**DUAL ACCESS CONTROL FOR CLOUD-BASED DATA STORAGE AND SHARING**

**ABSTRACT:**

Cloud-based data storage service has drawn increasing interests from both academic and industry in the recent years due to its efficient and low cost management. Since it provides services in an open network, it is urgent for service providers to make use of secure data storage and sharing mechanism to ensure data confidentiality and service user privacy. To protect sensitive data from being compromised, the most widely used method is encryption. However, simply encrypting data (e.g., via AES) cannot fully address the practical need of data management. Besides, an effective access control over download request also needs to be considered so that Economic Denial of Sustainability (EDoS) attacks cannot be launched to hinder users from enjoying service. In this paper, we consider the dual access control, in the context of cloud-based storage, in the sense that we design a control mechanism over both data access and download request without loss of security and efficiency. Two dual access control systems are designed in this paper, where each of them is for a distinct designed setting. The security and experimental analysis for the systems are also presented.

**EXISTING SYSTEM**

* The first work with consideration of user personal privacy was introduced by Nishide *et al.* [8], where the access policy was partially hidden by dividing attribute into two parts as value and name, while only hiding the value. Due to the hidden policy, the adversary cannot get any information about the users. However, their scheme is impractical since its computation cost is too high. In 2009, Waters proposed a CP-ABE scheme with dual system encryption technique [7]. It provided a new way for privacy preserving in CP-ABE.
* Then Lai *et al.* [9], [10] used this technique to issue two hidden access policy CP-ABE schemes (HP-CP-ABE). Both of them have been proven to achieve full security. The first one [9] only supports AND gate, and the second one [10] supports linear secret share scheme (LSSS) [11], which is a more expressive access structure. However, the size of both secret keys and ciphertext increases linearly with the number of attributes.
* Then Rao *et al.* [12] introduced another HP-CP-ABE scheme with full security. In this scheme, its security also relies on composite-order group, but the size of secret keys and ciphertext achieves constant which improves the efficiency compared with [9] and [10]. However, this scheme only supports AND gate, which is not expressive. Zhang *et al.* [13] proposed a hierarchical HP-CP-ABE scheme, where they used the technique proposed by Abdalla *et al.* [14]. It achieves constant size secret keys and supplies fast decryption.
* To address this problem, Zhang *et al.* [16] introduced an HP-CP-ABE scheme with authority verification phase to decrease users’ computational consumption. The authority verification phase can help users check whether they are the valid users or not. However, privacy leakage is found in the match phase.

**DISADVANTAGES**

* + In the existing work, system is either coarse grained or short of scalability as the number of users increases.
  + The existing doesn’t use 256 or 512 bit encrypted keys.

**PROPOSED WORK**

Identity key ,We proposed identity key verification technique for dual access control in which every user will receive unique identity key at the time of registration, if he wants to download any document he have to submit his identity key. System will send that identity key to key manager server automatically; key manager server will verify user’s identity with access attributes associated with the document Modified CP-ABE Technique for Documents encryption In Attribute Based Encryption scheme both the user secret key and the cipher-text are associated with a set of attributes. A user is able to decrypt the cipher-text if and only if at least a threshold number of attributes overlap between the cipher-text and user secret key. Different from traditional public key cryptography such as Identity Based Encryption [1], ABE is implemented for one-to many encryption in which cipher-texts are not necessarily encrypted to one particular user, it may be for more than one number of users. In Sahai and Waters ABE scheme, the threshold semantics are not very expressive to be used for designing more general access control system. Attribute Based Encryption (ABE) in which policies are specified and enforced in the encryption algorithm itself.

Existing CP-ABE depends how attributes and policy are associated with cipher texts and users’ decryption keys. In a CP-ABE scheme, a ciphertext is associated with a monotonic tree access structure and a user’s decryption key is associated with set of attributes. However, basic CP-ABE schemes are still not fulfilling the enterprise requirements of access control which require considerable flexibility and efficiency. CP-ABE has limitations in specifying policies and managing user attributes. In a CP-ABE scheme, decryption keys only support user attributes that are organized logically as a single set, so users can only use all possible combinations of attributes in a single set issued in their keys to satisfy policies. In CP-ABE, all the attributes are associated with the secrete key and it takes more computation time in case of any changes in attributes. Therefore to improve CP-ABE technique, we proposed attribute key concept. Using attribute key concept, system will generate unique attribute key for every document and the document will be encrypted using attribute key. The user’s attributes will be managed in key manager server with reference to attribute key.

**ADVANTAGES**

* The system is more secure due to the data contents which have been kept confidential to unauthorized individuals and collaborating users, including the curious cloud servers.
* The system is more secured since the Users from different groups cannot decrypt the cipher text by collaboration.

**SYSTEM REQUIREMENTS**

➢ **H/W System Configuration:-**

➢ Processor - intel i3,i5,i7

➢ RAM - 4 GB (min)

➢ Hard Disk - 256 GB

➢ Key Board - Standard Windows Keyboard

➢ Mouse - Two or Three Button Mouse

➢ Monitor - SVGA

**Software Requirements:**

* Operating System - Windows 7,8,10,11
* Coding Language - Java/J2EE(JSP,Servlet)
* Front End - J2EE
* Server - Apache Tomcat
* Database - MySQL