World Research Journal of Engineering and Technology
ISSN: 2278-8530 & E-ISSN: 2278-8549, Volume 1, Issue 1, 2012, pp.-04-07.
Available online at http://www.bioinfo.in/contents.php?id=117

ACHIEVING SHELTERED, SCALABLE AND FINE-GRAINED DATA ACCESS CONTROL IN CLOUD COMPUTING

BHARTIYA A.S., AGRAWAL L.S., GAWANDE Y.V. AND RAPARTIWAR S.S.

Comp Sci and Engg, JDIET, Yavatmal.
*Corresponding Author: Email- sweetestannu.10@gmail.com, agrawal.latika@gmail.com, yashashree.gawande@gmail.com and rap_sagar@rediffmail.com

Received: March 06, 2012; Accepted: May 12, 2012

Abstract- To keep sensitive user data confidential against untrusted servers, existing solutions usually apply cryptographic methods by disclosing data decryption keys only to authorized users. However, in doing so, these solutions inevitably introduce a heavy computation overhead on the data owner for key distribution and data management when fine grained data access control is desired, and thus do not scale well. The problem of simultaneously achieving fine-grainedness scalability and data confidentiality of access control actually still remains unresolved. This paper proposed some services for data safekeeping and access control when users outsource sensitive data for sharing on cloud servers. This paper addresses this challenging open issue by, on one hand, defining and enforcing access policies based on data attributes, and, on the other hand, allowing the data owner to delegate most of the computation tasks involved in fine grained data access control to unfrosted cloud servers without disclosing the underlying data contents. Our proposed scheme enables the data owner to delegate tasks of data file re-encryption and user secret key update to cloud servers without disclosing data contents or user access privilege information. We achieve this goal by exploiting and uniquely combining techniques of attribute-based encryption (ABE), proxy re-encryption, and lazy re-encryption. Our proposed scheme also has salient properties of user access privilege confidentiality and user secret key accountability and achieves fine - graininess, scalability and data confidentiality for data access control in cloud computing. Extensive analysis shows that our proposed scheme is highly efficient and provably secures under existing security models .
Keywords- Security, ABE

Citation: Bhartiya A.S., et al. (2012) Achieving Sheltered, Scalable and Fine-Grained Data Access Control In Cloud Computing. World Research Journal of Engineering and Technology, ISSN: 2278-8530 & E-ISSN: 2278-8549, Volume 1, Issue 1, pp.-04-07.

Introduction

Cloud computing has many problems. Cloud computing is a promising computing paradigm which recently has drawn extensive attention from both academia and industry. By combining a set of existing and new techniques from research areas such as Service -Oriented Architectures and virtualization, cloud computing is regarded as such a computing paradigm in which resources in the computing infrastructure are provided as services over the Internet. Data security, as it exists in many other applications, is among these challenges that would raise great concerns from users when they store sensitive information on cloud servers. These concerns originate from the fact that cloud servers are usually operated by commercial providers which are very likely to be outside of the trusted domain of the users. Data confidential against cloud servers is hence frequently desired when users outsource data for storage in the cloud. In some practical application systems, data confidentiality is not only a security/privacy issue, but also of juristic concerns. Furthermore, we observe that there are also cases in which cloud users themselves are content providers. They publish data on cloud servers for sharing and need fine-grained data access control in terms of which user (data consumer) has the access privilege to which types of data. In the healthcare case, for example, a medical center would be the data owner who stores millions of healthcare records in the cloud. It would allow data consumers such as doctors, patients, researchers and etc., to access various types of healthcare records under policies admitted by HIPAA. To enforce these access policies, the data owners on one hand would like to take advantage of the

abundant resources that the cloud provides for efficiency and economy; on the other hand, they may want to keep the data contents confidential against cloud servers. We address this open issue and propose a secure and scalable fine-grained data access control scheme for cloud computing. Our proposed scheme is partially based on our observation that, in practical application scenarios each data file can be associated with a set of attributes which are meaningful in the context of interest. The access structure of each user can thus be defined as a unique logical expression over these attributes to reflect the scope of data files that the user is allowed to access. As the logical expression can represent any desired data file set, fine-graininess of data access control is achieved. To enforce these access structures, we define a public key component for each attribute. Data files are encrypted using public key components corresponding to their attributes. User secret keys are defined to reflect their access structures so that a user is able to decrypt a cipher text if and only if the data file attributes satisfy his access structure. Such a design also brings about the efficiency benefit, as compared to previous works, in that, 1) the complexity of encryption is just related the number of attributes associated to the data file, and is independent to the number of users in the system; and 2) data file creation/deletion and new user grant operations just affect current file/user without involving system-wide data file update or re-keying. One extremely challenging issue with this design is the implementation of user revocation, which would inevitably require re-encryption of data files accessible to the leaving user, and may need update of secret keys for all the remaining users. If all these tasks are performed by the data owner himself/herself, it would introduce a heavy computation overhead on him/her and may also require the data owner to be always online. To resolve this challenging issue, our proposed scheme enables the data owner to delegate tasks of data file re-encryption and user secret key update to cloud servers without disclosing data contents or user access privilege information. We achieve our design goals by exploiting a novel cryptographic primitive, namely key policy attribute-based encryption.

Literature Survey

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things r satisfied, ten next steps is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites.

System Study

Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During 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 feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

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 and 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.

Technical Feasibility- This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social Feasibility- The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

System Analysis

Existing System-Existing solution applies cryptographic methods by disclosing data decryption keys only to authorized users. These solutions inevitably introduce a heavy computation overhead on the data owner for key distribution and data management when fine grained data access control is desired, and thus do not scale well.

Proposed System-In order to achieve secure, scalable and fine-grained access control on outsourced data in the cloud, we utilize and uniquely combine the following three advanced cryptographic techniques.

- Key Policy Attribute-Based Encryption (KP-ABE).
- Proxy Re-Encryption (PRE)
- Lazy re-encryption

System Design

Data Flow Diagram / Use Case Diagram / Class Diagram / Sequence Diagram

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

Implementation

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

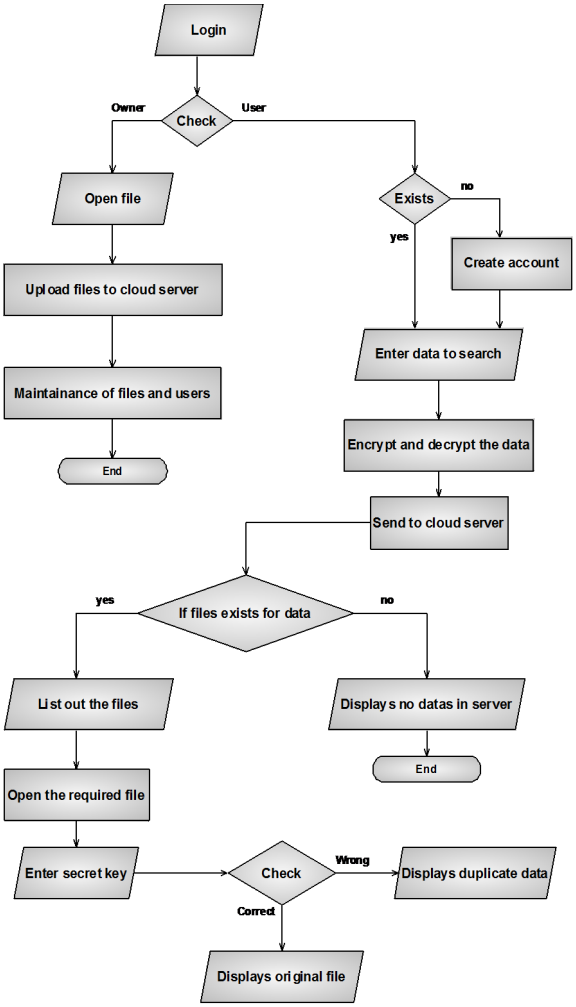

Fig. 1- Data Flow Diagram

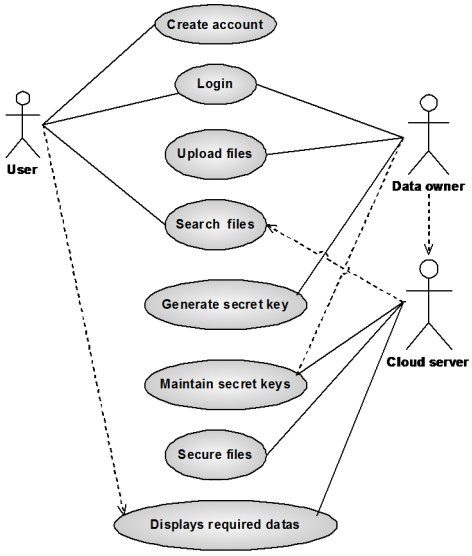

Fig. 2- Use Case Diagram

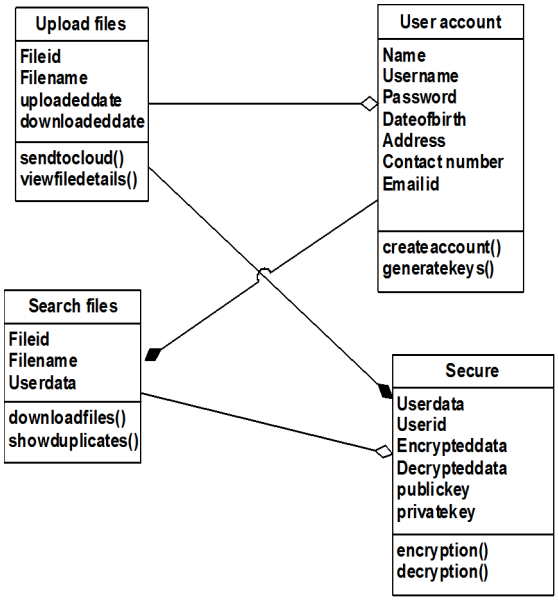

Fig. 3- Class Diagram

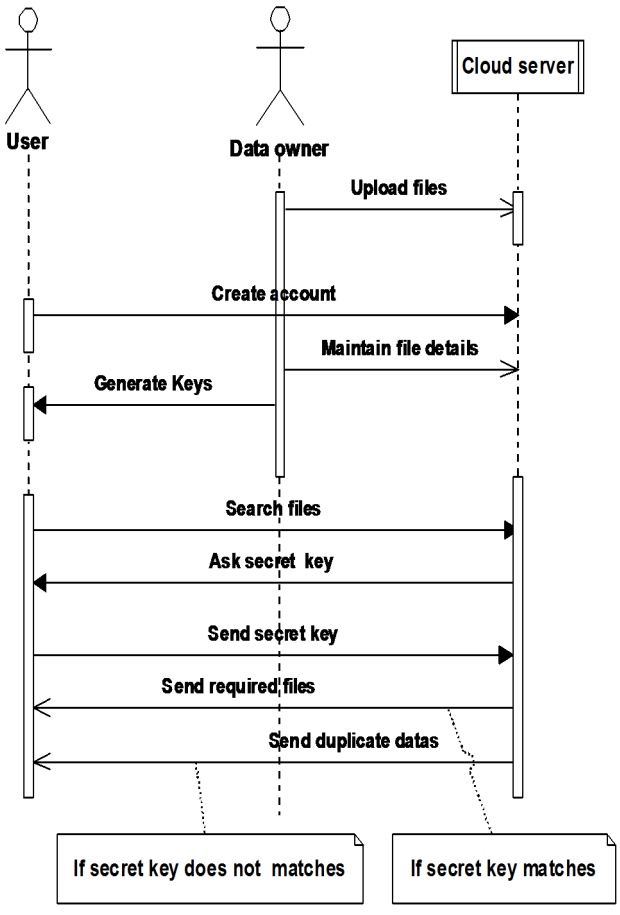

Fig. 4- Sequence Diagram

Main Modules

Module Description

Key Policy Attribute-Based Encryption (KP-ABE)

KP-ABE is a public key cryptography primitive for one-to-many communications. In KP-ABE, data are associated with attributes for each of which a public key component is defined. User secret

key is defined to reflect the access structure so that the user is able to decrypt a cipher text if and only if the data attributes satisfy his access structure. A KP-ABE scheme is composed of four algorithms which can be defined as follows:

- Setup Attributes
- Encryption
- Secret key generation
- Decryption

Setup Attributes-This algorithm is used to set attributes for users. From these attributes public key and master key for each user can be determined. The attributes, public key and master key are denoted as

Attributes- U = {1, 2. . . N}
Public key- PK = (Y, T1, T2, . . . , TN)
Master key- MK = (y, t1, t2, . . . , tN)

Encryption-This algorithm takes a message M, the public key PK, and a set of attributes I as input. It outputs the cipher text E with the following format-

$$E = (I, \tilde{} E, \{Ei\}i)$$

where ˜E = MY, Ei = Ti.

Secret Key Generation

This algorithm takes as input an access tree T, the master key MK, and the public key PK. It outputs a user secret key SK as follows.

$$SK = \{ski\}$$

Decryption

This algorithm takes as input the cipher text E encrypted under the attribute set U, the user's secret key SK for access tree T, and the public key PK.

Finally it output the message M if and only if U satisfies T.

Proxy Re-Encryption (PRE)

Proxy Re-Encryption (PRE) is a cryptographic primitive in which a semi-trusted proxy is able to convert a cipher text encrypted under Alice's public key into another cipher text that can be opened by Bob's private key without seeing the underlying plaintext. A PRE scheme allows the proxy, given the proxy re-encryption key

$$rka\leftrightarrow b,$$

to translate cipher texts under public key pk1 into cipher texts under public key pk2 and vise versa.

Lazy Re-Encryption:

The lazy re-encryption technique and allow Cloud Servers to aggregate computation tasks of multiple operations. The operations such as

- Update secret keys
- Update user attributes.

System Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

System Specification

System Requirements

Hardware Requirements

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 Mb.

Software Requirements

- Operating system : - Windows XP.
- Coding Language : DOT NET
- Data Base : SQL Server 2005

Conclusion

This paper aims at fine-grained data access control in cloud computing. One challenge in this context is to achieve finegrainedness, data confidentiality, and scalability simultaneously, which is not provided by current work. In this paper we propose a scheme to achieve this goal by exploiting KPABE and uniquely combining it with techniques of proxy re-encryption and lazy re-encryption. Moreover, our proposed scheme can enable the data owner to delegate most of computation overhead to powerful cloud servers. Confidentiality of user access privilege and user secret key accountability can be achieved. Formal security proofs show that our proposed scheme is secure under standard cryptographic models.

References

[1] Kallahalla M., Riedel E., Swami Nathan R., Wang Q. and Fu K. (2003) *Proc. of FAST* 03.
[2] Armbrust M., Fox A., Griffith R., Joseph A. D., Katz R.H., Konwinski A., Lee G., Patterson D.A., Rabkin A., Stoica I. and Zaharia M. (2009) *University of California, Berkeley*, Tech. Rep. USB-EECS.
[3] Goyal V., Pandey O., Sahai A. and Waters B. (2006) *Proc. ofCCS*.
[4] Wang Q., Wang C., Li J., Ren K. and Lou W. (2009) *Proc. of ESORICS*.
[5] Naor D., Naor M. and Lotspiech J.B. (2001) *Proc. of CRYPTO*.
[6] Atallah M., Frikken K. and Blanton M. (2005) *Proc. of CCS*.
[7] Li J., Li N. and Winsborough W.H. (2005) *Proc. of CCS*.