Privacy-Preserving Using Multi- Keyword Search over Encrypted Big Data Storage

[1]R.Latha, [2] V.Tamilarasi
[1]Assistant Professor, [2] PG Scholar, Department of MCA
[1]Latha@velhightech.com , [2]chelsitamil0@gmail.com
Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College.
Avadi, Chennai-62

Abstract **- Security is a prime concern for any service that provides big data storage. The data of an individual should remain confidential and should be accessed only by any authenticated person. The required features are obtained by introducing a new technique for providing big data storage i.e. a privacy-preserving multi keyword search over encrypted big data storage. In this technique the latest proposal of "Coordinate matching" i.e., "as matching keys as prospective", is a well-organized similarity evaluate among such multi-keyword semantics to refine the consequence relevance, and has been generally worn in the plaintext Information recovery (IR) community. However, how to be appropriate it in the encrypted cloud data hunt system remains a very difficult task because of inherent protection and privacy obstacles, including various strict necessities like the information privacy, the index privacy, the keyword privacy with multi storage and multi sharing mechanisms.**

Index terms: **privacy, multi-keyword, co-ordinate matching, big data.**

I.INTRODUCTION

Cloud computing is a new technology that is changing the way IT hardware and software are designed and purchase [1]. As a latest model of computing, cloud computing provides profuse benefits including easy access, decreased costs, quick deployment and flexible resource management, etc. Enterprises of all sizes can control the cloud to increase originality and teamwork.

Despite the profuse benefits of cloud computing, for privacy concerns, individuals and enterprise users are averse to delegate their perceptive data, including emails, personal health records and government

Confidential files, to the cloud. This is because once sensitive data are subcontract to a remote cloud; the analogous data owners drop direct control of these information [2]. Cloud service providers (CSPs) would promise to guarantee owners' data security using mechanisms like pragmatic and firewalls. However, these mechanisms do not protect owners' data privacy from the CSP itself, since the CSP possesses full manage of cloud hardware, software, and owners' information. Encryption on perceptive data before outsourcing can preserve data privacy against CSP. However, data encryption makes the traditional information operation service based on plain text keyword search a very challenging problem. A trivial solution to this difficulty is to download all the encrypted information and decrypt them locally. However, this method is observably impractical because it will cause a enormous amount of communication overhead. Therefore, developing a secure search service over encrypted cloud data is of principal importance.

Secure search over encrypted data has newly attracted the interest of many researchers. Song et al. [3] first define and solve the difficulty of secure search over encrypted data. They propose the conception of searchable encryption, which is a cryptographic archaic that enables users to perform a keyword-based search on an encrypted dataset, just as on a plaintext dataset. Searchable encryption is moreover developed by [6], [7], [8]. However, these scheme are concerned mostly with particular or Boolean keyword Search. Extend these techniques for ranked multi-keyword search over encrypted with cloud data will incur serious computation and storage costs. Secure search over encrypted cloud information is first defined by Wang et al. [9] and more over developed by [10], [11], [12], [13]. These research not only diminish the computation and storage cost for secure keyword search over encrypted cloud information, but also enrich the category of search task, including secure ranked multi-keyword search, fuzzy keyword search, and comparison search. However, all these schemes are limited to the particular-owner model. As a material of fact, most cloud servers in practice do not just serve one data owner; instead, they often support multiple data owners to share the remuneration brought by cloud computing.

"Coordinate matching" [4], i.e., "as matching keys as prospective", is an well-organized similarity measure among such multi-keyword semantics to

refine the consequence relevance, and has been generally used in the plaintext Information retrieval (IR) community. However, how to be appropriate it in the encrypted cloud data hunt system remains a very challenging task because of inherent protection and privacy obstacles, including various strict necessities like the information privacy, the index privacy, the keyword privacy with multi storage and multi sharing mechanisms.

The crisis of multi-keyword ranked search over encrypted cloud data (MRSE) while conserve strict system-wise privacy in the cloud computing model. Among a variety of multi-keyword semantics, the proficient similarity measure to confine the consequence of data documents to the search query. Specifically, we use "inner produce similarity" [4], i.e., the number of inquiry keywords appearing in a document, to quantitatively estimate such connection measure of that document to the search query. During the key construction, each document is connected with a binary vector as a sub key where all bit represents whether analogous keyword is contained in the file. The search inquiry is also described as a dual vector where each bit means whether related keyword appears in this search request, so the comparison could be exactly measured by the inner product of the inquiry vector with the data vector. However, frankly outsourcing the information vector or the query vector will violate the key privacy or the hunt privacy. To meet the confront of sustaining such multi-keyword semantic without privacy breaches, a basic initiative for the MRSE using secure inner product computation, which is adapted from a secure k-nearest neighbor (KNN) technique [4], and then give two significantly superior MRSE schemes in a bit by bit manner to achieve various stringent privacy necessities in two peril models with increased attack capabilities. Our contributions are summarized as follows,

1) The difficulty of multi-keyword ranked search over encrypted cloud data, and launch a set of strict privacy requirements for such a secure cloud data utilization system.
2) The MRSE schemes based on the connection appraise of "coordinate matching" while conference different privacy requirements.
3) An proficient data user endorsement protocol, which not only prevents attackers from eavesdrop secret keys and pretending to be dishonest data users performing searches, but also enables data user verification and revocation.

.

The residue of this paper is planned as follows. In Section II, we introduce the system model, the peril model, our design goals, and the beginning Section III describes

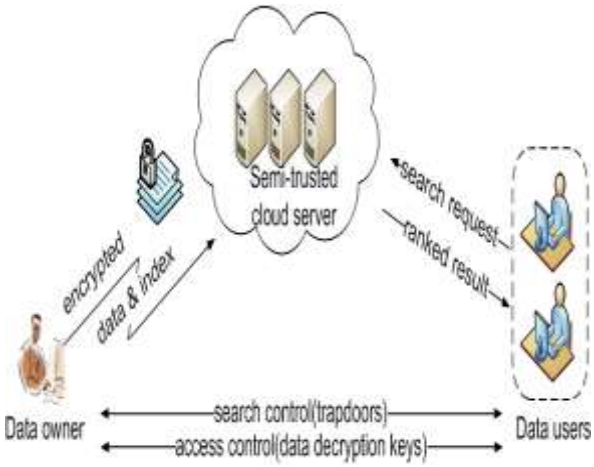

Fig. 1: Architecture of the search over encrypted cloud data

The MRSE framework and privacy necessities followed by section IV, which describes the planned schemes. Section v presents model results. We discuss linked work on both particular and Boolean keyword searchable encryption in Section VI, and conclude the paper in Section VII.

II. PROBLEM FORMULATION

A. STRUCTURE MODEL

A cloud information user service involving three different entities, as illustrated in Fig. 1: the data owner, the information user, and the cloud server. The data owner has a collection of data documents F to be outsourced to the cloud server in the encrypted form $\mathcal{C}$. To enable the searching capability over C for effective data utilization, the data owner, earlier than outsourcing, will first build an encrypted searchable index I from F, and then outsource both the index I and the encrypted file collection $\mathcal{C}$ to the cloud server. To search the document collection for t given keywords, an allowed user acquires a corresponding trapdoor T through search control mechanisms, e.g., broadcast encryption [8]. Upon receiving T from a data user, the cloud server is responsible to search the index I and return the corresponding set of encrypted file. To improve the file retrieval accuracy, the search result should be ranked by the cloud server according to some grade criteria (e.g., coordinate matching, as will be introduced shortly). Moreover, to reduce the communication cost, the data user may send an optional number k along with the trapdoor T so that the cloud server only sends back top-k documents that are most relevant to the search query. Finally, the

access control mechanism [16] is employed to manage decryption capabilities given to users.

B.PERIL MODEL

The big data server is measured as "honest-but-curious" in our model, which is consistent with linked works on cloud security [16], [17]. particularly, the cloud server acts in an "honest" fashion and correctly follows the selected protocol requirement .However, it is "curious" to infer and scrutinize data (including index) in its storage and message flows established during the protocol so as to find out additional information. Based on what information the cloud server knows, we consider two peril models with different attack capabilities as follows.

Known Cipher text Model In this model, the big data server is hypothetical to only know encrypted dataset $\mathcal{C}$ and searchable key *I*, both of which are outsourced from the data owner.

Known setting Model In this stronger model, the cloud server is supposed to possess more data than what can be access in the known cipher text model. Such records may include the correspondence relationship of given search requests (trapdoors), as well as the dataset related algebraic information. As an instance of likely attacks in this case, the cloud server could use the known trapdoor information collective with document/keyword occurrence [18] to assume/identify certain keywords in the query.

C. DESIGN GOAL

To allow ranked search for effective consumption of outsourced cloud data under the aforementioned model, our system design should simultaneously get security and performance guarantees as follows.

Multi-keyword Ranked Search: To intend search schemes which allow multi-keyword inquiry and provide Result similarity ranking for valuable data retrieval, instead of persistent undifferentiated results.

Privacy-Preserving: To prevent the cloud server from erudition further information from the dataset and the key, and to meet privacy necessities specified in section III-B.

Efficiency: Above goal on functionality and privacy should be achieve with low communiqué and calculation overhead

Data owner scalability: The planned scheme should allow latest data owners to enter this structure without disturbing other data owners or information users, i.e., the scheme should maintain data owner scalability in a plug-and-play model.

Data user revocation: The planned scheme should guarantee that only legitimate data users can perform accurate searches. Moreover, once a information user is revoked, he can no longer perform accurate searches over the encrypted cloud data.

D. NOTATIONS

F – the plaintext document collection, denoted as a set of m data documents $\mathcal{F} = (F1, F2, \ldots, Fm)$.
· $\mathcal{C}$ – the encrypted document collection stored in the cloud server, denoted as $\mathcal{C} = (C1, C2, \ldots Cm)$.
· $\mathcal{W}$ – the dictionary, i.e., the keyword set consisting of n
keyword, denoted as $\mathcal{W} = (W1,2, \ldots, Wn)$.
· I – the searchable index associated with $\mathcal{C}$, denoted as $(I1, I2, \ldots, Im)$ where each sub index I is built for Fi.
· $\tilde{}\mathcal{W}$ – the subset of $\mathcal{W}$, representing the keywords in a search request, denoted as $\tilde{}\mathcal{W} = (Wj1,2, \ldots, Wjt)$.
· $T\ \tilde{}\ \mathcal{W}$ – the trapdoor for the search request $\tilde{}\mathcal{W}$.
$F\ \mathcal{W}$ – the ranked id list of all documents according to their relevance to $\tilde{}\mathcal{W}$.

E. COORDINATE MATCHING

"Coordinate matching" [4], i.e., "as matching keys as prospective", is a well-organized similarity community and measure which uses the number of query keywords appearing in the file to quantify the relevance of that document to the query. When users know the accurate subset of the dataset to be retrieve, Boolean queries perform well with the precise search requirement individual by the user. In cloud computing, however, this is not the convenient case, given the huge amount of outsourced data. Therefore, it is more supple for users to specify a list of indicating their interest and retrieve keywords the most relevant documents with a rank order.

III.PRELIMINARIES

Before we introduce our detailed construction, we first briefly introduce some techniques that will be used in this paper.

A. BILLINEAR MAP

Let G and G1 denote two cyclic groups with a prime order p. We further denote g and g1 as the generator of G and G1, respectively. Let ˆe be a bilinear map ˆe : G×G → G1, then the following three conditions are satisfied: 1) Bilinear: ∀a, b ∈ Z* p, ˆe(ga, gb) = ˆe(g, g)ab. 2) Non-degenerate: ˆe(g, g)≠ 1. 3) Computable: ˆe can beefficiently computed.

B. BILINEAR DIFFIE-HELLMAN PROBLEM AND BILINEAR DIFFIE-HELLMAN ASSUMPTION

The Bilinear Diffie-Hellman (BDH) problem in (G,G1, ˆe) is described as follows, given random g ∈ G, and ga, gb, gc for some a, b, c ∈ Z*p, compute ˆe(g, g)abc ∈ G1. The BDH assumption is presented as follows, given (G,G1, ˆe), g ∈ G, and ga, gb, gc for some a, b, c ∈ Z*p, an adversary A has advantage ϵ in solving BDH when Pr[A(ga, gb, gc) = ˆe(g, g)abc] ≥ ϵ. The BDH assumption tells that the benefit ϵ is negligible for any polynomial time A.

IV. MATCHING DIFFERENT-KEY ENCRYPTED KEYWORDS

Various data owners are often concerned in practical cloud applications. For privacy concerns, they would be averse to split secret keys with others. Instead, they prefer to use their own secret keys to encrypt their responsive data (keywords, files). When keywords of dissimilar data owners are encrypted with different secret keys, the coming question is how to find different-key encrypted keywords among multiple data owners.[5] In this section, to enable secure, efficient and fitting searches over encrypted cloud data owned by multiple data owners, we systematically design schemes to achieve the following three requirements: First, different data owners use different secret keys to encrypt their keywords. Second, authentic data users can make their trapdoors without knowing these secret keys. Third, upon receiving trapdoors, the cloud server can find the corresponding keywords from different data owners' encrypted keywords without significant the real value of keywords or trapdoors.

A. KEYWORD ENCRYPTION

For keyword encryption, the following condition should be content: first, different data owners use their have secret keys to encrypt keywords. Second, for the similar keyword, it would be encrypted to different cipher-texts each time. These properties benefit our scheme for two reasons. First, losing the key of one data owner would not lead to the disclosure of next owners' data. Second, the cloud server cannot see any relationship among encrypted keywords. Given the *h*th keyword of data owner *Oi*, i.e., *wi; h*, we encrypt *wi;h* as follows.

$$wi;h = (gki;w{\cdot}ro{\cdot}H(wi;h), gki;w{\cdot}ro) \quad (1)$$

Where *ro* is by chance generated number each time, which helps enhance the security of ˆ *wi;h*. For easy description and understanding, we let

$$E'a = gki;w{\cdot}ro{\cdot}H(wi;h) \text{ and } Eo = gki;w{\cdot}ro$$

The data owner delivers *Ea′* and *Eo* to the administration server, and the management server further re-encrypts *Ea′* with his secret keys *ka*1 and *ka*2 and gets *Ea*.

$$Ea = (Ea' \cdot gka1)ka2 \quad (2)$$

Therefore ˆ *wi;h* = (*Ea,Eo*). The administrative server further submits ˆ *wi;h* to the cloud server. Note that, since the management server only does simple computations on the encrypted data, he cannot learn any sensitive information from these chance encrypted data without knowing the secret keys of data owners.

B.TRAPDOOR GENERATION

To make the data users make trapdoors securely, conveniently and efficiently, our proposed scheme should satisfy two main conditions. First, the data user does not need to request a large amount of data owners for secret keys to make trapdoors. Second, for the same keyword, the trapdoor make each time should be different. To meet this condition, the trapdoor generation is conducted in two steps: First, the data user make trapdoors based on his search keyword and a random number. Second, the management server re-encrypts the trapdoors for the authenticated data user.

Assume a data user wants to search keyword *wh′* , so he encrypts it as follows:

$$T'wh' = (Gh(wh'){\cdot}ru, gru) \quad (3)$$

Where *ru* is a by chance generated y number each time. As we can see, during the trapdoor generation process, secret keys of data owners are not required. Additionally, with the help of chance variable *ru*, for the same keyword *wh′* , we can generate two different trapdoors which prevent attackers from knowing the relationship among trapdoors.

Upon receiving *T′ wh′* , the administration server first generates a random number *ra*, and then re-encrypts *T′ wh′* as follows:

$$Twh' = (gH(wh'){\cdot}ru{\cdot}ka1{\cdot}ka2{\cdot}ra \ , \ gru{\cdot}ka1 gru{\cdot}ka1{\cdot}ra) \quad (4)$$

For easy description and understanding, we let $T1 = gH(wh')\cdot ru\cdot ka1\cdot ka2\cdot ra$,$gru\cdot ka1$, $T3= gru\cdot ka1\cdot ra$, hence, $Twh' = (T1, T2, T3)$. Finally, the administration server submits $Twh='$ to the cloud server.

C. KEYWORDS MATCHING AMONG DIFFERENT DATA OWNERS

The cloud server stores all encrypted files and keywords of different data owners. The administration server will also store a secret data $Sa = gka1\cdot ka2\cdot ra$ on the cloud server. Upon receiving a query request, the cloud will search over the data of all these data owners. The cloud processes the search request in two steps. First, the cloud matches the queried keywords from all keywords stored on it, and it gets a candidate file set. Second, the cloud ranks files in the candidate file set and finds the most top-k relevant files. We introduce the matching strategy here, while leaving the task of introducing the ranking strategy in the next section. When the cloud obtains the trapdoor
Twh' and encrypted keywords (Eo,Ea), he first computes

$$\hat{e}(Sa, T2) = \hat{e}(gra\cdot ka1\cdot ka2 , gru\cdot ka1) = \hat{e}(g, g)ra\cdot ka1\cdot ka2\cdot ru\cdot ka1 \quad (5)$$

Then he can judge whether $wh' = wi;h$ (i.e., an encrypted keyword is located) holds if the following equation is true.

$$\hat{e}(Ea, T3) = \hat{e}((gki;w\cdot ro\cdot H(wi;h)\cdot gka1)ka2,gru\cdot ka1\cdot ra) = \hat{e}(g,g)(ki;w\cdot ro\cdot H(wi;h)+ka1)\cdot ka2\cdot ru\cdot ka1\cdot ra \quad (6)$$

$$= \hat{e}(g, g)ki;w\cdot ro\cdot H(wi;h)\cdot ka2\cdot ru\cdot ka1\cdot ra \cdot \hat{e}(Sa, T2) = \hat{e}(gki;w\cdot ro , gH(wi;h)\cdot ka2\cdot ru\cdot ka1\cdot ra) \cdot \hat{e}(Sa, T2) = \hat{e}(Eo, T1) \cdot \hat{e}(Sa, T2)$$

V.RELATED WORK

In primary phase, Authentication: - This is password based or key authentication.1. Cloud users ask for login page. 2. The cloud provider displays the login screen. 3. Cloud user login with username and password. 4. Cloud provider verify is suitable username and password by searching in DB in cloud storage.5.If user information not valid show error message else display second phase of authentication.

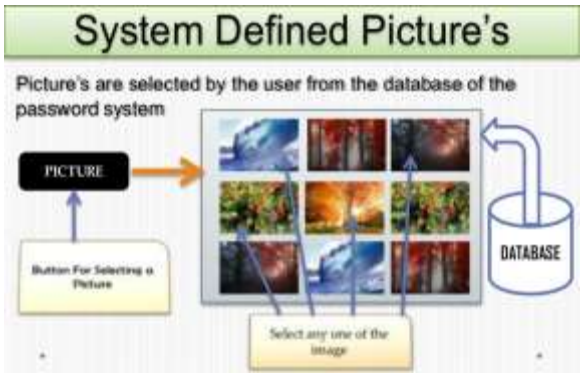

Figure 2

In secondary phase authentication steps: Then user enters the graphical password authentication 1. Cloud provider displays graphical login screen, in which many images showed. 2. The cloud user chooses his password image into the multiple images. 3. A cloud provider check is suitable graphical image by searching in DB in cloud storage. If user image is not valid show error message else display the full image. 4. Then user clicks on the specific place (location) on the image. 5. Cloud provider check is suitable graphical image location password by searching in DB in cloud storage. 6. If user password is suitable you will successfully authenticated with cloud server. Otherwise display error message

VI. CONCLUSION

The complexity of multi-keyword ranked search over encrypted cloud data, and establish a variety of privacy requirements. Among a variety of multi-keyword semantics, we decide the proficient similarity measure of "coordinate matching", i.e., as many matches as possible, to effectively detain the relevance of outsourced credentials to the query keywords, to quantitatively analyze such connection measure. For meeting the confront of supporting multi-keyword semantic without privacy breaches, we knew idea a basic idea of MRSE using secure inner product computation. Then we give two considerably improved MRSE schemes to achieve various stringent privacy requirements in two different peril models. Thorough analysis investigating privacy and efficiency guarantees of new idea schemes is given, and experiments on the real-world dataset show our new idea schemes introduce low overhead on both calculation and communiqué.

REFERENCES

[1] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 50–55, 2009.

[2] S. Kamara and K. Lauter, "Cryptographic cloud storage," in *RLCPS, January 2010, LNCS. Springer, Heidelberg*.

[3] A. Singhal, "Modern information retrieval: A brief overview," *IEEE Data Engineering Bulletin*, vol. 24, no. 4, pp. 35–43, 2001.

[4] I. H. Witten, A. Moffat, and T. C. Bell, "Managing gigabytes: Compressing and indexing documents and images," Morgan Kaufmann Publishing, San Francisco May 1999.

[5] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. of S&P*, 2000.

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

[7] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in *Proc. of ACNS*, 2005.

[8] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable
symmetric encryption: improved definitions and efficient constructions," In *Proc. of ACM CCS*, 2006.

[9] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. of EUROCRYPT*, 2004.

[10] M. Bellare, A. Boldyreva, and A. ONeill, "Deterministic and efficiently searchable encryption," in *Proc. of CRYPTO*, 2007.

[11] M. Abdulla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable encryption
revisited: Consistency properties, relation to anonymous ibe, and extensions," *J. Cryptol.*, vol. 21, no. 3, pp. 350–391, 2008.

[12] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *Proc. of IEEE INFOCOM'10 Mini-Conference*, San Diego, CA, USA, March 2010.

[13] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. S. III, "Public key encryption that allows pir queries," in *Proc. of CRYPTO*, 2007.

[14] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in *Proc. of ACNS*, 2004, pp. 31–45.

[15] L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," in *Proc. of ICICS*, 2005.

[16] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proc. Of INFOCOM*, 2010.

[17] S. Zerr, E. Demidova, D. Olmedilla, W. Nejdl, M. Winslett, and S. Mitra, "Zerber: r-confidential indexing for distributed documents," in *Proc. of*
EDBT, 2008, pp. 287–298.

[18] S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+r: Top-k retrieval from a confidential index," in *Proc. of EDBT*, 2009, pp. 439–449.

