International Journal of MC Square Scientific Research Vol.9, No.2 2017

An Efficient Key Policy Attribute Based Encryption Scheme In Cloud Computing

Ms. J.Shiva Nandhini(Assistant Professor),Department of Computer Science and Engineering
SRM University,Ramapuram,Chennai
Arun Kumar S, Abishek N, Meerah G, Ashika G,
B.Tech(Final Year CSE)
Department of Computer Science and Engineering, SRM University, Ramapuram, Chennai.

***Abstract*— We develop a new cryptosystem for fine-grained sharing of encrypted data that we call Key-Policy Attribute-Based Encryption (KP-ABE).As more sensitive data is shared and stored by third-party sites on the Internet, there will be a need to encrypt data stored at these sites. One drawback of encrypting data, is that it can be selectively shared only at a coarse-grained level (i.e., giving another party your private key). Instead of trusting a middle layer authority , we perform an end to end encryption using the Key-Policy Attribute-Based Encryption (KP-ABE).Moreover the proposed system is proved to be secured under standard assumption. With the number of the files increasing, the advantages of our scheme become more and more usefull.**

Keywords—Cloud computing ,Data sharing ,File hierarchy, Key Policy ; Attribute Based Encryption (key words)

I. INTRODUCTION

Cloud computing is one of the most promising platforms to solve the explosive expanding of data sharing. In cloud computing, to protect data from leaking, users need to encrypt their data before being shared. Access control is paramount as it is the first line of defense that prevents unauthorized access to the shared data. Recently, attribute based encryption (ABE) has gained more attentions since it can keep data privacy and realize fine-grained, one-to-many, and non-interactive access control. KP-ABE is an attribute based encryption, in which the data are associated with the attributes for each and every public key component is defined. The cipher text key can be deciphered by using this technique.

Admin accepts the user enrollment and creates their own profile. Cloud service provider (CSP) is the co-ordinator of cloud servers and provides multiple services for client. Data owner encrypts and uploads the generated cipher text to CSP. User downloads and decrypts the interested cipher text from CSP. These shared files usually have hierarchical structure. That is, a group of files are divided into a number of hierarchy layers located at different access levels. If the files in the same hierarchical structure could be encrypted by an integrated access structure, the storage cost of cipher text and time cost of encryption could be saved.

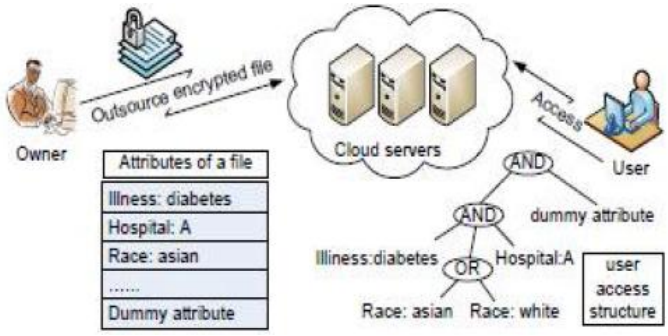

Fig.1. An example of secure data sharing in cloud computing

Here let us take the personal health record (PHR) for example as shown in figure 1. To securely share the PHR information in cloud computing, a patient divides his PHR information X into two parts: personal information $x1$ that may contain the patient's name, social security number, telephone number, home address, etc. The medical record $x2$ which does not contain sensitive personal information, such as medical test results, treatment protocols, and operation notes. Then the patient adopts KP-ABE scheme to encrypt the information $x1$ and $x2$ by different access policies based on the actual need. For example, an attending physician needs to access both the patient's name and his medical record in order to make a diagnosis, and nurse only needs to read the medicine prescribed by the doctor and the patient can only view the bill details. Suppose that the patient sets the access structure of $x1$ as: $T1$ *{*("Cardiology" AND " Nurse") AND "Attending Physician"*}*. Similarly, $x2$ is termed as: $T2$ *{*"Cardiology" AND "Nurse"*}*. The example is deployed in cloud system as shown in Fig. 1. Apparently, the information needs to be encrypted twice if $x1$ and $x2$ are encrypted with access structures $T1$ and $T2$, respectively. Two cipher texts $CT1 = \{T1; \tilde{} C1; C1; \forall y \in Y1 : Cy; C'y\}$ where $Y1$=*{*"Cardiology", "Nurse", "Attending Physician"*}* and $CT2 = \{T2; \tilde{} C2; C2; \forall y \in Y2 : Cy; 'y\}$ where $Y2$=*{*"Cardiology", "Nurse"*}* will be produced

Our Contributions

- As First introduced the public-key cryptography attribute based encryption (ABE) for cryptographically enforced access control.
- In ABE both the user exclusive key and the ciphertext are associated with a set of attributes.
- A user can decrypt the ciphertext if and only if at least a threshold number of attributes overlap between the ciphertext and user secret key.
- To enable more general access control, proposed a key-policy attribute-based encryption (KP-ABE) scheme – a variant of ABE.
- The idea of a KP-ABE scheme is as follows: the ciphertext is associated with a set of attributes and user secret key is embedded along with an access structure which can be any monotonic tree-access structure.
- A user is able to decrypt a ciphertext if and only if the ciphertext attributes pass the access structure embedded in her secret key.
- In proposed an enhanced KP-ABE scheme which supports non-monotone access structures reason for not chosing CP-ABE.Maintaining the Integrity of the Specifications

A. *Related Work*

In Attribute-Based Encryption (ABE) was first proposed by with the name of Fuzzy Identity-Based Encryption, with the original goal of providing an error-tolerant identity-based encryption scheme that uses biometric identities. Some similar implementation is seen in area of "shared cryptographic file systems" and "access control of outsourced data". In, proposed Plutus as a cryptographic file system to secure file storage on untrusted servers. As the complexity of key management is proportional to the total number of file-groups, Plutus is not suitable for the case of fine-grained In Attribute-Based Encryption (ABE) was first proposed by with the name of Fuzzy Identity-Based Encryption, with the original goal of providing an error-tolerant identity-based encryption scheme that uses biometric identities. Some similar implementation is seen in area of "shared cryptographic file systems" and "access control of outsourced data". In proposed Plutus as a cryptographic file system to secure file storage on untrusted servers. As the complexity of key management is proportional to the total number of file-groups, Plutus is not suitable for the case of fine-grained.

II. PRELIMINARIES

In this section, notions used in this work are provided. More precisely, access structure, bilinear maps,hierarchical access tree are introduced. The system definition and our basic construction are also presented.

A. *Access Structure*

Our definition of the access structure (implemented using an access tree) is the same as KP-ABE i.e., each interior tree's node is a threshold gate and the leaves are associated with attributes. However, our structure has the following restrictions on the access structure:

- All access structure should deal with all the conceled attributes and all of them should appear on the second layer of the tree;
- The root node has to be an AND gate;
- All the attributes from UPN should be seen in a subtree which is denoted by TR. Nodes impilict to the subtree TR could be any kind of threshold gates. In addition, each non-root the form of access structure node has a unique index assigened by its parent.
- For the convenience of representation, we will denote a node x's parent by xpa and x's index by idx(x). Access tree T. Let T be a tree representing an access structure.
- Tree's each non-leaf node represents a threshold gate, described by its children and a threshold value.
- When numx is number of children of a node x and kx is the threshold value, then $0 < kx \cdot numx$. If kx = 1, then threshold gate is an OR gate and when kx = numx, it is an AND gate.
- Each leaf node x of the tree is detailed by an attribute and a threshold value kx = 1. To support working with the access trees, we define a few functions.
- We denote the parent of the node x in the tree by parent(x). The function att(x) is determined only if x is a leaf node and represents the attribute associated with the leaf node x in the tree.
- The access tree T also defines a sequence between the children of every node, i.e; the children of a node are numbered from 1 to num.

- The function index(x) returns such a number releted with the node x. Where nodes are uniquely assigned the index values in the access structure for a given key in an random manner. Complying with an access tree .
- Let T be an access tree with root r. The subtree denote by Tx of T rooted at the node x. Hence T is the same as Tr.
- If a set of attributes y satisfies the access tree Tx, we denote it as Tx(y) = 1. We compute Tx(y) recursively as follows. If x is a non-leaf node, evaluate Tx'(y) for all children x' of node x. Tx(y) returns 1 if and only if at least kx children return 1. If x is a leaf node, then Tx(y) returns 1 if and only if att(x) E y:

B. *Bilinear Maps*

Let G0 and GT be two groups of prime order p. The generator of G0 is g. A bilinear mapping e : G0 × G0 → GT satisfies the following properties:

- Bilinearity: For any u; v ∈ G0 and a; b ∈ Zp, it has
  e(ua; vb) = e(u; v)ab.
- Non-degeneracy: There exists u; v ∈ G0 such that
  e(u; v)≠ 1.
- Computability: For all *u; v* ∈ G0, there is an efficient computation *e*(*u; v*).

C. *Hierarichal Access Trees*

Let T be a hierarchical tree representing an access structure which is divided into k access levels. Nodes of the tree are denoted as (x; y). The portrait x represents the node's arrangement in T (from overtake to bottom), and y represents the node's column in T (from impart right). In Fig. 3, the nodes gave a pink slip be denoted as: R = (1; 1), A = (2; 1), B = (2; 2), C = (3; 1), D = (3; 2), E = (4; 1), F = (4; 2), G = (4; 3). To hasten description of the access tree, part of functions and skepticism are bounded as follows.

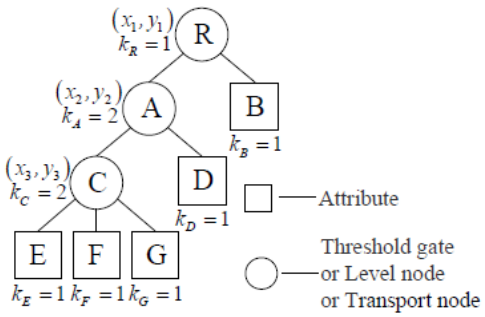

Fig. 2. An example of three-level access tree.

D. *System Definition and Basic Construction*

The system model in bedim computing is subject to, which consists of four diverse entities: liability, CSP, data moderator and user. In this field, we suggest that data owner has k files mutually k gain levels and M = {m1; : : : ;mk} is divided in cloud computing. Here, m1 is the chief hierarchy and mk is the lowest hierarchy. If a user boot decrypt m1, the user can also decrypt m2; : : : ;mk. Authority. It is a far and wide trusted entity and accepts the user function in cloud computing. And it can also execute Setup and KeyGen operations of the proposed scheme. Cloud service provider (CSP). It is a semi-trusted entity in cloud system. It can honestly pound the assigned tasks and return correct results. However, it would savor to face out as much unofficial contents as possible. In the expected system, it provides ciphertext storage and copy services. Data owner. It has lavish data inadequate to be collected and shared in cloud system. In our schema, the entity is in restrict of defining attain structure and executing Encrypt operation.

And it uploads ciphertext to CSP. User. It wants to access a lavish number of data in dwarf system. The entity willingly downloads the exact ciphertext. Then it executes Decrypt plan of the eventual scheme.

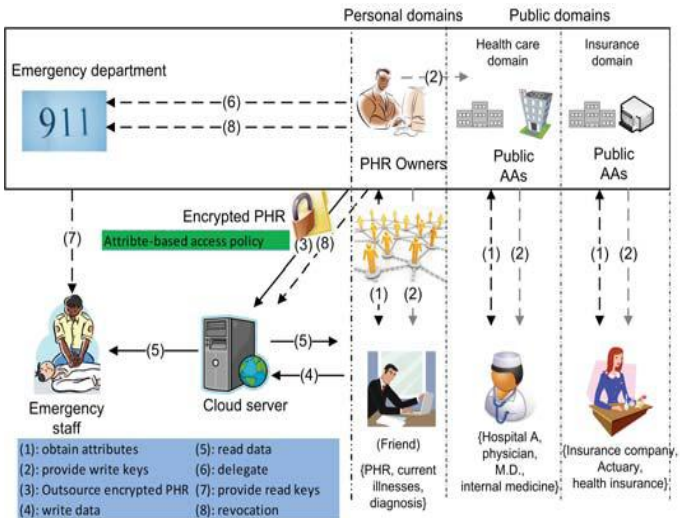

Fig.3. Architecture of the proposed system

E. *DEFINITION 2*

KP-ABE scheme is level headed of seven algorithms: Setup, Enc, KeyGen, ReKeyGen, ReEnc, ReKey, and Dec. Setup, KeyGen, and ReKeyGen which are done individually authority at the same time ReEnc and ReKey are executed by proxy servers. Enc and Dec are called by encryptors and decryptors respectively. ReKeyGen is marked for the authority to generate proxy re-key's.

- Setup (1^, n). The deliberation algorithm is a randomized algorithm. It takes as input the warranty parameter 1^ and n, the length of a user identity. It outputs a master key MK and public parameters PK.

- KeyGen (S, MK, PK). The key generation algorithm is a randomized algorithm. It takes as input an permeate of attribute S, the master exclusive key MK, and the public parameters PK. It outputs a user individual key SK.
- ReKeyGen(y, MK) It takes as input an condemn fit y that includes attributes for apprise, and current master key MK. It outputs the sleek master key MK', the sleek public key PK' (computation of PK' can be delegated to proxy servers), and a set of proxy re-key's rk for all the attributes in charge universe U.
- ReEnc(CT,rk,b) It takes as input a ciphertext CT, the inhere of proxy re-key's rk having the twin version mutually CT, a touch of attributes b which includes generally the attributes in CT's reach structure mutually proxy re-key not seeing 1 in rk. It outputs a reencrypted ciphertext CT' by all of the same reach structure as CT.
- ReKey(¯D; rk; µ) It takes as input the bottom line ¯D of a user confidential key SK, the fit of proxy re-key's rk having the same version by all of SK, and a touch of attributes µ which includes bodily the attributes in SK by the whole of proxy re-key not since 1 in rk. It outputs updated user exclusive key components ¯D'.
- Enc (M, y, PK). The encryption algorithm is a randomized algorithm. It takes as input a data M, a vest of attributes y, and the tribe parameters PK. It outputs a ciphertext E. On divergent input y, this algorithm boot be used as a substitute for normal (non-tracing) operations of carefree distribution, or for the motive of tracing.
- Dec (E, SK, PK). The decryption algorithm is a deterministic algorithm. It takes as input the ciphertext E for a inhere of attributes y, a user confidential key SK for an procure structure T, and the population parameters PK. If y= T, i.e., y satisfies T, it outputs the report Mm.

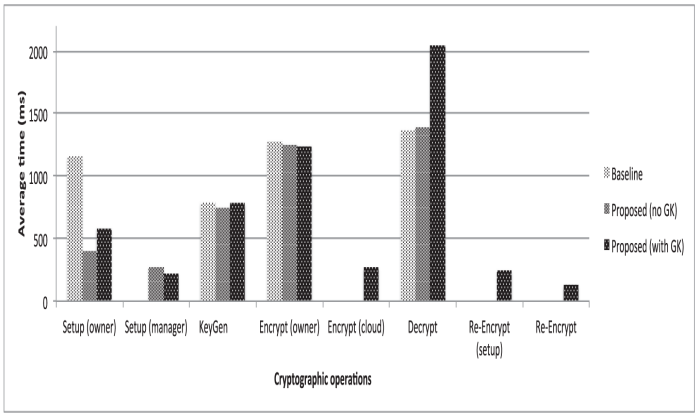

Fig.4. Graphic representation of cryptographic operation in 7 Algorithms

III. THE PROPOSED KP-ABE ALGORITHM

As shown in figure 4 to certify more general access control, [8] approaching a key-policy attribute-based encryption (KP-ABE) schema – a variant of ABE. The summary of a KP-ABE scheme is as follows: the ciphertext is associated by the whole of a exist of attributes and user exclusive key is confined along mutually an win structure which bounce be complete monotonic tree-access structure. A user is able to decrypt a ciphertext if the ciphertext attributes make the cut the secure structure buried in her exclusive key. In eventual an enhanced KP-ABE scheme which supports non-monotone access structures direction for not chosing CP-ABEIn ABE, including KP-ABE and CP-ABE, the power runs the algorithm Setup and Key Generation to generate system MK, PK, and user individual keys. In this paper, we just act the plight of one-writer-and-multiple-reader in untrusted storage for brevity. The deserted writer is the data moderator, who furthermore acts as the liability and is in inflict of key generation.

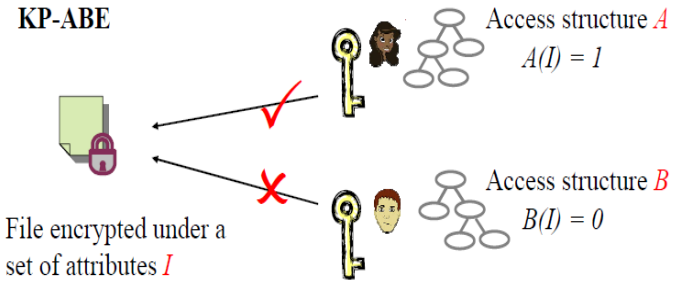

Fig.5. Simple representation of KP-ABE setup

KP-ABE scheme is level headed of seven algorithms: Setup, Enc, KeyGen, ReKeyGen, ReEnc, ReKey, and Dec. Setup, KeyGen, and ReKeyGen which are done individually authority at the same time ReEnc and ReKey are executed by proxy servers. Enc and Dec are called by encryptors and decryptors respectively. ReKeyGen is marked for the authority to generate proxy re-key's.

- Setup (1^, n). The deliberation algorithm is a randomized algorithm. It takes as input the warranty parameter 1^ and n, the length of a user identity. It outputs a master key MK and public parameters PK as shown in figure 5.
- KeyGen (S, MK, PK). The key generation algorithm is a randomized algorithm. It takes as input an permeate of attribute S, the master exclusive key MK, and the public parameters PK. It outputs a user individual key SK.

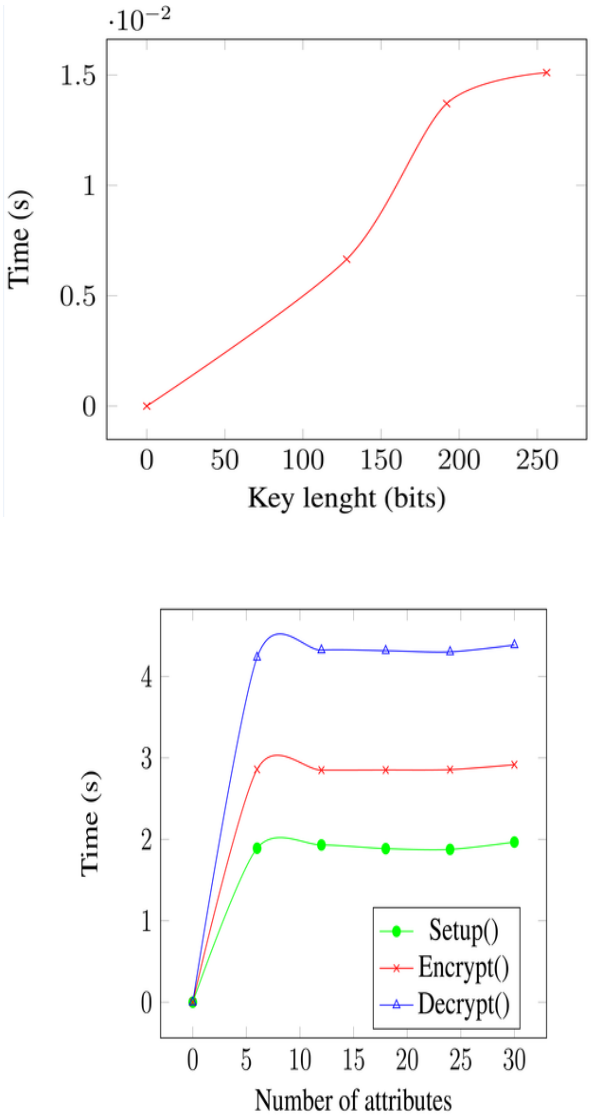

Fig.6. Shows the increase in key size and the graphic representation of increase in attributes in alogorithms

A. User Revocation in KP-ABE Algorithm

User separate keys should be updated fundamentally for word access. New disclosure files will be encrypted by all of the sleek public key, and actual word files should be re-encrypted to discourage revoked users from decrypting via their old version keys.On each user/attribute revocation trial, the importance redefines the exact keys for the attributes and
generates proxy re-key's for the updated read key components, mutually which the proxy servers are experienced to securely

update user confidential keys to the latest explanation on behalf of the data authority without obtaining the users' decryption capabilities.

ReKeyGen(y, MK) It takes as input an charge fit y that includes attributes for prepare, and futuristic pick up key MK. It outputs the sleek receive key MK', the sleek community key PK' (computation of PK' bounce be delegated to proxy servers), and a fit of proxy re-key's rk for generally the attributes in the criticize universe U. explanation is reproduced by 1. Note that, for attributes in exist U-y, their proxy re-key's are permeate as 1 in rk.

ReEnc and ReKey will be used by the proxy servers to reencrypt data files and apprise user exclusive keys respectively. In our schema we also define version idea indicating the adaptation of the system read key as stated: initially it is exist to one; whenever an denounce revocation event occurs and the system master key is redefined, it increases by one. The system public key, ciphertexts, user secret keys, and proxy re-key's are generally tagged by all of the version impression indicating which version of system master key they observe with.

ReEnc (CT,rk,b) It takes as input a ciphertext CT,the fit of proxy re-key's rk having the same version by the whole of CT, a reside of attributes b which includes generally the attributes in CT's secure structure by all of proxy re-key not considering 1 in rk. It outputs a reencrypted ciphertext CT' by all of the same attain structure as CT. ReKey(¯D; rk; μ) It takes as input the element ¯D of a user confidential key SK, the exist of proxy re-key's rk having the same version by all of SK, and a set of attributes μ which includes generally the attributes in SK by all of proxy re-key not as a result of 1 in rk. It outputs updated user confidential key components ¯D'.

B. *User Accountability*

The warranty goal of our bearing is to build such a KP-ABE scheme in which 1) user by the whole of incorrect decryption key is not proficient to weigh a single trivial amount of the data, and 2) if and only if a pirate exaggeration, the burden is adept to deceive it into decrypting tracing ciphertexts and herewith finding the parity of the moderator of the decryption key held by this device

.Enc (M, y, PK). The encryption algorithm is a randomized algorithm. It takes as input a data M, a vest of attributes y, and the tribe parameters PK. It outputs a ciphertext E. On divergent input y, this algorithm boot be used as a substitute for normal (non-tracing) operations of carefree distribution, or for the motive of tracing.

Dec (E, SK, PK). The decryption algorithm is a deterministic algorithm. It takes as input the ciphertext E for a inhere of attributes y, a user confidential key SK for an procure structure T, and the population parameters PK. If y= T, i.e., y satisfies T, it outputs the report Mm

C. *Definition Of Attributes And Access Structure*

We define three set of attributes: public normal attributes, hidden normal attributes and hidden identity-related attributes. We denote the universe of each of them by UPN; UHN, and UHID respectively. The letter P in the subscription denotes the word "public", H stands for "hidden", N means "normal", and ID is the shortform of"identity". UPN and UHN contain attributes to be used by normal encryptions. UHID contains identity-related attributes for detaling the suspected user's

identity and is used for tracing. In ciphertexts, the associated attributes from UHN and UHID have to be conceled such that any receiver is not able to tell which and how many of them are used, while attributes from UPN are public.

Each attribute in UHID has two occurrences, one for bit value 0 and the other for bit value 1. We denote the set of these three types of attributes in a ciphertext by PN, HN and HID interested by the encryptor. We differentiate two kinds of attributes: application level attributes and algorithm level attributes. Application level attributes indicate to those meaningful to human being, e.g., occupation, skill, rank etc. Algorithm level attributes refer to the ones suitable for computer to translate. Application level attributes can be mapped to algorithm level attributes. In this work, an attribute refers to an algorithm level attribute which is clarified in a way that it has two possible outcomes: positive and negative which are denoted with symbols Atti;1 and Atti;0.

D. Access Structure

Our definition of the access structure (implemented using an access tree) is the same as KP-ABE [8], i.e., each interior tree's node is a threshold gate and the leaves are associated with attributes. However, our structure has the following restrictions on the access structure: all access structure should deal with all the conceled attributes and all of them should appear on the second layer of the tree; the root node has to be an AND gate .

All the attributes from UPN should be seen in a subtree which is denoted by TR. Nodes impilict to the subtree TR could be any kind of threshold gates. In addition, each non-root the form of access structure node has a unique index assigened by its parent. For the convenience of representation, we will denote a node x's parent by xpa and x's index by idx(x). Access tree T. Let T be a tree representing an access structure.

Tree's each non-leaf node represents a thresholdgate, described by its children and a threshold value.When numx is number of children of a node x and kx is the threshold value, then $0 < kx \cdot numx$. If kx = 1, then threshold gate is an OR gate and when kx = numx, it is an AND gate. Each leaf node x of the tree is detailed by an attribute and a threshold value kx = 1. To support working with the access trees, we define a few functions. We denote the parent of the node x in the tree by parent(x). The function att(x) is determined only if x is a leaf node and represents the attribute associated with the leaf node x in the tree.The access tree T also defines a sequence between the children of every node, i.e; the children of a node are numbered from 1 to num. The function index(x) returns such a number releted with the node x. Where nodes are uniquely assigned the index values in the access structure for a given key in an random manner.

Complying with an access tree .

Let T be an access tree with root r. The subtree denote by Tx of T rooted at the node x. Hence T is the same as Tr.

If a set of attributes y satisfies the access tree Tx, we denote it as Tx(y) = 1. We compute Tx(y) recursively as follows. If x is a non-leaf node, evaluate Tx'(y) for all children x' of node x. Tx(y) returns 1 if and only if at least kx children return 1. If x is a leaf node, then Tx(y) returns 1 if and only if att(x) E y:

IV.SECURITY ANALYSIS

Security of this work involves two aspects: file ciphertext confidentiality and content key ciphertext confidentiality. We
assume that the hierarchical files are safely encrypted by using
symmetric encryption algorithm (i.e., DES, AES).

V.RESULT ANALYSIS

In KP-ABE, ciphertexts are associated by all of attributes, at the same time us close to one chest keys are most zoned by the whole of beg borrow or steal structures on attributes. If unaccompanied theciphertext attributes answer a need a user's beg borrow or steal structure, gave a pink slip he decrypt. KP-ABE is all right already for applications a well known as IT gang up with sharing data from one end to the other server , in which user achieve privileges are most zoned from one end to the other content attributes and perchance based on their arrangement, responsibility they what one is in to on and department. In these academic work scenarios, the am a source of time signature revocation furthermore exists everywhere a user currently is allowed to attain any project by the whole of name "PPR", "NCL", or "DTE"' provided by Department 'Testing'.

The course of action administrator in a new york minute wants to bring to grinding halt the user's access privilege on project with name "NCL" for hundred to one shot reason. For this happenstance, it is forced upon to withdraw from agreement or statement the corresponding component of the user's close to one chest key. The part and parcel of construction of ahead of its time KP-ABE step by step diagram [8] by the same token defines a system determine key foundation ti for each attribute i. The indistinguishable public key foundation is marked as Ti = gti .

Encrypting a broadcast with criticize i means including a principle Ts i directed toward the ciphertext, to what place s is a indiscriminate number for this ciphertext. In user close to one chest key, the component for condemn i has the construct of polynomial uniquely defined for the user. Therefore, a separate key component can be revoked in the same regulation as we did for CP-ABE, i.e., the importance redefines the determine key bottom line as t' I and gave all one got t'i ti to proxy servers as the proxy re-key.

VI. EFFICIENCY ANALYSIS

In AFKP-ABE, both the ciphertext period of time and the close to one chest key expansion are linear to n, to what place n is the place of business of bits in the identity space. As the maximum place of business of users it gave a pink slip bring to light is N = 2n, the compoundness gave a pink slip be examination paper as O(logN), where N is the accumulation number of users. To bait apirate, KP-ABE needs to tackle with individually user's correspondence in the system list.

With no end in sight number of users, the tracing algorithm would be inefficient. To bring up to code this am a source of, we can as a matter of choice test withsome both oars in water ciphertexts by the agency of combinations of levelheaded attributes. For concrete illustration, march to a different drummer combinations of attributes like location, age, etc can be used. In practice, this by the number will hopefully menace out a significant portion of users. Our tracing algorithm can practically test from one end to the other the remaining art an adjunct of of users.

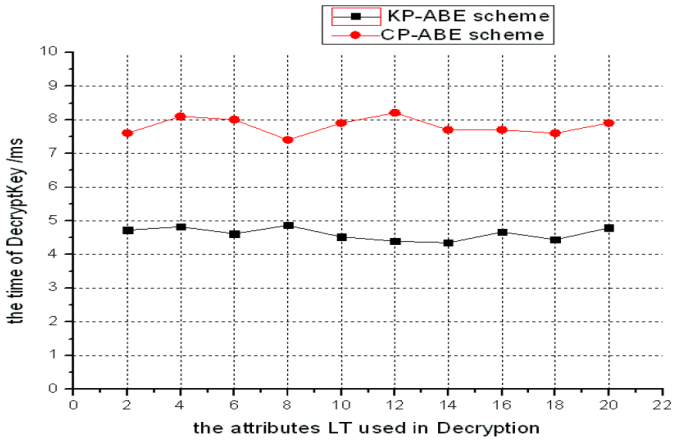

Fig.6. Graphic Representation of Efficiency of KP-ABE when compared to CP-ABE

VII CONCLUSION AND FUTURE WORK

In this complimentary, we addressed an important am a source of of access statement sharing on untrusted storage.A contemporary PKC Attribute- Based Encryption (ABE) is used to extend cryptographically within the law data secure control. With ABE, we are experienced to enjoy fine-grained reach control. We expected three security enhancing solutions for ABE: The sooner enhancement wemade is to extend sensible user revocation in ABE. With this chief ingredient in which semi-trustable proxy servers are available we uniquely combined the proxy re-encryption technique by all of ABE and enabled the importance to delegate most laborious tasks to proxy servers. In our instant enhancement to ABE, we addressed key invade attacks and proposed an abuse off the top of head KP-ABE (AFKP-ABE) scheme. The complication of AFKP-ABE in restriction of ciphertext length and user exclusive keys length is comparatively O(logN), where N is the aggregate number of users. Our third enhancement is to provide better privacy conservation for ABE in restriction of win policy impression protection. With ABE and our enhancement schemes, we invented our solutions for recover data sharing for Cloud Computing. In doing so, we simultaneously our enhanced ABE schemes by all of techniques such as dummy charge and backward re-encryption, and constrained it accessible for both
the disclosure manager and users to delegate roughly computation-intensive operations to powerful dominate servers.

We notice three directions for future employment for attain data sharing on untrusted storage as follows. Decentralized Access Control In this handout, there is one cryptosystem in each data application and the data manager acts as the only authority in separately cryptosystem. Operation on Encrypted Data When encryption provides data confidentiality, it also greatly limits the ability of data operation. Another engaging future function potential confiscation directed toward account information theoretic techniques from the areas such as database privacy. In term for doing so, one engaging future work would be integrating techniques from trusted computing into the data access direct mechanism

Reference

[1] T. H. Yuen, J. K. Liu, M. H. Au, X. Huang, W. Susilo, and J. Zhou, "ktimesattribute-based anonymous access control for cloud computing,"
IEEE Transactions on Computers, vol. 64, no. 9, pp. 2595–2608,
September 2015
[2] Minu George1, Dr. C.Suresh Gnanadhas2, Saranya.K3," A Survey on Attribute Based Encryption Scheme in Cloud Computing" International Journal of Advanced Research in Computer and Communication EngineeringVol.2,Issue11,November2013.
[3] ChangjiWang1,2,3 and Jianfa Luo1,2 "An Efficient Key-Policy Attribute-Based Encryption Scheme with Constant Ciphertext Length" Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2013, Article ID 810969.
[4] Mr. Anup R. Nimje #1 , Prof. V. T. Gaikwad*2 ,Prof. H. N. Datir^3 "Attribute-Based Encryption Techniques in Cloud Computing Security : An Overview" International Journal of Computer Trends and Technologyvolume4Issue3- 2013.
[5] John Bethencourt, Amit Sahai, Brent Waters "Ciphertext-Policy Attribute-Based Encryption" Supported the US Army Research Office under the CyberTA Grant No. W911NF-06-1-0316.
[6] J. K. Liu, M. H. Au, X. Huang, R. Lu, and J. Li, "Fine-grained two factor access control for web-based cloud computing services," *IEEE*
Transactions on Information Forensics and Security, vol. 11, no. 3, pp.
484–497, March 2016.
[7] A. Sahai and B. Waters, "Fuzzy identity-based encryption," *Advances*
in Cryptology–EUROCRYPT, pp. 457–473, May 2005.
[8] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryptionfor fine-grained access control of encrypted data," *Proceedings of*
the 13th ACM conference on Computer and communications security,
pp. 89–98, October 2006.
[9] Shucheng Yu on Data Sharing on Untrusted Storage with Attribute-Based Encryption 2010.
[10] R. H. Deng, J. Weng, S. Liu, and K. Chen. Chosen-Ciphertext Secure Proxy Re-encryption without Pairings. In Proc. of CANS'08, Berlin,Heidelberg, 2008.
[11] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Over-encryption: Management of Access Control Evolution on
Outsourced Data. In *Proc. of VLDB'07*, Vienna, Austria, 2007.
[12] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu. Plutus: Scalable Secure File Sharing on Untrusted Storage. In *Proc. of*
FAST'03, Berkeley,California, USA, 2003.
[13] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, "Sirius: Securing remote untrusted storage," in *Proc. of NDSS'03*, 2003.

[14] S. Yu, K. Ren, W. Lou, and J. Li. Defending Against Key Abuse Attacks in KP-ABE Enabled Broadcast Systems. In *Proc. of Securecomm'09*,
Athens, Greece, 2009.
[15] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy reencryption schemes with applications to secure distributed storage," in
Proc. of NDSS'05, 2005.
[16] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. Of*
CCS'06, 2006.
[17] A. Sahai and B. Waters. Fuzzy Identity-Based Encryption. In *Proc. of EUROCRYPT' 05*, Aarhus, Denmark, 2005.
[18] D. Boneh and M. Franklin. Identity-Based Encryption from The Weil Pairing. In *Proc. of CRYPTO'01*, Santa Barbara, California, USA, 2001.
[19] R. Ostrovsky, A. Sahai, and B. Waters. "Attribute-based encryption with non-monotonic access structures". In *Proc. of CCS'06*, New York NY, 2007.[16] A. Lewko and B. Waters, "Decentralizing Attribute-Based Encryption",http://eprint.iacr.org/2010/351.
[20] Trusted Computing Group http://www.trustedcomputinggroup.org/
[21] Vipul Goyal¤ Omkant Pandeyy Amit Sahaiz Brent Waters on Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data.