International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 07
June 2017

# A Review of the Research on Public Key Encryption of Dual Server with Keyword Search for Secure Cloud Storage

**1 M.MADHAVI 2 D.VENKATA SIVAREDDY**

**Abstract**— Searchable encryption is of increasing interest for protecting the data privacy in secure searchable cloud storage. In this paper, we investigate the security of a well-known cryptographic primitive, namely, public key encryption with keyword search (PEKS) which is very useful in many applications of cloud storage. Unfortunately, it has been shown that the traditional PEKS framework suffers from an inherent insecurity called inside keyword guessing attack (KGA) launched by the malicious server. To address this security vulnerability, we propose a new PEKS framework named dual-server PEKS (DS-PEKS). As another main contribution, we define a new variant of the smooth projective hash functions (SPHFs) referred to as linear and homomorphic SPHF (LH-SPHF). We then show a generic construction of secure DS-PEKS from LH-SPHF. To illustrate the feasibility of our new framework, we provide an efficient instantiation of the general framework from a Decision Diffie–Hellman-based LH-SPHF and show that it can achieve the strong security against inside the KGA.

.

## I. INTRODUCTION

**C**LOUD storage outsourcing has become a popular application for enterprises and organizations to reduce the burden of maintaining big data in recent years. However, in reality, end users may not entirely 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. Searchable encryption can be realized in either symmetric or asymmetric encryption setting. In [2], Song et al. proposed keyword search on ciphertext, known as Searchable

Symmetric Encryption (SSE) and afterwards several SSE schemes [3], [4] were designed for improvements. Although SSE schemes enjoy high efficiency, they suffer from complicated secret key distribution. Precisely, users have to securely share secret keys which are used for data encryption. Otherwise they are not able to share the encrypted data outsourced to the cloud. To resolve this problem, Boneh *et al.* [5] 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 ciphertexts) with the encrypted data. The receiver then sends the trapdoor of a to-be-searched keyword to the server for data searching. Given the trapdoor and the PEKS ciphertext, the server can test whether the keyword underlying the PEKS ciphertxt is equal to the one selected by the receiver. If so, the server sends the matching encrypted data to the receiver

## 2 Related Work

In this subsection, we describe a classification of PEKS schemes based on their security.

**1) Traditional PEKS***:* Following Boneh et al.'s seminal work [5], 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) Secure Channel Free PEKS***:* The original PEKS scheme [5] 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 a PEKS system.

The keyword ciphertext 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 securitymodel [13] for SCF-PEKS where the attacker is allowed toobtain the relationship between the non-challenge ciphertextsand the trapdoor. They also presented an SCF-PEKS schemesecure under the enhanced security model in the random oracle model. Another extension on SCF-PEKS is by Emura *et al.* [15]. They enhanced the security model by introducing the *adaptively secure* SCF-PEKS, wherein an adversary is allowed to issue test queries adaptively.

**3) Against Outside KGA***:* Byun *et al.* 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.* [5] was susceptible to keyword guessing attack. Inspired by the work of Byun *et al*, Yau *et al* 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 also showed off-line keyword guessing attacks against the (SCF-)PEKS schemes in [13] . The first PEKS scheme secure against outside keyword guessing attacks was proposed by Rhee *et al.* In the notion of trapdoor in distinguish ability was proposed and the authors showed that trapdoor in distinguish ability is a sufficient condition for preventing outside keyword-guessing attacks. Fang *et al.* [21] 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.

***4) Against Inside KGA****:* Nevertheless, all the schemes mentioned above are found to be vulnerable to keyword guessing attacks from a malicious server (i.e., inside KGA). Jeong *et al.* showed a negative result that the consistency/ correctness of PEKS implies insecurity to inside

KGA in PEKS. Their result indicates that constructing secure and consistent PEKS schemes against inside KGA is impossible under the original framework. A potential solution is to propose a new framework of PEKS. In [10], Peng *et al.* proposed the notion of Public-key Encryption with Fuzzy Keyword Search (PEFKS) where each keyword corresponds to an exact trapdoor and a fuzzy trapdoor. The server is only provided with the fuzzy trapdoor and thus

can no longer learn the exact keyword since two or more keywords share the same fuzzy keyword trapdoor. However, their scheme suffers from several limitations regarding the security and efficiency. On one hand, although the server cannot exactly guess the keyword, it is still able to know which small set the underlying keyword belongs to and thus the keyword privacy is not well preserved from the server. On the other hand, their scheme is impractical as the receiver has to locally find the matching ciphertext by using the exact trapdoor to filter out the non-matching ones from the set returned from the server. A new framework for peks In this section, we formally define the Dual-Server Public Key Encryption with Keyword Search (DS-PEKS) and itssecurity model.

**3 Definition of DS-PEKS**

DS-PEKS scheme mainly consists of *(*KeyGen, DS − PEKS, DS − Trapdoor, FrontTest, BackTest*)*. To be more precise, the KeyGen algorithm generates the public/ private key pairs of the front and back servers instead of that of the receiver. Moreover, the trapdoor generation algorithm DS − Trapdoor defined here is public while in the traditional PEKS definition [5], [13], the algorithm Trapdoor takes as input the receiver's private key. Such a difference is 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 ciphertext to recover the encrypted keyword. As a result, it is impossible to achieve the semantic security as defined in [5] 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 difference between the traditional PEKS and our proposed DS-PEKS is that the test algorithm is divided into two algorithms, FrontTest and BackTest run by two independent servers. This is essential for achieving security against the inside keyword guessing attack. In the DS-PEKS system, upon receiving a query from the receiver, the front server pre-processes the trapdoor and all the PEKS ciphertexts using its private key, and then sends some *internal testing-states* to the back server with the corresponding trapdoor and PEKS ciphertexts hidden. The back server can then decide which documents are queried by the receiver

using its private key and the received internal testing-states from the front server.

The formal definition of DS-PEKS is as follows.

Definition 1 (DS-PEKS): A DS-PEKS scheme is defined by

the following algorithms.

• Setup(1λ). Takes as input the security parameter λ, generates the system parameters P;

• KeyGen(P). Takes as input the systems parameters P, outputs the public/secret key pairs (pkFS, skFS), and (pkBS, skBS) for the front server, and the back server respectively;

• DS − PEKS(P, pkFS, pkBS, kw1). Takes as input P, the front server's public key pkFS, the back server's public key pkBS and the keyword kw1, outputs the PEKS ciphertext CTkw1 of kw1;

• DS − Trapdoor(P, pkFS, pkBS, kw2). Takes as input P, the front server's public key pkFS, the back server's public key pkBS and the keyword kw2, outputs the trapdoor Tkw2 ;

• FrontTest(P, skFS,CTkw1 , Tkw2 ). Takes as input P, the front server's secret key skFS, the PEKS ciphertext CTkw1 and the trapdoor Tkw2 , outputs the internal testing-state CIT S;

• BackTest(P, skBS,CIT S). Takes as input P, the back server's secret key skBS and the internal testing-state CIT S, outputs the testing result 0 or 1; Correctness. It is required that for any keyword kw1, kw2, and CTkw1 ← DS − PEKS(P, pkFS, pkBS, kw1), Tkw2← DS − Trapdoor(P, pkFS, pkBS, kw2), we have BackTest(P, skBS,CIT S) =_ 1 kw1 = kw2,0 kw1 _= kw2. where CIT S ←FrontTest(P, skFS,CTkw1 , Tkw2 ).

**3 Security Models**

In this subsection, we formalise the following security models for a DS-PEKS scheme against the adversarial front and back servers, respectively. One should note that both the front server and the back server here are supposed to be "honest but curious" and will not collude with each other. More precisely, both the servers perform the testing strictly following the scheme procedures but may be curious about the underlying keyword. We should note that the following security models also imply the security guarantees against the outside adversaries which have less capability compared to the servers.

**1) Adversarial Front Server***:* In this part, we define the security against an adversarial front server. We introduce two games, namely semantic-security against chosen keyword 4 *Semantic-Security Against Chosen Keyword Attack.* In the following, we define the semantic-security against

chosen keyword attack which guarantees that no adversary is able to distinguish a keyword from another one given the corresponding PEKS ciphertext. That is, the PEKS ciphertext does not reveal any information about the underlying keyword to any adversary. Formally, we introduce an experiment in Fig. 1 for the SS-CKA security definition against the adversarial front server. In the experiment, the adversary *A* is given the public/private key pair of the front server and the public key of the back server. In the find phase, *A* can test any pair of PEKS ciphertext and keyword by querying the oracle *OT* and eventually output two challenging keywords *(kw*0*, kw*1*)* with the hint information "state." With a random bit $b \in \{0, 1\}$ as input,

the experiment generates and then sends the PEKS ciphertext *CT∗ kw* of keyword *kwb* to *A*. During the guess phase, *A* can continue the query to *OT* and finally output its guess *b*_. The guess *b*_ is a valid output of the experiment if and only if that *A* has never queried *OT* with the challenge keywords.We refer to such an adversarial front server *A* in the above experiment as an SS-CKA adversary and define its advantage as AdvSS−CKA $_{FS,A}(\lambda) = \Pr[b = b\_] - 1/2$.

**2 Indistinguishability Against Keyword Guessing Attack**. This

security model captures that the trapdoor reveals no information

about the underlying keyword to the adversarial front

server. We define the security experiment as shown in Fig. 2.

The experiment is similar to that of SS-CKA experiment

except that in the challenge phase, the adversary is given the

trapdoor instead of the PEKS ciphertext.

We refer to such an adversarial front server *A* in the

above experiment as an IND − KGA adversary and define its

1In this paper, we use two different terms, namely semantic security and

indistinguishability, to define the security for the keyword ciphertext and the

trapdoor, respectively. However, as for normal public key encryption, these

two terms are equivalent.

Fig. 2. IND-KGA experiment for adversarial front server.

advantage as

AdvIND−KGA

$_{FS,A}(\lambda) = \Pr[b = b\_] - 1/2$.

*2) Adversarial Back Server:* The security models of

SS − CKA and IND − KGA in terms of an adversarial back

server are similar to those against an adversarial front server.

III. *Semantic-Security Against Chosen Keyword Attack.* Here

the SS − CKA experiment against an adversarial back server

is the same as the one against an adversarial front server except

that the adversary is given the private key of the back server

instead of that of the front server. We omit the details here

for simplicity. We refer to the adversarial back server *A* in

the SS − CKA experiment as an SS − CKA adversary and

define its advantage as

AdvSS−CKA

*BS,A (λ)* = Pr[*b* = *b*_] − 1/2.

IV. *Indistinguishability Against Keyword Guessing Attack.*

Similarly, this security model aims to capture that the trapdoor

does not reveal any information to the back server and hence

is the same as that against the front server except that the

adversary owns the private key of the back server instead

of that of the front server. Therefore, we also omit the details here. We refer to the adversarial back server *A* in

the IND − KGA experiment as an IND − KGA adversary and

define its advantage as

AdvIND−KGA

*BS,A (λ)* = Pr[*b* = *b*_] − 1/2. CONCLUSION

In this paper, we proposed a new framework, named

Dual-Server Public Key Encryption with Keyword Search

(DS-PEKS), that can prevent the inside keyword guessing

attack which is an inherent vulnerability of the traditional

PEKS framework. We also introduced a new Smooth Projective

Hash Function (SPHF) and used it to construct a generic

DS-PEKS scheme. An efficient instantiation of the new SPHF

based on the Diffie-Hellman problem is also presented in

the paper, which gives an efficient DS-PEKS scheme without

pairings.

ACKNOWLEDGEMENT

The authors thank the anonymous reviewers for their

invaluable feedback on this work.

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. Secur. Privacy (ACISP)*, 2015, pp. 59–76.

[2] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE Symp. Secur. Privacy*, May 2000, pp. 44–55.

[3] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2004, pp. 563–574.

[4] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," in *Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS)*, 2006, pp. 79–88.

798 IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 11, NO. 4, APRIL 2016

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

[6] R. Gennaro and Y. Lindell, "A framework for password-based authenticated key exchange," in *Proc. Int. Conf. EUROCRYPT*, 2003, pp. 524–543.

[7] B. R. Waters, D. Balfanz, G. Durfee, and D. K. Smetters, "Building an encrypted and searchable audit log," in *Proc. NDSS*, 2004, pp. 1–11.

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

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

[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, Nov. 2013.

[11] G. Di Crescenzo and V. Saraswat, "Public key encryption with

searchable keywords based on Jacobi symbols," in *Proc. 8th Int. Conf. INDOCRYPT*, 2007, pp. 282–296.

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

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

[14] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Improved searchable public key encryption with designated tester," in *Proc. 4th Int. Symp. ASIACCS*, 2009, pp. 376–379.

[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)*, 2006, pp. 75–83.

[17] W.-C. Yau, S.-H. Heng, and B.-M. Goi, "Off-line keyword guessing attacks on recent public key encryption with keyword search schemes," in *Proc. 5th Int. Conf. ATC*, 2008, pp. 100–105.

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

[19] H. S. Rhee, W. Susilo, and H.-J. Kim, "Secure searchable public key encryption scheme against keyword guessing attacks," *IEICE Electron. Exp.*, vol. 6, no. 5, pp. 237–243, 2009.

[20] 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," *J. Syst. Softw.*, vol. 83, no. 5, pp. 763–771, 2010.

**AUTHOR'S PROFILE:**

1.M.MADHAVI

Mekalamadhavi20@gmail.com

2. D.VENKATA SIVAREDDY HOD

lionshivareddy@gmail.com