International Advance Journal of Engineering Research (IAJER)
Volume 3, Issue 8 (August- 2020), PP 15-21
ISSN: 2360-819X
www.iajer.com

Research Paper **Open Access**

Secure and Privacy Preserving-Keyword Search Retrieval over Cloud Data

Ridhi Thakur
Information Technology Sreenidhi Institute of Science and Technology. Yamnampet, Ghatkesar, Telangana 501301, India.

ABSTRACT:- A few years ago, customers who would want to store their records in the storage area would need to buy their own service. But to maintain an owner server it takes plenty of financial charges like purchasing hard disk, installing the operating structures and software et cetera and additionally unable to save the records for long term which leads to the loss of data availability so, because of these motives for preserving the facts for long time with infrastructure cloud computing has come into existence. Here the cloud is offering us with everything that I mentioned above for availability. The cloud service provider who will be providing the cloud will store the information for a long-term. After buying the storage area as cloud from the CSP, only the simplest and remaining maintenance will be taken care by the CSP. But after purchasing the cloud Customers used to store their information into the cloud however, they suffered with issues regarding protection., because the CSP can only provide the user with space whereas, the protection and security should be taken care by the client itself. So, forgetting that security and protection the customers might be covering might be converting the data into a ciphertext configuration and keep into the cloud storage. Later, the authorize customers could convey the ciphertext into a straightforward text. Even do they provide cryptographic techniques, conserving and maintaining the direct entrance from cloud storage engine little bit of security breach in threat.to overcome problems like these, the data owner who desires to upload the data, can do it by using keywords which are beneficial for discerning in the cloud for getting results. Trusted users can search with the help of keywords in the cloud server for accessing the documents securely. Numerous previous systems offer a single cloud server system for searching the information in the cloud, however, due to this approach there is a threat to getting the keywords by using non-authorize customers for having access to the cloud records.to guard the cloud information, I are implementing a new approach with multiple cloud servers that search and verify the authorized customers keywords and sends it to the user with the proper Co-ordination of every server. That means that the data or name store the documents in a cipher text format and searching keywords should be sent to the cloud server, and after receiving certain keywords, first the cloud server can convert the keyword as a hash code with the hashing technique and also store it. Now, the second cloud server all also convert that the data owner may store the documents in a cipher text format and searching keywords should be sent to the cloud server, and after receiving searching keywords, first the cloud so you can convert the keyword as a hash code with the hashing technique and also store it. Now, the second cloud server all also converts the keyword as a hash code and both the cards are set in the cloud server. When the user desires for accessing the cloud data, need to send a searching request to the cloud servers with keywords. If any of the keywords are matching the results are sent to the user and they can convert the ciphertext and hence view the plain Thus, by using this approach any unauthorized user cannot get the cloud data effortlessly without interacting with the multiple cloud servers.

Keywords: DS-PEKS, Cryptography, Ciphertext, Keyword search, Security, Retrieval, Hash code, Cloud

I. INTRODUCTION

The main purpose of this approach is to maintain the important records of the cloud from non-authorized users and to do so I use searchable cryptography techniques for searching with keywords in the cloud storage for getting results as data files. Here, the related keywords are sent to the user emails that are registered in this system as user for having access to the cloud data securely.

Present System

In Present System using single server searchable cryptography, data owner can need to upload the file data along with keywords also store in cloud with plain text format due to which for non-authorized user can read the keywords and searching with knowing keywords over cloud storage for accessing the cloud data which

is not recommended for secure searchable cryptography techniques. The previous many old searchable cryptography techniques are used to provide security in single cloud server but they all are unable to restrict to knowing the keywords by unauthorized people.

Suggested System

The Suggested System using multiple servers for secure searchable over cloud. But using single key cryptography is owner of the data needs to encrypt the file data and keywords after can store in cloud which is not safety for authorized user, because for searching with keywords users should be know the key for accessing the cloud data which is shared by data owner. While sharing the keywords and keys if any malicious users knew by them then there is a chance to breaching the security by searching with knowing keywords and keys. So that for protecting cloud data, data owner needs to use multiple keys cryptography techniques. In this technique data owner can only encrypt the file data with single key cryptography system and later the keywords are should be forwarded to multiple cloud servers. After receiving the keywords from data owner then first cloud server can convert the keyword in hashing text by using first key of cloud server one and again the keywords are sent to second cloud server for converting keywords as hashing text by using of second cloud server first key.

In this system multiple servers are collaborated to each other for protecting sensitive data from malicious users. While searching keyword by users they first send a keyword request to multiple servers not to cloud directly. After receiving the keywords from users then each sever participating in searching scheme. Like first cloud server converting the keyword into hashing code with second key and share to another cloud sever, this server which is second cloud server can also converting the keyword as hashing code then matching these two hashing codes if they are matching same then cloud server can return the results as a data files to users or else they can return the results as No Records Found to users. The results are can be access by users from cloud server by decrypting them if their keywords are matching. To enabling this kind of searching hashing techniques, I can protect the cloud data from malicious users.

II. SYSTEM ANALYSIS

Literature Survey

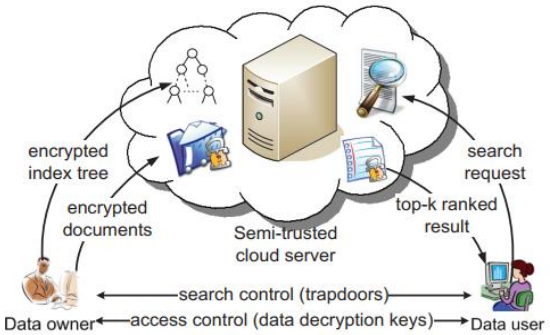

Fig.1 Using Ranked search over encrypted cloud

From Figure 1 data owner can sending the cipher indexes (keywords) and cipher documents into cloud server for data arability. After data users they can search with multiple keywords and these keywords are first send to trapdoor and trapdoor can be search in cloud storage with user's keywords. The cloud server can return top most data files to data users. But after receiving the cipher documents user can send key request to data owner because of decrypting those data owner can accept the data user request and send the related data file keys to data users. Using of this keys data user can decrypting as pain documents. But this system is not providing secure searching scheme because of sharing keys and keywords to data users publicly which is not recommended using for searchable crypto techniques.

Survey on Public-Key Encryption:

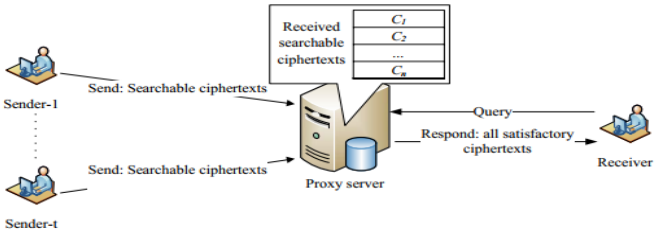

Fig.2 Searchable encryption with Proxy System

The multiple keys cryptography technique used for encrypting the keywords and document files with one key and store into proxy server. The previous system provides searching technique with single key while searching with plain text keywords in cloud storage then there is a chance to knowing the keywords by cloud server which is leads to lock of security, because of cloud server is a semi trusted. To not readable keywords by CS, Sender should be encrypting the keywords and documents with Receiver first key and store into cloud server. Later Receiver can convert the keywords as cipher text format with first key and send to proxy server, the proxy server can search in cloud with cipher keywords and get cipher documents and send to Receiver. After receiving the cipher document receiver can converting as plain documents with second key. By using of this system no one or cloud server cannot read the keywords and cannot accessing the cloud data by attackers who are not authorized to this system.

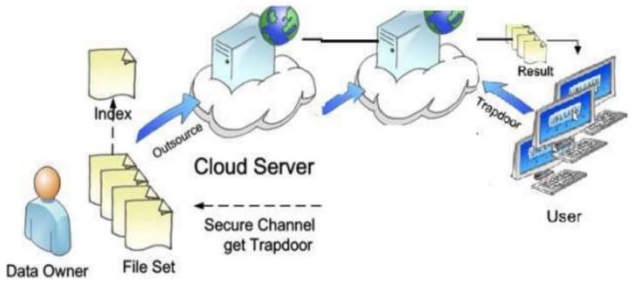

Fig. 3 PEKS framework

With help of RSA cryptosystem owner of the data can securely preserve their cipher documents into the cloud server. The implementation of this system is data owner encrypt their sensitive data which name is file set and index before store in cloud. Here data owner chooses the trusted users and share the indexes and decryption keys through secure channel (i.e. email). The trusted user can get indexes and decryption key then search with index in cloud and get relevant cipher documents and decrypt those cipher documents with decryption key with securely. But sharing the bulk of indexes and keys are burden to data owner because he/she needs to online every time as Ill as it takes heavy computation for ciphering and deciphering the file sets.

System design:

Program design has been divided into three categories such as interface design, UML design with advantages created by enterprises in a simple way and user case by User case, process streaming series, class diagrams include details of particular groups in the process with strategies to be included in the paper, if our paper involves a UML.

Execution:

Implementation is a process in which I strive to provide the logical results of research carried out at the design stage, and the central and critical aspect of the paper comes into play in this stage in a significant part of Coding in Business Logic.

System Modules:

Multiple Server Cryptosystem Smooth Projectile Hash Functions (MSCSPHFs):

MSCSPHFs are basically groupings of L-language function sets. Some similar keys document these characteristics, where the hazard key may be seen as the private key and the paper ion key as the free key. Both capacities will contribute to a comparable outcome with the word W[E]: an observatory W [e] is only implemented with the hashing key and with the paper ion key. Clearly, if W 6 selects L, there is no other investigator, and the excellence property is autonomous to hp. As a consequence, Hash can't be thought about in either situation, understanding hp.

Data Owner:

It has large data should have been put away and partaken in cloud framework. In our scheme, the element is responsible for characterizing Keywords file and execute text encryption process. It also transmits chip text to air and even transfers keywords (kw) to servers. These two servers will authenticate the keywords to be stored in the cloud.

Data User:

It wants to access a huge volume of information in the cloud network. The entity first downloads the

corresponding cipher text. It then performs the procedure Decrypt of the planned scheme. Here first, before you access the cipher code, you should authenticate the front server with the data consumer scans for keywords, so that the keywords are sent to the front server so that both keywords and back server are both encrypted and keywords are checked in the cloud, if any of the keywords fit, and encrypted files must instead be sent to the data user. These files may be decrypted and vied by users of the app.

III. MULTIPLE SERVER CRYPTOSYSTEM:

Multiple Service Cryptosystem Scheme primarily consists of a wider precise, the key Generation rule set creates the public / non-public key pair front service and back server instead of the single user. In addition, the Multiple Server Trapdoor trapdoor algorithm mentioned here is public while the Trapdoor algorithm uses the private key for a recipient as an input into the traditional multi-server cryptosystem context. This differentiation is based on the various mechanisms used by both systems. Due to the fact that there is only one most effective server in a conventional multi-server encryption system, the Server may launch a guessing attack against a keyword ciphertext to recover the encrypted keyword if the trapdoor creation algorithm is public. Consequently, the somaticized security cannot earn miles. But in the sense of the Multiple Application Cryptosystem I will demonstrate it later. Another distinction between our proposed Cryptosystem Multiples Servers and the traditional Cryptosystem Multiple Servers is that the test algorithm is divided into two different algorithms: The Front Test and the Back-Test run by both independent servers. This requires defense from guessing in the keyword. When the client or the user is asked a question regarding the Cryptosystem Multi-Domain Network, the front domain settings the trapdoor and all Multi-Server Cryptosystem chipboards with its personal keys, meaning that all internal tests are routed to the back end with relative trapping tools and multi-server Cryptosystem chipboards. Then, one of the two servers decide which records the customer or the consumer requires using his / her private key and then internal monitoring states from the server are required. (FS).

System Architecture:

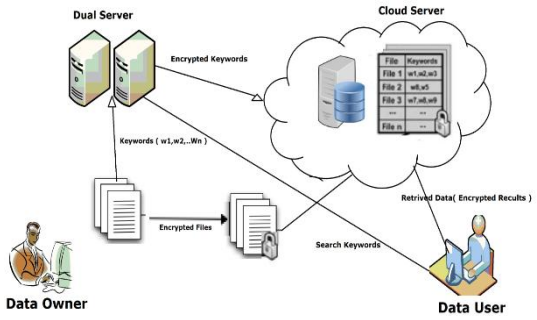

Fig.4 Paper Architecture diagram

The Figure.6 contains data owner data user, multiple server and cloud server. The data owner can encrypt documents and send to cloud server and keywords are sent to multiple servers. The data user can search with keywords and send to multiple servers then they can perform the testing on cloud server later they can send the results to data users if test case is valid only. After receiving the cipher documents at user then they can decrypt the documents.

Feasibility Study

The Feasibility study is done on my application .In this study we have seen how our application needs to developed and how it can make money for the users and how it will helpful for public and also how safety is our application how much time will it take need manpower for development need technologies for application need hardware for the paper running and maintains performance of the application all this are covered in this seedy.

Functional Requirements

System Init ():

In this requirement it can generate system parameter randomly for generating multiple keys for multiple servers.

Key Generation ():

In this requirement it takes input as a system parameter and it returns multiple server first key and second key.

Multiple Server Enc ():

In this requirement it takes input as a multiple each server first key and keyword then it returns hashing

keywords and store in cloud server.

Multiple Server Trapdoors ():

In this requirement it takes input as a user keyword and multiple server first key and convert the keywords as hashing code and send to multiple servers for testing whether the hashing code is available or not.

First Server Test:

In this requirement it takes input as first server second key and user kw then it returns new hash code and compare with old hash code then it returns true otherwise it returns false.

Second Server Test:

In this requirement it takes input as second server second key and user kw and it returns new hash code and compare with old hash code then it returns true or else it returns false. Later they can compare both statuses if both are true then it returns results to users or else it returns acknowledgment like No Records Found.

Application needs Non-Functional Requisites

Expanded System admin security:

Supervisor to abstain from the improper use of the application by PC which is supposed to be secured and available.

Compactness:

The potential of this paper is not very helpful, rendering it possible to understand and react to the same degree by the user.

Quality:

The tools available in the implementation of this sub-structure are very successful in providing us with the required inquiries.

Time taken: The reaction time of our application is quick.

Diverseness: Our application can be stretched.

Security: Our paper is high secured. With data base encryption

Present System Algorithm

Setup(1^λ). Take as input the security parameter λ, run the algorithm SPHFSetup and generate the global parameters param, the description of the language $\mathcal{L}$ and a collision-resistant hash function $\Gamma : KW \to \mathcal{Y}$. Set the system parameter $P =< \text{param}, \mathcal{L}, \Gamma >$.

KeyGen(P). Take as input P, run the algorithms < HashKG, ProjHash > to generate the public/private key pairs $(pk_{FS}, sk_{FS}), (pk_{BS}, sk_{BS})$ for the front server and the back server respectively.

$$pk_{FS} \leftarrow \text{HashKG}(P), \quad sk_{FS} = \text{ProjKG}(P, pk_{FS}), \quad pk_{BS} \leftarrow \text{HashKG}(P), \quad sk_{BS} = \text{ProjKG}(P, pk_{BS}).$$

DS-PEKS($P, pk_{FS}, pk_{BS}, kw_1$). Take as input P, pk_{FS}, pk_{BS} and the keyword kw_1, pick a word $W_1 \in \mathcal{L}$ randomly with the witness w_1 and generate the PEKS ciphertext CT_{kw_1} of kw_1 as following.

$$x_1 = \text{ProjHash}(P, pk_{FS}, W_1, w_1), \quad y_1 = \text{ProjHash}(P, pk_{BS}, W_1, w_1), \quad C_1 = x_1 \circledast y_1 \circledast \Gamma(kw_1).$$

Return $CT_{kw_1} =< W_1, C_1 >$ as the keyword ciphertext.

DS-Trapdoor($P, pk_{FS}, pk_{BS}, kw_2$). Take as input P, pk_{FS}, pk_{BS} and the keyword kw_2, pick a word $W_2 \in \mathcal{L}$ randomly with the witness w_2 and generate the trapdoor T_{kw_2} of kw_2 as follows.

$$x_2 = \text{ProjHash}(P, pk_{FS}, W_2, w_2), \quad y_2 = \text{ProjHash}(P, pk_{BS}, W_2, w_2), \quad C_2 = x_2 \circledast y_2 \circledast \Gamma(kw_2)^{-1}.$$

Return $T_{kw_2} =< W_2, C_2 >$ as the trapdoor.

FrontTest($P, sk_{FS}, CT_{kw}, T_{kw}$). Takes as input P, the front server's secret key sk_{FS}, the PEKS ciphertext $CT_{kw_1} =< W_1, C_1 >$ and the trapdoor $T_{kw_2} =< W_2, C_2 >$, pick $\Delta w \in W$ randomly, generate the internal testing-state C_{ITS} as follows.

$$W = W_1 \circledcirc W_2, \quad x = \text{Hash}(P, sk_{FS}, W), \quad C = C_1 \circledast C_2 \circledast x^{-1}, \quad W^* = \Delta w \otimes W, C^* = \Delta w \bullet C.$$

Return $C_{ITS} =< W^*, C^* >$ as the internal testing-state.

BackTest(P, sk_{BS}, C_{ITS}). Takes as input P, the back server's secret key sk_{BS} and the internal testing-state $C_{ITS} =< W^*, C^* >$, test as follows.

$$\text{Hash}(P, sk_{BS}, W^*) \stackrel{?}{=} C^*$$

If yes output 1, else output 0.

IMPLEMENTATION

Technologies Used

Software used for the Paper:
JAVA, Apache Server, MySQL (Database), (EDITOR) Notepad++

I used a one-tiered architecture in our paper in Application Creation paper as our client would be built in a single system with the three levels of application growth like interface layer, where I use our software improvements to build a GUI application like JavaScript, HTML-5CSSJSP Pages, etc. I associated a connection from business layer to base layer of information by making use of JDBC (an API) and finalize base layer of information by building up the application's data form.

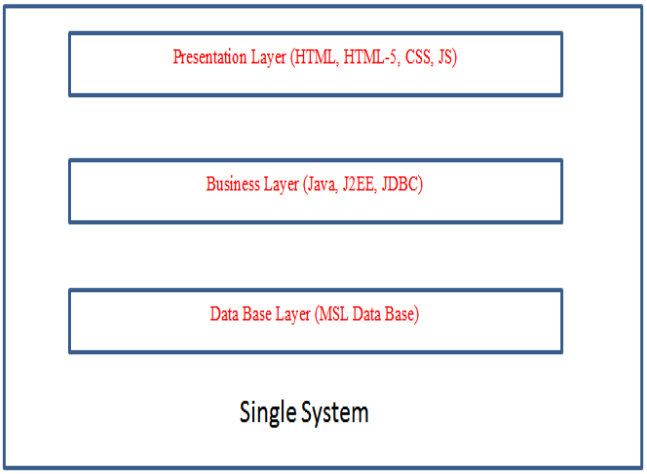

Fig.5 Paper Development using Single Tier framework

Enhancement of the approach using Java:
Setting-up the Software in our system

The downloaded software structure which is the oracle site has a free source according to the software we had introduced in out framework and for what we have set the framework way of java in our location of OS. We have utilized the principle rationale of our algorithm by java's core ideas for application of website We associated base level information by exploiting total concepts of JSP and connectivity of data base through JDBC. Thus, we accomplished the paper application by the primary tier framework.

Connectivity of the data through MYSQL

We have taken chosen free source programming MYSQL all the way from given site and run in our remote device we utilized it for developing out tables associated to data base of the paper according to extent prerequisite's in any event for easy to understand MYSQL access. And also, we exploited the SQLYog which is a naive software for visualizing the data graphically which provides an uncomplicated interface to the user.

The apache tomcat (local) web server job

Our paper basically necessitates a web server, so now we have made use of a free sharp source programming again. That's how our entire main code of the task will perform in webapps of that server structure area where the running of the application takes place onto the internet access so that clients can go through the paper.

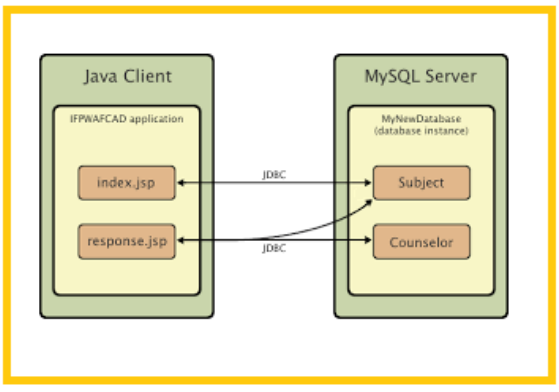

Fig.6 The structure of application development.

IV. CONCLUSION

In this paper, another method called Dual Server Public Key Encryption with Keyword Search (DS-PEKS) is suggested which would prevent the inner phrase from speculating the traditional PEKS framework 's weakness. We have also shown another Smooth Projectile Hash Function (SPHF), which we use to construct a traditional plot DS-PEKS competitive release of the latest SPHF based on the Diffie-Hellman problem is also added to our research that offers a strong and efficient DS-PEKS plot without matches.

REFERENCES

[1]. R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "another general framework for secure open key encryption with catchphrase search, "in Proc. twentieth Australasian Conf. Inf. Secure. Protection (ACISP), 2015, pp. 59–76.

[2]. D. X. Tune, D. Wagner, and A. Perrig, "Viable methods for searches on scrambled information," in Proc. IEEE Symp. Secure. Security, May 2000, pp. 44–55.

[3]. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Request preserving encryption for numeric information," 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 proficient constructions, "in Proc. thirteenth ACM Conf. Compute. Secure. (CCS), 2006,pp. 79–88.\

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

[6]. R. Gennaro and Y. Lindell, "A structure for secret key basedauthenticated key trade," 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 accessible review log," in Proc. NDSS, 2004, pp. 1–11.

[8]. M Abdalla et al., "Accessible encryption returned to: Consistency properties, connection to mysterious IBE, and augmentations," in Proc. 25thAnnu. Int. Conf. CRYPTO, 2005, pp. 205–222.

[9]. D. Khader, "Open key encryption with catchphrase search based on k-strong IBE," in Proc. Int. Conf. Computer. Sci. Appl. (ICCSA), 2006, pp. 298–308.

[10]. P. Xu, H. Jin, Q. Wu, and W. Wang, "Open key encryption with fuzzy catchphrase search: A provably secure plan under keyword guessing assault," IEEE Trans. Computer., vol. 62, no. 11, pp. 2266–2277, Nov. 2013