CIPHERTEXT-POLICY ATTRIBUTE-BASED ENCRYPTION FOR SECURE MOBILE APPLICATIONS IN CLOUD

S. ANGEL LEOLA SRUTHI[1], M.E., A.UMAMAGESWARI[2]
[1]ME Student, Associate Professor, [1,2] Department of Computer Science and Engineering,
[1,2]DMI College of Engineering, Chennai, India
[1]angelleolasruthi@gmail.com, [2]r.umaramesh@gmail.com

***Abstract*— The mainstay of this project is to design an effective cloud environment for preserving the privacy of mobile user data stored in the cloud. Portable devices such as smart phones or tablets that are significantly more limited than desktop computers in terms of memory, processors, useful operating life, and available network bandwidth so we are moving towards cloud. A mobile user may act as a data owner request key from the cloud manager, encrypts the data and then it uploads the data in the cloud. In the cloud re-encryption is done. Ciphertext-Policy Attribute Based Encryption (CP-ABE) approach is used for security and scalability. Regular auditing is done for integrity and availability.**

***Index Terms*— CP-ABE approach, cloud environment, cloud manager, cloud auditing.**

I. INTODUCTION

Mobile cloud computing at its simplest, refers to an infrastructure where both the data storage and data processing happen outside of the mobile device. Mobile cloud applications move the computing power and data storage away from the mobile devices and into powerful and centralized computing platforms located in clouds, which are then accessed over the wireless connection based on a client.

Today smartphones are employed with rich cloud services by integrating applications that consume web services. These web services are deployed in cloud. There are several Smartphone operating systems available such as Google's Android, Apple's iOS, RIM BlackBerry, Symbian, and Windows Mobile Phone. It must also be recognized that the recent trend in contemporary cloud computing applications is for cloud data to be accessed primarily by resource-constrained mobile devices, a practice known as mobile cloud computing. This device category includes users of smartphones and tablets; in some applications, it is appropriate to consider smart wireless sensors in the same mix. Hence, any protocol providing additional security must not add burdensome costs to a mobile user; specifically, the number of transmissions must be minimized to conserve the battery and over- the-air data usage fees, and the amount of computation must also be minimized to avoid adding significant delays to the user experience while further decreasing battery life. Another important requirement is for data to be addressable with fine-grained access controls on the record-level or finer, to provide flexibility. A single user log-in is largely insufficient in today's complex data retrieval tasks.

The following are the contributions made in the proposed work:

1. Storing data in a cloud environment is done in secure fashion. The cloud provider is unable to read stored data. The protocol is designed to be efficient for resource-constrained mobile users. An improvement is made over a traditional attribute based encryption scheme.
2. Additional security is provided through re-encryption in the cloud service provider. This additional security measure is an optional variant applicable to highly sensitive data subject to frequent access.
3. Re-encryption, as a process of transforming the stored ciphertext, it does not require removal of attributes and subsequent key regeneration, and may be administered by a trusted authority without involvement of the data owner.
4. The real-time performance of the proposed system is demonstrated on commercially popular mobile and cloud platforms.
5. Auditing is provided for integrity and availability of the data in the cloud.

In Section 2, literature survey on key management to secure cloud data storage is presented, with a focus on attribute-based schemes in the context of applications accessed by mobile devices. In Section 3, a system model encompassing a mobile cloud computing system is presented. In Section 4, the proposed algorithm for attribute-based encryption and re-encryption suitable for mobile users of the cloud is presented. In Section 5, the results of the implementation of proposed scheme on actual mobile devices and an operational cloud system are presented and discussed. Section 6, provides concluding remarks and the future enhancement.

II. LITERATURE SURVEY

Many solutions may be used to exchange encrypted data with a cloud provider in a secure manner, such that the cloud provider is not directly entrusted with key material, auditing is carried for ensuring availability of data.

The main disadvantage of a public key management system such as RSA [2] (which stands for the authors Rivest, Shamir, and Adleman) depends on the difficulty of factoring large integers. High traffic cost is another issue since the single key must be shared with all authorized users. Users may join and leave the authorized user set frequently, leading to constant key re-generation and redistribution through additional communication sessions to handle user revocation; in a highly scalable system, such events may occur at relatively high frequency. Wireless communication, is expensive and results in rapid battery drain, especially when transmitting [3].

Encrypted data should ideally be stored in the cloud so that it cannot be accessed by cloud provider. This notion is dependent on the keys being securely managed by an entity outside of the provider's domain. The difficulty arises when new users join the system, and existing ones leave, necessitating new keys to be generated. The encrypted data should ideally be transformed such that it may be unlocked with new keys, without an intermediate decryption step that would allow the cloud provider to read the plaintext; this process is known as data re-encryption. Although it appears to be a promising technique in managing encrypted data as access rights evolve over time, current solutions in the literature do not address the issue of high scalability to a sufficient and satisfactory degree; nor do they necessarily strive to lessen the computational and communication burden on users connecting to the cloud from resource- constrained mobile devices. The technique of ciphertext-policy attribute-based encryption (CP-ABE) [4] offers numerous advantages. It allows a user to obtain access to encrypted data in the cloud based on the possession of certain attributes that satisfy an access structure defined in the cloud, rather than the possession of a key that must be disseminated to all interested parties in advance. The requisite attributes may be determined by a data owner in advance; this owner is responsible for generating the user data to be shared, encrypting it, and uploading it to the cloud. The owner may not necessarily be required in every read transaction. Normally, a scheme based on CP-ABE relies upon the data owner granting access permission through an access tree, which requires his or her constant availability.

In some works, key material is distributed among multiple parties; for instance, a data owner and a trusted authorizer may function in concert to grant access permission to other users [5]; the solution, however, is not tailored for a mobile environment due to its computational demands, the required constant availability of the data owner, and time- based expiration of access leading to frequent key retrieval. Revocation of an authorized user is particularly hard to accomplish efficiently in CP-ABE and is usually addressed by extending attributes or keys with expiration dates. A tree of revocable attributes may need to be maintained and a trusted party assigned to validate revocation status. A mechanism using linear secret sharing and binary tree techniques is one example [6], but mobile users have to incur the communication cost of continually requesting new keys. Also, the data owner is typically also a mobile user, and thus cannot manage access control on demand for all due to its transient connectivity. Revocation has been proposed that relies on stateless key distribution and access control on the attribute level [7], but requires a trusted authority and encumbers the owner with a cryptographic pairing operation that is computationally costly.

Hierarchical Identity Based Encryption (HIBE) and CP-ABE are combined in another related work [20], to distribute user keys hierarchical domain masters are used; this is done at the cost of increased storage requirements for key material held by users and when generating ciphertext greater amount of processing is carried out. Progressive elliptic curve encryption scheme has been proposed for trusted data sharing [8]. However, it depends upon a writer uploading encrypted data to the cloud, then to perform re-encryption credentials are distributed to the cloud, and also to the reader on each data access attempt; it is impractical when applied to resource-constrained mobile devices and networks. A scheme has been proposed to handle revocation in a scalable system, which uses cloud provider for distributing portions of key material and for automatic re-encryption [9]; based on user identity it requires access solely this is one of the limitation in this work.

One method is to re-encrypt the stored content during retrieval. Such a technique has been applied to an encrypted file storage system where a content owner encrypts blocks of content with unique, symmetric content keys, and these keys are then further encrypted to form a lockbox [10]; users communicate with an access control server to decrypt them. The problem is that the content owner manages access control for all other users, which is a great burden, and requires dynamic re-encryption of the same data whenever multiple users access it. In the model herein, one-time re- encryption only occurs whenever membership changes, presumably a less frequent occurrence than that of data access. Other approaches require a trusted proxy for each decryption [11], which increases the communication cost. Proxy re-encryption has also been combined with CP- ABE [12] such that re-encryption keys are computed by the cloud provider based on a secret that is pre shared between the data owner and the provider, as well as the provider's internal clock. The re-encryption keys must be computed for all attributes in the access structure, which could be very numerous. Another idea is to securely embed the data key within the header of the record stored in the cloud [13]; a privileged manager group is responsible for generation of re-encryption keys, but it must also distribute the secret header key to the recipient to complete the process.

Attribute revocation approach has been suggested, and in response, an authority redefines master key components for the attributes, user secret keys are updated, and proxy server re-encrypts the data [14]; the revocation is dependent upon modifications to attributes this make it difficult, resulting in costly key updates on each revocation. A technique that combines CP-ABE with proxy re-encryption [15] does not appear to be highly efficient for mobile users: for instance, the decryption process requires processing two access sub trees instead of one.

ABE and proxy re-encryption are merged together in another related work, fine-grained access control of resources are allowed while offloading re-encryption activity to the cloud provider [16]. It has various differences to the scheme that will be proposed. The data owner is involved in generating a key for each new user that joins or leaves the system, rather than offloading this task; it is not only a prohibitive cost for a mobile user, but also impractical due to the user's mobility. Another difference is that a secret key must be regenerated and redistributed for each user, in lazy fashion, whenever user revocation occurs, rather than allowing users to upgrade a common group key, which reduces the communication cost and results in higher efficiency. Furthermore, the re-encryption occurs due to attribute redefinition and the scheme is based on key-policy attribute-based encryption (KP-ABE) and not CP-ABE, where the ciphertext is associated with a policy. A multi authority system has been proposed [17] that uses attribute-based access control to avoid the single point of failure of a single key authority, and relies upon the data owner communicating with attribute authorities to generate multiple encryption keys for hosted content.

The costs of communicating with multiple authorities and storing multiple keys could become prohibitive for mobile users. In a related work [18], each user submits multiple secret keys issued by authorities to a server to generate a decryption token for each ciphertext that is used in concert with the user's global secret key to perform a decryption. The same costs associated with multiple authorities apply, however, and such authorities complicate and add to the expense of system engineering. The authors are unaware of a similar scheme where fine- grained operations in attribute-based cryptography have been reassigned across system components to minimize the workload of mobile users; nor are other techniques found with performance benchmarked on commercial mobile and cloud systems, as herein, useful in assessing real-world viability.

The model is used for auditing the data for checking integrity of cloud [21] secure dynamic auditing method for data owner is used this helps to keep away from the security and integrity risk of data. New standards for publishing audit data is proposed [22] here various challenges and directions of auditing are discussed. Hence these are the various related works on key management and auditing.

III. SYSTEM MODEL

System model consist of the architecture diagram of the entire work. This explains clearly what the proposed work is.

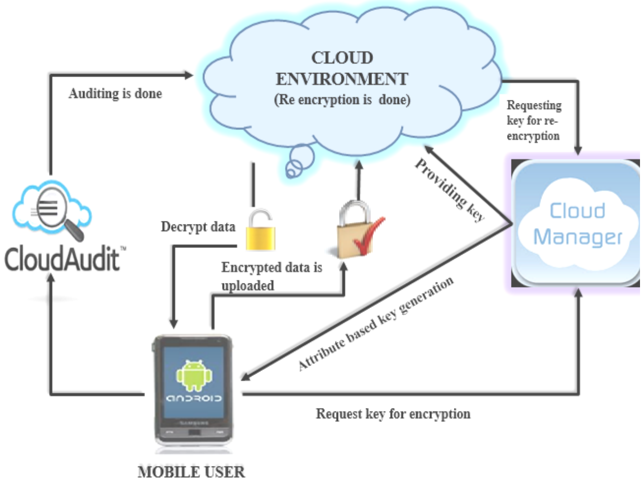


Figure 1. Architecture Diagram

Cloud environment

The public cloud is used here for storing the details of the mobile users. Public cloud allows system and services to be easily accessible to the general public. Public cloud may be less secure because of its openness. Re-encryption done in the cloud provides additional security to the data.

Cloud manager

Cloud manager provides key for encryption, re-encryption and decryption. Manager is a trusted party. The mobile users request keys for encryption, manager also generates keys for re-encryption which is performed in the cloud.

Mobile users

Mobile users may be Google's Android, RIM Blackberry, Apple iOS, etc., in this work android mobile users are selected. Android users encrypted their data and then upload their data to the cloud. They can also decrypt the data when they need it.

Cloud audit

Cloud solutions has to be auditable in order to enable continuous evaluation of whether the security level of a cloud supplier's specific solution is sufficient to be used for a given system or solution. At the same time, a cloud supplier must be able to provide adequate information about audit and assurance to meet the customers' risk assessment and be compliant with legislation. For auditing a protocol is been used.

IV. METHODOLOGY USED

The technique used is Cipertext-policy attribute based encryption were the encryption is done according to the attribute of the mobile user under this technique symmetric key encryption such as AES is used for encryption, re-encryption and decryption string matching algorithm is used for cloud auditing.

A. AES encryption:

AES encryption consist of four different stages one of permutation and three of substitution: substitute bytes, shift rows, mix columns and add round key are the various stages in the encryption. The same steps are carried for the re-encryption.

$$c = E_k(m)$$

Where, C is cipher text; E is encryption, m is the original message, E_k is key for encryption. This is how the encryption takes place.

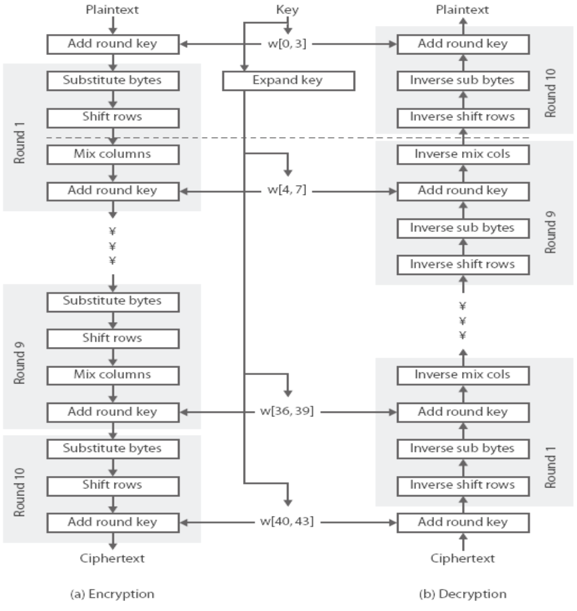


Figure 2. AES structure

The following algorithm shows the step by step process for encryption in AES:

Step 1: Start with the plain text
Step 2: Derive the set of round keys from the cipher key.
Step 3: Initialize the state array with the block data (plaintext).
Step 4: Add the initial round key to the starting state array.
Step 5: Perform various rounds of state manipulation.
Step 6: Perform the final round of state manipulation.
Step 7: Copy the final state array out as the encrypted data (cipher text).
Step 8: The cipher text is again given for the re-encryption process.
Step 9: The same steps are followed for the re-encryption process.

B. AES decryption:

AES decryption is the process of transforming information to make it readable to only those possessing special knowledge, usually referred to as a key decryption is not identical to encryption since steps done in reverse this has inverse shift rows, inverse sub bytes, and inverse mix columns and add round key.

$$D_k(c) = D_k(E_k(m)) = m$$

Where, D_k is the key for decryption, m is the original message, C is the cipher text. This is how the decryption process takes place.

The following algorithm shows the step by step process for decryption in AES:

Step 1: Start with the cipher text which has been re-encrypted in the cloud.
Step 2: Derive the set of round keys
Step 3: Round 1 is started which consist of inverse sub bytes, inverse shift rows and inverse mix columns and round key.
Step 4: Several rounds of operations are performed.
Step 5: Final round consist of inverse sub bytes, inverse shift rows and round keys alone
Step 6: The plain the final output from the decryption algorithm.

C. String Matching Algorithm for Auditing:

The context of the problem is to find out whether one string is contained in another string. This problem correspond to a part of more general one, called "pattern recognition". The strings considered are sequences of symbols, and symbols are defined by an alphabet. The size and other features of the alphabet are important factors in the design of string-processing algorithms. So this technique is used for auditing whether any other user is using their data.

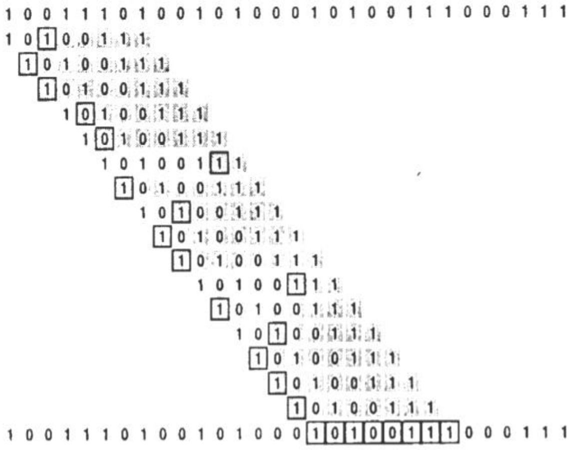


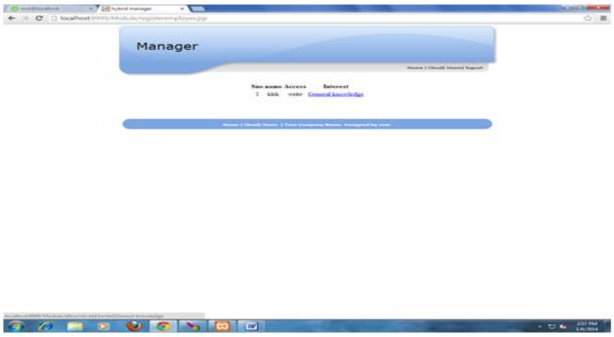
Figure 3. String Matching

The following algorithm shows the step by step process of string matching algorithm:

Step 1: Select the data to be audited
Step 2: The patterns will be searched
Step 3: Next counter will be given to move to the next string.
Step 4: This will be carried for the entire pattern of data.
Step 5: If there is any mismatch then the data is altered or any of vulnerabilities are available
Step 6: If the pattern are correct then the data are safe.

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

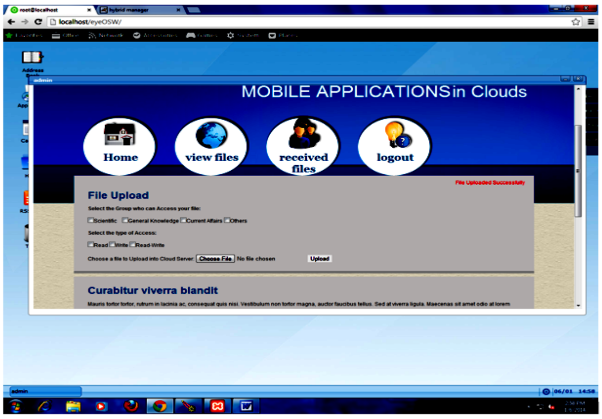
The proposed methodology has been done in public cloud using eyeOS software. Here the AES algorithm is used for encryption, re-encryption and decryption this ensures security of mobile users data stored in data. String matching algorithm is used for cloud auditing this is used for ensuring data integrity and availability. In the proposed methodology re-encryption provides high security.


Manager
Home | Cloud| Users| logout
Home | Cloud| Users | Your Company Name. Designed by vnn.

Sno	name	Access	Interest
1	kkk	write	General knowledge

Figure 4. Manager providing keys for encryption

Here the manager provides the key for encryption process, then the mobile users use this key for encrypting the data and then it uploads the data figure 5. Shows uploading the data in the cloud.



File Uploaded Successfully

File Upload

Select the Group who can Access your file:

□Scientific □General Knowledge □Current Affairs □Others

Select the type of Access:

□Read □Write □Read-Write

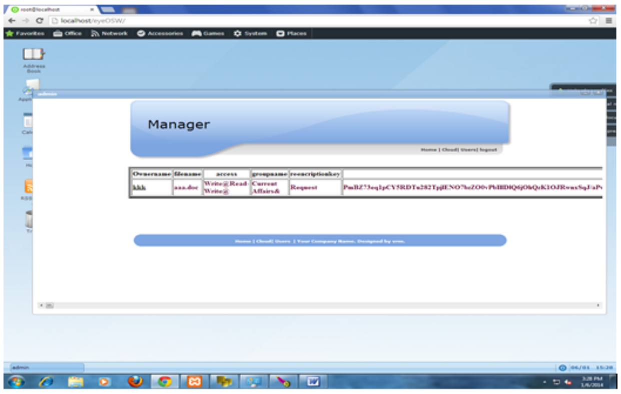
Choose a file to Upload into Cloud Server: Choose File No file chosen Upload

Curabitur viverra blandit

Mauris tortor tortor, rutrum in lacinia ac, consequat quis nisi. Vestibulum non tortor magna, auctor faucibus tellus. Sed at viverra ligula. Maecenas sit amet odio at lorem

Figure 5. Uploading file successfully

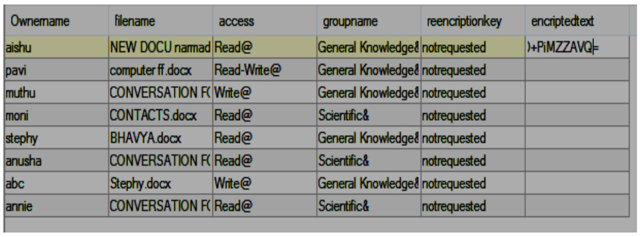
Here the re-encryption keys are provided by the cloud manager and re-encryption is done in the cloud environment itself. The manager provides key for decryption and the user decrypts the data from the cloud environment.


Manager
Home | Cloud| Users| logout
Home | Cloud| Users | Your Company Name. Designed by vnn.
admin

Ownername	filename	access	groupname	reencriptionkey	
kkk	aaa.doc	Write@Read-Write@	Current Affairs&	Request	PmBZ73eq1pCY5RDTu282TpjIENO7hzZO0vPhlIDlQ6jOhQzK1OJRwuxSqJ/aP

Figure 6. Re-encryption

Cloud auditing is done regularly for ensuring availability this is the novel modification in the proposed work.



Ownername	filename	access	groupname	reencriptionkey	encriptedtext
aishu	NEW DOCU narmad	Read@	General Knowledge&	notrequested	)+PiMZZAVQ=
pavi	computer ff.docx	Read-Write@	General Knowledge&	notrequested	
muthu	CONVERSATION F(	Write@	General Knowledge&	notrequested	
moni	CONTACTS.docx	Read@	Scientific&	notrequested	
stephy	BHAVYA.docx	Read@	General Knowledge&	notrequested	
anusha	CONVERSATION F(	Read@	Scientific&	notrequested	
abc	Stephy.docx	Write@	General Knowledge&	notrequested	
annie	CONVERSATION F(	Read@	Scientific&	notrequested	

Figure 7. Cloud Audit

VI. CONCLUSION

Secure mobile applications in cloud is implemented using ciphertext-policy attribute based encryption technique. Symmetric key encryption is used for encryption, re-encryption and decryption. Protocol for cloud audit is proposed for ensuring cloud integrity and availability. This work carried in the public cloud in future in it can be done in private or hybrid cloud.

REFERENCES

[1] P.K. Tysowski and M.A. Hasan, “Hybrid Attribute-Based Encryption and Re-Encryption for Scalable Mobile Applications in Clouds,” Technical Report 13, Centre for Applied Cryptographic Research (CACR), Univ. of Waterloo, 2013.

[2] R.L. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” Comm. ACM, vol. 26, no. 1, pp. 96-99, Jan. 1983.

[3] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, “Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications,” Proc. Ninth ACM SIGCOMM Conf. Internet Measurement Conf. (IMC ’09), pp. 280-293, 2009.

[4] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-Policy Attribute-Based Encryption,” Proc. IEEE Symp. Security and Privacy (SP ’07), pp. 321-334, 2007.

[5] A. Tassanaviboon and G. Gong, “OAuth and ABE Based Authorization in Semi-Trusted Cloud Computing: Aauth,” Proc. Second Int’l Workshop Data Intensive Computing in the Clouds (DataCloud-SC ’11), pp. 41-50, 2011.

[6] X. Liang, R. Lu, and X. Lin, “Ciphertext Policy Attribute Based Encryption with Efficient Revocation,” Technical Report BBCR, Univ. of Waterloo, 2011.

[7] J. Hur and D.K. Noh, “Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems,” IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 7, pp. 1214-1221, July 2011.

[8] G. Zhao, C. Rong, J. Li, F. Zhang, and Y. Tang, “Trusted Data Sharing over Untrusted Cloud Storage Providers,” Proc. IEEE Second Int’l Conf. Cloud Computing Technology and Science (CLOUDCOM ’10), pp. 97-103, 2010.

[9] P.K. Tysowski and M.A. Hasan, “Towards Secure Communication for Highly Scalable Mobile Applications in Cloud Computing Systems,” Technical Report 33, Centre for Applied Cryptographic Research (CACR), Univ. of Waterloo, 2011.

[10] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, “Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage,” ACM Trans. Information and System Security, vol. 9, pp. 1-30, Feb. 2006.

[11] S. Jahid, P. Mittal, and N. Borisov, “EASiER: Encryption-Based Access Control in Social Networks with Efficient Revocation,” Proc. Sixth ACM Symp. Information, Computer and Comm. Security (ASIACCS ’11), pp. 411-415, 2011.

[12] Q. Liu, G. Wang, and J. Wu, “Clock-Based Proxy Re-Encryption Scheme in Unreliable Clouds,” Proc. 41st Int’l Conf. Parallel Processing Workshops (ICPPW), pp. 304-305, Sept. 2012.

[13] J.-M. Do, Y.-J. Song, and N. Park, “Attribute Based Proxy Re- Encryption for Data Confidentiality in Cloud Computing Environments,” Proc. First ACIS/JNU Int’l Conf. Computers, Networks, Systems and Industrial Eng. (CNSI), pp. 248-251, May 2011.

[14] S. Yu, C. Wang, K. Ren, and W. Lou, “Attribute Based Data Sharing with Attribute Revocation,” Proc. Fifth ACM Symp. Information, Computer and Comm. Security (ASIACCS ’10), pp. 261-270, 2010.

[15] Y. Ming, L. Fan, H. Jing-Li, and W. Zhao-Li, “An Efficient Attribute Based Encryption Scheme with Revocation for Out- sourced Data Sharing Control,” Proc. First Int’l Conf. Instrumentation, Measurement, Computer, Comm. and Control, pp. 516-520, 2011.

[16] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing,” Proc. IEEE INFOCOM ’10, pp. 534-542, 2010.

[17] K. Yang and X. Jia, “Attributed-Based Access Control for Multi- Authority Systems in Cloud Storage,” Proc. IEEE 32nd Int’l Conf. Distributed Computing Systems (ICDCS), pp. 536-545, 2012.

[18] K. Yang, X. Jia, K. Ren, and B. Zhang, “DAC-MACS: Effective Data Access Control for Multi-Authority Cloud Storage Systems,” Proc. IEEE INFOCOM, pp. 2895-2903, 2013.

[19] J. Wang, “Java Realization for Ciphertext-Policy Attribute-Based Encryption,” http://github.com/wakemecn, 2012.

[20] G. Wang, Q. Liu, and J. Wu, “Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Storage Services,” Proc. 17th ACM Conf. Computer and Comm. Security (CCS ’10), pp. 735-737, 2010.

[21] S. V. Marshal, “Secure Audit service by using TPA for Data Integrity in Cloud System” International Journal of Innovative Technology and Exploring Engineering (IJITEE), September 2013.

[22] H. Rasheed, “Auditing for Standards Compliance in the Cloud: Challenges and Directions” The International Arab Journal of Information Technology, July 2003.