Survey on Searchable Public-key Cipher Texts for Privacy Preserving Keyword Search

Vina M. Lomte
Assistant Professor,
Dept. of Computer Engineering
RMD Sinhgad School of Engineering, Pune,
Maharashtra, India

Gauri S. Patil
M.E. Student,
Dept. of Computer Engineering,
RMD Sinhgad School of Engineering, Pune,
Maharashtra, India

ABSTRACT
The Public Key Encryption along with Keyword Search allows one to search the data that is in encrypted form with a keyword without showing any information. This paper gives the detail study on searchable Public-Key Ciphertexts with Hidden Structures (SPCHS) that fastenthe keyword search without sacrificing the security of encrypted keywords. In SPCHS, the keyword ciphertexts is structured by hidden relation and by using a trapdoor function used in cryptography to keywords to disclose minimum information to search algorithm. In SPCHS Schema,cipher texts have hidden star like structure. The SPCHS construction is based on IBKEM i.e. Identity BasedKeyword Encapsulation Management that splits the computation in two parts first that perform heavy computation and other cipher text produced by light computation. The generic SPCHS construction is built with IBE i.e. Identity Based Encryption and Collision-free full-identity malleability IBKEM.

Keywords
Public-key searchable encryption, semantic security, Identity-based key encapsulation mechanism, Identity based encryption.

1. INTRODUCTION
Public Key Encryption performed along with Keyword Search allowssearching the encrypted data with a keyword without disclosing any information [1]. A public key encryption along with keyword search i.e. PEKS [4] scheme does not allow the user to carry decryption of the encrypted keyword or decrypt the encrypted massage. This not to decrypt property has limited the applicability of a PEKS scheme. Considering a scenario that two user A and B where B wants to share documents with user A, B has two options to achieve this work. B stores the documents in the mobile devices such as flash drive and portable hard disk, and sends it to A. However there are many uncertain situations in delivering process; for example, devices might have the unauthorized access so to cause huge damage to the company. Another option for B to share documents is stored in the server that is considered as the storage media. Traditionally, users store their respective data in the server and take the server as a storage media. Users can access and change the data in short period of time, and they can authorize other users in order to use the data for some special purposes. However, the security, integrity and confidentiality of data in the remote server cannot be guaranteed because data cannot be controlled and supervised by user directly [7]. Since document is transformed into cipher text, it creates another problem that how user can obtain the encrypted data without decrypting them. Thus one of the solutions is user download all encrypted data and decrypted them so that user can find the right document they want without revealing any information to the server administrators. This can cause transfer cost and storage space whenever Queried data [8]. Another solution is to setup keywords for each encrypted document with specific keywords they wish to query. For example user A wish to only retrieve the document which contain word W, downloading whole encrypted data is not suitable solution. Inorder to provide the efficient search performance without losing the semantic security in PEKS. Since semantic security regarding keywords privacy is difficult to get [11]. The major hurdle is to have linear search complexity of existing schemes [1].Thus to increase search performance in PKES without losing semantic security, organize cipher text designed with hidden relation. If keyword searchable cipher texts have hidden star like structure then search for cipher texts containing keywords can be fasten [1]. In accordance to the semantic security, defining the Searchable public-key cipher texts with hidden (SPCHS) [1]. In this concept the keywords searchable Cipher texts with the hidden structures is generated in public key settings. It consists of the keyword search trapdoor that discloses partial relation in order to find out the all matching cipher texts. In this the semantic security is needed for both the hidden structures and the keywords. On the other side the previous PEKS do not contain hidden structures among the PEKS cipher texts .In previous scheme the semantic security is defined to only keywords [9]. The resultant SPCHS was constructed in order to generate keywords searchable cipher texts with hidden star-like structure. The generic SPCHS is constructed with IBE and Collision free full Identity malleable IBKEM. Hence, the new IBKEM scheme builds SPCHS schemes securely in the standard model with the same search performance as similar to the SPCHS construction from scratch in the RO model.

2. LITERATURE REVIEW
Search on encrypted data has been investigated in recent years. From a Cryptographic perspective, the existing works fall into two categories, i.e., symmetric searchable encryption and public-key searchable encryption. Searchable symmetric encryption (SSE) [2] allows a party to outsource the storage of its data to another party (a server) in a private manner, while maintaining the ability to selectively search over it. This problem has been the focus of active research in recent years. Public Key Encryption with Keyword Search (PEKS) scheme enable one to search the encrypted data with a keyword without revealing any information and preserving its semantic security [1].[1]Proposedsearchable public-key ciphertexts with hidden structures (SPCHS) for keyword search as fast as possible without sacrificing semantic security of the encrypted keywords. In SPCHS, all keyword-searchable ciphertexts are structured by hidden relations, and with the search trapdoor corresponding to a keyword, the minimum information of the relations is disclosed to a search algorithm as the guidance to find all matching cipher texts efficiently. [2] introduced a new

methodfor constructing sub-linear SSE (Searchable symmetric encryption) schemes. The approach is highly parallelizable and dynamic. Previous the only method for achieving sub-linear time search is the inverted index approach, which requires the search algorithm to access a sequence of memory locations. A new approach for designing SSE schemes that yields constructions with sub-linear search time but that has none of the limitations of the inverted index approach. In particular approach is simple, highly parallel and can easily handle updates. Scheme also achieves the following important properties: (a) it enjoys a strong notion of security, namely security against adaptive chosen-keyword attacks; (b) compared to existing sub-linear dynamic SSE schemes updates in our scheme do not leak any information, apart from information that can be inferred from previous search tokens; (c) it can be implemented efficiently in external memory (with logarithmic I/O overhead). The technique is simple and uses a red-black tree data structure. [3] Provides Asymmetric searchable encryption (ASE) schemes which support two special features, namely message recovery and flexible search authorization. The message recovery feature requires that a cipher text not only allows the data owner to recover the plaintext but also allows third-party servers to search in it. The flexible searchable authorization feature requires that the data owner can authorize a third-party server in three different ways: (1) authorize the server to search any message at the data owner's interest by assigning a message-dependent trapdoor (i.e. the server can only determine whether the message encoded in the trapdoor is equal to the plaintext inside a cipher text); (2) authorize the server to search any message at the server's interests by assigning a master trapdoor (i.e. the server can choose a message at its will and see whether it is equal to the plaintext inside any cipher text); (3) authorize the server to perform both types of searches. [4] Proposed PEKS, where a proxy server, who responds the keyword queries of a receiver, can know the content of keywords by implementing KGA. Moreover, it is efficient under the practical condition that the size of the keyword space is not more than the polynomial level. [6] Givesbroader view on what can be achieved regarding trapdoor privacy in asymmetric searchable encryption schemes, and bridge the gap between previous definitions, which give limited privacy guarantees in practice against search patterns. The paper proposes the notion of Strong Search Pattern Privacy for PEKS and constructs a scheme that achieves this security notion.

3. SYSTEM ARCHITECTURE

In Searchable Public-key Cipher text with Hidden Structure(SPCHS),Keyword searchable ciphertexts with hidden structure is generated in public key setting; with the keyword search trapdoor[6] ,partial relation is disclosed to find all matching ciphertexts.Advantage of SPCHS over the traditional scheme is it provides semantic security for both keywords and hidden structures .Scheme generate keyword searchable ciphertexts with hidden star like structure.

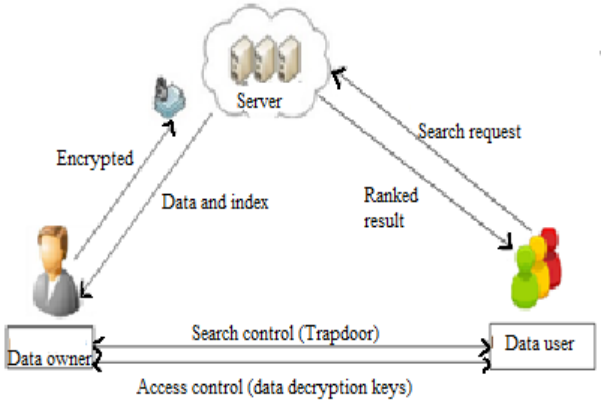

Fig 3.1: System Architecture

The architecture of the system is described in detail in the given figure [1]; in this the data owner uploads the data in the cloud with the help of essential storage considering as block storage. Before uploading the data, it is encrypted. The document stored in the cloud is in form of the encrypted form. The text mining process is a natural language processing used to retrieve the file and appropriate tools is used to obtain the files and contents. Natural Language Processing (NLP) process is used to extract the meaningful words in available in the file content. Data user tries to search a necessary user's query in the cloud server. The cloud server will perform the mapping of the keywords and the keywords mentioned by the user during the search of the related files. The cloud server gives the related filename to user based on the keywords mapping. To view the information, the user click the filename, on clicking the file name user request the requested file to cloud server and in response the server send the user details and file name to the respective data owner. Then data owner is familiar with all public key of user so as to it can be used to encrypt data by the data users private key and its public part public key and encrypted key is send to the server in turn the server will send that key related information to user, then user decrypt the key by using the provided private key. After that the data users acquires the private key of the data owner and then access the required data. In the entire system the search user can use various necessary keywords to search the data it is interested in to. The data owner uses the term and inverse term frequency to choose the necessary keywords. The data and index are both encrypted in order to preserve the security of both the documents and the index. The search user provides the key containing the keyword to the server. Server uses this key to provide the ranking based result to the search users. These results are obtained by the search user to get the most relevant search matching the keyword to obtain the accurate document. The hidden relation has enhanced the system performance by improving the search over the encrypted data [1]. The complexity of the system depends on the queried cipher text. The obtained one hidden relation helps to obtain another relation in the chain.

4. SEARCHABLE PUBLIC-KEY CIPHER TEXT WITH HIDDEN STRUCTURE

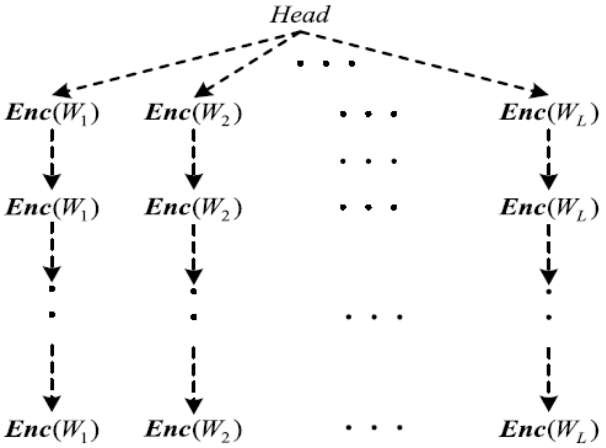

Fig: 4.1 Hidden star-like structures

The system have targeted on to increase search performance in PKES without sacrificing semantic security, where sender separately encrypt a file and its extracted keywords and send the resulting cipher texts to a server, the receiver when want to retrieve the file containing a specific keyword, it delegate a keyword search trapdoor to server; server finds the encrypted files containing the queried keywords without knowing the original file or keyword and return the corresponding encryptedfile to receiver and receiver decrypt that file. The keyword searchable cipher text form hidden star-like structure as shown in figure[1]. [1] Here the dashed arrows denote the hidden relations. Enc (W_i) denotes the searchable Cipher text of keyword Wi. All cipher texts have same keywords that form chain by correlated hidden relation also hidden relation exists from public head to first cipher texts of each chain. With keyword search trapdoor and head, the server check the first matching cipher texts through the corresponding relation from head. By carrying this all matching cipher texts can be found .Thus search time depend on the actual number of cipher texts containing the queried keyword rather than total number of all cipher texts.

4.1 Scheme from collision-free full-identity malleable IBKEM

A SPCHS scheme is formed with IBE Collision-free full - Identity Malleable IBKEM with semantic security [1]. Several interesting properties are identified i.e. collision-freeness and full-identity malleability in some IBKEM instances, and formalized these properties to build a SPCHS. Given are two collision-free full-identity malleable IBKEM instances that are completely secured. In IBKEM, a sender encapsulates a key K to an intended receiver ID. Of course, receiver ID can DE capsulate and obtain K, and the sender knows that receiver ID will obtain K. However, a non-intended receiver ID1 may also try to DE capsulate and obtain K1. [1] It is observed that:

1. It is usually the case that K and K1 are independent of each other from the view of the receivers.
2. In some IBKEM the sender may also know K1 obtained by receiver ID1.

This can be referred to the former property as collision-freeness and to the latter as full-identity malleability. An IBKEM scheme contains some properties, depending on this property it is said to be collision-freefull-identitymalleable if it possesses both properties. If both underlying IBKEM and IBE have semantic security and the privacy of receiver's identities, the SPCHS is semantically secure containing this properties.Collision-free full-identity malleable IBKEM [1]:

1. K and ID = Sender encapsulate key K to an intended receiver ID, of course receiver ID can be DE capsulated to obtain K.
2. K1 and ID1= Non-intended receiver ID1 will also try to DE capsulate toobtain K1.

Two cases are observed:

1. Collision-free - It is case that K and K1 are independent of each other from receiver view.
2. Full-identity malleability -In some IBKEM, the sender may also know K1 obtained by receiver ID1.

Semantic security to SPCHS, includes the probabilistic polynomial time (PPT) Adversary that is allowed to know all structure public parts, query the private part of chosen structure, Query the trapdoor for keywords and query cipher texts of keywords.SS-CKSA is Semantic Security of SPCHS to Chosen Keyword and Structure Attack [10]; the adversary will choose two challenge keyword-structure pairs. This SS-CKSA security means that for ciphertexts of one of two challenge keyword structure pair.

5. SPCHS CONSTRUCTION

The SPCHS definition consists of five algorithms[1].

Step 1: SystemSetup{W, 1^k}, the system inputs keyword space W and security parameter 1^k.Run the pair of master keys (PK, SK) = Setup (1^k,ID). It outputs the pair of master public and secret key {PK, SK}where, PK contains keyword W cipher texts C.

Step 2: StructureInitialization{PK}, the system inputs the public key and generates the encapsulated key. Initialize the hidden structure. It outputs the hidden relation {Pri, Pub} and initializing a hidden structure.

Step 3: StructuredEncryption{PK,W,Pri}, inputs the PK,W,Pri. Search for the keyword in Pri and output the cipher text with keyword W and update Pri. Where, Pri is hidden relation in C.

Step 4: Trapdoor {SK, W}, inputs the secret key and keyword W and gives keyword search trapdoor Tw of W

Step 5: StructureSearch{PK,Pub,C,Tw}, inputs the public key, Pub, C, Tw. And discloses partial relation to find out cipher texts containing keywords W with hidden structure

The generic SPCHS is constructed based on the properties observed in the IBKEM.

6. MATHEMATICAL MODEL

Collision-free full-identity malleable IBKEM[1].

Set input value**s {**W,(PK,SK), C, (Pub,Pri),Tw,(K,C),1^K}

Output value**:=** {(PK,SK), (PK_{IBKEM} ,PK_{IBE}), (SK_{IBKEM} ,SK_{IBE}), (Pri=(u) ,Pub = (C)), Tw , FIM(Wi,u)}

(1) $(PK_{IBKEM},SK_{IBKEM})=Setup_{IBKEM}(1_K,ID_{IBKEM})$ and $(PK_{IBE},SK_{IBE})=Setup_{IBE}(1_K,ID_{IBE})$

(2) $(K,C)=Encaps_{IBKEM}(PK_{IBKEM},W,u)$

(3) $C =(FIM(W,u),Enc_{IBE}(PK_{IBE},W,Pt(u,W)))$

(4) StructuredSearch*(*PK, Pub,C, Tw_i*)*

7. CONCLUSION AND FUTURE SCOPE

Public Key Encryption along with Keyword Search (PEKS) scheme allows one to search the encrypted data with a keyword without revealing any information. Thus the searchable public key ciphertext with hidden structure for keyword search (SPCHS)makes search as fast as possible without sacrificing its semantic security .To increase search performance in PKES without sacrificing semantic security ,the ciphertext are designed with hidden relation. If keyword searchable have hidden star like structure then search for cipher texts containing keywords can be fasten.

The future scope is to investigate on the authentication and consider various access control issues in searchable encryption technique.

8. REFERENCES

[1] PengXu, Qianhong Wu, Wei Wang, Willy Susilo, Josep Domingo-Ferrer, and Hai Jin, Member, IEEE, "Generating Searchable Public-Key Cipher texts With Hidden Structures for Fast Keyword Search,"IEEE Transaction On Info forensic and security, Vol.10,No.9(2015).

[2] S. Kamara and C. Papamanthou, "Parallel and dynamic searchable symmetric Encryption," in Financial Cryptography and Data Security, vol. 7859, pp.258-274, (2013).

[3] Q. Tang and X. Chen, "Towards asymmetric searchable encryption with message recovery and flexible search authorization," in Proc. ASIACCS, (2013), pp. 253264.

[4] P. Xu, H. Jin, Q. Wu, and W. Wang, "Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack," IEEE Trans. Compute., vol. 62, no. 11, pp. 22662277, (2013).

[5] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in Proc. IEEE INFOCOM, Mar. 2010, pp. 15.

[6] A. Arriaga, Q. Tang, and P. Ryan, "Trapdoor privacy in asymmetric searchable encryption schemes, "vol. 8469, D. Pointcheval and D. Vergnaud, Eds. Berlin, Germany: Springer-Verlag, (2014), pp.3150.

[7] D. J. Park, K. Kim, and P. J. Lee, "Public key encryption with conjunctive field keyword search," in Information SecurityApplications(Lecture Notes in Computer Science), vol. 3325, C. H. Lim and M. Yung, Eds. Berlin, Germany: Springer-Verlag, 2005, pp. 73–86.

[8] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in Applied Cryptography and NetworkSecurity(Lecture Notes in Computer Science), vol. 3089, M. Jakobsson, M. Yung, and J. Zhou, Eds. Berlin, Germany: Springer-Verlag, 2004,pp. 31–45.

[9] L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," in Information and CommunicationsSecurity(Lecture Notes in Computer Science), vol. 3783, S. Qing, W. Mao, J. López, and G. Wang, Eds. Berlin, Germany: Springer-Verlag, 2005, pp. 414–426.

[10] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in Pairing-Based Cryptography—Pairing(Lecture Notes in Computer Science), vol. 4575, T.Takagi, T.Okamoto, E.Okamoto, and T Okamoto, Eds. Berlin, Germany: Springer-Verlag, 2007, pp. 2–22.

[11] E.-K.Ryu and T.Takagi "Efficient conjunctive keyword-searchable encryption," in *Proc.* 21st Int. Conf. *Adv. Inf.* Netw. Appl. Workshops, May 2007, pp. 409–414.