INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN : 2454-9924

Efficient Searchable Symmetric Encryption Multi-Keyword Ranked Search over Encrypted Cloud Data

Ranjit.VNV[1], V.Dakshayani[2]

M.Tech Student[1], Assistant professor[2]

Department of Computer Science Engineering[1&2]

MJR college of engineering & Technology, Pileru A.P,India.[1&2]

Abstract:

As cloud storage is widely adopted in various applications, how to protect data privacy while allowing efficient data Searching and retrieving a distributed environment remains a challenging research issue. Already searchable encryption schemes Still not enough on desired functionality and security / privacy perspectives. Specifically, supports multi-keyword search below Hiding the multi-user setting, search pattern and access pattern and preventing keyword essay attacks (KGA) are the most challenging Tasks. In this paper, we present a new searchable encryption scheme that solves the above problems at once, which does Adoption in distribution systems is practical. It does not enable multi-keyword search on encrypted data under Multi-Writer / Multilayer Setting but guarantees data and search pattern privacy. To avoid KGA, our scheme adopts a multi-server architecture, this speed up the search response, shares the workload and reduces the risk of key leakage by allowing only authorized servers Test whether the search token matches the stored cipher. A novel subcommittee decision process is also designed as a major technique Is underlying our scheme and can be used in applications other than keyword search. Finally, we demonstrate and assess safety Ability to compute and communicate to demonstrate the practicality of our scheme.

Keywords: **Searchable Encryption, Multi-Keyword Search, Multi-User Access, Search Pattern, Access Pattern.**

INTRODUCTION:

With the emergence of cloud computing due to its attractive advantages over traditional data storage, cloud storage has become one of the most popular and essential cloud services for industrial and personal users. According to the statistics portal website Statista, the data center storage capacity worldwide will be 2,300 exabytes by 2021 [1]. With such rapid growth in cloud storage, data security and privacy are inevitable considerations that need to be addressed well to prevent monetary loss or reputation damage due to cloud data leaks. Therefore, it is natural to apply cryptographic methods such as data encryption mechanisms to ensure the privacy of sensitive information stored in the cloud. However, such direct privacy protection does not work for cloud storage facilities with significant capabilities as it does not allow the cloud server to perform a quick search on data stored based on user request. To

ISSN : 2454-9924

address this issue, searchable encryption schemes have been introduced in the literature.

In the seminary work of Boneh et al. [2], the concept of public-key encryption was introduced with Keyword Search (PEKS). In the PEKS scheme, it is assumed that there are three entities: the data owner (or author), the data user (or reader) and the storage server. To share data with the user via Jukiao Liu, Gumin Yang, Willie Susilo and Joseph Tonian Institute of Cyber security and Cryptology, University of Wollongong, Australia, 2522. Email: fxl691, Guang, Wsusilo, dong @. au. Jimeng Liu College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China, 350002. Email: snbnix@gmail.com. Xian Shen School of Computer and Software Nanjing University of Information Science and Technology Nanjing, China, 210044. Email: s shenjian@126.com. * Related author. Manuscript received,; Edited,. The storage server, the owner first extracts a keyword from the data and then generates the keyword encryption (known as searchable cipher) with the intended user's public key. The actual data, which can be encrypted separately, is submitted to the server along with its searchable cipher. Only the intended user can then generate a search token (aka Trapdoor) based on his / her private key and keyword of interest and then send the token to the server, testing whether the Trapdoor is compatible with the searchable cipher and informing the user of the search result. In this model, if the owner wants to share the same data with different users, it must repeat the operation above and produce multiple searchable ciphers that are not practical or scalable for distribution environments. For example, if the administrative staff of a large company with tens of thousands of employees uploads a control document to their member cloud server, it will have to generate tens of thousands of searchable ciphers, resulting in massive computational and storage overhead. Therefore, an effective searchable encryption scheme that supports multi-user access is more needed for multi-user environments, where searchable ciphers can be matched with trapdoors from different authorized users.

Similar to the demand of multi-user search, multimode search is another desirable feature of searchable encryption. For a data document with multiple keywords, the plain PEKS scheme requires the production of the same number of searchable ciphers. In addition, given the set of trapdoors for multiple search keywords, each trapdoor needs to be tested repeatedly against all searchable ciphers associated with a document. Therefore, more efficient searchable encryption that supports multi-keyword search is also required.

In secure cloud storage that supports keyword search, security issues involve not only data privacy, but also search query privacy. Unfortunately, the PEKS scheme by Boneh et al. [2] Search query privacy against offline keyword attacks (KGAs) [3] cannot be guaranteed [3] because anyone (including the cloud server) can check whether the trapdoor fits the searchable cipher Therefore, it is important to block searchable encryption schemes from the KGA to protect user privacy.

Various solutions have been proposed to prevent KGA attacks. There are usually two ways to block KGA: make the server itself unable to generate searchable cipher, and then enable KGA; And

ISSN : 2454-9924

discontinuation of public examination. The first method led to new cryptographic primitives, such as public-key authentication encryption with keyword search (PAEKS) [4] where the data owner's private key is used to generate and authenticate searchable ciphers. However, PAEKS data takes the user's public key as input to generate searchable ciphers, thus failing to support multi-user search.

The second approach, which disables public testing, inevitably requires secrecy to be used in the testing algorithm. Since the test is performed by a storage server, if the secret server knows, it can be KGA undetected. Therefore, a distribution testing approach is needed to reduce reliance on a single server. Specifically, the secret used for testing can be divided into two (or multiple) shares, one held by a public cloud storage server and the other by an internal server of the organization. The function of the internal server is to cooperate with the cloud storage server in performing the search operation, secondly to prevent KGA from doing so. Such an approach can also be extended to a multi-internal server setting. For example, you can set up two internal servers that live in two sections / branches / groups so that each internal server can handle search queries from one section. In such a distribution environment, allowing a multi-reader and multitasked to upload and retrieve data is an important requirement. We should allow multiple data users (or readers) to search the encrypted document uploaded by the data owner (or author), and vice versa.

Finally, it is also important to ensure search pattern privacy, which means that the cloud storage server cannot detect documents that are relevant to the search query (i.e., cloud storage cannot tell whether two search queries give the same or different results, even if they were created by the same user).

RELATED WORKS:

Once the concept of searchable encryption (SE) is placed in it is divided into two categories, searchable symmetric encryption (SSE) and public key encryption with keyword search (PEKS) [2]. SSE with sensitive encryption indicators to significantly speed up the search operation can develop cipher text stream from the prototype of semantic scanning to various advanced structures without various indicators.

As PEKS has attracted more attention from researchers over the past two decades, PEKS works with different features and functions are designed. Several schemes supporting multiple-keyword search have been proposed in the literature [9], [10], [11]. However, these proposed schemes do not support multiple readers and authors at once and do not achieve satisfactory performance. In particular, the trapdoor and cipher size of [9], [11] is as simple as the number of keywords in the processed document or the number of keywords indicated in the query. Additionally, the public key size of [10] is also proportional to the set size. Therefore, it is a challenge to create a public key, trapdoor and cipher with small or fixed size, reduce the computational cost on trapdoor and cipher, and make it easier to expand the universal keyword set.

According to PEKS's syntax, to enable multi-reader access to a single message, multiple searchable ciphers of the same keyword must be generated for different readers, thereby multiplying the computational and storage overhead. Current works that support multi-user access more or less rely on

ISSN : 2454-9924

SSE or broadcast encryption. Their multi-user access represents an author and multiple readers. Sun et al. Cipher text-Policy Attribute-Based Encryption (CP-ABE) was used in conjunction with the cross-tag proposed in to support multi-reader access as well as multi-keyword functionality [16]. In their scheme, an author assigns secret keys to readers, i.e. accessing data outsourced by different authors, each reader must maintain a set of secret keys. In addition, the number of searchable ciphers that can be stored is P w2W jDB [w] j, where DB [w] represents all documents that contain the keyword and W represents the universal keyword set. Each document can be accompanied by multiple searchable ciphers, resulting in a large storage overhead on a large scale system. In 2019, Xu et al. Proposed a lattice-based PEKS scheme that transfers identity from an anonymous identity-based (ID-based) scheme by replacing it with keywords. Their structure is actually ID-based PEKS, which maps the reader identity to the matrix so that the author can use the reader identity to encrypt.

Keyword essay attack (KGA) is a common attack against PEKS. As readers know the public keys, anyone can generate searchable ciphers for the desired keywords and test them against the search trapdoor. To prevent KGA, cryptographic primitives such as public key authentication with keyword search (PAEKS) and public-key encryption with blurred keyword search (PEFKS) have been proposed. The KGA search pattern undermines privacy. Concealment of the search pattern and accessibility pattern should also be considered when building searchable encryption schemes. However, the search pattern privacy is not preserved in many existing schemes where the adversary can tell whether the underlying keywords of the two queries are the same. The access pattern refers to the identifiers of matching documents, which are found in most searchable encryption schemes. Although Oblivius RAM is a viable solution to the problem, current ORAM architectures are still too expensive to be practical.

METHODOLOGY:

In our design, the system consists of the following parties: a Key Generation Center (KGC), a Cloud Platform (CP), multiple Internal Servers (IS's), Data Providers (DPs) and Fig. 1: System model. Request Users (RUs).

- KGC: The Key Generation Center is in charge of generating public parameters, system keys, and keys of CP and IS's, as well as distributing corresponding keys to CP and IS's. For instance, the administrator of an organization can play the role of KGC
- CP: The Cloud Platform is responsible for storing documents and corresponding searchable cipher texts uploaded by DPs. It handles search queries with the assistance from IS's, and generates the (encrypted) searching result for RUs.
- IS: Internal Servers of an organization undertake partial computation to assist CP in handling a search query. In practice, IS's can be designated servers in an organization.
- DP: Each Data Provider generates its own public and secret key pair based on the public parameters, computes the searchable cipher texts according to the keywords associated with a document, and stores the

ISSN : 2454-9924

document with the searchable cipher text on CP.

- RU: Each Request User generates its own public and secret key pair based as DP, and computes the trapdoor for specific keywords of interest. It decrypts the search results sent back from CP and obtains the indexes of the documents satisfying the search query.

ALGORITHMS:

DT-PKC

Distributed Two-Trapdoor Public-Key Cryptosystem (DTPKC) is an useful tool for dealing with integer operations across different encrypted domains by splitting a strong key into shares [25], which is based on partial homomorphism encryption (PHE) [26] and threshold cryptosystems [27], attaining more competitive computation performance than solutions using fully homomorphism encryption (FHE) [28].

Basic Algorithms:

The DT-PKC algorithms in [25] are as follows: Keygen: If the security parameter k is given, it will provide a strong private key SK, a public key pki and a weak private key ski of party i. Encryption (NC): If the message m and party i's public key are given by pki, it provides the cipher [[m]] pki. Decryption with a weak private key (WDec): When the cipher [[m]] pki and the weak private key are given ski, the original message is m. Strong Private Key Splitting (SKeyS): When a strong private key is given SK, it provides two partially strong private keys SK (1) and SK (2). Partial decryption with partial strong private key Step One (PSDec1): When the cipher [[m] pki and partial strong private key SK (1) are given, it executes the partial decryption algorithm PDOSK (1) () and provides the step CT (1) i. Partial decryption with partial strong private key step two (PSDec2): Step One partial cipher CT (1) i, cipher [[m] pki and partial strong private key SK (2), which runs the partial decryption algorithm PDTSK (2) (;) And outputs the original message m. Cipher Refresh (CR): If the cipher [[m]] pki is given, it returns another cipher [[m]] 0 pki of the same message.

Derived Protocols:

The following DT-PKC derived protocols take the same inputs, two ciphers [[x]] pka, [[y] pkb, partially strong private keys SK (1), SK (2) and public keys pka, pkb, pkc. The syntax is as follows: Secure Addition Protocol (SAD) across domains: When input is given, it additionally provides the encryption of [[x + y]] pkc. Secure Multiplication Protocol (SMD) across domains: When input is provided, it provides the encryption of the multiplier [[x y]] pkc. Less secure than protocol (STL): when input is given, it provides the cipher [[u]] pkc, where u = 0 means x y and u = 1 means x <y. Secure Equivalency Protocol (SEQ): When input is given, it provides the cipher [[f]] pkc, where f = 0 means x = y and x 6 = y.

Subset Decision Mechanism:

Assume that the universal set is W = fw 1; ; w0g whose binary representation is (b □1 ;; b0) = (1 ;; 1), and its corresponding decimal integer is SUM. There are two subsets of WT and Wt whose binary representation is (T □1 ;; T0) and (t □1 ;; t0), respectively. Their corresponding decimal integers are T and t, respectively. Wt Our method of determining Wt is to make sure I am not. ti = 1 and Ti = 0. The method of determination is described as follows. Depending on the bit binary representations of T, its complement is (: T □1 ;;: T0), whose integer

ISSN : 2454-9924

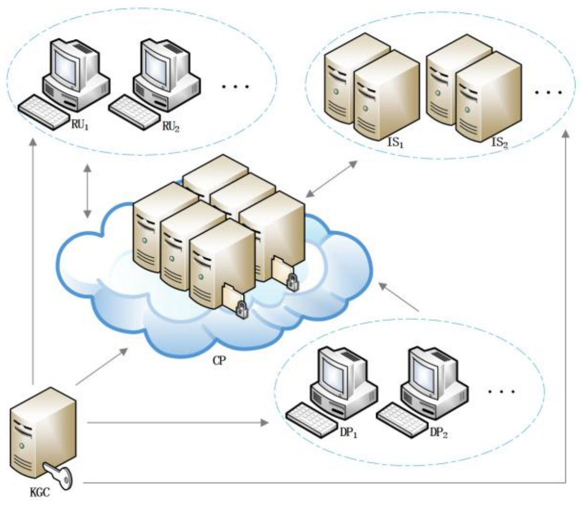
is: T. Then we calculate the bitwise addition of T to the CI and: t. Finally, ci = 2, Wt WT; Otherwise, Wt 6 WT. The formal procedure is described in Algorithm 1.

```
Algorithm 1 Subset Decision
Input: A universal set $\mathcal{W} = \{w_{\mu-1}, \cdots, w_0\}$, two subsets
    $\mathcal{W}_T, \mathcal{W}_t \subseteq \mathcal{W}$.
Output: Whether $\mathcal{W}_t \subseteq \mathcal{W}_T$.
1: Compute the binary representations $(T_{\mu-1}, \cdots, T_0)$,
   $(t_{\mu-1}, \cdots, t_0)$ of $\mathcal{W}_T, \mathcal{W}_t$ .
2: Compute the complement $(\neg T_{\mu-1}, \cdots, \neg T_0)$ of
   $(T_{\mu-1}, \cdots, T_0)$.
3: Set $i = 0, R = 1$.
4: while $i < \mu$ do
5:     $c_i = \neg T_i + t_i$,
6:     $d_i = 2 - c_i$,
7:     $R = R \cdot d_i$,
8: end while
9: if $R = 0$ then
10:    return $\mathcal{W}_t \not\subseteq \mathcal{W}_T$.
11: else
12:    return $\mathcal{W}_t \subseteq \mathcal{W}_T$.
13: end if
```

In addition, our other observation is that WT 6 speeds up the algorithm when WT is present. That is, WW WT, sum = t +: T SUM = 2 □ 1. Accordingly, if sum> SUM, we have Wt 6 WT. The procedure with this modification is described in Algorithm 2. For ease of understanding, we take Table 2 as a toy example to illustrate the subcommittee decision mechanism in Algorithm 2. Proper analysis: Here we demonstrate the accuracy of the algorithm 2. For output 6WT, which divides into two cases? Case-1 is a total SUM but the existing bits are bitwise addition, suppose at least.

```
Algorithm 2 Subset Decision with Modification
Input: A universal set $\mathcal{W} = \{w_{\mu-1}, \cdots, w_0\}$, two subsets
    $\mathcal{W}_T, \mathcal{W}_t \subseteq \mathcal{W}$.
Output: Whether $\mathcal{W}_t \subseteq \mathcal{W}_T$.
1: Besides computations in Step 1, 2 of Algorithm 1, com-
   pute the decimal integers $T, t$ of $\mathcal{W}_T, \mathcal{W}_t, SUM = 2^\mu - 1$
   of $\mathcal{W}, \neg T = SUM - T, sum = \neg T + t$.
2: if $sum > SUM$ then
3:     return $\mathcal{W}_t \not\subseteq \mathcal{W}_T$.
4: else
5:     Go to Step 3 of Algorithm 1.
6: end if
```

ARCHITECTURE:



CONCLUSION:

In this paper, we have introduced the concept of SE-EPOM and formalized its security definitions. Subsequently, we designed the concrete SE-EPOM scheme in the structure distributed with our novel subcommittee decision mechanism and it satisfied our proposed security requirements. In addition, the scheme has attractive features such as multi-keyword search, fixed size trapdoor and cipher support, hiding search pattern and access pattern during search and enabling multi-author / multi-reader setting. Finally, an evaluation of the comparable schemes and our scheme shows that the overall performance of our distributed SEEPOM scheme outperforms other solutions. We leave our future work to secure SE-EPOM design with less rounds of communication between CPs and IS.

REFERENCES:

[1]“Forecast number of personal cloud storage consumers/ users worldwide from 2014 to 2020 (in

ISSN : 2454-9924

millions),” https://www.statista.com/statistics/638593/ worldwide-data-center-storage-capacity-cloud-vs-traditional/, accessed August 30, 2018.

[2] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in International conference on the theory and applications of cryptographic techniques, 2004, pp. 506–522.

[3] J. W. Byun, H. S. Rhee, H.-A. Park, and D. H. Lee, “Off-line keyword guessing attacks on recent keyword search schemes over encrypted data,” in Workshop on Secure Data Management, 2006, pp. 75–83.

[4] Q. Huang and H. Li, “An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks,” Information Sciences, vol. 403, pp. 1–14, 2017.

[5] 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, 2000, pp. 44–55.

[6] E.-J. Goh et al., “Secure indexes.” IACR Cryptology ePrint Archive, vol. 2003, p. 216, 2003.
[7] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and efficient constructions,” Journal of Computer Security, vol. 19, no. 5, pp. 895–934, 2011.

[8] K. Kurosawa and Y. Ohtaki, “Uc-secure searchable symmetric encryption,” in International Conference on Financial Cryptography and Data Security, 2012, pp. 285–298.

[9] P. Wang, H. Wang, and J. Pieprzyk, “Keyword field-free conjunctive keyword searches on encrypted data and extension for dynamic groups,” in International conference on cryptology and network security, 2008, pp. 178–195.

[10] B. Zhang and F. Zhang, “An efficient public key encryption with conjunctive-subset keywords search,” Journal of Network and Computer Applications, vol. 34, no. 1, pp. 262–267, 2011.