Secure Keyword Search with Public Key Encryption by Cloud storage

*T. Nagamani[1], Dr. V.Senthilkumar[2], S.Varuna[3]

[1]Assistant Professor, Department of Computer Science and Engineering
[2]Assistant Professor (Sr.G), Department of Civil Engineering
[3]Assistant Professor, Department of Computer Science and Engineering Bannari Amman Institute of Technology, Sathyamangalam, Erode, India
Corresponding Author: *T. Nagamani[1]

ABSTRACT

Cloud computing allows the users to outsource their data using cloud storage servers in order to reduce the economic cost. Cloud computing and storage solutions provide users and enterprises to store and process their data in third-party data centers that may be located far from the user–ranging in distance from across a city to across the world. Encryption is a potential way to protect the confidentiality of the outsourced data, but it also introduces much difficulty to performing effective searches over encrypted information. Although traditional searchable encryption scheme, Public key Encryption with Keyword Search (PEKS), allow users to securely search over encrypted data through keywords, these techniques support only boolean search. Unfortunately, it is demonstrated that the customary PEKS framework suffers from an inalienable insecurity called inside Keyword Guessing Attack (KGA) launched by the malevolent server. To address this security vulnerability, it is proposed a new PEKS framework named Dual-Server PEKS (DS-PEKS).

Keywords: Boolean search, Data Privacy, Data Retrieval, Keyword Guessing Attack, Keyword Privacy, Keyword Search Scheme, Keyword generation.

Date of Submission: 01-08-2017 Date of acceptance: 14-08-2017

I. INTRODUCTION

Cloud storage outsourcing is becoming a popular application for enterprises and organizations to reduce the load of maintaining big data in recent years. However, in reality, end users may not completely trust the cloud storage servers and may prefer to encrypt their data before uploading them to the cloud server in order to protect the data privacy. This usually makes the data utilization more difficult than the traditional storage where data is kept in the absence of encryption. One of the typical solutions is the searchable encryption which allows the user to retrieve the encrypted documents that contain the user-specified keywords, where given the keyword trapdoor, the server can find the data required by the user without decryption. Boneh *et al.* [2] introduced a more flexible primitive, namely Public Key Encryption with Keyword Search (PEKS) that enables a user to search encrypted data in the asymmetric encryption setting. In a PEKS system, using the receiver's public key, the sender attaches some encrypted keywords (referred to as PEKS cipher texts) with the encrypted data. The receiver then sends the trapdoor of a to-be-searched keyword to the server for data searching. The server can test whether the keyword underlying the PEKS cipher text is equal to the one selected by the receiver from the given trapdoor and the PEKS cipher text. If so, the server sends the matching encrypted document to the receiver.

II. RELATED WORK

In this subsection, Classification of PEKS schemes is described based on their security.

2.1 Traditional PEKS: Following Boneh et al.'s seminal work [2], Abdalla et al. [8] formalized anonymous IBE (AIBE) and presented a generic construction of searchable encryption from AIBE. They also showed how to transfer a hierarchical IBE (HIBE) scheme into a public key encryption with temporary keyword search (PETKS) where the trapdoor is only valid in a specific time interval. Waters et al. [7] showed that the PEKS schemes based on bilinear map could be applied to build encrypted and searchable auditing logs. In order to construct a PEKS secure in the standard model, Khader [9] proposed a scheme based on the k-resilient IBE and also gave a construction supporting multiple-keyword search. The first PEKS scheme without pairings was introduced by Di Crescenzo and Saraswat [11]. The construction is derived from Cock's IBE scheme [12] which is not very practical.

2.2 Secure Channel Free PEKS: The original PEKS scheme [2] requires a secure channel to transmit the trapdoors. To overcome this limitation, Baek et al. [13] proposed a new PEKS scheme without requiring a secure channel, which is referred to as a secure channel-free PEKS (SCF-PEKS). The idea is to add the server's public/private key pair into PEKS system. The keyword cipher text and trapdoor are generated using the server's public key and hence only the server (designated tester) is able to perform the search. Rhee et al. [14] later enhanced Baek et al.'s security model [13] for SCF-PEKS where the attacker is allowed to obtain the relationship between the non-challenge cipher texts and the trapdoor. They also presented an SCF-PEKS scheme secure under the enhanced security model in the random.

2.3 Against Outside KGA: Byun et al. [16] introduced the off-line keyword guessing attack against PEKS as keywords are chosen from a much smaller space than passwords and users usually use well-known keywords for searching documents. They also pointed out that the scheme proposed in Boneh et al. [2] was susceptible to keyword guessing attack. Inspired by the work of Byun et al. [16], Yau et al. [17] demonstrated that outside adversaries that capture the trapdoors sent in a public channel can reveal the encrypted keywords through off-line keyword guessing attacks and they so showed off-line keyword guessing attacks against the CF-)PEKS schemes in [13] and [18]. The first PEKS scheme secure against outside keyword guessing attacks was proposed by Rhee et al. [19]. In [20], the notion of trapdoor indistinguishability was proposed and the authors showed that trapdoor indistinguishability is a sufficient condition for preventing outside keyword-guessing attacks. Fang et al. [5] proposed a concrete SCF-PEKS scheme with (outside) KGA resilience. Similar to the work in [15], they also considered the adaptive test oracle in their proposed security definition.

III. EXISTING SEARCHABLE ENCRYPTION FRAMEWORK

Existing searchable encryption frameworks such as PEKS [1], [2], [3], [4–6] etc. were based on bilinear pairing and trapdoor functions. Consider a scenario where the user wants to upload their files to a remote server. Initially, user and server must agree on a set of cryptographic parameters for secure file storage and retrieval. In order to store a file in a secure manner, user encrypts the file along with its associated keyword using their private key.

$$I = E\mathbf{K}\ (F, KW)\ldots\ldots\ldots\ldots\ldots\ldots. (1)$$

Where,
I – Index of the encrypted file and keyword
K – Encryption Key (User's Public or Private Key)
F – File that needs to be stored in a secure manner on remote server
KW (KW**1**, KW**2**… KW**n**) – Keywords related to the file name and content

Index I is created by the encryption of file and keyword using the user's private key. In order to search data the user generates Trapdoor (K, KW). This trapdoor is used by the server to verify whether the given keyword is present in the index I. If it exists, then server returns the appropriate document related to that keyword.

3.1 Limitations of existing work

The traditional PEKS schemes suffer from an inalienable insecurity regarding the trapdoor keyword privacy, namely, Keyword Guessing Attack (KGA). The reason leading to such security vulnerability is that anyone who knows receiver's public key can generate the PEKS cipher text of arbitrary keyword themselves. Specifically, from the keyword space, the adversarial server can choose a guessing keyword by given a trapdoor and then use that keyword to generate a PEKS cipher text. The server then can test whether the guessing keyword is the one underlying the trapdoor. This guessing-then-testing procedure can be repeated until the correct keyword is found. Such a guessing attack has also been considered in many password based systems. However, the attack can be launched more efficiently against PEKS schemes since the keyword space is roughly the same as a normal dictionary (e.g., all the meaningful English words)

3.2 Proposed Secure Keyword Encryption Framework

In this section, it is formally defined the Dual Server - Public Key Encryption with Keyword Search (DS-PEKS) and its security architecture model.

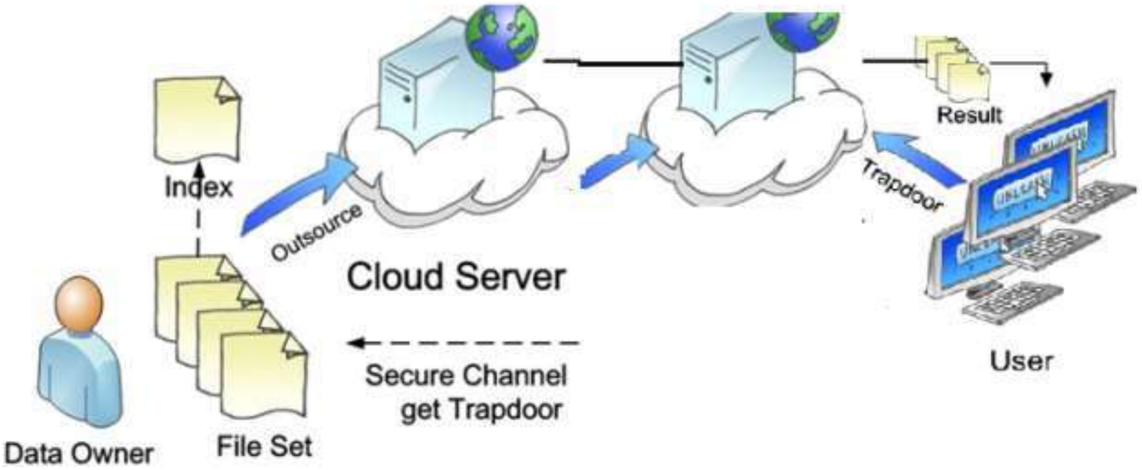

Figure 1: System architecture

3.2.1 Data Owner

Firstly, Data owner in Fig.1 must register with cloud server and then login (username must be unique) must be established. Then, owner send request to Public Key Generator (PKG) to generate Key on the registered user name. While browsing the file, owner makes the request for Public key to encrypt the data and upload the data to cloud service provider. Finally, it is to be verified from the cloud.

3.2.2 Public Key Generator

After receiving the request from the users, key is generated. In this way, all the keys are stored based on the user names. By checking the username, private key is to be provided then revoke the end user (File Receiver if they try to hack file in the cloud server and revoke the user after updating the private key for the corresponding file based on the user).

3.2.3 Key Update

All the files are received from the data owner then these are stored in cloud. Then, the data integrity is to be checked in the cloud and inform to end user about the data integrity. Request is to be sent to PKG in order to update the private key of the user based on the date parameter (Give some date to update the Private Key). Then, make to list all the files and all the updated Private Key details based on the date and users and also make to list all File attackers and File Receive Attackers.

3.2.4 Design of algorithm

A DS-PEKS scheme mainly consists of (KeyGen, DS - PEKS, DS - Trapdoor, FrontTest, BackTest).To be more precise, the Key Generation algorithm generates the public/ private key pairs of the front and back servers (PU_{BS}, PR_{BS}, PU_{FS}, PR_{FS}) instead of that of the receiver. Moreover, the trapdoor generation algorithm, DS - Trapdoor defined here, is public, while, in the traditional PEKS definition [2], [13], the Trapdoor algorithm takes the receiver's private key as input. Such a difference is mainly due to the different structures used by the two systems. In the traditional PEKS, since there is only one server, if the trapdoor generation algorithm is public, then the server can launch a guessing attack against a keyword cipher text to recover the encrypted keyword. As a result, it is impossible to achieve the semantic security as defined in [2] and [13]. However, as we will show later, under the DS-PEKS framework, we can still achieve semantic security, when the trapdoor generation algorithm is public. Another main difference between the traditional PEKS and our proposed DS-PEKS is that the test algorithm is divided into two algorithms, Front server testing algorithm and Back server testing algorithm run by the two independent servers. This is essential for achieving security against the inside keyword guessing attack (KGA).

In the DS-PEKS system, the query is received from the receiver, then the front server pre-processes the trapdoor and all the PEKS cipher texts using its private key, and then sends some internal testing-states to the back server with the corresponding trapdoor and PEKS cipher texts hidden. The back server can then decide which documents are queried by the receiver using its private key and also receive the internal testing-states from the front server, that corresponding document will be received by the user. The algorithms involved in DS-PEKS are as follows.

a. Key generation Algorithm:
The output, public key-private key pair of the front server (PU_{FS}, PR_{FS}) and back server(PU_{BS},PR_{BS}), can be calculated using system parameter P as input.
b. Dual server-PEKS algorithm:
The output, cipher text CT_{KW1}, is generated using the input parameters called P, the front server's public key PU_{FS}, the back server's public key PU_{BS} and the keyword KW1.
c. Dual server – Trapdoor algorithm:
The output, the trapdoor T_{KW2}, is generated using the input parameters P , the front server's public key PU_{FS} , the back server's public key PU_{BS} and the keyword KW2.
d. Front server Testing Algorithm:
The output, the internal testing-state C_{ITS}, is generated using the input parameters P , the front server's private key PR_{FS} , the PEKS ciphertext CT_{KW1} and the trapdoor T_{KW2}.
e. Back server Testing Algorithm:
This algorithm outputs the testing results either 0 or 1 by using the back server's private key PR_{BS} and the internal testing-state C_{ITS}.
f. Consistency
The correctness required by DS-PEKS ensures that the test function always outputs the correct answer.

IV. RESULTS AND DISCUSSION

4.1 Performance Evaluation

In this section, performance is evaluated by making the comparison between existing schemes and our scheme in terms of computation, size and security.

All the existing schemes [2], [10], [14] require the pairing computation during the generation of PEKS cipher text and testing. Hence, these schemes are less efficient than our scheme, which does not need any pairing computation. In our scheme, the computation cost of PEKS generation and testing are calculated.

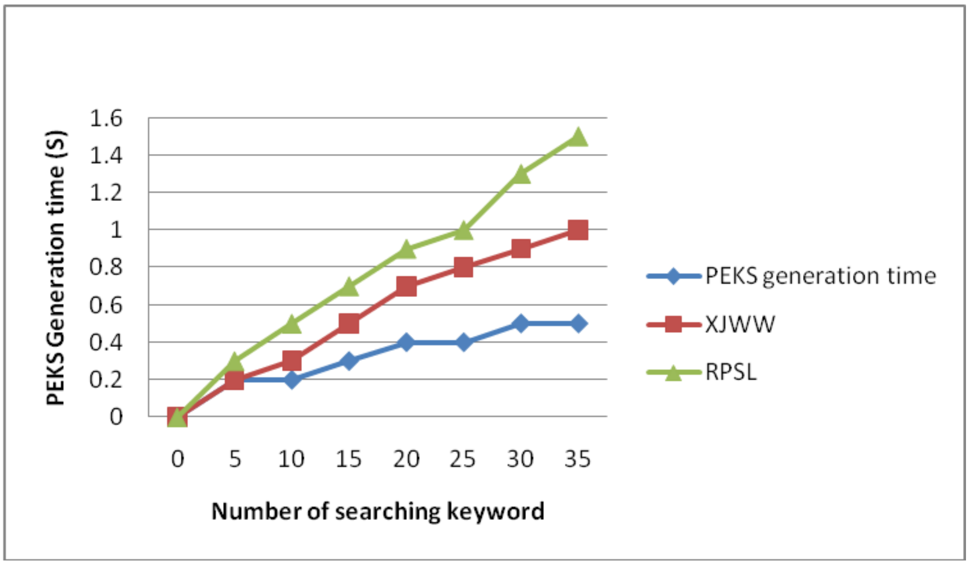

Figure 2: Computation cost of PEKS generation in different schemes.

As shown in Figure. 2, our scheme is the most efficient in terms of PEKS computation. It is because that our scheme does not include pairing computation. Particularly, the scheme [10] requires the most computation cost due to 2 pairing computation per PEKS generation. All the existing schemes do not involve pairing computation, as for the trapdoor generation concerned and also the computation cost is much slower than that of PEKS generation.

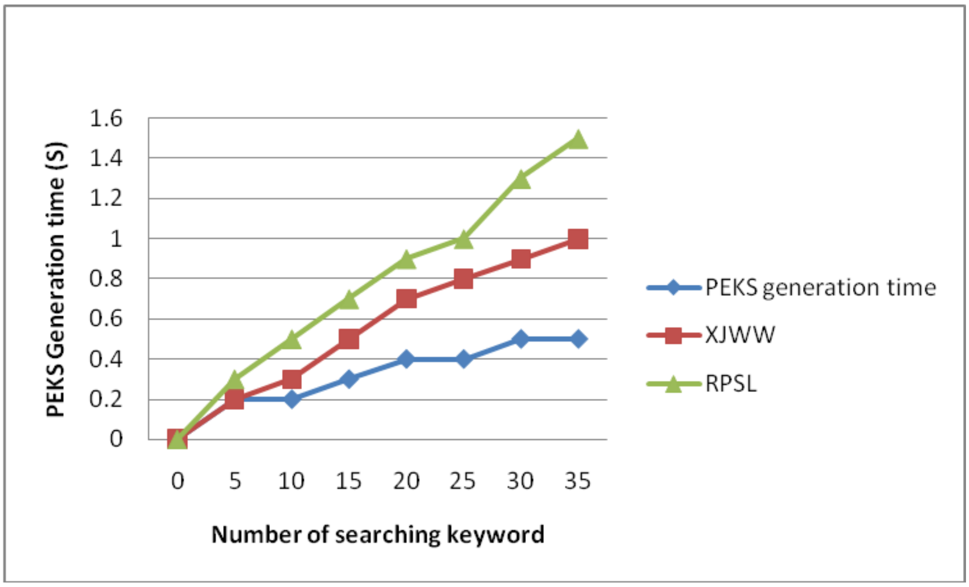

Figure 3. Computation cost of testing in different schemes.

It is worth noting that trapdoor generation in our scheme is slightly higher than those of existing schemes due to the additional exponentiation computations. When the searching keyword number is 30, the total computation cost of our scheme is about 0.5 seconds. As illustrated in Fig. 3, the scheme [10] cost the most time due to an additional pairing computation in the exact testing stage. One should note that this additional pairing computation done on the user side instead of the server. Therefore, it could be the computation burden for users who may use a light device for searching the data. In our scheme, although it also requires another stage for the testing, our computation cost is actually lower than that of any existing scheme as it do not require any pairing computation and all the searching work is handled by the server.

V. CONCLUSION

In this paper, the new proposed framework, named, Dual Server Public Key Encryption with Keyword Search (DS-PEKS), that can prevent the Keyword Guessing Attack(KGA) which is an inherent vulnerability of the traditional PEKS framework. Furthermore, the proposed scheme allows the server to participate in the encryption operation, thus a data owner could pay less computational cost for encryption, without leaking any information about the plaintext. It is determined that the suitability and efficiency of DS-PEKS scheme is for practical use in the cloud environment.

REFERENCES

[1]. R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "A new general framework for secure public key encryption with keyword search," in Proc. 20th Australasian Conf. Inf. Security Privacy (ACISP), pp. 59–76, 2015.

[2]. Boneh D, Di Crescenzo G, Ostrovsky R, Persiano G, "Public key encryption with keyword search", in Proc. Int. Conf. Advances in Cryptology -EUROCRYPT, pp. 506–522, 2004.

[3]. Fang L, Susilo W, Ge C, Wang J. A, "Secure channel free public key encryption with keyword search scheme without random oracle", Cryptology and Network Security, pp.248–258, 2009.

[4]. Park, Dong Jin, Kihyun Kim, and Pil Joong Lee, "Public Key Encryption with Conjunctive Field Keyword Search", Vol. 4, pp. 73–86, 2004.

[5]. Fang L, Susilo W, Ge C, Wang J., "Public key encryption with keyword search secure against keyword guessing attacks without random oracle." Information Sciences, pp. 221- 241, 2013

[6]. Bellare M, Rogaway P, "Random oracles are practical: A paradigm for designing efficient protocols", Proceedings of the 1st ACM conference on Computer and communications security, pp. 62–73, 1993.

[7]. Canetti, Ran, Oded Goldreich, and Shai Halevi, "The random oracle methodology, revisited", Journal of the ACM (JACM), 51(4), pp. 557–94, 2004.

[8]. Abdalla, Michel, et al. "Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions.", in Proc. 25th Annu. Int. Conf. CRYPTO, Vol. 3621, pp. 205– 222, 2005.

[9]. D. Khader, "Public key encryption with keyword search based on K-resilient IBE", in Proc. of Int. Conf. Comput. Sci. Appl. (ICCSA), pp. 298–308, 2006.

[10]. 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. Comput., vol. 62, No. 11, pp. 2266–2277, 2013.

[11]. G. Di Crescenzo and V. Saraswat, "Public key encryption with searchable keywords based on Jacobi symbols" , in Proc. 8th Int. Conf. INDOCRYPT, pp. 282–296, 2007.

[12]. Cocks, Clifford, "An identity based encryption scheme based on quadratic residues", in Cryptography and Coding. Cirencester, U.K.: Springer, pp. 360–363, 2001.

[13]. J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited", in Proc. Int. Conf. Comput. Sci. Appl. (ICCSA), pp. 1249–1259, 2008.

[14]. H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Trapdoor security in a searchable public-key encryption scheme with a designated tester," Journal of Systems and Software 83.5,pp. 763-771, 2010.

[15]. K. Emura, A. Miyaji, M. S. Rahman, and K. Omote, "Generic constructions of secure-channel free searchable encryption with adaptive security", Secur. Commun. Netw., vol. 8, No. 8, pp. 1547–1560, 2015.

[16]. 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 Proc. 3rd VLDB Workshop Secure Data Manage. (SDM), pp. 75–83, 2006.

[17]. Yau, Wei-Chuen, Swee-Huay Heng, and Bok-Min Goi. , "Off-line keyword guessing attacks on recent public key encryption with keyword search schemes," in Proc. 5th Int. Conf. ATC, pp. 100–105, 2008.

[18]. Baek, Joonsang, Reihaneh Safavi-Naini, and Willy Susilo, "On the integration of public key data encryption and public key encryption with keyword search", in Proc. 9th Int. Conf. Inf. Secur. (ISC), Vol. 4176, pp. 217–232, 2006.

[19]. Rhee, Hyun Sook, Willy Susilo, and Hyun-Jeong Kim., "Secure searchable public key encryption scheme against keyword guessing attacks," IEICE Electronics Express 6.5, vol. 6, no. 5, pp. 237–243, 2009.

[20]. Rhee, Hyun Sook, J. H. Park, W. Susilo, and D. H. Lee, "Trapdoor security in a searchable public-key encryption scheme with a designated tester," Journal of Systems and Software 83.5, vol. 83, no. 5, pp. 763–771, 2010.

International Journal of Computational Engineering Research (IJCER) is UGC approved Journal with Sl. No. 4627, Journal no. 47631.

* T. Nagamani " Secure Keyword Search with Public Key Encryption by Cloud storage " International Journal of Computational Engineering Research (IJCER) 7.8 (2017): 40-45.