An Efficient and Secure Solution for Attribute Revocation Problem Utilizing CP-ABE Scheme in Mobile Cloud Computing

Vijay H. Kalmani
Research Scholar
Suresh Gyan Vihar University,
Jaipur, India

Dinesh Goyal, PhD
Research Supervisor and
Principal, Gyan Vihar School of
Engineering and Technology,
Suresh Gyan Vihar University,
Jaipur, India

Sanjay Singla, PhD
Research Supervisor
Suresh Gyan Vihar University,
Jaipur, India

ABSTRACT

With the advent of business apps which allow users to form dynamic groups so that they can store data on cloud servers and share the data within their user groups through their mobile devices. A major concern comes here that mobile users need the security of their group data which should not be accessible to other group users. To solve the issue, ABE or Attribute Based Encryption techniques are employed as they are vastly recognized as a valid and robust mechanism to provide fine access control over the data to legitimate users. At the same time, as there are complex computations involved in key issuing and data encryption by AAs' (Attribute Authorities) and decryption by legitimate users, there exist some efficiency issues. Rekeying plays a major role in dynamic systems where nodes come-in and move-out. As revocation of user rights requires the system to secure data from moved out users, rekeying has to be done on entire data set belonging to that attribute users in the group. However, the cost of re-keying is another concern for system efficiency which should not be compensated with a compromise on data security. There are many research works carried out earlier on data security for web applications using ABE, but there are limited studies on CP-ABE in mobile computing with multi-authority data storage system. A system is implemented which allows user groups to register, CAs'(Certificate Authorities) to allow registrations of Users and AAs and assign public Keys, AAs to manage attributes and revoke user access with re-keying and a centralized server for data persistence. Experimental results show the effectiveness of proposed solution and efficiency of re-keying mechanism while evoking user access rights on system architecture.

Keywords

Attribute-Based Encryption, CP-ABE, Mobile Data Security, Re-Keying, User Access Control

1. INTRODUCTION

Centralized data storage techniques are the vital element for global accessibility of user data at any moment of time. Cloud service providers offer such services to host the outsourced data of any user through the web or mobile device [1]. As mobile devices have certain restriction due to limited hardware capabilities, most of the computations are carried at hosting server end. These hosting services bring out new challenges for data access control due to the huge amount of data storage by numerous users. The cloud servers cannot be blindly trusted by data owners for security for their data, there have to be certain policies to make sure the data is protected from misuse. CP-ABE or Ciphertext-Policy Attribute-Based Encryption [2], [3] is considered as one among the most appropriate methodologies for access control of users' data in centralized storage servers, as it provides the data owner unrestricted but controlled access on data. In CP-ABE mechanism, there exist an entity which is accountable for attribute administration and key allocations i.e. Attribute Authority (AA). The AA can be any entity who acts as an admin of the system such as HR manager or group administrator of the network based game, etc. The visitors of the systems have to register in order to be a data owner, CA allots the public key and approves the registration and AA assigns attribute to the user as per the group allotment. Data gets encrypted through the attribute assigned by the AA. A user can decrypt the data only when they belong to the same attribute group else data will remain encrypted and secured on the server.

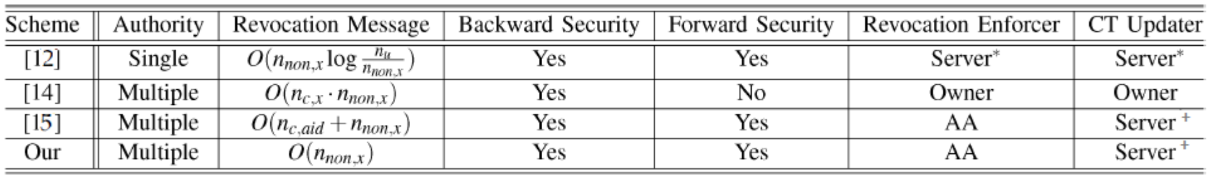
Till recent research works, there is evidence of only two types of ABE which are being proposed: CP-ABE (Ciphertext Policy-Attribute Based Encryption) and KP-ABE (Key-Policy Attribute Based Encryption). In KP-ABE, private keys are used for the access policies, whereas, in CP-ABE, it is provided in ciphertext [6].

Universally, there exist two kinds of CP-ABE mechanisms: single-authority CP-ABE [2], [3], [4], [5] in which attributes are controlled by a single AA and the second one is multi-authority CP-ABE [7], [8], [9] in which multiple attributes are administered by multiple authorities. Multi-authority CP-ABE is more suitable in case of cloud storage systems as data access has to be provided to huge no. of different users.

In case of multi-authority based data storage, Data Owners (DO) attributes can be altered dynamically. A new user can be allotted a new attribute by AA or any existing group user may lose the attribute to revoke their access rights. Though, existing attribute revocation schemes [10-15] depends too on a trustworthy server or are short of efficiency, those were not appropriate to tackle with the attribute revocation issues in data access management in multi-authority based data storage systems.

A novel revocable multi-authority CP-ABE mechanism is introduced in which a proficient and safe revocation scheme is projected to resolve the attribute revocation issues in the mobile cloud environment. As illustrated in below table, proposed attribute revocation scheme is proficient because it reduces delay in the process and computations. Also, it is safe because it provide a dual security mechanism. First, it restricts any revoked user to decrypt new encrypted data which requires a new attribute to decrypt and not the attribute which revoked user was having. Second, it allows new users to enjoy access to old data if a new user has been assigned the attribute which is required for decryption.

Brief Evaluation of Attribute Revocation Schemes for CP-ABE Architectures



Scheme	Authority	Revocation Message	Backward Security	Forward Security	Revocation Enforcer	CT Updater
[12]	Single	$O(n_{non,x} \log \frac{n_u}{n_{non,x}})$	Yes	Yes	Server*	Server*
[14]	Multiple	$O(n_{c,x} \cdot n_{non,x})$	Yes	No	Owner	Owner
[15]	Multiple	$O(n_{c,aid} + n_{non,x})$	Yes	Yes	AA	Server +
Our	Multiple	$O(n_{non,x})$	Yes	Yes	AA	Server +

*: Cloud Servers which are completely Trusted; + Partially Trusted Cloud Server

n_u represents the no.of users in the architecture; $n_{non,x}$ represents the no. of existing users who contain the revoked attribute x and $n_{c,x}$ is the no. of Ciphers that hold the attributes which are revoked x. $n_{c,aid}$ represents the cumulative no. of attributes belongs to AA_{aid} in entire ciphers

The proposed mechanism doesn't necessitate cloud server to be completely trusted as there is involvement of AAs for attribute assignment and validation of keys. This mechanism can assure the security of data and fine-grained access control due to its dual security mechanism and novel revocation scheme in multi-authority systems.

2. RELATED WORK

The idea of Attribute-Based Encryption pioneered by A. Sahai and B. Waters [16] as fuzzy identity-based encryption was initially brought by Pandey et al. [20] as the ABE. The distinguished parts in which ABE is bifurcated into are KP-ABE (Key-Policy Attribute Based Encryption) and CP-ABE (Ciphertext-Policy Attribute-Based Encryption) [20]. The structure of KP-ABE was presented in the same journal [20], as the primary CP-APE architecture referring tree-based organization in standard representation is demonstrated by Bethencourt et al. in 2007 [21]. Gradually, numerous other architectures referring to different types of access organizations were offered [22-24]. With regard to revocation of the attribute in Attribute-Based Encryption, an easy and simple revocation mechanism was introduced in [25] to attain scalable as well as fine-grained access control. To trim down the overload at the local machine, it is naturally expected to transfer steep computational jobs to remote machine. In fact, the issue that how to safely transfer diverse sorts of steep computations has drawn substantial considerations from hypothetical IT community. Pantazopoulos et al. [26] offered architecture to safely transfer of technical calculations such as matrix multiplication or quadrature. However, the proposition used the disguise mechanism and, therefore, resulted into the disclosure of confidential data. Atallah et al [27] look into the issue and offered an ingenious procedure to safely transfer progression assessment with dual servers. In addition, Benjamin et al. [28] dealt with the issue of safe outsourcing of extensively relevant linear arithmetic calculations. On the other hand, the suggested protocols requisite the steep procedures of homomorphic cipher conversion. Atallah et al. in [29] additionally researched on this issue and offered enhanced protocols based on the purported weak secret hiding supposition. Lately, Ren et al. [30] offered resourceful methods for safe transfer of linear program calculation.

Although a number of mechanisms have been brought in to safely outsource different types of steep computations, those were not appropriate for reducing ABE computational load of operations at the client device. To attain this objective, the conventional method is to make use of server-supported methods [31-33]. Nevertheless, prior researchers were aiming to increase the speed of exponentiation utilizing distrustful servers. Just making use of these methods in ABE would not have the desired efficiency. An additional way may be to enhance current universal outsourcing mechanisms or transferring computations [34-37] [19] depending on complete homomorphic ciphering or else interactive verification method. Though, Gentry [19] has presented that even for feeble safety factors on “bootstrapping” process of the homomorphic ciphering, it may require minimum thirty seconds on a very good resourceful system. Thus, yet the secrecy of data and operations can be safeguarded by those means but the computational load still remains a big issue to resolve.

Few of the studies conducted before in the same line were [17] and [18]. In [17], a revolutionary concept of outsourcing the deciphering of data in ABE was introduced whilst in [18] the researchers offered the Secrecy Safeguarding CP-ABE mechanism that permits to safely outsource ciphering and deciphering of the data to 3d party servers. In contrast with our study, both of the prior studies are not concentrating much on removing computation overhead on the user machine. We assume users to have mobile devices which lack efficiency compared to PCs and try to overcome computation load on user device issue by dividing the overall operations load evenly.

3. SYSTEM ARCHITECTURE

In the proposed architecture, we keep data access control in multi-authority based cloud data storage, as illustrated in Fig. 1. Totally there are five categories of entities in the scheme: Data Owners (DOs), Data Consumer (Group Users), a Certificate Authority (CA), the Attribute Authorities (AAs) and a centralized Mobile Cloud Server.

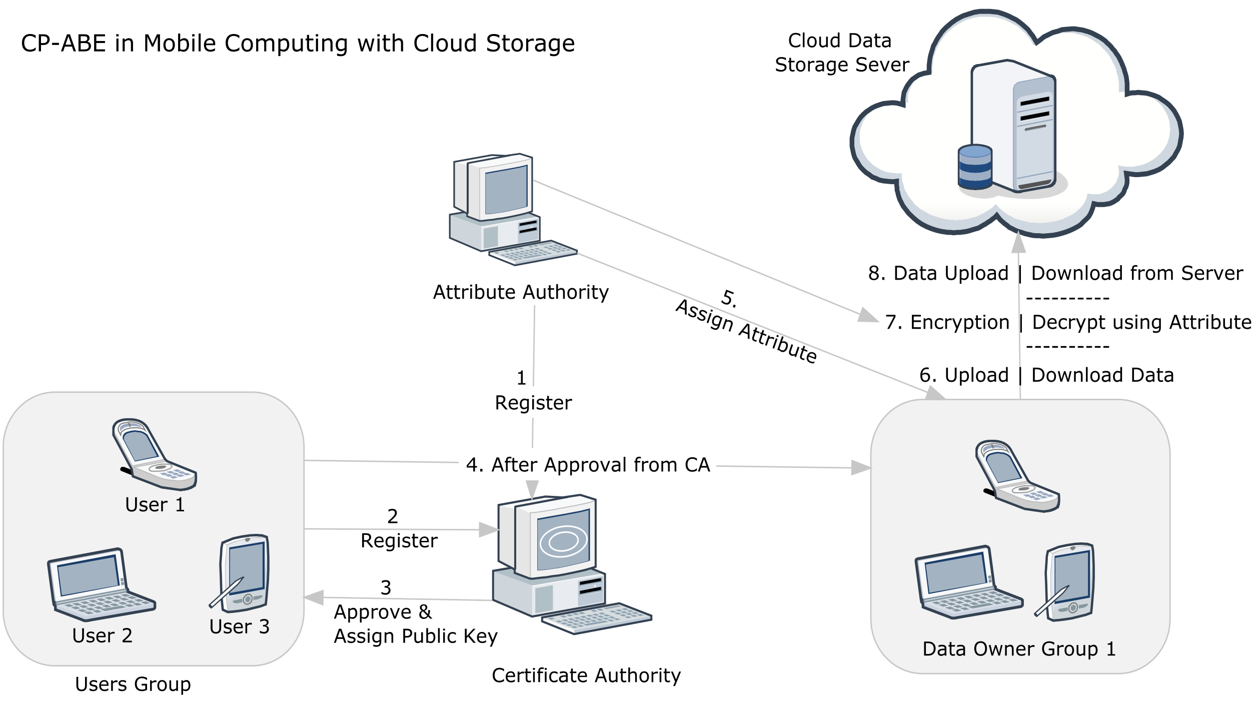

CP-ABE in Mobile Computing with Cloud Storage
Cloud Data Storage Sever
Attribute Authority
5. Assign Attribute
8. Data Upload | Download from Server
----
7. Encryption | Decrypt using Attribute
----
6. Upload | Download Data
1 Register
4. After Approval from CA
User 1
2 Register
3 Approve & Assign Public Key
User 2
User 3
Users Group
Certificate Authority
Data Owner Group 1

Figure 1: Architecture Diagram

It is assumed CA to be a genuine entity who distributes the authorization certificates to the members in this organization. It is the authority who makes approval for the new users and Attribute Authorities. Every genuine user in this scheme gets a distinguished registration ID and a PK (Public Key) is generated for them by the CA. Thus, the certificate authority is not responsible for generating or assigning any attribute to the users. Every Attribute Authority is an autonomous entity which takes care of generating attributes, assigning it to the users, revoking user's access controls and rekeying. In our system, the single attribute is linked with an AA, but every AA can administer any amount of attributes. Each AA has complete power over the arrangement and organization of its attributes, also it is liable to generate a secret attribute key for each attribute it generates so that the data can be encrypted using the same secret key as per the role and group of the data owner.

Every user of the system has a universal registration ID of them which distinguishes them from other users. One user may be eligible for multiple attributes which are assigned by multiple AAs. User gets a secret key connected with the attributes it owns assigned by respective AA. A user may login through their device and act as data owner to upload their content which can be accessed by their group mates. The content uploaded by data owners get encrypted using the secret key they own for their attribute. The access policy gets bounded to the data uploaded by the owners as it get encrypted and stored on the centralized cloud server. Consumers of the data must and should share the same attribute in order to decrypt and access the information.

4. PROPOSED SCHEME STAGES

Stage 1. System Setup: In this stage Certificate Authority and Attribute Authority setup is done using these functions:

- CAInit(1^{λ}) – (UMK, UPP, (UPK$_{uid,}$ UPK'$_{uid}$), (USK$_{uid}$, USK'$_{uid}$), Cert(uid)). The Certificate Authority initialization function is controlled fully by CA itself. It doesn't need anything apart from the inbuilt security token λ. It spawns a Universal Master Key UMK for the architecture and a Universal Public Parameter UPP. Every user is assigned with a distinguished Universal Public Keys (UPK$_{uid}$, UPK'$_{uid}$), Universal Secret Keys (USK$_{uid}$, USK'$_{uid}$) and Cert(uid) attached to the user.
- AAInit(G$_{aid}$) – (ASK$_{aid}$, APK$_{aid}$, {AVK$_{x_{aid}}$, APK$_{x_{aid}}$} $_{x_{aid} \in U_{aid}}$).

  The Attribute Authority (AA) initialization function is executed by every AA. It considers Global G$_{aid}$ administered by AA$_{aid}$ as an input parameter. It result in AA$_{aid}$'s Secret Key and AAs Public Key set i.e. (ASK$_{aid}$, APK$_{aid}$) with a pair of AA's Version and Public Attribute Keys i.e. {AVK$_{x_{aid}}$, APK$_{x_{aid}}$} $_{x_{aid} \in U_{aid}}$ for entire Attributes administered by AAs.

Stage 2. Attribute Authorities Generate Secret Key:

- SecKeyGen(UPP, UPK$_{uid}$, UPK'$_{uid,}$ USK$_{uid}$, ASK$_{aid}$, SetAt$_{uid,aid}$, {AVK$_{x_{aid}}$,APK$_{x_{aid}}$} $_{x_{aid} \in S_{uid,aid}}$) → ASK$_{uid,aid}$. Every Attribute Authority executes the functions to generate Secret Keys. Generation of the ASK$_{uid,aid}$ which is used for data decryption require AAs to receive Universal Public Parameter (UPP), Universal Public keys (UPK$_{uid}$, UPK'$_{uid}$) and a user's Universal Secret Key (USK$_{uid}$), Attribute Authority's (AA$_{aid}$) Secret Key ASK$_{aid}$, attribute set SetAt$_{uid,aid}$ which represents the user ID and attribute ID and matching AA's Version AVK$_{x_{aid}}$ and Public Attribute keys APK$_{x_{aid}}$ as an input.

Stage 3. Encryption of Outsourced Data: Initially the data *d* is converted into cipher by AES algorithm and then the content keys get encrypted using the following function:

- EncData(UPP,$\{APK_{aid_k}\}$ $_{aid_k \in I_A}$, $CK, \mathbb{AP}$) → ECData. Data encryption function is executed by Data Owners to convert content keys to cipher. The parameters considered while generating encrypted data ECData is Universal Public Parameters (UPP), a pair of public keys $\{APK_{aid_k}\}$ $_{aid_k \in I_A}$ for entire Attribute Authorities in the cipher set $I_{A,}$ the content key CK with an access policy $\mathbb{AP}$. It is presumed that Encrypted Data internally had $\mathbb{AP}$.

Stage 4. Decryption of Server Data: User initially executes decryption function to achieve the CK and utilize it for decryption of the data.

- DecData (ECData, UPK$_{\text{uid}}$, USK'$_{\text{uid}}$, $\{$ ASK$uid, aid_k\}$ $_{aid_k \in I_A}$)→ CK.

  Users execute the decryption function to decrypt the encrypted Data. It collects the ECData which has inbuilt $\mathbb{AP}$, UPK$_{\text{uid,}}$ USK'$_{\text{uid,}}$ and a pair of secret keys $\{$ ASK$uid, aid_k\}$ $_{aid_k \in I_A}$through each Attribute Authorities. The cipher is decrypted and content Key CK is returned on the condition that if the user's attribute $\{$SetAt$uid, aid_k\}$ $_{aid_k \in I_A}$matches with the$\mathbb{AP}$.

Stage 5. Revoking User Access: This stage involves 3 phases namely Generating Modified Key by Attribute Authorities, Updating Modified Key for Other Users in the Group and Re-Encryption of Data.

- MKeyGen (ASK$_{\text{aid'}}$, $\tilde{R}_{aid'}$, $VKey_{\tilde{x}_{aid}}$) → ($\widetilde{VKey}_{\tilde{R}_{aid'}}$$MK_{8,\tilde{R}_{aid'}}$, $MK_{c,\tilde{R}_{aid'}}$).

  Associated Attribute Authorities who administers revoked attribute $\tilde{R}_{aid'}$ executes the Modify Key generation function. This function takes input of parameters such as AAs Secret Key ASK$_{\text{aid'}}$, revoked attribute $\tilde{R}_{aid'}$ and its present Version Key i.e. $VKey_{\tilde{x}_{aid}}$. It generates a fresh $\widetilde{VKey}_{\tilde{R}_{aid'}}$, Modified Key $MK_{8,\tilde{R}_{aid'}}$and , $MK_{c,\tilde{R}_{aid'}}$ for updation of secret key and ciphers.

- ASKUpdate (ASK$_{\text{uid,aid'}}$, $MK_{8,\tilde{R}_{aid}}$)→ $\widetilde{ASK}_{uid,aid'}$.

  The security key associated with the old attribute ASK$_{\text{uid,aid'}}$ is exchanged with the security key of new attribute $\widetilde{ASK}_{uid,aid'}$ by validating modified key $MK_{8,\tilde{R}_{aid}}$, for all users who remain in the group. Thus, the revoked user doesn't get any chance to decrypt previous data.

- ECDataUpdate (ECData, , $MK_{c,\tilde{R}_{aid'}}$) → $\widetilde{ECData}$.

  The task to re-encrypt the data is done within the storage server. It processes the existing encrypted data with the revoked attribute $\tilde{R}_{aid'}$ to bring data to original state and again process the plaintext data with $MK_{c,\tilde{R}_{aid'}}$ to generate new encrypted data with a modified key.

5. EXPERIMENTAL RESULTS

5.1 Experiment Setup

To conduct the experiment two systems have been developed. First, an android based app for user registration, sign-in, uploading and downloading the data. Second, we developed an online application which has multiple modules such as Certificate Authority (CA), Attribute Authority (AA) and centralized cloud storage server. CA and AAs need to login in order to operate the system, AAs and Users may send a registration request to CA for approval and allotment of distinguished ID and public key. Until and unless AA do not allot attribute to the user, the user won't be able to login to the system. AAs may create any number of attribute and have the power to revoke or change any attribute for any particular user or whole group. For encryption and decryption, AES algorithm is used as it is proved to be of higher security trust. Android app uses HTTP web-service for server interaction and JSON objects are used for data binding and transmission over the server to the app.

5.2 Results

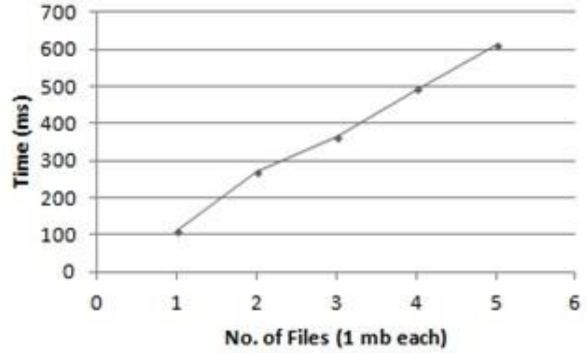


Figure 2: Registration ID & PK Generation Time

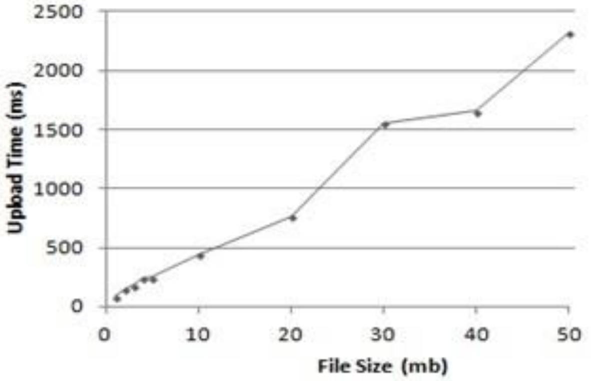


Figure 3: File Uploading Time

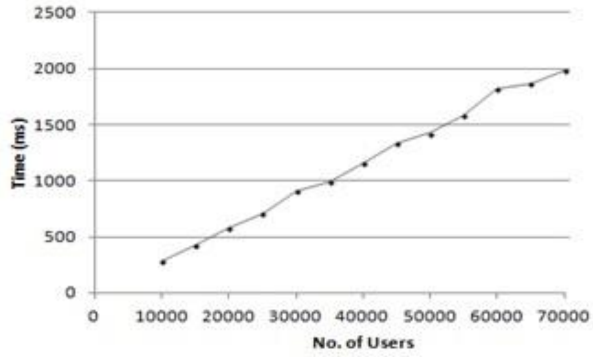


Figure 4: Rekeying Time

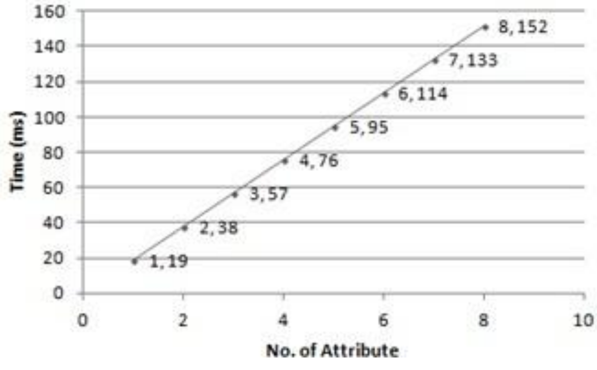


Figure 5: Attribute Assignment Time

These results are generated for a real-time system developed using J2EE and android programming. The delays are calculated programmatically and are subject to vary as the device performance to minor proportion.

Above results demonstrate various scenarios where the delay is measured. Fig. 2 shows the time taken for generation of registration ID and a public key for different no. of users simultaneously. Fig. 3 shows the uploading time of files of different sizes. This measure is done to estimate the delay in upload, encryption, and db update process. Fig 4 shows the rekeying time for different no. of files stored on the server. As revocation involves keying of files with a new attribute so no. of files get decrypt and encrypt costs little delay on server operation. At last Fig. 5 shows the attribute assignment time for users. More no. of attributes assigned to a user will have more time delay in proportion as only one attribute can be assigned at a time to the user. These results prove the efficiency of the system as each type of operation take very little time which is calculated in milliseconds

6. CONCLUSION

This research mainly concentrates on providing scalable and well-organized application solution which follows CP-ABE scheme on mobile cloud computing. The proposed mechanism provides a secure and efficient solution for attribute revocation problem and offers fine-grained access to legitimate users. Our system supports multi-authority CP-ABE scheme where multiple attribute authorities may exist to provide users more accessibility and flexibility to use the system. The advantage here is, the computation cost is marginally decreased due to the distribution of workload among multiple entities. It is safe to outsource data on distrustful servers as well due to better access control and security. This proposed mechanism can be applied on various online social groups or business apps which allow users to form groups and share the uploaded data within the group.

7. REFERENCES

[1] P. Mell and T. Grance, “The NIST Definition of Cloud Computing”, National Institute of Standards and Technology, Gaithersburg, MD, USA, Tech. Rep., 2009.

[2] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-Policy Attribute-Based Encryption”, in Proc. IEEE Symp., Security and privacy (S&P’07), 2007, pp. 321-334.

[3] B. Waters, ‘‘Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization,’’ in Proc. 4th Int’l Conf. Practice and Theory in Public Key Cryptography (PKC’11), 2011, pp. 53-70.

[4] V. Goyal, A. Jain, O. Pandey, and A. Sahai, ‘‘Bounded Ciphertext-Policy Attribute-Based Encryption," in Proc. 35th Int'l Colloquium on Automata, Languages, and Programming (ICALP'08), 2008, pp. 579-591.

[5] A.B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B.Waters, ‘‘Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption,’’ in Proc. Advances in Cryptology-EUROCRYPT’10, 2010, pp. 62-91.

[6] Jin Li 0002, Jingwei Li, Xiaofeng Chen, Xinyi Huang and Yang Xiang, "Securely Outsourcing Attribute-based Encryption with Checkability", in IEEE Trans. Parallel Distributed System, 25(8):2201-2210, 2014.

[7] M. Chase, ‘‘Multi-Authority Attribute-Based Encryption," in Proc. 4th Theory of Cryptography Conf. Theory of Cryptography (TCC'07), 2007, pp. 515-534.

[8] M. Chase and S.S.M. Chow, ‘‘Improving Privacy and Security in Multi-Authority Attribute-Based Encryption,’’ in Proc. 16th ACM Conf. Computer and Comm. Security (CCS’09), 2009, pp. 121-130.

[9] A.B. Lewko and B. Waters, ‘‘Decentralizing Attribute-Based Encryption,’’ in Proc. Advances in Cryptology-EUROCRYPT’11, 2011, pp. 568-588.

[10] S. Yu, C. Wang, K. Ren, and W. Lou, ‘‘Attribute Based Data Sharing with Attribute Revocation,’’ in Proc. 5th ACM Symp. Information, Computer, and Comm. Security (ASIACCS'10), 2010, pp. 261-270.

[11] M. Li, S. Yu, Y. Zheng, K. Ren, andW. Lou, ‘‘Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption,’’ IEEE Trans. Parallel Distributed Systems, vol. 24, no. 1, pp. 131-143, Jan. 2013.

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

[13] S. Jahid, P. Mittal, and N. Borisov, ‘‘Easier: Encryption-Based Access Control in Social Networks with Efficient Revocation,’’ in Proc. 6th ACM Symp. Information, Computer, and Comm. Security (ASIACCS'11), 2011, pp. 411-415.

[14] S. Raj, A. Nayak, and I. Stojmenovic, ‘‘DACC: Distributed Access Control in Clouds,’’ in Proc. 10th IEEE Int’l Conf. TrustCom, 2011, pp. 91-98.

[15] K. Yang and X. Jia, ‘‘Attribute-Based Access Control for Multi-Authority Systems in Cloud Storage," in Proc. 32th IEEE Int'l Conf. Distributed Computing Systems (ICDCS'12), 2012, pp. 1-10.

[16] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in Advances in Cryptology - EUROCRYPT 2005, ser. Lecture Notes in Computer Science, R. Cramer, Ed. Springer Berlin / Heidelberg, 2005, vol. 3494, pp. 457–473.

[17] M. Green, S. Hohenberger, and B. Waters, “Outsourcing the decryption of ABE ciphertexts,” in Proceedings of the 20th USENIX conference on Security, ser. SEC’11. Berkeley, CA, USA: USENIX Association, 2011, pp. 34–34.

[18] Z. Zhou and D. Huang, “Efficient and secure data storage operations for mobile cloud computing,” Cryptology ePrint Archive, Report 2011/185, 2011.

[19] C. Gentry and S. Halevi, “Implementing gentry’s fully-homomorphic encryption scheme,” in Advances in Cryptology – EUROCRYPT 2011, ser. Lecture Notes in Computer Science, K. Paterson, Ed. Springer Berlin / Heidelberg, 2011, vol. 6632, pp. 129–148.

[20] O. Pandey, V. Goyal, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in Proceedings of the 13th ACM conference on Computer and communications security, 2006, pp. 89–98.

[21] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in IEEE Symposium on Security and Privacy 2007, may 2007, pp. 321–334.

[22] L. Cheung and C. Newport, “Provably secure ciphertext policy ABE,” in Proceedings of the 14th ACM conference on Computer and communications security, ser. CCS ’07, 2007, pp. 456–465.

[23] T. Nishide, K. Yoneyama, and K. Ohta, “Attribute-based encryption with partially hidden encryptor-specified access structures,” in Applied Cryptography and Network Security, ser. Lecture Notes in Computer Science, S. Bellovin, R. Gennaro, A. Keromytis, and M. Yung, Eds. Springer Berlin / Heidelberg, 2008, vol. 5037, pp. 111–129.

[24] K. Ren, J. Li, B. Zhu, and Z. Wan, “Privacy-aware attribute based encryption with user accountability,” in Information Security, ser. Lecture Notes in Computer Science, P. Samarati, M. Yung, F. Martinelli, and C. Ardagna, Eds. Springer Berlin / Heidelberg, 2009, vol. 5735, pp. 347–362.

[25] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving secure, scalable, and fine-grained data access control in cloud computing,” in Proceedings of the 29th conference on Information communications, ser. INFOCOM’10. Piscataway, NJ, USA: IEEE Press, 2010, pp. 534–542.

[26] K. Pantazopoulos, M. J. Atallah, J. R. Rice, and E. E. Spafford, “Secure outsourcing of scientific computations,” in Trends in Software Engineering, ser. Advances in Computers, M. V. Zelkowitz, Ed. Elsevier, 2002, vol. 54, pp. 215 – 272.

[27] M. J. Atallah and J. Li, “Secure outsourcing of sequence comparisons”, International Journal of Information Security, vol. 4, pp. 277– 287, 2005.

[28] D. Benjamin and M. J. Atallah, “Private and cheating-free outsourcing of algebraic computations,” in Proceedings of the 2008 Sixth Annual Conference on Privacy, Security and Trust, ser. PST ’08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 240–245.

[29] M. J. Atallah and K. B. Frikken, “Securely outsourcing linear algebra computations,” in Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ser. ASIACCS ’10. New York, NY, USA: ACM, 2010, pp. 48–59.

[30] K. Ren, C. Wang, and J. Wang, “Secure and practical outsourcing of linear programming in cloud computing,” in IEEE International Conference on Computer Communications (INFOCOM), 2011, pp. 820–828.

[31] K. Bicakci and N. Baykal, “Server assisted signatures revisited,” in Topics in Cryptology - CT-RSA 2004, ser. Lecture Notes in Computer Science, T. Okamoto, Ed. Springer Berlin / Heidelberg, 2004, vol. 2964, pp. 1991–1992.

[32] M. Jakobsson and S. Wetzel “Secure server-aided signature generation”, in Public Key Cryptography, 2001, pp. 383–401.

[33] S. Hohenberger and A. Lysyanskaya, “How to securely outsource cryptographic computations,” in Theory of Cryptography, ser. Lecture Notes in Computer Science, J. Kilian, Ed. Springer Berlin / Heidelberg, 2005, vol. 3378, pp. 264–282.

[34] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum, “Delegating computation: interactive proofs for muggles,” in Proceedings of the 40th annual ACM symposium on Theory of computing, ser. STOC ’08. New York, NY, USA: ACM, 2008, pp. 113–122.

[35] C. Gentry, “Fully homomorphic encryption using ideal lattices,” in Proceedings of the 41st annual ACM symposium on Theory of computing, ser. STOC ’09. New York, NY, USA: ACM, 2009, pp. 169–178.

[36] R. Gennaro, C. Gentry, and B. Parno, “Non-interactive verifiable computing: Outsourcing computation to untrusted workers,” in Advances in Cryptology - CRYPTO 2010, ser. Lecture Notes in Computer Science, T. Rabin, Ed. Springer Berlin / Heidelberg, 2010, vol. 6223, pp. 465–482.

[37] K.-M. Chung, Y. Kalai, F.-H. Liu, and R. Raz, “Memory delegation,” in Advances in Cryptology - CRYPTO 2011, ser. Lecture Notes in Computer Science, P. Rogaway, Ed. Springer Berlin / Heidelberg, 2011, vol. 6841, pp. 151–168.