IJMTARC – VOLUME – V – ISSUE – 21, JAN-MAR, 2018

ISSN: 2320-1363

A Hybrid Approach on Secure Storage of Data in the Cloud by Dual Server Public Key Encryption System

D.YADAIAH

Assistant Professor, Dept of CSE, University College for Women, Koti, Hyderabad, TS, INDIA

Abstract: Searchable encryption is of expanding enthusiasm for ensuring the information protection in secure searchable distributed storage. In this paper, we examine the security of outstanding cryptographic crude, to be specific, public key encryption with keyword search (PEKS) which is exceptionally helpful in numerous uses of distributed storage. Lamentably, it has been demonstrated that the conventional PEKS system experiences an inborn weakness called inside keyword guessing attack (KGA) propelled by the pernicious server. To address this security powerlessness, we propose another PEKS structure named dual-server PEKS (DS-PEKS). As another fundamental commitment, we characterize another variation of the smooth projective hash capacities (SPHFs) alluded to as linear and homomorphic SPHF (LH-SPHF). We at that point demonstrate a nonexclusive development of secure DS-PEKS from LH-SPHF. To delineate the attainability of our new system, we give an effective 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.

Keywords: DS-PEKS, Keyword Guessing Attack (KGA), SPHFs, LH-SPHF.

1. INTRODUCTION

Distributed storage outsourcing has transformed into a notable application for endeavors and relationship to diminish the heaviness of keeping up gigantic data recently. In any case, when in doubt, end customers may less trust the cloud limit servers and may jump at the chance to encode their data some time as of late exchanging them to the cloud server remembering the true objective to guarantee the data security. This when in doubt makes the data use more troublesome than the standard storing where data is kept in the nonattendance of encryption. One of the standard plans is the searchable encryption which allows the customer to recuperate the encoded reports that contain the customer decided watchwords, where given the catchphrase trapdoor, the server can find the data required by the customer without deciphering.

Searchable encryption can be recognized in either symmetric on the other hand veered off encryption setting. In [2], Song et al. proposed watchword look on figure content, known as Searchable Symmetric Encryption (SSE) and along these lines a couple of SSE designs [3], [4] were expected for upgrades. In spite of the way that SSE designs acknowledge high capability, they encounter the evil impacts of tangled riddle key apportionment. Accurately, customers need to securely share puzzle keys which are used for data encryption. Else they are not prepared to share the mixed data outsourced to the cloud. To decide this issue, Boneh et al. [5] exhibited a more versatile crude, to be particular Public Key Encryption with Keyword Search (PEKS) that engages a customer to look for encoded data in the amiss encryption setting. In a PEKS system, using the authority's open key, the sender joins some encoded watchwords (allowed to as PEKS figure writings) with the encoded data. The recipient by then sends the trapdoor of a to-be-looked for catchphrase to the server for data chasing. Given the trapdoor and the PEKS figure message, the server can test whether the catchphrase crucial the PEKS cipher text is identical to the one picked by the beneficiary. Given this is valid; the server sends the planning mixed data to the beneficiary.

Regardless of being free from riddle key course, PEKS designs encounter the evil impacts of an inherent insecurity concerning the trapdoor catchphrase security, to be particular inside Keyword Guessing Assault (KGA). The reason inciting to such a security weakness is, to the point that any individual who knows recipient's open key can make the PEKS figure content of self-emphatic watchword himself. Specifically, given a trapdoor, the adversarial server can pick a conjecturing catchphrase from the watchword space and after that usage the catchphrase to create a PEKS figure content. The server at that point can test whether the conjecturing catchphrase is the one fundamental the trapdoor. This theorizing then-testing system can be repeated until the point when the correct catchphrase is found. Such an estimating attack has furthermore been considered in various watch word based systems. In any case, the strike can be impelled more beneficially against PEKS designs since the watchword space is by and large the same as a standard word reference (e.g., all the vital English words), which has a significantly humbler size than a watchword dictionary (e.g., each one of the words Containing 6 alphanumeric characters). It is noteworthy that in SSE designs, just riddle key holders can create the watchword figure content and from this time forward the hostile server isn't prepared to dispatch within KGA. As the watchword reliably demonstrates the assurance of the customer data, it is along these lines of helpful importance to beat this security risk for secure searchable encoded data outsourcing.

2. DS-PEKS FRAMEWORK

Contrasted with [1], we have reexamined and propelled the work significantly in the going with points of view. In any case, in the preliminary work [1] where our non particular DS-PEKS advancement was shown, we demonstrated neither a strong improvement of the straight what's more, homomorphism SPHF nor a sensible instantiation of the DS-PEKS structure. To fill this hole and diagram the believability of the framework, in this paper (Section 6), we regardless show that a direct and homomorphism tongue LDH can be gotten from the Diffie-Hellman supposition and by then form a strong direct and homomorphism SPHF, suggested as SPHFDH, from LDH. We give a formal confirmation that SPHFDH is correct, smooth and pseudo-unpredictable improvement. We at that point display a strong DS-PEKS plot from SPHFDH. To examine its execution, we first give a connection between's current designs and our arrangement and after that evaluate its execution in trials. We too reexamined the preliminary adjustment [1] to update the introduction what's more, weightiness. In the related work part, broke down to the preliminary version, we incorporate more composed works and give a clearer portrayal of the present designs in light of their security. We show the security models of DS-PESK as tests to make them clearer. Furthermore, to make the thoughts of SPHF and our as of late portrayed vary clearer, we incorporate Fig. 4 and Fig. 5 to feature their key properties.

A DS-PEKS plot fundamentally contains (KeyGen, DSPEKS, DS-Trapdoor; Front Test; BackTest). To be more correct, the KeyGen estimation makes general society/private key arrangements of the front and back servers instead of that of the authority. Additionally, the trapdoor time computation DS-Trapdoor described here is open while in the standard PEKS definition [5], [13], the count Trapdoor takes as information the gatherer's private key. Such a qualification is relied upon to the various structures used by the two systems. In the standard PEKS, since there is only a solitary server, if the trapdoor period figuring is open, at that point the server can dispatch a theorizing ambush against a catchphrase ciphertext to recover the mixed catchphrase. Accordingly, it is hard to achieve the semantic security as described in [5], [13]. In any case, as we will seem later, under the DS-PEKS framework, we can regardless achieve semantic security when the trapdoor time figuring is open. Another qualification between the standard PEKS and our DS-PEKS is that the test count is disengaged into two figurings, FrontTest and BackTest continue running by two free servers. This is essential for achieving security against within watchword conjecturing attack.

In the DS-PEKS system, in the wake of getting an inquiry from the authority, the front server pre-frames the trapdoor what not the PEKS figure writings using its private key, and a short time later sends some inside testing-states to the back server with the contrasting trapdoor and PEKS figure writings concealed. The back server would then be able to pick which reports are addressed by the gatherer using its private key and they got inside testing-states from the front server.

Regardless, Setup is performed and the structure parameters are made. In light of the made structure parameters, interchange frameworks are executed. It is depicted underneath:

1) Setup (1). Takes as info the security parameter, creates the framework parameters P.

2) KeyGen (P): Takes as info the frameworks parameters P, yields the public/mystery key sets (pkFS; skFS), and (pkBS; skBS) for the front server, and the back server individually.

3) DS-PEKS (P; pkFS; pkBS; kw1): Takes as information P, the front server's public key pkFS, the back server's public key pkBS and the keyword kw1, yields the PEKS ciphertext CTkw1 of kw1.

4) DS-Trapdoor (P; pkFS; pkBS; kw2): Takes as information P, the front server's public key pkFS, the back server's public key pkBS and the keyword kw2, yields the trapdoor Tkw2.

5) FrontTest (P; skFS; CTkw1 ; Tkw2): Takes as information P, the front server's mystery key skFS, the PEKS ciphertext CTkw1 and the trapdoor Tkw2 , yields the interior testing-state CITS.

6) BackTest (P; skBS; CITS): Takes as information P, the back server's mystery key skBS and the inside testing-state CITS, yields the testing result 0 or 1

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

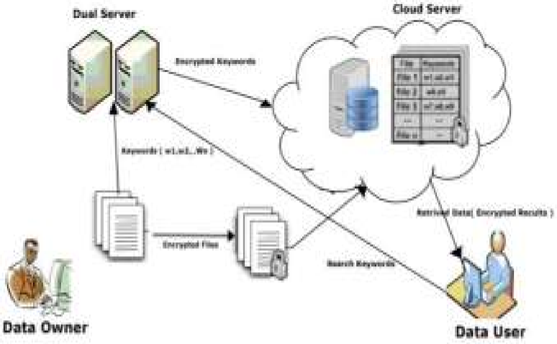

Fig -1: Dual-Server Architecture

The figure 1 shows the architecture of the new proposed scheme DS-PEKS which have two servers front server and back server. 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 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:** Front server and back server encrypt the keywords sent by the user and owner using their public keys simultaneously. Next front server will do testing on user keyword using its private key and send these testing's to the back server. Then using private key the back server 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 cipher texts using its private key, and then forwards some internal testing-states to the back server with the corresponding trapdoor and PEKS cipher texts hidden.

Back Server: In this module, the back server 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.

3. LITERATURE SURVEY

Cloud computing speaks to the present most energizing computing design move in data innovation [1]. Yet, security and protection are seen as essential hindrances to its vast adoption[2]. Here, layout a few basic security challenges and persuade encourage examination of security answers for a reliable public cloud condition [3]. Cloud computing is the most recent idea for the since quite a while ago envisioned vision of computing 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 insurance perils. Regardless, this commonly recommends one needs to give up handiness for security. For example, if a client wishes to recuperate just reports containing certain words, it was not previously known how to let the data amassing server play out the request and answers the inquiry without loss of data mystery [4].

The issue of looking for on data that is encoded using a public open key system. Consider customer Bob who sends email to customer Alice mixed under Alice's open key. An email entry needs to test whether the email contains the watchword \urgent" with the objective that it could course the email as requirements be. Alice, of course does not wish to enable the way to unscramble each one of her messages. We done and build up an instrument that engages Alice to give a key to the gateway that enables the way to test whether the word \urgent" 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 straightforwardly 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 keyword which is we need to search [5].

The conventional property in this arrangement allows the server to examine for a catchphrase, given the trapdoor. Consequently, the verifier can simply use an untrusted server, which makes this thought greatly practical. Taking after Boneh et al's. Work, there have been resulting works that have been proposed to update this thought. Two key thoughts fuse the gathered catchphrase hypothesizing strike and secure channel free, proposed by Byun et al. likewise, Baek et al., independently. The past comprehends the path that before long, the space of the catchphrases used is amazingly obliged, while the last thinks about the departure of secure channel between the recipient and the server to make PEKS practical. Heartbreakingly, the present improvement of PEKS secure against catchphrase theorizing strike is simply secure under the sporadic 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 ciphertext strike, and against watchword theorizing attacks, in spite of the way that these musings seem, by all accounts, to be the most businesslike utilization of PEKS primitives [6].

Another 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 strike) by portraying another security illustrate 'novel ciphertext vagary [7].

In 2000, to accomplish this errand, Song et al. [1] first proposed the idea of searching the encoded information with specific words. It was the primary searchable symmetric encryption conspires. In that there are two approaches to search on the ciphertext, which is to develop a list for each word W and play out a successive sweep without a list. In another needn't bother with second space to store the record, however it is slower. A few other Searchable Symmetric Encryption (SSE) schemes and a while later a couple of SSE designs [2], [3] were proposed for upgrades.

Therefore, Boneh et al. additionally proposed another plan that searches the scrambled information in view of keyword [4]. It was the principal uneven searchable encryption arranged by Boneh et al. [4], public key encryption with keyword Search. This calculation can identify which encoded outsourced document has a particular keyword without letting different gatherings, for example, Cloud Service suppliers and unapproved clients to master anything all through search and recovering procedure.

In [5] Baek et al. who enhanced PEKS plot into a safe channel (SSL) free PEKS conspire (SCF-PEKS) which disposes of a thought of secure channel (SSL) amongst clients and a server. In SCF-PEKS conspire, the information proprietor utilizes the server's public key and beneficiary's public key to encode the keywords each time he stores the scrambled data to the server. At whatever point a beneficiary or information client needs to search the encoded information connected with a particular keyword, the information client can send the trapdoor (questioned keyword) to get the information by means of a public system since just the server has the coordinating private key which can test whether the PEKS ciphertext matches the trapdoor. Be that as it may, the trapdoors can be caught by the outside attackers can infer the inserted keyword in light of the fact that the trapdoor moved in the public system.

In 2006, Byun et al. [6] called attention to that PEKS may be attacked by the disconnected keyword-guessing attacks. Since keywords are picked over essentially significantly littler space than passwords and clients more often than not utilize well-known keywords (low entropy) for looking for data [6]. Along these lines, attackers can catch the trapdoor and have opportunity to assume keyword.

In 2008, Yau et al. [7] likewise showed that outside attackers that catch the trapdoors sent in a public channel can uncover scrambled keywords by performing disconnected keyword guessing attacks.

In 2013 , Xu et al. [8] proposed a novel idea called public-key encryption with fluffy keyword search (PEFKS), by which the un-trusted server just achieves the fluffy search trapdoor as a substitution for of the correct search trapdoor, and characterize its semantic security under picked keyword attack (SS-CKA) and lack of definition of keywords under non-adaptively picked keywords and keyword guessing attack. PEFKS is the main plan to oppose against keyword guessing.

In 2010, Rhee et al. [9] examined a protected searchable public key encryption conspire with an assigned analyzer (dPEKS). They improved the current security model to join the sensible capacities of dPEKS attackers and presented t "trapdoor indistinctness" This plan is the main dPEKS technique that is secure against keyword-guessing attacks.

V. CONCLUSION

The current systems on keyword-based encryption, which are generally utilized on the plaintext information, can't be specifically connected on the scrambled information. Downloading every one of the information from the cloud and unscramble locally is clearly unfeasibly .keeping in mind the end goal to enhance the security issues in the cloud conditions; we proposed a productive plan that anticipates inside keyword guessing attack known as Dual-Server Public Key Encryption with Keyword Search (DS-PEKS) by means of smooth projective hash capacities. Our proposed framework is proficient and financially savvy. As a future work, the venture can be reached out to lessen high cost of calculation for executing trapdoors and cipher texts and we can add any new calculations to give greater security. As an expansion document content key can be recovered through collector's mail id.

REFERENCES

[1] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "A new general framework for secure public key encryption with keyword search," in Proc. 20th Australasian Conf. Inf. Secur. Privacy (ACISP), 2015, pp. 59–76.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] R. Gennaro and Y. Lindell, "A framework for passwordbased authenticated key exchange," 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 searchable audit log," in Proc. NDSS, 2004, pp. 1–11.

[8] 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.

[9] D. Khader, "Public key encryption with keyword search based on K-resilient IBE," in Proc. Int. Conf. Comput. Sci. Appl. (ICCSA), 2006, pp. 298–308.

[10] 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.

[11] 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.

[12] C. Cocks, “An identity based encryption scheme based on quadratic residues,” in Cryptography and Coding. Cirencester, U.K.: Springer, 2001, pp. 360–363.

[13]J. Baek, R. Safavi-Naini, and W. Susilo, “Public key encryption with keyword search revisited,” in Computational Science and Its Applications - ICCSA, 2008, pp. 1249–1259.

[14]H. S. Rhee, J. H. Park,W. Susilo, and D. H. Lee, “Improved searchable public key encryption with designated tester,” in ASIACCS, 2009, pp. 376–379.

[15]K. Emura, A. Miyaji, M. S. Rahman, and K. Omote, “Generic constructions of secure-channel free searchable encryption with adaptive security,” Security and Communication Networks, vol. 8, no. 8, pp. 1547– 1560, 2015.