# An Efficient Data Sharing Scheme in Mobile Cloud Computing using Attribute based Encryption

**Vidhya Vijayan[1], Shandry K K[2]**

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

**Abstract** – *The use of cloud storage reduces the burden of storage in the user devices. Due to the security issues in the cloud storage the client application needs to provide security to the outsourced cloud data. Now a days, mobile devices become popular and mobile devices are used to store/retrieve data to/from the cloud storage from anywhere at any time. Thus, cloud security issues become more and more severe. There are different studies conducted to solve cloud security problems. Due to the limited computational capabilities of mobile devices, most of the techniques are not applicable for the mobile cloud. Thus, a lightweight scheme is needed that can provide both confidentiality and integrity to the cloud data. Also, an efficient access privilege management is also needed.*

***Key Words***: **Mobile cloud computing, Encryption, Data sharing, data integrity, Confidentiality.**

## 1. INTRODUCTION

Cloud computing provides large computational capabilities and huge storage spaces for data outsourcing. It provides huge storage space. Data stored in the cloud by a data owner means it goes out from their control i.e., to the third party, the Cloud Service Provider (CSP). CSP is considered as a semi-trusted authority. So, it is important to provide security and confidentiality to the user's data. Nowadays, mobile devices become more and more popular. And mobile devices are used to store/retrieve data from the cloud. But mobile devices have limited computing power and resources. To compensate for this limitation, it is essential to use the resources provided by the CSP to store and retrieve the data.

Now, various mobile applications are widely used. Mobile Cloud Computing (MCC) provides rich computational resources to mobile users. The data stored in the cloud may be confidential. Thus, it is very important to provide security to the outsourced data. There are many types of research had done that addresses the security in cloud computing. And these researchers propose many techniques that may solve these issues to some extent. No techniques can solve these security issues completely.

Data confidentiality and data integrity are the main challenges to be considered when handling outsourced data. Most of the work that addresses the security of cloud data is based on cryptographic techniques. Cryptographic techniques can provide security to the outsourced data by encrypting the data before it is being outsourced. In these cryptographic techniques, the data owner needs to share a key to whom they want to share the encrypted data. This secret key is used to decrypt the data. Also for the simplicity, the data owner can divide the data users into groups and send the key to each group that they want to share the data. In these approaches, data management is also an issue that needs to be considered. Also, it needs to provide an efficient access control mechanism so that only authorized user can access the original data. This is a challenging issue.

The mobile cloud computing has characteristics i.e., distributed, mobility and centralized management is to overcome the limitations of storage capacity and computing power of mobile devices and reduces the performance of the implementation of complex computing. Due to the limitations of the various resources of the mobile terminal equipment, mobile cloud services are also facing more complex security and privacy protection problems [1]. Along with the data confidentiality, it is very important to consider the data integrity of the outsourced data. Also, it is important to provide efficient user privilege management functionalities so that the owner can grant/revoke data access privileges to each user. Also, it must need to handle the problem that arises during user revocation. All the above issues must need to handle carefully, during data outsourcing to the cloud storage.

## 2. RELATED WORK

Homomorphic encryption is a form of encryption which directly performs ciphertext operation, with operation results being automatically encrypted, which lets anyone manipulate what is encrypted, even without knowing the secret key [7]. Processing encrypted data homomorphically requires more computation than processing the data unencrypted. To get low overhead, [8] use the recent batch homomorphic evaluation techniques. Also, introduce some other optimizations that can speed up homomorphic evaluation. In [9], they describe two improvements to Gentry's fully homomorphic scheme based on ideal lattices and its analysis: they provide a more aggressive analysis of one of the hardness assumptions (related to the Sparse Subset Sum Problem) and introduce a probabilistic decryption algorithm that can be implemented with an algebraic circuit of low multiplicative degree. Combined together, these improvements lead to a faster fully homomorphic scheme. This technique is secure over untrusted cloud but this technique is more complex in real applications.

Fully homomorphic encryption includes two basic homomorphism types. They are the multiply homomorphic encryption algorithm and additively homomorphic encryption algorithm. Homomorphic encryption algorithm supports only addition homomorphism and multiplication homomorphism before 2009 [10]. Fully homomorphic encryption is to find an encryption algorithm, which can be any number of addition algorithm and multiplication algorithm in the encrypted data. But this cryptographic technique is too complex to implement in real applications.

Attribute-Based Encryption (ABE) is a cryptographic technique proposed by Sahai and Waters [2]. It is derived from Identity-Based Encryption (IBE) [3] and is particularly suitable for one-to-many data sharing scenarios in a distributed and open cloud environment. Attribute-based encryption is divided into two categories: one is the Ciphertext-Policy Attribute-Based Encryption (CP-ABE), in which the access control policy is embedded in the ciphertext, the other one is Key-Policy Attribute-Based Encryption (KP-ABE), in which the access control policy is embedded in the user's key attributes.

In KP-ABE, the key issuer decides who can access the encrypted data and the encryptor exerts no control over who has access to the encrypted data. And in CP-ABE, the encryptor decides who has access to the encrypted data. Thus, CP-ABE is widely used. And CP-ABE is collusion resistant. In CP-ABE, encrypted data can be kept confidential even if the storage server is untrusted. A user’s private key will be associated with an arbitrary number of attributes expressed as strings. When a data owner encrypts a message in our system, they specify an associated access structure over attributes. A user will only be able to decrypt a ciphertext if that users attributes pass through the ciphertext’s access structure. Access structures are described by a monotonic ’access tree’, where nodes of the access structure are composed of threshold gates and the leaves describe attributes. A ciphertext-policy attribute-based encryption scheme consists of four fundamental algorithms: Setup, Encrypt, KeyGen, and Decrypt. Also, CP-ABE is conceptually closer to traditional access control methods such as Role-Based Access Control (RBAC) [4].

In CP-ABE, the attribute keys cannot be reclaimed once they are distributed. Thus, when a data user’s attribute is revoked, ensure data privacy becomes a difficult problem [5]. Unlike the existing CP-ABE schemes, Obtrovsky et al. [6] propose a more advanced CP-ABE scheme that can support non-monotonic access structures.

All these proposals consume a large amount of storage and computational resources, which are not available for mobile devices. Thus Ruixuan, et al. [23] proposes a Lightweight Data sharing Scheme (LDSS). This scheme is suitable for mobile devices and can provide security to the outsourced data. In this scheme, computationally intensive operations are conducted on proxy servers. Also, this scheme introduces lazy re-encryption to reduce revocation overhead.

Provable data possession (PDP) is used to provide integrity to data, that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. G. Ateniese presents two provably-secure PDP schemes [11] that are more efficient than previous schemes. The model allows the server to access small portions of the file in generating the proof, no need to access the entire file. First provably-secure scheme (S-PDP) is for remote data checking. The second scheme is a more efficient version of this scheme (E-PDP) that proves data possession using a single modular exponentiation at the server. Both schemes use homomorphic verifiable tags. Because of the homomorphic property, tags computed for multiple file blocks can be combined into a single value. The client pre-computes tags for each block of a file and then stores the file and its tags with a server. At a later time, the client can verify that the server possesses the file by generating a random challenge against a randomly selected set of file blocks. Using the queried blocks and their corresponding tags, the server generates a proof of possession. The client is thus convinced of data possession, without actually having to retrieve file blocks.

The first (S-PDP) provides a strong data possession guarantee, while the second (E-PDP) achieves better efficiency at the cost of weakening the data possession guarantee. The schemes are based on the KEA1 assumption which was introduced by Damgard [12]. The more efficient variant of S-PDP is E-PDP that achieves better performance. The E-PDP scheme reduces the computation on both the server and the client to one exponentiation.

An algebraic signature-based cloud data integrity auditing scheme can satisfy the security properties of data confidentiality, privacy preserving and data integrity. Jian Shen, et al. [13] proposes algebraic signature-based data integrity auditing scheme that ensures the cloud data integrity and confidentiality with batch auditing. This auditing scheme has less computation overhead for the data integrity auditing. This scheme also supports data dynamics by using only one cloud server. Due to the characteristics of the algebraic signatures, this batch auditing can support single data-block auditing and multiple data-block auditing. A cloud data dynamics mechanism is also included based on matrix operations. This data dynamics can provide data deletion, insertion and update.

Algebraic signatures have homogeneity and algebraic properties, which can be used in remote data possession proofs [14] [15]. Algebraic signatures are defined in the finite field. Their algebraic property is such that the signature of some data blocks is equal to the sum of the corresponding data blocks' signatures [16] [17]. The algebraic signature is similar to the hash functions MD5 and SHA in cryptography. If MD5 and SHA are used for data integrity checking on a remote server, the user's side needs to retrieve all of his/her data, resulting in huge computation and communication overhead at the user's side. The algebraic signature enables the cloud to return a part of the data signatures for data integrity checking, which saves bandwidth in remote cloud server data integrity checking applications. This technique costs lower communication and computation overhead [14] [15] [18].

This is a secure and efficient PDP scheme called IBPSMPDP, based on the limitations of the computational and storage capabilities of mobile terminals. Yuan, et al. proposes identity-based proxy signature multiple-file data integrity verification scheme called Identity-Based Proxy Signature Multiple-File PDP (IBPSMPDP) [19] for mobile cloud computing. This is based on the traditional PDP, combines the advantages in [20] with [21] and [22]. This technique effectively reduce the cost of public key certificate management and storage, the combination of bilinear pairings makes the key and signature length of the scheme shorter and the security is based on the computational Diffie- Hellman (CDH) problem. It consists of 5 entities:

(1) Mobile User: Mobile users or data owners store their data on the cloud server through a variety of interfaces of mobile terminal devices to interact with the cloud, they can also add files to the cloud.

(2) Cloud Service Provider (CSP): Providing a variety of data storage and resource services for mobile users via the mobile Internet.

(3) Proxy Party: Proxy party can be mobile user's home computer or companies that have the strong computing power, providing computing service and data signing for mobile user (data owner), as well as reducing the computational pressure of the mobile user.

(4) Private Key Generation (PKG): To generate the system's key pairs according to the system security parameters, generate the public key pair of the user and the proxy on the basis of the identity information sent by them and so on.

(5) Verifier: Verify that the data stored on the cloud storage server is complete.

To check the integrity of the outsourced data, the proxy signature party with the higher computing power will generate the signature of the file being outsourced. During the integrity checking, the verifier sends a challenge to the cloud server which can generate the integrity evidence Proof. Then, the verifier determines the integrity of the data stored in the cloud based on the integrity evidence returned by the cloud storage server. But this technique is complex in practical applications.

A hashing algorithm is a cryptographic hash function. It is a mathematical algorithm that maps data of arbitrary size to a hash of a fixed size. It is designed to be a one-way function i.e., infeasible to invert. Some common hashing algorithms include MD5, SHA-1, SHA-2, NTLM, and LANMAN. In [24], uses MD5 hashing algorithm to provide integrity. It computes the hash function continuously to provide integrity. In [25], Secure Hashing Algorithm (SHA-2) is used to generate a message digest by passing the original message along with shared variable to the hash function. This is done by both the end- user and the auditor and the value obtained from the hash function is compared and hence the data integrity is verified.

In [26], Guoping Wang proposes an efficient implementation of SHA-1 hashing algorithm. The SHA-1 algorithm takes as an input message with a maximum length of less than 264 bits and produces a 160-bit message digest. The input is processed in 512-bit blocks. The algorithm processing includes the following steps: Padding, Appending Length, Appending Length, Processing message in 16-word Blocks, Final Output Processing. SHA-1 algorithm is collision resistant and one- way function. Also, different versions of SHA algorithm like, SHA1, SHA2, SHA, etc., can be used for integrity based on the application.

## 3. SYSTEM ARCHITECTURE

The proposed data sharing method has the following components:

(1) Data Owner (DO) – DO uploads the data to the mobile cloud. DO determine the access control policies.

(2) Data User (DU) – DU retrieves data from the mobile cloud.

(3) Trusted Authority (TA) – TA can grant/revoke access permission to users.

(4) Encryption Service Provider (ESP) - ESP encrypts the data for the DO.

(5) Decryption Service Provider (DSP) - DSP decrypts the data for the DU.

(6) Cloud Service Provider (CSP) – CSP stores the data and executes the operations requested by the DU.

(7) Integrity verifier – Verifies the integrity of the cloud data.

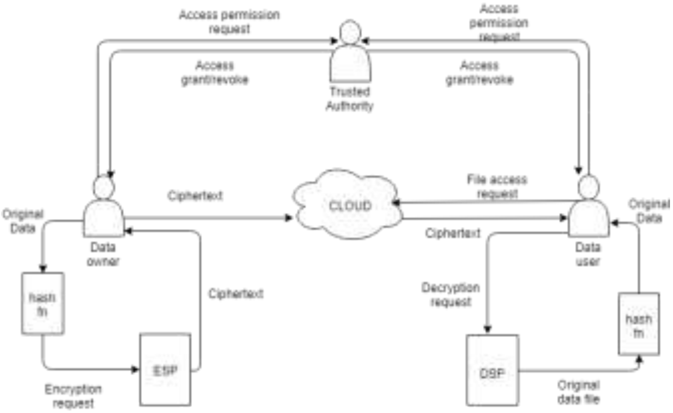

**Fig -1**: System architecture

The fig-1 shows the system architecture. The DO sends data to the cloud. Since the cloud is not trusted, data has to be encrypted before it is uploaded. DO define the access control policy. The data files are encrypted using the symmetric encryption scheme. And the symmetric key for the data encryption is also encrypted using Attribute-Based Encryption (ABE). In this scheme access control policy is embedded in the ciphertext. DU who obtains the key that satisfies the access control policy can decrypt the ciphertext.

The encryption and decryption are done by the ESP and DSP to reduce the computational overhead in the mobile devices. The key for the data encryption is generated by the Trusted Authority (TA). The TA is responsible for user management. In this system the users are categorized in different groups. And each data owner can determine who can access the data at the time of data outsourcing. Thus, TA need not be online at every data access. Thus, it can be used efficiently at any time.

The integrity verifier verifies the integrity of the cloud data by using hash function. The hash function produces hash value for every file at the time of data outsourcing and save this hash value in the database. At the time of data request, the integrity verifier recomputed the hash value and checks it with the previous hash value computed at the time of data outsourcing. If the both hash values are same the requested file is given to the data requester.

## 4. IMPLEMENTATION

Microsoft Visual Studio is used to implement the system. It is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Development framework used is ASP.NET with C# programming language. ASP.NET is a framework for developing dynamic web applications. Database used is SQL Server Management Studio. SQL Server Management Studio (SSMS) is one of the most important tools used with SQL Server. It provides a user-interface for common database tasks.

Secure file sharing cloud is used to store files. Symmetric encryption is used to secure data. Attribute based encryption is used to secure the encryption key. To reduce the overhead on the mobile devices, proxy servers are used. The encryption and decryption tasks are done in these proxy servers. Trusted authority is used to manage the users. SHA-1 hash function is used to provide integrity to the outsourced data.

## 5. RESULTS

The system provides security to the outsourced cloud data. The system uses AES encryption algorithm for data confidentiality and a data integrity verifier is also used to check the integrity of the cloud data. For the performance analysis, different sized encrypted files are uploaded into the cloud and calculate the time of uploading the encrypted file to the cloud. Also, the time of uploading the encrypted data into the cloud is also calculated for a different system i.e., Robust and auditable access control scheme (RAAC). It is also a system for secure data in the cloud storage. It also uses encryption and decryption.

Compare the proposed system (Mobile data sharing) with the RAAC by plotting a chart that shows the performance of both systems. The time analysis chart shows the time taken by the two systems to upload a file into the cloud against the file size. Chart-1 shows the time analysis. From the analysis chart, it is clear that the performance of the proposed system is better, when compared to the RAAC. The proposed system takes minimum time for data uploading than existing system.

This is because our proposed system implements encryption and decryption as a separate module. Thus, it reduces the storage burden at the client devices. And our system performs better and faster than previous system.

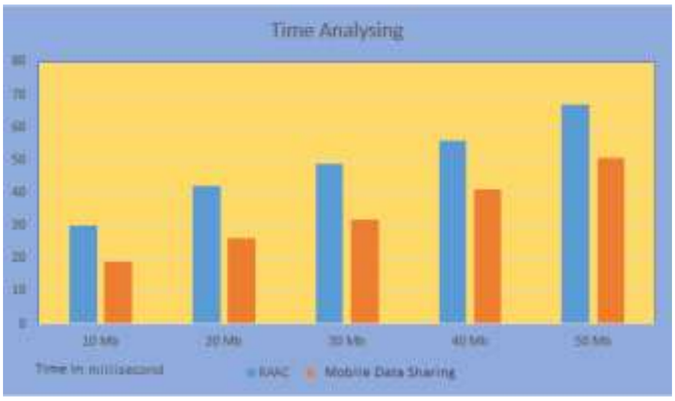

**Chart -1**: Time analysis

## 6. CONCLUSIONS

The main issues in the cloud are data confidentiality and integrity. The homomorphic encryption technique is more secure but it is not applicable for real applications. The traditional attribute-based encryption technique is secure, but it is not suitable for mobile applications. Thus a new lightweight scheme called LDSS based on ciphertext policy attribute based encryption (CP-ABE) can be used that can provide security to the cloud data and it is suitable for mobile devices.

In this project, a lightweight data outsourcing scheme is proposed for mobile cloud environment. An integrity verification mechanism based on SHA-1 hashing algorithm is also included. An efficient access privilege management mechanism is also included. This system can provide both confidentiality and security to the mobile devices. This is more secure scheme that is accessible to the mobile devices. And our system performs faster by reducing the storage burden in the client side.

## REFERENCES

[1] **ZHANG, X., and TU, P.**, 2016. "Application of storage and segmentation encoding technology in mobile cloud security". Journal of Computer Applications, 4, p. 012.

[2] **Sahai, A., and Waters, B.**, 2005. "Fuzzy identity-based encryption". In Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, pp. 457–473.

[3] **Boneh, D., and Franklin, M.**, 2001. "Identity-based encryption from the weil pairing". In Annual international cryptology conference, Springer, pp. 213–229.

[4] **Sandhu, R. S., Coyne, E. J., Feinstein, H. L., and Youman, C. E.**, 1996. "Role-based access control models". Computer, 29(2), pp. 38–47.

[5] **Lai, J., Deng, R. H., Li, Y., and Weng, J.**, 2014. "Fully secure key-policy attribute-based encryption with constant-size ciphertexts and fast decryption". In Proceedings of the 9th ACM symposium on Information, computer and communications security, ACM, pp. 239–248.

[6] **Ostrovsky, R., Sahai, A., and Waters, B.**, 2007. "Attribute-based encryption with non-monotonic access structures". In Proceedings of the 14th ACM conference on Computer and communications security, ACM, pp. 195–203.

[7] **Li, J., Song, D., Chen, S., and Lu, X.**, 2012. "A simple fully homomorphic encryption scheme available in cloud computing". In Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on, Vol. 1, IEEE, pp. 214–217.

[8] **Gentry, C., Halevi, S., and Smart, N. P.**, 2012. "Fully homomorphic encryption with polylog overhead". In Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, pp. 465–482.

[9] **Stehl´e, D., and Steinfeld, R.**, 2010. "Faster fully homomorphic encryption". In International Conference on the Theory and Application of Cryptology and Information Security, Springer, pp. 377–394.

[10] **Bedra, A.**, 2010. "Getting started with google app engine and clojure". IEEE Internet Computing, 14(4), pp. 85–88.

[11] **Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., and Song, D.**, 2007. "Provable data possession at untrusted stores". In Proceedings of the 14th ACM conference on Computer and communications security, ACM, pp. 598–609.

[12] **Damgard, I.**, 1991. "Towards practical public key systems secure against chosen ciphertext attacks". In Annual International Cryptology Conference, Springer, pp. 445–456.

[13] **Shen, J., Liu, D., He, D., Huang, X., and Xiang, Y.**, 2017. "Algebraic signatures-based data integrity auditing for efficient data dynamics in cloud

computing”. IEEE Transactions on Sustainable Computing.

[14] **Chen, L.**, 2011. “Using algebraic signatures for remote data possession checking”. In Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2011 International Conference on, IEEE, pp. 289–294.

[15] **Schwarz, T. S., and Miller, E. L.**, 2006. “Store, forget, and check: Using algebraic signatures to check remotely administered storage”. In null, IEEE, p. 12.

[16] **Shen, J., Liu, D., Shen, J., Liu, Q., and Sun, X.**, 2017. “A secure cloud-assisted urban data sharing framework for ubiquitous-cities”. Pervasive and mobile Computing, 41, pp. 219–230.

[17] **Mokadem, R., and Litwin, W.**, 2006. “String-matching and update through algebraic signatures in scalable distributed data structures”. In Database and Expert Systems Applications, 2006. DEXA’06. 17th International Workshop on, IEEE, pp. 708–711.

[18] **Zhu, Y., Ahn, G.-J., Hu, H., Yau, S. S., An, H. G., and Hu, C.-J.**, 2013. “Dynamic audit services for outsourced storages in clouds”. IEEE Transactions on Services Computing, 6(2), pp. 227–238.

[19] **Yuan, Y., and Xie, F.**, 2017. “Identity-based proxy signature multiple-file pdp for mobile cloud computing”. In Computational Science and Engineering (CSE) and Embedded and Ubiquitous Computing(EUC), 2017 IEEE International Conference on, Vol. 1, IEEE, pp. 381–387.

[20] **Yan, L., Shi, R., Zhong, H., Cui, J., Zhang, S., and Xu, Y.**, 2015. “Integrity checking protocol with identity based proxy signature in mobile cloud computing”. Journal on Communications, 36(10), pp. 278–286.

[21] **Xiao, D., Yang, L., Liu, C., Sun, B., and Zheng, S.**, 2015. “Efficient data possession auditing for real-world cloud storage environments”. IEICE TRANSACTIONS on Information and Systems, 98(4), pp. 796–806.

[22] **Xiao, D., Yang, Y., Yao, W., Wu, C., Liu, J., and Yang, Y.**, 2012. “Multiple-file remote data checking for cloud storage”. Computers & Security, 31(2), pp. 192–205.

[23] **Ruixuan Li,Chenglin Shen Heng He, Xiwu Gu, Zhiyong Xu and Cheng-Zhong Xu**., 2018. “A Lightweight Secure Data Sharing Scheme for Mobile Cloud Computing”. IEEE Transactions on Cloud Computing, 6(2), pp. 344-357.

[24] **S. Suganya, P.M Durai Raj Vincent** "Improving Cloud Security by Enhancing Remote Data Integrity Checking Algorithm" International Conference on Innovations in Power and Advanced Computing Technologies, 2017.

[25] **Rubal Deep Gill , Neha Kapur , Harpreet Singh Gill** " Increase Security of Data With Respect to Both Confidentiality and Integrity over Cloud " International Journal of Applied Engineering Research Volume 13, Number 10 (2018) pp. 7388-7391

[26] **Guoping Wang** ,"An Efficient Implementation of SHA-1 Hash Function" IEEE International Conference on Electro/Information Technology, 2006.

## BIOGRAPHIES

**Vidhya Vijayan**, post-graduation student in College of Engineering, Kidangoor. Specialization in Computer and Information Science. She received the graduation in Information technology from Caarmel Engineering college, perunad. Areas of interest are cloud computing and cryptography.

**Shandry K K,** is currently Assistant Professor in College of Engineering, Kidangoor. She had completed her MTech in Computer and Information Science at Cochin University of Science and Technology. Her main area is bigdata and data mining. She has attended conferences and published papers on data mining