Dual-Server Public-Key Encryption with Keyword Search using Smooth Projection Hashing for Secure Storage on Cloud

Veligatla Suneel Goud* & Kompella Venkata Ramana**
*M.Tech Scholar, Dept. of Computer Science and Systems Engg. Andhra University, Visakhapatnam, India.
**Professor, Dept. of Computer Science and Systems Engg. Andhra University, Visakhapatnam, India.

Received Jan. 07, 2018 Accepted Feb. 05, 2018

ABSTRACT *Now a day's cloud computing has become one of the fascinating domains which was used by almost all MNC and IT companies. Generally this is formed by interconnecting a large number of systems connected all together for remote servers hosted on internet to store, access, retrieve data from remote machines not from local machines. As the cloud server has the capability to store a lot of valuable data on its memory block, a lot of users can connect with the centralized location to access, retrieve and modify the data which is stored on the cloud server. Till now there was no mechanism available to store the data in an encrypted manner in all public clouds and even private clouds. In this proposed work, we mainly try to examine the security level 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, the traditional PEKS mainly suffer with a problem like inside keyword guessing attack (KGA) launched by the malicious server. We formalize a new PEKS framework named Dual-Server Public Key Encryption with Keyword Search (DS-PEKS) to address the security vulnerability of PEKS.By conducting various experiments on our proposed approach, we finally, came to an conclusion that our proposed approach is best suited to provide security for the data which is stored inside the cloud server and also to provide secure keyword search for this sensitive data.*

***Key words**: Centralized Location Encrypted Manner, Cryptography Primitive, Keyword Search, Guessing Attack.*

1. INTRODUCTION

As we all know that in recent days there was a lot of user's attention towards the cloud data storage for storing and retrieving the data to and from the cloud server. As the data is been increasing day by day almost all the companies are unable to store their valuable data on their own individual devices, so in this situation they opt for a new data storage area known as Cloud Data Storage [1], [2]. Generally cloud service providers allow the users to access their services for a low economical and ascendable marginal cost compared with primitive data storage services. Generally the data which is stored in the cloud server is mainly used for sharing within the users of same group or between the users of different group with a valid authentication. Some of the best cloud data storage services are as follows: Google Drive, DriveHq Server, DropBox and ICloud. As these all are the best among various types of cloud service providers in which the data can be stored either in public cloud or private cloud, sometimes can be stored in both combine known as Hybrid Cloud.

2. MAIN CONTRIBUTIONS FOR DOING THIS PROPOSED WORK

The main contribution for doing this proposed paper contains four main reasons like:

1. Initially we try to propose a Novel PEKS framework named Dual-Server *Public Key Encryption with Keyword Search with Smooth Projective Hashing* (PEKS-SPH) to address the security vulnerability of primitive PEKS which is already proposed in the literature.
2. Design a novel variant of *Smooth Projective Hash Function* (SPHF), referred to as *linear and homomorphic SPHF.*
3. Next we try to show a generic construction of PEKS using the proposed Lin-Hom SPHF.
4. To illustrate the feasibility of this novel framework, an efficient instantiation of Linear-Homomorphic SPHF.

2.1 BACKGROUND WORK

In this section we will mainly discuss about the background work that was carried out in order to prove the performance of our proposed Dual Server Public Key Encryption with Keyword Search (PEKS-SPHF)

2.1.1 MAIN MOTIVATION

In this section we will initially try to find out the system model and assumptions that were used in the current paper. Now let us look about them in detail:

Traditional PEKS: A well known authors like Boneh[16], and another well known author like Abdalla [17] mainly constructed the anonymous IBE (also known as AIBE) and they try to design a novel searchable encryption from AIBE.In this study they mainly try to construct a hierarchical IBE (HIBE) scheme into a well known public key encryption with temporary keyword search (PETKS).In order to construct a PEKS model, another well known author like Khader also proposed a scheme based on the k-resilient IBE and also gave a construction supporting multiple-keyword search.As the traditional PEKS is best suited for providing security for the data but failed in handling the access policies for various users within the data storage.

Secure Channel Free PEKS: This is an enhanced version for the primitive PEKS model, where the primitive PEKS scheme requires a secure channel to transmit the trapdoors. To overcome this limitation,a well known author like Baek et al. try to 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. later enhanced Baek et al.'s security model for SCF-PEKS where the attacker is allowed to obtain the relationship between the non-challenge ciphertexts and the trapdoor. They also presented an SCF-PEKS scheme secure under the enhanced security model in the random oracle model. Another extension on SCF-PEKS is by Emura et al. They enhanced the security model by introducing the adaptively secure SCF-PEKS, wherein an adversary is allowed to issue test queries adaptively.

3. PROPOSED NOVEL DUAL-SERVER PUBLIC KEY ENCRYPTION WITH KEYWORD SEARCH USING SMOOTH PROJECTION HASHING (DS-PEKS)

In this section we will find out the proposed novel Dual-Server Public Key Encryption with Keyword Search (DS-PEKS) protocol that was used in current thesis in order to give high level of security for the sensitive data which is stored and accessed to and from the cloud server.

PRELIMINARY KNOWLEDGE

A Public Key Encryption with Keyword Search using smooth projection hashing scheme mainly consists of following attributes like

1. **KeyGen,**
2. **DS – PEKS**
3. **DS – Trapdoor**
4. **FrontTest**
5. **BackTest**

In order to discuss about these attributes more precisely, initially we discuss about the KeyGen algorithm which 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 ,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 [18]. 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.

- Setup(1^{λ}). *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* (pk_{FS}, sk_{FS}), *and* (pk_{BS}, sk_{BS}) *for the front server, and the back server respectively;*
- DS – PEKS($P, pk_{FS}, pk_{BS}, kw_1$). *Takes as input* P, *the front server's public key* pk_{FS}, *the back server's public key* pk_{BS} *and the keyword* kw_1, *outputs the PEKS ciphertext* CT_{kw_1} *of* kw_1;
- DS – Trapdoor($P, pk_{FS}, pk_{BS}, kw_2$). *Takes as input* P, *the front server's public key* pk_{FS}, *the back server's public key* pk_{BS} *and the keyword* kw_2, *outputs the trapdoor* T_{kw_2};
- FrontTest($P, sk_{FS}, CT_{kw_1}, T_{kw_2}$). *Takes as input* P, *the front server's secret key* sk_{FS}, *the PEKS ciphertext* CT_{kw_1} *and the trapdoor* T_{kw_2}, *outputs the internal testing-state* C_{ITS};
- BackTest(P, sk_{BS}, C_{ITS}). *Takes as input* P, *the back server's secret key* sk_{BS} *and the internal testing-state* C_{ITS}, *outputs the testing result 0 or 1;*

Correctness. It is required that for any keyword kw_1, kw_2, and $CT_{kw_1} \leftarrow$ DS – PEKS($P, pk_{FS}, pk_{BS}, kw_1$), $T_{kw_2} \leftarrow$ DS – Trapdoor($P, pk_{FS}, pk_{BS}, kw_2$), we have

$$\text{BackTest}(P, sk_{BS}, C_{ITS}) = \begin{cases} 1 & kw_1 = kw_2, \\ 0 & kw_1 \neq kw_2. \end{cases}$$

In the PEKS-SPH 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.

4. SMOOTH PROJECTIVE HASH FUNCTION (SPHF)

This is the main element for the construction of dual-server public key encryption with keyword search and the notion is represented as *smooth projective hash function* (SPHF),by two well known authors like Cramer and Shoup [19]. Here we can discuss about the original definition of an SPHF in the below paragraphs.

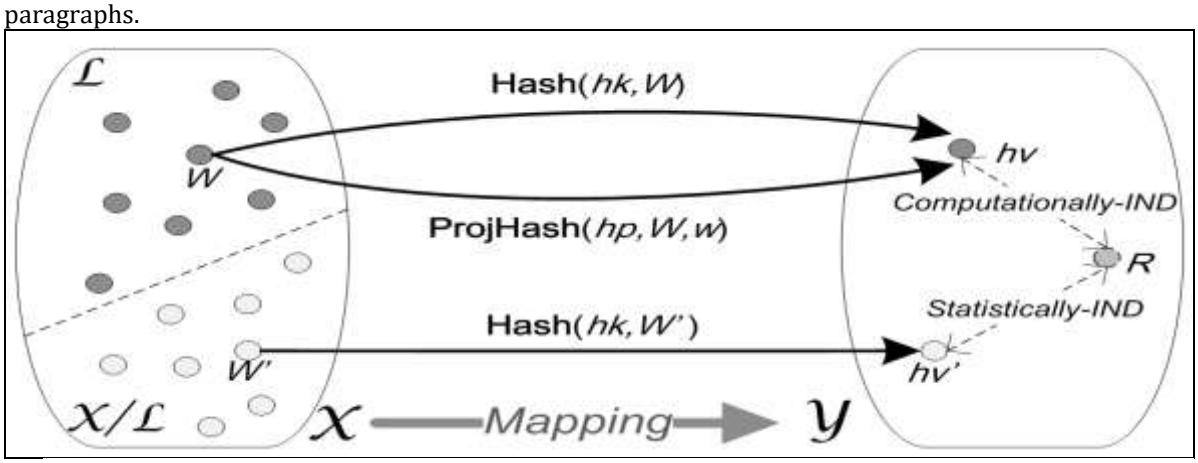


FIGURE.1. REPRESENTS THE ARCHITECTURE OF AN SMOOTH PROJECTIVE HASH FUNCTION (SPHF)

From the above figure 3,we can clearly represent the architecture flow of an SPHF.Here we assume the domain with X and an NP language problem with L, where L contains a subset of the elements of the domain X, i.e., $L \subset X$. Formally, an SPHF system over a language $L \subset X$, onto a set Y, is defined by the following five attributes:

They are as follows:

SPHFSetup (1λ): generates the global parameters param and the description of an NP language instance L.
HashKG$(L,$ param$)$: generates a hashing key hk for L.
ProjKG$($hk, $(L,$ param$))$: derives the projection key hp from the hashing key hk.
Hash$($hk, $(L,$ param$),W)$: outputs the hash value hv $\in Y$ for the word W from the hashing key hk.
ProjHash$($hp, $(L,$ param$),W,w)$: outputs the hash value hv1 $\in Y$ for the word W from the projection key hp and the witness w for the fact that $W \in L$.
The *correctness* of an SPHF requires that for a word $W \in L$ with w the witness,

$$\text{Hash}(\text{hk}, (L, \text{param}), W) = \text{ProjHash}(\text{hp}, (L, \text{param}), W, w).$$

Another property of SPHFs is *smoothness*, which means that for any $W \in X \backslash L$, the following two distributions are statistically indistinguishable:

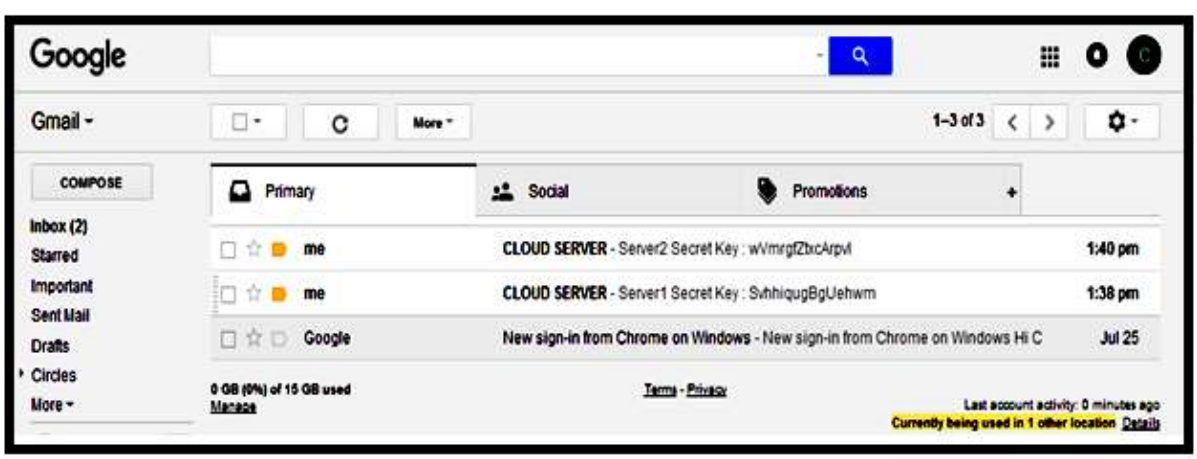
$V1 = \{(L, \text{param}, W, \text{hp}, \text{hv}) | \text{hv} = \text{Hash}(\text{hk}, (L, \text{param}), W)\}$,
$V2 = \{(L, \text{param}, W, \text{hp}, \text{hv}) | \text{hv} \$\leftarrow Y\}$,

In summary, an SPHF has the property that the projection key uniquely determines the hash value of any word in the language L but gives almost no information about the hash value for any point in $X \setminus L$.

5. RESULT ANALYSIS

In this section we mainly describe about the result analysis at the end of our application. Here we can see the server window that clearly represents that server can view all the file details along with set of user details and also the requests that was raised by the end users. Here the users can connect to this centralized server in order for accessing the files to and from the cloud server.

From the below window we can clearly find out that the data user will get two access keys from the servers in order for making the file downloaded into the PC in a plain text manner.For this he need to request the two servers individually and if the two servers front server and back server gives approval for data download then only he can download the data in a lain text manner.If any of the key is not received for the data user,he/she cant able to access the data and they cant able to view the data in a plain text manner.



6. CONCLUSION

In this paper, we for the first time have proposed a novel framework, named Dual-Server Public Key Encryption with Keyword Search (DS-PEKS), that can mainly protect the sensitive data which is stored inside the server space from the inside keyword guessing attack which is an inherent vulnerability of the traditional PEKS framework. Here we mainly proposed a novel Smooth Projective Hash Function (SPHF) to

construct the unique keys for the end users from the two servers dynamically without having a chance of creating duplicate keys for the end users. By conducting various experiments on our proposed DS-PEKS algorithm, our comparison results clearly tell that our proposed approach is best in providing security for the sensitive data which is stored inside the server space.

7. REFERENCES

1. C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing," IEEE Trans. Services Computing, vol. 5, no. 2, pp. 220-232, Apr.-June 2012.
2. X. Huang *et al.*, "Cost-effective authentic and anonymous data sharing with forward security," *IEEE Trans. Comput.*, vol. 64, no. 4,pp. 971–983, Apr. 2015.
3. K. Liang, J. K. Liu, D. S. Wong, and W. Susilo, "An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing," in *Proc. 19th ESORICS*, 2014,pp. 257–272.
4. J. K. Liu, M. H. Au, W. Susilo, K. Liang, R. Lu, and B. Srinivasan, "Secure sharing and searching for real-time video data in mobile cloud," *IEEE Netw.*, vol. 29, no. 2, pp. 46–50, Mar./Apr. 2015.
5. C.-K. Chu, W.-T. Zhu, J. Han, J.-K. Liu, J. Xu, and J. Zhou, "Security concerns in popular cloud storage services," *IEEE Pervasive Comput.*,vol. 12, no. 4, pp. 50–57, Oct./Dec. 2013.
6. Raul Isea The Present-Day Meaning Of The Word Bioinformatics, Global Journal of Advanced Research, 2015.
7. Ilzins, O., Isea, R. and Hoebeke, J. Can Bioinformatics Be Considered as an Experimental Biological Science 2015
8. *Ehrlich, M; Wang, R. (19 June 1981). "5-Methylcytosine in eukaryotic DNA". Science.**212** (4501): 1350–1357*
9. T. Jiang, X. Chen, J. Li, D. S. Wong, J. Ma, and J. Liu, "TIMER: Secure and reliable cloud storage against data re-outsourcing," in *Proc. 10th Int.Conf. ISPEC*, 2014, pp. 346–358.
10. 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.
11. 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.
12. 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.
13. 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.
14. 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.
15. C. Cocks, "An identity based encryption scheme based on quadratic residues," in *Cryptography and Coding*. Cirencester, U.K.: Springer,2001, pp. 360–363.
16. 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
17. 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.
18. 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.
19. 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.

By giving people the power to share, we're making the world more transparent.

~ Mark Zuckerberg