# SECURE AND EFFICIENT DATA SHARING IN CLOUD COMPUTING WITH PRIVILEGE VERIFICATION

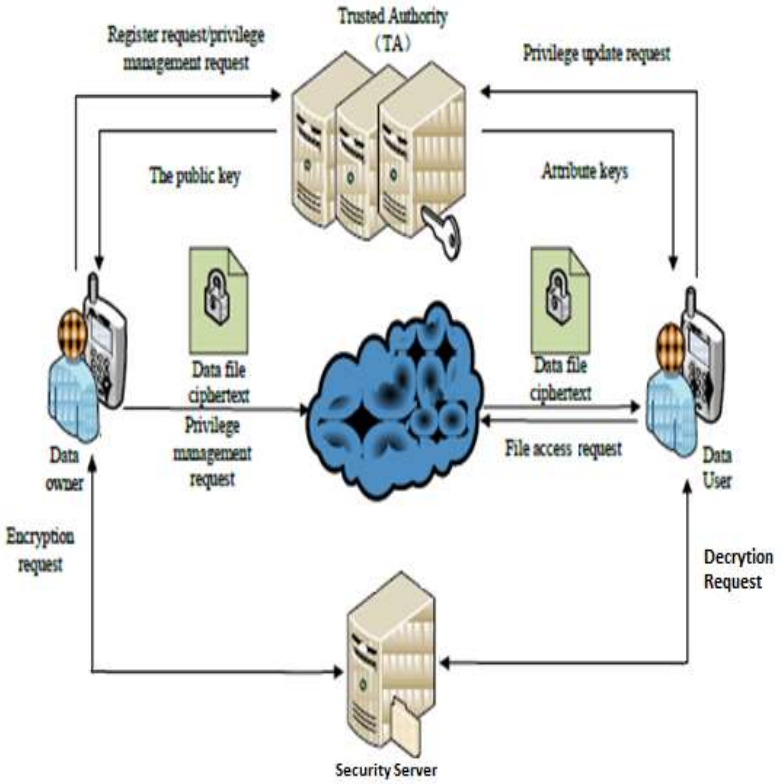
Ajith .M[1], kalaiyarasan.C[2], Kanagaraj.S[3], Som Seker.S[4], Ashok kumar.V[5,](M.E), Assistant professor,Vijayanand.S[6] (M.E, Phd),
The Kavery Engineering College,Computer Science & Engineering , Mecheri.

**ABSTRACT:**
*the development of cloud computing, mobile devices can store/retrieve personal data from anywhere at any time.as a consequence of the data security problem in mobile cloud becomes more and more severe and prevents further development of mobile cloud. There are substantial studies that have been conducted to improve the cloud security. the most of them are not fit for mobile cloud first mobile devices only have limited computing resources and power. Solutions with low computational overhead are in great need for mobile cloud applications. In this paper, we propose a lightweight data sharing scheme (LDSS) for mobile cloud computing. It depends on CP-ABE, an access control technology used in normal cloud environment, but changes the structure of access control tree to make it suitable for mobile cloud in the context. In addition KP-ABE is implemented in TA to ensure attribute based key encryption and decryption from user. The combination of CP and KP ABE attain better performance compared to existing approaches.*

***INDEX:*** **Attribute-based encryption, secure systems, applied cryptography**

## 1. INTRODUCTION:

With the development of cloud computing and the famous of smart mobile devices, people are gradually getting accustomed with new area of data sharing models .which the data is stored on the cloud and the mobile devices are used to store/retrieve the data from the cloud. Typically, mobile devices only have limited storage space and computing power. On the contrary, the cloud has enormous amount of resources.In such a scenario, to achieve the satisfactory performance, it is essential to use the resources provided by the cloud service provider (CSP) to store and share the data.Variouscloud mobile applications have been widely used. In these applications, people (data owners) can upload their photos, videos, documents and other files to the cloud and share these data with other people (data users) they like to share. CSPs also provide data management functionality for data owners. Since personal data files are sensitive, data owners are allowed to choose whether to make the data files public or can only be shared with specific data users. Clearly, data privacy of the personal sensitive data is a big concern for many data owners.data is secure against the CSP. However, the data encryption brings new problems. so they can grant/revoke data access privileges easily on the data users. There have been substantial researches on the issue of data access control over cipher text In these researches, they have the following common assumptions. First, the CSP is considered honest andcurious. Second, all the sensitive data are encrypted before uploaded to the Cloud. Third, user authorization on certain data is achieved through encryption/decryption key distribution. In general, wecan divide these approaches into four categories: simple cipher text access control, hierarchical accesscontrol, access control based on fully holomorphic encryption [1][2] and access control based attributebased encryption (ABE). All these proposals are designed for non-mobile cloud environment .They consume large amount of storage and computation resources, which are not available for mobile devices. According to the experimental results in [26], the basic ABE operations take much longer time on mobile devices than laptop or desktop computers. It is at least 27 times longer to execute on a smart phone than a personal computer (PC)




. (1.1.overview of secure and efficient)

This means that an encryption operation which takes one minute on a PC will take about half an hour to finish on a mobile device. Moreover, current solutions don't solve the user privilege change problem very well. Such an operation could result in very high canceled cost. This is not applicable for mobile devices as well. Clearly, there is no proper solution which can effectively solve the secure data sharing problem in mobile cloud. The mobile cloud becomes excess and more popular, providing an efficient secure data sharing mechanism in mobile cloud is in urgent need. To address this issue, in this paper, we propose a Lightweight Data Sharing Scheme (LDSS) for mobile cloud computing environment. The main contributions of LDSS are as follows:

I. (1) We design an algorithm called LDSS-CP-ABE based on Attribute-Based Encryption (ABE) method to offer efficient access control over cipher text.

II. (2) We use proxy servers for encryption and decryption operations. In our approach, computational intensive operations in ABE are conducted on proxy servers, which greatly reduce the computational overhead on client side mobile devices. Meanwhile, in LDSS-cipher text policy-attribute based encryption, in order to maintain data privacy, a version attribute is also added to the access structure. The decryption key format is modified so that it can be sent to the proxy servers in a secure way.

III. (3) Introduce to re-encryption and explanation of field attributes to reduce the canceled overhead when dealing with the user revocation problem.

IV. (4) Finally, we implement a data sharing prototype framework based on Light weight secure data sharing scheme. The experiments show that secure data can greatly reduce the overhead on the client side, which only introduces a minimal additional cost on the server side. That an approach is useful to implement a data sharing security scheme on mobile devices. The results also show that data sharing hasbetterperformance.

**2. RELATED WORK:**

In 2005 Sahai and Waters [1] introduced the concept of ABE. There are two categories of ABE, KP-ABE and Ciphertext PolicyABE. Goyal et al. [15] proposed the rst KP-ABE, in which a ciphertext is related to a set of attributes, and each private key corresponds to an access policy over the attributes. The decryption can be fullled correctly if and only if the attribute set of the cipher text satises the access policy on the decryptor's private key. Reversely, Bethencourt et al. [2]proposed Ciphertext Policy-ABE where the ciphertext is associated with an access policy and the private key is related to an attribute set. Note that we here mainly focus on reviewing Ciphertext Policy-ABE. Later on, Cheung and Newport [10] proposed a provably secure Cipher text policy-ABE scheme which only supports AND gates over attributes. The rst fully expressive Cipher text Policy-ABE was proposed by Waters [26]. Using dual system encryption, Lew koetal. 1] proposed a fully secure Cipher text Policy-ABE which leads to some loss of efficiency compared to the most e client scheme proposed in [26]. Recently, Attrapadung et al. [3] proposed a CP-ABE with constant-size cipher texts.The

Following introduction of decryption rights delegation by Mambo and Okamoto [3], Blaze et al. [5] formalized proxy re-encryption and proposed a seminal bidirectional prescheme. After that, Ivan and Dodis [1] formalized the de nation of bidirectional and unidirectional proxy functions. In 2005, Ateniese et al. [1,2] proposed three unidirectional preschemes with CPA security. Later on, many classic preschemes (e.g., [8,16,21]) have been proposed.To implement in the attribute-based cryptographic setting, Liang et al. [20] denned CiphertextPolicy-ABPRE, and proposed a concrete construction based on a CiphertextPolicy-ABE scheme [10] in which access policy is only represented as AND gates on positive and negative attributes. Mizuno Doi [4] proposed a hybrid PRE where thescheme can bridge ABE and IBE in the sense that ciphertexts generated in the context of ABE can be converted to the ones which can be decrypted in the IBE setting. Lou et al. [2] proposed a CiphertextPolicy-ABPRE scheme which supports AND gates on multi-valued and negative attributes. The aforementioned CiphertextPolicy-ABPRE schemes, however, are only secure against CPA and supports AND gates over attributes. The construction of a CCA secure CiphertextPolicy-ABPRE supporting any monotonic access policy remains open. This paper deals with thisproblem.

**2.1.ATTRIBUTE-BASED ENCRYPTION:**

now we give an overview of Attribute-Based Encryption algorithms.The security Sahai-Waters [26] (ABE) cryptosystem as implemented in this paper is specifically detailed. We focus our efforts on providing the description of the scheme and intuition for its construction. the proof of security Sahai and Waters [26].Attribute-Based Encryption can be viewed as a generalization of Identity-Based Encryption [5,9,30]. In a user's identity is a string such as "bobsmith@yahoo.com".A party in the system can encrypt a message to this particular user with only the knowledge of the recipient's identity and the system's public parameters. In particular the encryption algorithm does not need to have access to a separate public key certificate of the recipient.In Attribute-Based Encryption a user's identity is com-posed of a set, S, of strings which serve as descriptive at-tributes of the user. For example, a user's identity could consist of attributes describing their university, department, and job function. A party in the system can then specify another set of attributes so such that a receiver can only decryption a message if his identity S has at least k attributes.

**2.2.4.SAHAI-WATERS CONSTRUCTION:**

Our second observation is based upon our experience in building an implementation of the security Sahai-Waters construction. We have found T requires a great deal of computational effort. It is easily seen that the number of exponentiations required to solve T is equal to n + 1.users are certified in an ABE system is analogous to certification in a PKI. Similar to a traditional PKI, a user presents the authority with a set of credentials that prove their right to fulfill an attribute. Instead of map-ping a user to an identity, certification establishes that the user fulfills the semantic of the attribute. Such semantics are specific to the supported community (e.g. job function in a business system, clubs belonged to in a social network). This process is repeated for all attributes appropriate to each user. Key distribution is significantly simplified in such a system, as public keys are simply the combination of the cryptosystem's public parameters and attribute names.The revocation process is significantly different in a Attribute Based Encryption system as attributes, not users or keys, are revoked. In fact, there is no way to revoke a user, save revoking all of his attributes. Like traditional PKI systems, revocation can impact all users who either have or use an attribute. Un-like traditional PKI systems, however, the compromise of a particular attribute may not mandate its revocation. As Section 4 details, it is the specific application of multiple attributes that defines policy. The compromise of any single attribute may therefore be a necessary but not sufficient condition for its revocation. Consequently, it may be desirable to revoke all, a subset or none of the compromised user attributes. Explored in depth in Section 6.3, we consider both online and offline revocation approaches.A superficial reading of the above issues may lead one to falsely conclude that ABE systems must be online. The creation of keys, certification of users, and adding attributes are largely isomorphic to certification issuance operations present in current PKI. Revocation can also be handled offline (however, online approaches such as OCSP.A superficial reading of the above issues may lead one to falsely conclude that ABE systems must be online. The creation of keys, certification of users, and adding attributes are largely isomorphic to certification issuance operations present in current PKI. Revocation can also be handled offline (however, online approaches such as OCSP.

**3.SYSTEM STUDY:**

**3.1.FEASIBILITYSTUDY:**

The processing of the project is analyzed in this phase and business plan is put forth with a very general plan for the project and some cost estimates. During the system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For processing analysis, some understanding of the major requirements for the system essentials.

Three key considerations involved in the feasibility analysis are:

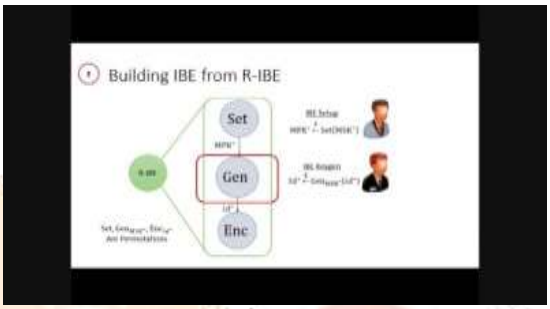
- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

### 3.1.2.ECONOMICAL FEASIBILITY:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research TO development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### 3.1.3.TECHNICAL FEASIBILITY

This study is carried out to check the technical processing, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical term resources. This will lead to high demands on the available technical resources.




Data users are clients register their details for storing and retrieving of data in network. A data user can have multiple attributes corresponding to multiple data files. A data owner can define a set of attributes for its data files. The data accesses are managed by access control policy specified by data owners.

## 4.IMPLEMENTATION:

execution of the project theoretical design is turned into a working system and is giving confidence on the new system for the users work will efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, an evaluation, of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The more complex system being implemented, the more involved will be the system analysis and the design effort required just for implementation on network processing. An implementation co-ordination committee based on policies of individual organization has been appointed. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system. Implementation is the final and important phase, the most critical stage in achieving a successful new system and in giving the users confidence. That the new system will work is effective.The system can be implemented only after through testing is done and if it found to working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

## 4.MODULES:

1. Data user
2. Data owners
3. Trusted Authority
4. User Verifications

### 4.1.DATA USER:

Data users are clients register their details for storing and retrieving of data in network. A data user can have multiple attributes corresponding to multiple data files. A data owner can define a set of attributes for its data files. The data accesses are managed by access control policy specified by data owners.

**4.2.DATA OWNERS:**

Data Owners can upload their documents and other files to the cloud and share these data with other people (data users) they like to share. CSPs also provide data management functionality for data owners. Since personal data files are sensitive, data owners are allowed to choose whether to make their data files public or can only be shared with specific data users. Clearly, data privacy of the personal sensitive data is a big concern for many data owners. People upload their data files onto the cloud, they are leaving the data in a place where is out of their control, and the CSP may spy on user data for its commercial interests and/or other reasons.[7] Second, people have to send password to each data user if they only want to share the encrypted data with certain users, which is very cumbersome. To simplify the privilege management, the data owner can divide data users into different groups and send password to the groups which they want to share the data.fig(7),

**4.2.1.ENCRYPTION MODES.**

Encryption of data at rest has different considerations than the traditional communication encryption model. For example, to enable random-access, FDE implementations treat each disk sector as an autonomous unit and assign sector-specific IVs for chaining modes such as CBC. These IVs are long-term and must be easily derived from or stored in the local system. When FDE is implemented with a CBC-mode cipher, information leakage about the plaintext disk content may occur without knowledge of the encryption key or cipher used (see e.g., [18]).

**4.2.2.DECRYPTION MODE:**

structure operations are pairings following by Decryption as describe by Sahai and Waters has the following form [26]: denotes a pairing operation. In the equation above, there are 2k pairings and k exponents. Decryption can be optimized to reduce the number of bilinear map operations by bringing the Lagrange coefficients in:Tweak able block cipher modes (e.g., LRW and XTS) have been designed specifically for disk encryption to prevent attacks such as watermarking, malleability, and copy-and-paste. These attacks are particularly important for PDE, as they may be used to identify hidden volumes without recovering any hidden plaintexts. This optimization reduces the number of bilinear map operations from 2k to k + 1 at the expense of increasing the number of exponentiations from k to 2k. Because bilinear map operations are more computationally intensive than exponentiation's, this optimization increases the overall speed of decryption.The default Android FDE uses CBC. We choose to move away from the Android default and instead, use XTS-AES [23, 34] to prevent known attacks against CBC. XTS-AES is a code book mode (i.e., no block chaining) and uses a secondary "tweak" key to make unpredictable use of the disk sector index. XTS-AES is not an authenticated mode, and as such is considered malleable [23]. However, unlike CBC, XTS is not malleable at a bit granularity: a modified ciphertext block will decrypt to a random plaintext block, preventing an attacker from making a predictable change. The absence of authentication tags also allows for a copy-and-paste attack (i.e., successful decryption of sectors that have been moved from other disk locations). Using CBC with random IVs will garble only the first block, but successfully decrypt all subsequent blocks in the moved sector.secure level bitlockerXTS-advance encryption standard does not on block chaining, and uses the tweak to entangle plaintext/ciphertext block pairs with their disk sector location. As such, all blocks in a moved sector will decryption to random plaintext. A watermark attack relies on predictable IVs, and is mounted by convincing the user to encrypt and store a file that has been specifically crafted to effectively zero out the IVs. The watermark manifests itself as identical ciphertext blocks at the beginning of consecutive disk sectors. The attacker can then examine the encrypted storage and locate the watermark. Both XTS-AES (Mobiflage) and CBC with ESSIV (Android FDE), effectively prevent watermarking attacks.

**5.EXISTING SYSTEM :**

An encryption operation which takes one minute on a PC will take about half an hour to finish on a mobile device. Furthermore, current solutions don't solve the user privilege change problem very well. Such an operation could result in very high revocation cost.[6] This is not applicable for mobile devices as well. Clearly, there is no proper solution which can effectively solve the secure data sharing problem in mobile cloud.[1] As the mobile cloud becomes more and more popular, providing an systematic secure data sharing mechanism in mobile cloud is in urgent need.However, the system needs to obtain large amount of information of keys. Borrows the access control method used in conventional distributed storage, separating users into different groups according to access rights and assign different keys to groups. This reduces the overhead of key management, but it cannot satisfy the demand for fine-grained access control. Hierarchical access control has good performance in reducing the overhead of key distribution in cipher text access control. As a result, there is substantial research on cipher text access control based on hierarchical access control method. In hierarchical access control method, keys can be derived from private keys and a public token table. However, the operation on token table is complicated and generates high cost. Besides, the token table is stored in the cloud. Its privacy and security cannot be guaranteed.

**4.3.TRUSTED AUTHORITY (TA):**

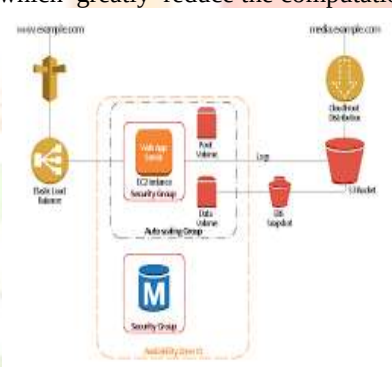
A trusted authority (TA) is introduced. It is responsible of generating public and private keys, and distributing attribute keys to users. With this mechanism, users can share and access data without being aware of the encryption and decryption operations. We assume trust authority is entirely credible, and a trusted channel exists between the TA and every user. The fact that a trusted channel exists doesn't mean that the data can be shared through the trusted channel, for the data can be in a large amount. TA is only used to transfer keys (in a small amount) securely between users.[8] In addition, it's requested that TA is online all the time because data users may access data at any time and need TA to update attribute keys.

**4.4.USER VERIFCATION:**

Data User logins onto the system and sends, an authorization request to Trusted Agent. The authorization request includes attribute keys which Data User already has. Trusted Agent A accepts the authorization request and checks whether Data User has logged on before. If the user hasn't logged on before. TA calls Key Generation to generate attribute keys for Data User. Trusted Agent compares the attribute description field in the attribute key with the attribute description field stored in database. If they are not match, for each inconsistent bit in description field, if it is 1 on data user's side and 0 on TA's side, it indicates that DU's attribute has been revoked, and then TA does nothing on this bit. If it is reversed scenario, it indicates that DU has been assigned with a new attribute, then TA generates the corresponding attribute key for DU.[7] If they are match for TA checks the version of every attribute key of DU.[2] If it's not the same with the current version, then TA updates the corresponding attribute key for DU.

**6.PROPOSED SYSTEM:**

A Lightweight Data Sharing Scheme for mobile cloud computing environment has been proposed. It designs an algorithm called LDSS-Cipher text Policy based on Attribute - Based Encryption (ABE) method to offer efficient access control over cipher text. proxy server using for encryption and decryption operations.[8] In our approach, computational intensive operations in attribute based encryption are conducted on proxy servers. which greatly reduce the computational overhead on client side mobile devices.



(4.3.overview of trusted authority)

Meanwhile, in LDSS-ciphertext policy-ABE, in order to maintain data privacy, aversion attribute is also added to the access structure. The decryption key format is modified so that it can be sent to the proxy servers in a secure way. (3)introducing encryption and re encryption the field of attributes to reduce the revocation overhead when dealing with the user revocation problem.[5] (4) Finally, we implement a data sharing prototype framework based on light weight data sharing secure. The experiments show that can greatly reduce the overhead on the client side, which only introduces a minimal additional cost on the server side. Such an approach is beneficial to implement a realistic data sharing security scheme on mobile devices. The results also show that secure data has better performance compared to the existing ABE based access control schemes over cipher text.

**7.CONCLUSION:**

In recent years, many studies on access control in cloud are based on attribute-based encryption algorithm (ABE). However, traditional ABE is not suitable for mobile cloud because it is computationally intensive and mobile devices only have limited resources. In this paper, we propose LDSS to address this issue. It introduces a novel LDSS-CP-ABE algorithm to migrate major computation overhead from mobile devices onto proxy servers, thus it can solve the secure data sharing problem in mobile cloud. To enable more security KP-ABE based encryption is used to decrypt file. Hence it reduces work of ESP and DSP and it eradicate the issue if symmetric key known to both ESP and DSP which is discussed in existing method. The experimental results show that LDSS can ensure data privacy in mobile cloud and reduce the overhead on users' side in mobile cloud. In the future work, we will design new approaches to ensure data integrity. To further tap the potential of mobile cloud, we will also study how to do ciphertext retrieval over existing data sharing schemes**.**

**8. REFERNCES:**

TABLE I. http://java.sun.com/products/archive/jaas/.

TABLE II. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores. In *ACMconference on* Computer and Communications Security, pages 598–609,2007.

TABLE III. E. Bertino, P. Bonatti, and E. Ferrari. Trbac: A temporal role-based access control model. *ACM* Transactions on Information and SystemSecurity *(TISSEC)*, 4(3):191–233, 2001.

TABLE IV. C. Fruhwirth. New methods in hard disk encryption. Tech-nical report, Vienna University of Technology (July 2005). http://clemens.endorphin.org/nmihde/nmihde-A4-ds.pdf.

TABLE V. M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu. Plutus: Scalable secure le sharing on untrusted storage. In Proceedings of the USENIX Conference on File and Storage Technologies, pages29{42, 2003.

TABLE VI. V. Kher and Y. Kim. Securing distributed storage: challenges, techniques, and systems. In Proceedings of the ACM workshop on Storage security and survivability, pages 9{25, 2005.

TABLE VII. Craig Gentry and Zul kar Ramzan. Single-database private information retrieval withconstant communication rate. In Lu s Caires, Giuseppe F. Italiano, Lus Monteiro,