An Efficient Scheme For Cloud Services Based On Access Policies

V.Suma #1,K.Vijay Kumar#2,

#1 Student, PVP Siddhartha Institute of Technology,Kanuru,Vijayawada, Krishna(dt),

#2 Assistant. Professor, PVP Siddhartha Institute of Technology,Kanuru,Vijayawada, Krishna(dt),

Abstract: Recent years have witnessed the trend of leveraging cloud based services for large scale content storage, processing, and distribution. Cloud computing, as an emerging computing paradigm, enables users to remotely store their data into a cloud so as to enjoy scalable services on-demand. Access control is one of the most important security mechanisms in cloud computing. Traditional access control models often assume that the entity enforcing access control policies is also the owner of data and resources. This assumption no longer holds when data is outsourced to a third-party storage provider, such as the cloud. Existing access control solutions mainly focus on preserving confidentiality of stored data from unauthorized access and the storage provider. Attribute-based access control provides a flexible approach that allows data owners to integrate data access policies within the encrypted data. In this paper, we implement an efficient temporal access control encryption scheme for cloud services with the help of cryptographic integer comparisons and a proxy-based re-encryption mechanism on the current time. We also use a dual comparative expression of integer ranges to extend the power of attribute expression for implementing various temporal constraints.

I INTRODUCTION

Cloud computing as one of current most exciting technology areas, denotes an architectural shift towards thin clients and scalably centralized provision of computing and storage resources on-demand. By combining emerging techniques, such as virtualization and service-oriented computing, three types of servers are available in a pay-as-you-go manner, i.e., infrastructure as a service (IaaS), where users make use of a cloud service provider's (CSP's) computing, storage, and networking infrastructure to deploy any arbitrary software, e.g., Amazon EC2; platform as a service (PaaS), where users deploy user-created or acquired applications written with programming languages and tools supported by CSPs, e.g., Microsoft Windows Azure; and software as a service (SaaS), where users make use of CSPs' software running on a cloud infrastructure, e.g, Google Docs.

Cloud computing provides an extensible and powerful environment for growing amounts of services and data by means of on-demand self-service. It also relieves the client's burden from management and maintenance by providing a comparably low-cost, scalable, location-independent platform. However, cloud computing is also facing many challenges for data security as the users outsource their sensitive data to clouds, which are generally beyond the same trusted domain as data owners. To address this problem, access control is considered as one of critical security mechanisms for data protection in cloud applications. Unfortunately, traditional data access control schemes usually assume that data is stored on trusted data servers for all users.

Later, attribute-based access control has been introduced into cloud computing to encrypt outsourced sensitive data in terms of access policy on attributes describing the outsourced data, and only authorized users can decrypt and access the data. Since the access control policy of every object is embedded within it, the enforcement of policy becomes an inseparable characteristic of the data itself. This is in direct contrast to most currently

available access control systems, which rely directly upon a trusted host to mediate access and maintain policies.

However, existing attribute-based solutions are difficult to provide full features of temporal data access control due to following reasons:

• The system models of existing systems cannot support dual comparative expressions (DTE), in which two range-based comparative constraints must be embedded into the outsourced files as well as the user's private key.

• The existing systems don't support current time, which is essentially an important factor for enforcing temporal access control.

In this paper, we implement a temporal access control solution along with a proxy-based re-encryption mechanism to address above mentioned problems for cloud computing. The proposed scheme is originated from the needs of practical cloud applications, in which each outsourced resource can be associated with an access policy on a set of temporal attributes, e.g., period-of-validity, opening hours, or hours of service. Each user can also be assigned a license with several privileges based on the comparative attributes. To enforce the valid matches between access policies and user's privileges, we introduce a proxy-based re-encryption mechanism with respect to the current time. This design brings about several efficient benefits, such as flexibility, supervisory, and privacy protection, compared with prior work.

II RELATED WORK

In a traditional public key cryptosystem (PKC), each user has a public/private key pair, and messages encrypted with a recipient's public key can only be decrypted with the corresponding private key. When a sender wants to encrypt a file to n recipients, he first obtains the authenticated public keys of all the recipients, and then encrypts the file using each recipient's public key, respectively. Finally, the n copies of the corresponding ciphertexts are stored in a cloud. Therefore, both the computational cost for encryption, and the length of the ciphertexts, are proportional to the total number of intended recipients. Obviously, it should not be applied directly while sharing data on cloud servers, since they are inefficient to encrypt a file to multiple recipients, and fail to support attribute- based access control and key delegation.

Fiat et al first introduced the concept of the broadcast encryption (BE), in which a broadcaster encrypts a message for some subset S of users who are listening on a broadcast channel, so that only the recipients in S can use their private keys to decrypt the message. The first proposal of a BE system is secure against a collusion of k users, which means that such a scheme may be insecure if more than k users collude. In a BE system, there are only two parties: a broadcaster and multiple users, where the broadcaster generates the secret keys for all the users, and can broadcast an encrypted message to some subset of the users. Obviously, a BE system achieves "one-to-many encryption" with general performance, however, it may not applied directly while sharing data on cloud servers, since it fails to support attribute-based access control and key delegation.

Shamir proposed the idea of identity-based cryptography, but a fully functional identity- based encryption (IBE) scheme was not found until recent work by Boneh et al and Cocks. An IBE scheme is a PKC, where any arbitrary string corresponding to a unique user information is a valid public key. The corresponding private key is computed by a trusted third party (TTP) called the private key generator (PKG). Compared with the traditional PKC, the IBE system eliminates online look-ups for the recipient's authenticated public key, but introduces the key escrow problem.

In an IBE system, there is only one PKG to distribute private keys to each user, which is undesirable for a large network because the PKG has a burdensome job. Horwitz et al, dedicated to reducing the workload on the root PKGs, introduced the concept of a HIBE system. They constructed a concrete two-level HIBE scheme, in which a root

PKG needed only to generate private keys for domain-level PKGs that, in turn generated private keys for all the users in their domains at the next level. Their scheme, with total collusion resistance on the upper level and partial collusion resistance on the lower level, has chosen ciphertext security in the random oracle model.

In recent work, Gentry et al proposed a fully secure HIBE scheme by using identity-based broadcast encryption with key randomization; Waters achieved full security in systems under simple assumption by using dual system encryption. Among others, by making use of the "valuable" property of the G-HIBE scheme, Liu et al proposed an efficient sharing of the secure cloud storage services (ESC) scheme, where a sender can specify several users as the recipients for an encrypted file by taking the number and public keys of the recipients as inputs of a HIBE system. The limitation of their scheme is that the length of ciphertexts grows linearly with the number of recipients, so that it can only be used in the case that a confidential file involves a small set of recipients. The HIBE system naturally achieves key delegation, and some HIBE schemes achieve "one-to-many encryption" with general performance, however, it may not be applied directly while sharing data on cloud servers, since it fails to support an attribute-based access control.

III BASIC DEFINITIONS

A) Goal:

Our main design goal is to help the data owner achieve temporal data access control on files stored in cloud servers. Although this kind of access control is based on finegrained access control introduced for outsourced data services, we intent to ensure that all kind of temporal access policy can be securely and efficiently implemented for outsourced data services.

B) System Model

Considering a cloud-based data storage service involving three different entities, as illustrated in Fig. 1: data owner, cloud server, and many data users (e.g., computers, mobile devices, or general equipments). In addition, in order to implement temporal access control, we require a clock server designed to always provide exactly the same current time by communicating with each other.

Basic to ensure the data access compliant with the assigned policy, fine-grained access control has been introduced into the outsourced storage service. We extend this kind of access control mechanisms to support temporal access control encryption (TACE) described as follows:

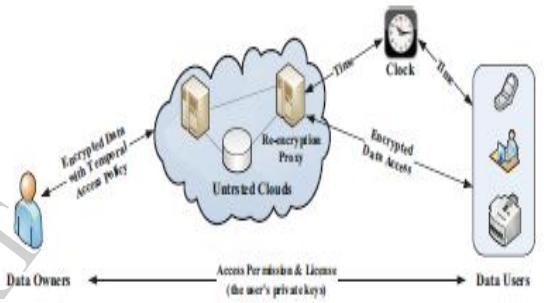

Fig 1: Temporal Constraints

• First, the data owner makes use of a temporal access policy *P* to encrypt data before store it to clouds.

• Second, once receiving an access request from a user, the cloud service checks whether corresponding temporal constraints can be satisfied in *P* with respect to the current time *tc*, then employs a re-encryption method to convert the encrypted data into another ciphertext *Ctc* that embed current time *tc* and sent it the user.

• Finally, the authorized user can use her/his private key *SK* with access privilege *L* to decrypt *Ctc* . In this model, we assume the cloud service is a semi-trusted service that can use the correct time to re-encrypt data.

C) Notations

For sake of clarity, we introduce following notations:

• *A*: the set of attributes *A* = *{A*1, ... *,Am}*;

• *Ak*(*ti, tj*): the range constraint of attribute *Ak* on [*ti, tj*], i.e., *ti* ≤ *Ak* ≤ *tj* ;

• *P*: the access control policy expressed as a Boolean function on AND/OR logical operations, generated by the grammar: *P* ::= *Ak*(*ti, tj*)|*P* AND *P*|*P* OR *P*;

• *L*: the access privilege assigned into the user's licence, generated by *L* ::= *{Ak*(*ta, tb*)*}Ak€A.*

The definitions of *P* and *C* can meet the basic requirements of dual temporal expressions.

IV TACE Framework and Model

With focusing on temporal access control and reencryption mechanism in cloud computing, the TACE scheme consists of:

1) Setup(1*,*A*): Takes a security parameter *κ* and a list of attributes *A* as input, outputs the master key *MK* and the public-key *PKA*;

2) GenKey(*MK, uk,L*): Takes the user's ID number *uk* as input, the access privilege *L* and *MK*, outputs the user's private key *SKL*;

3) Encrypt(*PKA,P*): Takes a temporal access policy *P* and *PKA* as input, outputs the ciphertext header *HP* and a random session key *ek*;

4) ReEncrypt(*PKA,HP, tc*): Takes a current time *tc* and a ciphertext header *HP* and *PKA* as input, outputs a new ciphertext header *Htc* ;

5) Decrypt(*SKL,Htc*): Takes a user's private key *SKL*, and a ciphertext header *Htc* on the current time *tc* as input, outputs a session key *ek*;

First, given a scheme based on our TACE framework, we must guarantee that this scheme can follow the principle in secure temporal control: Let *Ak€ A* be a range-based temporal attribute and (*P,L*) be a constraint-privilege pair with *Ak*, where *Ak*[*ti, tj*] € *P* and *Ak*[*ta, tb*] € *L*. Given a current time *tc*, secure temporal control requires that the access is granted if and only if *tc € [ti, tj*] and *tc € [ta, tb*]. This means that the TACE scheme can must also obey this rule as follows: Given the above-mentioned (*P,L*), we can compute (*MK, PKA*) ← *Setup*(1_,*A*), *SKL* ← *GenKey*(*MK, uk,L*), and (*HP, ek*) ← *Encrypt*(*PK,P*). Such that, we hold Pr [*Hc* ← *ReEncrypt*(*PKA,HP, tc*); *Decrypt*(*SKL,Htc*) = *ek*] = 1, if and only if the access is granted over (*P,L*) and *tc* according to fine-grained access control model.

V PERFORMANCE

Flexibility: TACE-based cryptosystem can provide more flexible access control based on temporal constraints as follows: a) Date control on Year, Month, and Day.

Supervisory: Traditional cryptosystems, that only contains both encryption and decryption processes, has not an efficient method to monitor the usage of encrypted data. TACEbased cryptosystem introduces a proxy-based re-encryption mechanism that can apply the current time to determine whether the user's download request is reasonable, and rely on the re-encryption technologies to produce a new version of data under the current time. Such a proxy service can also integrate with other rich information to determine the legitimacy of user behaviors.

Privacy Protection: In our system model, the access policies are enforced entirely dependent upon temporal attribute matches between ciphertexts and private keys in the client side. In the re-encryption process, cloud servers do not require any user information which is used to enforce access policies. Hence, this mechanism ensures that user privacy, including user identity and access privilege in the user's private key, will not be disclosed to cloud servers.

VI CONCLUSION

Cloud computing is one of the current most important and promising technologies. For the sake of enjoying a more comprehensive and high-quality service, we proposed scheme which simultaneously achieves flexibility, high performance, and full key delegtaion. In recent years, cryptographic access control has been introduced as a new access control paradigm to manage dynamic data sharing systems in cloud computing. Attribute-based encryption (ABE) is proposed in 2005 to realize a fine-grained ttribute-based access control mechanism. In this paper, we use the temporal access control in cloud computing. Based on a temporal access control encryption to support time range comparisons and re-encryption mechanism to handle current time controls and temporal constraints.

VII REFERENCES

[1] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *ACM Conference on Computer and Communications Security*, 2007, pp. 195–203.

[2] Y. Zhu, H. Hu, G.-J. Ahn, M. Yu, and H. Zhao, "Comparison-based encryption for fine-grained access control in clouds," in *CODASPY, ACM*, 2012, To appear.

[3] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *INFOCOM, IEEE*, 2010, pp. 534–542.

[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *CCS, ACM*, 2006, pp. 89–98.

[5] E. Bertino, P. A. Bonatti, and E. Ferrari, "TRBAC: A temporal rolebased access control model," *ACM Trans. Inf. Syst. Secur.*, vol. 4, no. 3, pp. 191–233, 2001.

[6] J. B. Joshi, E. Bertino, U. Latif, and A. Ghafoor, "A generalized temporal role-based access control model," *IEEE Transactions on*
Knowledge and Data Engineering, vol. 17, pp. 4–23, 2005.

[7] A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In Proceedings of CRYPTO 1984, volume 196 of LNCS, pages 47-53.

[8] C. Gentry and S. Halevi. Hierarchical Identity Based Encryption with Polynomially Many Levels. In Proceedings of TCC 2009, volume 5444 of LNCS, pages 437-456.

[9] B. Waters. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In Proceedings of CRYPTO 2009, volume 5677 of LNCS, pages 619-636.

[10] D. Boneh and X. Boyen. Efficient Selective-ID Secure Identity Based Encryption without Random Oracles. In Proceedings of EUROCRYPT 2004, volume 3027 of LNCS, pages 223-238.

[11] D. Boneh, X. Boyen, and E. Goh. Hierarchical Identity Based Encryption with Constant Size Ciphertext.In Proceedings of EUROCRYPT 2005, volume 3494 of LNCS, pages 440-456.