JOURNAL FOR INNOVATIVE DEVELOPMENT IN PHARMACEUTICAL AND TECHNICAL SCIENCE

Volume-2,Issue-4 (Apr-2019)
ISSN (O) :- 2581-6934

Secured Cloud Based Quick Search for Temporary Keyword

Ms.Y.U.Selvalaxmi [1], R.A.Anushma [2]
[1]Assistant Professor ME , [2]Student ME
[1 2] Computer Science and Technology, Rajas Institute of technology for women.

Abstract— Cloud computing makes computer system resources, especially storage and computing power, available on demand without direct active management by the user. The cloud providers are not fully trusted. So, it is necessary to outsource data in the encrypted form. A new cryptographic primitive called key policy attribute based temporary keyword search is introduced. By using the key policy only, the extracted keywords are required to obtain the files from the cloud. In the proposed system, along with the key policy the quick search algorithm is introduced. It provides the searching process very quickly. It is used to find the extract keyword that is matched with the keyword that was stored in the cloud to obtain the files in the secure manner. It provides more security and requires less search time compared with the existing systems.

Keywords—Searchable encryption, attribute-based encryption, provable security, temporary keyword search, cloud security

1. INTRODUCTION

These days, cloud computing plays an imperative role in our daily life, sinceit provides efficient, reliable and scalablepossessionsfor data storage and computational activities at a very low price. However, the direct access of the cloud to the sensitive dataof its users threatens their privacy. A trivial solution to address this problem is encrypting data before outsourcing it to the cloud. However, searching on the encrypted data is very difficult. Public key encryption with keyword search (PEKS) is a cryptographic primitive which was first introduced by Boneh et al. [1] to facilitate searching on the encrypted data. In PEKS, each data owner who knows the public key of the intended data user generates a searchable cipher text by means of his/her public key, and outsources it to the cloud. Then, the data user extracts a search token related to an arbitrary keyword by using his/her secret key, and issues it to the cloud. The cloud service provider (CSP) runs the search operation by using the received search token on behalf of the data user to find the relevant results to the intended keywords. Zheng et al. [2] introduced the notion of attribute-based keyword search (ABKS) to allow a data owner to control the access of data users for searching on his/her outsourced encrypted data.

They used attribute-based encryption (ABE) [3] to construct a searchable cryptographic primitive in the multi-sender/multi receiver model. In their work, the legitimate data users can enlist the cloud to run the search operation on behalf of them without requiring any interaction with the data owner. In a secure ABKS scheme, a data owner cannot obtain any information about the keywords which the data users intend to look for. However, in all of the PEKS and ABKS schemes, once the cloud receives a valid search token related to a certain keyword, the cloud can investigate the keyword's presence in the past and any future cipher text. So, if the adversary realizes the corresponding keyword of the target search token, then she will be able to get some information about the next documents which will be outsourced to the cloud.

Therefore, it will be more secure to limit the time period in which the search token can be used. Motivated by this problem, Abdalla et al. [4] introduced the notion of public key encryption with temporary keyword search (PETKS) which restricts the validation of the token to a certain time period. They applied anonymous identity-based encryption [5] in their generic scheme. In addition, Yu et al. [6] proposed another public key searchable encryption in the context of temporary keyword search. Despite the good features of their schemes, these schemes do not provide the facility for data owners to enforce their intended access policy. In this paper,

we propose a novel notion of Key-Policy Attribute-Based Temporary Keyword Search (KP-ABTKS).

In KP-ABTKS schemes, the data owner generates a searchable cipher text related to a keyword and the time of encrypting according to an intended access control policy, and outsources it to the cloud. After that, each authorized data user selects an arbitrary time interval and generates asearch token for the intended keyword to find the ciphertext.Then, he/she sends the generated token to the cloud to run the search operation. By receiving the token, the cloud looks for the documents contain the intended keyword. The search result on a ciphertext is positive, if

(i) The data user's attributes satisfy the access control policy,

(ii) The time interval of the search token encompasses the time of encrypting, and

(iii) The search token and the ciphertext are related to the same keyword. To show that the proposed notion can be realized,

we also propose a concrete instantiation for this new cryptographic primitive based on bilinear map.

Related Work

Due to the increasingeminenceof cloud computing, an ever accumulativenumber of information proprietors are inspired to re-fitting their information to cloud servers for incredible comfort and decreased expense in infotheadministrators. Be that as it may, delicate information ought to be encoded previously re-appropriating for security necessities, which obsoletes information usage like keyword based archive recovery. In this paper, we present a safe multikeyword ranked search conspire over scrambled cloud information, which all the while underpins dynamic update activities like erasure and addition of archives. In particular, the vector space demonstrates and the generally utilized TF *IDF show are joined in the file development and inquiry age.

We build an uncommon tree-based file structure and propose an "Greedy Depth-first Search" calculation to give proficient multi-keyword ranked search. Ciphertext-Policy Attribute - Based Encryption (CP-ABE) permits to scramble information under an get to policy, determined as a consistent mix of attributes. Such ciphertexts can be unscrambled by anybody with a lot of attributes that fulfill the entrance policy.

We propose a Ciphertext-Policy Attribute-Based Encryption, which is based on an ongoing secret sharing technique called Linear Integer Secret Sharing Scheme (LISS). In this scheme, the encryptor can determine the entrance policy regarding LISS grid M, over the attributes in the framework. The scheme is specifically secure under Decisional Bilinear Diffie– Hellman (DBDH) presumption.

In previous privacy-preserving multi-authority attribute-based encryption (PPMA-ABE) plans, a client can procure mystery keys from numerous specialists with them knowing his/her qualities and moreover, a focal expert is required. Strikingly, a client's personality data can be extricated from his/her some delicate traits. Henceforth, existing PPMAABE plans can't completely secure clients' protection as different experts can team up to distinguish a client by gathering and examining his properties. Also, ciphertext-arrangement ABE (CPABE) is a progressively effective open key encryption where the encryptor can choose adaptable access structures to encode messages.

Along these lines, a testing and critical work is to develop a PPMA-ABE conspire where there is no need of having the focal expert and besides, both the identifiers and the ascribes can be ensured to be known by the specialists.

A security safeguarding decentralized CP-ABE (PPDCPABE) is proposed to diminish the trust on the focal expert furthermore, secure clients' protection. In our PPDCP-ABE conspire, each specialist can work freely with no joint effort to starting the framework and issue mystery keys to clients. Besides, a client can get mystery keys from numerous specialists without them knowing anything about his worldwide identifier (GID) and qualities.

Proposed System

In the proposed system, Key policy Attribute-Based Temporary Keyword Search (KP-ABTKS) was used. This scheme consists of four entities including data owner, data user, cloud server and Trusted Third Party (TTP).Each data owner according to an access control policy generates a searchable cipher text based on an arbitrary keyword and the time of encrypting. Each data user for searching a keyword in

ISSN:-2581-6934|www.jidps.com

a specific time interval, generates a search token which is valid just for that time interval. The data users can generate the search tokens without interacting with the data owners. The cloud server based on the received search token can find the encrypted documents which contain the intended keyword and are generated in the specified time interval. Then, it returns the search result to the data users whose attributes satisfy the access control policy enforced by the data owner

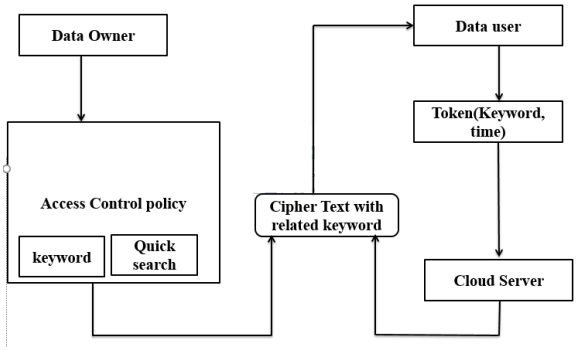

Fig: Architecture Diagram

Quick search algorithm with KB-ABTKS are introduced. TTP generate the key for both the data owner and data user. By using the key, the data owner uploads the data in the cloud. The data user generates the token by using the key provided by the data owner and the quick search algorithm is implemented.It provides the better search compared to the existing system. It reduces the searching time and generate the token in very less time.

Key-policy attribute based temporary keyword search (KP-ABTKS)

In this area, we propose new "Key-Policy Property Based Temporary Keyword Search (KP-ABTKS)". This plan comprises of four substances including information proprietor, information client, cloud server and Trusted Third Party (TTP) which are depicted as pursues:

1) Data proprietor: Is an element who encodes its archives under a self-assertive access control strategy and redistributes them to the cloud. They thinks about the season of encoding in creating the ciphertexts. We should feature that the information proprietor additionally encodes his/her reports under his/her subjective access control arrangement. Notwithstanding, in this paper we focus on the encryption of the separated watchwords from reports.

2) Data client: Is a substance that is searching for archives which contains an expected watchword, and is encoded in a decided time interim. The time interim is subjectively chosen by the information client.

3) Cloud Server (CS): Is a substance with amazing calculation and capacity assets. CS stores a monstrous sum of encoded information, and gets the pursuit tokens to look for the required records in the interest of the information client. The cloud finds the applicable records, and sends them back to the information client.

4) Trusted Third Party (TTP): Is a completely confided in substance who gets every client's entrance tree, and produces their mystery keys comparing to his/her qualities set exhibited in his/her entrance tree. At that point, the TTP sends back the clients' certifications through a safe and confirmed channel. Every data owner as indicated by an entrance control arrangement produces an accessible ciphertext dependent on a self-assertive watchword and the season of scrambling.

Every data client for looking through a temporary keyword search in an explicit time interim, produces a seek token which is substantial only for that time interim. The information clients can create the hunt tokens without collaborating with the information proprietors. The cloud server dependent on the got pursuit token can discover the encoded archives which contain the planned watchword and are produced in the predefined time interim. At that point, it restores the output to the information clients whose properties fulfill the entrance control arrangement upheld by the data owner. Formal meaning of KP-ABTKS The proposed KP-ABTKS plot comprises of five calculations, Setup; KeyGen; Enc; TokenGen; Search. These calculations are depicted as pursues:

• (msk, pp) ←Setup (1 λ): This calculation is controlled by the TTP. It takes the security parameter λ as info and produces the ace mystery key msk and the general population parameter pp.

• sk← KeyGen(msk; Tr): This calculation creates a mystery key sk for the client with the entrance tree, Tr. The TTP decides the entrance tree Tr and runs this calculation.

• cph← Enc(ω; ti ; Atts; pp): This calculation produces a accessible ciphertext identified with the temporary keyword

ISSN:-2581-6934|www.jidps.com

search ! also, time of encoding ti as indicated by a characteristic set, Atts which is controlled by the information proprietor.

• st ←TokenGen(sk, ω , [ts , te]): The information client runs this calculation to produce the look token st for seeking the ciphertexts which are scrambled in the time interim [ts,te], and contain the temporary keyword search ω, as indicated by its mystery key sk.

• {0,1} := Search(cph, st): For each put away ciphertext cph furthermore, the pursuit token st which is related with explicit temporary keyword ! furthermore, trait set Atts, this calculation returns 1 if the ma|ority of the accompanying conditions are met at the same time:

◦ Tr(Atts) = 1,

◦ cph*← Enc(ω * ; ti ; Atts)

◦ st∗←TokenGen(sk; ω * ; [ts ; te])

◦ ti € [ts , te] Else, it returns 0

The Proposed Concrete Construction OFKP-ABTKS

With the inspiration of the ABKS scheme, the proposed construction is obtained. The detail of the construction is presented as follows :(msk, pp) Setup (1 λ): This is a randomized algorithm whichis run by the TTP to generate the master secret key and the publicparameters. Based on the security parameter λ, this algorithm selects a bilinear map e : G1 × G1 ← G2, where G1 and G2 are cyclic groups of orderλ-bit prime number q.

Let H1 = {0,1} * → G1 and H2 ={0,1} ∗ → Zq be two cryptographic one-way hashfunctions. It first selects P ∈R G1 as the generator of G1 and two random values, s, sr ∈R Zq . Then, it sets the public parameter and the master secret key as follows: pp := (H1,H2, e, P,sP, srP,G1, G2) msk := (s; sr) skj←KeyGen(msk,Tr): The TTP determines the accesstree of the j-th cloud user, Trj , and runs this randomized algorithm to generate his/her secrete key, skj . This algorithm runs Share(Trj, Srs−1) as a subroutine to allocate the secret share qn(0) to each leaf node n ∈ lvs(Trj) with regard to the access tree Trj . For this aim, the TTP first selects a random value tj ∈R Zq, and computes An = qn(0)P + tjH1(att(n)) and Bn = tjsP for each leaf n 2 lvs(Trj). Then, the secret key skj is set as follows: skj :=Trj,{(An,Bn)|n ∈ lvs(Trj)}

(4) cph← Enc(ω,ti,Atts,pp): The information proprietor runs this calculation on the temporary keyword ω , the time example of scrambling ti , the planned traits set Atts and general society parameters, pp as its contributions to produce a quality based accessible ciphertext for redistributing it to the cloud.

This randomized calculation chooses two irregular values r1; r2 2R Zq, and encrypts the keyword! according to the following steps: W = r1r2sP W' = r1srP W'' = r1H2(ω)sP + r1r2P W = H2(ti) ∀attj ∈ Atts : Wj = r1r2H1(attj) cph := (Atts,W,W',W'', W,{Wj|attj ∈ Atts} (5) st TokenGen(skj , ω ,Tenc = [ts ,te], pp): A data user withthe access tree Trj and the secret key skj runs this randomizedalgorithm to generate a search token for the keyword ω.

They wants to find the cipher texts including! and are encrypted ina specified time interval, Tenc = [ts ,te]. For this aim, he/sheselects z0 ∈R Zp, computes An = z0An and Bn = z0Bn for each leaf node n ∈ lvs(Trj), and finally generates the search tokens as follows: l = te –ts St(x) = H2(ω) +i=1∏j=0 (x−H2(ts + j)) = (H2(ω) + a1') + a2x +···+ alx i-1 = a1 + a2x +···+ alxl−1 st1 = {st1,j : st1,j = z0ajsP,∀j ∈ I = {1...,l} st2 = z0srP st =: (st1,st2,Trj,{(An',Bn')|n ∈ lvs(Trj)}) (6) (0,1) := Search(st, cph): This algorithm selects the largest subset S of the attribute set Atts satisfying the access tree Trj . If S is empty, this algorithm returns 0; otherwise, acts as follows: ∀attj ∈ S : En = e(A'n,W0)/e(B' n,Wj) = e(P,P)z0r1r2sqn(0) It should be mentioned that we have att(n) = att|, for n 2 lvs(Tr|). Eroot := Combine(Trj, { En|att(n) ∈ S) = e(P, P)z0r1r2sqn(0) = e(P, P)z0r1r2ss−1s r = e(P, P)z0r1r2sr (7) Then, the cloud computes st∗ as follows. St*= ∑ j=1 W j-1 st1,j (8) Finally, this algorithm returns 1 if e(W', st∗) . Eroot =e(st2; W '') and 0, otherwise.

CONCLUSION

Distributed storage is an imperative issue in cloud registering. We tended to this issue and presented the thought of keypolicy attribute based temporary keyword search (KPABTKS). As indicated by this idea, every datum client can produce an inquiry token which is substantial just temporarily interim. We proposed the main solid development for this new cryptographic crude dependent on bilinear guide. We formally demonstrated that our plan is provably secure in the arbitrary

ISSN:-2581-6934|www.jidps.com

prophet demonstrate. The multifaceted nature of encryption calculation of our proposition is direct regarding the quantity of the included characteristics. In expansion, the quantity of required matching in the inquiry calculations is free of the quantity of the expected time units determined in the hunt token and it is direct regarding the number of properties. Execution assessment of our plan in term of both computational expense and execution time demonstrates the down to earth parts of the proposed plan.

REFERENCES

[1] Q. Zheng, S. Xu, and G. Ateniese, "Vabks: Verifiable attribute-based keyword search over outsourced encrypted data," in INFOCOM, 2014 Proceedings IEEE. IEEE, 2014, pp. 522–530.

[2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Advances in Cryptology–EUROCRYPT 2005. Springer, 2005, pp. 457–473.

[3] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions," in Advances in Cryptology–CRYPTO 2005. Springer, 2005, pp. 205–222.

[4] Y. Yu, J. Ni, H. Yang, Y. Mu, and W. Susilo, "Efficient public key encryption with revocable keyword search," Security and Communication Networks, vol. 7, no. 2, pp. 466–472, 2014.

[5] E.-J. Goh et al., "Secure indexes." IACR Cryptology ePrint Archive, vol. 2003, p. 216, 2003. [6] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward efficient multikeyword fuzzy search over encrypted outsourced data with accuracy improvement," IEEE Transactions on Information Forensics and Security, vol. 11, no. 12, pp. 2706–2716, 2016.

[7] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multikeyword ranked search over encrypted cloud data," IEEE Transactions on parallel and distributed systems, vol. 25, no. 1, pp. 222–233, 2014.

[8] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. S. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 3, pp. 312–325, 2016.

[9] A. Awad, A. Matthews, Y. Qiao, and B. Lee, "Chaotic searchable encryption for mobile cloud storage," IEEE Transactions on Cloud Computing, vol. PP, no. 99, pp. 1–1, 2017.

[10] J. Li, D. Lin, A. C. Squicciarini, J. Li, and C. Jia, "Towards privacypreserving storage and retrieval in multiple clouds," IEEE Transactions on Cloud Computing, vol. 5, no. 3, pp. 499–509, July 2017. [11] J. Li, R. Ma, and H. Guan, "Tees: An efficient

search scheme over encrypted data on mobile cloud," IEEE Transactions on Cloud Computing, vol. 5, no. 1, pp. 126–139, Jan 2017.

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

[13] H. Yin, Z. Qin, J. Zhang, L. Ou, and K. Li, "Achieving secure, universal, and fine-grained query results verification for secure search scheme over encrypted cloud data," IEEE Transactions on Cloud Computing, vol. PP, no. 99, pp. 1–1, 2017.

[14] K. Liang and W. Susilo, "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage," IEEE Transactions on Information Forensics and Security, vol. 10, no. 9, pp. 1981–1992, 2015.

[15] J. Han, W. Susilo, Y. Mu, and J. Yan, "Attribute-based oblivious access control," The Computer Journal, vol. 55, no. 10, pp. 1202– 1215, 2012.

[16] "Attribute-based data transfer with filtering scheme in cloud computing," The Computer Journal, vol. 57, no. 4, pp. 579–591, 2014.

[17] Y. Shi, Q. Zheng, J. Liu, and Z. Han, "Directly revocable keypolicy attribute-based encryption with verifiable ciphertext delegation," Information Sciences, vol. 295, pp. 221–231, 2015.

[18] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in Security and Privacy, 2007. SP'07. IEEE Symposium on. IEEE, 2007, pp. 321–334.

[19] H. Deng, Q. Wu, B. Qin, J. Domingo-Ferrer, L. Zhang, J. Liu, and W. Shi, "Ciphertext-policy hierarchical attribute-

based encryption with short ciphertexts," Information Sciences, vol. 275, pp. 370–384, 2014.

[20] A. Balu and K. Kuppusamy, "An expressive and provably secure ciphertext-policy attribute-based encryption," Information Sciences, vol. 276, pp. 354–362, 2014.

[21] J. Han, W. Susilo, Y. Mu, J. Zhou, and M. H. A. Au, "Improving privacy and security in decentralized ciphertext-policy attributebased encryption," IEEE Transactions on Information Forensics and Security, vol. 10, no. 3, pp. 665–678, 2015.

[22] J. Han, W. Susilo, Y. Mu, and J. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption," IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 11, pp. 2150–2162, 2012.

ISSN:-2581-6934|www.jidps.com