Protecting Your Data: Attribute-Based Keyword Search Authorization in the Cloud Environment

[#1]p.Archana, M.Tech Student, [#2]dr.M.Sujatha, Assoc Professor,
Department Of Cse, Jyothishmathi Institute Of Technological Sciences, Karimnagar T.S.India.
Corresponding Author: p.Archana, M.Tech Student

Abstract: *Searchable encryption (SE) has been a promising technology which allows users to perform search queries over encrypted data. However, the most of existing SE schemes cannot deal with the shared records that have hierarchical structures. In this paper, we devise a basic cryptographic primitive called as attribute-based keyword search over hierarchical data (ABKS-HD) scheme by using the ciphertext-policy attribute-based encryption (CP-ABE) technique, but this basic scheme cannot satisfy all the desirable requirements of cloud systems. In this paper, we focus on a different yet more challenging scenario where the outsourced dataset can be contributed from multiple owners and are searchable by multiple users, i.e. multi-user multicontributor case. Inspired by attribute-based encryption (ABE), we present the first attribute-based keyword search scheme with efficient user revocation (ABKS-UR) that enables scalable fine-grained (i.e. file-level) search authorization. Our scheme allows multiple owners to encrypt and outsource their data to the cloud server independently. Users can generate their own search capabilities without relying on an always online trusted authority. Fine-grained search authorization is also implemented by the owner-enforced access policy on the index of each file. Further, by incorporating proxy re-encryption and lazy reencryption techniques, we are able to delegate heavy system update workload during user revocation to the resourceful semitrusted cloud server. We formalize the security definition and prove the proposed ABKS-UR scheme selectively secure against chosen-keyword attack. Finally, performance evaluation shows the efficiency of our scheme.*

Index Terms*—, hierarchical structures, ciphertext-policy, chosen-plaintext attack, Searchable encryption, attribute-based encryption,, chosen-keyword attack.*

Date of Submission: 19-07-2018 Date of acceptance: 02-08-2018

I. Introduction

Cloud computing has become a promising technology due to its impressive features, i.e., large storage capacity and flexible accessibility. By outsourcing the sensitive data to a cloud server, individuals and enterprises are relieved from the burden of local data management and maintenance. However, as data owners cannot have the full physical control over their data, data security and privacy concerns remain significant barriers to the adoption of cloud computing. The basic idea is to encrypt the shared data before outsourcing them to the cloud servers, whereas the encryption mechanism limits the flexibility of data retrieval to some extent. In addition, it is a naive solution to download all ciphertexts and decrypt them locally because this will incur a waste of computation and bandwidth resources. Accordingly, how to securely and efficiently retrieve cloud data is of prime importance in the scenarios of cloud storage [2], [3]. To solve the problem of searching over encrypted data, the SE technique [4], [5], [6], [7], [8], which allows cloud server to retrieve encrypted data on behalf of data owners without loss of data confidentiality, has made specific contributions in terms of security, efficiency and functionality. So far, a lot of work under various security models has been proposed in order to gain different search functionalities, such as single keyword search, multi-keyword search, fuzzy keyword search, etc. Although the SE technique has attracted much attention in the industrial and academicals fields over the last decade, it is still not sufficient as data owners also want to achieve the fine-grained data sharing and decentralized access control. To the best of our knowledge, the traditional server-based access control mechanisms are no longer suitable for cloud storage as the cloud server cannot be completely trusted by data owners. At present, the CP-ABE technology [9], which can gain one-to-many encryption rather than one-to-one, has turned to be a viable tool to tackle the problem of fine-grained access control.

This paper focuses on the problem of search over encrypted data, which is an important enabling technique for the encryption-before-outsourcing privacy protection paradigm in cloud computing, or in general in any networked information system where servers are not fully trusted. Much work has been done, with majority focusing on the single-contributor scenario, i.e. the dataset to be searched is encrypted and managed by a single entity, which we call owner or contributor in this paper. Under this setting, to enable search over encrypted data, the owner has to either share the secret key with authorized users [4], [7], [8], or stay online to generate the search trapdoors, i.e. the “encrypted” form of keywords to be searched, for the users upon request [9], [10]. The same symmetric key will be used to encrypt the dataset (or the searchable index of the dataset) and to generate the trapdoors. These schemes seriously limit the users’ search flexibility.

Consider a file sharing system that hosts a large number of files, contributed from multiple owners and to be shared among multiple users (e.g. 4shared.com, mymedwall.com). This is a more challenging multi-owner multi-user scenario. How to enable multiple owners to encrypt and add their data to the system and make it searchable by other users? Moreover, data owners may desire fine-grained search authorization that only allows their authorized users to search their contributed data. By fine-grained, we mean the search authorization is controlled at the granularity of per file level. Symmetric cryptography based schemes [4], [7], [8] are clearly not suitable for this setting due to the high complexity of secret key management. Although authorized keyword search can be realized in single-owner setting by explicitly defining a server-enforced user list that takes the responsibility to control legitimate users’ search capabilities [11], [12], i.e. search can only be carried out by the server with the assistance of legitimate users’ complementary keys on the user list, these schemes did not realize fine-grained owner-enforced search authorization and thus are unable to provide differentiated access privileges for different users within a dataset. Asymmetric cryptography is better suited to this dynamic setting by encrypting individual contribution with different public keys. For example, Hwang et al. [13] implicitly defined a user list for each file by encrypting the index of the file with all the public keys of the intended users. However, extending such user list approach to the multiowner setting and on a per file basis is not trivial as it would impose significant scalability issue considering a potential large number of users and files supported by the system. Additional challenges include how to handle the updates of the user lists in the case of user enrollment, revocation, etc., under the dynamic cloud environment.

In this paper, we address these open issues and present an authorized keyword search scheme over encrypted cloud data with efficient user revocation in the multi-user multi-data contributor scenario. We realize fine-grained owner-enforced search authorization by exploiting ciphertext policy attribute based encryption (CP-ABE) technique. Specifically, the data owner encrypts the index of each file with an access policy created by him/her, which defines what type of users can search this index. The data user generates the trapdoor independently without relying on an always online trusted authority (TA). The cloud server (CS) can search over the encrypted indexes with the trapdoor on a user’s behalf, and then returns matching results if and only if the user’s attributes associated with the trapdoor satisfy the access policies embedded in the encrypted indexes. We differentiate attributes and keywords in our design. Keywords are actual content of the files while attributes refer to the properties of users. The system only maintains a limited number of attributes for search authorization purpose. Data owners create the index consisting of all keywords in the file but encrypt the index with an access structure only based on the attributes of authorized users, which makes the proposed scheme more scalable and suitable for the large scale file sharing system. In order to further release the data owner from the burdensome user membership management, we use proxy re-encryption [14] and lazy re-encryption [15] techniques to shift the workload as much as possible to the CS, by which our proposed scheme enjoys efficient user revocation. Formal security analysis shows that the proposed scheme is provably secure and meets various search privacy requirements. Performance evaluation also demonstrates its efficiency and practicality.

Our contributions can be summarized as follows:

1) We design a novel and scalable authorized keyword search over encrypted data scheme supporting multiple data users and multiple data contributors. Compared with existing works, our scheme supports fine-grained owner-enforced search authorization at the file level with better scalability for large scale system in that the search complexity is linear to the number of attributes in the system, instead of the number of authorized users.
2) Data owner can delegate most of computationally intensive tasks to the CS, which makes the user revocation process efficient and is more suitable for cloud outsourcing model.
3) We formally prove our proposed scheme selectively secure against chosen-keyword attack.

II. Related Work

A. Keyword Search over Encrypted Data

1) Secret key vs. Public key: Encrypted data search has been studied extensively in the literature. Song et al. [4] designed the first searchable encryption scheme to enable a full text search over encrypted files. Since this seminal work, many secure search schemes have been proposed to boost the efficiency and enrich the search functionalities based on either secret-key cryptography (SKC) or public key cryptography (PKC) presented an efficient single keyword encrypted data search scheme by adopting inverted index structure. The authors in designed a dynamic version of with the ability to add and delete files efficiently. To enrich search functionalities, the first privacy-preserving multi keyword ranked search scheme over encrypted cloud data using “coordinate matching” similarity measure. Later on, presented a secure multi-keyword text search scheme in the cloud enjoying more accurate search results by “cosine similarity measure” in the vector space model and practically efficient search process using a tree-based secure index structure. Compared with symmetric search techniques, PKC-based search schemes are able to generate more flexible and more expressive search queries. In devised the first PKC-based encrypted data search scheme supporting single keyword query. The scheme from supports search queries with conjunctive keywords by explicitly indicating the number of encrypted keywords in an index. Predicate encryption is another promising technique to fulfil the expressive secure search functionality. For example, the proposed scheme in supports conjunctive, subset, and range queries, and disjunctions, polynomial equations, and inner products could be realized.

2) Authorized keyword search: To grant multiple users the search capabilities, user authorization should be enforced. In the authors adopt a server-enforced user list containing all the legitimate users' complementary keys that are used to help complete the search in the enterprise scenario to realize search authorization. But these SKC-based schemes only allow one data contributor in the system. Hwang et al. [13] in the public-key setting presented a conjunctive keyword search scheme in multi-user multi-owner scenario. But this scheme is not scalable under the dynamic cloud environment because the size of the encrypted index and the search complexity is proportional to the number of the authorized users, and to add a new user, the data owner has to rewrite all the corresponding indexes. By exploiting hierarchical predicate encryption, Li et al. [20] proposed a file-level authorized private keyword search (APKS) scheme over encrypted cloud data. However, it incurs additional communication cost, since whenever users want to search, they have to resort to the attribute authority to acquire the search capabilities. Moreover, this scheme is more suitable for the structured database that contains only limited number of keywords. The search time there is proportional to the total number of keywords in the system, which would be inefficient for arbitrarily-structured data search, e.g., free text search, in the case of dynamic file sharing system.

B. Attribute-based Encryption

There has been a great interest in developing attribute based encryption due to its fine-grained access control property. Designed the first key policy attribute-based encryption (KP-ABE) scheme, where ciphertext can be decrypted only if the attributes that are used for encryption satisfy the access structure on the user private key. Under the reverse situation, CP-ABE allows user private key to be associated with a set of attributes and ciphertext associated with an access structure. CP-ABE is a preferred choice when designing an access control mechanism in a broadcast environment. Since the first construction of CP-ABE, many works have been proposed for more expressive, flexible and practical versions of this technique. Cheung et al. [23] proposed a selectively secure CP-ABE construction in the standard model using the simple boolean function, i.e. AND gate. By adopting proxy re-encryption and lazy re-encryption techniques, also devised a selectively secure CP-ABE scheme with the ability of attribute revocation, which is perfectly suitable for the data-outsourced cloud model.

III. Problem Formulations

In this section, we will present the system model, threat model, the definition of ABKS-HD scheme, security model and design goals, respectively.

3.1 System & Threat Models

In ABKS-HD, the system involves four different entities, namely Trusted Authority (TA), Cloud Service Provider (CSP), data owner and data users, as illustrated in Fig. 4. TA generates the public keys and master keys, where the master keys are owned by itself. Data owner generates ciphertexts by utilizing public keys and access policies before sending them to CSP. When data user intends to issue a search query, he needs to obtain his secret key from TA by submitting his attributes. After that, data user sends the trapdoor as well as his attributes to CSP to gain the authorized results. Assume that there are L records M = {m1, · · · , mL} which are divided into L access levels, where m1 has the highest hierarchy and mL has the lowest hierarchy. If a specific data user can decrypt the record mi, he can also decrypt the record mj , where 1 ≤ i < j ≤ L. Take Fig. 2 as an example, the access levels of nodes R, A, B, C, D, E, F, G, H, I decrease in turn and each node is associated with a record, namely (R,m1),· · · ,(I,m10). The data user with attribute D can decrypt not only record m5 but also records m6, m7, m8, m9, m10.

For instance, each PHR data may be divided into two parts, namely personal information M1 which contains the patient's gender, social security number, name, etc, and medical record M2 which is composed of treatment protocols, medical test results, etc. Later, a specific doctor needs to access M1 so as to make a diagnosis, while the chemist who focuses on studying cancer can only access the M2. Preferably, the patient needs to encrypt the personal information and medical record individually with different access policies (T1, T2) to securely share his PHR data. Unfortunately, this inevitably incurs extra computation and storage costs. To this end, the cloud data in the same access level can be encrypted with the integrated access policy T, as illustrated in Fig. 2.

Next, we give an introduction for each entity as follows:

- TA. It is responsible for generating the system parameters and secret keys of the specific data users.
- CSP. The cloud server which is assumed to have adequate storage capacity can provide many services, i.e., data storage, computation and retrieval. Although it can honestly conduct data storage and retrieval operations, it will still spy out as much sensitive information as possible.

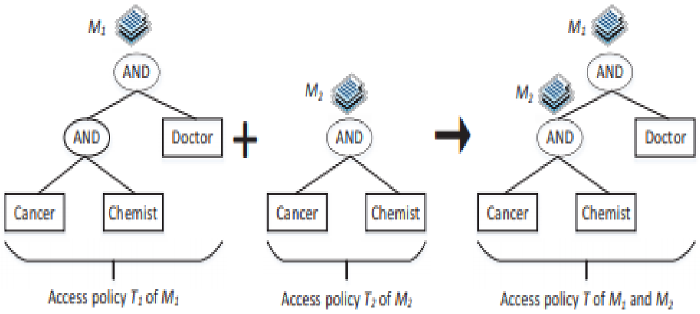

M_1
AND
AND
Doctor
Cancer
Chemist
+
M_2
AND
Cancer
Chemist
M_1
M_2
AND
AND
Doctor
Cancer
Chemist
Access policy T_1 of M_1
Access policy T_2 of M_2
Access policy T of M_1 and M_2

Fig. 1. An example of integrated access policy.

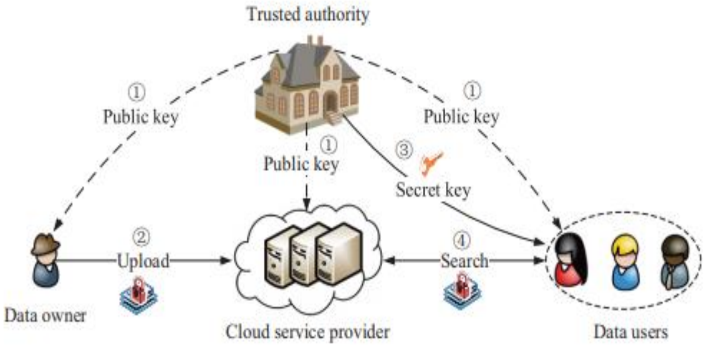


Fig. 2. System model of ABKS-HD scheme.

- Data owner. The entity that has a huge amount data to be stored and shared in cloud server is in charge of specifying access structure and generating ciphertexts for indexes and record key set.
- Data user. If he is proved to be a legal entity, then he can issue search queries according to his interested keywords and execute decryption operations.

As for TA, it is a completely trusted entity and takes charge of system initialization. As the CSP is always supported by third-parties, it is assumed to be an honest-but curious entity which honestly follows the designed protocols but may be curious to find out the valuable information. Besides, the data owner is also considered to be fully trusted, and data users who can decrypt ciphertexts cannot collude with other malicious ones.

3.2 Overview of ABKS-HD

Scheme Based on the keyword set W = {w}, data owner first extracts keywords from the record set M = {m1, · · · , mL} and builds indexes for it, as shown in Fig. 5. Then, he encrypts each record mi with different symmetric record key ki , where 1 ≤ i ≤ L. Finally, he encrypts the record key set K = {k1, · · · , kL} and indexes with our proposed ABKS-HD scheme. When a specific data user wants to access the ciphertexts containing his intended keyword, he must deliver his attribute set and trapdoor generated from his queried keyword to CSP. After that, the CSP returns the relevant ciphertexts if and only if his attribute set S (or trapdoor) matches with the access structure (indexes). It is worth noticing that only the authorized data user can obtain his corresponding record keys and decrypt the returned encrypted records.

1. **Setup(1k).** On input the security parameter k, TA runs this algorithm to output the public key PK and master key MSK.
2. **Key Gen(PK, MSK, S).** When gaining an attribute set S, TA conducts this algorithm to output the secret key SK for the specific data user.

3. **Enc(PK, W,M, K, Γ).** This algorithm is run by the data owner who has a record set M with different access levels. Then, he generates the record ciphertexts C, encryption key ciphertexts CT and indexes I according to the symmetric key set K and keyword set W. Finally, he sends the tuple (C, CT, I) to the CSP.
4. **Trap(PK, SK, S, w′).** A specific data user first conducts this algorithm to generate the trapdoor (or search token) Tw′ associated with his queried keyword w ′ . Then, he delivers his attribute set S and Tw′ to the CSP.
5. **Search(PK, S, Tw′ , CT, I).** On input Tw′ and S, the CSP issues this operation and returns the encrypted records C ′ which not only contain the keyword w ′ but also can be accessed by the specific data user.
6. **Dec(PK, SK, S).** The specific data user issues this algorithm to gain the symmetric record keys {ki , ki+1, · · · , kL} and decrypt the returned ciphertexts C ′ .

3.3 Security Models

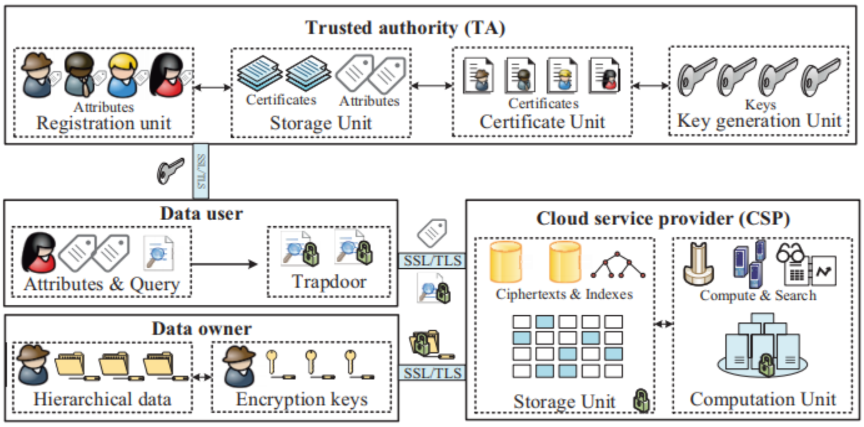
For the security of ABKS-HD scheme, the confidentiality of record set and its symmetric encryption key set should be guaranteed. Besides, the data user's secret key SK is associated with an attribute set, and the ciphertexts CT are described by the access structure, while the security model of our scheme requires that our scheme should resist the CPA. Next, we will show the CPA security game between the adversary A and challenger B as follows:

- Init. A first chooses a challenging access structure Γ ∗ and delivers it to B.
- Setup. B runs the Setup(1k) algorithm to generate the public key PK and returns it to A.
- Query phase 1. A first selects a series of attribute sets S1, · · · , Sη to query for the secret keys. Then, B answers these queries by running the KeyGen(PK, MSK, St). But there exists one restriction that St ∈/ Γ ∗ , where t ∈ [1, η].
- Challenge. A first selects two messages m0, m1 which are to be challenged on. Then, B selects a random bit ℓ ∈ {0, 1} ∗ and encrypts the message mℓ with an access structure Γ ∗ . Finally, B sends the ciphertexts CT ∗ to A.
- Query phase 2. A repeats the queries for secret keys as the same as the queries in the query phase 1.
- Guess. A outputs a guess bit ℓ ′ ∈ {0, 1} ∗ . If ℓ ′ = ℓ, A wins this security game; otherwise, it fails. The A's advantage in winning the CPA game is denoted as AdvCP A A (1k) = |P r[ℓ ′ = ℓ] – 1/2 |.

IV. Securities and Performance Analysis

In this section, we first prove that the security of ABKS-HD scheme can be guaranteed by two theorems. Next, we give its performance analysis in terms of theoretical and practical computation complexities

4.1 Security The security of ABKS-HD scheme has two aspects, namely CPA security and CKA security, which can be proved by the following two theorems, respectively. As for the CPA security, our scheme can guarantee the confidentiality of record encryption key



Cloud system architecture of our scheme.

4.2 Performance

In this section, we analyze the efficiencies of our proposed schemes in terms of theoretical analysis and actual performance with the state-of-the-art ABKS-UR [21] scheme. For the theoretical analysis, we mainly focus on the storage and computation costs. Given the element lengths in LG, LGT 1 and LZp , respectively, we show the public key size, master key size, secret key size and trapdoor size of aforementioned schemes in TABLE 3, and we notice that our schemes have much less storage costs than that of the ABKS-UR

[21] scheme as |S| ≪ |Att|. Furthermore, except for the index size, our enhanced ABKS-HD-I and ABKS-HD-II schemes do not increase the storage costs when compared with the basic ABKS-HD.

For comparison convenience, we mainly consider several time-consuming operations, i.e., bilinear pairing operation OP which maps two elements in group G to group GT , hash operation OH1 which maps the arbitrary string to group G, exponentiation operation OE in group G and exponentiation operation OET in group GT . In TABLE 4 we assess the computation overhead of KeyGen algorithm, Enc algorithm, Trap algorithm and Search algorithm, respectively.

Although we notice that the Enc algorithm in our schemes has higher computation burden, it does not affect a specific data user's search experience as it is just one-time cost. In conclusion, our schemes enjoy better performance in the secret key generation, trapdoor generation and ciphertexts retrieval phases. Compared with the basic ABKSHD scheme, our improved ABKS-HD-I and ABKS-HDII schemes can achieve more practical features (including multi-keyword search and user revocation) without incurring a great amount of computation overhead. Although the ABKS-UR [21] scheme solves the problems of user revocation and keyword search simultaneously, it brings in a large amount of storage and computation cost and cannot satisfy the requirements of cloud storage as ours. Thus, our schemes are feasible in a broad range of applications. However, to evaluate the actual performance of aforementioned schemes, we need to present the experimental simulations using real-world Enron Email Dataset2 which includes half a million records from 150 users. This public email dataset used in many SE schemes contains half a million records from about 150 users, mostly senior management of Enron, and the Enron corpus contains a total of about 0.5M message. The experiments are implemented on an Ubuntu Server 15.04 with Intel Core i5 Processor 2.3 GHz by using C and Paring Based Cryptography (PBC) Library. In PBC Library, the Type A is denoted as E(Fq) : y 2 = x 3 + x, the group G and group GT of order p are subgroups of E(Fq), where the parameters p and q are equivalent to 160 bits and 512 bits, respectively. Then, we have LZp = 160 bits, LG = LGT = 1024 bits. For comparison convenience, we set |Att| ∈ [1, 100], |S| ∈ [1, 50], L ∈ [1, 10000], n ∈ [1, 1000] in accordance with the CP-ABE scheme [9], and all of the experimental results are averages of 100 trials. Meanwhile, to compare with the state-of-theart ABKS-UR [21] scheme, we just show the experimental results of KeyGen, Enc, Trap and Search algorithms as follows.

For comparison, we fix the value of |Att| as 100 and vary the number of a specific data user's attributes |S|. As illustrated in Fig. 8 (a), we notice that our proposed schemes can greatly improve the efficiency of KeyGen algorithm. In addition, the computation overhead of secret key generation increases almost linearly with the number of data user's attributes, while that of the ABKS-UR scheme gradually increases with the number of attributes in system (|Att|). Due to |S| ≪ |Att|, the more computation costs of KeyGen algorithm in our schemes can be saved. For instance, the key generation time in our schemes and the ABKS-UR scheme is 1.28s and 0.47s when |S| = 20, and the saved computation cost in our schemes is 63.2% approximately. The saving rate jumps from 41.9% to 9.9% when |S| ranges from 30 to 50. Due to |S| ≪ 50 in practice, the efficiencies in our schemes are improved in terms of key generation time.

V. Conclusion

In this paper, we design the first attribute-based keyword search scheme in the cloud environment, which enables scalable and fine-grained owner-enforced encrypted data search supporting multiple data owners and data users. Compared with existing public key authorized keyword search scheme our scheme could achieve system scalability and fine-grainedness at the same time. Different from search scheme with predicate encryption, our scheme enables a flexible authorized keyword search over arbitrarily-structured data. In addition, by using proxy re-encryption and lazy reencryption techniques, the proposed scheme is better suited to the cloud outsourcing model and enjoys efficient user revocation. Moreover, we formally prove the proposed scheme semantically secure in the selective model. Thus, our proposed schemes are feasible and efficient in practice. As part of our future work, we try to explore the expressive search, such as fuzzy keyword search, range search, etc.

References

[1]. S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in Proc. of INFOCOM. IEEE, 2010, pp. 1–9.

[2]. M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attributebased encryption," IEEE TPDS, vol. 24, no. 1, pp. 131–143, 2013.

[3]. S. Kamara and K. Lauter, "Cryptographic cloud storage," in Financial Cryptography and Data Security. Springer, 2010, pp. 136–149.

[4]. D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. of S&P. IEEE, 2000, pp. 44–55.

[5]. Y. Huang, D. Evans, J. Katz, and L. Malka, "Faster secure two-party computation using garbled circuits." in USENIX Security Symposium, vol. 201, no. 1, 2011.

[6]. C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.

[7]. [7] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in Proc. of CCS. ACM, 2006, pp. 79–88.

[8]. S. Kamara, C. Papamanthou, and T. Roeder, “Dynamic searchable symmetric encryption,” in Proc. of CCS. ACM, 2012, pp. 965–976.

[9]. J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in Proc. IEEE Symposium on Security and Privacy (S&P’07), 2007, pp. 321–334.

[10]. S. Tabibian, A. Akbari, and B. Nasersharif, “A fast hierarchical search algorithm for discriminative keyword spotting,” Information Sciences, vol. 336, pp. 45–59, 2016. [11] S. Wang, J. Zhou, J. K. Liu, J. Yu, J. Chen, and W. Xie, “An efficient file hierarchy attribute-based encryption scheme in cloud computing,” IEEE Transactions on Information Forensics and Security, vol. 11, no. 6, pp. 1265–1277, 2016.

[11]. 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.

[12]. P. Zhang, Z. Chen, K. Liang, S. Wang, and T. Wang, “A cloud-based access control scheme with user revocation and attribute update,” in Proc. Australasian Conference on Information Security and Privacy (ACISP’16), 2016, pp. 525–540.

[13]. J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, “User collusion avoidance cp-abe with efficient attribute revocation for cloud storage,” IEEE Systems Journal, 2017.

[14]. Q. Zheng, S. Xu, and G. Ateniese, “Vabks: verifiable attributebased keyword search over outsourced encrypted data,” in Proc. IEEE international conference on Computer communications (INFOCOM’14), 2014, pp. 522–530.

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

p.Archana, M.Tech Student "Protecting Your Data: Attribute-Based Keyword Search Authorization in the Cloud Environment." IOSR Journal of Computer Engineering (IOSR-JCE) 20.4 (2018): 25-31.