International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

Securing Cloud Storage Dual-Server Using Public-Key Encryption with Keyword Search

Veligatla Suneel Goud & Prof. Kompella Venkata Ramana
[#1] M.Tech Scholar , Department of Computer Science and System Engineering, Andhra University College of Engineering (A), Visakhapatnam, AP, India.
[#2] Professor, Department of Computer Science and System Engineering, Andhra University College of Engineering (A), Visakhapatnam, AP, India.

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.

As we all know that there are many applications of cloud computing, such as data sharing for remote systems [3], [4], [5], [6], data storage from a remote systems to a centralized location [7], [8], [9], big data management systems [10], medical information system etc.All the cloud users try to access cloud-based applications or cloud server through a web browser to store or access the data to and from the cloud server. There are several benefits of web-based cloud computing services like the ease of accessibility, reduced storage costs and on time data supply. Although they are many principles that govern the principle of cloud computing, still it provides great advantages especially in terms of security and privacy. As we all know that sensitive data will be reside in the cloud server for sharing and access to and from the remote access, the major issue that arise in the cloud based services is authentication of the cloud server. Initially the cloud user or end user need to register into the cloud server and then once he/she got registered, then the user should substitute his valid credentials for login into the system for various applications and services access. During this login into the cloud, the two main problems that arise in the traditional cloud based systems is account/password based authentication is not strictly privacy preserving, the second mainly problem that arise in the primitive cloud based services is as we all know that the data from the cloud server will be accessed from different people from different locations and hence it may be very easy for a hacker to install some spyware software to learn the login password in any way from the web browser.

In our proposed work searchable encryption can be realized in either symmetric or asymmetric encryption setting. In [10], Song et al. designed a keyword search on ciphertext, known as Searchable Symmetric Encryption (SSE) and later they were many SSE schemes [11], [12] also designed for the enhancement of searchable encryptions. As we all know that proposed SSE schemes have high efficiency, but still they suffer with a complicated case like secret key distribution. In the above contexts the users have to securely share secret keys which are used for data encryption and the same keys need to be share with others in order for downloading the data from the cloud server. In order to solve all the above issues, a well known author like Boneh *et al.* [13] try to propose a more flexible primitive, namely Public Key Encryption with Keyword Search (PEKS) in order to provide more security for the data in an encrypted manner inside the cloud server. Given the trapdoor and the PEKS ciphertext, the server can test whether the keyword underlying the PEKS ciphertext is equal to the one selected by the receiver. If so, the server sends the matching encrypted data to the receiver.

2. CONTRIBUTIONS FOR 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* (DS-PEKS) to address the security vulnerability of primitive PEKS which is already proposed in the literature.
2. Next we try to 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 DS-PEKS using the proposed Lin-Hom SPHF.
4. Finally in order to illustrate the feasibility of our novel framework, an efficient instantiation of our SPHF, which is almost compared with the Diffie-Hellman algorithm which is available in the literature.

2.1 MOTIVATION

There are four types of services available in the cloud storage and one among them is DaaS which is the focus of our proposed work by securing the service of DaaS. We also prove that the proposed framework also gives the best security for the data stored on the cloud storage servers [14]. Now let us discuss about each and every service in detail as follows:

A. IaaS (Infrastructure as a Service)
B. PaaS(Platform as a Service)
C. SaaS(Software as a Service)
D. DaaS (Data /Data Base as a Service)

A. IaaS (Infrastructure as a Service)

This is the first service out of various services that are available in the cloud. This service mainly deals with application level and it is basically used to set the infra-structure for the users. This service is mainly used to create infrastructure for the set of PCs that are linked in an area. The persons who come under this service is IT Professionals,

B. PaaS (Platform as a Service)

The second important service in the cloud computing is Platform as a Service, where this is mainly used for customization of cloud server. Here in this service we try to set the platform for the users, where the developer comes under this service. Here the cloud server customizes which type of platforms is needed for their company usage is seen in this service.

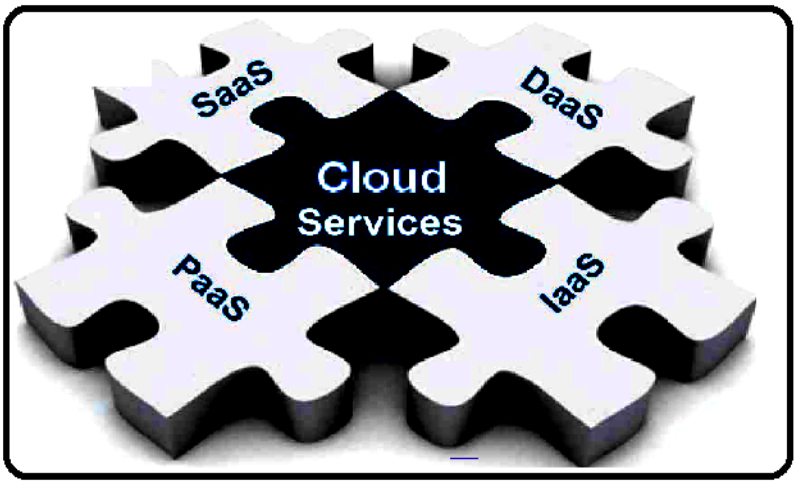


FIGURE.1. REPRESENTS THE VARIOUS CLOUD SERVICES THAT ARE AVAILABLE IN REAL TIME CLOUD

C. SaaS (Software as a Service)

The third service one among the best services in cloud computing is Software as a

Service, where this is mainly used for a consumer to use the cloud service provider's applications running on a cloud IaaS. Generally business end-users come under this service where all the software's that are required for running the cloud are processed in this service.

D. DaaS (Data/Database as a Service)

This is the last one among the set of cloud services that was launched and included in various cloud client services is DaaS, which is clearly seen in above figure 2.This DaaS service is used mainly for storing the data in the form of encrypted manner [15]. As this is having various advantages compared with other cloud client services, it has a small limitation like the data which is stored in this DaaS is not stored in the encrypted manner which is stored in the plain manner. So in this proposed thesis we try to encrypt the data before it is uploaded into the cloud using DaaS service.

Also we enabled this DaaS service by providing extended security by using a dual server technique, where the cloud server will try to generate two keys for providing security for the sensitive data. The user need to have the two keys with him for downloading the data in a plain text manner from the cloud server, if he/she fail to enter valid keys during authentication, the data can't be downloaded in a plain text manner.

Also we discuss about the similarity that takes place between the traditional PEKS and also the secure channel free PEKS which is used in the proposed work.

2.2 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.

2.3 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 (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.

3.1 PRELIMINARY KNOWLEDGE

A Dual-Server Public Key Encryption with Keyword Search 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 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

3.2 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.

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

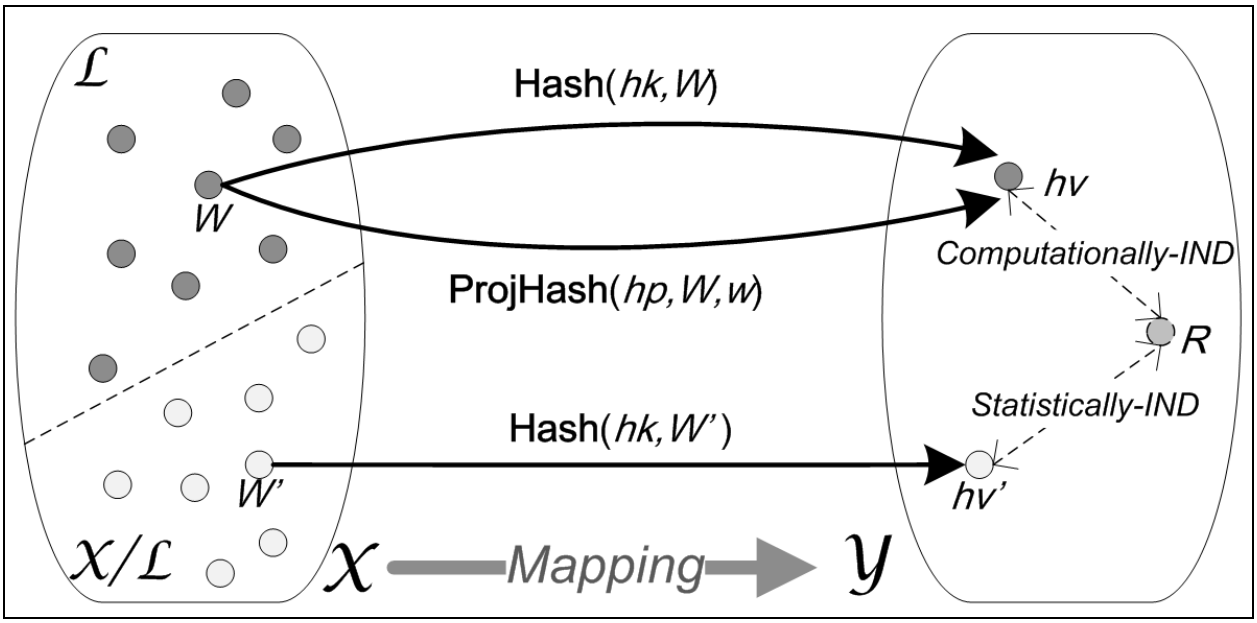


FIGURE.2. REPRESENTS THE ARCHITECTURE OF AN SMOOTH PROJECTIVE HASH FUNCTION (SPHF)

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:

1. SPHFSetup
2. HashKG
3. ProjKG
4. Hash
5. ProjHash

SPHF-Setup (1^{λ})**:** 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 ∈ *Y* for the word *W* from the hashing key hk.

ProjHash(hp, *(L,* param*),W,w)*: outputs the hash value hv1 ∈ *Y* for the word *W* from the projection key hp and the witness *w* for the fact that *W* ∈ *L*.

The *correctness* of an SPHF requires that for a word *W* ∈ *L* with *w* the witness,

Hash(hk, *(L,* param*),W)* = ProjHash(hp, *(L,* param*),W,w)*.

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

*V*1 ={*(L,* param*,W,* hp,hv*)*|hv=Hash(hk, *(L,* param*),W)*},

*V*2 ={*(L,* param*,W,* hp,hv*)*|hv $←*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* \ *L*.

4. IMPLEMENTATION PHASE

Implementation is the stage where the theoretical design is converted into programmatically manner. In this stage we will divide the application into a number of modules and then coded for deployment. We have implemented the proposed concept on Java programming language with JEE as the chosen language in order to show the performance this proposed novel IPath protocol. The front end of the application takes JSP,HTML and Java Beans and as a Back-End Data base we took My-SQL Server. The application can be executed either on a single PC or it can be executed on multiple PC's all connected over a LAN. The application is divided mainly into following 4 modules. They are as follows:

1. System Construction Module
2. Semantic-Security against Chosen Keyword Attack Module
3. Front Server Module
4. Back Server Module

Now let us discuss about each and every module in detail as follows:

4.1 SYSTEM CONSTRUCTION MODULE

In the first module, we develop the system with the entities required to provide 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. Here the users are of two types: One is Data Owner and other is Data user. The data owner is the person who can upload the files into the cloud server in a secure manner and they can maintain indexes for the uploaded files. Here the Data user is one who can access the files from the cloud server by providing the credentials

that are required for accessing the files in a plain manner.

2) **Cloud Service Provider (CSP): T**he 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.All these roles can be clearly shown in below figure 4.

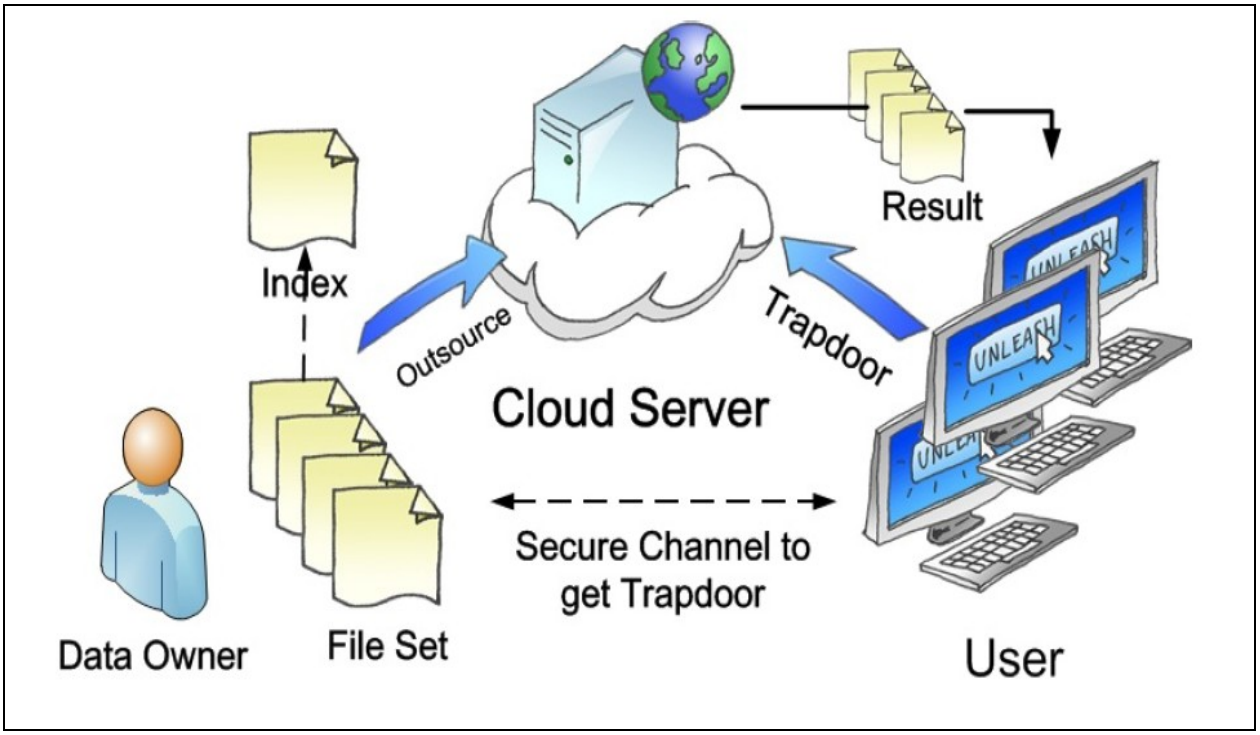


FIGURE.3. REPRESENTS THE ARCHITECTURE OF DUAL-SERVER PUBLIC KEY ENCRYPTION WITH KEYWORD SEARCH (DS-PEKS)

4.2 SEMANTIC-SECURITY AGAINST CHOSEN KEYWORD ATTACK MODULE

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.

4.3 FRONT SERVER MODULE

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

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

testing-states to the back server with the corresponding trapdoor and PEKS ciphertexts hidden.

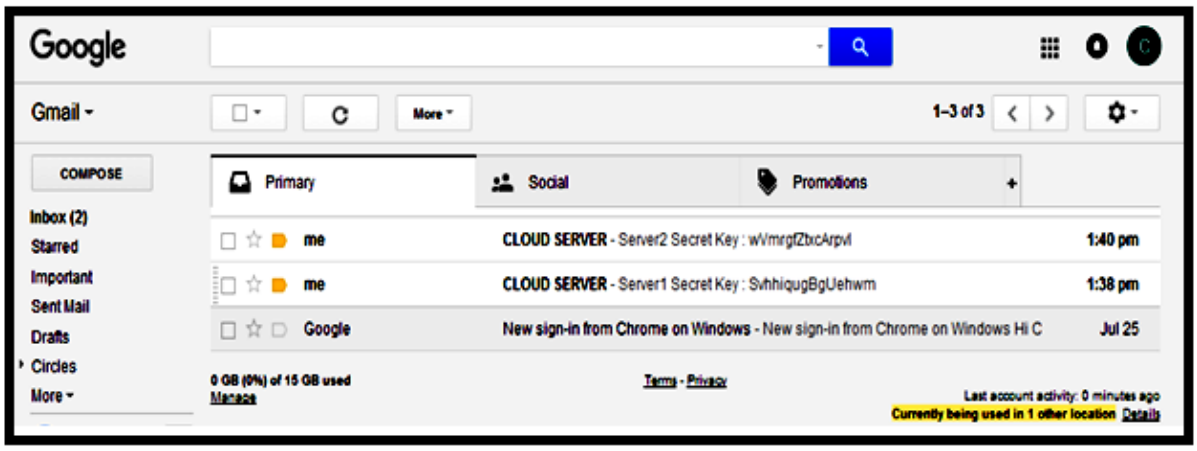
4.4 BACK SERVER MODULE

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.

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.