Privacy-Preserving Multi-Keyword Ranked Search

Mr.Haider Kareem Mohammed
Research Scholar, Department of Computer Science, Madurai Kamaraj University, Madurai - 625 021

Abstract- Sensitive data, such as emails, personal health records, photo albums, tax documents, financial transactions, and other types of data, may need to be encrypted by data owners before being outsourced to the commercial public cloud [58]; however, this renders the traditional data utilization service based on plaintext keyword search obsolete. Due to the high cost of bandwidth in cloud scale systems, the simple approach of downloading all the data and decrypting locally is clearly impracticable. Furthermore, aside from eliminating local storage management, storing data in the cloud is useless unless it can be searched and used quickly. Thus, investigating a privacy-preserving and effective search service for encrypted cloud data is critical. Given the possibility for a big number of on-demand data users and a significant number of outsourced data documents in the cloud, this challenge is particularly tough to solve, as it is exceedingly difficult to meet performance, system usability, and scalability criteria.

Key Words : Cloud, Privacy, encrypt, system, performance, storage, management

INTRODUCTION

Cloud computing is the long-awaited realization of computing as a utility, in which cloud users can store their data remotely in the cloud and access high-quality applications and services on demand from a common pool of programmable computer resources [31, 32, 103]. Individuals and businesses are both motivated to outsource their local complicated data management system to the cloud because of its excellent flexibility and cost savings.

On the one hand, the enormous number of documents necessitates the cloud server performing result relevance ranking rather than returning undifferentiated results in order to meet the effective data retrieval need. Instead of searching through every match in the content collection, data consumers can quickly get the most relevant information using a ranked search system [96]. In the "pay-as-you-use" cloud model, ranked search can also significantly save wasteful network traffic by sending back only the most relevant material. Such a ranking system, however, should not leak any keyword-related information for privacy reasons.

However, due of the inherent security and privacy hurdles, such as data privacy, index privacy, keyword privacy, and many other rigorous criteria, implementing it in an encrypted cloud data search engine remains a difficult undertaking

Searchable encryption [7,18,22,23,35,40,51,65,86,98] is a useful technology that considers encrypted data as documents and allows a user to securely search for documents of interest using a single keyword. These approaches, however, would not be acceptable for use in a secure large-scale cloud data usage system because they were designed as crypto primitives and cannot support such high service-level requirements as system usability, user searching experience, and easy information finding.

Although various recent designs [16, 24, 29, 52, 55, 59, 63, 67, 93] have been presented to support Boolean keyword search in an attempt to increase search flexibility, they are still insufficient to provide consumers with appropriate result ranking functionality Our early research [104, 106] were aware of this issue and provided solutions to the secure ranked search over encrypted data problem, but only for single-keyword queries. The question of how to develop an efficient encrypted data search mechanism that enables multi-keyword semantics while avoiding privacy violations is still open.

Direct outsourcing of the data vector or query vector, on the other hand, will compromise index or search privacy. We propose a basic idea for the MRSE using secure inner product computation, which is adapted from a secure k-nearest neighbor (kNN) technique [120], and then give two significantly improved MRSE schemes in a step-by-step manner to achieve various stringent privacy requirements in two threat models with increased attack capabilities. The following is a list of our contributions:

We investigate the topic of multi-keyword ranked search over encrypted cloud data for the first time and propose a set of stringent privacy requirements for such a secure cloud data utilization system.

Based on the similarity measure of "coordinate matching," we present two MRSE techniques that meet distinct privacy requirements in two separate threat models.

We look at improving our ranked search method to accommodate more search semantics and dynamic data operations.

A thorough investigation of the suggested schemes' privacy and efficiency guarantees is provided, and experiments on a real-world dataset indicate that the proposed methods do indeed introduce low cost on computation and communication.

PROBLEM FORMULATION

System Model

As shown in Fig. 3.1, a cloud data hosting service involves three different entities: the data owner, the data user, and the cloud server. The data owner has a set of data documents F that will be encrypted and sent to the cloud server. Before outsourcing, the data owner will first generate an encrypted searchable index I from F, and then outsource both the index I and the encrypted document collection C to the cloud server to provide searching capability over C for effective data usage. An authorized user obtains a matching trapdoor T through search control techniques, such as broadcast encryption [40], to search the document collection for t supplied keywords.

The cloud server is responsible for searching the index I and returning the corresponding collection of encrypted documents after receiving T from a data user. To improve document retrieval accuracy, the cloud server should rank the search results based on specific ranking criteria (e.g., coordinate matching, as will be introduced shortly). Furthermore, the data user may transmit an optional number k along with the trapdoor T to have the cloud server only give back the top-k pages that are most relevant to the search query, reducing transmission costs. Finally, the data collection can be changed in terms of inserting new documents, modifying current documents, and removing existing documents using the access control method [129].

THREAT MODEL

In our concept, the cloud server is "honest-but-curious," which is consistent with related cloud security research [111, 129]. The cloud server, in particular, acts in a "honest" manner and appropriately follows the authorized protocol specification. However, inferring and analyzing data (including index) in its storage and message flows received via the protocol to gain more information is "curious." We investigate two threat models with varying attack possibilities based on the information the cloud server has. Ciphertext Models Known The cloud server in this approach is only expected to know encrypted dataset C and searchable index I, which are both outsourced from the data owner.

Model of the Background The cloud server in this stronger model is expected to know more than what can be accessible in the known ciphertext model. The correlation relationship of given search requests (trapdoors), as well as dataset-related statistical information, are examples of such information. As an example of possible attacks in this circumstance, the cloud server might deduce/identify certain terms in the query using known trapdoor information along with document/keyword frequency [131].

NOTATIONS

F – the plaintext document collection, which is represented by a set of m data documents F = (F1, F2, . . ., Fm).

C – the cloud server's encrypted document collection, denoted by C = $(C_1, C_2, \ldots, C_m)$.

W – the dictionary, i.e., the keyword set consisting of n keyword, denoted as
W = $(W_1, W_2, \ldots, W_n)$.

I – the searchable index for C, denoted as $(I_1, I_2, \ldots, I_m)$ with each subindex I_i generated for F_i.

$\widetilde{W}$= W – the subset of W that represents the keywords in a search request as $\widetilde{W} = (W_{j1}, W_{j2}, \ldots, W_{jt})$.

$T_{\widetilde{W}}$ - a backdoor for the $\widetilde{W}$ search request.

F_{W} – the ranking id list of all papers sorted by relation to $\widetilde{W}$

Coordinate Matching Preliminary

"Coordinate matching" [119], a mix of conjunctive and disjunctive search, is an intermediate similarity metric that leverages the amount of query terms appearing in the document to assess the text's relevance to the query. Boolean searches work effectively with the precise search requirement specified by the user when the user knows the exact subset of the dataset to be obtained. Given the vast volume of data that is outsourced in cloud computing, this is not the case. As a result, users have more flexibility in specifying a list of keywords that indicate their interest and retrieving the most relevant pages in a rank order.

MRSE Framework and Privacy Requirements

We define the framework for multi-keyword ranked search over encrypted cloud data (MRSE) in this part, as well as several severe system-wide privacy requirements for such a secure cloud data use system.

MRSE Framework

Because the data owner might easily encrypt and then outsource data using typical symmetric key encryption, operations on the data documents are not included in the framework for ease of presentation. The MRSE system is made up of four algorithms that focus on the index and query.

Setup (1^{ℓ}) the data owner generates a symmetric key as SK using a security parameter ℓ as input.

Create Index (F, SK) the data owner creates a searchable index I based on the dataset F, which is then encrypted with the symmetric key SK and sent to the cloud server. The document collection can be encrypted and outsourced independently after the index is built.

Trapdoor ($\widetilde{\mathrm{W}}$) This algorithm constructs a comparable trapdoor $T_{\widetilde{\mathrm{W}}}$ using t keywords of interest in $\widetilde{\mathrm{W}}$as input.

Query (T_{W} , k, I) When the cloud server receives the question (T_{W} , k), it uses trapdoor T_{W} to do a ranked search on the index I, and then returns F_{W}, the ranked id list of the top-k documents sorted by their resemblance to $\mathrm{W}^{\sim}$.

The search control and the access control are not covered in this dissertation. The former controls how authorized users obtain trapdoors, while the latter controls user access to outsourced materials.

Privacy Requirements for MRSE

In related literature, such as searchable encryption, the representative privacy assurance is that the server should learn nothing except search results. We study and construct a set of severe privacy requirements particularly for the MRSE architecture using this broad privacy description.

In terms of data privacy, the data owner can encrypt the data before outsourcing using typical symmetric key cryptography, effectively preventing the cloud server from looking into the outsourced data. In terms of index privacy, if the cloud server deduces any relationship between keywords and encrypted documents from the index, it can teach a document's principal subject, even the content of a brief document [131]. As a result, the searchable index should be built to prevent the cloud server from committing such an association attack. While data and index privacy guarantees are required by default in the associated literature, the following search privacy criteria are more complex and harder to meet.

Privacy is a key word. Because users prefer not to have their searches revealed to others, such as the cloud server, the most significant problem is to conceal what they are looking for, namely the keywords signaled by the appropriate trapdoor. Although the trapdoor can be created in a cryptographic manner to safeguard the query keywords, the cloud server might perform some statistical analysis on the search results to produce an estimate. Document frequency (i.e., the number of documents containing the keyword) is sufficient as a type of statistical information to identify the keyword with high probability [130]. This keyword-specific information can be used to reverse-engineer the keyword if the cloud server has some background knowledge about the dataset.

Unsinkability of Trapdoors Instead of being deterministic, the trapdoor generating function should be randomized. The cloud server, in particular, should not be able to deduce the relationship between any two trapdoors, such as determining whether the two trapdoors are produced by the same search request. Otherwise, generating a deter- monistic trapdoor would provide the cloud server the benefit of accumulating frequencies of different search requests for different keyword(s), potentially violating the aforementioned keyword privacy criterion. As a result, the foundational safeguard against trapdoor unsinkability is to include enough no determinacy in the trapdoor generation mechanism.

Privacy-Preserving and Efficient MRSE

We suggest using "inner product similarity" [119] to objectively evaluate the efficient similarity measure "coordinate matching" to perform multi-keyword ranked search efficiently. D_i is a binary data vector for document F_i, and Q is a binary query vector indicating the keywords of interest, with each bit $D_i[j] \in \{0, 1\}$ representing the existence of the corresponding keyword W_j in the query $\widetilde{W}$.

The inner product of their binary column vectors, i.e. $Q[j] \in \{0, 1\}$, is used to indicate the similarity score of document F_i to query$\widetilde{W}$. The cloud server must be provided the capacity to compare the similarity of different documents to the query in order to rank them. However, data vector D_i, query vector Q, and their inner product D_i Q should not be accessible to the cloud server in order to maintain tight system-wide privacy.

In this section, we first propose a basic idea for the MRSE based on secure inner product computation, which is adapted from a secure k-nearest neighbor (kNN) technique, and then show how to significantly improve it in the MRSE framework to be privacy-preserving against various threat models in a step-by-step manner.

SECURE INNER PRODUCT COMPUTATION

Secure in Computation

The Euclidean distance between a data record pi and a query vector q is used to identify k nearby database records in the secure k-nearest neighbor (kNN) approach [120]. The secret key is made up of one (d+1)-bit vector S and two (d+1)(d+1) invertible matrices M1, M2. To begin, every data vector pi and query vector $\vec{p}_i$ are extended to (d + 1)-dimension vectors as pi and q, respectively, with the (d + 1)-th dimension set to 0.5||p2|| and 1. Furthermore, the query vector q is scaled by a random number r > 0 as follows: (rq, r). Then, $\vec{p}_i$ is split into two random vectors as $\{\vec{p}_i\,', \vec{p}_i\,''\}$, and $\vec{q}$ is also split into two random vectors as $\{\vec{q}\,', \vec{q}\,''\}$.

Note here that vector S *functions* as a splitting indicator. Namely, if the j-th bit of S *is* 0, $\vec{p}_i\,'[j]$ and $\vec{p}_i\,''[j]$ are set as the same as $\vec{p}_i[j]$, while $\vec{q}\,'[j]$ and $\vec{q}\,''[j]$ are set to two random numbers so that their sum is equal to $\vec{q}[j]$; if the j-th bit of S is 1, the splitting process is similar except that $\vec{p}_i$ and $\vec{q}$ are switched. The split data vector pair $\{\vec{p}_i\,', \vec{p}_i\,''\}$is encrypted as $\{p_{ia}, p_{ib}\}$, where $p_{ia} = M_1^T\vec{p}_i\,'$ and $p_{ib} = M_2^T\vec{p}_i\,''$; the split query vector pair $\{\vec{q}\,', \vec{q}\,''\}$ is encrypted as $\{q_a, q_b\}$, where $q_a = M_1^{-1}\vec{q}\,'$ and $q_b = M_2^{-1}\vec{q}\,''$. In the query step, the product of data vector pair and query vector pair, i.e., $-0.5r(\|p_i\|^2 - 2p_i \cdot q)$, is serving as the indicator of Euclidean distance $(\|p_i\|^2 - 2p_i \cdot q + \|q\|^2)$ to select k nearest neighbors.

Known Ciphertext Model Security Analysis Let the encrypted data record and query vector be the attacker's knowledge, as in [120]. By definition, the attacker knows the encrypted values $\{p_{ia}, p_{ib}\}$.for any data record p_i. The attacker must model as two random (d+1)-dimensional vectors if he does not know the splitting configuration. The transformation matrices are solved using the equations $M^T\vec{p}_i\,' = p_{ia}$ and $M^T\vec{p}_i\,'' = p_{ib}$, where M_1 and M_2 are two $(d+1) \times (d+1)$ Unknown matrices. $2(d+1)$ unknowns in p_{ia} and p_{ib} and $2(d+1)^2$ unknowns In M_1 and M_2.The attacker does not have enough knowledge to solve for the transformation matrices because there are only 2(d + 1) equations, which is less than the number of unknowns. As a result, we believe that in the given cipher text model, this kNN computing approach is secure.

Secure Inner Product Computation

We need to make some changes to the safe kNN computation strategy to fit the MRSE framework because it uses inner product similarity instead of Euclidean distance. One technique to achieve this is to remove the dimension extension; the final result becomes the inner product as $rp_i \cdot q$.

Analysis of Efficiency While the encryption of either a data record or a query vector requires two $O(d^2)$ multiplications of a d× d matrix and a d-dimension vector, the final inner product computation requires two O(d) multiplications of two d-dimension vectors (d).

Analysis of Security Because the splitting vector S is unknown under the known ciphertext model, $\vec{p}_i\,'$ and $\vec{p}_i\,''$are treated as two random d-dimensional vectors. We have 2dm unknowns in m data vectors and $2d^2$ unknowns in $\{M_1, M_2\}$to solve the linear equations formed by data vector encryption. We don't have enough information to solve data vectors or $\{M_1, M_2\}$ because we only have 2dm equations, which are less than the number of unknowns. In the same way, q' and $\vec{q}''$ can be thought of as two random d-dimensional vectors.

MRSE I: Privacy-Preserving Scheme in Known Ci phertext Model

For our MRSE architecture, the customized secure inner product computation approach is insufficient. The main reason is because the only randomness involved in the trapdoor generation is the scale factor r, which does not provide enough nondeterminacy in the overall scheme to meet the trapdoor unlinkability and keyword privacy requirements. We've updated our MRSE I scheme to provide a more advanced design for the MRSE.

MRSE I Scheme

Instead of just eliminating the extended dimension from the query vector as we first planned, we keep this dimension extending procedure but assign a new random number t to the extended dimension in each query vector in our more advanced approach. This new randomization is supposed to make it more difficult for the cloud server to figure out the relationship between the incoming trapdoors.

Furthermore, as specified in the keyword privacy requirement, randomness in the search result should be properly regulated to obscure document frequency and reduce the likelihood of keyword reidentification. Incorporating some randomization into the final similarity score is a good method to achieve what we want. Specifically,

we inject a dummy keyword into each data vector and assign a random value to it, as opposed to the query vector's randomness. Instead of (n + 1), each individual vector Di is extended to (n + 2) dimensions, with a random variable I representing the dummy keyword stored in the extended dimension. The following is the entire technique for achieving ranked search with multiple keywords over encrypted data.

Setup The data owner creates random numbers a $(n + 2)$ bit vector as S and two $(n + 2) \times (n + 2)$ invertible matrices $\{M_1, M_2\}$. The secret key SK is in the form of a 3-tuple as $\{S, M_1, M_2\}$.

BuildIndex(F, SK) A binary data vector is created by the data owner. D_i for every document F_i, where each binary bit $D_i[j]$ whether the corresponding keyword is present W_j appears in the document F_i. Subsequently, every plaintext subindex $\vec{D}_{\mathrm{i}}$ is dimension extending and splitting methods were used to create on D_i. These procedures are similar with those in the secure kNN computation except that the $(n+1)$-th entry in $\vec{D}_{\mathrm{i}}$ is set to a random number ε_i, and the $(n+2)$-th entry in $\vec{D}_{\mathrm{i}}$ is set to 1 during the dimension extending. $\vec{D}_{\mathrm{i}}$ is therefore equal to $(D_i, \varepsilon_i, 1)$. Finally, the subindex $I_i = \{M^T \vec{D}_i'$, $M^T \vec{D}_i''\}$ is built for every encrypted document C_i.

Trapdoor(W) With t keywords of interest in W as input, one binary vector Q is generated where each bit $Q[j]$ indicates whether $W_j \in$ W is true or false. Q is first extended to n + 1-dimension which is set to 1, and then scaled by a random number $r \neq 0$, and finally extended to a $(n + 2)$-dimension vector as $\vec{Q}$ where the last dimension is set to another random number t. $\vec{Q}$ is therefore equal to (rQ, r, t). Following the identical splitting and encrypting procedures as previously, the trapdoor $T\sim$ is generated as $\{M_1^{-1}\vec{Q}', M_2^{-1}\vec{Q}''\}$.

Query(T_W, k, I) With the trapdoor T_W, the cloud server computes the similarity scores of each document F_i as in equation 3.1. WLOG, we assume $r > 0$. After sorting all scores, the cloud server returns the top-k ranked id list $F_{\tilde{W}}$.

The final similarity scores would be: With t in the query vector and ε_i in each data vector, the final similarity scores would be:

The final score in the original situation is just $rD_i \,.\, Q$, which preserves the scale relationship between two queries on the same keywords. Due to the unpredictability of both t and ε_i in our enhanced scheme, this issue is no longer valid, demonstrating the usefulness and improved security strength of our MSRE L process.

Analysis

The three components of design goals outlined in section 3.2 are used to analyze this MRSE L system.

Efficiency and functionality Assume that a document has a certain amount of query terms F_i is $x_i = D_i \cdot Q$. From equation 3.1, the final similarity score as $y_i = I_i \cdot T\tilde{W} = r(x_i + \varepsilon_i) + t$ is a linear function of x_i, where the coefficient r positive random integer is assigned to r.

However, because the random factor ε_i is included in the similarity score, the final search result based on similarity score sorting may not be as accurate as in the original method. We can let ε_i follow a normal distribution N (μ, σ^2) for search accuracy, with the standard deviation serving as a flexible trade-off parameter between search accuracy and security. From the standpoint of efficiency, σ is expected to be smaller in order to achieve high precision, demonstrating strong document purity. We set a measure called precision P_k to capture the fraction of returned top-k documents that are included in the genuine top-k list in order to quantify search accuracy. The accuracy of the real-world dataset will be evaluated in detail in section 3.6.

From a performance standpoint, our MRSE scheme based on internal products is an exceptional solution. The generating technique of each subindex or trapdoor in stages like BuildIndex or Trapdoor comprises two multiplications of a (n + 2) × (n + 2) matrix and a (n + 2)-dimension vector with complexity O. (n2). The final similarity score in the Query is determined by multiplying two (n + 2)-dimension vectors with complexity O. (n).

Privacy Traditional symmetric key encryption techniques could be used to protect data privacy; however this is outside the scope of this research.

Because such vector encryption approach has been shown secure in the known ciphertext model [120], the index privacy is well preserved if the secret key SK is kept hidden. In comparison to the adapted secure inner product computation described in Section 3.4.1.2, we add two more dimensions to the vectors. When it comes to data encryption, In $M_2^T \vec{D}_i'$= I_i' 1 I and $M_2^T \vec{D}_i''$= I_i'', the number of equations as 2(n + 2)m is still smaller than the number of unknowns as the total of 2(n + 2)m 2 I unknowns in m data vectors and 2(n + 2)^2 unknowns in {M_1, M_2}. As a result, the attacker is unable to fix the problems. The inclusion of dimensions will only strengthen the scheme's security [120].

Our basic technique may generate two completely different trapdoors for the same query $\widetilde{W}$ thanks to the randomization introduced by the splitting procedure and the random numbers r and t. Because of the deterministic quality of trapdoor production, this nondeterministic trapdoor generation can guarantee trapdoor unlinkability, which is an unsolved privacy leakage concern in related symmetric key based searchable encryption methods [40]. Furthermore, with the right setting for the random factor ε_i even the final score results can be effectively obfuscated, preventing the cloud server from understanding the links between given trapdoors and keywords.

CONCLUSION

We define and solve the problem of multi-keyword ranked search over encrypted cloud data for the first time in this chapter, as well as provide a variety of privacy requirements. To effectively capture the relevance of outsourced documents to the query keywords, we choose the efficient similarity measure of "coordinate matching," i.e., as many matches as possible, from among various multi-keyword semantics, and use "inner product similarity" to quantitatively evaluate such similarity measure. We present a basic idea of MRSE employing safe inner product computing to tackle the difficulty of enabling multi-keyword semantic without privacy breaches. Then, in two different threat models, we present two revised MRSE strategies to meet distinct demanding privacy criteria.

We also look at improving our ranked search method by adding support for other search semantics, such as TF IDF, and dynamic data operations. A thorough investigation of the suggested schemes' privacy and efficiency guarantees is provided, and experiments on a real-world dataset indicate that our proposed methods have low overhead on both computation and communication.

REFERENCE

[1] Allison Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *Proc. of EUROCRYPT*, 2010.

[2] Cohen. Enron email dataset. http://www.cs.cmu.edu/~enron/. Reza Curtmola, Juan A. Garay, Seny Kamara, and Rafail Ostrovsky. Search- able symmetric encryption: improved definitions and efficient constructions. In *Proc. of ACM CCS*, 2006.

[3] Cong Wang, Qian Wang, Kui Ren, Ning Cao, and Wenjing Lou. Toward secure and dependable storage services in cloud computing. *Services Computing, IEEE Transactions on*, 5(2):220 –232, april-june 2012.

[4] Giuseppe Ateniese, Roberto Di Pietro, Luigi V. Mancini, and Gene Tsudik. Scalable and efficient provable data possession. In *Proceedings of the 4th in- ternational conference on Security and privacy in communication netowrks*, SecureComm ’08, pages 9:1–9:10, New York, NY, USA, 2008. ACM.