Message Privacy with Load Balancing using Attribute based Encryption

Jyoti Yogesh Deshmukh
Department of Computer Engineering, PVPIT,
Bavdhan, Pune, India-411021

Arati M. Dixit
Department of Computer Engineering, PVPIT,
Bavdhan, Pune, India-411021

ABSTRACT
The notion of attribute-based encryption (ABE) was proposed as an economical alternative to public-key infrastructures. ABE is also a useful building block in various cryptographic primitives such as searchable encryption. For ABE, it is not realistic to trust a single authority to monitor all attributes and hence distributing control over many attribute-authorities is desirable. A multi-authority ABE scheme can be realized with a trusted central authority (CA) which issues part of the decryption key according to a user's global identifier (GID). However, this CA may have the power to decrypt every cipher text, and the use of a consistent GID allowed the attribute-authorities to collectively build user's attributes. Decentralized ABE scheme can eliminate the burden of heavy communication and collaborative computation. It is observed that privacy-preserving decentralized key-policy ABE scheme has claimed to achieve better privacy for users and is provably secure in the standard model. However, after carefully revisiting the scheme, it is observed that existing system cannot resist the collusion attacks, hence fails to meet the basic security definitions of the ABE system. This paper proposes a solution without the trusted CA and without compromising users' privacy, thus making ABE more usable in practice. The privileged users are the users who will exactly match policy attributes with decentralized authority. To the best of our knowledge this framework of privileged users enhances the access control mechanism by avoiding the collusion.

Collusion attack occurs when more than one user try to occupy same resource at a time. Proposed system resists collusion attack at every execution. After some encryption and decryption there can be load on the system. This load can be reduced using this system in terms of processing speed. When system load is increased backup server will be initialized to reduce system load and speedup the processing of cryptography. The message privacy is therefore enhanced with load balancing using attribute based encryption (LB-ABE), as it provides an added support of rebalancing which inherently supports optimally more user work-load.

Keywords
Attribute-based Encryption, Global Identifier, Privacy, Decentralized Authority, Access Control.

1. INTRODUCTION
In traditional access control schemes [1], user's sensitive data access can be controlled by a central authority. This system can be working efficiently with some limited advantages. First, as the scheme is having central authority, it becomes difficult task to manage numerous identities of different authorities in a distributed system for validation purpose. Second, all controls and validations are to the central authority, so by default all users and resources need to trust central authority. In this case if authority is malicious then the system will be in big trouble. In attribute based access control [3] users will be validated with set of descriptive attributes, which are more than single identity. These attributes can have an access structure for secure sharing of data. Therefore, attribute-based access control schemes are efficient to share data securely with many users without taking care of their identities. To overcome the disadvantage of central authority, decentralized and distributed access control schemes are proposed. After these schemes, a decentralized attribute based access control with privacy preserving is addressed to provide the great secure sharing of sensitive data with multiple users. Attribute-based encryption (ABE) introduced by Sahai and Waters [4] is a more proficient encryption scheme and it can articulate an intricate access structure. In an ABE scheme, both the user's secret keys and the ciphertext are labeled with sets of attributes. The encrypter can encrypt a message under a set of attributes. Prior to decrypting the ciphertext, the receiver must obtain the secret keys from the central authority. The receiver can decrypt the ciphertext and obtain the data if and only if there is a match between its secret keys and the attributes listed in the cipher text. In attributes there is compulsory field of date to specify its access period, so that it will be valid within that period only. After specified period file will not be available to owner as well as to users. Essentially, there are two kinds of ABE schemes: Key Policy ABE (KPABE). In these schemes, the secret keys are associated with an access structure, while the cipher text is labeled with a set of attributes [4], [5]. Cipher text Policy ABE (CPABE). In these schemes, the cipher text is associated with an access structure, while the secret keys are labeled with a set of attributes [3]. Attrapadung and Imai proposed a dual policy ABE scheme which combines a KPABE scheme with CPABE scheme. In this scheme, two access structures are created. One is for the objective attributes labeled with the cipher text, and the other is for the subjective attributes held by the users. Furthermore, there is only one access structure in both KPABE and CPABE schemes.

Centralized Key policy attribute based encryption is supporting attribute based encryption. Where all messages are created with their attributes and some policies designed by same attributes and stores encrypted messages. These messages will be encrypted with a key and at the time decryption, same key will be used. Overall operation in CKPABE can be summarized as shown in Figure 1, with the help of following steps:

1. Data owner will create a message with attributes and policy. These details will be submitted to centralized authority for key generation. 2. Once key is generated it is issued to data owner 3. Using this key message will be encrypted by data owner. 4. User will send its global identifiers to data owner. 5. If those details are validated by any of data owner then file will be downloaded to user. 6. Valid user will request for key generation to centralized authority. 7. Centralized authority will issue key to user for decryption downloaded file to user.

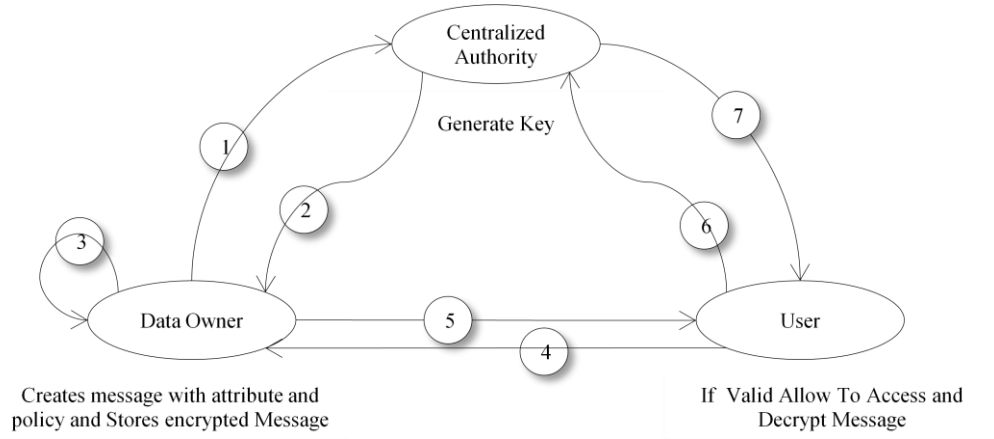


Figure 1 Centralized KPABE

Decentralized Key policy attribute based encryption is supporting attribute based encryption. Where all messages are created with their attributes and some policies designed by same attributes and stores encrypted messages to data store. These messages will be encrypted with a key and at the time of decryption, same key will be used. DKPABE will be explained with the help of Figure 2, and stepwise explanation. 1. Data owner will create a message and with some attributes and with combination of it one policy will be designed. These details will be submitted to Decentralized Authority to generate key. 2. Decentralized authority will issue the generated key to data owner. 3. With the help of key decentralized authority will encrypt the message. 4. These encrypted messages will be stored in data store in organized format. 5. User will send its details to data sore for validation and verification. 6. Data sore will go through validate user details. 7. After validation data store will allow user to download the file. 8. Valid user will request decentralized authority for keys by which message was encrypted. 9. Finally decentralized authority will issue the same key to user for decryption of message.

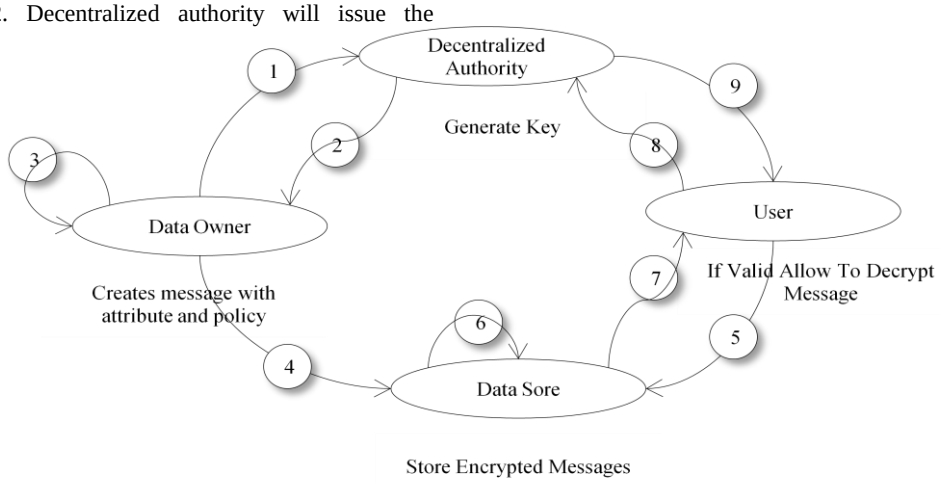


Figure 2 Decentralized KPABE

2. RELATED WORK

Fuzzy Identity based encryption is a set of descriptive attributes [1]. For decryption of ciphertext encrypted with an identity allowed by a private key for an identity, there should be identity match of both. In this scheme biometric inputs are considered as identities which are used to enable encryption. This scheme has error tolerance property because of biometric identities, which intrinsically will have some noise each time they are sampled. The manufacture is an IBE of a message under numerous attributes that invent a (fuzzy) identity. Multiauthority system, there are many authorities. That's why any party can act as authority and there is establishment of initial set of common reference parameters. Any party can be ABE authority, only the thing it have to create a public key and on request issue private keys to different users according to their attributes. A user can encrypt data in terms of any Boolean formula over attributes issued from any chosen set of authorities. The largest technical hurdle is to make it collusion resistant. As the system is multiauthority, users will come from completely different authority. There can be possibility of collusion attack due to multiauthority system, so prevention technique is there to tie keys together. This system is secure using the recent dual system encryption methodology where the security proof works by first converting the ciphertext and private keys to a semi-functional form and then comes security. The fully functional IBE scheme has chosen ciphertext security in the random oracle model assuming a variant of the computational Diffe-Hellman problem [4]. This system is based on bilinear maps between groups. The fine combination on elliptic curves is an example of such a map. Specific definitions for protected identity based encryption schemes and a number of applications for such systems are the outcomes of this system.

An attribute based encryption scheme (ABE) is a cryptographic primordial in which every user is recognized by a set of attributes, and some function of these attributes is used to determine the ability to decrypt each Ciphertext [5].

This central authority would jeopardize the whole system if it's corrupted. The proposed MAFIBE could be extended to the threshold multiauthority attribute based encryption (MAABE) scheme and be further unmitigated to a proactive MAABE scheme. In several distributed systems a user should only be able to access data if a user posses a certain set of identification or attributes [6]. If any server storing the data is compromised, then the privacy of the data will be compromised. By using these techniques encrypted data can be kept off the record even if the storage server is not trustworthy. This method is safe and sound against collusion attacks. In this system attributes are used to describe a user's credentials, and a party encrypting data determines a policy. Thus, these methods are theoretically nearer to traditional access control methods such as Role Based Access Control. In an identity based encryption scheme, each user is identified by a unique character sequence [8]. An attribute based encryption scheme (ABE), is a scheme in which each user is recognized by a set of attributes, and some function of those attributes is used to verify decryption ability for each ciphertext, it is called as policy. There is a single authority attribute encryption scheme. This scheme can tolerate a capricious number of corrupt authorities and also how to apply the techniques to get a multiauthority edition of the large universe fine grained access control ABE. Ciphertext Policy Attribute Based Encryption (CPABE) allows encrypting data under an access policy, specified as a logical combination of attributes [10]. Such cipher texts can be decrypted by anyone with a set of attributes that hysterics the policy. This scheme introduces the model of Distributed Attribute Based Encryption (DABE), where a random number of parties can be there to maintain attributes and their related secret keys. This is in barren contrast to the classic CPABE schemes, where all secret keys are distributed by one central trusted party. CPABE scheme is very effective, as it requires only some serial operations on attributes and policy during encryption and decryption.

3. PROPOSED LB-ABE METHOD

3.1 Overview:

State flow diagram of LB-ABE system is shown in Figure 3, consists of two blocks as described below:

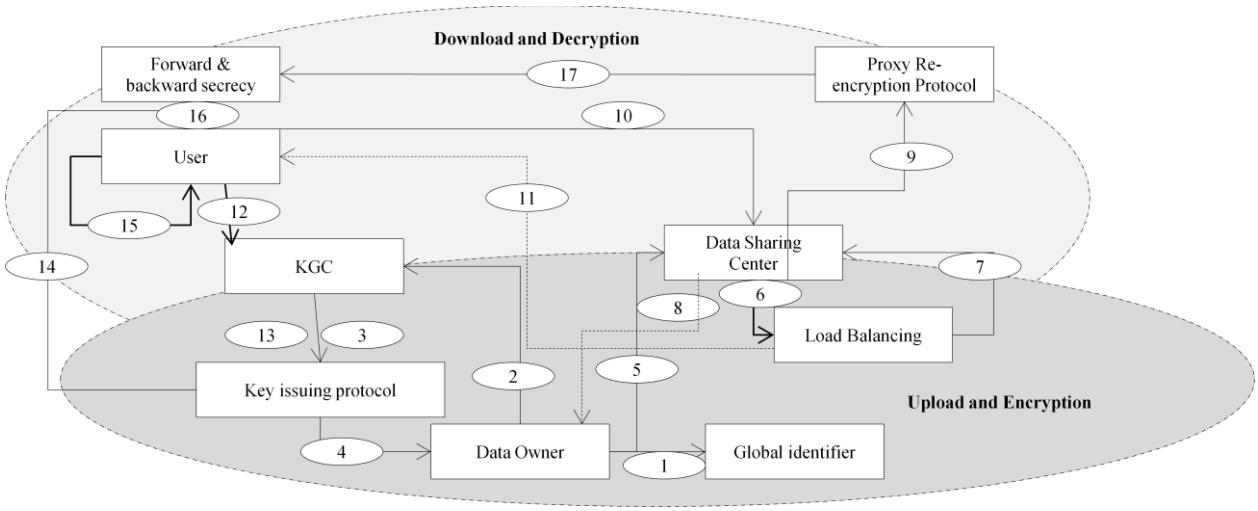


Figure 3 LB-ABE State Flow Diagram

3.1.1 Uploading and Encryption: In this block files will be uploaded and encrypted by Data owner.

1. Data owner will create message along with global identifiers like attributes and policy.
2. These details will be submitted by data owner to KGC (key generation center).
3. KGC will call key issuing protocol (KIP) for processing the information.
4. KIP will issue key to data owner.
5. Using this key data owner will encrypt the message and store it in data sharing center.
6. Uploading of message will go through load balancing algorithm.
7. If necessary data store will store data according to load balancing, otherwise it will store data normally.
8. After successful uploading of message data store will acknowledge data owner.
9. Final stage of uploading and encryption will be categorizing the stored files of data sharing center.

3.1.2 Downloading and Decryption: In this block files will be downloaded and decrypted by user.

10. User will send its identifiers to data sharing center for validation.
11. After validation user will get a response in terms of downloaded file.
12. Valid user will then request for key to KGC.
13. KGC will call KIP.
14. KIP will generate a key and issue it to user.
15. With the help of this key user will decrypt the file.
16. Forward and backward secrecy will monitor user decrypted file.

Table 1: Notations used

Notations	Notations used
SK	Secret key
PK	Private key
GID	Global identifier
CT	Ciphertext
KG	Key generation
AU	Authority
DO	Data owner
P	Policy
AP	Attribute policy
E	Encryption
D	Decryption

3.2 LB-ABE System Architecture:

LB-ABE system architecture is explained in Figure 4, AU_1, AU_2, …, AU_m and DO_1, DO_2,…, DO_m are data owners to generate attributes and design policy. That's why data owners will direct have two way associations with dataset of attribute and policy. This association will in combination have association with key generation center. After processing by KGC key will be generated and issued to authority and dataset association. After getting key data will be encrypted and stored into data store DS_1, DS_2, …, DSm. After that if user wants to access data it will request to data store and if it gets verified then allowed to access data of data store. There can be number of users available.

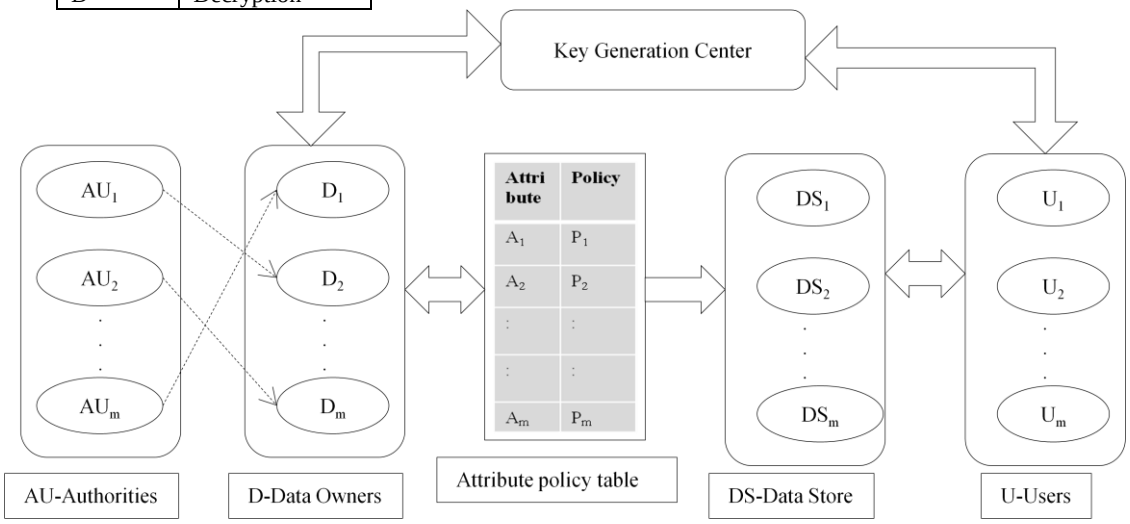


Figure 4 LB-ABE system architecture

3.3 LB-ABE System Algorithm:

A LB-ABE system consists of the following steps:

- *All inclusive attribute and policy setup*. This algorithm takes as input a security parameter ℓ and outputs the system parameters params.
- *Access structure generation.* Each authority Ai generates his secret-public key pair $\mathcal{K}(1^l) \rightarrow (Sk_i, PK_i)$ and an access structure ΔI for i=1,2,…..,N.
- *Key generation.* each authority Ai takes as input his secret key SKi, a global identifier GID and a set of attributes A^i_{GID}, and outputs the secret keys SK^i_U where $A^i_{GID}=A_{GID} \cap Ai$, A_{GID} and A_i denote the attributes corresponding to the GID and monitored by A_i respectively.
- *Encoding.* This algorithm takes as input the system parameters params, a message M and a set of attributes AC, and outputs the cipher text CT, where $A_C = \{A^1_c, A^2_c, \ldots..A^N_C\}$ And $A^i_C=A_c \cap A_i$.
- *Load balancing.* Numbers of encrypted files are stored in data store and after number of executions system space is low that time load balancing algorithm is working.
- *Decoding.* This algorithm takes as input the global identifier GID, the secret keys, $\{SK^i_U\}$ $i \epsilon I_C$ and the cipher text CT, and outputs the message M, where I_c is the index set of the authorities A_i such that $A^i_c \neq \{\phi\}$.

A decentralized key-policy attribute based encryption scheme is correct if

Pr

$$\left\langle Decryption\left(GID, \{SK^i_U\}\, i \epsilon I_C, CT\right) = M \;\middle|\; \begin{array}{c} Global\ Setup(1^l) \rightarrow params \\ Authority\ \ Setup(1^l) \rightarrow (SKi, PKi, \Delta \boldsymbol{i}); \\ Key\ Gen(SKi, GID, A^i_{GID}) \rightarrow SK^i_U; \\ Encryption(params, M, Ac) \rightarrow CT; \\ \{A_{GID} \cap Ai \in \Delta \boldsymbol{i}\} i \epsilon I_C, \end{array} \right\rangle$$

=1

Where the probability is taken over the random coins of all the algorithms in the protocol.

3.4 LB-ABE Mathematical Model

Following are the steps suggested in mathematical model for LB-ABE system:

- *Comprehensive setup:* There are i number of authorities like AU = $\{AU_1, AU_2, \ldots, AU_m\}$ and data owners DO = $\{DO_1, DO_2, \ldots, DO_m\}$where AU is authority and D is data owner.
- *Key Generation*: Data authority will create data with some attributes and policy. P_{pt} = $\{AP_i\}$. Then this plaintext will be applied with key which is generated with some function applied on attributes and policy (AP) where P ∈ A i.e. $K_{pt} = f(P_{pt})$.
- *Uploading of encrypted files*: by applying the key generated in previous step to plaintext created by data authority in first step. $E_{pt} = K_{pt}$ (PT). Store this encrypted data in data store. Here after some executions system

storage become full so load balancing algorithm will be applied.

- *Data Access*: if user is privileged user then data will be available to user. $K_u = K_{pt}$ if and only if $A_u = A_{DA}$ and $P_u = P_{DA}$. If this is the condition then only user is privileged user.
- *Downloading of decrypted files:* after having authenticated access to data stored in data storing center user will decrypt data with the same key using which it is encrypted. $D_{ct} = K_u (CT)$.

3.5 LB-ABE Security Requirements

LB-ABE security requirements can be described as following:

- *Synchronization of uploading files to system*: User's identifier is embedded in its identifier secret keys and it is encapsulated secret keys of authorities so that these keys can be tied together to secure against collusion attack. At the time of encrypting a message, all public keys of authorities are aggregated and randomized by the values. To decrypt the ciphertext only secret keys from same identifier can be used. Direct combination of different secret keys cannot be possible to get resultant secret key as a result of combination of different keys and also cannot be possible to split by malicious users. Identities of two users suppose U1 and U2 obtain secret keys for attributes which satisfy the access structures specified by the authorities of data.
- *Authorized access control:* To have access control, the LB-ABE system can only express a threshold access structure. To express an access structure for any possible combinations of attributes and policies, it is possible to implement the access tree structure technique introduced by Goyal et al.. Let T be a tree which specifies an access structure, and defines an ordering between the children of every node x from 1 to nx, where nx denotes the number of the children of the node x. Each non leaf node of T represents a threshold gate which consists of the number of its children and a threshold value.

In LB-ABE method decentralized authority will have its own master key and its main function is to generate keys whether that will be private or public based on policy set according to its attributes for encryption of the message. After encryption that message will be stored in data store in the same encrypted format. Data store will have the permission to validate the user, here the LB-ABE system analyzing for valid users. Valid users should satisfy the policy set by owner for particular message. This is how user will be analyzed. After successful message analysis will be possible to decrypt.

4. EXPERIMENTAL ANALYSIS

Consider computer department of Padmabhooshan vasantdada patil institute of technology(PVPIT). There are number of users present like ME co-ordinator(Master of Engineering co-ordinator), Project guides and finally students. So HOD(head of department) can be considered as key generation center, ME-coordinator can be considered as data storing center, project guides can be considered as data owner and students can be considered as users.

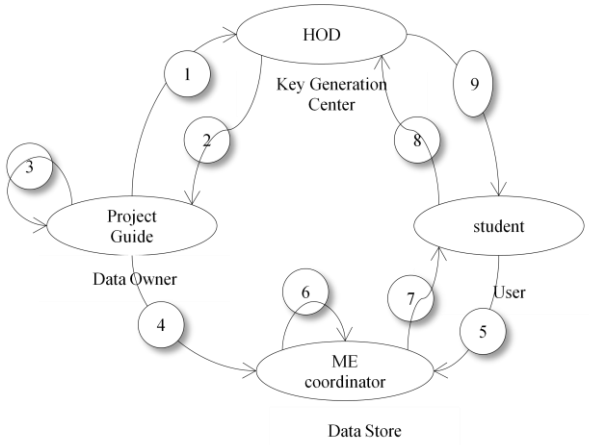


Figure 5 PVPIT case study flow diagram

The sequence of events occuring in PVPIT case study are expressed in Figure 5. The following is sequence of events:

1. Project guide creates records with some attributes and policy. This information is sent to HOD.
2. HOD will go through the information and by applying function it will generate key and issue it to Project guide.
3. By using issued key Project guide will encrypt the record.
4. Encrypted record will be stored in data store i.e. to ME coordinator.
5. To access to record of any student he or she will first send the request with some attributes and policy information.
6. ME coordinator will then verify the information of student and categorize the student into valid or invalid user.
7. After validating student ME coordinator will give access to priviledged student. 8. After validation student will send request to HOD for keys. 9. Then HOD will issue same key by which the file is encrypted.

For PVPIT case study, following are some components:
AU = {HOD, ME coordinator, Project guide},
DO = {{AU}, students},
User = {{AU}, students},

Some basic attributes that PVPIT components can have:
Basic AT={name, class, subject, roll_No., designation, address, gender, category, Mo_No., academic_year, result, division, percentage, branch, collegename, attendance, salary, experience_T(teaching), experience_I(industry), education, 10^{th}_Marks, 12^{th}_Marks, Diploma_Marks, blood_gr, mail_id, department, GFM(guardian faculty member), class_teacher, EBC(economically backward class), AreaOfInterest, semester, books_issued, residence, account_no, guardian, test_marks, Unit_test, DOB(date of birth)},

Random attributes and policy of PVPIT student:
Student AT = { name, class, subject, roll_No., address, gender, category, Mo_No., academic_year, result, division, percentage, branch, collegename, attendance, 10^{th}_Marks, 12^{th}_Marks, Diploma_Marks, blood_gr, mail_id, department, GFM, class_teacher, EBC, AreaOfInterest, semester, books_issued, residence, account_no, guardian, test_marks, DOB }
Sample policy = {roll_No, gender, DOB, GFM, books_issued, Account_no}

Random attributes and policy of PVPIT HOD:
HOD AT = { name, class, subject, designation, address, gender, category, Mo_No., academic_year, result_analysis, division, branch, collegename, attendance_record, salary, experience_T, experience_I, education, books_issued, account_no, test_record, DOB}

Sample policy = {name, department, Mo_No, result_analysis, test_record}

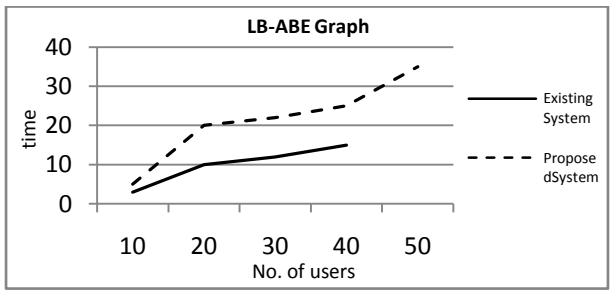
Random attributes and policy of PVPIT ME coordinator:
ME coordinator AT = { name, class, subject, designation, address, gender, category, Mo_No., academic_year, result_analysis, division, branch, collegename, attendance_record, salary, experience_T, experience_I, education, books_issued, residence, account_no, test_record, fees_record, university_updates,DOB}

Sample policy = {name, class, division, fees_record, university_updates, academic_year}

Random attributes and policy of PVPIT project_guide:
Project_guide AT = { name, class, subject, address, gender, category, Mo_No., academic_year, result_analysis, division, branch, collegename, attendance_record, salary, experience_T, experience_I, education, books_issued, residence, account_no, test_record, fees_record, university_updates, guidance_record, DOB}

Sample policy = {name, experience, education, residence, guidance_record,}

Suppose project guide will create an accessment record of every student to whom they are project guide. This accessment record will be provided with an attributes such as student roll_no, name, class, contact_details, address, area-of_interest, papers_published, implementation_details, report_status and remark. Out of these attriibutes for every student there will be different policy according to their status. So project guide itself categorizing the student record to check their improved status. Now project guide will submit these details to HOD, and then he will process this data to create a key and issue it to project guide for encrypting the data. Project guide will encrypt the data by using key issued by HOD and store it to ME coordinator. Then if student want to check their status or accessment report then they have to submit their details to ME coordinator, if these attribute details are matching to the database then he will allow to download the record. Then student will submit its attribute and policy to HOD to get a key. After issuing key from HOD student can decrypt information and get accessment report.



There are many project guides in department; each one may create too many records. The encrypted record will be stored to ME coordinator. Existing system will store the records randomly and LB-ABE system will store the records using load balancing algorithm. Means when storage space will become almost full that time load balancing algorithm will start working.

5. CONCLUSION

The decentralized attribute based encryption scheme attracted a lot of attention as in centralized attribute based encryption there is possibility of losing trust in case of sharing information of one data owner by other. In order to resist the collusion attacks in the decentralized ABE schemes, the global identifier GID is used to tie all the user's secret keys from multiple authorities together. However, this will risk the user being traced and impersonated by the corrupted authorities. In this paper, it proposes a privacy-preserving decentralized ABE scheme to protect the user's privacy. In LB-ABE scheme, all the user's secret keys are tied to its identifier to resist the collusion attacks while the multiple authorities cannot know anything about the user's identifier. Notably, each authority can join or leave the system freely without the need of reinitializing the system and there is no central authority.

LB-ABE system check for most recently required resource by any user and allot it to that particular user request. After each of these executions, system will become almost full. Then for other user executions load balancing algorithm will pop up to best fit every operation performed by users.

6. REFERENCES

[1] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption,"EUROCRYPT '05: Proc. Advances in Cryptology, R. Cramer, ed.,pp. 457-473, May 2005.

[2] A. Lewko and B. Waters, "Decentralizing Attribute – Based Encryption," EUROCRYPT '11: Proc. 30th Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology,K.G. Paterson, ed., pp. 568-588, May 2011.

[3] B.C. Neuman and T. Ts'o, "Kerberos: An Authentication Sewice for Computer Networks," IEEE Comm. Magazine, vol. 32, no. 9,pp. 33-38, Sept. 1994.

[4] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," CRYPTO '01: Proc. Advances in Cryptology, J. Kilian, ed., pp. 213-229, Aug. 2001.

[5] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure Threshold Multi-Authority Attribute Based Encryption without a Central Authority,"INDOCRYPT '08: Proc. Int'l Conf. Cryptology in India,D.R. Chowdhury, V. Rijmen, and A. Das, eds., pp. 426-436, Dec.2008.

[6] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," Proc. IEEE Symp. Security and Privacy (S&P '07), pp. 321-34, May 2007.

[7] J. Li, Q. Huang, X. Chen, S.S.M. Chow, D.S. Wong, and D. Xie, "Multi-Authority Ciphertext-Policy Attribute-Based Encryption with Accountability," Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS '11), pp. 386-390, 2011.

[8] M. Chase, "Multi-Authority Attribute Based Encryption," Proc.Theory of Cryptography Conf. (TCC '07), S.P. Vadhan, ed., pp. 515-534, Feb. 2007.

[9] N.P. Smart, "Access Control Using Pairing Based Cryptography,"CT-RSA '03: Proc. RSA Conf. The Cryptographers' Track, pp. 111-121,2003.

[10] S. Muller, S. Katzenbeisser, and C. Eckert, "Distributed Attribute-Based Encryption," Proc. 11th Int'l Conf. Information Security and Cryptology (ICISC '08), P.J. Lee and J.H. Cheon, eds., pp. 20-36,Dec. 2008.