Vol 13, Issue 01, Jan /2022
ISSN NO:0377-9254

LIGHT WEIGHT DATA SHARING SCHEME FOR CLOUD COMPUTING

[1]Ambati Hemalatha
M. Tech, [C.S.E]
Priyadarshini institute of technology & science for women Chintalapudi – 522306, Andhra Pradesh.
[2]N. Vijaya Gopal
Assistant Professor
Priyadarshini institute of technology & science for women Chintalapudi – 522306, Andhra Pradesh.

Abstract—With the notoriety of distributed computing, cell phones can store/recover individual information from anyplace whenever. Therefore, the information security issue in versatile cloud turns out to be increasingly serious and forestalls further improvement of portable cloud. There are significant investigations that have been directed to further develop the cloud security. Notwithstanding, the greater part of them are not pertinent for portable cloud since cell phones just have restricted processing assets and power. Arrangements with low computational overhead are in extraordinary requirement for portable cloud applications. In this paper, we propose a lightweight information sharing plan (LDSS) for versatile distributed computing. It takes on CP-ABE, an entrance control innovation utilized in typical cloud climate, however changes the design of access control tree to make it reasonable for portable cloud conditions. LDSS moves a huge part of the computational serious access control tree change in CP-ABE from cell phones to outer intermediary servers. Moreover, to diminish the client renouncement cost, it acquaints trait depiction fields with execute lethargic disavowal, which is a prickly issue in program based CP-ABE frameworks. The exploratory outcomes show that LDSS can viably decrease the overhead on the cell phone side when clients are sharing information in portable cloud conditions.

Index Terms—mobile cloud computing, data encryption, access control, user revocation

1 INTRODUCTION

With the advancement of distributed computing and the notoriety of brilliant cell phones, individuals are step by step getting acclimated with another time of information sharing model in which the information is put away on the boisterous and the cell phones are utilized to store/recover the information from the cloud. Normally, cell phones just have restricted extra room and figuring power. Actually, the cloud has huge measure of assets. In such a situation, to accomplish the acceptable presentation, it is vital for utilize the assets given by the cloud specialist organization (CSP) to store and share the information.

These days, different cloud portable applications have been broadly utilized. In these applications, individuals (information proprietors) can transfer their photographs, recordings, reports and different documents to the cloud and offer these information with others (information clients) they like to share. CSPs additionally give information the executives usefulness to information proprietors. Since individual information records are touchy, information proprietors are permitted to pick whether to make their information documents public or must be imparted to explicit information clients. Obviously, information protection of the individual touchy information is a major worry for some information proprietors.

Vol 13, Issue 01, Jan /2022
ISSN NO:0377-9254

The cutting edge honor the board/access control components given by the CSP are either not adequate or not extremely helpful. They can't meet every one of the necessities of information proprietors. To start with, when individuals transfer their information records onto the cloud, they are avoiding the information where is with regards to their control, and the CSP might keep an eye on client information for its business advantages as well as different reasons. Second, individuals need to send secret key to every information client to impart the scrambled information to specific clients, which is exceptionally lumbering. To improve on the honor the board, the information proprietor can isolate information clients into various gatherings and send secret phrase to the gatherings which they need to share the information. Be that as it may, this methodology requires fine-grained admittance control. In the two cases, secret word the executives is a major issue.

Clearly, to take care of the above issues, individual delicate information ought to be scrambled before transferred onto the cloud with the goal that the information is secure against the CSP. Nonetheless, the information encryption brings new issues. Instructions to give proficient access control system on ciphertext unscrambling so just the approved clients can get to the plaintext information is testing. What's more, framework should offer information proprietors compelling client honor the executives ability, so they can allow/deny information access honors effectively on the information clients. There have been significant explores on the issue of information access command over ciphertext. In these investigates, they have the accompanying normal presumptions. To start with, the CSP is viewed as genuine and inquisitive. Second, every one of the touchy information are encoded before transferred to the Cloud. Third, client approval on specific information is accomplished through encryption/unscrambling key dissemination. As a rule, we can isolate these methodologies into four classifications: straightforward ciphertext access control, progressive access control, access control dependent on completely homomorphic encryption [1][2] and access control dependent on trait based encryption (ABE). This large number of recommendations are intended for non-versatile cloud climate. They burn-through enormous measure of capacity and calculation assets, which are not accessible for cell phones. As per the trial results in [26], the essential ABE tasks take significantly longer time on cell phones than PC or work stations. It is somewhere quite a bit longer to execute on an advanced mobile phone than a (PC). This implies that an encryption activity which requires one moment on a PC will take about thirty minutes to complete on a cell phone. Moreover, current arrangements don't take care of the client honor change issue well indeed. Such an activity could bring about extremely high disavowal cost. This isn't pertinent for cell phones also. Obviously, there could be no legitimate arrangement which can adequately tackle the protected information sharing issue in versatile cloud. As the versatile cloud turns out to be increasingly famous, giving a proficient secure information sharing instrument in portable cloud is in critical need.

To resolve this issue, in this paper, we propose a Lightweight Data Sharing Scheme (LDSS) for versatile distributed computing climate. The fundamental commitments of LDSS are as per the following:

(1) We plan a calculation called LDSS-CP-ABE dependent on Attribute-Based Encryption (ABE) technique to offer productive access command over ciphertext.

(2) We use intermediary servers for encryption and unscrambling tasks. In our methodology, computational concentrated tasks in ABE are directed on intermediary servers, which extraordinarily decrease the computational overhead on customer side cell phones. In the mean time, in LDSS-CP-ABE, to keep up with information security, a rendition property is additionally added to the entrance structure.

The decoding key organization is adjusted with the goal that it very well may be shipped off the intermediary servers in a protected manner.

(3) We present sluggish re-encryption and depiction field of qualities to lessen the denial overhead when managing the client repudiation issue.

(4) Finally, we carry out an information sharing model structure dependent on LDSS. The trials show that LDSS can significantly lessen the overhead on the customer side, which just presents a negligible extra expense on theserver side. Such a methodology is gainful to execute a practical information sharing security plot on cell phones.

The outcomes likewise show that LDSS has better execution contrasted with the current ABE based admittance control plans over ciphertext.

2. OUR PROPOSED MECHANISM

We propose LDSS, a system of lightweight information sharing plan in versatile cloud (see Fig. 1). It has the accompanying six parts.

(1) Data Owner (DO): DO transfers information to the versatile cloud and offer it with companions. DO decides the entrance control strategies.

(2) Data User (DU): DU recovers information from the versatile cloud.

(3) Trust Authority (TA): TA is liable for producing and disseminating property keys.

(4) Encryption Service Provider (ESP): ESP gives information encryption tasks to DO.

(5) Decryption Service Provider (DSP): DSP gives information decoding tasks to DU.

(6) Cloud Service Provider (CSP): CSP stores the information for DO. It reliably executes the tasks mentioned by DO, while it might look over information that DO has put away in the cloud.

As displayed in Fig. 1, a DO sends information to the cloud. Since the cloud isn't tenable, information must be encoded before it is transferred. The DO characterizes access control strategy as access control tree (allude to Definition 2 in Section 3.2) on information records to appoint which credits a DU ought to get to get to a specific information document. In LDSS, information records are completely encoded with the symmetric encryption instrument, and the symmetric key for information encryption is additionally scrambled utilizing property based encryption (ABE). The entrance control strategy is inserted in the ciphertext of the symmetric key. Just a DU who gets quality keys that fulfill the entrance control strategy can decode the ciphertext and recover the symmetric key. As the encryption and unscrambling are both computationally concentrated, they present significant weight for portable clients. To mitigate the overhead on the customer side cell phones, encryption specialist organization (ESP) and unscrambling specialist organization (DSP) are utilized. Both the encryption specialist co-op and the unscrambling specialist organization are additionally semi-trusted. We change the customary CP-ABE calculation and plan a LDSS-CP-ABE calculation to guarantee the information security while re-appropriating computational errands to ESP and DSP.

Vol 13, Issue 01, Jan /2022
ISSN NO:0377-9254

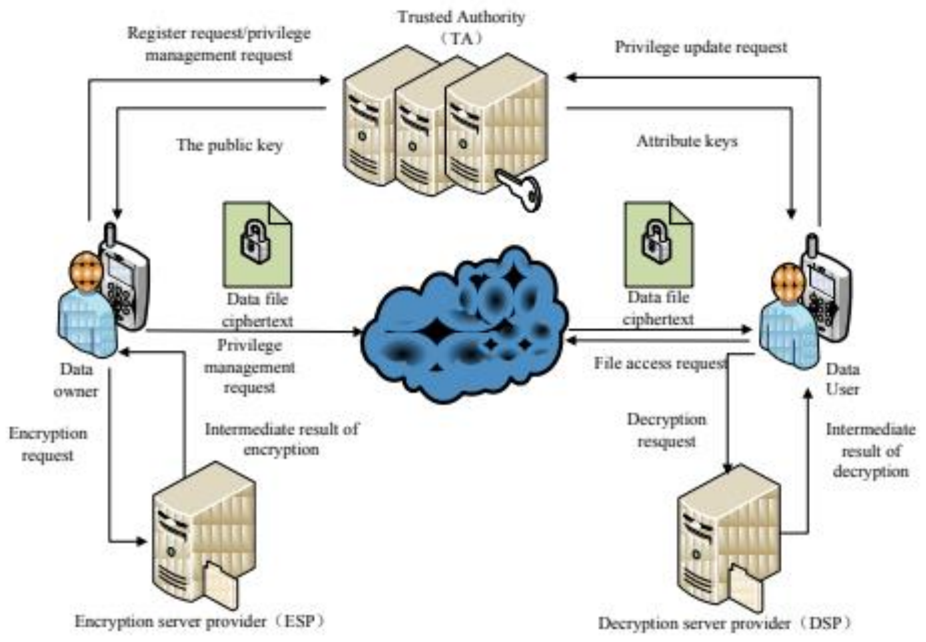
Trusted Authority (TA)
Register request/privilege management request
Privilege update request
The public key
Attribute keys
Data file ciphertext
Data file ciphertext
Privilege management request
File access request
Data owner
Data User
Encryption request
Intermediate result of encryption
Decryption resquest
Intermediate result of decryption
Encryption server provider（ESP）
Decryption server provider（DSP）

Fig. 1. A lightweight data-sharing scheme (LDSS) framework.

2.1 LDSS-CP-ABE Algorithm

To better illustrate LDSS-CP-ABE algorithm, we first define the following terms.

Definition 1: Attribute

An attribute defines the access privilege for a certain data file. Attributes are assigned to data users by data owners. A data user can have multiple attributes corresponding to multiple data files. A data owner can define a set of attributes for its data files. The data accesses are managed by access control policy specified by data owners.

Let A = {A1, A2, A3, ..., An} be the set of attributes for a data owner. Each data user u also has a set of attributes Au, which is a non-empty subset of A, namely Au □ {A1, A2, A3, ..., An}.

For example, assume A is {relatives, colleagues,classmates, friends, teachers, peers, Hubei, Beijing, Shanghai, degree of intimacy}. A data user's subset Au could be {friend, Hubei, degree of intimacy=3}. The access control policy for a data file M could be: ((friends and degree of intimacy > 1 and Hubei) or (relatives and peers)), which means a data user cannot access M unless these conditions are met.

Definition 2: Access Control Tree

Access control tree is the specific expression of access control policies, in which the leaf nodes are attributes, and non-leaf nodes are relational operators such as and, or, n of m threshold. Each node in an access control tree represents a secret, and the secret of a top node can be split into multiple secrets by secret sharing scheme and distribute to lower level nodes. Correspondingly, if we know the secrets of leaf nodes, we can deduce the secret of non-leaf nodes by calculating recursively from bottom to top. Fig. 2 shows the access control tree for the example described in Definition 1.

Definition 3: Version Attribute.

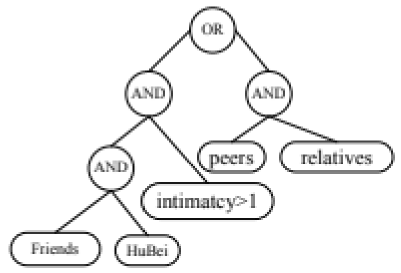

Fig. 2. The access control tree.

Version attribute is introduced in LDSS-CP-ABE algorithm to ensure security. It is an addition to the original access control tree, forming a new root node of and. We have the following definitions. T: The new access tree with version attributes.

S: The secret related to the root of T.

Ta, Ra, Sa: Ta is the initial access control tree and the left subtree of T. Ra is the root of Ta. Sa is the secret related to Ra.

Tv, Rv, Sv: Tv is the right subtree of T and contains only one node, which represents the version attribute Rv. Sv is the secret related to Rv.

Both Sa and Sv are derived from S based on the secret sharing scheme.

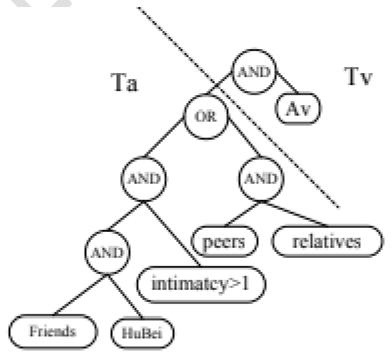

Fig. 3. The access control tree with version attributes.

control tree with version attributes is shown in Fig. 3.

LDSS-CP-ABE algorithm is designed using above definitions. It includes four sub-functions:

Setup(A, V): Generate the master key MK, the public key PK based on attribute set A of the Data Owner and the version attribute V .

Vol 13, Issue 01, Jan /2022
ISSN NO:0377-9254

KeyGen(Au, MK): Generate attribute keys SKu for a data user U based on his attribute set Au and the master key MK.

Encryption(K, PK, T): Generate the ciphertext CT based on the symmetric key K, public key PK and access control tree T .

Decryption(CT,T,SKu): Decrypt the ciphertext CT using the access control tree T and the attribute keys SKu .

We clarify these capacities explicitly beneath. To start with, work Setup() is called by the confided in outsider (TA) to produce the expert key and the public key. The expert key is utilized to create property keys and the public key is utilized to encode information documents. The course of this capacity is given in Function 1.

```
Function 1: Setup()
INPUT: The attribute set $A$ , the version attribute $V$.
OUTPUT: The master key $MK$, the public key $PK$.
1. Construct a p-order bilinear group $G_0$ of generator $g$
   and a bilinear mapping $e: G_0 * G_0 = G_1$ .
2. Randomly choose $a, b \in Z_p$ and calculate
   $g^b, e(g,g)^a$ .
3. For each attribute $a_i$ in $A$, randomly choose $t_i \in Z_p$,
   and calculate $X_i = g^{t_i}$.
4. For V, randomly choose $t_v \in Z_p$ , and calculate
   $X_v = g^{t_v}$ .
5. Return the master key $MK$ and the public key $PK$,
   Wherein $MK=\{a,b\}$, $PK=\{ G_0 , g , g^b, e(g,g)^a, \{X_i\}_{i=1}^{k}, X_v\}$.
```

Second, function KeyGen() is used to generate attribute keys for users, as shown in Function 2.

```
Function 2: KeyGen()
INPUT: The attribute set $A_u$, the master key $MK=\{a,b\}$.
OUTPUT: Attribute keys associated with $A_u$.
1. Randomly choose a parameter $r \in Z_p$ , and
   calculate $SK_r = g^{(a+r)/b}$ .
2. For each attribute $a_i$ in $A_u$, randomly choose
   $r_i \in Z_p$, and calculate $SK_a = \{g^{r_i}, g^r \cdot X_i^{r_i}\}_{i=1}^{j}$ .
3. For $V$, randomly choose $r_v \in Z_p$ , and calculate
   $SK_v = \{g^{r_v}, g^r \cdot X_v^{r_v}\}$ .
4. Return $SK_u = \{SK_r, SK_a, SK_v\}$ .
```

Vol 13, Issue 01, Jan /2022
ISSN NO:0377-9254

Function 3: Encryption()

```
INPUT: The symmetric key $K$, public key $PK$, access control tree $T$ (including the left subtree $T_a$, right subtree $T_v$, and left subtree has $num$ leaf nodes).
OUTPUT: The ciphertext $CT$.
1. Randomly choose $S \in Z_p$ as the secret of $T$, and
   calculate $CT_k$=$\{g^{bS}, K \bullet e(g,g)^{aS}\}$.
2. Get the value of the two children (namely $S_a$, $S_v$) of
   the root node according to the access control tree.
3. Calculate $CT_v$=$\{g^{S_v}, g^r \cdot X_v^{S_v}\}$.
4. Return $CT$=$\{CT_k, CT_a, CT_v\}$.
```

Function 4: Decryption()

```
INPUT: Ciphertext $CT$, the access control tree $T$ (including the left subtree $T_a$, right subtree $T_v$, and left subtree has $num$ leaf nodes), $SK_u$ (attribute keys of user $U$).
OUTPUT: The plaintext of $K$.
1. Randomly choose $t$, and get $SK_u'$=$\{SK_r' = SK_r^{\ t}, SK_a,$
   $SK_v\}$.
2. For every leaf node $z$ of $T_a$, calculate DecryptLeaf($CT_a$,
   $SK_u'$, $z$) = $e(g, g)^{q_z(0)}$.
3. For the leaf node in right subtree, calculate
   DecryptLeaf($CT_v$, $SK_u'$, $V$) = $e(g, g)^{q_v(0)}$.
4. Let $CT_k\text{-}1 = g^{bS}$, $CT_k\text{-}2 = K \bullet e(g, g)^{aS}$, calculate $K$
   $= \dfrac{CT_k - 2}{e(CT_k - 1, SK_r')^{\frac{1}{t}}/Fx} = \dfrac{CT_k - 2}{e(CT_k - 1, SK_r')^{\frac{1}{t}}/e(g, g)^{rS}}$.
```

3. IMPLEMENTATION

LDSS scheme is designed for data sharing in mobile cloud. The whole process of LDSS includes system initialization, file sharing, user authorization, and file access operations. It also has to support attribute revocation and file update operations.

System Initialization

In system initialization, Function 1 is executed. The specific process is described as follows.

(1) When the data owner (DO) registers on TA, TA runs the algorithm Setup() to generate a public key PK and a master key MK. PK is sent to DO while MK is kept on TA itself.

(2) DO defines its own attribute set and assigns attributes to its contacts. All these information will be sent to TA and the cloud.

(3) TA and the cloud receive the information and store it.

Vol 13, Issue 01, Jan /2022
ISSN NO:0377-9254

File Sharing

The process of file sharing uses Function 3 to encrypt data files. The specific process is described as follows.

(1) DO selects a file M which is to be uploaded and encrypts it using a symmetric cryptographic mechanism (such as AES, 3DES algorithm) with a symmetric key K, generating ciphertext C.

(2) DO assigns access control policy for M and encrypts

K with the assistance of ESP using Function 3, generating the ciphertext of K (CT). (3) DO uploads C, CT and access control policy to the cloud.

User Authorization

The course of client approval executes Function 2 to produce trait keys for information clients. The particular cycle is portrayed as follows.

(1) DU logins onto the framework and sends, an approval solicitation to TA. The approval demand incorporates quality keys (SK) which DU as of now has.

(2) TA acknowledges the approval solicitation and checks whether DU has signed on previously. Assuming the client hasn't signed on previously, go to step (3) , in any case go to step (4).

(3) TA calls Function 2 to create trait keys (SK) for DU.

(4) TA looks at the characteristic portrayal field in the quality key with the property depiction field put away in information base. Assuming they are not match, go to step (5), in any case go to step (6).

(5) For each conflicting piece in portrayal field, assuming it is 1 on information client's side and 0 on TA's side, it demonstrates that DU's property has been repudiated, then, at that point, TA doesn't does anything on this piece. Assuming it is switched situation, it demonstrates that DU has been relegated with another characteristic, then, at that point, TA produces the comparing quality key for DU.

(6) TA checks the variant of each quality key of DU. Assuming that it's not something similar with the current adaptation, then, at that point, TA refreshes the relating characteristic key for DU. In the phase of client approval, TA refreshes characteristic keys for DU as indicated by the quality depiction field, which is put away with SK. It portrays which credits DU has and their relating forms. TA likewise keeps quality depiction field of DU in data set. At the point when DO changes the trait of DU, the characteristic portrayal field on the TA side is additionally refreshed. In this way, when DU logins on the framework, the characteristic depiction field on itself might be not the same as that of TA. TA needs to refresh the trait keys for DU as indicated by the characteristic depiction field comparably portrayed previously.

Access Files

At the point when DU solicitations to get to a specific information document, Function 4 is utilized to unscramble information. The particular cycle is portrayed as follows:

(1) DU sends a solicitation for information to the cloud.

Vol 13, Issue 01, Jan /2022
ISSN NO:0377-9254

(2) Cloud gets the solicitation and checks assuming the DU meets the entrance necessity. Assuming that DU can't meet the prerequisite, it rejects the solicitation, in any case it sends the ciphertext to DU.

(3) DU gets the ciphertext, which incorporates ciphertext of information documents and ciphertext of the symmetric

key. Then, at that point, DU executes the Function 4 to decode the ciphertext of the symmetric key with the help of DSP.

(4) DU utilizes the symmetric key to unscramble the ciphertext of information documents.

Privilege Revoked

DO can renounce credits from a DU. The cycle is as per the following.

(1) DO illuminates TA and the cloud that one quality has been denied from a particular DU.

(2) TA and the cloud update the data of DU in information base.

(3) DO marks the relating piece of the trait depiction field of information records. This system executes the nonconcurrent handling of quality renouncement and trait keys update tasks. At the point when DO needs to deny one characteristic from a DU, TA just updates the information base and doesn't refresh quality keys for DU at the same time.

Documentation Updates

Because of apathetic re-encryption, when DO denies one trait from a DU, the disavowed characteristic isn't refreshed. At the point when the information document is refreshed, assuming it has one property that has been renounced, this trait ought to be refreshed. The particular cycle is as per the following.

(1) DO checks on the off chance that there is any piece in the portrayal field of information documents has been set to '#'.

(2) DO illuminates TA which credits ought to be refreshed. Every one of the qualities that ought to be refreshed structure a set is called Anew.

(3) TA picks another worth in G0 for each trait in Anew to supplant the first one, and updates the portrayal field of DO in DO-PK/MK table, changing the comparing property depiction touch to the new worth.

(4) TA sends another PK to DO, and DO utilizes the new PK to scramble information documents.

(5) DO sets the '#' piece of the portrayal field of the relating information record to 1. This activity is basic for languid re-encryption. Assuming that the framework refreshes ascribes following the property denial activity, unreasonable overhead happens. Considering that DU definitely know the substance of an information document in the wake of getting to it, there is no compelling reason to re-encode this information record with another symmetric key right away. The DU who has been repudiated the entrance honors ought not have the option to get to the refreshed substance. In the present circumstance, the framework should re-scramble the information record. Consequently, in LDSS, characteristic updates are

Vol 13, Issue 01, Jan /2022
ISSN NO:0377-9254

deferred until related information records are refreshed. To conclude which quality ought to be refreshed, the comparing bit in the portrayal field must be set apart as '#'.

4. CONCLUSION AND FUTURE WORK

As of late, many examinations on access control in cloud depend on quality based encryption calculation (ABE). Nonetheless, conventional ABE isn't reasonable for portable cloud since it is computationally serious and cell phones just have restricted assets. In this paper, we propose LDSS to resolve this issue. It presents a clever LDSS-CP-ABE calculation to move significant calculation overhead from cell phones onto intermediary servers, subsequently it can take care of the protected information sharing issue in portable cloud. The exploratory outcomes show that LDSS can guarantee information protection in versatile cloud and decrease the overhead on clients' side in portable cloud. Later on work, we will configuration new ways to deal with guarantee information trustworthiness. To additional tap the capability of portable cloud, we will likewise concentrate on the best way to do ciphertext recovery over existing information sharing plans.

5. REFERENCES

[1] Gentry C, Halevi S. Implementing gentry's fully-homomorphic encryption scheme. in: Advances in Cryptology–EUROCRYPT 2011. Berlin, Heidelberg: Springer press, pp. 129-148, 2011.

[2] Brakerski Z, Vaikuntanathan V. Efficient fully homomorphic encryption from (standard) LWE. in: Proceeding of IEEE Symposium on Foundations of Computer Science. California, USA: IEEE press, pp. 97-106, Oct. 2011.

[3] Qihua Wang, Hongxia Jin. "Data leakage mitigation for discertionary access control in collaboration clouds". the 16th ACM Symposium on Access Control Models and Technologies (SACMAT), pp.103-122, Jun. 2011.

[4] Adam Skillen and Mohammad Mannan. On Implementing Deniable Storage Encryption for Mobile Devices. the 20th Annual Network and Distributed System Security Symposium (NDSS), Feb. 2013.

[5] Wang W, Li Z, Owens R, et al. Secure and efficient access to outsourced data. in: Proceedings of the 2009 ACM workshop on Cloud computing security. Chicago, USA: ACM pp. 55-66, 2009.

[6] Maheshwari U, Vingralek R, Shapiro W. How to build a trusted database system on untrusted storage. in: Proceedings of the 4th conference on Symposium on Operating System Design & Implementation-Volume 4. USENIX Association, pp. 10-12, 2000.

[7] Kan Yang, Xiaohua Jia, Kui Ren: Attribute-based fine-grained access control with efficient revocation in cloud storage systems. ASIACCS 2013, pp. 523-528, 2013.

[8] Crampton J, Martin K, Wild P. On key assignment for hierarchical access control. in: Computer Security Foundations Workshop. IEEE press, pp. 14-111, 2006.

[9] Shi E, Bethencourt J, Chan T H H, et al. Multi-dimensional range query over encrypted data. in: Proceedings of Symposium on Security and Privacy (SP), IEEE press, 2007. 350- 364

Vol 13, Issue 01, Jan /2022
ISSN NO:0377-9254

[10] Cong Wang, Kui Ren, Shucheng Yu, and Karthik Mahendra Raje Urs. Achieving Usable and Privacy-assured Similarity Search over Outsourced Cloud Data. IEEE INFOCOM 2012, Orlando, Florida, March 25-30, 2012

[11] Yu S., Wang C., Ren K., Lou W. Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing. INFOCOM 2010, pp. 534-542, 2010

[12] Kan Yang, Xiaohua Jia, Kui Ren, Bo Zhang, Ruitao Xie: DAC- MACS: Effective Data Access Control for Multiauthority Cloud Storage Systems. IEEE Transactions on Information Forensics and Security, Vol. 8, No. 11, pp.1790-1801, 2013.

[13] Stehlé D, Steinfeld R. Faster fully homomorphic encryption. in: Proceedings of 16th International Conference on the Theory and Application of Cryptology and Information Security. Singapore: Springer press, pp.377-394, 2010.

[14] Junzuo Lai, Robert H. Deng ,Yingjiu Li ,et al. Fully secure key- policy attribute-based encryption with constant-size ciphertexts and fast decryption. In: Proceedings of the 9th ACM symposium on Information, Computer and Communications Security (ASIACCS), pp. 239-248, Jun. 2014.

[15] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute- based encryption. in: Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP). Washington, USA: IEEE Computer Society, pp. 321-334, 2007.

[16] Liang Xiaohui, Cao Zhenfu, Lin Huang, et al. Attribute based proxy re-encryption with delegating capabilities. in: Proceedings of the 4th International Symposium on Information, Computer and Communications Security. New York, NY, USA: ACM press, pp. 276-286, 2009.

[17] Pirretti M, Traynor P, McDaniel P, et al. Secure atrribute-based systems. in: Proceedings of the 13th ACM Conference on Computer and Communications Security. New York, USA: ACM press, pp. 99-112, 2006.

[18] Yu S., Wang C., Ren K., et al. Attribute based data sharing with attribute revocation. in: Proceedings of the 5th International Symposium on Information, Computer and Communications Security (ASIACCS), New York, USA: ACM press pp. 261-270, 2010.

[19] Sandhu R S, Coyne E J, Feinstein H L, et al. Role-based access control models. Computer, 29(2): 38-47, 1996. [20] Tian X X, Wang X L, Zhou A Y. DSP RE-Encryption: A flexible mechanism for access control enforcement management in DaaS. in: Proceedings of IEEE International Conference on Cloud Computing. IEEE press, pp.25-32, 2009