Graduate Research in Engineering and Technology (GRET)

Volume 1 | Issue 2 Article 1

October 2013

ACHIEVING AVAILABILITY AND DATA INTEGRITY PROOF IN HIERARCHICAL ATTRIBUTE ENCRYPTION SCHEME USING HYBRID CLOUD

PALLAVI R
Department of Computer Science and Engineering, Siddaganga Institute of Technology, Tumkur, Karnataka, India., pallavi.ramanand@gmail.com

R APARNA
Department of Computer Science and Engineering, Siddaganga Institute of Technology, Tumkur, Karnataka, India., raparna@sit.ac.in

ACHIEVING AVAILABILITY AND DATA INTEGRITY PROOF IN HIERARCHICAL ATTRIBUTE ENCRYPTION SCHEME USING HYBRID CLOUD

PALLAVI R[1], DR. R APARNA[2]

[1]M.Tech Student, Department of Computer Science and Engineering, Siddaganga Institute of Technology, Tumkur, Karnataka, India.
[2]Associate Professor, Department of Computer Science and Engineering, Siddaganga Institute of Technology, Tumkur, Karnataka, India.
Email: pallavi.ramanand@gmail.com, raparna@sit.ac.in

Abstract— It has been widely observed that the concept of cloud computing has become one of the major theory in the world of IT industry. Data owner decides to release their burden of storing and maintaining the data locally by storing it over the cloud. Cloud storage moves the owner's data to large data centers which are remotely located on which data owner does not have any control. However, this unique feature of the cloud poses many new security challenges. One of the important concerns that need to be addressed is access control and integrity of outsourced data in cloud. Number of schemes has been proposed to achieve the access control of outsourced data like hierarchical attribute set based encryption [HASBE] by extending cipher-text-policy attribute set based encryption [CP-ABE]. Even though HASBE scheme achieves scalability, flexibility and fine grained access control, it fails to prove the data integrity in the cloud. Hence integrity checking concept has been proposed for HASBE scheme to achieve integrity. Though the scheme achieves integrity it fails to provide the availability of data to the user even when fault had occurred to data in the cloud. However, the fact that owner no longer have physical possession of data indicates that they are facing a potentially formidable risk for missing or corrupted data, because sometimes the cloud service provider deletes the data which are either not used by client from long-time and which occupies large space in the cloud without the knowledge or permission of data owner. Hence in order to avoid this security risk, in this paper we propose a hybrid cloud concept. Hybrid cloud is a cloud computing environment in which an organization provides and manages some internal resources and external resources. A hybrid cloud is a composition of at least one private cloud and at least one public cloud. This concept provides the availability and data integrity proof for HASBE scheme.

Keyword— *Access Control, Hybrid Cloud, Data Security, Availability and Integrity*

I. INTRODUCTION

CLOUD computing is a new computing paradigm that is built on virtualization, parallel and distributed computing and service oriented architecture. In the last several years cloud computing has emerged as one of the most influential paradigms in the IT industry and has attracted extensive attention from both academic industries. Cloud computing is defined as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Cloud storage is an important service of cloud computing which allows data owner to move data from their local computing systems to the cloud. By hosting their data in the cloud, data owners can avoid the initial investment of expensive infrastructure setup, large equipments and daily maintenance cost. The data owners only need to pay the space they actually use. Another reason is that data owners can rely on the cloud to provide more reliable services, so that they can access data from anywhere and at any time. Individuals or small sized companies usually don't have the resources to keep their servers as reliable as the cloud does have. With the high costs of data storage devices as well as the rapid rate at which data is being generated it proves costly for enterprises or individual users to frequently update their hardware. Hence data owner decides to release their burden of storing and maintaining the data locally by storing it over the cloud. While the cloud service provider are commercial enterprise, which cannot be totally trusted. Cloud storage moves the owner's data to large data centers which are remotely located, on which data owner does not have any control. However, this unique feature of the cloud poses many new security challenges like data security, privacy, availability and data integrity in cloud computing due to its internet-based data storage and management.

Access control is the major security issue. To achieve flexible and fine grained access control, a number of schemes [12]-[15] have been proposed. But this access schemes are applicable to system where data owners and service providers are within the same trusted domain. If the data owner and service provider are present in different domain, then a new access control scheme called attribute-based encryption [ABE] is proposed by Yu et al [17]. The attribute–based encryption [ABE] has key-policy

attribute-based encryption [KP-ABE] and cipher text-policy attribute based encryption [CP-ABE]. Attribute based encryption [ABE] schemes provides the fine-grained access control but fails to provide the flexibility in attribute management and scalability in multiple-levels of attribute authorities. Hence to achieve scalability, flexibility and fine grained access control, Hierarchical attribute set based encryption [HASBE] by extending cipher-text-policy attribute set based encryption [CP-ABE] scheme is proposed by Zhinguo et al [22]. Hierarchical attribute set based encryption [HASBE] scheme seamlessly extends the attribute set based encryption [ASBE] scheme to handle the hierarchical structure of system users.

Even though HASBE scheme achieves scalability, flexibility and fine grained access control, it fails to prove the data integrity in the cloud. In HASBE scheme the data owners have to give up their data to the cloud service provider for storage and business operation, while the cloud service provider are commercial enterprise which cannot be totally trusted. Data will be the important asset for organization, and owner will face serious consequences if the confidentiality of data is disclosed to the public. Thus data owners must make sure that their data are kept confidentially to outsiders. However, the fact that owner no longer have physical possession of data indicates that they are facing a potentially formidable risk for missing or corrupted data, because sometimes the cloud service provider modifies the data without the knowledge or permission of data owner. Hence in order to achieve integrity, data integrity checking concept for HASBE scheme has been proposed [23].

Even though after proposing the integrity concept for HASBE scheme it fails to provide the availability of the data to the user which is one of the major drawback. Sometimes in the cloud environment the data which is stored will be deleted by the cloud service provider without the permission of data owner, since data are either not used by client from long-time or it may occupies large space in the cloud. Hence here we can note that in HASBE scheme, there is no method called availability scheme to ensure that user will get the data even in the case of fault occurrence in the cloud. Hence it is the major drawback of HASBE scheme.

In order to overcome the above security risk, in this paper we propose a hybrid cloud concept which provides availability proof for HASBE scheme. Hybrid cloud is a cloud computing environment in which an organization provides and manages some internal resources and external resources. A hybrid cloud is a composition of at least one private cloud and at least one public cloud.

The rest of the paper is organized as follows. Section II provides an overview on related work. Then we present our proposed method with system model in Section III. In Section IV, we give the implementation details of proposed system. In section V, we analyze the security of proposed method by comparing with Zhiguo et al.'s scheme. We present our results in section VI and lastly we conclude the paper in Section VII.

II. RELATED WORK

In this section, we review the notion of hierarchical attribute set based encryption [HASBE] scheme with data integrity concept which fails to prove the availability of data to the user even during the fault occurrence situation in the cloud.

Access control is the major security issue. To achieve flexible and fine grained access control, a number of schemes [12]-[15] have been proposed. But, this access schemes are applicable to system where data owners and service providers are within the same trusted domain. If the data owner and service provider are present in different domain, then a new access control scheme called attribute-based encryption [ABE] is proposed by Yu et al [17]. The primary drawback of the attribute based encryption [ABE] scheme is that it lacks expressibility. Several efforts are introduced to solve the expressibility problem. In the ABE scheme, cipher texts are not encrypted to one particular user as in traditional public key cryptography. Rather, both cipher texts and user's decryption key are associated with a set of attributes or a policy over attributes. A user is able to decrypt a cipher text only if there is a match between the decryption key and cipher text. ABE schemes are classified into key-policy attribute-based encryption [KP-ABE] and cipher text-policy attribute based encryption [CP-ABE], depending on how attributes and access policy are associated with cipher texts and user decryption keys.

In KP-ABE scheme [16], cipher-texts are associated with the set of attributes and user key are associated with monotonic tree access structure. Only when attributes of cipher text matches with access structure of user key, user can decrypt the cipher text. The main problem with KP-ABE scheme is that the encryptor is not able to decide who can access or decrypt the encrypted data except choosing descriptive attribute for the data and has no choice other than to trust the key issuer. Hence KP-ABE scheme is not naturally suitable to certain application. In CP-ABE scheme [17], cipher-texts are associated with access structure and user keys are associated with attribute. If attributes of user key matches with the access structure of cipher-text, then user can decrypt the cipher-text. The main problem in CP-ABE scheme is, it won't provide flexibility and fine grained access. Since the attributes present in the user keys are organized logically as a single set, users can only use all possible combination of attributes in single set. Hence we note that, attribute based encryption [ABE] schemes provides the fine-grained access control, but fails to provide the flexibility in

attribute management and scalability in multiple-levels of attribute authorities.

Hence to achieve scalability, flexibility and fine grained access control, Hierarchical attribute set based encryption [HASBE] by extending cipher-text-policy attribute set based encryption [CP-ABE] scheme is proposed by Zhinguo et al [22]. Hierarchical attribute set based encryption [HASBE] scheme seamlessly extends the attribute set based encryption [ASBE] scheme to handle the hierarchical structure of system users. In HASBE scheme, the user keys are associated with attribute and cipher-texts are associated with access structure. If attributes of user key matches with the access structure of cipher-text then user can decrypt the cipher-text. Since attributes present in the user keys are organized into a recursive set structure, we note that HASBE scheme achieves scalability, flexibility and fine grained access control of outsourced data in cloud.

Even though HASBE scheme, proposed by Zhinguo et al achieves scalability, flexibility and fine-grained access control, it fails to prove data integrity in the cloud, which the major drawback of HASBE scheme. Since owner no longer has physical possession of data, there is a chance of data corruption /deletion by service provider without the knowledge or permission of data owner. To overcome this drawback the data integrity checking concept for HASBE scheme has been proposed [23]. Even after proposing the integrity concept for the HASBE scheme, it fails to prove the availability of the data to the user, even during the fault occurrence situation in the cloud.

III. PROPOSED SYSTEM

In this section, we present a hybrid cloud concept to overcome the drawback of Hierarchical attribute set based encryption [HASBE] scheme and to provide the availability of the data to the user.

A. System Model

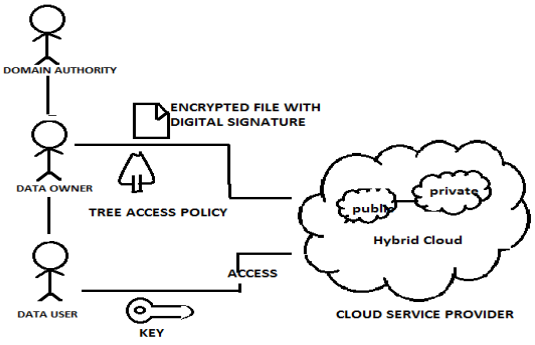

FIG 3.1.SYSTEM MODEL

As depicted in Fig. 1, the cloud computing system under

consideration consists of four types of parties: *a cloud service provider, domain authority, data owner* and *data user.*

Domain authority is the root authority and responsible for managing the data owner. Cloud service provider manages a cloud to provide the storage service. Data owner is responsible for creating the data user and for generating new file with digital signature which is further encrypted and stored in public cloud and private cloud. Data owner is also responsible for adding the access structure to file. To access the shared data file, data user must download the encrypted data file of their interest from the cloud and then decrypt them.

IV. IMPLEMENTATION

In this section, we discuss some main operations like: system setup, data owner grant, data user grant, generating new file with digital signature, data integrity check, file access, availability check and file deletion.

The above desirable operations are implemented by using six algorithms, Setup, KeyGen, Message-Digest, Digital Signature, Encrypt, and Decrypt:

- *Setup (d)*: Here d is the depth of key structure. By taking input a depth parameter d. It gives a public key (PK) and master key (MK).

- *KeyGen (MK, u, A):* By taking the input as master key (MK), user identity and attributes of key structure, it gives a secret key SKu for user u.

- *Message-Digest (M, h*): By taking the message M and hash function (h) as input, it gives the message-digest (MD) as output.
  .
- *Digital-Signature (MK, MD*): By taking the message-digest (MD) and master key (MK) as input, it outputs a digital signature.

- *Encrypt (PK, M):* By taking the public key (PK), and a message (M), as input. It outputs a cipher-text (CT).

- *Decrypt (CT, SKu):* By taking cipher-text (CT) and secret key of user (SKu) as input, it outputs a message (M). If the attributes associated with the user secret key (SKu) matches with the access structure of cipher-text (CT), then it outputs a message M which is the original correct message. Otherwise, m is null.

The modules we consider to perform the above operations are *Domain Authority, Data Owner* and *Data User.*

A. Domain Authority

Domain authority is called as root authority which is responsible for creating the data owner. Domain authority calls the system setup algorithm to generate the public key(PK) and master key(MK), where public key will be made public to other parties and master key will be kept secret.

Setup (d=2) → (PK, MK). Here d is the depth of key structure. Here in this paper we consider a key structure of depth 2, and it can be extended to any depth *d*. The algorithm selects a bilinear group G of prime order *p* with generator *g* and then chooses random exponent $\alpha, \beta_i \in Z_p$, $V_i \{1, 2\}$. To support key structure of depth *d*, *i* will range from 1 to *d*. This algorithm sets the public key and master key as follows:

PK= $(G, g, h1=g^{\beta 1}, f1=g^{1/\beta 1}, h2=g^{\beta 2}, f2=g^{1/\beta 2}, e(g, g)^{\alpha})$

MK= $(\beta_1, \beta_2, g^{\alpha})$

B. Data Owner

Data owner is created by the domain authority. Data Owner caries the operation like creating the data user, generating a new file with digital signature, data integrity check and file deletion.

1) Creating the Data Users

Data users are created by data owner. At the time of creation, data owner calls the keygen algorithm which gives the secret key for a data user. The secret key is generated by making use of master key and user attributes i.e. key structure of user, where this secret key is sent to the user at the time of creation.

a) Key Structure

Key structure defines unique labels for set in it. The depth of the key structure is the level of recursions in the recursive set which is similar to definition of depth for a tree. Here we consider a key structure with depth 2, members of the set at depth 1 can either be attribute elements or sets. Depth 2 may only be attributes elements.

The Fig 4.1 key structure of user represents the attribute of a person who is student in CSE department.

The key structure of user and master key is combined to generate the secret key. The Fig 4.2 represents the secret key of user which is the combination of the master key, user id, user department and user designation

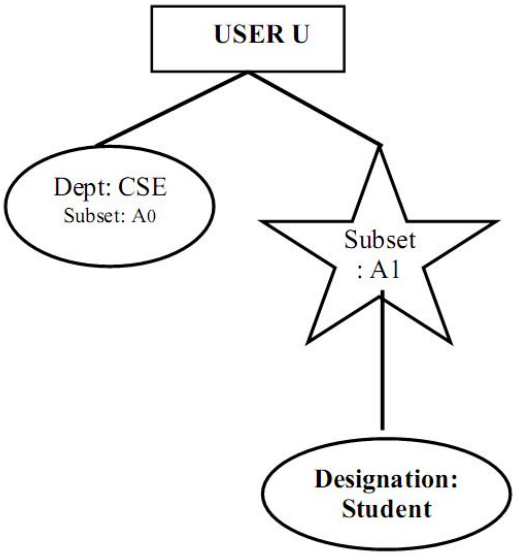

Fig 4.1. Key Structure of User

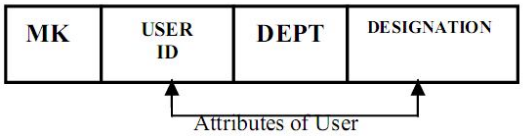

Fig 4.2. Secret Key

Once the secret key is generated it is sent to the user. The depth of the key structure can be increased by adding the extra attributes. Data owner can add an attribute like expiration-time to user's secret key which indicates the time until which the key is considered to be valid. Once the time expires then the key will be considered as invalid and user will no longer have the file access rights. To perform this key expiration-time operation, access structure associated with data files must include check on expiration-time attribute as a numerical comparison.

For e.g.: assuming a user 'U' has a key with expiration-time 'X' and a data file whose access policy is associated with expiration time 'Y', then user 'U' can decrypt the data file only when X>=Y.

2) Generating New File with Digital Signature

Once the data owner creates the file, he picks the unique ID for this data file. With the help of hash function the data owner generates the message digest for the corresponding file where further this message digest is encrypted and called as digital signature. This digital signature is appended to file and owner

calls the encrypt algorithm which returns cipher-text which is stored in public cloud and redudant copy of that file is saved in a private cloud to achieve a availability. Owner also defines the tree access structure for created files.

a) Access Structure

Access structure is the structure given to the file by the data owner. This access structure ensures that whether the user have the rights to access/download the file or not.

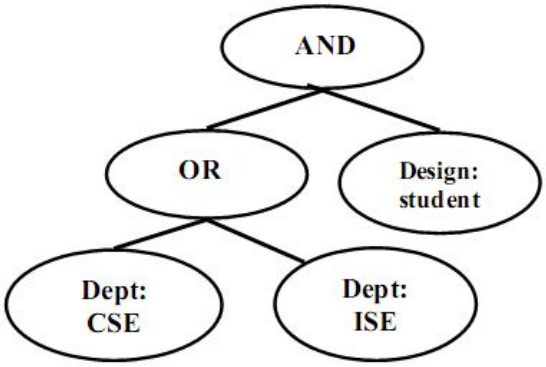

Fig 4.3. Access Structure

This kind of access structure is given to the file by the data owner. The given example of access structure indicates that, the file can be accessed by the student who belongs to either department of 'CSE' or department of 'ISE'.

3) Data Integrity Check

At the sampling period of time, the owner performs the data integrity operation by sending the request for uploaded file by specifying the file name or file id to the cloud. Once the data owner receives the file from the cloud, initially it decrypts the file and by using the hash function, owner generates the message digest [MD1]. Then by decrypting the digital signature which was appended to the file earlier, the owner will get the message digest [MD2]. By comparing the message digest MD1 and MD2, the owner verifies the integrity status. If MD1 and MD2 are same then owner ensures that, data which is present in cloud is not altered.

4) File Deletion

Encrypted data files can be deleted only at the request of the data owner. To delete an encrypted file, the data owner must send the request by specifying the file name or file id to the cloud. Only upon successful verification of the data owner and the request, the cloud service provider deletes the requested data file.

C. Data User

Data user is created by the owner. At the time of creation, secret key is sent to the user by the data owner. The secret key is the combination of master key and user attributes like user id, user department and user designation.

1) File Access

If there is a need for user to download the file, then user must send the request by specifying the file name or file id with secret key to the gateway server. Gateway server accepts the request and extracts the attribute value present in the secret key of user. Gateway server sends the request for access details to access control tree by specifying file name and attributes. Once the access details of specified file is obtained by access control tree, then server matches the access structure of file with user attributes. If attributes and access structure matches then gateway server sends the request to the cloud storage and picks the requested file and calls the decrypt(CT, SKu) algorithm which decrypts the cipher-text and sends the decrypted file to the user, if match is not found, then user access rights to the requested file is denied.

2) File Availability

Availability can be achieved by using the hybrid cloud concept where the redundant copy of the file will be saved in the private cloud. Even though the file which is requested by the user has not found in public cloud due to its deletion by cloud service provider, user will get the requested file with the use of hybrid cloud concept. Initially gateway server will send the request for the public cloud by specifying requested file id, if the particular file has not found in the public cloud then gateway server will send the request to private cloud where the redundant copy is stored and picks the particular file from the private cloud, decrypts it and sends the file to user.

V. SECURITY ANALYSIS

In this section, we compare our scheme with the one proposed by [23] on security features.

- *Scalability*: compared with [23] scheme our scheme also achieves scalability, by shifting the authority rights to data owner which decreases the workload of root authority.
- *Flexibility*: compared with [23] scheme our scheme also achieves flexibility, by allowing the user attributes to organized in a recursive set structure, our scheme supports the compound attributes and multiple numerical assignments for a given attribute conveniently.

- *Fine-grained access*: compared with [23] scheme our scheme also achieves fine-grained access control, by allowing data owner to define expressive and flexible access policy for data files.

- *Data Integrity*: compared with [23] our scheme also provides the data integrity proof, by allowing the owners to ensure the integrity of data periodically using the data integrity checking concept.

- *Availability*: compared with [23] scheme our scheme provides the availability, even though the file which is stored in public cloud gets deleted by cloud service provider. Availability is achieved by using hybrid cloud concept where the redundant copy of the file is stored in private cloud.

- *Expressiveness*: compared with [23] scheme, our scheme provides expressiveness, where user keys are associated with attributes rather than access policy.

VI. RESULTS

In this section we are taking the setup operation and new user grant operation and observing the time taken to generate the key with different depths in setup operation and observing the time taken to generate the secret key with number of attributes in key structure in new user grant operation.

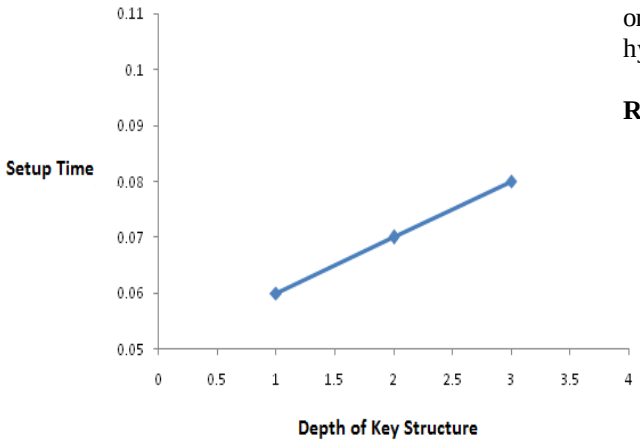

Fig 6.1 Setup operation

Fig. 6.1 shows the time required to setup the system for a different depth of key structure. Our scheme can be extended to support any depth of key structure. Setup can be completed in constant time for a given depth.

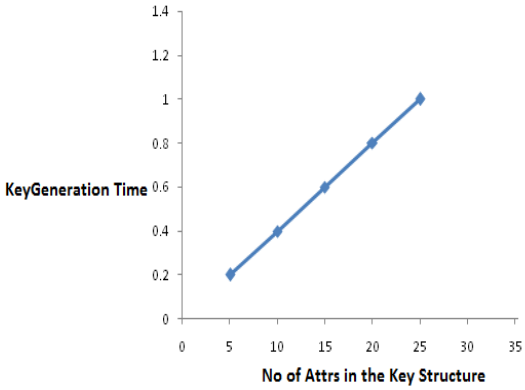

Fig 6.2. New User Grant

Fig. 6.2 shows the time required to generate the key considering number of attributes in the key structure.

VII. CONCULISON

In this paper, we proposed the Hybrid Cloud concept with for HASBE scheme, to overcome the drawback which was present in the HASBE scheme and to prove the availability and data integrity in the cloud. Even though HASBE scheme achieves scalability, flexibility, data integrity and fine grained access control, it fails to provide the availability of the data to user, even during the fault occurrence situation in the cloud. However, the fact that owner no longer have physical possession of data indicates that they are facing a potentially formidable risk for missing the data, because sometimes the cloud service provider deletes the data in the cloud without the knowledge or permission of data owner. Hence in order to avoid this security risk, we introduce a hybrid cloud concept in this paper.

REFRENCES

[1] R.Buyya, C.ShinYeo, J.Broberg I Brandic,"cloud computing emerging it platform-vision,hype,reality for delivering computing as the 5th utility,"Future Generation Comput.Syst., vol. 25, pp.599-616, 2009.

[2] Amazon Elastic Compute cloud (amazon ec2)[online].Available: http://aws.amazon.com/ec2/

[3] Amazon web services(AWS)[online]. Available: https://s3.amazonzws.com/

[4] R.Martin, "IBM brings cloud computing to earth with massive new data centers," information week Aug.2008[online].Available :http://www.informationweek.com/news/hardware/data_centers/209901523

[5] Google App engine[online]. Available: http:://code.google.com/appengine/

[6] K.Barlow and J.Lane, "like technology from an advanced alien culture: google apps for education at ASU,"in proc. ACM SIGUCCS User services conf.,2007

[7] B.Barbara, "salesforce.com: Raising the level of networking," Inf. Today, vol.27,2010.

[8] J.Bell, hosting enterprise data in the cloud-part 9: investmentvaluezetta,tech. rep.,2010

[9] A.Ross," technical perspective: a chilly sense of security,"commun.ACM,vol.52,pp.90-90,2009

[10] D.E.Bell and L.J.Lapadula, secure computer systems: unified exposition and multics interpretation the MITRE Corporation, Tech.Rep.,1976

[11] K. J. Biba, Integrity Considerations for Secure Computer Systems the MITRE Corporation, Tech. Rep., 1977.

[12] H. Harney, A. Colgrove, and P. D. McDaniel, "Principles of policy in secure groups," in *Proc. NDSS*, San Diego, CA, 2001.

[13] P. D. McDaniel, A. Prakash, "Methods and limitations of security policy reconciliation," in *Proc. IEEE Symp. Security and Privacy*, Berkeley, CA, 2002.

[14] T. Yu and M. Winslett, "A unified scheme for resource protection in automated trust negotiation," in *Proc. IEEE Symp. Security and Privacy*, Berkeley, CA, 2003.

[15] J. Li, N. Li, and W. H. Winsborough, "Automated trust negotiation using cryptographic credentials," in *Proc. ACM Conf. Computer and Communications Security (CCS)*, Alexandria, VA, 2005.

[16] V. Goyal, O. Pandey, A. Sahai, and B.Waters, "Attibute-based encryption for fine-grained access control of encrypted data," in *Proc. ACM Conf. Computer and Communications Security (ACM CCS)*, Alexandria,VA, 2006.

[17] S. Yu, C. Wang, K. Ren, and W. Lou, "Achiving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. IEEE INFOCOM 2010*, 2010, pp. 534–542.

[18] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attributebased encryption," in *Proc. IEEE Symp. Security and Privacy*, Oakland, CA, 2007.

[19] R. Bobba, H. Khurana, and M. Prabhakaran,"Attribute-sets: A practically motivated enhancement to attribute-based encryption," in *Proc. ESORICS*, Saint Malo, France, 2009.

[20] A. Sahai and B. Waters, "Fuzzy identity based encryption," in *Proc. Acvances in Cryptology—Eurocrypt*, 2005, vol. 3494, LNCS, pp. 457–473.

[21] G.Wang, Q. Liu, and J.Wu, "Hierachical attribute-based encryption for fine-grained access control in cloud storage services," in *Proc. ACM Conf. Computer and Communications Security (ACM CCS)*, Chicago, IL, 2010.

[22] Zhiguo Wan, Jun'e Liu, and Robert H. Deng, Senior Member, IEEE, "HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing.

[23] Pallavi R, R Aparna, "Data Integrity Scheme for Hierarchical Attribute Encryption in Cloud Computing," in International Conference on computer science and information technology.,2013.