www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 5 May 2015, Page No. 12009-12014

Secure Role Based Cloud System By Integrating Trust and Role Based Encryption Scheme

Bokefode Jayant D[1], Ubale Swapnaja A.[2], Pingale Subhash V.[3], Rajguru Abhijit A.[4]

[1]Solapur University, Sinhgad College of Engineering,Korti,
Pandharpur, Solapur,India
bokefode.jayant@gmail.com

[2] Solapur University, Sinhgad College of Engineering,Korti,
Pandharpur, Solapur,India
swapnaja.b.more@gmail.com

[3] Solapur University, Sinhgad College of Engineering,Korti,
Pandharpur, Solapur,India
sub.pingale83@gmail.com

[4] Solapur University, Sinhgad College of Engineering,Korti,
Pandharpur, Solapur,India
abhijitcse08@gmail.com

Abstract: ***Now a day's cloud based systems are getting more attraction as the number of users data is rapidly increasing. The organizations are worried about the storing of data and its security. The Role Based Encryption Scheme (RBAC) provides a way to the users for managing and sharing the data in a cloud. The RBAC will combine cryptographic techniques and access controls to secure the data. With the use of these techniques the owner of data will encrypt the data in such a way that the appropriate user of the role can decrypt and view the data. In this paper we provide a hybrid cloud in which the organization data will be stored in a private cloud while the authorized users are allowed to store their data in public cloud. The role based encryption (RBE) scheme will be helpful to solve the security issue. In this paper we are integrating the RBE scheme using the AES algorithm for encryption and decryption of the data. The paper also shows the mathematical model for calculating the trust of the user. This can be used by the owner when he wants to upload any data on the cloud.***

Keywords: Role Based Access Control, AES, RSA, Cloud computing, Trust Management

1. Introduction

Cloud computing is one of the emerging and promising field in Information Technology. It provides services to an organization over a network with the ability to scale up or down their service requirements. Cloud computing services are established and provided by a third party, who having the infrastructure. Cloud computing having number of benefits but the most organizations are worried for accepting it due to security issues and challenges having with cloud. Security requirements required at the enterprise level forces to design models that solves the organizational and distributed aspects of information usage. Such models need to present the security policies intended to protect information against unauthorized access and modification stored in a cloud. The proposed work describes the approach for modeling the security requirements from the perspective of job functions and tasks performed in an organization by applying the cryptography concepts to store data on cloud with the smallest amount of time and cost for encryption and decryption processes. In this work, we used RSA and AES algorithm for encryption and decryption of data and role based access control model is used to provide access according to the role played by user. This paper also shows the mathematical model for calculating the trust of the user. This model gives the uploading rights to the user when he/she recommended by the Administrator and Owner when users exceeds the specified experience and trust threshold value.

There are various online services in market which provides the storage for the users so that the user can access it from anywhere. But the increase in cost and complexity issue is becoming critical for all these online services. The cloud providers such as the cloud storage service providers are providing solutions to these problems of storing and managing users' data online. The online service providers can outsource their data on the public cloud and can provide quality of service to the users. But the security issue arises when it comes to public cloud as because it's an open platform anyone can access this cloud. There have been various security issues as how to prevent unauthorized access of data. One way to provide the security will be the use of RBAC scheme [2] which is managing the access rights for each of the user. There are various models has been proposed in the literature some of them uses keys. There exist many hierarchy access control schemes [3] [4] which have been developed based on HKM schemes, and approaches using HKM schemes for enforcing RBAC policies for storing of data are discussed in [7]. If there is a large number of owners and users involved, setting up the key infrastructure overhead can be very high. When a user's access permission is revoked, all the keys known to this user also all the public values related to these keys need to changed, which makes these schemes impractical. In ABAC, access is granted based on attributes of the user. Systems define combination of attributes as the access policies, and users need to prove that they have these attributes in order to gain access. In 2006, the first attribute-based encryption (ABE) scheme was proposed in [5] based on the work in [6], and some other ABE schemes have been proposed afterwards. In these schemes, data is encrypted to a set of attributes, and users who have the private keys associated with these attributes can decrypt the data. These works have provided an alternative approach to secure the data stored in a distributed environment using a different access control mechanism. In traditional system there is a central authority which allocates the access permission for each of the users. The data stored in distributed fashion which will need multiple authorities. Therefore there is need to trust these many number of authorities to specify correct access policies. In cloud storage systems the data owner may wish specify some policies for those who can access their data from the cloud. These policies will be applied to the cloud providers as well. They must take permission from the owner to grant the access to data. The owner will use the encryption policy and put the data on to the public cloud. The data will be accessed to those who have given access by the owners in the cloud policy no other user can access the data. The paper is organized as follows. The section II describes the literature review. Section III describes our proposed system and its preliminaries. The Section IV presents the mathematical model of the trust. Section V shows the result of this paper and section VI concludes the paper.

2. Literature Survey

There are several cryptographic schemes, such as the the RBAC model was introduced in 1992 [1]. In this model the user can inherit the permission from other roles. In this the user who has given membership to the role has access to the permission of the given role as well as the permission that are inherited from other role. The schemes in [8] [9], developed for enforcing access policies on the data. There is a two layer encryption model proposed in the [10] which uses the Base Encryption Layer (BEL) and Surface Encryption Layer (SEL) to prevent a service provider from accessing the content but the service provider can run queries if user which has keys and need to decrypt the data. These schemes combine cryptographic technology and protect the privacy of the data. These schemes ensure that the data will be available to those users with specific role that are authorized by the data owners can only decrypt the data no other user including cloud provider can decrypt the data. The paper [11] proposes a new Role Based Encryption (RBE) scheme in which the user management is decentralized to the individual roles, the user membership is managed by the single role also it keeps the cipher text and decryption key constant in size. There is another scheme proposed in [12] in which there are various RBAC schemes such as constrained RBAC, flat RBAC and hierarchical RBAC. The cryptographic RBAC will integrate the RBE scheme with the RBAC model for enforcing the access policies in untrusted environment of the cloud. This scheme allows data to be encrypted for specific role and the members of roles will be able to decrypt and view the data. In this paper, we consider trust models for cloud storage systems that are using cryptographic RBAC schemes like the RBE, where each individual role can manage their user memberships without the need of involving the administrators. It is worth noting that the security of a RBAC system using one of these schemes is under the assumption that the managers of roles behave in a trusted manner so they do not breach the RBAC policies. However, in a cloud storage system that uses RBAC to control the access to the data, an authorized user may be excluded from accessing the permissions of the role that have been legitimately assigned to the user by a malicious administrator of the system; or an unauthorized user of the system may be granted the membership of the role by malicious administrators. Such issues rely on trust aspects in these systems. In such systems, data owners need to consider the trust of the roles with whom they wish to interact instead of the administrator of system. When the owner evaluate the trust value of role, will only proceed with encrypting data to the role if the trust value of the role is above a certain trust threshold. In RBAC, access decisions are depends on the roles the individual users have. It simplifies the administration and management of permissions; roles can be updated without updating the permissions for every user on an individual basis. The user which has inherited the role form another user will also be able to decrypt the data. The encryption will provide more security to the cipher text and those who are members of that role will only can decrypt the data.

3. Proposed System

3.1 Components of Architecture

The cloud deployment models should be assessed in terms of security. They are public, private, hybrid clouds. These models are summarily described in the following subsections.

1) Public Cloud: Public cloud is a third party cloud provider which resides outside the infrastructure of the organizations and organizations outsource their users' encrypted data to the public cloud. Since the public cloud is untrusted, data stored in the public cloud could be accessed by unauthorized parties, such as employees of the cloud provider and users from other organizations who are also using services from the same cloud. Therefore only public information and encrypted data will be stored in the public cloud. An untrusted public cloud may deny a user's request for accessing stored data in the cloud or provide users with incorrect data. Such behaviors will result in the users not being able to access the data stored in cloud but will not cause violation of RBAC policies. These behaviors can be detected, as a user can observe the failure immediately after s/he communicates with the public cloud. In this case, organization may choose to change the cloud provider to a more reliable one, especially if the current provider is found to be malicious.

2) Private Cloud: Private cloud is built on an internal data centre that is hosted and operated by a single organization. The organization only stores critical and confidential information in this private cloud. The amount of this in public cloud, so this cloud does not need to have the information is relatively small comparing to the data stored capacity to handle large volumes of data. The private cloud only provides interfaces to the administrator and role managers of the role-based system and to the public cloud. Users do not have direct access to the private cloud. This helps to reduce the attack surface of the private cloud. The purpose of using a private cloud is to ensure that correct and up-to-date information about the organization's structure and user membership are used in the decision making. To achieve efficient user revocation, the private cloud is assumed to be honest-but-curious in order to use the proposed scheme in this architecture. That is, the cloud will faithfully execute the scheme and will not collaborate with revoked users.

3) Hybrid Cloud: This cloud is a mixture of the two or more clouds. In this the public cloud and private cloud both are used. In this it integrates the advantages of each one for overcoming the others obstacle. The private cloud will not be available for the user. The user will only interact with the public cloud and the administrator of the system will be allowed to access the private cloud. This model is managed both by the third party entity and organization. It can be placed in the onsite or off site location.

4) Role Manager: A role manager is the party who manages the relationship between users and roles. When updating the user membership of a role, the role manager needs to compute new role parameters and update them in the private cloud. None of users are affected by this operation, so role managers do not need to communicate with users, and they only need to interact with the private cloud. Before a user is included into a role, the role manager will need to authenticate the user in order to ensure that the user is qualified user.

3.2 System Modules

RBE scheme has the following four types of entities. SA is the authority who generates the keys for users and roles, and to define the role hierarchy. RM is a role manager who manages the user membership of a role. Owners are the parties who want to store their data securely in the cloud. Users will want to access and decrypt the stored data in the cloud. We define the following algorithms for our RBE scheme:

1) Setup: Setup takes as input the security parameter _ and outputs a master secret key mk and a system public key pk. mk will be kept secret by system admin while pk will be for available for all users.

2) Extract: Executed by the SA to generate the key associated with the identity ID. if ID is user, the decryption key is returned to the user. If ID is for role, the secret key is returned to the role manager.

3) Manage Role: Executed by the SA to manage a role with the identity IDR in the role hierarchy.

4) Encrypt: Encrypt is executed by the owner of a message M. This algorithm takes as input the system public key pk, the role public parameters pubR, and outputs a tuple (C, K), where C will be a ciphertext, and K is the key used to encrypt the message M. The system uses a secure encryption scheme Enc, which takes K as the key space, for encrypting messages. The members of role can only decrypt the message encrypted by the secure encryption scheme. After completing this the owner will upload this encrypted message on the cloud.

5) Trust Evaluation: When owner want to encrypt the data for any role he will first look out for the trust value for given role. Upon receiving the trust value from the trust engine the owner decides whether to trust the user for given role or not. If the owner trusts the given role he will encrypt and store the data onto the cloud. If the owner founds any leak of the data for unqualified user he will report it to the role behavior auditor. Upon receiving the feedback from the owner the auditor will first checks that is the owner is authorized? If he is a valid owner the feedback will be forward to the central repository. This will affect the trust value as the trust engine is taking the input from this repository. When the owner doesn’t know who leaked the data, he can report this to user behavior failure record for a role. As we are using the role inheritance it will affect other roles as well. The role later on can check out who has leaked the data. When the identity will be matched with malicious user it can be reported to the user behavior auditor.

6) Decrypt: Executed by a user who is a member of the role R. This algorithm takes as input the system pk (public key), pubR (role public parameters), dk (user decryption key), the part C (ciphertext downloaded from cloud) , and outputs message encryption key K 2 K. The key can then be used to decrypt the ciphertext part to obtain the message M.

3.3 Experience Based Trust

Experience based trust uses the past history of the user to build the trust on the user. There are a range of other attributes and credentials such as different types of privileges, the state of the platform being used as well as reputations, recommendations and histories that come into play in decision making. Recently, a number of models have been developed using soft trust techniques to determine the trustworthiness of systems. Experience-based trust model is one such trust management system which enables the trust decisions to be made based on the historical behavior of an entity. Such a system allows an entity to rate the transactions with other entities, and the trustworthiness of an entity is determined using the collection of ratings of the transactions that other entities have had with this entity. Most experience based trust systems derive the trustworthiness of an entity from both its own experience and the feedbacks on the transactions provided by other entities which have had interactions with the entity concerned in the past. Let us consider a simple example of such a system. When a client finishes a transaction with a service provider he/s gives a feedback as either “positive” or “negative” depending on whether or not it is satisfied with the transaction. The feedback record is uploaded by the client to a trust central repository. When another client wants to evaluate the trustworthiness of the service provider it obtains the collection of feedback records from the central repository, by adding up the total number of each different type of feedback, the client gets a evidence. Then the client makes the decision whether or not to continue the transaction with the service provider based on whether this tuple exceeds a certain threshold.

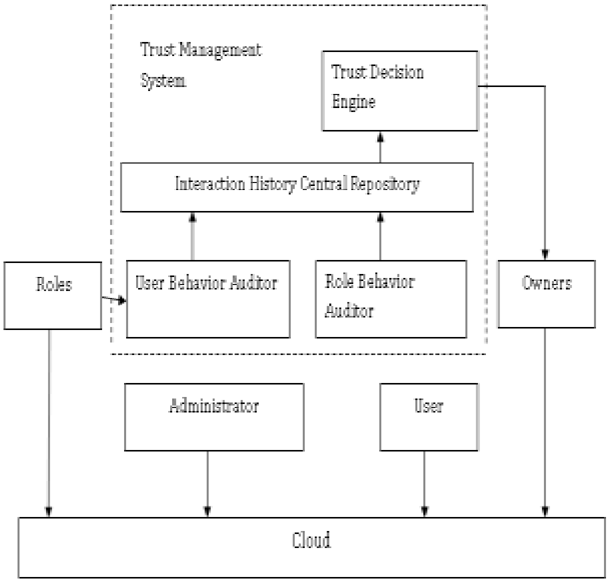


Figure 1: Trust Architecture

3.4 System Architecture

In this we first needs to create the user, assign roles to the users. This procedure contains following operations. In this we use Advanced Encryption Standard (AES) [13] [14] algorithm for generating the encryption and decryption keys. When the user is created the decryption key will be generated and will send to the user by the administrator of the system. This can now be used for decrypting of the data when user wants to gain access for any data from cloud. When administrator creates the role manager it will generate the secret key for given role. When there is need to add or remove user the role manager will perform this work. When the user wants to decrypt the data he will first request for the cipher text from the public cloud. As the decrypting values are stored in private cloud this request will be forwarded to the private cloud which will return the tuple to the public cloud with identity of user and cipher texts. The private cloud signs the given message. When user receives the cipher text he will first check for the signature whether it is received from the private cloud. After validation the user can run the decryption algorithm to recover the data. Consider the system architecture shown in Figure 1. Since our trust models are based on cryptographic RBAC schemes, our system contains all the entities that a cryptographic RBAC scheme has, which include an administrator, roles, users and owners. The administrator is the certificate authority of the RBAC system. The administrator generates the system parameters and issues all the necessary credentials. In addition, the administrator manages the role hierarchy structure of the system. To put a role into the role hierarchy structure, the administrator needs to compute the parameters for the role. These parameters represent the position of the role in the role hierarchy. They are stored in the cloud, and are available publicly.

Roles are the entities that associate users and owners together. Each role has its own role parameters which defines the user membership. The owner can be a user and can also be an external party. In this we consider the owner as separate entity even though user can be an owner and vice versa. The user will be an entity who wants to get access to the data stored in the public cloud. For accessing the data the user needs to be a qualified user and needs to have the decryption key provided by the administrator of the system. For this he needs to send request to the cloud and after receiving the response he can proceed further for accessing the data. With these components in architecture we have other components which are explained as follows. In this we have central repository which will needs to store all the interaction happened between the user and the owner. It will also store the trust related record. When the trust value needs to be calculated the trust engine will use this record for its reference. The entities which are outside the trust management system will not be able to access this repository. Another entity is the Role behavior audit which keeps track of the feedbacks stored for particular role. These feedbacks will be stored again in the central repository. The feedbacks which are given by the authorized user will be stored in the repository other will be discarded. The user behavior audit keeps track of the behavior of the users. The main purpose is to allow roles to report malicious user. If the role founds the malicious user it will report it to the central repository this will be then updated in the User behavior failure record. Another entity is The trust decision engine which takes the interaction history as input and will output you the certain trust value for the given role to the owner so that the owner can decide whether to trust the role or not.

3.5 System Parameters

and decryption of the data. The main purpose of using this algorithm is to provide more security to the data which will be uploaded on to the cloud. The system is developed in asp.net. For the cloud we are using the Linux as a platform with i5 processor. The use of latest processor will reduce the response time for uploading and delivering of the data to the owner and user respectively. The size of the decryption key is another important factor in cloud storage system. The decryption key needs to be portable as users may use the storage service from different clients. The experimental results show that the size of the decryption key is 48 bytes, which is convenient for the users. A non-constant size decryption key will usually make it difficult for the users to decide the memory requirements that are needed on the client devices to store the keys.

4. Mathematical Model

1) Trust Degree: Trust degree Tdij is used to evaluate the degree of trust from a domain set of possible trust values that trustor Ti in

$Tdij(Ti,Tj, Sk, t)$ where $i \neq j$; $0 \leq Tdij(Ti, Tj, Sk, t) \leq 1$

Where, Sk is kth service and t is defined as time. Trust degree has value between 0 and 1.Trust degree is calculated using direct trust Tddir or recommendation trust Tdrecom or if new entity joining a cloud environment first time then ignorance value Tdiv is assigned.

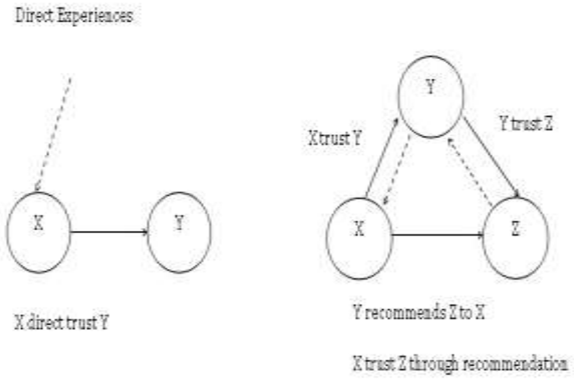


Figure 2: Trust Architecture

$$\ni Tdij(Ti,Tj,Sk,t) = \{Tdij(Ti,Tj,Sk,t)\; Tddir(Ti,Tj,Sk,t)\Theta\; Tdrecom(Ti,Tj,Sk,t)\Theta\; Tdiv(Ti,Tj,Sk,t)\} \qquad (1)$$

$$\{\, 0, ifnd = 0, nr\; Tdij(Ti,Tj,Sk,t) = Dt(Ti,Tj,Sk,t), ifnd \in \{1,2 \ldots\}, nr = 0$$

$$Rt(Ti,Tj,Sk,t), ifnr \in \{12 \ldots\}, nd = 0 \qquad (2)$$

Where, Dt(Ti ,Tj, Sk ,t) and Rt(Ti ,Tj, Sk ,t) are the direct trust degree and recommendation trust degree of trustor Ti, in view of trustee Tj about kth service Sk at time t, nd and nr are the number of direct trust degree and recommendation trust degree.

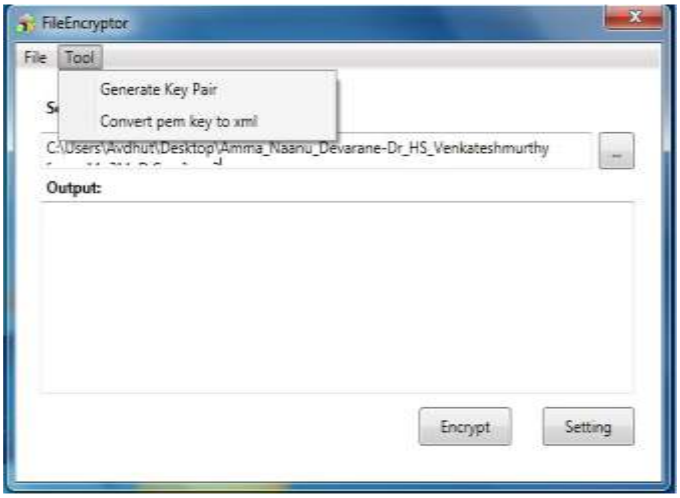
2) Trust Level: Trust level represents the trustworthiness using degree of trust. The Table I shows the satisfactory levels of the trust.

Table 1: Satisfactory Level

Level	Label	Trustworthiness
I	No Option	Tdij = 0
II	Low Distrust	0 < Tdij < 0.5
III	Medium Trust	Tdij = 0.5
IV	High Trust	0.5 < Tdij < 1
V	Complete Trust	Tdij =1

5. Result

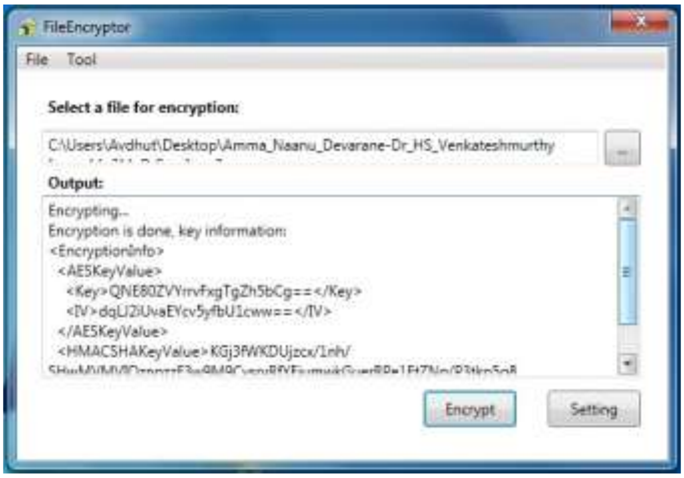
For encrypting any file first we need to create the pair of keys i.e. public key and private key. Here we use the RSA algo for generating the public and private key. The figure 3 shows the result.



```
C:\Users\Avdhut\Desktop\Amma_Naanu_Devarane-Dr_HS_Venkateshmurthy
```

Figure 3: Generate Key Pair

After that ,owner can encrypt the file. But for encrypting it he needs to provide the public key. Here we use the AES key value for encrypting the message. The figure 4 shows the encryption result.



Select a file for encryption:

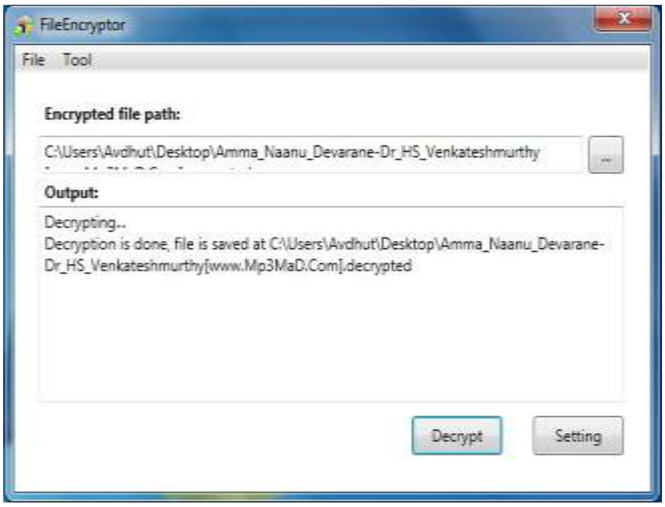
```
C:\Users\Avdhut\Desktop\Amma_Naanu_Devarane-Dr_HS_Venkateshmurthy
```

Output:

```
Encrypting...
Encryption is done, key information:
<EncryptionInfo>
  <AESKeyValue>
    <Key>QNE80ZVYrrvFxgTgZh5bCg==</Key>
    <IV>dqLJ2iUvaEYcv5yfbU1cww==</IV>
  </AESKeyValue>
  <HMACSHAKeyValue>KGj3fWKDUjzcx/1nh/
```

Figure 4: Encrypting Message

When the user wants to download the message from cloud, he must now the private key. The uploaded message will be available for the user when he provides the private key. Figure 5 shows the result of decryption



Encrypted file path:

```
C:\Users\Avdhut\Desktop\Amma_Naanu_Devarane-Dr_HS_Venkateshmurthy
```

Output:

```
Decrypting...
Decryption is done, file is saved at C:\Users\Avdhut\Desktop\Amma_Naanu_Devarane-
Dr_HS_Venkateshmurthy[www.Mp3MaD.Com].decrypted
```

Figure 5: Decrypting Message

6. Conclusion

In this paper, we have addressed trust issues in cryptographic role-based access control systems for securing data storage in a cloud environment. The paper has proposed trust model for owners in RBAC systems which are using cryptographic RBAC schemes to secure stored data. The trust model can be integrated into the RBAC environment which will help the owners whether to trust particular role in the system or not. Finally we have also shown the results for generating the key pair, encrypting the message and decrypting the message by using the AES algorithm.

7. Acknowledgement

I thank to my respected guides Prof. Ubale Swapnaja A. and Prof.Pingale Subhash V. for their valuable guidance. I like to express deepest appreciation towards Dr. K. J. Karande,Principal, SKN SINHGAD COLLEGE OF ENGINEERING, and Prof.Abhijit A.Rajguru (Head. Dept. of C.S.E)whose invaluable guidance supported us in completing this paper.

References

[1] [1] D. F. Ferraiolo and D. R. Kuhn, ”Role-based access controls”, in 15th National Computer Security Conference, vol. 1-2. National Institute of Standards and Technology, National Computer Security Center, 1992, pp. 554 – 563.

[2] Avdhut Suryakant Bhise and Phursule R.N. Article: “A Review of Role based Encryption System for Secure Cloud Storage.”. International Journal of Computer Applications 109(14):15-20, January 2015.

[3] S. G. Akl and P. D. Taylor, “Cryptographic solution to a problem of access control in a hierarchy”, ACM Trans. Comput. Syst., vol. 1, no. 3, pp. 239–248, 1983.

[4] M. J. Atallah, K. B. Frikken, and M. Blanton, “Dynamic and efficient key management for access hierarchies”, in Proc. ACM Conf. Comput.Commun. Sec., Nov. 2005, pp. 190–202.

[5] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data”, in Proc. ACM Conf. Comput. Commun. Sec., Oct./Nov. 2006, pp. 89–98.

[6] A. Sahai and B. Waters, “Fuzzy identity-based encryption”, in Proc. EUROCRYPT, 2005, pp. 457–473.

[7] H. R. Hassen, A. Bouabdallah, H. Bettahar, and Y. Challal, “Key management for content access control in a hierarchy”, Comput. Netw.,vol. 51, no. 11, pp. 3197–3219, 2007.

[8] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P.Samarati, “A data outsourcing architecture combining cryptography and access control”, in CSAW , 2007, pp. 63–69.

[9] Y. Zhu, H. Hu, G.-J. Ahn, H. Wang, and S.-B. Wang, “Provably secure role-based encryption with revocation mech-anism”, J. Comput. Sci. Technol., vol. 26, no. 4, pp. 697–710, 2011.

[10] S. D. C. D. Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P.Samarati, “Over-encryption: Management of access control evolution on outsourced data”, in VLDB. ACM, 2007, pp. 123–134.

[11] Lan Zhou, Vijay Varadharajan, and Michael Hitchens, “AchievingSecure Role-Based Access Control on Encrypted Data in Cloud Storage”, IEEE transactions on information forensics and security, vol. 8, no. 12,pp. 1947-1960, December 2013.

[12] R. S. Sandhu, D. F. Ferraiolo, and D. R. Kuhn, “The nist model for role based access control: towards a unified

standard", in ACM Workshop on Role-Based Access Control, ser.RBAC00, 2000, pp. 47–63.

[13] ogdanov, Andrey, Dmitry Khovratovich, and Christian Rechberger, "Biclique Cryptanalysis of the Full AES", Microsoft Research (2011): n. pag. Biclique Cryptanalysis of the Full AE. Microsoft, 2011. Web. 7 Mar. 2013.

[14] Bokefode J.D, Ubale S. A, Apte Sulabha S,Modani D. G, Analysis of DAC MAC RBAC Access Control based Models for Security, International Journal of Computer Applications, Volume 104 – No.5, October 2014

[15] Stallings, William "The advanced encryption standard", Cryptologia 26.3 (2002): 165-188.