A Lightweight Secure Data Sharing Scheme for Mobile Cloud Computing

R.Deepalakshmi#1,K.Guanasundari#2,Mrs. G.Deepa, MCA.,M.Phil.,Ph.D., #3

Pg Scholar, Assistant Professor &DSCASW College

Department of MCA &Dhanalakshmi Srinivasan College of Arts and Science for Women's- Perambalur, Tamilnadu, India

[1] deepa.asd1995@gmail.com,[2]Sundariasd31@gmail.com,[3]deepasanmathi@gmail.com

Abstract

With the recognition of cloud computing, mobile devices will store/retrieve personal knowledge from anyplace at any time. Consequently, the info security downside in mobile cloud becomes a lot of and a lot of severe and prevents more development of mobile cloud. There area unit substantial studies that are conducted to enhance the cloud security. However, most of them don't seem to be applicable for mobile cloud since mobile devices solely have restricted computing resources and power. Solutions with low machine overhead area unit in nice want for mobile cloud applications. In this paper, we have a tendency to propose a light-weight knowledge sharing theme (LDSS) for mobile cloud computing. It adopts CP-ABE, associate degree access management technology employed in traditional cloud surroundings, however changes the structure of access management tree to create it appropriate for mobile cloud environments. LDSS moves an oversized portion of the machine intensive access management tree transformation in CP-ABE from mobile devices to external proxy servers. Furthermore, to cut back the user revocation value, it introduces attribute description fields to implement lazy-revocation, that could be a thorny issue in program based mostly CP-ABE systems. The experimental results show that LDSS will effectively scale back the overhead on the mobile device aspect once users area unit sharing knowledge in mobile cloud environments.

Keywords: Mobile Cloud Computing, Data Encryption, Access Control, User Revocation

INTRODUCTION

Various cloud 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 additionally give knowledge management practicality for knowledge homeowners. Since personal knowledge files square measure sensitive, knowledge homeowners square measure allowed to decide on whether or not to create their knowledge files public or will solely be shared with specific knowledge users. Clearly, knowledge privacy of the non-public sensitive knowledge could be a huge concern For several knowledge homeowners. The progressive privilege management/access management mechanisms provided by the CSP square measure either not sufficient or not terribly convenient. They cannot meet all the necessities of information homeowners. First, when 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. 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. However, this approach requires fine-grained access control. In both cases, password management is a big issue. Apparently, to solve the above problems, personal sensitive data should be encrypted before uploaded onto the cloud so that the data is secure against the CSP. However, the data encryption brings new problems. How to provide efficient access control mechanism on ciphertext decryption so that only the authorized users can access the plaintext data is challenging. In addition, system must offer data owners effective user privilege management capability, 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 ciphertext. In these researches, they have the following common assumptions. First, the CSP is considered honest and curious. 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, we will divide these approaches into four categories: straightforward ciphertext access management, hierarchical access control, access control based on fully homomorphic encryption and access control based on attribute-based encryption (ABE). All these proposals square measure designed for non-mobile cloud surroundings. They consume great deal of storage and computation resources, which are not available for mobile devices. According to the experimental results in, 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). 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. Furthermore, current solutions don’t solve the user privilege change problem very well. Such AN operation might end in terribly high revocation price. This is not applicable for mobile devices moreover. Clearly, there's no correct resolution which might effectively solve the secure knowledge sharing drawback in mobile cloud. As the mobile cloud becomes more and more popular, providing an efficient secure data sharing mechanism in mobile cloud is in urgent need.

RELATED WORK

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.

We describe a working implementation of a variant of Gentry’ s fully homomorphic encryption scheme (STOC 2009), similar to the variant used in an earlier implementation effort by Smart and Vercauteren (PKC 2010). Smart and Vercauteren implemented the underlying “somewhat homomorphic” scheme, but were not able to implement the bootstrapping functionality that is needed to get the complete scheme to work. We show a number of optimizations that allow us to implement all aspects of the scheme, including the bootstrapping functionality. Our main optimization is a key-generation method for the underlying somewhat homomorphism encryption, that does not require full polynomial inversion. This reduces the asymptotic complexity from ˜O(n2:5) to ˜O (n1:5) when working with dimension-n lattices (and practically reducing the time from many hours/days to a few seconds/minutes). Other optimizations include a batching technique for encryption, a careful analysis of the degree of the decryption polynomial, and some space/time trade-offs for the fully-homomorphism scheme. We tested our implementation with lattices of several dimensions, corresponding to several security levels. From a “toy” setting in dimension 512, to “small,” “medium,” and “large” settings in dimensions 2048, 8192, and 32768, respectively. The public-key size ranges in size from 70 Megabytes for the “small” setting to 2.3 Gigabytes for the “large” setting. The time to run one bootstrapping operation (on a 1-CPU 64-bit machine with large memory) ranges from 30 seconds for the “small” setting to 30 minutes for the “large” setting.

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.

We present a fully homomorphic encryption scheme that is based solely on the (standard) learning with errors (LWE) assumption. Applying known results on LWE, the security of our scheme is based on the worst-case hardness of \short vector problems" on arbitrary lattices. Our construction improves on previous works in two aspects: 1. We show that \somewhat homomorphic" encryption can be based on LWE, using a new re- linearization technique. In contrast, all previous schemes relied on complexity assumptions related to ideals in various rings. 2. We deviate from the \squashing paradigm" used in all previous works. We introduce a new dimension-modulus reduction technique, which shortens the cipher texts and reduces the decryption complexity of our scheme, without introducing additional assumptions. Our scheme has very short cipher texts and we therefore use it to construct an asymptotically efficient LWE-based single-server private information retrieval (PIR) protocol. The communication complexity of our protocol (in the public-key model) is k _ polylog(k) + log jDBj bits per single-bit query (here, k is a security parameter).

ADAM SKILLEN AND MOHAMMAD MANNAN. ON IMPLEMENTING DENIABLE STORAGE ENCRYPTION FOR DEVICES. THE 20TH ANNUAL NETWORK AND DISTRIBUTED SYSTEM SECURITY SYMPOSIUM (NDSS), FEB. 2013.

Data confidentiality can be effectively preserved through encryption. In certain situations, this is inadequate, as users may be coerced into disclosing their decryption keys. In this case, the data must be hidden so that its very exis- tence can be denied. Steganographic techniques and deni- able encryption algorithms have been devised to address this specific problem. Given the recent proliferation of smartphones and tablets, we examine the feasibility and ef- ficacy of deniable storage encryption for devices. We evaluate existing, and discover new, challenges that can compromise plausibly deniable encryption (PDE) in a mo- bile environment. To address these obstacles, we design a system called Mobiflage that enables PDE on de- vices by hiding encrypted volumes within random data on a device's external storage. We leverage lessons learned from known issues in deniable encryption in the desktop environment, and design new countermeasures for threats specific to systems. Key features of Mobiflage in- clude: deniable file systems with limited impact on through- put; efficient storage use with no data expansion; and re- striction/prevention of known sources of leakage and dis- closure. We provide a proof-of-concept implementation for the Android OS to assess the feasibility and performance of Mobiflage. We also compile a list of best practices users should follow to restrict other known forms of leakage and collusion that may compromise deniability.

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.

Providing secure and efficient access to large scale outsourced data is an important component of cloud computing. In this paper, we propose a mechanism to solve this problem in owner-write-users-read applications. We propose to en- crypt every data block with a different key so that exible cryptography-based access control can be achieved. Through the adoption of key derivation methods, the owner needs to maintain only a few secrets. Analysis shows that the key derivation procedure using hash functions will intro- duce very limited computation overhead. We propose to use over-encryption and/or lazy revocation to prevent revoked users from getting access to updated data blocks. We design mechanisms to handle both updates to outsourced data and changes in user access rights. We investigate the overhead and safety of the proposed approach, and study mechanisms to improve data access efficiency.

EXISTING PROCESS

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. Privilege management/access management mechanisms provided by the CSP ar either not spare or not terribly convenient. They cannot meet all the necessities of knowledge homeowners. 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. However, this approach requires fine-grained access control. In both cases, password management is a big issue.

DISADVANTAGE

- Difficult to share key
- Less security
- Not user friendly

PROPOSED PROCESS

Nowadays, various cloud mobile applications have been widely used. In these applications, people (data owners) can upload their photos, documents and other files to the cloud and share these data with other people (data users) they like to share.

CSPs additionally offer information management practicality for information house owners.

Since personal information files ar 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 To propose a Lightweight Data Sharing Scheme (LDSS) for mobile cloud computing environment. The main contributions of LDSS are as follows:

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

(2) Here, 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 decipherment key format is changed in order that it are often sent to the proxy servers in an exceedingly secure approach.

(3) To introduce lazy re-encryption and description field of attributes to reduce the revocation overhead when dealing with the user revocation problem.

(4) Finally, implement a data sharing prototype framework based on LDSS.The experiments show that LDSS will greatly scale back the overhead on the consumer facet, which only introduces a minimal additional cost on the server side.Such AN approach is useful to implement a sensible information sharing security theme on mobile devices.

The results additionally show that LDSS has higher performance compared to the present ABE based mostly access management schemes over ciphertext.

ADVANTAGE

- Secured sharing of data
- Easy access
- More user friendly

PROCESS

- Certification of files
- Privacy protection
- Request generation
- Forward security

- Access the files

CERTIFICATION OF FILES

Cloud storage is predicated on extremely virtualized infrastructure and is like broader cloud computing in terms of accessible interfaces, near-instant snap and measurability, multi-tenancy, and metered resources.Certification of files refers to uploading the files in the cloud. Data Owner can upload the files into the cloud. Data owner upload the files in an Encrypted format in the cloud for providing more security to the particular data.

PRIVACY PROTECTION

Data owner uploads the data or a file only in an encrypted format for providing security. Using X-OR key encryption algorithm, a key will be randomly generated for uploading the data. (key is also in an encrypted format). When the data owner uploads the file into the cloud, data owner can hide some data or a file (data which the owner doesn't want to a public data) among all the files in the cloud/server. Hence user cannot view the hidden files in the server. Whenever data owner wants to display the data, he changes the hidden file into the unmasked data.

REQUEST GENERATION

User can view all the files or except the hidden files or data in the server. If any user wants to access the particular file or data in the server, then he sends a request to the particular data owner. User cannot access the data or a file in the cloud, without the permission of the data owner. Hence data owner can view all the user requests, and verify it. Then the user request will be forwarded to the Trusted Third Party Authenticator and the TTP will send the authentication to the user.

FORWARD SECURITY

Forward security is mainly focused on trusted third party. After data owner view the user requests, he forwards the secret key to the trusted third party. Trusted third party auditor is to verify the user whether he is authorized user or not. Unauthorized user cannot access the data. After confirming the user is authorized, then only the trusted third party authenticator sends the secret key to the particular user email ID.

ACCESS THE FILES

Finally, the secret key will be mailed to the particular user. User can access the file or a data in the server only using the particular secret key. The secret key is to decrypt the file and download the file from the server.

ARCHITECTURE

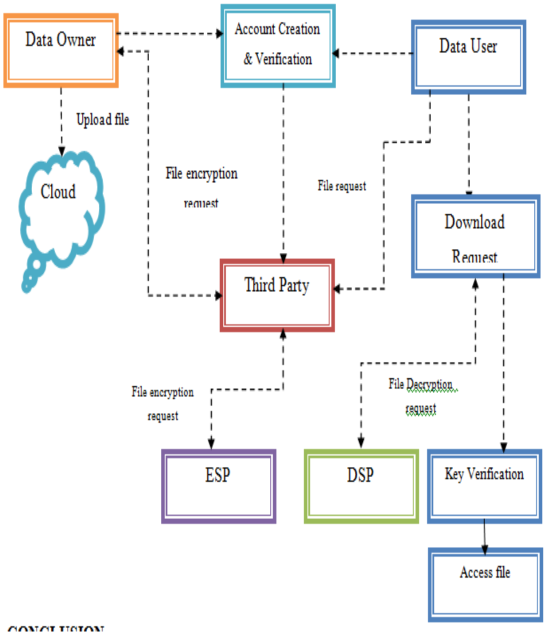

Figure 1: Architecture

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 cloud because it is computationally intensive and   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 devices onto proxy servers, thus it can solve the secure data sharing problem in cloud. The experimental results show that LDSS can ensure data privacy in cloud and reduce the overhead on users' side in cloud.

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: 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), , Jun. 2011.

[4] Adam Skillen and Mohammad Mannan. On Implementing Deniable Storage Encryption for 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, Xiaohua Jia, Kui Ren: 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. 350-364

[10] Cong Wang, Kui Ren, Shucheng Yu, and Karthik Mahendra Raje Urs. Achieving Usable and Privacy-assured Similarity Search over Outsourced Cloud Data. IEEE INFOCOM 2012, Orlando, Florida, March 25-30, 2012.

[11] Yu S., Wang C., Ren K., Lou W. Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing. INFOCOM 2010, pp. 534-542, 2010

[12] Kan Yang, Xiaohua Jia, Kui Ren, Bo Zhang, Ruitao Xie: DAC-MACS: Effective Data Access Control for Multiauthority Cloud Storage Systems. IEEE Transactions on Information Forensics and Security, Vol. 8, No. 11, pp.1790-1801, 2013.

[13] Stehlé D, Steinfeld R. Faster fully homomorphic encryption. in: Proceedings of 16th International Conference on the Theory and Application of Cryptology and Information Security. Singapore: Springer press, pp.377-394, 2010.

[14] Junzuo Lai, Robert H. Deng ,Yingjiu Li ,et al. 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 (ASIACCS), pp. 239-248, Jun. 2014.

[15] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute based encryption. in: Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP). Washington, USA: IEEE Computer Society, pp. 321-334, 2007.

BIOGRAPHICAL NOTES

	Miss.DEEPA.G. Received M.C.A.,M.Phil,.,Degree in computer Science. She has 4 Years of Teaching Experience. She is Guided 7 Students in M.Phil.,She had Presented 1 Papers in International Conference. She is Currently Working has Associate Professor in Department of Computer Applications in Dhanalakshmi Srinivasan College of Arts & Science for Women,Perambalur,TamilNadu,India.
	Miss.R.DEEPALAKSHMI.,PG Scholar,[Department of Computer Applicatons],pursuing MCA in Dhanalakshmi Srinivasan College of Arts & Science for Women,Perambalur-621212,TamilNadu,(India)
	Miss.K,GNANASUNDARI..PG Scholar,[Department of Computer Applicatons],pursuing MCA in Dhanalakshmi Srinivasan College of Arts & Science for Women,Perambalur-621212,TamilNadu,(India)