer and communications security. ACM, 2006, pp. 79–88

[12] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," in IEEE INFOCOM, April 2011, pp. 829–837.

[13] S. Kamara and C. Papamanthou, "Parallel and dynamic searchable symmetric encryption," in Financial Cryptography and Data Security. Springer, 2013, pp. 258–274.

[14] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Ros¸u, and M. Steiner, "Highly-scalable searchable symmetric encryption with support for boolean queries," in Advances in Cryptology–CRYPTO 2013. Springer, 2013, pp. 353–373.

[15] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, "Dynamic searchable encryption in very large databases: Data structures and implementation," in Proc. of NDSS, vol. 14, 2014.

[16] W. Zhang, S. Xiao, Y. Lin, T. Zhou, and S. Zhou, "Secure ranked multi-keyword search for multiple data owners in cloud computing,"in Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on. IEEE, 2014, pp. 276–286.