# A NEW FRAMEWORK FOR SECURE SEARCHABLE CLOUD STORAGE

**Rubeena A. Attar[1] , Shivaputra S. Panchal[2]**

[1]*M.Tech Student, Department of Computer Science and Engineering, BLDEA's V.P. Dr.P.G.Halakatti College of Engineering & Technology Vijayapur, Karnataka, India*

[2] *Assistant Professor, Department of Computer Science and Engineering, BLDEA's V.P. Dr.P.G.Halakatti College of Engineering & Technology Vijayapur, Karnataka, India*

---------------------------------------------------------------------***---------------------------------------------------------------------

**Abstract** – *Today Cloud storage has become popular technology for data storage. However end users will not completely trust the cloud. So they encrypt and store the data. So, essential information should be encrypted before outsourcing for privacy concerns, but this makes difficult for the users of the cloud to utilize it. One technique is searchable encryption where user can search encrypted data in the cloud by using well known scheme called Public key Encryption with keyword search (PEKS) but one drawback for this system is it bears from Keyword Guessing Attack (KGA). To address this security, we suggest a new version of PEKS structure named as Dual-Server Public Key Encryption with Keyword Search (DS-PEKS) that overcome this security issue of keyword guessing by attackers. To show security of this new framework we show its structure from Smooth Projective Hash Functions (SPHF) which uses hashing technique. We show security of DS-PEKS scheme and how it can accomplish the solid security against inside KGA. It uses two servers that is frontserver and backserver for providing high and strong security.*

***KeyWords***: **Keyword Search, PEKS, Secure Cloud Storage, Encryption,Inside Keyword Guessing Attack, Smooth Projective Hash Function.**

## 1. INTRODUCTION

Today there is a growing popularity of cloud computing, large number of users and data owners are motivated to store their data to cloud servers for large convenience and reduced cost required for data management. Also it provides scalable data storage and computational services. In other words, customers can remotely outsource large amount of data and workloads to the cloud and benefit from unlimited computing resources and applications in the on-demand high-quality services. Relieving the burden for storage management, accessing to the data independent of locations, reducing in capital expenditure on hardware, software and staff are among the advantages of cloud computing brings to its users. The data can be accessed from anywhere by using their credentials created at the time of registration to the cloud. By storing their data into the cloud, the data owners can be relieved from the burden of data storage and maintenance so as to enjoy the on-demand high quality data storage service. However, the fact that data owners and cloud server are not in the same trusted domain may put the outsourced data at risk, as the cloud server may no longer be fully trusted in such a cloud due to a number of reasons: the cloud server may leak data information to unauthorized bodies or be attacked. It follows that sensitive data generally should be encrypted before outsourcing for data confidentiality. To make the most of these data, Searchable encryption is one technique for protection of the information. Searchable encryption is a cryptography primitive that enables users to search through outsourced encrypted data without exposing keywords to the untrusted server. Public Key Encryption with Keyword Search (PEKS) is such a method but it suffers from inside Keyword Guessing Attack (KGA). KGA attack is offline attack that is unauthorized access from authorized users. We will provide a new scheme of PEKS called as Dual-Server Public Key Encryption with Keyword Search (DS-PEKS) as to overcome this issue in the cloud.

## 2. RELATED WORK

Several works have been done on how to search the encrypted data from cloud.

In 2000, to achieve this task, Song et al. [1] first proposed the notion of searching the encrypted data with certain words. It was the first searchable symmetric encryption scheme. In that there are two ways to search on the ciphertext, which is to build up an index for each word W and perform a sequential scan without an index. In another one do not need second space to store the index, but it is slower. Several other Searchable Symmetric Encryption (SSE)schemes and afterwards a few SSE plans [2], [3] were proposed for enhancements.

As a result, Boneh et al. further proposed a new scheme that searches the encrypted data based on keyword [4]. It was the foremost asymmetric searchable encryption planned by Boneh et al. [4], public key encryption with keyword Search. This algorithm is able to detect which encrypted outsourced file has a specific keyword without letting other parties such as Cloud Service providers and unauthorized users to learn anything throughout search and retrieving process.

In [5] Baek et al. who improved PEKS scheme into a secure channel (SSL) free PEKS scheme (SCF-PEKS) which eliminates an idea of secure channel (SSL) between users and a server. In SCF-PEKS scheme, the data owner uses the server's public key and receiver's public key to encrypt the keywords each time he stores the encrypted information to the server. Whenever a receiver or data user wants to search the encrypted data linked with a specific keyword, the data user can send the trapdoor (queried keyword) to get the data via a public network since only the server has the matching private key which can test whether the PEKS ciphertext matches the trapdoor. However, the trapdoors can be captured by the outside attackers can derive the embedded keyword because the trapdoor transferred in the public network.

In 2006, Byun et al. [6] pointed out that PEKS might be attacked by the off-line keyword-guessing attacks. Since keywords are chosen over significantly much smaller space than passwords and users usually use familiar keywords (low entropy) for seeking information [6]. Therefore, attackers can capture the trapdoor and have chance to presume keyword.

In 2008, Yau et al. [7] also demonstrated that outside attackers that capture the trapdoors sent in a public channel can reveal encrypted keywords by performing offline keyword guessing attacks.

In 2013 , Xu et al. [8] proposed a novel concept called public-key encryption with fuzzy keyword search (PEFKS), by which the un-trusted server only attains the fuzzy search trapdoor as a replacement for of the exact search trapdoor, and define its semantic security under chosen keyword attack (SS-CKA) and indistinguishability of keywords under non-adaptively chosen keywords and keyword guessing attack. PEFKS is the first scheme to resist against keyword guessing.

In 2010, Rhee et al. [9] studied a secure searchable public-key encryption scheme with a designated tester (dPEKS). They enhanced the existing security model to incorporate the realistic abilities of dPEKS attackers and introduced t "trapdoor indistinguishability" This scheme is the first dPEKS method that is secure against keyword-guessing attacks.

## 3. EXISTING SYSTEM

In present systems, in spite of of being open from secret key sharing, 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 liability is that anyone who knows receiver's public key can produce the PEKS ciphertext of random 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 continued until the right keyword is matched. Such a guessing attack has also been considered in several 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), which has a much smaller size than a password dictionary (e.g., all the words containing 6 alphanumeric characters). It is worth noting that in SSE schemes, only secret key holders can generate the keyword ciphertext and hence the adversarial server is not able to launch the inside KGA. As the keyword always specifies the privacy of the user data, it is therefore of most important to overcome this security threat for securing searchable encrypted data outsourcing.

## 4. PROPOSED SYSTEM

In the conventional 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. Another difference between the traditional PEKS and our proposed DS-PEKS is that the test algorithm is divided into two algorithms, FrontTest and BackTest controlled 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 come to a decision to which documents are queried by the receiver using its private key and the received internal testing-states from the front server.

Formally DS-PEKS is defined as follows.

A DS-PEKS primarily comprises of (KeyGen, DSPEKS, DS-Trapdoor,FrontTest, BackTest).

1. Setup ($1^{\rho}$)
   inputs is the security parameter $\rho$ to generates the system parameters K.

2. KeyGen(K)
   Inputs the systems parameters K that outputs the public/secret key (pk/sk) pairs for the front server(FS), and the back server(BS).

3. DSPEKS(K, $pk_{FS}$, $pk_{BS}$, kw1)
   Inputs K, the front server's public key, the back server's public key and the keyword kw1, outputs the PEKS ciphertext $CT_{kw1}$ of kw1.

4. DS-Trapdoor(K, $pk_{FS}$, $pk_{BS}$, kw2)
   Inputs K, the front server's public key pkF S, the back server's public key $pk_{BS}$ and the keyword kw2, outputs the trapdoor $T_{kw2}$.

5. FrontTest(K, $sk_{FS}$, $CT_{kw1}$, $T_{kw2}$)
   Inputs K, the front server's secret key $sk_{FS}$, the PEKS ciphertext $CT_{kw1}$ and the trapdoor $T_{kw2}$, outputs the internal testing-state ITS.

6. BackTest(K, $sk_{BS}$, ITS)
   Inputs K, the backserver's secret key $sk_{BS}$ and the internal testing-state ITS, outputs the testing result 0 or 1.

$$BackTest\ (P, skBS, CITS) = \begin{cases} 1, & kw1 = kw2, \\ 0, & kw1 \neq kw2. \end{cases}$$

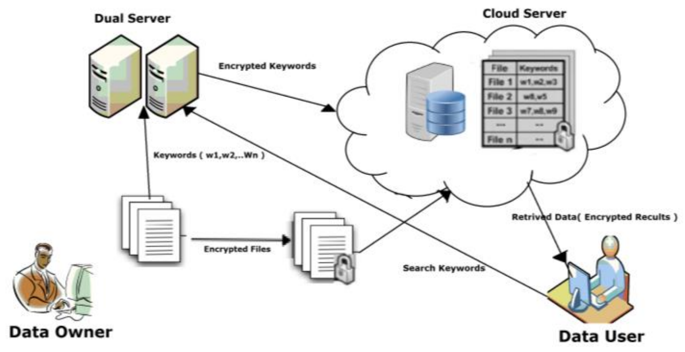

**Fig -1**: Dual-Server Architecture

The figure 1 shows the architecture of the new proposed scheme DS-PEKS which have two servers frontserver and backserver. There are three main modules:

1. **Data Owner**: Register with cloud server and login (username must be unique). Send request to the cloud admin. Browse file and generate keywords for the filr and then request content key to encrypt the data, Upload data to cloud server. These keywords will be sent to the cloud.
2. **Data User**: Register with cloud server and login (username must be unique). Send request to the cloud admin. Login and search by entering user's choice keyword. This keyword will be sent to the dual server.
3. **Dual Server**: Frontserver and backserver encrypt the keywords sent by the user and owner using their public keys simultaneously. Next frontserver will do testing on user keyword using its private key and send these testings to the backserver. Then using private key the backserver will test these testing and return results to the queried user, that is in brief,

**Front Server:**
After getting the query keyword from the receiver, the front server pre-processes the trapdoor and all the PEKS ciphertexts using its private key, and then forwards some internal testing-states to the backserver with the corresponding trapdoor and PEKS ciphertexts hidden.

**Back Server:**
In this module, the backserver makes a decision that which documents are queried by the receiver using its private key and the received internal testing-states from the front server. Also Given a searchable encryption of the keyword w' by user and a trapdoor for w by owner, the server should be able to find out all messages having keyword w' (if w' = w) and learn nothing more about the keywords . Also, the server shouldn't learn anything about the encrypted information itself.

## 5. Smooth Projective Hash Functions (SPHF)

The DS-PEKS is constructed using smooth projective hash function (SPHF), an idea given by Cramer and Shoup [10].

Formally, SPHF is defined over language L as <Hash, ProjHash> containing keys (hk,hp);

hk is hashing key(a private key) hp is projection key(public key).

SPHF is defined as:

1. HashKG (L): generates a hashing key hk for the language L

2. ProjKG (hk,L): generates the projection key hp from the hashing key hk

3. Hash (hk, L, W): outputs the hash value of the word W from the hashing key hk;

4. ProjHash (hp,L;W,w): outputs the hash value of the word W from the projection key hp, and the witness w that W∈ L.

## 6. ADVANTAGES OF DS-PEKS SYSTEM

- Confidentiality
  The private records of users should be kept in secrecy from both unauthorized system users and attackers. So our system does not disclose any information to the unauthorized systems. Even the searched keywords of users are kept confidential by encrypting it into server side. All the attached keywords of the file are also encrypted.

- Secure model
  Our work uses dual server model and have larger security alignment. It provides two way

authentications by double encryption of the keywords at frontserver and backserver. It uses only servers public and private keys.

- No need of sharing of keys
  No concept of the encryption keys and the decryption keys are generated separately for multi-client setting. Every user who uploads the file will encrypt it using one special key called content key which will be accessible by the user when he retrieves the results from the server.

- Abide offline KG attacks:
  Our work fights against offline guessing of keyword attacks and offers higher efficiency. No authorized user can able to guess the keywords because we are also encrypting the keywords in our new scheme.

- When it comes to trapdoor generation, our scheme don't involve pairing computation, the computation price is reduced compared to PEKS generation. No users keys are required which reduces cost of computation and no channel required between user and receiver.

## 7. CONCLUSION AND FUTURE WORK

The existing techniques on keyword-based encryption, which are widely used on the plaintext data, cannot be directly applied on the encrypted data. Downloading all the data from the cloud and decrypt locally is obviously impractically .In order to improve the security issues in the cloud environments; we proposed an efficient scheme that prevents inside keyword guessing attack known as Dual-Server Public Key Encryption with Keyword Search (DS-PEKS) via smooth projective hash functions. Our proposed system is efficient and cost effective.

As a future work, the project can be extended to reduce high cost of computation for executing trapdoors and ciphertexts and we can add any new algorithms to provide more security. As an extension file content key can be retrieved through receiver's mail id.

## REFERENCES

1. D. W. Dawn, X. Song and A. Perrig, "Practical techniques for searches on encrypted data," in Security and Privacy, 2000. S P 2000. Proceedings. 2000 IEEE Symposium on, pp. 44 –55, 2000.
2. D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searcheson encrypted data," in Proc. IEEE Symp. Secur. Privacy, May 2000,pp. 44–55.
3. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preservingencryption for numeric data," in Proc. ACM SIGMOD Int. Conf. Manage.Data, 2004, pp. 563–574.
4. D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano."Public key encryption with keyword search," in Advances in Cryptology-Eurocrypt 2004,pp. 506-522.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption," in Proc. Int.Conf. EUROCRYPT, 2002, pp. 45–64.

## BIOGRAPHIES

M.Tech Student, Department of Computer Science and Engineering, BLDEA's V.P. Dr. P.G.Halakatti College of Engineering & Technology Vijayapur, Karnataka, India

Working as an Assistant Professor, Department of Computer Science and Engineering, BLDEA's V.P. Dr. P.G. Halakatti College of Engineering & Technology Vijayapur, Karnataka, India.