International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

# Shielding of Data Access Control for Multiauthority Cloud Storage Systems using Advanced Encryption Standard

**Abin K Ninan[1], Rex Benny[2], Soumya Sara Koshy[3]**

[1, 2]B.Tech Student, Department of Computer Science and Engineering, MBCCET, Peermade, Kerala, India

[3]Assistant Professor, Department of Computer Science and Engineering, MBCCET, Peermade, Kerala, India

**Abstract:** ***Data access control for multiauthority cloud storage systems (DAC-MACS) is a beneficial way to ensure data security of the cloud storage system. The two main challenging issues of the current cloud storage systems are data outsourcing and untrusted cloud servers. The existing access control policies cannot be applied further as they either produce multiple encrypted copies of the same data or it requires a fully trusted cloud server. In DAC-MACS, there exist multiple attribute authorities which have the capabilities to issue its own attributes independently without any assistance. A new extensive data access control scheme (NEDAC-MACS) with multiple attribute authorities is used which will reduce the workload of a single Attribute Authority (AA). It achieves revocation security by the use of file token specified. Using this scheme the owner's data can be accessed by users with the use of the unique secret key alongside with ciphertext token issued by the admin. The uploaded data can be accessed by the users on the owner's approval. In case if the owner uploads some inappropriate content, the users can report or flag to the server which leads to the blocking of the owner by the Certified Authority (CA). After the owner gets blocked after providing access control for the users, the data can be accessed further with the version key issued by the Attribute Authority and updated by admin as a new secret key and download the file. A 256 bit symmetric block cipher Advanced Encryption Standard (AES) is used to enhance the security. AES is an assuring method that provides access control of encrypted data and provides more secure attribute revocation.***

**Keywords:** Access control, Revocation security, CP-ABE, Multiauthority cloud

## 1. Introduction

Cloud Computing plays a dominant role in the area of security and privacy of data in cloud storage systems. It is required to protect important data from attackers. Cipher text policy attribute based Encryption (CP-ABE) is a convincing technique for access control of encrypted data. Because of the inefficiency of decryption and lack of immediate revocation the existing CP-ABE schemes cannot be directly applied to the multiauthority cloud storage systems. Data access control (DAC) [3] is the selective restriction of access to users. The DAC issues are related to security since all users get access of uploaded data, certain techniques and access policies need to be used to avoid unauthorized users to access data. Files can be stored in cloud server using effective encryption standard. Its prime requirement is a trusted authority which could manage all the attributes and distributes keys in the system. Users accessing the encrypted data in unauthorized manner only gets scrambled code, since the original data is in encrypted format. Existing CP-ABE schemes cannot be directly applied to the access control for multi-authority cloud storage systems, due to the inefficiency of decryption and revocation. J. Bethencourt introduced the mechanism of CP-ABE, the user can decrypt ciphertext only if user receive attribute and secret key from attribute authority and satisfies access policies incorporated in the cipher text [1]. K. Yang et al. proposed basic data access control scheme for multi-authority cloud storage system (DAC-MACS) [3] and an extensive data access control scheme (EDAC-MACS). By the use of most effective, new extensive data access control for multiauthority cloud storage systems (NEDAC-MACS) [8] two vulnerabilities can be solved, during the first attack, the revoked user can obtain other users key to update its secret key and obtain token to decrypt information as a non-revoked user. By the second attack the revoked user has the ability to get the update key to decrypt secret information as a non-revoked user. More secure attribute revocation is achieved by NEDAC-MACS. However, the data access control (DAC) issue of cloud computing systems has been escalated by the surge in attacks such as collusion, wiretapping and distort, so that DAC must be designed with sufficient resistance. DAC issues are mainly related to the security policies provided to the users accessing the uploaded data, and the techniques of DAC must specify their own defined security access policies and the further support of policy updates, based on which each valid user can have access to some particular sets of data whereas invalid users are unauthorized to access the data. Paper is organized as follows. Section II describes related works in data access control and Attribute based encryption. Section III presents implementation of NEDAC-MACS. Implementation is described in Section IV. Different algorithms such as secret key generation, AES algorithm, Token generation used is given in Section V. Finally, Section VI presents conclusion.

## 2. Related Work

Data access control systems based on cipher text policy attribute-based encryption (CP-ABE) are efficient, fine grained and revocable access schemes [3]. One approach to alleviate attacks is to store the outsourcing data in encrypted form. However, due to the normally semi trusted cloud and its arrangement issues of administration rights, cloud-based

access control approaches with traditional encryption are no longer applicable to cloud storage systems. Sanai and Waters laid a theoretical foundation for solving above encryption problem by introducing the new concept of attribute-based encryption (ABE) whose prototype is the identity-based encryption (IBE). The ABE notion has been the promising cryptographic approach on which more intensive research is based. V. Goyal et al first proposed the key-policy attribute based encryption for fine-grained access control (KP-ABE) [6]. In KP-ABE, the data was encrypted by attribute set, and decryption was possible only when the user's policy tree matched the attribute set in the cipher text. Shortly after KP-ABE, J. Bethencourt introduced the mechanism of CP-ABE [1], in which the user received attributes and secret keys from the attribute authority and was able to decrypt cipher text only if it held sufficient attributes that satisfied the access policy embedded in the cipher text. Furthermore, the constructed CP-ABE scheme is deemed as one of the most appropriate techniques for data access control in cloud storage systems, since it can be configured to some DAC schemes which do not require the data owners to distribute keys and furnish the data owners with more efficient and attribute-level control on defined access policies offline. A myriad of data access control techniques based on CP-ABE are proposed to construct the efficient, secure, fine-grained and attribute-level-revocable access schemes in a semi-trusted cloud storage system [4]. However, based on the Dole-Yao model, security goals such as active attack resistance, data confidentiality, anti-collusion, and attribute-revocation security of most solution designs cannot be all perfectly guaranteed since the capable Dole-Yao adversaries can overhear, intercept, replay, and synthesis arbitrary information in the open communication channels. For example, in context of attribute revocation in the scenario of Yang etc. [3].

## 3. Architectural Model

### 3.1 Architectural Model of DAC-MACS

Data Access Control is based on the promising CP-ABE technique are proposed to construct the efficient, secure, fine grained and revocable access schemes. Data access control issue of cloud computing systems has been increased by the attacks such as collusion, wiretapping and distort, so that DAC must be designed with sufficient resistance. DAC is used to ensure data security in cloud storage systems. A cloud storage system with multiple attribute authorities (DAC-MACS) has five types of entities: Certified authority (CA), users, cloud servers, admin, and attribute authority (AA). CA sets up the system and accepts the registration of all the users and AAs in the system. Every AA is responsible for issuing secret key and version key. Data owners give permission to authenticated user to access requested data and Admin provides file tokens.

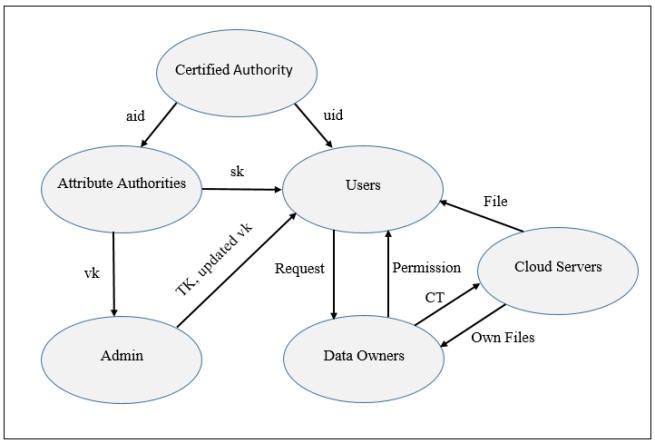


**Figure 1:** Architectural Model of DAC-MACS

DAC-MACS model consist of CA, Users, AA's, Admin and Cloud Server. CA registers Users and AA's. Secret key is sent to the users by the AA during the first time login. User request access for available files in cloud to the owner, when owner approve permission Admin sent token to users so file can be decrypted and downloaded. After giving access permission to users, if owner get revoked user cannot decrypt file using the secret key, in that case version key is generated by the AA and Admin update the version key as new secret key and user access original file. When a user upload files to cloud then become owner of the particular file and own files can be downloaded. DAC issues are mainly related to the security policies provided to the users accessing the uploaded data. Certain access techniques must be specified, so each valid user can have access to some particular sets of data whereas invalid users are unauthorized to access the data. Files are encrypted and stored in the cloud, if attempt to download files from cloud gets only cipher text. The framework of DAC-MACS consist of Initialization of the system, Secret key and version key generation by AA's, Data encryption by the data owners, Data decryption by users with the permission of data owners and admin, Attribute revocation using advanced encryption standard (AES). System Initialization can be done by the following steps:

**1) CA Setup:**
Globally trusted CA is setup using the CA setup algorithm. CA registers both users as well as the attribute authorities. CA can revoke the Owners by the inappropriate file contents posted by them.

**2) User Registration:**
User should register to CA using the uid assigned unique to users. When users upload file to cloud server they become the owners of the particular file and can decide who should access the data.

**3) AA Registration:**
AA should register to CA. The CA assigns aid, multiple AA's can be registered such that it will reduce the workload of a single attribute authority.

**4) AA Setup**
AA provides the secret key by the owner attributes and generates version key to update the secret key in case of revoked owners.

**5) Admin Setup**

Admin provide file tokens to decrypt the files with the user's secret key. In case of revoked users if permission of access is given previously to users admin update the version key generated by AA as new secret key.

**Table 1:** Entities and Description

| Entity | Description |
|---|---|
| CA | Global trusted authority sets up the system and accepts the registration of all the users and AAs in the system. Accept or Revoke the User. |
| AA | AA is responsible for generating a public attribute key for each attribute it manages and a secret key for each user associates with their attributes. Generates version key for the access of revoked users permitted files. |
| User and Owner | To decrypt a cipher text, each user may submit their secret keys issued by some AAs together with the file token. Permission for owners file other users are given initially by the same owner. |
| Admin | Admin sends file tokens, with the help of this token files are downloaded. Admin will updated the version key as new secret key to access the revoked user's file, if owner is blocked due to inappropriate content of other files. |
| Cloud Server | Stores the owner's data and provides data access service to users. Storage of encrypted files, files can be decrypted using secret key and file token. |

## 4. Implementation

An efficient and immediate attribute revocation method for multiauthority CP-ABE scheme that achieves both forward security and backward security. Techniques of DAC must specify their own defined security access policies and the further support of policy updates, based on which each valid user can have access to some particular sets of data whereas invalid users are unauthorized to access the data and revoked user access data of nonrevoked user. Due to the open and non-secure communication channel for attribute revocation, the revoked user can still break the backward revocation security both in DAC-MACS and EDAC-MACS. In NEDAC-MACS obtain more than two users Key Update Keys to update its Secret Key. NEDAC-MACS can with stand the static corruption of authorities since the file is decrypted based on the approval of multiple authorities not on the basis of single authority thus security is enhanced in cloud storage systems. In NEDAC-MACS the revoked user cannot update its secret key even by using some corrupted AAs. The implementation is summarized as follows:

1) User registration is done by the Certified Authority (CA) along with the secret key (SK) generated by the Attribute Authority (AA), users get the secret key to decode the encrypted file with the file token specified by the admin if owners permit to access the data to the users.
2) We propose multiple attribute authorities to generate secret key and version keys, it reduces the workload of a single attribute authority since the work is distributed equally.
3) User become data owner when files are uploaded to the cloud server and initial permission to access file is given by owner of particular file. Own files can be directly downloaded by the owner.
4) Data owners can be blocked by the CA based on the reports made by the users accessing the data. Count of the total complaints against the owners are used to send warning to the owner's. Based on human intelligence if the report's limits are exceeded against the inappropriate file contents hoisted by the owners.
5) Advanced Encryption Standard (AES) is a promising technique for access control of encrypted data. AES encryption algorithm is used for protecting classified information as well as the first publicly accessible and open cipher approved for top-secret information.
6) If data owners give access permission of uploaded files to users, admin will send the file tokens. After accepting the users request further if the owners gets revoked by CA, if the same owners other files contain some inappropriate contents, access permission is giving with the help of AA and admin. AA sends version key admin update this version key as new secret key, user also need to update as new secret key thus can access the revoked user's important files. Further decryption of cipher text can be done by using the updated secret key along with the file tokens.

## 5. Algorithms Used

### 5.1 Secret Key Generation

When data owners outsource their data with some attributes and is encrypted by attributes identity (aid) then it authenticates with user identity (uid), which is issued by CA. Each attribute authority assigns each valid user a set of attributes, then performs the SKeyGen algorithm. Secret key generation algorithm is a symmetric key generation algorithm, during the secret key user can download files from cloud in decrypted format with the file token. A unique secret key is given to each users during the first login time. Using the issued secret to the authenticated user, user can download the file for the permitted owners using the file tokens generated by the admin. Using the secret key users can decrypt files from cloud server, to access the data of revoked users with AAs generate version key and admin update the key as new secret key and need to be updated by user.

*Algorithm*

*Step 1. Get uid, guid of user.*
*Step 2. Get aid, gaid of Attribute Authority.*
*Step 3. Create a random number t.*
*Step 4. Rt = t * guid*
*Step 5. Calculate secret key sk*
*Step 6. sk = aid+Rt + uid*
*Step 7. Encrypt secret key sk*

### 5.2 AES Algorithm

The Advanced Encryption Standard (AES) is a symmetric block cipher to encrypt sensitive data. The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six time faster than triple DES. A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow. AES is an iterative rather than Feistel cipher, stronger

and faster than Triple-DES. It is based on substitution–permutation network. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations). AES performs all its computations on bytes rather than bits so AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix. AES encryption algorithm using a Symmetric key for encryption and decryption process. Firstly the original text i.e. clear text is converted into bytes and then for the AES algorithm to perform encryption, we need to generate key using the derived bytes and the symmetric key. In decryption process the encrypted text i.e. cipher text is converted into bytes, here too we will generate key using the derived bytes and the symmetric key. Unlike DES, the number of rounds in AES is variable and depends on the length of the key. Data file are uploaded to the cloud server in AES encrypted format by taking the file content as a string and encrypting the string and storing into new file and uploaded. Decryption of encrypted file are done by taking the uploaded file content again to a string, decrypting the string using the AES decryption to the original file content.

*Algorithm*
*Step 1. Create random class to generate random numbers.*
*Step 2. Multiply random number with user id and stored it in a new variables.*
*Step 3. Call aes_encrypted class for encrypting the number stored in the variable.*
*Step 4. Create aes key-size as 256 and block size as 128.*
*Step 5. Repeatedly hash the user password along with the salt.*
*Step 6. Divide the encrypted key and block size by 8.*
*Step 7. Convert the encrypted data as cipher text mode.*
*Step 8. Write the cipher text to the object using crypto stream class.*

### 5.3 Token Generation

Token generation is done by the Admin to access files to users. The user set queries for a decryption Token TK and CT by sending its secret keys, Then TK is computed by TKGen algorithm. File Tokens are provided to decrypt the encrypted file stored in the cloud server alongside with the secret key to download the file for permitted users, File access permission is initially given by the file owners. Users decrypt the encrypted files stored in the cloud sever with the secret key. File tokens are only provided by admin if and only if the owner give access permission to users for the specified file because, ultimate ownership is to the data owner. Permitted users can download the file using the secret key and the file token.

*Algorithm*
*Step 1. Get sk of user.*
*Step 2. Get guid of user.*
*Step 3. Create object r of Random class.*
*Step 4. Calculate token k,*
*Step 5. k = r.next( ) * (sk / guid)*
*Step 6. Encrypt token k.*

## 6. Conclusion

A new effective data access control scheme for multiauthority cloud storage systems (NEDAC-MACS) with multiple attribute authorities and using the symmetric AES encryption and decryption standard is proposed to withstand the vulnerabilities and thus enhance the revocation security. NEDAC-MACS ensure security against the corruption of authorities. The data owner can interact with the user directly for providing data access service. Owner's data can be accessed by the authenticated user's secret key and the file tokens sent by the admin. New users are added to the cloud server by the certified authority (CA) and attribute authorities generating the secret key thus implementing the multiauthority concept, when user upload data to cloud become data owners so that other users can access the requested data if permitted. If the data owner gets blocked by CA after giving access to user's based on the users report then data can be accessed by using the version key generated by AA, updated as the secret key by Admin. Thus the new extensive data access control for multiauthority cloud storage systems (NEDAC-MACS) can be done effectively using the AES encryption and multiple attribute authorities effectively.

## References

[1] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," Proc.IEEE Symp.Security & Privacy, pp. 321-334, 2007
[2] J. Hur, "Improving security and efficiency in attribute-based data sharing," IEEE Trans. Knowledge and Data Engineering, vol. 25, no. 10, pp.2271-2282, Oct. 2013
[3] K. Yang, X. Jia, and K. Ren, "DAC-MACS: Effective data access control for multiauthority cloud storage systems," IEEE Trans. Information Forensics and Security, vol. 8, no. 11, pp. 1790-1801, Nov. 2013
[4] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-Based Encryption with Non-Monotonic Access Structures," Proc. ACM Conf. Computer and Comm. Security, pp. 195-203, 2007
[5] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," J. Network and Computer Applications, vol. 34, no. 1, pp. 1-11, Jul. 2010
[6] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," Proc. ACM Conf. Computer and Comm. Security, pp. 89-98, 2006
[7] Vinoth Kumar P, Dr. P.D.R. Vijaya Kumar "Literature survey on revocable multiautority cipher text-policy attribute-based encryption(CP-ABE) scheme for cloud storage"
[8] Xianglong Wu, Rui Jiang, and Bharat Bhargava, Fellow, IEEE, "On the Security of Data Access Control for Multiauthority Cloud Storage Systems" IEEE Transactions on Services Computing Volume: PP,Year: 2015
[9] Zhiguo Wan, Jun'e Liu, and Deng, R.H., "HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing," IEEE

Trans. Information Forensics and Security, vol.7, no.2, pp. 743-754, April 2012

[10] Zhongma Zhu, Zemin Jiang, Rui Jiang, "The Attack on Mona: Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud," Proc.ISCC 2013, Guangzhou, Dec.7, 2013, pp. 185-189