JCSE International Journal of Computer Sciences and Engineering Open Access

Review Paper Vol.-7, Issue-2, Feb 2019 E-ISSN: 2347-2693

Extraction of Cipher Texts for Temporary Keywords Search on Confidential Data in the Cloud

C. Pradeepthi[1*] , V.J.Vijaya Geetha[2]

[1]Dept of CSE, SPMVV, Tirupathi, AP, India, E-mail: E-mail:
[2] Dept of CSE, SPMVV, Tirupathi, AP, India

DOI: https://doi.org/10.26438/ijcse/v7i2.692696 | Available online at: www.ijcseonline.org

Accepted: 16/Feb/2019, Published: 28/Feb/2019

Abstract: Temporary keyword search seek on secret information in a cloud condition is the principle focal point of this exploration. The cloud suppliers are not completely trusted. In this way, it is important to redistribute information in the encoded frame. In the attribute based keyword search (ABKS) plans, the approved clients can produce some hunt tokens and send them to the cloud for running the pursuit task. These hunt tokens can be utilized to remove all the ciphertexts which are created whenever and contain the comparing temporary keyword search. Since this may prompt some data spillage, it is increasingly secure to propose a plan in which the hunt tokens can just concentrate the ciphertexts produced in a predefined time interim. To this end, in this paper, we present another cryptographic crude called key policy -attribute based temporary keyword search(KPABTKS) which give this property. To assess the security of our conspire, we formally demonstrate that our proposed plan accomplishes the watchword mystery property and is secure against specifically picked watchword assault (SCKA) both in the arbitrary prophet demonstrate and under the hardness of Decisional Bilinear Diffie-Hellman (DBDH) suspicion. Moreover, we demonstrate that the intricacy of the encryption calculation is direct concerning the quantity of the included qualities. Execution assessment demonstrates our plan's reasonableness.

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

I. INTRODUCTION

Distributed computing assumes a critical |ob in our day by day life, since it gives productive, dependable and adaptable assets for information stockpiling and computational exercises at an extremely low cost. In any case, the immediate access of the cloud to the touchy data of its clients undermines their protection. A unimportant arrangement to address this issue is encoding information before redistributing it to the cloud. In any case, seeking on the scrambled information is exceptionally troublesome. public key encryption with keyword search(PEKS) is a cryptographic crude to encourage looking on the scrambled information.

In PEKS, every datum proprietor who knows general society key of the expected information client creates an accessible ciphertext by methods for his/her open key, and re-appropriates it to the cloud. At that point, the information client separates a hunt token identified with a self-assertive temporary keyword search by utilizing his/her mystery key, and issues it to the cloud. The cloud service provider(CSP) runs the inquiry task by utilizing the got pursuit token in the interest of the information client to locate the important outcomes to the proposed watchwords.

In a safe ABKS conspire, an information proprietor can't get any data about the watchwords which the information clients mean to search for. In any case, in the majority of the PEKS and ABKS plans, once the cloud gets a substantial inquiry token identified with a specific temporary keyword search, the cloud can examine the watchword's quality previously also, any future ciphertext. In this way, if the enemy understands the comparing watchword of the objective inquiry token, at that point she will have the capacity to get some data about the following archives which will be re-appropriated to the cloud. In this manner, it will be progressively secure to restrain the day and age in which the pursuit token can be utilized.

Propelled by this issue, Abdalla et al. presented the thought of open key encryption with brief watchword seek (PETKS) which confines the approval of the to ken to a specific era. They connected unknown character based encryption in their nonexclusive plan. Moreover, Yu et al.proposed another open key accessible encryption in the unique situation of impermanent temporary keyword search look. In spite of the great highlights of their plans, these plans don't give the office to information proprietors to implement their expected access approach. In this paper, we propose a novel idea of Key-Policy Attribute-Based Temporary keyword Search (KP-ABTKS). In KP-ABTKS plans, the information proprietor creates an accessible ciphertext identified with a temporary keyword search and the season of scrambling as indicated by a planned access control arrangement, and redistributes it to the cloud. From that point onward, each approved information client chooses a subjective time interim and produces a scan token for the expected watchword to discover the ciphertext.

At that point, he/she sends the produced token to the cloud to run the look task. By getting the token, the cloud searches for the reports contain the expected watchword. The query item on a ciphertext is certain, on the off chance that (I) the information client's characteristics fulfills the entrance control strategy, (ii) the time interim of the pursuit token envelops the season of encoding, and (iii) the pursuit token and the ciphertext are identified with a similar temporary keyword search. To demonstrate that the proposed thought can be acknowledged, we additionally propose a solid instantiation for this new cryptographic crude based on bilinear guide.

II. RELATED WORK

Due to the expanding prominence of cloud computing, an ever increasing number of information proprietors are inspired to re-appropriate their information to cloud servers for incredible comfort and decreased expense in information the executives. 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 multi-keyword 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 demonstrate 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.

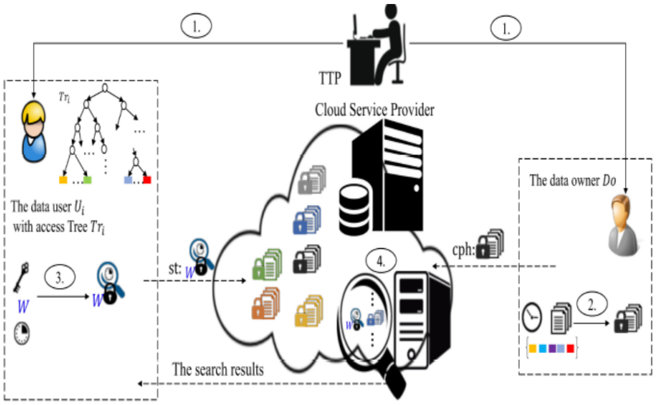
III. PROPOSED SYSTEM

1) The novel idea of KP-ABTKS, and propose a solid development for this new cryptographic crude which can be connected in the distributed storage administrations. The proposed solid plan is planned based on bilinear blending. In the proposed KP-ABTKS, each client is related to an entrance control approach. The information proprietor chooses a property set, and runs the encryption calculation with respect to it. On the off chance that an information client's traits set fulfills the entrance tree of the information proprietor, at that point he/she can produce a substantial pursuit token. The cloud applies the produced hunt token to locate the comparing ciphertexts which have been encoded in a period interim indicated by the information client.

2) We formally characterize two security definitions for KPABTKS in the standard model. One of them characterizes its security against specifically picked temporary keyword search assault (KPABTKS-SCKA), and the other one characterizes the watchword mystery of KP-ABTKS. We formally demonstrate that our proposed plan fulfills these security definitions under the hardness of the Decisional Diffe-Hellman (DDH) presumption.

3) We assess the execution of the proposed development of KP-ABTKS as far as both computational intricacy and the execution time. The execution assessment demonstrates the down to earth parts of our proposition.

IV. SYSTEM ARCHITECTURE



V. 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. He/She 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 an 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; t_i; Atts; pp): This calculation produces a accessible ciphertext identified with the temporary keyword search ! also, time of encoding ti as indicated by a characteristic set, Atts which is controlled by the information proprietor.
• st ←TokenGen(sk, ω , [t_s, t_e]): 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 got 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^{*}; t_i; Atts)
◦ st∗ ←TokenGen(sk; ω^{*}; [t_s; t_e])
◦ t_i € [t_s, t_e]
Else, it returns 0

VI. 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 order

λ-bit prime number q. Let H1 = $\{0,1\}^{*} \rightarrow$ G1 and H2 =$\{0,1\}^{*} \rightarrow$ Zq be two cryptographic one-way hashfunctions. It first selects P $\in_R$ G1 as the generator of G1 and two random values, s, s_r $\in$R Zq . Then, it sets the public parameter and the master secret key as follows:

pp := (H1,H2, e, P,sP, s_rP,G1, G2)

msk := (s; s_r)

$sk_j \leftarrow$KeyGen(msk,T_r): The TTP determines the accesstree of the j-th cloud user, Tr_j, and runs this randomized algorithm to generate his/her secrete key, sk_j. This algorithm runs Share($Tr_{j,}$ S_rs−1) as a subroutine to allocate the secret share qn(0) to each leaf node n ∈ lvs(Trj) with regard to the access tree Tr_j. For this aim, the TTP first selects a random value tj ∈R Zq, and computes An = qn(0)P + tjH1(att(n)) and Bn = t_jsP for each leaf n 2 lvs(Tr_j). Then, the secret key sk_j 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 t_i, 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 = r1r_2sP

W’ = r1s_rP

W’’ = r1H2(ω)sP + r_1r_2P

W = H2(ti)

∀attj ∈ Atts : Wj = r1r2H1(attj) cph := (Atts,W,W',W", W,{Wj|attj ∈ Atts} (5)

st TokenGen(sk_j , ω ,Tenc = [t_s,t_e], pp): A data user withthe access tree Tr_j and the secret key skj runs this randomizedalgorithm to generate a search token for the keyword ω. He/Shewants to find the ciphertexts including ! and are encrypted ina specified time interval, T_{enc} = [t_s,t_e]. For this aim, he/sheselects z0 ∈R Zp, computes $A_n = z_0A_n$ and $B_n = z_0B_n$ for each leaf node n ∈ lvs(Tr_j), and finally generates the search tokens as follows:

l = te –ts

$St(x) = H2(\omega) +^{i=1}\prod_{j=0} (x-H_2(t_s + j)) = (H_2(\omega) + a_{1'}) + a_2x +\cdots+ a_lx^{i-1} = a_1 + a_2x +\cdots+ a_lx_l-1$

st_1 = {st1,j : st1,j = z0ajsP,∀j ∈ I = {1...,l}}

st2 = z_0s_rP

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 Tr_j. If

S is empty, this algorithm returns 0; otherwise, acts as follows:

∀attj ∈ S :

$E_n = e(A'n,W_0)/e(B' n,W_j) = e(P,P)^{z0r1r2sqn(0})$

It should be mentioned that we have att(n) = att|, for n 2 lvs(Tr|).

E_{root} := Combine($Tr_{j,}$ { E_n|att(n) ∈ S)

= $e(P, P)^{z0r1r2sqn(0)}$

= $e(P, P)^{z0r1r2ss-1s}{}_r$

= $e(P, P)^{z0r1r2s}$r (7)

Then, the cloud computes st∗ as follows.

$St^{*=} \sum_{j=1} W^{j-1} st_{1,j}$ (8)

Finally, this algorithm returns 1 if e(W', $st^{*)}$.

E_{root} =e(st2; W '') and 0, otherwise.

VII. PERFORMANCE

The KP-ABTKS scheme consists of five algorithms: Setup, KeyGen, Enc, TokenGen and Search. Since the Setup algorithm is run offline, we exclude its computational cost in analyzing the performance of our scheme. Suppose that N is the number of the attributes which are chosen by the data owner; and |S| denotes the number of the attributes which are appeared in each access tree. In the KeyGen algorithm,|S| hash functions and 3|S| modular exponentiations in G1 are run. The Enc algorithm requires to execute (4 + N) modular exponentiations in G1 and (N + 2) hash functions. In the Token Gen algorithm, (2|S| + l + 1) modular exponentiation sin G1 and l hash functions are computed. Finally, the Search algorithm is executed by running 2(N + 1) pairings and l exponentiations. The computational cost of the Encalgorithm is linear with respect to the number of the intended attributes, N. Moreover, the number of required pairings in the Search algorithm is also linear with respect to the number of involved attributes, |S|. One of the salients features of our proposed temporary keyword search scheme is that the number of required pairings in the search algorithm is independent of the number of time units which are considered in the search token by the data user.

The storage overhead for each user is equal to 2|S|(log2 |G1|) where |G1| is the cardinality of the group G1. Besides, the communication overhead can be computed by adding the ciphertext size, (N + 4) log2 |G1|, and the search token size, (2|S| + l + 1) log2 |G1|.

With regards to brief temporary keyword search , we can infer to people in general key encryption with temporary keyword search seek (PETKS) and the Yu et al's. plot. These two plans have a place to the variation of conventional open key accessible encryption plans. Not at all like our proposed plan and the PETKS plot, in the presented plan in [6], the creators proposed a multikeyword open key accessible encryption conspire in which the seek token is pertinent to discover the ciphertext in extraordinary time occasion rather than a period interim.

In multi-client situations, for example, systems contain an extensive number of information clients, when the information proprietor intends to share the reports alongside a lot of approved information clients by utilizing two referenced plans ([4] and [6]), it very well may be seen that the request of computational intricacy is straight with the quantity of expected information clients. Thus, when the quantity of approved clients is expansive, these plans require overwhelming calculations. Be that as it may, our proposed conspire is progressively relevant in the huge systems, in light of the fact that the information proprietor can encode the reports utilizing an encryption calculation which its computational multifaceted nature is free of the number of approved information clients and is direct as for the number of properties. Regularly, the quantity of properties are restricted and we can compose a boundless number of clients with the current traits set. Along these lines, it tends to be reasoned that in the multi-client setting the computational multifaceted nature of our plan is much lower than [4] and [6]. Likewise, to the best of our insight, our plan is the main quality based accessible encryption conspire which underpins brief inquiry property.

VIII. CONCLUSION

Distributed storage is an imperative issue in cloud registering. We tended to this issue and presented the thought of key-policy 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 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 key-policy 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 attribute-based 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.