IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

Cloud Data Access with Multi-authorities

Aniket Salunke[1], Arjun Vekariya[2], Siddhesh Vartak[3], Sachin Sonawane[4]

Department of Computer Engineering, Atharva College of Engineering, Mumbai University, Mumbai, India[1,2,3]

Assistant Professor, Atharva College of Engineering, Mumbai University, Mumbai, India[4]

Abstract: Cloud computing is a revolutionary computing paradigm which enables flexible, on-demand and low-cost usage of computing resources. Those advantages, ironically, are the causes of security and privacy problems, which emerge because the data owned by different users are stored in some cloud servers instead of under their own control. To deal with security problems, various schemes based on the Attribute-Based Encryption have been proposed recently. However, the privacy problem of cloud computing is yet to be solved. This paper presents an anonymous privilege control scheme Anony Control to address not only the data privacy problem in a cloud storage, but also the user identity privacy issues in existing access control schemes By using multiple authorities in cloud computing system, our proposed scheme achieves anonymous cloud data access and fine-grained privilege control. Our security proof and performance analysis shows that Anony Control is both secure and efficient for cloud computing environment.

Keywords: Cloud computing, security and privacy, cloud data access.

I. INTRODUCTION

Cloud computing is a new concept of computing technique, by which computer resources are provided dynamically via Internet. It attracts considerable attention and interest from both academia and industry. However, it also has at least three challenges that must be handled before applied to our real life. First of all, data confidentiality should be guaranteed. When sensitive information is stored in cloud servers, which is out of users' control in most cases, risks would rise dramatically. The servers might illegally inspect users' data and access sensitive information. On the other hand, unauthorized users may also be able to intercept someone's data (e.g. server compromise). Secondly, personal information (defined by a user's attributes) is at risk because one's identity is authenticated according to his information. In fact, various techniques have been proposed and/or used to address the aforementioned problems.

Identity-based encryption (IBE) was first introduced by Shamir in 1985. In the IBE, the sender of a message can specify an identity such that only a receiver with matching identity can decrypt it. This is different from Public-key Encryption, in that the encrypter does not need to issue extra key to decrypter for each cipher text. In the IBE, the private key, which contains the identity of the holder, is distributed to every user only once when he joins the system. In fact, various techniques have been proposed and/or used to address the aforementioned problems. Identity-based encryption (IBE) was first introduced by Shamir in 1985 [1]. In the IBE, the sender of a message can specify an identity and only a receiver with matching identity can decrypt it. This is not Public-key Encryption,it is very much different from the public key encryption process.in this process the encrypter does not have to provide extra key to decrypter for each ciphertext for decryption process. The identity of the holder which is stored in the private key is distributed to every user only one time when user joins the system for the first time. Few years later, Sahai and Waters introduced a new type of Fuzzy Identity-Based Encryption [2] which is known as Attribute-Based Encryption(ABE). More general ABE schemes, Key-Policy Attribute-Based Encryption (KP-ABE) [3] and Ciphertext-Policy Attribute Based Encryption (CP-ABE) [4], are proposed by Goyalet al. and Bethencourtet al. In the KP-ABE [3], a ciphertext is associated with a set of attributes, which partially Represents the cipher text's encryption policy. If the access tree in his private key is satisfied by the attributes in the cipher text then only user can decrypt the cipher text.

The main contributions of this paper are:

1. The proposed scheme is able to protect user's privacy against each single authority.
2. The proposed scheme is tolerant against authority compromise, and compromising of up to (N −2) authorities does not bring the whole system down.
3. We provide detailed analysis on security and performance to show feasibility of our scheme.
4. We first implement the real toolkit of multi-authority based encryption scheme.

II. DEFINITIONS OF OUR SCHEME

A. System Model:

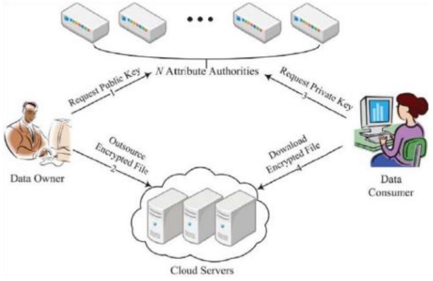

Fig. 1.Our system model

Powerful computation abilities are assumed to have by Authorities, who are supervised by government offices since keys act as IDs and partially contain users' PII (Personally Identifiable Information). Each authority can control the whole attribute set which is divided into N disjoint sets. One of the practical method which can be used to divide the attributes set is to divide them by category (e.g. Sex, Nationality, Designation, University etc. Only one type of attribute is aware in each authority in this way no useful information is leaked. Master keys of individual are compute at the initialization phase and the authorities jointly compute a system-wide public key. For all operations within the system the public key is used for computation and the master keys are used by attributes when the private is generated for data consumers.

B. Threats Model:

We assume the Cloud Servers are untrusted, who behave properly in most of time but may collude with malicious Data Consumers or Data Owners to harvest others' file contents to gain illegal profits but they should also gain. legal benefit when users' requests are correctly processed. The meaning of this is they will follow the protocol in general. In addition to that even if the Cloud Server illegally modifies data files for sake of monetary benefits (eg. Saving storage by deleting rarely access files) whether the data is intact can be detected by the TPA technique introduced in [5].

The N authorities in our system are assumed to be semi-honest. That is, they will follow our proposed protocol in general, but they will also try to find out as much information as possible individually. More specifically, we assume can that they are interested in user's attributes to achieve the identities, but they will never collude authority or with any user to harvest file contents even if it is highly beneficial. This assumption is similar to many previous researches on security issue in cloud computing (e.g. [6], [5]–[7]), and it is also reasonable since these authorities will be audited by government offices.3

Data Consumers can not be trusted as they are untrustful as they are random users including the attackers. They may collude with other Data Consumers to access what is not allowed for them.

C. Design Goal:

Our primary goal is to help Data Owners to share their private data with Data Consumers secretly and securely, where fine-grained privilege control is achievable, and to provide the guarantee the confidentiality of Data Consumers' identity information by decomposing a center authority to multiple ones while preserving tolerance to compromise attacks on the authorities. We assume that the identity of the information is not disclosed by the underlying network. This can be achieved by employing anonymized protocols (e.g.,[8]). In the rest of this paper, A^u is used to denote the attributes set of a user u. Akis used to denote the attribute authority k, and we also use a subscript k to denote the attributes set handled by Ak.

III. SECURITY ANALYSIS

A. User's Identity Information Confidentiality:

As our primary goal is to provide the security and confidentiality the attributes, which contain a user's identity and any other information, are separately controlled by different attribute authorities. Therefore, a user's attributes information is securely protected.

B. Trade-off between Tolerance and Complexity:

In the proposed scheme, an authority Akgenerates a set of random secret parameters $\{s_{kj}\}$ and shares it with other authorities, and the $x_k$is computed based on this parameters. Even if an adversary is able to compromise up to (N − 2) authorities, there are still.

Two parameters kept unknown to the adversary. So, the adversary is not able to guess the valid and he fails to construct a valid secret key. Hence, the scheme achieves compromise tolerance to up to (N − 2) authorities compromise. But, if we reduce the time complexity of the setup phase by dividing authorities into several clusters having C authorities in each, attackers can compromise C−1 authorities in a cluster to create valid master keys of that cluster.

Hence, there is a trade-off in between tolerance and complexity. However, because the number of authorities are typically not very huge in numbers, and the setup is one-time operation at the beginning of the system setup, we recommend using the original setup algorithm whose complexity is $O(N^2)$.

Finally, note that the compromised authorities are able to issue valid attribute keys for which they are in charge of, so the cipher texts whose privilege trees have only those attributes might be illegally decrypted if the attacker issue all possible attribute keys to himself. But, since the authorities are well protected servers, it is hard to compromise even one authority, and the probability of compromising enough authorities to illegally decrypt some cipher text is very low.

IV. IMPLEMENTATION

In this section, we give the experimental result of our scheme, which is conducted on the prototype of our scheme. To the best of our knowledge, this is the best possible implementation of a multi-authority attribute based encryption scheme.

Our prototype system provides five command line tools.
anonyabe-setup: Jointly generates a public key and N master keys.

anonyabe-keygen: Generates a part of private key for the possible sets of attributes.
anonyabe-enc: Encrypts a file under r privilege trees.
anonyabe-dec: Decrypts a file if possible.
anonyabe-rec : Re-encrypts a file under other privilege trees.

DOI 10.17148/IJARCCE.2016.5429

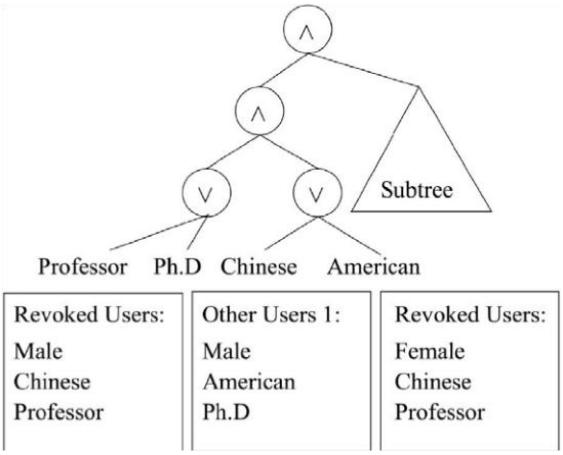

Fig. 2. An example of privilege tree after the re-encryption

This toolkit is based on the CP-ABE toolkit [4] which is in turn based on PBC library, and the whole system is implemented on a linux system with Intel i7 2nd Gen @ 2.7GHz and 2GB RAM.

V. CONCLUSION

This paper proposed privilege control of an anonymous attribute-based scheme Anony Control to address the problem of user privacy in a cloud storage server.

By using multiple authorities in cloud computing system, our proposed scheme achieves fine-grained privilege control, as well as the anonymity while conducting privilege control based on users' identity information. More importantly, this system can tolerate up to $N - 2$ authority compromise, which is very good option and highly preferable especially in such type of computing environment where Internet-based cloud computing environment is used.

Furthermore, although the data contents are fully outsourced to Cloud Servers, the Cloud Servers cannot read the contents unless their private keys satisfy the privilege tree T_0. We also conducted detailed performance and security analysis which shows that Anony Control is both secure and efficient for cloud storage system environment.

VI. ACKNOWLEDGEMENT

We would like to thank our project guide Prof. Sachin Sona wane sir for his co-operation and guidance. We express our gratitude to him for wholeheartedly supporting and giving his precious time for providing technical guidance for writing this paper.We are also thankful to our institute Atharva College of Engineering and Shri Sunil Rane sir the Executive President of Atharva Group of Institutes for providing us with the necessary infrastructural facilities.

We are also grateful to our principal Dr. S.P.Kallurkar sir and our project coordinators for providing us with the necessary guidelines for presenting our paper in this conference. Finally we would like to thank our friends and other technical staff of our branch who helped us for writing this paper.

REFERENCES

[1]. A. Shamir, "Identity-based cryptosystems and signature schemes," in Advances in cryptology, Springer, 1985, pp. 47–53.

[2]. A. Sahai and B. Waters, "Fuzzy identity-based encryption," Advances in Cryptology–EUROCRYPT 2005, pp. 557–557, 2005.

[3]. V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proceedings of the 13th ACM conference on Computer and communications security, 2006, pp. 89–98.

[4]. J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute based encryption," in IEEE Symposium on Security and Privacy, 2007, pp. 321–334.

[5]. C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in IEEE INFOCOM, 2010. pp. 1 – 9.

[6]. J. Liu, Z. Wan, and M. Gu, "Hierarchical attribute-set based encryption for scalable, flexible and fine-grained access control in cloud computing," Information Security Practice and Experience, pp. 98–107, 2011.

[7]. C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in IEEE ICDCS, 2010, pp. 253–26.