An Implementation of Dual -Server Public-Key Encryption with Keyword Search for Secure Cloud Storage

Shaikh Tofik S.[1], Nirmal M.D..[2]

[1]*Student, Computer Department, PREC Loni, Maharashtra, India*
[2] *Assistant Proffesor, Computer Department, PREC Loni, Maharashtra, India*

ABSTRACT

Accessible encryption is of expanding enthusiasm for ensuring the information protection in secure accessible distributed storage. In this paper, we explore the security of a notable cryptographic primitive, in particular, open key encryption with catchphrase seek (PEKS) which is extremely valuable in numerous uses of distributed storage. Tragically, it has been demonstrated that the customary PEKS system experiences a natural uncertainty called inside watchword speculating assault (KGA) propelled by the pernicious server. To address this security helplessness, we propose another PEKS system named double server PEKS (DS-PEKS). As another primary commitment, we characterize another variation of the smooth projective hash capacities (SPHFs) alluded to as direct and homomorphic SPHF (LH-SPHF). We at that point demonstrate a non specific development of secure DS-PEKS from LH-SPHF. To outline the attainability of our new structure, we give an productive instantiation of the general structure from a Decision Diffie–Hellman-based LH-SPHF and demonstrate that it can accomplish the solid security against inside the KGA in cloud computing.

Keyword - *encryption, cryptography, security, cloud computing.*

1. INTRODUCTION

The basic good position of appropriated stockpiling is its unavoidable customer accessibility moreover it's in every practical sense limitless data stockpiling limits. Despite such focal points gave by the cloud, the genuine test that residual parts is the stress over the mystery and security of data while grasping the appropriated capacity organizations [1]. For instance, decoded customer data set away at the remote cloud server can be vulnerable against external attacks began by unapproved untouchables and inside ambushes began by the unscrupulous cloud specialist co-op (CSPs) associations [2]. There are a couple reports that confirm data breaks related to cloud servers, due to harmful strike, theft or internal botches [3]. This raises sensitivity data may contains to a great degree sensitive individual affiliation/information. Circulated distributed storage outsourcing has transformed into an unmistakable application for endeavors and relationship to diminish the weight of keeping up colossal data lately.No withstanding, truth be told, end customers may not by any methods accept the cloud limit servers and might need to scramble their data some time as of late exchanging them to the cloud server with a particular ultimate objective to secure the data security. This ordinarily makes the data use more troublesome than the traditional stockpiling where data is kept in the nonappearance of encryption. One of the normal courses of action is the accessible encryption which allows the customer to recuperate the mixed records that contain the customer demonstrated catchphrases, where given the watchword trapdoor, the server can find the data required by the customer with no issue .

1.1 PROBLEM STATEMENT

The Problem is to determine how to securely search any document from cloud in form of encrypted data with the help of dual servers.

- Dual Server-public key encryption with keyword search (PEKS).
- How to Store data in Secure form on cloud.
- How to Store data in Secure form on cloud.

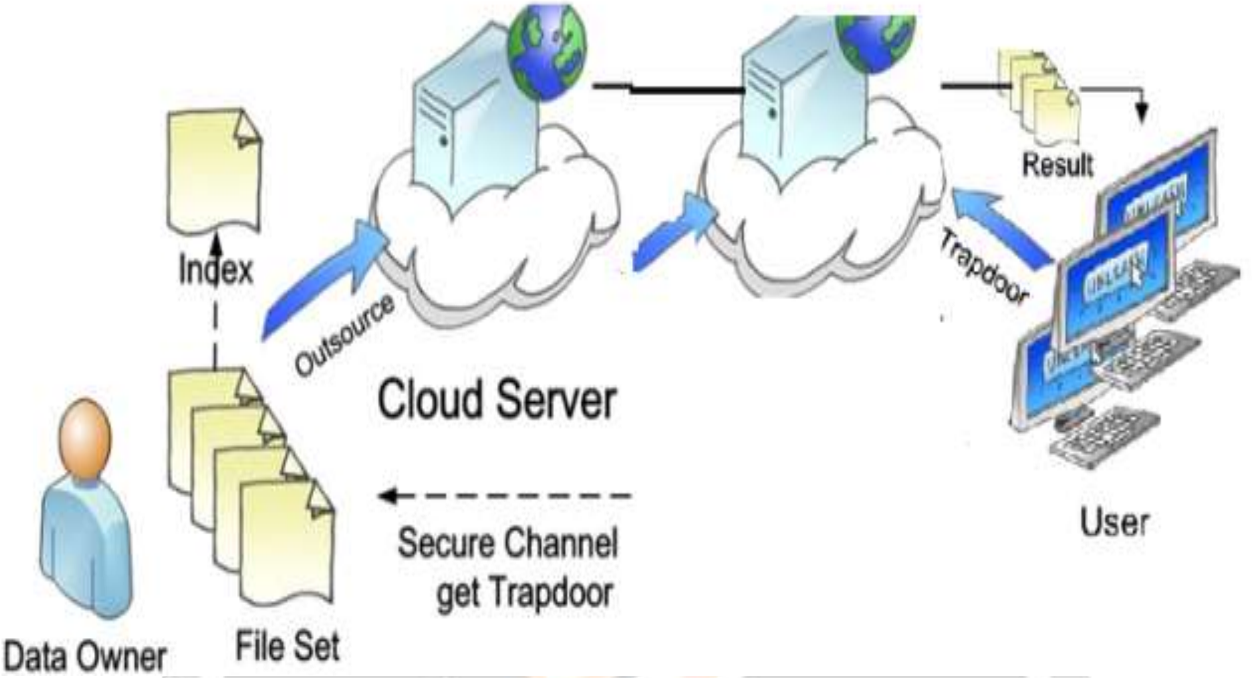
2. LITERATURE SURVEY

Distributed computing speaks to today’s most energizing registering design move in data technology [1]. However, security also, security are seen as essential snags to its expansive adoption[2]. Here, layout a few basic security challenges what's more, spur promote examination of security answers for a dependable open cloud environment [3]. Distributed computing is the most recent idea for the since quite a while ago envisioned vision of registering as a helpfulness. It is important to store data on data stockpiling servers, for example, mail servers and record servers in encoded casing to enhance security and assurance perils. Regardless, this ordinarily proposes one needs to give up convenience for security. For example, if a client wishes to recoup just reports containing certain words, it was not in advance known how to let the data stockpiling server play out the request and answer the inquiry without loss of data secrecy[4]. The issue of looking for on data that is encoded using an open key system. Consider customer Bob who sends email to customer Alice mixed under Alice's open key. An email section needs to test whether the email contains the watchword “with the objective that it could course the email as needs be. Alice, on the other hand does not wish to give the entryway the ability to unscramble each one of her messages. We done and build up an instrument that engages Alice to give a key to the entryway that enables the way to test whether the word “is a watchword in the email without realizing whatever else about the email. We imply this framework as Public Key Encryption with watchword Search. As another case, consider a mail server that stores diverse messages transparently mixed for Alice by others. Using our instrument Alice can send the mail server a key that will enable the server to recognize all messages containing some watchword which is we need to search [5]. The good property in this arrangement allows the server to filter for a catchphrase, given the trapdoor. In this manner, the verifier can just use an untrusted server, which makes this thought to a great degree sensible. Taking after Boneh et al's. work, there have been following works that have been proposed to redesign this thought. Two essential thoughts fuse the assumed catchphrase guessing attack and secure channel free, proposed by Byun et al. furthermore, Baek et al., independently. The past gets it the route that before long, the space of the catchphrases used is to a great degree obliged, while the last considers the departure of secure channel between the recipient and the server to make PEKS rational. Grievously, the current improvement of PEKS secure against catchphrase conjecturing strike is recently secure under the unpredictable prophet show, which does not reflect its security in this present reality. Also, there is no aggregate definition that gets secure channel free PEKS designs that are secure against picked catchphrase ambush, picked cipher text attack, and against watchword hypothesizing attacks, in spite of the way that these musings have all the earmarks of being the most down to business utilization of PEKS primitives[6].Aanother framework, called secure server-task open key encryption with catchphrase look for (SPEKS), was familiar with improve the security of dPEKS (which encounters the on-line catchphrase hypothesizing attack) by describing another security illustrate 'one of a kind cipher text in distinguishability[7].

3. PROPOSED SYSTEM

Public Key Encryption with Keyword Search (PEKS) that empowers a client to seek scrambled information in the topsy-turvy encryption setting. In a PEKS framework, utilizing the beneficiary's open key, the sender appends some scrambled watchwords (alluded to as PEKS figure writings) with the encoded information. The beneficiary at that point sends the trapdoor of a to-be-looked watchword to the server for information seeking. Given the trapdoor and the PEKS figure message, the server can test whether the catchphrase basic the PEKS figure content is equivalent to the one chose by the recipient. Provided that this is true, the server sends the coordinating encoded information to the beneficiary.

3.1 PROPOSED SYSTEM ARCHITECTURE:



3.1.1 Data Owner:

Register with cloud server and login(username must be unique). Send request to Public key generator (PKG) to generate Key on the user name. Browse file and request Public key to encrypt the data, Upload data to cloud service provider. Verify the data from the cloud .

3.1.2 Public Key Generator:

Receive request from the users to generate the key, Store all keys based on the user names. Check the username and provide the private key. Revoke the end user (File Receiver if they try to hack file in the cloud server and un revoke the user after updating the private key for the corresponding file based on the user).

3.1.3 Key Update:

Receive all files from the data owner and store all files. Check the data integrity in the cloud and inform to end user about the data integrity. Send request to PKG to Update the private key of the user based on the date parameter (Give some date to update Private Key). List all files, List all updated Private Key details based on the date and users, List all File attackers and File Receive Attackers.

3.1.4 Front Server:

After receiving the query from the receiver, 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.

3.1.5 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.

4. ALGORITHM

A DS-PEKS scheme is defined by the following algorithms.

_ Setup(1_). Takes as input the security parameter _, generates the system parameters P;

_ KeyGen(P): Takes as input the systems parameters P, outputs the public/secret key pairs (pkFS; skFS), and (pkBS; skBS) for the front server, and the back server respectively;

_ DS-PEKS(P; pkFS; pkBS; kw1): Takes as input P, the front server's public key pkFS, the back server's public key

pkBS and the keyword kw1, outputs the PEKS ciphertext CTkw1 of kw1;

_ DS-Trapdoor(P; pkFS; pkBS; kw2): Takes as input P, the front server's public key pkFS, the back server's public key pkBS and the keyword kw2, outputs the trapdoor

Tkw2 ;
_ FrontTest(P; skFS;CTkw1 ; Tkw2): Takes as input P, the front server's secret key skFS, the PEKS ciphertext

CTkw1 and the trapdoor Tkw2 , outputs the internal testing-state CITS;

_ BackTest(P; skBS;CITS): Takes as input P, the back server's secret key skBS and the internal testing-state CITS, outputs the testing result 0 or 1;

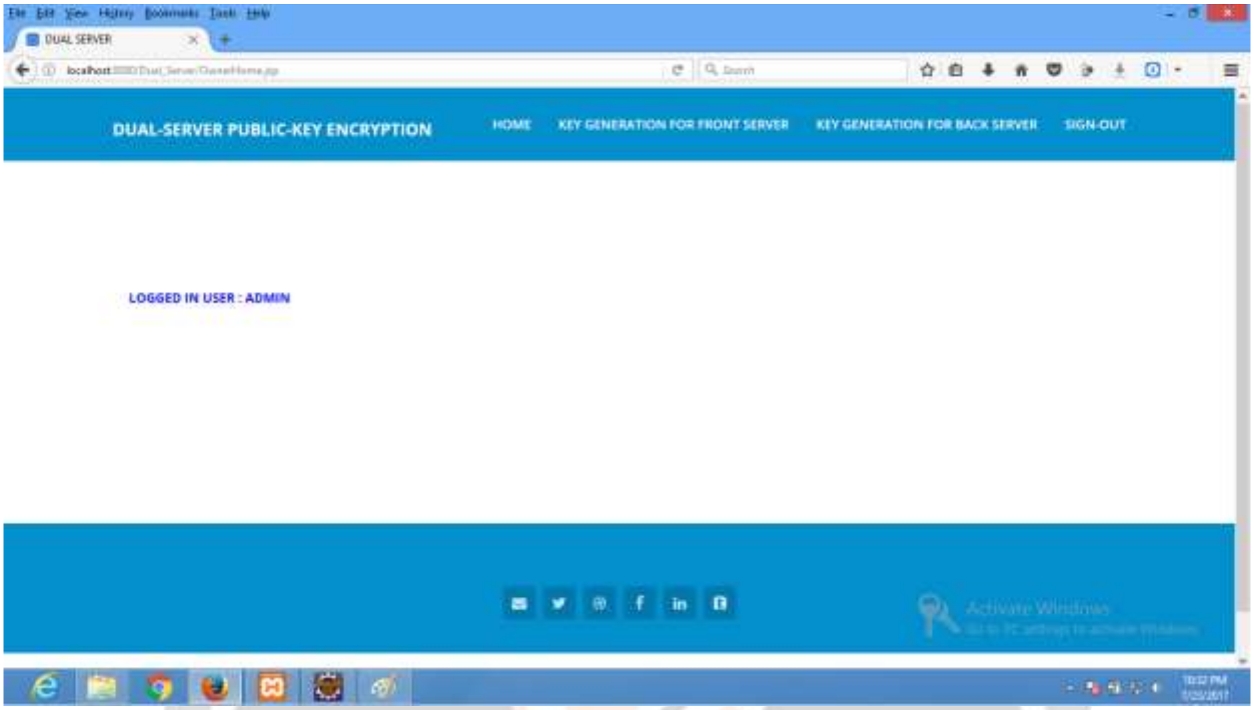
4.1 ADVANTAGES OF PROPOSED SYSTEM

All the current plans require the blending calculation amid the era of PEKS ciphertext and testing and thus are less proficient than our plan, which does not require any matching calculation.

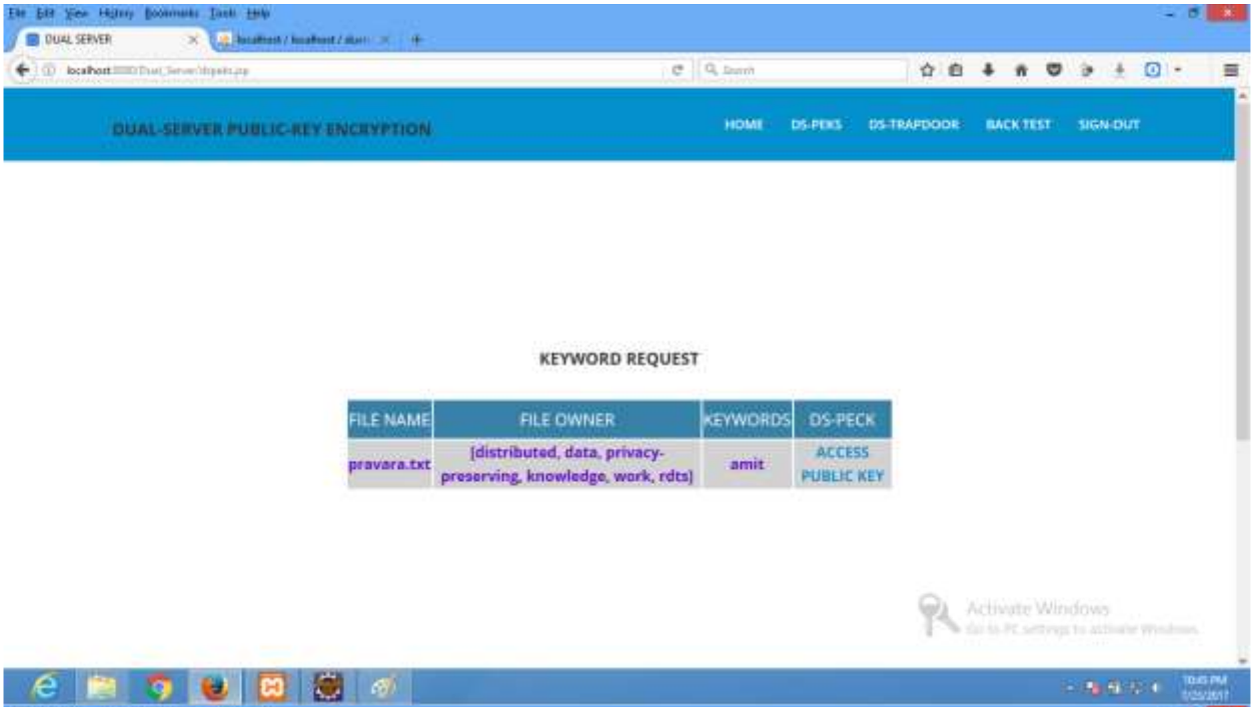
Our plan is the most productive as far as PEKS calculation. It is on account of that our plan does exclude matching calculation. Especially, the current plan requires the most calculation cost because of 2 blending calculation for each PEKS era.

In our proposed system, in spite of the fact that we additionally require another phase for the testing, our calculation cost is really lower than that of any current plan as we don't require any matching calculation and all the seeking work is taken care of by the server.

3. RESULTS

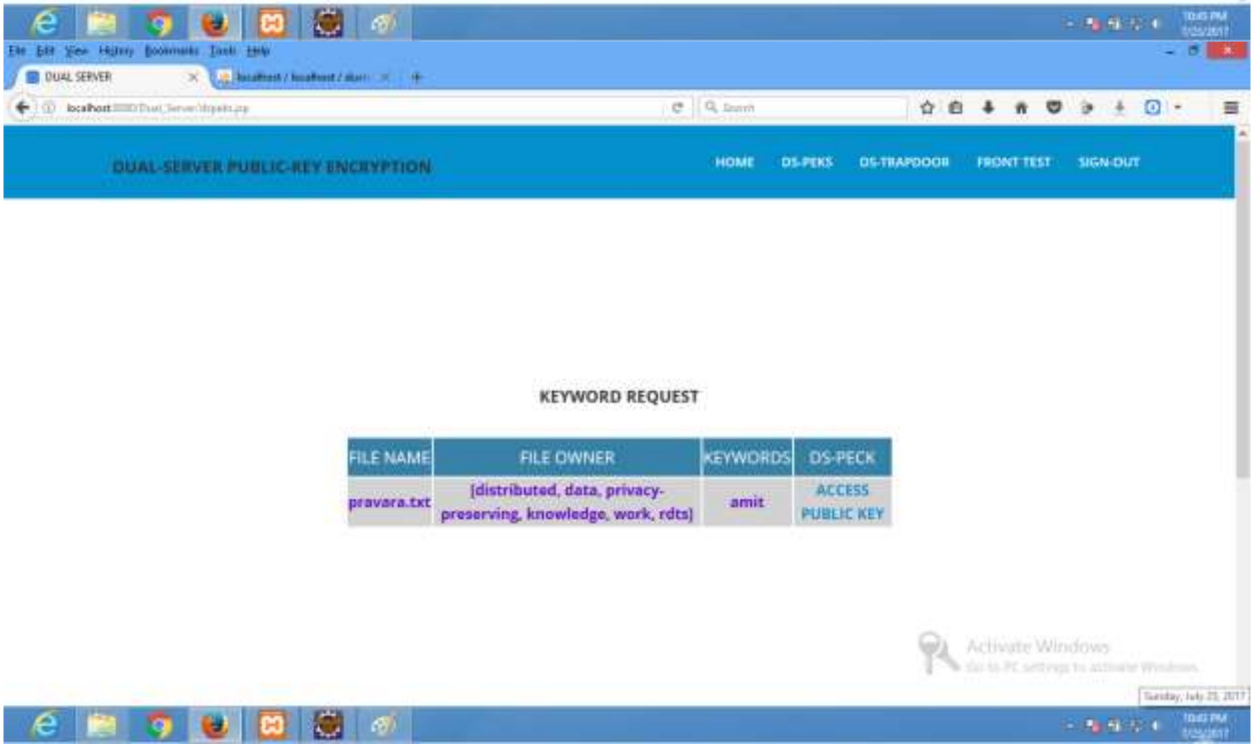

File Edit View History Bookmarks Tools Help
DUAL SERVER
localhost
DUAL-SERVER PUBLIC-KEY ENCRYPTION
HOME
KEY GENERATION FOR FRONT SERVER
KEY GENERATION FOR BACK SERVER
SIGN-OUT
Activate Windows

LOGGED IN USER : ADMIN



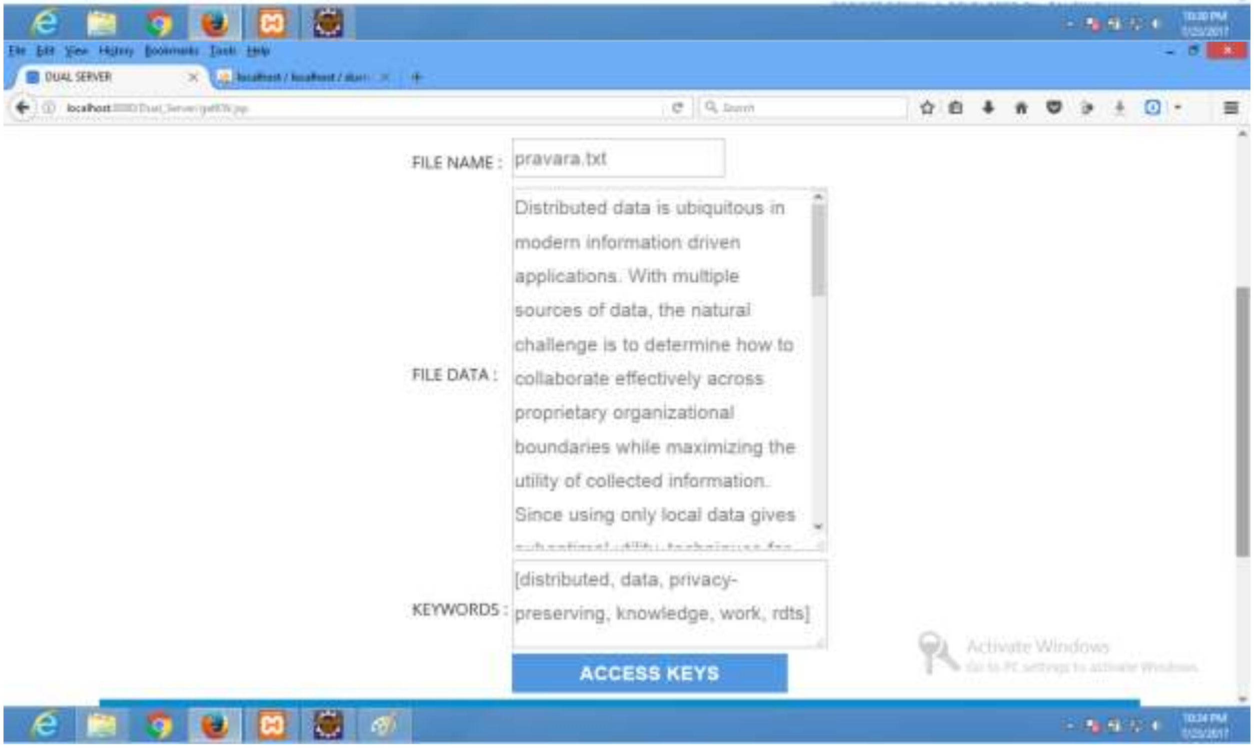
KEYWORD REQUEST

FILE NAME	FILE OWNER	KEYWORDS	DS-PECK
pravara.txt	[distributed, data, privacy-preserving, knowledge, work, rdts]	amit	ACCESS PUBLIC KEY



KEYWORD REQUEST

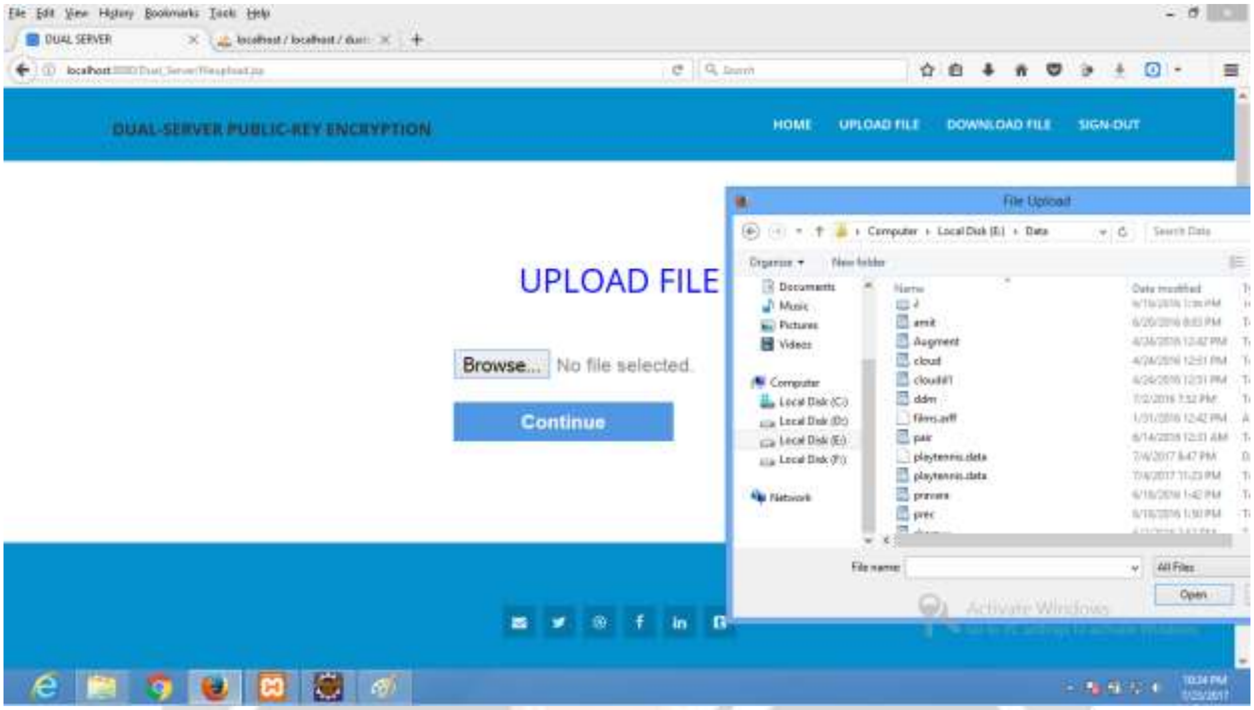
FILE NAME	FILE OWNER	KEYWORDS	DS-PECK
pravara.txt	[distributed, data, privacy-preserving, knowledge, work, rdts]	amit	ACCESS PUBLIC KEY



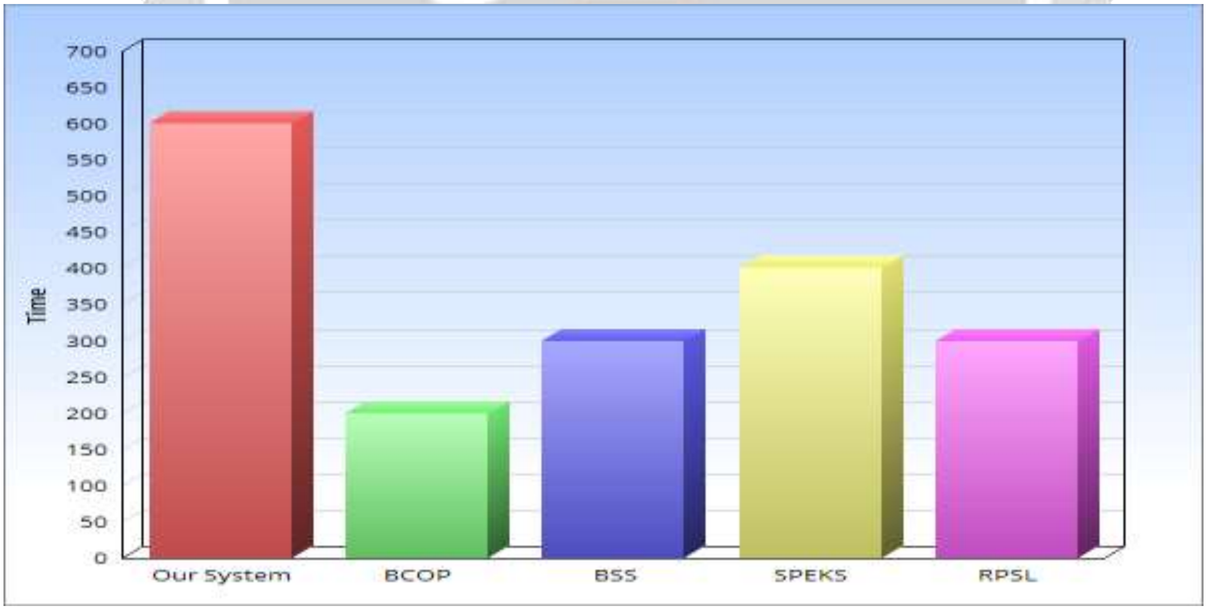
FILE NAME : pravara.txt

FILE DATA : Distributed data is ubiquitous in modern information driven applications. With multiple sources of data, the natural challenge is to determine how to collaborate effectively across proprietary organizational boundaries while maximizing the utility of collected information. Since using only local data gives

KEYWORDS : [distributed, data, privacy-preserving, knowledge, work, rdts]



6. COMPARATIVE TEST RESULTS



Our proposed system gives a tremendous improvement than conventional system also shown that it is valuable in various digital data storage fields, which show a higher level of security,efficiency and scalability of the system. Proposed system satisfies the usability factor
like Satisfaction, accuracy, effectiveness and efficiency.

Also proposed system is robust, but also it gives better security mechanism than conventional system.

7. CONCLUSION

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 impractical.

In this paper, we proposed another structure, named Dual- Server Public Key Encryption with Keyword Search (DSPEKS), that can keep within brutforce keyward attack which is an innate weakness of the PEKS system.

In future , According to technical view our proposed system is efficient and cost effective.

8. REFERENCES

1. Kamara S, Lauter K (2010) Cryptographic cloud storage. In: Sion R, Curtmola R, Dietrich S, Kiayias A, Miret JM, Sako K, Sebé F (eds) Financial Cryptography and Data Security, LNCS 6054. Springer, Berlin, Heidelberg, pp 136–149.

2. Hacigümüș H, Iyer B, Li C, Mehrotra S (2002) Executing sql over encrypted data in the database-service-provider model. In: Proceedings of SIGMOD, ACM, pp 216–227.

3. Subashini S, Kavitha V (2011) A survey on security issues in service delivery models of cloud computing. J Netw Comput Appl 34:1–11.

4. D. X. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in IEEE Symposium on Security and Privacy, 2000, pp. 44–55.

5. D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in EUROCRYPT, 2004, pp. 506–522.

6. L. Fang,W. Susilo, C. Ge, and J.Wang, “Public key encryption with keyword search secure against keyword guessing attacks without random oracle,” Inf. Sci., vol. 238, pp. 221–241, 2013.

7. R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, “A new general framework for secure public key encryption with keyword search,” in Information Security and Privacy - 20th Australasian Conference, ACISP, 2015, pp. 59–76**.**