# Fine Grained Attribute based Encryption and Decryption in Cloud

**[1]Sirisha Potluri, [2]P.Sree Lakshmi,**

**[1]Visiting Faculty, CSE Dept, IFHE, FST, Shankarpally,Medak Dist, Telangana, sirisha.vegunta@gmail.com**

**[2]Asst Professor, CSE Dept, BVRIT,Narsapur,Medak Dist, Telangana.**

**Abstract:** Cloud computing is one of the most emerging technologies in the world. Security is also very important in the cloud computing. Though there are no of security issues are there in cloud still there is lack of security of the data in the cloud. In this paper, Attribute-based access control provides a flexible approach that allows data owners to integrate data access policies within the encrypted data. In this paper, proposed system focus on effective encryption and decryption of the data is also implemented. 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, signifies a design shift towards thin clients and scalably provision of computing and storage resources on-interest. By consolidating rising methods, for example, virtualization and service-oriented computing, three types of servers are accessible in a pay-as-you-go way, i.e., infrastructure as a service (IaaS), where clients make utilization 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 clients send client made or procured applications composed with programming languages and tools supported by CSPs, e.g., Microsoft Windows Azure; and software as a service (SaaS), where clients make utilization of CSPs' software running on a cloud framework, e.g, Google Docs [1].

Cloud computing gives an extensible and intense 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 direct data 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, quality based access control has been acquainted into cloud computing with encrypt outsourced sensitive information as far as access policy on attributes describing the outsourced data, and only authorized users can decrypt and access the data. Since the data access control approach of each item is inserted inside of it, the authorization of arrangement turns into an indistinguishable normal for the information itself. This is in immediate differentiation 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 frameworks can't dual comparative expressions (DTE), in which two range based similar requirements must be inserted into the outsourced records and additionally the client's private key.

• The existing system doesn’t support current time, which is basically a critical element for implementing 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 [2]. 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

Public key cryptosystem (PKC), every client has an open/private key match, and messages encoded with a beneficiary's public key must be decoded with the relating private key. At the point when a sender needs to encrypt a file to n beneficiaries, he first acquires the validated public keys of the considerable number of beneficiaries, and after that encrypts the file utilizing every beneficiary's public key, separately. At last, the n duplicates of the relating ciphertexts are stored in a cloud. Hence, both the computational expense for encryption, and the length of the ciphertexts, is relative to the aggregate number of proposed beneficiaries [3]. Clearly, it should not be connected specifically while sharing information on cloud servers, since they are inefficient to encode a file to various beneficiaries, and fail to support attribute- based access control and key delegation.

Fiat first presented the idea of the broadcast encryption (BE), in which a broadcaster encodes a message for some subset S of clients who are listening on a telecast station, so that just the beneficiaries in S can utilize their private keys to decrypt the message. The first proposition of a BE system is secure against an intrigue of k clients, which implies that such a plan may be frail if more than k clients plot. In a BE system, there are just two gatherings: a telecaster and numerous clients, where the supporter produces the mystery keys for all the clients, and can broadcast a encrypted message to some subset of the clients. Clearly, a BE system accomplishes "one-to-numerous encryption" with general execution, then again, it may not connected specifically while sharing information 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 [4].

In an IBE system, there is stand out PKG to disseminate private keys to every client, which is undesirable for a substantial system on the grounds that the PKG has an oppressive occupation. Horwitz et al, committed to lessening the workload on the root PKGs, presented the idea of a HIBE framework. They developed a solid two-level HIBE plan, in which a root PKG required just to produce private keys for space level PKGs that, thusly produced private keys for all the clients in their areas at the following level. Their plan, with aggregate plot resistance on the upper level and incomplete conspiracy resistance on the lower level, has picked ciphertext security in the irregular prophet model [5].

In recent work, Gentry proposed a completely secure HIBE conspire by utilizing character based telecast encryption with key randomization; Waters accomplished full security in systems under straightforward suspicion by utilizing double system encryption. Among others, by making utilization of the "important" property of the G-HIBE plan, Liu proposed an efficient sharing of the protected cloud storage services (ESC) plan, where a sender can indicate a few clients as the beneficiaries for a scrambled file by taking the number and open keys of the beneficiaries as inputs of a HIBE framework. The restriction of their plan is that the length of figure writings becomes straightly with the quantity of beneficiaries, so it must be utilized as a part of the case that a confidential file includes a little arrangement of beneficiaries [6]. The HIBE framework normally accomplishes key appointment, and some HIBE plans accomplish "one- to-numerous encryption" with general execution, then again, 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:

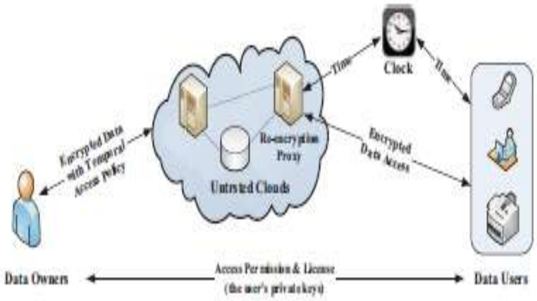
Clock
Time
Time
Re-encryption Proxy
Untrsted Clouds
Encrypted Data with Temporal Access policy
Encrypted Data Access
Data Owners
Access Per mission & License (the user's private keys)
Data Users

**Fig 1: Proposed Architecture**

• 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 [7].

• 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 [8].

## IV TACE Framework and Model

With focusing on temporal access control and re encryption 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

**Resilience:** TACE-based cryptosystem can provide more flexible access control based on temporal constraints as follows: a) Date control on Year, Month, and Day [9].

**Managerial:** Traditional cryptosystems, that only contains both encryption and decryption processes, has not an efficient method to monitor the usage of encrypted data. TACE based 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 [10].

**Security:** In our system model, the access policies are enforced entirely dependent upon temporal attribute matches between cipher texts 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 [11].

## VI CONCLUSION

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 attribute-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.