International Conference on Recent Innovations in Science, Engineering & Management
R.R. INSTITUTE OF TECHNOLOGY, Bengaluru (India)
10th - 11th May 2019 www.conferenceworld.in ISBN : 978-93-87793-87-3

Efficient Secure Data Sharing In Mobile Cloud Environment Using Light Weight Encryption

SumanthV[1] , ShivaSelvi[2] , Shreekanth[3],Bhimavva P[4], Mounika P.V[5]

[1,2,3,4,5]*Department of CSE ,RRIT,(India)*

ABSTRACT:

The data security problem are becomes more grievous and intercept in the development of mobile clouds. In the mobile clouds there many research are done to improve the mobile cloud security. But many clouds are not supporting to the mobile services since mobile devices has limited power and computing resources. To find the solutions to mobile cloud applications with low computational overhead is important task. In this research paper, we proposed LDSS technique to solve this issues for mobile cloud computing It adopts 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 environments. LDSS moves a large portion of the computational intensive access control tree transformation in CP-ABE from mobile devices to external proxy servers. . The experimental results show that LDSS can effectively reduce the overhead on the mobile device side when users are sharing data in mobile cloud environments.

Keywords: Mobile cloud computing,data encryption,access control,user revocation

[1] INTRODUCTION

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers.

1.1 Benefits of cloud computing

Consumers and organizations have many different reasons for choosing to use cloud computing services. They might include the following:

- Convenience
- Scalability
- Low cost
- Security

1.2 Applications of cloud computing

- Storing File Online**:**Cloud computing provide a benefit to store and access the software with the help of internet connection to the cloud. The interface provided is very easy to operate and is economical
- Video making and Editing Software:There are many software available which can access with the help of the cloud . This software helps to create and modify are stored in the cloud itself and we can access anytime
- File Converters**:**There are many applications which utilize to change to format to the file such that from HTML to pdf and so on. software at cloud and access from anywhere with the help of internet connection**.**

1.3 Issues in cloud computing

- Cost**:**Cloud : computing itself is affordable, but tuning the platform according to the company's needs can be expensive. Furthermore, the expense of transferring the data to public clouds can prove to be a problem for short-lived and small-scale projects .Companies can save some money on system maintenance, management, and acquisitions. But they also have to invest in additional bandwidth, and the absence of routine control in an infinitely scalable computing platform can increase costs.
- PasswordSecurity**:**Industrious password supervision plays a vital role in cloud security. However, the more people you have accessing your cloud account, the less secure it is. Access rights related to passwords and usernames should only be allocated to those who require them.
- Data privacy **:**Sensitive and personal information that is kept in the cloud should be defined as being for internal use only, not to be shared with third parties. Businesses must have a plan to securely and efficiently manage the data they gather.

[2] Problem Statement

With the popularity of cloud computing, mobile devices can store/retrieve personal data from anywhere at any time. Consequently, the data security problem in mobile cloud becomes more and more severe and prevents further development of mobile cloud, Solutions with low computational overhead are in great need for mobile cloud applications we propose a lightweight data sharing scheme (LDSS) for mobile cloud computing. It adopts 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 environments , LDSS moves a large portion of the computational intensive access control tree transformation in CP-ABE from mobile devices to external proxy servers

2.1 Existing system

- In general, we can divide these approaches into four categories: simple ciphertext access control, hierarchical access control, access control based on fully homomorphic encryption and access control based on attribute-based encryption (ABE). All these proposals are designed for non-mobile cloud environment

- Tysowski et al. considered a specific cloud computing environment where data are accessed by resource-constrained mobile devices, and proposed novel modifications to ABE, which assigned the higher computational overhead of cryptographic operations to the cloud provider and lowered the total communication cost for the mobile user.

Disadvantages of existing system:

- Data privacy of the personal sensitive data is a big concern for many data owners.
- The state-of-the-art privilege management/access control mechanisms provided by the CSP are either not sufficient or not very convenient.
- They cannot meet all the requirements of data owners.
- They consume large amount of storage and computation resources, which are not available for mobile devices
- Current solutions don't solve the user privilege change problem very well. Such an operation could result in very high revocation 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.

2.2 Proposed system

- We propose a Lightweight Data Sharing Scheme (LDSS) for mobile cloud computing environment.
- The main contributions of LDSS are as follows:
- We design an algorithm called LDSS-CP-ABE based on Attribute-Based Encryption (ABE) method to offer efficient access control over ciphertext.
- 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-CP-ABE, 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.
- We introduce lazy re-encryption and description field of attributes to reduce the revocation overhead when dealing with the user revocation problem.
- Finally, we implement a data sharing prototype framework based on LDSS.

Advantages of proposed system:

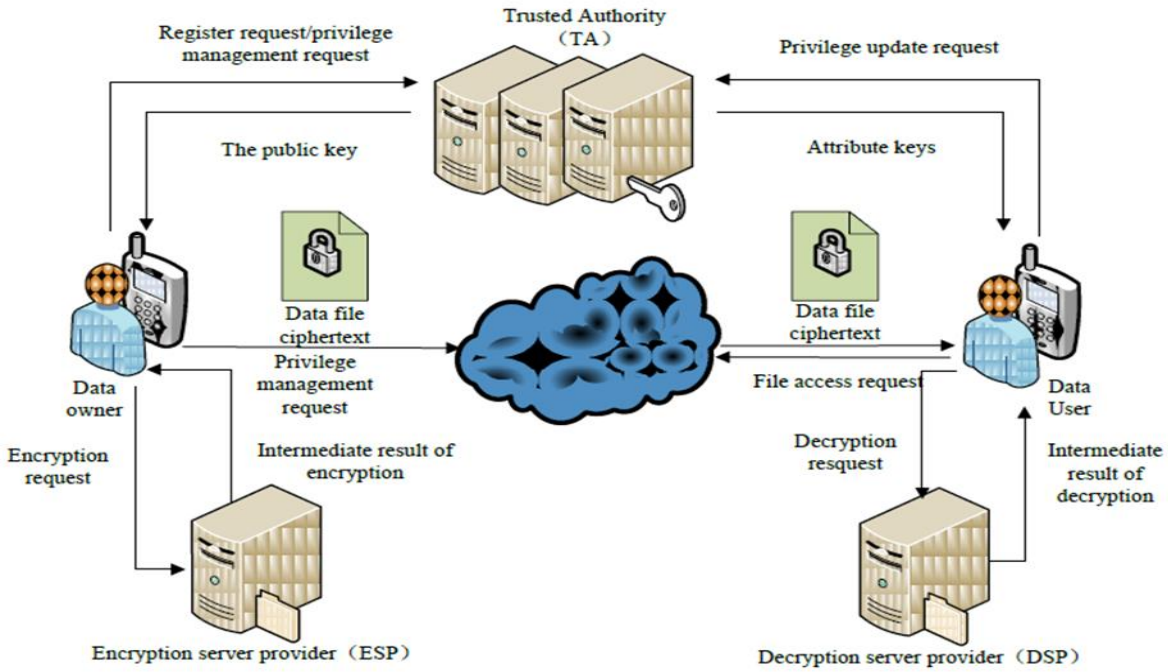
The experiments show that LDSS can greatly reduce the overhead on the client side, which only introduces a minimal additional cost on the server side.

- Such an approach is beneficial to implement a realistic data sharing security scheme on mobile devices.
- The results also show that LDSS has better performance compared to the existing ABE based access control schemes over ciphertext.
- Multiple revocation operations are merged into one, reducing the overall overhead

- In LDSS, the storage overhead needed for access control is very small compared to data files.

[3] Methodology

3.1 System Architechture



Data owner and data user both will registerboth will register with the trusted authority , trusted authority checks if the data user and data owner is authorized then the trusted authority generates the public key and attribute keys for the data user data owner gives encryption request to Encryption service provider , encryption service provider gives the intermediate result and the data owner can upload the ciphertext file to the cloud , if data user want to acess the file uploaded in cloud data user can send the decryption request to the Decryption server provider , decryption service provider gives the intermediate result and then data user can give file access request to cloud and access the file .

3.2 Modules

(1) Data owner : When the data owner (DO) registers on TA, TA runs the algorithm Setup() to generate a public key PK and a master key MK. PK is sent to DO while MK is kept on TA itself. DO defines its own attribute set and assigns attributes to its contacts. All these information will be sent to TA and the cloud. TA and the cloud receive the information and store it. DO uploads data to the mobile cloud and share it with friends. DO determines the access control policies. DO sends data to the cloud. Since the cloud is not credible, data has to be encrypted before it is uploaded. The DO defines access control policy in the form of access control tree on data files to assign which attributes a DU should obtain if he wants to access a certain data file.

(2) Data User :DU logins onto the system and sends, an authorization request to TA. The authorization request

includes attribute keys (SK) which DU already has.TA accepts the authorization request and checks the request and a generate attribute keys (SK) for DU. DU sends a request for data to the cloud.Cloud receives the request and checks if the DU meets the access requirement. DU receives the ciphertext, which includes ciphertext of data files and ciphertext of the symmetric key. DU decrypt the ciphertext of the symmetric key with the assistance of DSP. DU uses the symmetric key to decrypt the ciphertext of data files.

(3) Trusted Authority :To make LDSS feasible in practice, 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 TA 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. 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) Cloud Service Provider:CSP stores the data for DO. It faithfully executes the operations requested by DO, while it may peek over data that DO has stored in the cloud. DU sends a request for data to the cloud. Cloud receives the request and checks if the DU meets the access requirement. If DU can't meet the requirement, it refuses the request; otherwise it sends the ciphertext to DU. CSP manages the Uploaded Files.

Algorithms

3.3 Algorithms

(1) Setup( ) Algorithm:

Setup(A, V): Generate the master keyMK, the public keyPK based on attribute set A of the Data Owner and theversion attribute V .

Input: The attribute set A , the version attribute V.

Output**:** The master key MK, the public key PK.

- Construct a p-order bilinear group G0 of generator g and a bilinear mapping e:$G_0 * G_0 = G_1$.
- Randomly choose a,b€Z_p and calculate g^b, $e(g, g)^a$.
- For each attribute a_i in A, randomly choose t_i€Z_p , and calculate $X_i = g^{ti}$.
- For V, randomly choose t_v€Z_p , and calculate $x_v = g^{tv}$
- Return the master key MK and the public key PK, Wherein MK={a,b},
  PK={ G0, g , g^b, $e(g,g)^a$,$\{X_i\}i^k_{=1}$,X_v }.

(2) KeyGen(Au, MK): Generate attribute keys SKu for a data user based on his attribute set Au and the master key MK.

Input: The attribute set Au , the master key MK={a,b}.

Output: Attribute keys associated with Au

- Randomly choose a parameter r€Zp,and calculate $SKr = g^{(a+r)/b}$
- For each Attribute ai in Au , randomly choose ri€Zp , and calculate SKa={gri,gr.Xiri}^{j} i=1
- For V , randomly choose rv € Zp, and calculate SKv = {grv,$gr._{Xv}^{rv}$ }

- Return $SK_u=\{SK_r,SK_a,SK_v\}$

(3) Encryption(K, PK, T): Generate the ciphertext CT based on the symmetric key K, public key PK and access control tree T

Input**:** The symmetric key K, public key PK, access control tree T (including the left subtreeTa, right subtreeTv, and left subtree has num leaf nodes).

Output: The ciphertextCT.

- Randomly choose S€Zp as the secret of T, and calculate CTk={g^{bS}, K .e(g,g)^{aS} }.
- Get the value of the two children (namely Sa, Sv) of the root node according to the access control tree.
- Calculate CTv={g^{Sv}, $g^{r}.X_v^{Sv}$ }
- Return CT={CTk, CTa, CTv }.

[4] Conclusion and Future Work

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. 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.

References

[1] Gentry C, Halevi S. Implementing gentry's fully-homomorphic encryption scheme. in: Advances in Cryptology–EUROCRYPT 2011. Berlin, Heidelberg: Springer press, pp. 129-148, 2011.

[2] Brakerski Z, Vaikuntanathan V. Efficient fully homomorphic encryption from (standard) LWE.in: Proceeding of IEEE Symposium on Foundations of Computer Science. California, USA: IEEE press, pp. 97-106, Oct. 2011.

[3] Qihua Wang, Hongxia Jin. "Data leakage mitigation for discertionary access control in collaboration clouds".the 16th ACM Symposium on Access Control Models and Technologies (SACMAT), pp.103-122, Jun. 2011.

[4] Adam Skillen and Mohammad Mannan.On Implementing Deniable Storage Encryption for Mobile Devices.the 20th Annual Network and Distributed System Security Symposium (NDSS), Feb. 2013.

[5] Wang W, Li Z, Owens R, et al. Secure and efficient access to outsourced data. in: Proceedings of the 2009 ACM workshop on Cloud computing security. Chicago, USA: ACM pp. 55-66, 2009.

[6] Maheshwari U, Vingralek R, Shapiro W. How to build a trusted database system on untrusted storage.in: Proceedings of the 4th conference on Symposium on Operating System Design & Implementation-Volume 4. USENIX Association, pp. 10-12, 2000.

[7] Kan Yang, XiaohuaJia, KuiRen: Attribute-based fine-grained access control with efficient revocation in cloud storage systems. ASIACCS 2013, pp. 523-528, 2013.

[8] Crampton J, Martin K, Wild P. On key assignment for hierarchical access control.in: Computer Security Foundations Workshop. IEEE press, pp. 14-111, 2006.

[9] Shi E, Bethencourt J, Chan T H H, et al. Multi-dimensional range query over encrypted data. in: Proceedings of Symposium on Security and Privacy (SP), IEEE press, 2007. 350364

[10] Cong Wang, KuiRen, Shucheng Yu, and KarthikMahendraRajeUrs.Achieving Usable and Privacy-assured Similarity Search over Outsourced Cloud Data. IEEE INFOCOM 2012, Orlando, Florida, March 25-30, 2012