Developing Secure Cloud Storage System by Integrating Trust and Cryptographic Algorithms with Role based Access Control

Avdhut Suryakant Bhise
M.E. Student
Department of Computer Enginnering,
JSPM's ICOER, Wagholi, Pune University,
Maharashtra, India

R. N. Phursule
Asst. Professor
Department of Computer Engineering,
JSPM's ICOER, Wagholi, Pune University,
Maharashtra, India

ABSTRACT
Cloud computing is one of the rising and encouraging 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 the 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 work describe the way for modeling the security requirements from the view of tasks performed in an organization by using the cryptography concepts to store data on cloud with the less time and cost for process of encryption and decryption. In this work, the RSA and AES algorithms are used for encryption and decryption of data. The role based access control model is used to provide accessibility according to the role assigned to the user. This paper has the mathematical model for the trust calculation of the user. This system gives the rights for uploading to the user when he/she is authorized by the Administrator and Owner.

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

1. INTRODUCTION
The Cloud Computing provides three main services, Information as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (Saas) [1]. It reduces the cost of hardware required to store data that could have been used at user end. The cloud computing provides the number of advantages over the traditional computing and it includes: quickness, lower cost, scalability, device and location independency. Security in cloud computing is one of the most critical aspects due to importance and sensitivity of data stored on the cloud. But cloud computing has several major issues and concerns, such as data security, user access control, data integrity, trust and performances issues. For solving these problems, number of schemes is proposed under different systems and security models [2]-[6]. Whenever there is security of cloud computing, various security issues come up in a cloud. Some of the security problems and their solutions of them are reported below. Due to sharing computing resources with another company physical security is lost .User does not have knowledge and control of where the resources run and stored. It can be insured by using secure Data Transfer, maintaining the consistency or integrity of the data. Third, Privacy rights may be violated by cloud service providers and hackers. It can be ensured using cryptographic technique. Forth, when cryptographic technique is used then who will control the encryption/decryption keys? It can be ensured by giving permission to the users/customers. So implementing security in cloud computing is must which will break the difficulty of accepting the cloud by the organizations. There are varieties of security algorithms which can be implemented to the cloud. There are two types of algorithms symmetric key and asymmetric key [22], [15]-[18]. DES, Triple-DES, AES, and Blowfish etc are some symmetric algorithms can be used to implement cloud security [32]. RSA and Diffie-Hellman Key Exchange are the asymmetric algorithms these can be used to generate encryption and decryption key for symmetric algorithms. In cloud computing, symmetric key and asymmetric key algorithms is used to encrypt and decrypt the data [31]. In presented work, RSA algorithm is used to generate encryption and decryption keys for AES symmetric algorithm. Another major issue is how to manage user access to cloud storage system. For that different access control mechanism can be enforced for cloud users. Access Control is nothing but giving the authority to users to access the specific resources, applications and system [31]. There are three access control models, such as MAC (Mandatory access control model), DAC (Discretionary access control model) and RBAC Role based access control models [8]. These access control models specify the set of rules or criteria to access the system and its resources [7]. In MAC, The administrator has all the privileges to assign the user's roles according to his wishes. And end users does not have authority to change the access policies specified by the administrator therefore it is very restrictive and less used access control model. It can be used in a very perceptive environment. For example military, research centers [15]. In DAC, the end users have authority to change the access policy for any objects. But if an attacker gets control over the account it is too dangerous. So giving the complete authority to users is not good for any organization. In RBAC, first different roles or jobs can be specified and then these roles can be assigned to cloud user so these user can get access according to their jobs requirement. It is effectively and mostly used access control model within an organization because access to particular data and resources can be given according to the roles [7], [9]-[14]. In offered work, RBAC model is used for providing access control to the users.

2. LITERATURE SURVEY

Mainly there are two types of cryptographic algorithms used for encryption and decryption, such as Symmetric key algorithms, Asymmetric key algorithms and Combination key algorithms [33]. Encryption of data and its keys will make the secure cloud network and maintain the data privacy. Encryption is the process in which one can encode a message or data into unreadable format so intruder not able to read it. User has plain text when it is encoded into unreadable format using one of the encryption technique called as cipher text. After received by the correct receiver, he/she can decrypt it into the original plain text. Encryption is mostly used in network communications to achieve the data confidentiality.

2.1 Symmetric Encryption

This is one of the simplest encryption technique in which have only one secret key for encryption and decryption. There are number of symmetric key encryption algorithms in use which includes block ciphers like DES Blowfish, AES, Camellia, Serpent etc. and stream ciphers like FISH, Py,RC4, QUAD, SNOW etc[15]-[18].

2.2 Asymmetric Encryption

Asymmetric cryptography techniques known as public key cryptography, in which two separate keys, are used for encryption and decryption. Where one key is publicly available called as public key and it is used for encryption, and the other key is private key and it is used for the decryption.

Following techniques are the mostly used cryptographic techniques for cloud computing.

AES: In cryptography, the Advanced Encryption Standard (AES) is mostly used symmetric-key encryption standard. AES is a block cipher having block length of 128 bits block size, with key sizes of 128, 192 and 256 bits, respectively. AES allows for three different key lengths: 128, 192, or 256 bits. Encryption and Decryption for 128-bit keys needs 10 rounds of processing, for 192-bit keys needs 12 rounds of processing, and for 256-bit keys needs 14 rounds. All other rounds are identical for encryption and decryption; except for the last round in each case. It provides greater efficiency for software as well hardware also [20].

MD5: MD5 (Message-Digest algorithm 5) is a very famous and well known hash function and it generates a 128-bit resulting hash value. MD5 is commonly used in various applications to provide security, and it is also used to ensure the integrity of files. The MD5 value generated for specific file is considered as reliable fingerprint that can be used to check the integrity of the file contents. This algorithm had been implemented in different computer languages including C, Perl, and Java. In MD5 algorithm sender uses the public key provided by the receiver to encrypt the message and receiver uses its private key to decrypt the message [16].

DES: The DES (Data Encryption Standard) algorithm is one of the most used encryption algorithm in the world. DES was developed by IBM and it is symmetric block cipher.DES uses a 56-bit key for encrypting and decrypting a 64-bit block of data. The algorithm is more suitable to implement on hardware and not for software, because it gives low performance and it is time consuming [19].

RSA: RSA also called as public-key cryptography algorithm, it involves a public key and a private key. The public key can be used for encrypting messages and it is known to everyone. Messages can be decrypted using the private key. Data can be encrypted prior to storage, and establishes secure transmission channels [21].

3. PROPOSED SCHEME

3.1 Components of Architecture

Proposed scheme has the following four main entities. System Administrator is the authority who generates the username and password for the all users and secret key for the Role Manager, and to define the role hierarchy. Role Manager manages the user membership of a role. Owner is the person who has the authority to upload/store data securely in the cloud. Users will want to access and decrypt the stored data in the cloud.

3.1.1 Public Cloud:

It is a third party provider which resides outside the organizations and organizations outsource 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 [22]. An untrusted public cloud may disallow a user's request for accessing 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 [22]. Such behaviors can be detected, as a user can see the failure immediately after he/she 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.

3.1.2 Private Cloud:

It 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. It 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. 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 proficient user revocation, the private cloud is assumed to be truthful but curious in order to use the proposed scheme in this architecture. The cloud will execute the scheme and will not collaborate with revoked users.

3.1.3 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 [22]. This model is managed both by the third party entity and organization. It can be placed in the onsite or off site location.

3.1.4 Administrator:

The administrator is the main authority of the organization and secure cloud storage system. The administrator specifies the organization structure and creates the role hierarchy according to it.

3.1.5 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 enter the secret given by the administrator. 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. Prior a user is included into a role; the role manager has to authenticate the user in order to ensure that the user is qualified user.

3.1.6 Owner:

An owner can be a user within the organization who has the authority to encrypt and upload data in the cloud for other users to access; owners specify who can access the data according to the role-based policies. In the proposed model, owner manages the relationship between permissions and roles. Owner performs the encryption operation for that it does not require any secret key.

3.1.7 User:

These are the employees of the organization who has some job functionality according to their skills and need certain data from the public cloud to do this job functionality. Each and every user is authenticated by the administrator of the secure cloud storage system. Users are not having authority to update the organization structure. They are allowed only for downloading the data assigned for their roles.

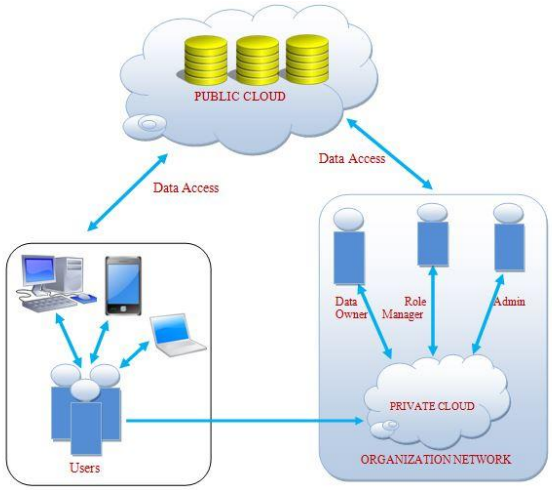

Fig 1. Secure Cloud Storage System

3.2 Experience Based Trust

It uses the past experience 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 and it is shown in figure1. 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. In proposed system, user does not have authority to upload data in the public cloud through system. When a user finishes a specified experience threshold value and got the recommendation from the Administrator and Data Owner he/s got the rights to upload data in a public cloud [22]. To receive recommendation from the administrator and data owner users past behavior and their transaction history will be considered. The received recommendations and his/her experience are uploaded in the central repository.

3.3 Working of SCSS

In this, first needs to create the user, assign roles to the users. This procedure contains following operations. In this work, Advanced Encryption Standard (AES) [20] algorithm used for encrypting and decrypting the data and RSA algorithm [21] is used to encrypt the secret key generated by the AES algorithm. When the roles in the system defined then for each role one public key and private key is created. This public key is used by the Data Owner to encrypt and upload the data in a public cloud and private key is used by the user to gain access for downloading data from the cloud. When administrator creates the role manager it will generate the secret key for that role and this secret key is used by Role Manager to assign role to users. When the user wants to decrypt the data he will first request for the cipher text from the public cloud [22]. As the decrypting values are stored in private cloud this request will be forwarded to the private cloud which will return the private key for decrypting the cipher texts. After confirmation the user can run the decryption algorithm to recuperate the data. User will get uploading rights when he/she finishes a specified experience threshold value and got the recommendation from the Administrator and Data Owner. To receive recommendation from the administrator and data owner users past behavior and their transaction history will be considered. The received recommendations and his/her experience are uploaded in the central repository. 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 thing is the Role behavior audit which keeps path of the feedbacks stored for particular role [22]. These feedbacks will be stored again in the central repository. Based on these conditions the trust decision engine takes decision whether or not the user will get rights for uploading or not [32]. The architecture of the trust management system is shown in figure1.

3.4 System Parameters

This system uses the AES for the purpose of encryption 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 public cloud we have taken instance from Microsoft azure and private cloud is created using the Linux with i5 processor and 8 GB ram. 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 significant 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 demonstrate that the size of the decryption key is 48 bytes, which is suitable for the users.

4. METHODOLOGY

In symmetric-key algorithm, the same secret key is used for both encryption and decryption, in contrast to asymmetric-key cryptography algorithm symmetric-key cryptography algorithm like AES (Advanced Encryption Standard) is high speed and also requires low RAM, but because of the same

secret key used for both encryption and decryption, it faces big problem of key transport from sender side to receiver side. But in asymmetric-key algorithm, it needs two different keys for encryption and decryption, one of which is private key and one of which is public key. The public key is used to encrypt plaintext; whereas the private key is used to decrypt cipher text [15]. As compared to symmetric-key algorithm, Asymmetric-key algorithm does not having problem while key exchanging and transporting key, but it is mathematically costly [15]-[18].

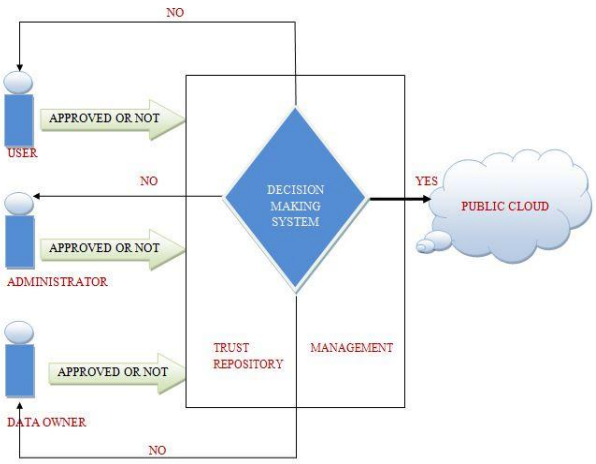

Fig 2. Trust Management Repository System

To solve the problem of key transport and get better performance these 2 algorithms can be combined together. In this data receiver generates the key pairs using asymmetric-key algorithm, and distributes the public key to sender. Sender uses symmetric-key algorithms to encrypt data, and then uses asymmetric-key algorithm to encrypt the secret key generated by the algorithms with the help of receiver's public key. Then receiver uses its private key to decrypt the secret key, and then decrypt data with the secret key. The asymmetric-key algorithm is used for encrypting the symmetric key, and it requires small computational cost. It similarly works like SSL. For encrypting the files or file data, AES (Advanced Encryption Standard) algorithm is used, and RSA (Rivest, Shamir and Adleman) is used to encrypt AES key [20], [21]. Description of AES and RSA algorithms is given below. AES starts with an Add round key stage followed by 9 rounds of 4 stages and a tenth round of 3 stages. The 4 stages are Shift rows, Substitute bytes, Add Round Key and Mix Columns. The 10^{th} round not performs the Mix Columns stage. These stages also apply for decryption. The nine rounds of the decryption algorithm consist of Inverse Shift rows, Inverse Substitute bytes, Inverse Add Round Key and Inverse Mix Columns. Again, the tenth round not performs the Inverse Mix Columns stage. For more details see [20].

RSA make use of measured exponential for encoding and decoding symmetric key that is secret key generated by the AES algorithm. Let us consider S as secret key and C as cipher key, then at encryption C=S mod n and at decryption S = C mod n. n is very large number which is generated during key generation process [21]. In proposed scheme, the administrator of the system defines different job functionalities required in a organization, then according to the needs of organization he add users or employees. The owner of the data encrypts the data in such a way that only the users with appropriate roles as specified by a RBAC policy can decrypt and view this data. The Role Manager will assign roles to users who are suitable for that role and he can also remove the users from assigned role. The cloud provider will not be able to see the contents of the data. A Role Manager is able to assign a role for particular user after the owner has encrypted the data or file for that role. A user assigned to particular role can be revoked at any time in which case, the revoked user will not have access rights to data or file uploaded for this role. Revocation of user from assigned role will not affect other users and roles in the system.

5. EXPERIMENT AND ANALYSIS

Various techniques used for applying Role based access control policies and encryption and decryption techniques to a Cloud storage system such as HKM, HIBE and ABE and RBE [15]. First approach is to concern the access control policies to transform the access control problem into a key management problem. Different approaches can be used to apply HKM schemes to enforce RBAC policies for data storage are discussed in [24]–[26]. But, these solutions have numerous limitations. For example, if the data owners and users are large then, it increase the overhead required to setting up the key infrastructure. Furthermore, when one of the user's access permission is deleted, then all the keys and public values known to this user need to be changed, which makes these schemes unfeasible.

Table 1. Comparison of Various Schemes with Proposed Scheme

Techniques	HKM	IBE	ABE	RBE	PROPO
Constant size	Yes	Yes	No	Yes	Yes
Constant size	Yes	No	No	Yes	Yes
Constant size	Yes	No	No	Yes	Yes
User	No	No	No	Yes	Yes
Role	No	No	No	No	Yes
Security	Low	Low	Low	Low	High

Table 2. Testing result of implemented scheme on different Processors

(Input Size) MB approximately	Intel P- 4 2.4 GHz	Intel Core2 Duo	Intel Core i3/i5/i7 & AMD
20-30	1.8	0.3	0.028
50-60	4.5	0.9	0.071
100-110	9.09	1.8	0.14
200-210	18.20	3.2	0.29
400-410	37	6.1	0.57
500-510	45.49	7.3	0.71
600-620	54.54	7.9	0.85
700-730	63.86	8.3	1
800-899	72.77	9	1.15
900-1000	81	9.8	1.29

One more scheme of keys management is Hierarchical ID-based Encryption (HIBE) [27], [28]. But in this technique the length of the user identity becomes longer when the depth of hierarchy increases. Another approach is role-based encryption scheme (RBE) in [29]. But, the user revocation in this scheme needs to update of all the role related parameters. Another approach was introduced in [30]. In this approach, the size of the cipher text increases linearly when role hierarchy increases. If single user belongs to different roles, then multiple keys need to be managed by this user. The implemented technique solves all these limitations. In this technique, for each and every role separate secret keys given to manage the user membership. Another scheme is ABAC; in this scheme access is given to the user depending on the attributes. In this techniques first attributes are defined as the access rights or rules, and to gain the access users have these attributes [31]-[34]. But, in this approach, the size of key is not constant and if one of these users can be revoked from the role hierarchy then the other user's keys are updated for the same role [32]. The result of implemented scheme is compared with all existing techniques shown in Table I. In this paper, the popular secret key algorithm that is AES with RSA is implemented, and their performance is calculated by encrypting different input files of varying contents and sizes. The algorithm is implemented in a uniform language (Asp.net), using their standard specifications, and tested on three different hardware platforms, to compare their performance. Result of comparison is shown in Table II.

6. CONCLUSION

The presented approach, allows an organization to upload data securely in a public cloud, while organizational information stored on a private cloud. The experience trust model was integrated into the SCSS. This helps administrator and owner to give uploading rights to users. The implemented approach provides great efficiency during encryption and decryption of the message. The implemented model also keeps the constant size of cipher text and decryption key. The experienced based trust has increased the security for the cloud. This proposed system is helpful for various organizations as it is implementing on role based access schemes which are much flexible for job functionality. It also provides secure storage of data using cryptographic techniques.

7. ACKNOWLEDGEMENTS

I specially thank to my friend Mr. Kale Babasaheb for constantly helping me throughout this work, his guidance for configuring the private cloud was much beneficiary. I also thank Onkar Mahajan for supporting me in this work.

8. REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, et al., "A view of cloud computing," Commun. ACM, vol. 53, no. 4, pp. 50–58, 2010.

[2] L. Zhou, V. Varadharajan, and M. Hitchens, "Enforcing role-based access control for secure data storage in the cloud," Comput. J., vol. 54, no. 13, pp. 1675–1687, Oct. 2011.

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

[4] 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.

[5] 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.

[6] D. Ferraiolo and R. Kuhn. Role-Based Access Controls. In Proceedings of the 15th NIST-NCSC National Computer Security Conference, pages 554–563, Bultimore, Maryland, USA, October 1992.

[7] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman. Role-Based Access Control Models. IEEE Computer,29(2):38–47, February 1996.

[8] D. Ferraiolo, R. Sandhu, S. Gavrila, R. Kuhn, and R. Chandramouli. Proposed NIST Standard for Role-Based Access Control. ACM Transactions on Information and Systems Security, 4(3):224–274,August 2001.

[9] G. Edjlali, A. Acharya, and V. Chaudhary. History-based Access Control for Mobile Code. In Proceedings of the 5th ACM Conference on Computer and Communication Security (CCS'98), pages 38–48, San Francisco, California, USA, November 1998.

[10] M. Abadi and C. Fournet. Access control based on execution history. In Proceedings of NDSSS'03, pages 107–121, 2003.

[11] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute-based Encryption for Fine-Grained Access Control of Encrypted Data," Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 89-98, 2006.

[12] R. Sandhu and X. Zhang. Peer-to-Peer Access Control Architecture Using Trusted Computing Technology. In Proceedings of the 10th ACM Symposium on Access Control Models and Technologies (SACMAT'05), pages 147–158, Stockholm, Sweeden, June 2005.

[13] J. Bethencourt, A. Sahai and B. Waters, "Cipher text-Policy Attribute-Based Encryption," Proceedings of the IEEE Symposium on Security and Privacy, pp. 321-334, 2007.

[14] K. Yang and X. Jia, "Attribute-based Access Control for Multi-Authority Systems in Cloud Storage," Proceedings of the 32nd IEEE International Conference on Distributed Computing Systems, pp. 536-545, 2012.

[15] Avdhut Suryakant Bhise and Phursule R.N., "A Review of Role based Encryption System for Secure Cloud Storage" In International Journal of Computer Application, Volume 109- No.14, January 2015.

[16] Bokefode Jayant D. and Ubale Swapnaja A., "Developing Secure Cloud Storage System by Applying AES and RSA Cryptography Algorithms with Role based Access Control Model", In International Journal of Computer Application, Volume 118-No. 12, May 2015.

[17] Singh, S preet, and Maini, Raman Comparison of Data Encryption Algorithms, International Journal of Computer science and Communication Vol. , No.1 January-June 2011, p.p. 125-127.

[18] Atul khate, Cryptography and Network Security, 2nd Ed, Tata Mcgraw hill, 2009, pp.87-2004.

[19] Davis, R., The Data Encryption Standard in Perspective,In Proc. of Communication Society magazine, IEEE, Volume 16 No 6, Nov. 1978, pp. 5-6.

[20] Daemen, J., and Rijmen, V. ,Rijndael: The Advanced Encryption Standard. Dr. Dobb's Journal, March 2001.

[21] R.L.Rivest, A.Shamir, and L.Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communication of the ACM, Volume 21 No. 2, Feb. 1978.

[22] Bokefode Jayant and Ubale Swapnaja, "Secure Cloud Storage System By Integrating Trust and Role Based Encryption Scheme", in IJECS Volume 4 Issue 5 May 2015.

[23] Prof. S. A. Ubale, Dr. S. S. Apte, Comparison of ACL Based Security Models for securing resources for Windows operating system,IJSHRE Volume 2 Issue 6, Page No 63.

[24] S. D. C. Di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, Over-encryption: Management of access control evolution on outsourced data, In Proc. VLDB, Sep. 2007, pp. 123–134.

[25] C. Blundo, S. Cimato, S. D. C. Di Vimercati, A. D. Santis, S. Foresti, S. Paraboschi, et al.,Efficient key management for enforcing access control in outsourced scenarios, In SEC (IFIP), vol. 297. New York, NY, USA: Springer-Verlag, May 2009, pp. 364–375.

[26] P. Samarati and S. D. C. di Vimercati, Data protection in outsourcing scenarios: Issues and directions," In Proc. ASIACCS, Apr. 2010, pp. 1–14.

[27] C. Gentry and A. Silverberg,, Hierarchical ID-based cryptography, in ASIACRYPT (Lecture Notes in Computer Science), vol. 2501. New York, NY, USA: Springer-Verlag, 2002, pp. 548–566.

[28] D. Boneh, X. Boyen, and E.-J. Goh, Hierarchical identity based encryption with constant size ciphertext, in EUROCRYPT (Lecture Notes in Computer Science), vol. 3494. New York, NY, USA: Springer-Verlag, May 2005, pp. 440–456.

[29] Y. Zhu, D. Ma, C. Hu, and D. Huang,,How to use attribute-based encryption to implement role-based access control in the cloud, In Proc. Int. Workshop Sec. Cloud Comput., 2013, pp. 33–40.

[30] Swapnaja A. Ubale, S. S. Apte, Bio-enable Security for Operating System by Customizing Gina, High Performance Architecture and Grid Computing Communications in Computer and Information Science Volume 169,2011, pp 179-185.

[31] Anjali D.V. and Dr. S.N. Chandrashekara, "Design and Implementation of Secure Cloud Storage System Using Hybrid Cryptography Algorithm With Role Based Access Control Model", in IJETR Volume – 5 Issue-1, May 2016.

[32] Rashmi Nigoti, Manoj Jhuria and Dr.Shailendra Singh, "A Survey Cryptographic Algorithm for Cloud Computing", in IJETCAS 13-123 2013.

[33] Ayushi, "A Symmetric Key Cryptographic Algorithm", in IJCA Volume 1-No. 15, 2010.