DATA CENTRIC ACCESS CONTROL SOLUTION WITH ROLE BAESD PROXY RE-ENCRYPTION

Akhila Raj[1], Dr. K. S. Angel Viji[2]

[1]Post Graduation Student, Dept. of Computer Science and Engineering, College of Engineering Kidangoor, Kerala, India
[2]Associate Professor, Dept. of Computer Science and Engineering, College of Engineering Kidangoor, Kerala, India

---

Abstract - *The security solutions currently available are based on perimeter security. However, cloud computing breaks the organization perimeters. At the point when data resides within the cloud, they reside outside the structure bounds of an organization. Users may loss control over their data and it raises reasonable security issues that block the adoption of cloud computing. Those issues includes questions like: Is that the cloud service provider (CSP) a truthful person or he accessing the data? Is it genuinely apply the access control policy defined by the user? This paper presents a data-centric access control solution with role-based expressiveness in which security is focused on protecting user data against the cloud service provider. In this project an identity-based and proxy re-encryption techniques are used to protect the authorization model. Data are doubly encrypted and stored in cloud to protect it against the cloud service provider that hold it. A de-duplication and a regeneration techniques are used to better access and management of data. Seed-block algorithm is used to regenerate the missing files.*

Keywords: **Authorization, Encryption, Proxy re-encryption, Role based access control.**

1. INTRODUCTION

Cloud computing is rising computing technology that uses Internet[1]. It consists of the use of computing resources that are delivered as a service over a network. In cloud computing model users have to give access to their data for storing and performing the desired business operations. Cloud computing is a large-scale distributed computing paradigm [2]. Hence cloud service provider must provide the trust and security, as there is valuable and sensitive data in huge amount stored on the clouds. There are concerns about flexible, scalable and fine grained access control in the cloud computing. There are some security requirements, such as data encryption, key management, identity authentication, and access control [3]. For this purpose, there have been many of the schemes, proposed for encryption. Such as simple encryption technique that is classically studied.

We are going to discuss about the Attribute-Based Encryption (ABE) schemes and how it has been developed and modified further into Key Policy. Attribute based encryption (KP-ABE). Cipher-text Policy Attribute Based Encryption (CP-ABE) and further it has been proposed as CP-ASBE and furthermore HABE and HASBE so on. This is according to how flexible, scalable and fine grained access control is provided by each scheme. Cloud computing has rapidly become a widely adopted paradigm for delivering services over the internet. Therefore cloud service provider must provide the trust and security, as there is valuable and sensitive data in large amount stored on the clouds. For protecting the confidentiality of the stored data, the data must be encrypted before uploading to the cloud by using some cryptographic algorithms.

The importance of access control is to authenticate the user to perform actions and operations. To restrict the subject or the group from accessing, a Discretionary Access Control (DAC) model is used. This model restricts part of the session's elements because it uses access control matrix for setting the policy. But this type of model support does not security level restriction and multi-policy. Mandatory Access Control (MAC) model is same like the DAC model, but a difference is that MAC support, security level restriction due to MAC sets a secret class on the target and the subject. An RBAC (Role-Based Access Control) approach, it is a role-centric model. In that, roles can be well accepted by their names, and they decide the permissions be granted to users. An ABAC (Attribute-Based Access Control) approach, It is an attribute-centric model. In this model, the permissions are granted to the user depends on their attributes and that attributes must select by expert staff or personnel.

A RBAC may require the large definition of roles for fine-grain authorization and ABAC is easier to set up without making an effort to determine the role as an RBAC model. On the other hand, In ABAC model, ABAC may result in a huge number of rules that is a system with n attributes then it would have possible rule combinations up to 2^n. An ABAC splits authorization rules from user attributes, making it difficult to decide permissions available to a particular user, while RBAC is role-centric and user privileges can be easily decided by the data owner. A rule-based approach following the RBAC scheme is to propose for authorization solution, where roles are used to assign the privileges to the user for data access. This approach can help to control and manage security based on the cloud access that is access data from a cloud by authorized users.

Present an authorization solution which provides a rule-based approach containing RBAC (Role-Based Access Control) scheme. A data-centric access control solution that is SecRBAC, for self-protected data that can run with untrusted CSPs (Cloud service provider) and thus provides extended Role-Based Access Control expressiveness on data.

2. LITERATURE SURVEY

2.1 Attribute based encryption (ABE)

Sahai and Waters [4] first introduced the attribute based encryption (ABE) for enforced access control through public key cryptography. The main goal for these models is to provide security and access control. The main aspects are to provide flexibility, scalability and fine grained access control. In classical model, this can be achieved only when user and server are in a trusted domain. But what if their domains are not trusted or not same? So, the new access control scheme that is Attribute Based Encryption (ABE) scheme was introduced which consist of key policy attribute based encryption (KP-ABE). As compared with classical model, KP-ABE provided fine grained access control. However it fails with respect to flexibility and scalability when authorities at multiple levels are considered.

In ABE scheme both the user secret key and the ciphertext are associated with a set of attributes. A user is able to decrypt the cipher-text if and only if at least a threshold number of attributes overlap between the cipher-text and user secret key. Different from traditional public key cryptography such as Identity-Based Encryption, ABE is implemented for one-to many encryption in which cipher-texts are not necessarily encrypted to one particular user, it may be for more than one number of users.

In Sahai and Waters ABE scheme, the threshold semantics are not very expressive to be used for designing more general access control system. Attribute-Based Encryption (ABE) in which policies are specified and enforced in the encryption algorithm itself. The existing ABE schemes are of two types. They are Key-Policy ABE (KP-ABE) scheme and Ciphertext-Policy ABE (CP-ABE) scheme.

2.2 Key Policy Attribute Based Encryption (KP-ABE)

To enable more general access control, V. Goyal, O. Pandey, A. Sahai, and B. Waters [4], [5] proposed a key-policy attribute-based encryption (KP-ABE) scheme. It is the modified form of classical model of ABE. Exploring KP-ABE scheme, attribute policies are associated with keys and data is associated with attributes. The keys only associated with the policy that is to be satisfied by the attributes that are associating the data can decrypt the data. Key Policy Attribute Based Encryption (KP-ABE) scheme is a public key encryption technique that is designed for one-to-many communications. In this scheme, data is associated with the attributes for which a public key is defined for each. Encrypter, that is who encrypts the data, is associated with the set of attributes to the data or message by encrypting it with a public key.

Users are assigned with an access tree structure over the data attributes. The nodes of the access tree are the threshold gates. The leaf nodes are associated with attributes. The secret key of the user is defined to reflect the access tree structure. Hence, the user is able to decrypt the message that is a ciphertext if and only if the data attributes satisfy the access tree structure. In KP-ABE, a set of attributes is associated with ciphertext and the user's decryption key is associated with a monotonic access tree structure. When the attributes associated with the ciphertext satisfy the access tree structure, then the user can decrypt the ciphertext. In the cloud computing, for efficient revocation, an access control mechanism based on KP-ABE and a re-encryption technique used together. It enables a data owner to reduce most of the computational overhead to the servers.

The KP-ABE scheme provides fine-grained access control. Each file or message is encrypted with a symmetric data encryption key (DEK), which is again encrypted by a public key, that is corresponding to a set of attributes in KP-ABE, which is generated corresponding to an access tree structure. The encrypted data file is stored with the corresponding attributes and the encrypted DEK. If and only if the corresponding attributes of a file or message stored in the cloud satisfy the access structure of a user's key, then the user is able to decrypt the encrypted DEK.

2.3 Cipher Text Policy Attribute Based Encryption (CP-ABE)

Sahai et. al[6] introduced the concept of another modified form of ABE called CP-ABE that is Ciphertext Policy Attribute Based Encryption. In CP-ABE scheme, attribute policies are associated with data and attributes are associated with keys and only those keys that the associated attributes satisfy the policy associated with the data are able to decrypt the data. CP-ABE works in the reverse way of KP-ABE. In CP-ABE the ciphertext is associated with an access tree structure and each user secret key is embedded with a set of attributes. In ABE, including KP-ABE and CP-ABE, the authority runs the algorithm Setup and Key Generation to generate system MK, PK, and user secret keys. Only authorized users (i.e., users with intended access structures) are able to decrypt by calling the algorithm Decryption.

In CP-ABE, each user is associated with a set of attributes. His secret key is generated based on his attributes. While encrypting a message, the encryptor specifies the threshold access structure for his interested attributes. This message is then encrypted based on this access structure such that only those whose attributes satisfy the access structure can decrypt it. With CP ABE technique, encrypted data can be kept confidential and secure against collusion attacks.

2.4 Cipher Text Policy Attribute-Set Based Encryption (CP-ASBE)

As compared to CP-ABE scheme in which the decryption keys only support user attributes that are organized logically as a single set, so users can only use all possible combinations of attributes in a single set issued in their keys to satisfy policies. To solve this problem, ciphertext-policy attribute-set based encryption (CP-ASBE or ASBE for short) is introduced by Bobba, Waters et al.[7], [8] ASBE is an extended form of CPABE which organizes user attributes into a recursive set structure.

Ciphertext Policy Attribute Set Based Encryption (CP-ASBE) is a modified form of CP-ABE. It differs from existing CP-ABE schemes that represent user attributes as a monolithic set in keys. It organizes user attributes into a recursive set based structure and allows users to impose dynamic constraints on how those attributes may be combined to satisfy a policy. The CP-ASBE consists of recursive set of attributes.

2.5 Identity Based Encryption (IBE) and Hierarchical Identity Based Encryption (HIBE)

In an identity-based encryption scheme, data is encrypted using an arbitrary string as the key and for decryption; a decryption key is mapped to the arbitrary encryption key by a key authority. Hierarchical Identity Based Encryption (HIBE) is the hierarchical form of a single IBE[9]. The concept of HIBE scheme can help to explain the definition of security. In a regular IBE (1-HIBE) scheme; there is only one private key generator (PKG) that distributes private keys to each users, having public keys are their primitive ID (PID) arbitrary strings.

A two-level HIBE (2-HIBE) scheme consists of a root PKG, domain PKGs and users, all of which are associated with PIDs. A users public key consists of their PID and their domains PID (in combine, called an address). In a 2-HIBE, users retrieve their private key from their domain PKG. Domain PKGs can compute the private key PK of any user in their domain, provided they have previously requested their domain secret key-SK from the root PKG. Similarly, is for number of sub-domains. There also includes a trusted third party or root certificate authority that allows a hierarchy of certificate authorities: Root certificate authority issues certificates for other authorities or users in their respective domains. The original system does not allow for such structure. However, a hierarchy of PKGs is reduces the workload on root server and allows key assignment at several levels.

2.6 Hierarchical Attribute-Base Encryption (HABE) and Hierarchical Attribute Set Based Encryption (HASBE)

This scheme Hierarchical attribute-based encryption (HABE) is derived by Wang et.al[7]. It is designed to achieve fine-grained access control in cloud storage services. It is a combination of HIBE and CP-ABE. In the HABE scheme, there are multiple keys with different usages. Therefore, we first provide a summary of the most relevant keys to serve as a quick reference. HASBE scheme is proposed and implemented by Zhiguo Wan et.al . The cloud computing system consists of five types of parties: a cloud service provider, data owners, data consumers, a number of domain authorities, and a trusted authority. The cloud service provider manages a cloud and provides data storage service.

Data owners encrypt their data files and store them in the cloud for sharing with data consumers. To access the shared data files, data consumers download encrypted data files of their interest from the cloud and then decrypt them. Each data owner/consumer is administrated by a domain authority. A domain authority is managed by its parent domain authority or the trusted authority. The trusted authority is responsible for managing top-level domain authorities. It is root level authority. For example, for an IT enterprise, employees are kept in the lowest domain level and above that there is department and above that there is top level of domain we call it as a trusted domain. It generates and distributes system parameters and also root master keys. And it authorizes the top-level domain authorities.

A domain authority delegates the keys to its next level sub-domain authorities. Each user in the system is assigned a key structure. Key specifies the attributes associated with the users decryption key. Zhiguo Wan et. al given a HASBE scheme for scalable, flexible, and fine-grained access control in cloud computing.

The HASBE scheme consists of hierarchical structure of system users by using a delegation algorithm to CPASBE. HASBE supports compound attributes due to flexible attribute set combinations as well as achieves efficient user revocation because of attributes assigned multiple values. Thus, it provides more scalable, flexible and fine grained access control for cloud computing. HASBE combines the functionalities of HIBE and ASBE. HASBE scheme seamlessly incorporates a hierarchical structure of system users. It uses a delegation algorithm to ASBE. Out of these schemes, the HASBE scheme provides more scalable, flexible and fine-grained access control than any other schemes in cloud computing.

3. PROXY RE-ENCRYPTION AND IDENTITY- BASED

ENCRYPTION

SecRBAC makes use of cryptography to protect data when moved to the Cloud[10]. Advanced cryptographic techniques are used to protect the authorization model in order to avoid the CSP being able to disclose data without data owner consent. Concretely, the solution is based on Proxy Re-

Encryption (PRE). A PRE scheme is a cryptographic scheme that enables an entity called proxy to re-encrypt data from one key to another without being able to decrypt it. That is, given a couple of key pairs α and β, the proxy could re-encrypt a ciphertext c_α encrypted under a public key to another ciphertext c_β that can be decrypted using β private key. Using this kind of cryptography, a user u_α can encrypt a piece of data m using his own public key pub_α to obtain a ciphertext c_α. A re-encryption key $rk_{\alpha\rightarrow\beta}$ can be generated for a proxy to re-encrypt from α to β, thus transforming c_α to another ciphertext c_β. Then, another user u_β can use his own private key $priv_\beta$ to decrypt c_β and obtain the plain piece of data m.

The following set of functions is provided by IBPRE. It constitutes the cryptographic primitives for the proposal:

$$setup(k, p) \rightarrow (p, msk) \quad (1)$$

$$keygen(p, msk, id_\alpha) \rightarrow sk_\alpha \quad (2)$$

$$encrypt(p, id_\alpha, m) \rightarrow c_\alpha \quad (3)$$

$$rkgen(p, sk_\alpha, id_\alpha, id_\beta) \rightarrow rk_{\alpha\rightarrow\beta} \quad (4)$$

$$reencrypt(p, rk_{\alpha\rightarrow\beta}, c\alpha) \rightarrow c_\beta \quad (5)$$

$$decrypt(p, sk_\alpha, c_\alpha) \rightarrow m \quad (6)$$

A brief description of each function follows.

1) **Initializes the cryptographic scheme**: It takes as input a security parameter k to initialize the cryptographic scheme (e.g., parameters to generate an elliptic curve) and outputs both the Master Secret Key msk and a set of public parameters p that is used as input for the rest of functions.
2) **Generates Secret Keys**: It takes as input the msk and an identity id_α; and outputs the Secret Key sk_α corresponding to that identity.
3) **Encrypts data**: It takes as input an identity id_α and a plain text m; and outputs the encryption of m under the specified identity c_α.
4) **Generates Re-encryption Keys**: It takes as input the source and target identities id_α and id_β as well as the Secret Key of the source identity sk_α; and outputs the Re-encryption Key $rk_{\alpha\rightarrow\beta}$ that enables to re-encrypt from id_α to id_β.
5) **Re-encrypts data**: It takes as input a ciphertext ca under identity idα and a Re-encryption Key rkα→β; and outputs the re-encrypted ciphertext c_β under identity id_β.
6) **Decrypts data**: It takes as input a ciphertext c_α and its corresponding Secret Key sk_α; and outputs the plain text m resulting of decrypting c_α.

4. PROPOSED FRAMEWORK

The purpose of this paper is to make data accessing from a cloud, as easy as possible. So this is created as an organization basis. For that purpose a website is created using Microsoft Visual Studio as the integrated development environment and .NET with C# language as the framework.

4.1 Problem Statement

Let O be an organization, E = $(e_1, e_2,\ldots, e_n)$ be the set of employees in that organization. Let R = $(r_1, r_2,\ldots, r_n)$ be the set of roles to be assigned to the employees and F = $(f_1, f_2,\ldots,f_n)$ be the set of double encrypted files in the cloud, then,

$$e_i \rightarrow f_i, \text{ iff } r_i(e_i) \geq pr(f_i) \quad (1)$$

In equation(1) where, $e_i$is the employee, f_i is double encrypted file, r_i is the role and $pr(f_i)$ permitted role of file. The encryption is done as,

$$CS=RE(p, rk_{\alpha\rightarrow\beta}, [E(p,id_\alpha,m)]) \quad (2)$$

In equation(2) where, p is the set of public parameter, $rk_{\alpha\rightarrow\beta}$ is the re-encryption key, id_α is the identity and m is the plain text. Decryption is done as,

$$m=RD(p,sk_\beta, [D(p, id_\alpha, c_\alpha)]) \quad (3)$$

In equation (3) where, m is the plain text, p is the set of public parameter, sk_β is the double- decryption key, id_α is the identity and c_α is the ciphertext.

4.2 System Architecture and Working

The proposed system consists of two types of data one is normal data and another one is compressed data. In that, we are presenting a data-centric access control solution with improved role-based expressiveness in which security is focused on protecting user data regardless the cloud service provider (CSP) that holds it. When data move to the cloud, data owner generates a self-protected package. This contains the authorization rules, the encrypted data objects, and the corresponding re-encryption keys. The Fig-1 shows the basic flow and working of the proposed architecture.

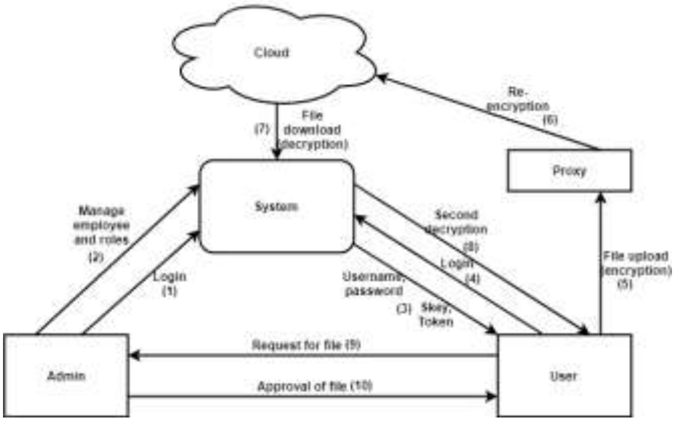

Fig -1: System Architecture

The Fig-1 shows the overall working of the system. It include mainly three modules such as admin(CEO), user/employee, and proxy. Admin have the overall control to the system and he have the responsibility to manage employees and roles. And also control over files. Employees are responsible for uploading files to the system. When uploading files into the cloud two algorithms are used. Symmetric and asymmetric algorithms. When an employee upload a file, it will symmetrically encrypted by using AES algorithm. Then file upload to the cloud by the proxy, the encrypted file will again encrypted using the asymmetric algorithm, RSA algorithm.

Double encryption is done to protect the data. Data owner upload files to the system. Firstly the user/ employee upload file to the system, at that time the file is encrypted using a symmetric encryption algorithm. Then the proxy re-encrypt the file using an asymmetric encryption algorithm and upload that file to the cloud storage.

De-duplication and regeneration are here for advanced management of data. De-duplication means when an employee upload an already existed file to the cloud, it will not upload to the cloud. Instead that existed file stored in system and also a reference will updated for the newly arrived file. Regeneration means generate a missing file. Which means, suppose a file is missing from the cloud. When an employee trying to download that missing file, he cannot access it. So to avoid this problem, copy of each file is XORed and saved in the system.

Working of seed block algorithm is like, when an employee upload a file to the system, copy of the file is taken and XOR the file and saved it into the system. When an employee trying to download a missing file, the seed block algorithm first decrypt the XORed file and encrypt and also re-encrypt and upload to the cloud. Normally proxy module upload the files to the cloud. But in the case of a missing file double encryption is done by the seed block algorithm itself.

5. RESULTS AND ANALYSIS

Based on the literature survey SecRBAC is the better method for data accessing and for encrypting the file. This project is for making data accessing as easy as possible. This project presents SecRBAC, a data-centric access control solution for self-protected data that can run in untrusted CSPs and provides extended Role-Based Access Control expressiveness. The proposed authorization solution provides a rule-based approach following the RBAC scheme, where roles are used to ease the management of access to the resources. This approach can help to control and manage security and to deal with the complexity of managing access control in cloud computing.

A data-centric approach is used for data self-protection, where cryptographic techniques such as Proxy Re-Encryption Encryption (PRE) is used. They allow to re-encrypt data from one key to another without getting access and to use identities in cryptographic operation. These techniques are used to protect both the data and the authorization model. Each piece of data is ciphered with its own encryption key linked to the authorization model and rules are cryptographically protected to preserve data against the service provider access or misbehavior when evaluating the rules.

Double encryption and also double decryption is done to protect the data. Firstly the user/ employee upload file to the system, at that time the file is encrypted using a symmetric encryption algorithm. Then the proxy re-encrypt the encrypted file using an asymmetric encryption algorithm and upload that file to the cloud. The symmetric algorithm here used is AES and that of symmetric algorithm used is RSA.

Modifications done with this project are de-duplication and regeneration. De-duplication means when an employee upload an already existed file to the cloud, it will not upload to the cloud. Instead that existed file stored in system and also a reference will updated for the newly arrived file. Next modification is regeneration. Regeneration means generate a missing file. Which means, suppose a file is missing from the cloud. When an employee trying to download that missing file, he cannot access it. So to avoid this problem, copy of each file is XORed and saved in the system. A regeneration algorithm, namely seed block algorithm, is used to avoid such problem.

The analysis done with this project is, one way is used the same symmetric algorithm, AES algorithm for both the encryption and double encryption and also for decryption and double decryption. The reason for choosing AES algorithm for both the encryption is, we can upload any type of files to the cloud. And the second way is used the AES and RSA algorithms to upload and download files to and from cloud. That is, employee encrypt file using symmetric algorithm AES and proxy using the asymmetric algorithm RSA.

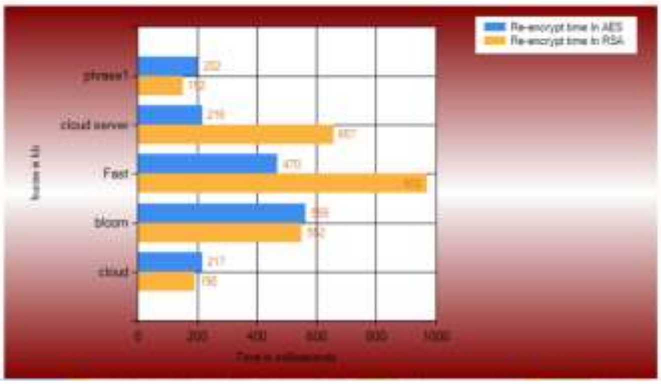

Chart-1: Comparison of re-encryption time in AES and RSA

The Chart-1 shows a graph that include comparison of re-encryption time of using AES and using RSA algorithms. The first encryption algorithm for both method is symmetric AES algorithm. So the time taken to encrypt a single file is same. So here consider re-encryption time for analysis. That is, time taken to upload the same file to cloud using RSA takes more time than AES. The X-axis shows time in milliseconds and Y-axis shows file name in kb.

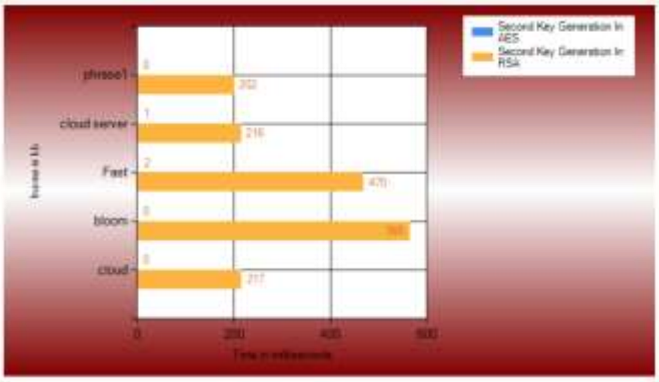

Chart-2: Second key generation time

The Chart-2 shows a graph that shows the difference in second key generation time. It can see that the time taken to generate the key for RSA algorithm make big difference from that of AES algorithm. The X-axis shows time in milliseconds and Y-axis shows file name in kb.

6. CONCLUSION

A data-centric authorization solution has been proposed for the secure protection of data in the Cloud. SecRBAC allows managing authorization and provides enriched role-based expressiveness including role hierarchies. Access control computations are delegated to the CSP, being this not only unable to access the data, but also unable to release it to unauthorized parties. Advanced cryptographic techniques have been applied to protect the authorization model. A re-encryption key complement each authorization rule as cryptographic token to protect data against CSP misbehavior. The solution is independent of any PRE scheme or implementation as far as three specific features are supported. An IBPRE scheme has been used in this paper in order to provide a comprehensive and feasible solution. To avoid duplicated files de-duplication is done and to manage missing files regeneration technique is used.

REFERENCES

1. **Sphurti Atram, N.R. Borkar**, A Paper on Attribute Based Encryption Schemes in Cloud Computing, International Journal of Computer Science and Mobile Computing,*Vol. 6, Issue. 5, May 2017, pg.260 – 266.*

2. **Shawish A., Salama M.**,Cloud Computing: paradigms and technologies, In Inter-cooperative collective intelligence: Techniques and application Springer,pp. 39-67.

3. **Huang J. Y., Chiang C. K. , and Liao,** 2013. An efficient attribute-based encryption and access control scheme for cloud storage environment. In International Conference on Grid and Pervasive Computing, Springer, pp. 453–463.

4. **Bethencourt, J., Sahai, A., and Waters, B.**, 2007. Attribute-based encryption. In Security and Privacy, 2007. SP'07. IEEE Symposium on, IEEE, pp. 321–334

5. **Goyal, V., Pandey, O., Sahai, A., and Waters, B.,** 2006. Attribute-based encryption for fine-grained access control of encrypted data . In Proceedings of the 13th ACM conference on Computer and communications security, Acm, pp. 89–98.

6. **Wang, G., Liu, Q., and Wu, J.**, 2010. Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. In Proceedings of the 17th ACM conference on Computer and communications security, ACM, pp. 735–737.

7. **Bobba, R., Khurana, H., and Prabhakaran, M.**, 2009. Attribute-sets: A practically motivated enhancement to attribute-based encryption. In European Symposium on Research in Computer Security, Springer, pp. 587–604.

8. **Sahai, A., and Waters, B.,** 2012. Attribute-based encryption for circuits from multilinear maps. arXiv preprint arXiv:1210.5287.

9. **Wan, Z., Liu, J., and Deng, R. H.,** 2012. Hasbe: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing.

IEEE transactions on information forensics and security, 7(2), pp. 743–754.

10. **Perez, J. M. M., Perez, G. M., and G´omez, A. F. S.**, 2017. Secrbac: Secure data in the clouds. IEEE Transactions on Services Computing, 10(5), pp. 726–740.

11. **K. Jung, K. I. Kim, A. K. Jain**, Text information extraction in images and video: a survey, Pattern recognition 37 (5) (2004) 977–997.

BIOGRAPHIES

Akhila Raj, She is a post graduation student in College of Engineering Kidangoor. Specialization in Computer and Information Science. She received the graduation in computer science and engineering from Caarmel Engineering College, Ranny.

Dr.K.S. Angel Viji, is currently working as Associate Professor, Department of Computer Science and Engineering, College of Engineering Kidangoor. She is having 12 years of teaching experience and 8 years of research experience. She did her BE and M.E in computer science and engineering under Anna University Chennai. She did her Ph.D in Noorul Islam university. Her area of interest includes medical image processing and network security. She is a member of IEEE. She is having more than 40 national and international conference and journal publications.