ISSN:2348-2079

Volume-6 Issue-2

International Journal of Intellectual Advancements and Research in Engineering Computations

# Dual-server public-key encryption with keyword search for secure cloud storage

[1]**Sukumar P,** Department of Computer Science and Engineering, Velalar College of Engineering and Technology

[2]**Sharmila M,** [2]**Swathi G,**[2]**Ranjani C,** [2]**Nandhini G,**

Department of Computer Science and Engineering, Velalar College of Engineering and Technology

[1] Assistant Professor, [2-5]UG Students

Email id: [1]spsukumaran@gmail.com , [2] sharmilam2026@gmail.com, [2] psgswathi@gmail.com, [2]ranjanichandran2@gmail.com, [2] nandhinigopal6@gmail.com.

**ABSTRACT:**

Searchable encryption is increasing interest for protecting the data privacy in secure searchable cloud storage. 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, a new PEKS framework named dual-server PEKS (DS-PEKS). As another main contribution, a new variant of the smooth projective hash functions (SPHFs) referred to as linear and homomorphic SPHF (LH-SPHF). To show a generic construction of secure DS-PEKS from LH-SPHF. To illustrate the feasibility of new framework, 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.

**KEYWORDS**

Guessing attack, Homomorphic, Diffie-Hellman

**INTRODUCTION:**

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers.

Cloud computing is a typical example in the emerging trends of Information Technology.It enables seamless access to shared pool of configurable system resources.It can be easily managed with minimal effort.It provides security and accessibility simultaneously. Cloud computing involves the use of computing resources to deliver services over network.It entrusts remote services with user's data , software and computation.The data which was stored under a third party system was vulnerable to theft.It requires an encryption to protect the data and with the traditional PEKS, it was still under siege.It prompted to deliver more secured way to store the data in the cloud, resulted in the creation of DS-PEKS.

**RELATED WORK**

Classification of PEKS is described based on their security. To 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. To show that the PEKS schemes based on bilinear map could be applied to build encrypted and searchable auditing logs. In orderconstruct a PEKS secure in the standard model, we proposed a scheme based on the k-resilient IBE and also gave a construction supporting multiple-keyword search. The first PEKS scheme without pairings. Secure Channel Free PEKS: The original PEKS scheme requires a secure channel to transmit the trapdoors. To overcome this limitation, 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. 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.We 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 was susceptible to keyword guessing attack demonstrated that outside adversaries that capture the trapdoors sent in a public channel can reveal the encrypted keywords through offline keyword guessing attacks and they so showed off-line keyword guessing attacks against the CF-)PEKS schemes. The first PEKS scheme secure against outside keyword guessing attacks was proposed, the notion of trapdoor was proposed and there is a sufficient condition for preventing outside keyword-guessing attacks. We proposed a concrete SCF-PEKS scheme with (outside) KGA resilience. They also considered the adaptive test oracle in their proposed security definition.

**EXISTING SYSTEM:**

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 = EK (F, KW)

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 (KW1, KW2… KWn) – 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. 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.Baek*et al.* proposed a ew PEKS scheme without requiring a secure channel, which is referred to as a secure channel-free PEKS (SCF-PEKS).security model for SCF-PEKS where the attacker is allowed to obtain the relationship between the non-challenge ciphertexts and the trapdoor.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.Despite of being free from secret key distribution, PEKS schemes suffer from an inherent insecurity regarding the trapdoor keyword privacy, namely *inside Keyword Guessing Attack* (KGA). The reason leading to such a security vulnerability is that anyone who knows receiver's public key can generate the PEKS ciphertext of arbitrary keyword himself. Specifically, given a trapdoor, the adversarial server can choose a guessing keyword from the keyword space and then use the keyword to generate a PEKS ciphertext. 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.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.

**PROPOSED SYSTEM:**

A new PEKS framework named *Dual-Server Public Key Encryption with Keyword Search* (DS-PEKS) to address the security vulnerability of PEKS.A new variant of *Smooth Projective Hash Function* (SPHF), referred to as *linear and homomorphic SPHF*, is introduced for a generic construction of DS-PEKS.To illustrate the feasibility of new framework, an efficient instantiation of our SPHF based on the Diffie-Hellman language. All the existing schemes require the pairing computation during the generation of PEKS ciphertext and testing and hence are less efficient than our scheme, which does not need any pairing computation.It is because that scheme does not include pairing computation. Particularly, the existing scheme requires the most computation cost due to 2 pairing computation per PEKS generation.In our scheme, although we also require

another stage for the testing, our computation cost is actually lower than that of any existing scheme as we do not require any pairing computation and all the searching work is handled by the server.

**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 (PUBS, PRBS, PUFS, PRFS) instead of that of the receiver. Moreover, the trapdoor generation algorithm, DS - Trapdoor defined here, is public, while, in the traditional PEKS definition, 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 . 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.

Secure Keyword Search with Public Key Encryption by Cloud storage

a. Key generation Algorithm: public key-private key pair of the front server (PUFS, PRFS) and back server(PUBS,PRBS), can be calculated using system parameter.

b. Dual server-PEKS algorithm: cipher text CTKW1, is generated using the input parameters, the front server's public key PUFS, the back server's public key PUBS and the keyword KW1.

c. Dual server – Trapdoor algorithm: The trapdoor TKW2, is generated using the input parameter, that are front server's public key PUFS , the back server's public key PUBS and the keyword KW2.

d. Front server TestingAlgorithm: the internal testing-state CITS, is generated using the input parameters P , the front server's private key PRFS , the PEKS ciphertext CTKW1 and the trapdoor TKW2.

e. Back server Testing Algorithm: This algorithm outputs the testing results either 0 or 1 by using the back server's private key PRBS and the internal testing-state CITS.

**SYSTEM ARCHITECTURE:**

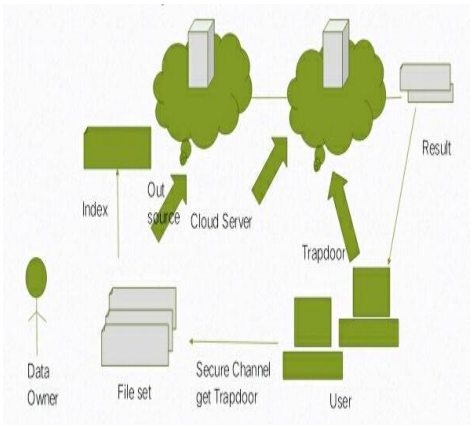



Fig.1

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.The user can retrieve the data from server by using trapdoor. Once trapdoor gets keyword from user, encrypts it and the server analyses the keyword under the ciphertext is same as selected by the receiver.The server sends the matching encrypted data to the receiver.

**MODULES:**

i) System Construction Module
ii)Semantic-Security against Chosen Keyword Attack
iii)Front Server
iv)Back Server

**MODULES DESCSRIPTION:**

**i)System Construction Module**

In the first module, we develop the system with the entities required to provde our system. 1) Cloud User: the user, who can be an individual or an organization originally storing their data in cloud and accessing the data. 2) Cloud Service Provider (CSP): the CSP, who manages cloud servers (CSs) and provides a paid storage space on its infrastructure to users as a service. We propose a new framework, namely DS-PEKS, and present its formal definition and security models. We then define a new variant of smooth projective hash function (SPHF). A generic construction of DS-

PEKS from LH-SPHF is shown with formal correctness analysis and security proofs. Finally, we present an efficient instantiation of DS-PEKS from SPHF.

**ii)Semantic-Security against Chosen Keyword Attack**

In the module, we develop 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.

**iii)Front Server:**

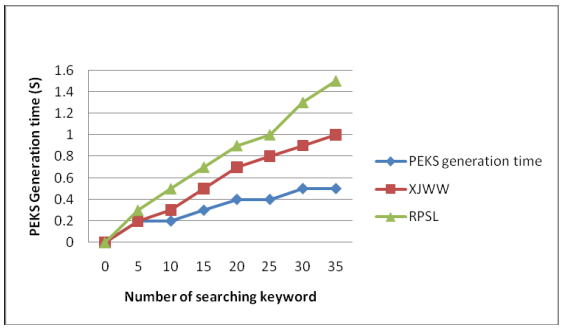
After receiving the query from the receiver, the front server pre-processes the trapdoor and all the PEKS ciphertexts using its private key, and then sends some internal testing-states to the back server with the corresponding trapdoor and PEKS ciphertexts hidden.

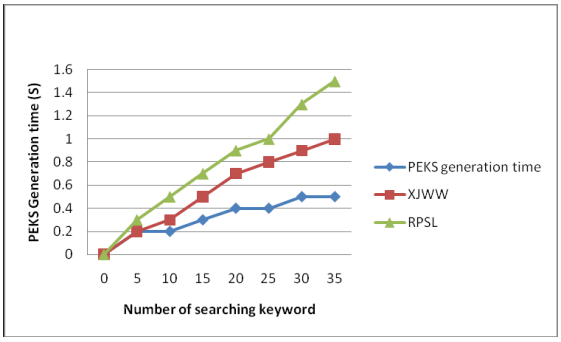
**iv)Back Server:**

In this module, 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.

## RESULTS

Performance is evaluated by making the comparison between existing schemes and our scheme in terms of computation, size and security. All the existing system require the pairing computation during the generation of PEKS cipher text and testing. Hence, these schemes are less efficient than our scheme. Because our method does not need any pairing computation. In our scheme, the computation cost of PEKS generation and testing are calculated.




When the searching keyword number is 30, the total computation cost of our scheme is about 0.5 seconds. As illustrated in Fig, the scheme [10] cost the mosttime due to an additional pairing computation in the exact testing stage. One should note that this additionalpairing computation done on the user side instead of the server. Therefore, it could be the computation burdenfor users who may use a light device for searching the data.

In our scheme, it also requires another stage for the testing but our computation cost is actually lower than that of any existing scheme.Our scheme does notrequire any pairing computation and all the searching work is handled by the server.

## CONCLUSION:

**I**n 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.

## REFERENCE:

**[**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 PilJoong 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, OdedGoldreich, and ShaiHalevi, "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.