An Efficient Multi Authority Data Access Control using Identity Based Signature Schema in cloud computing

Uriti Bhagya Latha[1],Behara Vineela[2]

[1]*Final M.Tech Student*, [2]Asst.professor

[1,2]Dept of CSE, *Sarada Institute of Science, Technology and Management (SISTAM), Srikakulam, Andhra Pradesh*

Abstract: *Data access control is an effective way to ensure the data security in the cloud. However, due to data outsourcing and untrusted cloud servers, the data access control becomes a challenging issue in cloud storage systems. Existing access control schemes are no longer applicable to cloud storage systems, because they either produce multiple encrypted copies of the same data or require a fully trusted cloud server. In this paper we are proposed mainly three concepts i.e. multi authority of users, key generation, encryption and decryption of cloud storage data. By implementing multi authority of users in cloud system we are using identity based digital signature schema. Another concept for generation of key using random code key generation process. In this paper data encryption and decryption process we are using extended tiny encryption algorithm. In this paper we are also implementing mailing concepts for sending second level. By using second level we can get first level for the purpose of data encryption and decryption. By implementing those concepts we can improve the efficiency of data accessing rate and also provide more security of data can be stored into cloud system.*

Keywords: *multi authority, key generation, security, cloud computing, signature*.

I. INTRODUCTION

Cloud storage is an important service of cloud computing [1], which offers services for data owners to host their data in the cloud. This new paradigm of data hosting and data access services introduces a great challenge to the data access control. Because the cloud server may give data access to the users who do not have the access permission for profit gain, the data owners can no longer trust the cloud servers and rely on them to do data access control. Cipher text identity based signature schema is regarded as one of the most suitable technologies for data access control in cloud storage systems, because it gives the data owner more direct control on access policies. In this schemes, the access policy checking is implicitly conducted inside the cryptography. That is, there is no one to explicitly evaluate the policies and make decisions on whether allows the user to access the data. In this scheme, there is an authority that is responsible for attribute management and key distribution. (The authority can be the registration office in a university, the human resource department in a company or the government education organization, etc.) The data owner defines the access policies and encrypts data according to the policies. Each user will be issued a secret key that reflects its attributes. A user can decrypt the data only when its attributes satisfy the access policies.

There are two challenging issues in the design of multiauthority access control schemes for cloud storage systems. The first issue is the problem of *Collusion*. Multiple users holding attributes from different authorities may collude together to obtain illegal access to the data. Existing multiauthority CP-ABE schemes [2], [3] usually rely on a global authority to collect and verify users' attributes and generate the secret keys for them. With the global authority, the collusion problem can be solved by using the key randomization mechanism as in the single authority schemes. However, the global authority is too powerful and it becomes a vulnerable point for security attacks and the performance bottleneck for large scale systems. Some multi-authority CP-ABE schemes [4], [5] are proposed to remove the global authority, but they still lack of scalability or efficiency. The other issue is the difficulty of *Attribute Revocation*. Existing attribute revocation methods designed for single authority CP-ABE [6]–[7] cannot be applied to multi-authority scenario. That is because, in multiauthority systems, there is no party to deal with the attribute revocation while still keeping the system secure against the collusion attack.

In this paper, we design an efficient multi-authority identity based signature schema without using a global authority and propose a multi-authority access control scheme for cloud storage systems. With no global authority, existing techniques for key randomization in multi-authority schemes are no longer applicable, because there is no such a global authority to tie all the pieces together. In our method, we introduce a certificate authority to assign a global user identifier to each user as in [4] and an authority identifier to each authority. The user identifier can uniquely identify a user in the system and it is used together with the secret keys issued by different authorities for data decryption, such that it is impossible for two users to collude together to gain illegal access of data. We also propose a new

technique to solve the attribute revocation problem in multi-authority systems. To improve the efficiency of attribute revocation, we move the work of re-encrypting the cipher text to the server by using proxy encryption method, such that there is no need for the server to decrypt the cipher text before re-encryption (i.e., the server cannot get the content key). The main contributions of this work can be summarized as follows.

1. We design an access control framework for multiauthority systems and propose an efficient and secure multi-authority access control scheme for cloud storage.

2) We design an efficient multi-authority identity based signature schema that does not require a global authority and can support any LSSS access structure.

3) We propose an efficient attribute revocation method for multi-authority while still keeping the system secure against the collusion attack.

The remaining of this paper is organized as follows. In Section II, we give the related work on data access control. After the definition of system model and security model in Section III, is to describe implementation of proposed system.. Finally, the conclusion is given in Section VII.

II. RELATED WORK

Cryptographic techniques are well applied to access control for remote storage systems [5]–[6]. The data owners encrypt files by using the symmetric encryption approach with content keys and then use every user's public key to encrypt the content keys. However, the key management is very complicated when there are a large number of data owners and users in the system. Also, the key distribution is not convenient in the situation of user dynamically joining or leaving the system, since it requires each data owner to be online all the time. Some methods [7]–[8] deliver the key management and distribution from the data owners to the remote server under the assumption that the server is trusted or semi-trusted. However, the server is cannot be trusted by the data owners in cloud storage systems and thus these methods cannot be
applied to access control for cloud storage systems

Some new cryptographic methods are proposed to the multi-authority ABE problem [4]–[9], [10], [11]. Chase [4] proposed a solution that introduced a global identifier to tie users' keys together. The proposed scheme also relies on a central authority to provide a final secret key to integrate the secret keys from different attribute authorities. However, the central authority would be able to decrypt all the cipher text in Chase's scheme, since it holds the master key of the system. Thus, the central authority would be a vulnerable point for security attacks and a performance bottleneck for large scale systems. Another limitation of Chase's scheme is that it can only express a strict "AND" policy over predetermined set of authorities. To improve Chase's scheme, Muller *et al.* [12] proposed a multi-authority ABE scheme that can handle any expressions in LSSS access policy, but it also requires a central authority. Chase *et al.* [13] also proposed a method to remove the central authority by using a distributed PRF (pseudo-random function). But it has the same limitation to strict "AND" policy of pre-determined authorities. Lin *et al.* proposed a decentralized scheme based on threshold mechanism. In this scheme, the set of
authorities is pre-determined and it requires the interaction among the authorities during the system setup. This scheme can tolerate collusion attacks for up to m colluding users, where m is a system parameter chosen at setup time. In , Lewko *et al.* proposed a new comprehensive scheme, which does not require any central authority. It is secure against any collusion attacks and it can process the access policy expressed in any Boolean formula over attributes. However, their method is constructed in composite order bilinear groups that incurs heavy computation cost. They also proposed a multi-authority CP-ABE scheme constructed in prime order group, but they did not consider attribute revocation, which is one of the major challenges in multi-authority access control for cloud storage.

III. PROPOSED SYSTEM

Cloud storage is an important service of cloud computing, which offers services for data owners to host their data in the cloud. This new paradigm of data hosting and data access services introduces a great challenge to the data access control. Because the cloud server may give data access to the users who do not have the access permission for profit gain, the data owners can no longer trust the cloud servers and rely on them to do data access control. Before retrieve data from cloud storage system each user will identify the given users are authenticated users or not. After performing authentication process the cloud service will generate key for encryption and decryption process stored data. If any user want to retrieve data from the cloud they are verify the status and also retrieve key from the data base. After retrieving key we can decrypt the data and get original plain format data. The implementation procedure of proposed system is as follows.

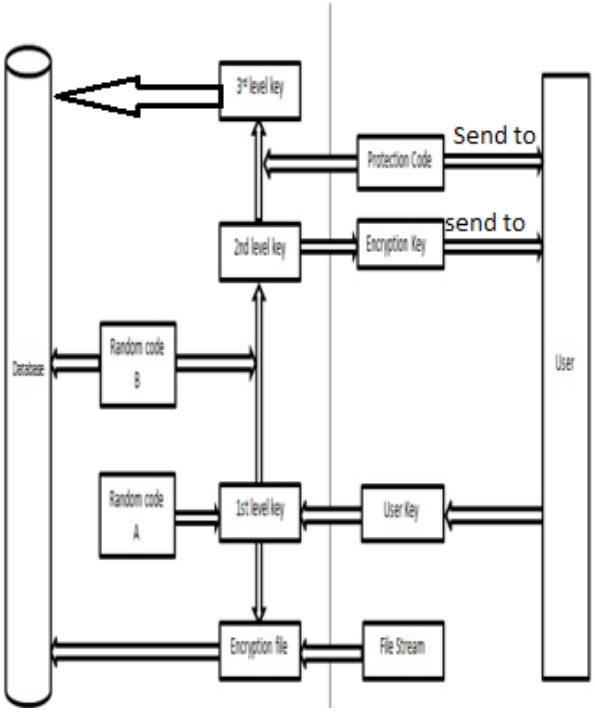
Identity based digital signature schema:

In this module each user will registered into cloud storage system. After completion of

registration each user will get username, password and also the verification code. The cloud service will send verification for individual users and using that code the users will generate signature. The users will send the signature to cloud service and get authentication status. The cloud service will generate signature for each user and compare both signatures. If the signatures are equal the cloud service will send authentication status to individual users. The cloud service will send verification code to users using mail.

Key Generation and File Encryption:

In the process of file encryption, the code A is randomly generated, and the string in request stream is encoded with code A to generate the first level encryption key. Subsequently, the data owner would use the first level encryption key to encrypt files using the extended tiny encryption algorithm. Finally, a new file is generated based on the original file and the first level encryption key. The new file is stored in the cloud storage system. This is the first level encryption. In the second level, the random code B is generated, and the first level encryption key are encoded with code B to generate the second level encryption key. The code B is stored in the database, and the new file of second level encryption key is generated and sent to the user by using the mail which is developed by using the smtp protocol. In case of losing the second level encryption key, the system generates the third level encryption key based on the second level encryption key and a protection code which is randomly generated by the system. The third level encryption key is stored in the database. In conclusion, the random code B and the third level key are stored in the database, and the encrypted file is stored in the cloud storage system. The user needs to save the second level encryption key, and remember the protection code. Other files or keys used in the processes need not be saved. The cloud service would send user a common encryption key (the common encryption key is also the second level encryption key, which is sent to the user). To upload a file, the data owner needs to upload file to be encrypted and stored into cloud service. The user would decrypt the general encryption key to the first level encryption key using the database stored random code B.



The pseudo code for encryption process is as follows.

```
void encipher(unsigned int num_rounds, uint32_t v[2], uint32_t const key[4])
{
   unsigned int i;
   uint32_t v0=v[0], v1=v[1], sum=0, delta=0x9E3779B9;
   for (i=0; i < num_rounds; i++)
{
      v0 += (((v1 << 4) ^ (v1 >> 5)) + v1) ^ (sum + key[sum & 3]);
      sum += delta;
      v1 += (((v0 << 4) ^ (v0 >> 5)) + v0) ^ (sum + key[(sum>>11) & 3]);
   }
   v[0]=v0; v[1]=v1;
}
```

File decryption process:

If the user wants to download the files, the second level encryption key must be provided. Using the second level encryption key and the database-stored random code B, the system can decrypt the first level encryption key and decrypt the files, then send the files in the form of stream to the client. The decrypted files would be generated in the client side.

The decryption process of extended tiny encryption algorithm is as follows.

```
void decipher(unsigned int num_rounds, uint32_t v[2], uint32_t const key[4])
 {
   unsigned int i;
   uint32_t v0=v[0], v1=v[1], delta=0x9E3779B9, sum=delta*num_rounds;
   for (i=0; i < num_rounds; i++)
{
      v1 -= (((v0 << 4) ^ (v0 >> 5)) + v0) ^ (sum + key[(sum>>11) & 3]);
      sum -= delta;
      v0 -= (((v1 << 4) ^ (v1 >> 5)) + v1) ^ (sum + key[sum & 3]);
   }
   v[0]=v0; v[1]=v1;
}
```

After encrypt the file each user will get original file with a secure manner. By implementing those concepts we can provide authentication of each user in cloud and also provide more efficient data access control policy. Because in this paper we are using mailing concepts for sending second level key and also send verification code of individual users or clients.

IV. Conclusions

In this paper, we defined a new access control framework for multi-authority systems in cloud storage and proposed an efficient and secure multi-authority access control scheme. We first designed an efficient multi-authority scheme that does not require a global authority and can support any less access structure. Then, we proved that our multi authority using identity based signature scheme is provably secure in the random oracle model. We can also propose other concepts for generation of shared key for encryption and decryption file. After encrypt the file we can stored into cloud storage system. In this paper the generation of encryption key can be done by cloud service and send that key to data owner. Before sending key to data owner the cloud service also send second level key to all users in cloud. By using the second level key each user will get first level encryption key. Using that key each user will perform the decryption process and get original plain format data. in this we are using extended tiny encryption algorithm for encryption and decryption cloud stored data. by implementing those concepts we can provide more efficiency for data accessing and also provide more security of cloud storage data.

V. References

[1]. P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, Tech. Rep., 2009.

[2]. M. Chase, "Multi-authority attribute based encryption," *Theory of Cryptography*, vol. 4392, pp. 515–534, 2007.

[8] S. M¨uller, S. Katzenbeisser, and C. Eckert, "Distributed attribute-based encryption," *Information Security and Cryptology*, pp. 20–36, 2009.

[3] M. Chase and S. Chow, "Improving privacy and security in multiauthority attribute-based encryption," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 121–130.

[4] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," *Advances in Cryptology–EUROCRYPT 2011*, pp. 568–588, 2011.

[5] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*. ACM, 2010, pp. 261–270.

[6] J. Hur and D. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Transactions on Parallel and Distributed Systems*, 2010.

[7] S. Jahid, P. Mittal, and N. Borisov, "Easier: encryption-based access control in social networks with efficient revocation," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. ACM, 2011, pp. 411–415.

[8]. M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Plutus: Scalable secure file sharing on untrusted storage," in *Proceedings of the 2nd USENIX Conference on File and Storage Technologies*. Berkeley, CA, USA: USENIX Association, 2003, pp. 29–42.

[9]. D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Advances in Cryptology–CRYPTO 2001*. Springer, 2001, pp. 41–62.

[10] D. Li, X. Du, X. Hu, L. Ruan, and X. Jia, "Minimizing number of wavelengths in multicast

routing trees in wdm networks," *Networks*, vol. 35, no. 4, pp. 260–265, 2000.

[11]. A. Lewko and B. Waters, "Decentralizing attribute-based encryption," *Advances in Cryptology–EUROCRYPT 2011*, pp. 568–588, 2011.

[12]. H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," *Information Sciences*, vol. 180, no. 13, pp. 2618–2632, 2010.

[13] J. Li, Q. Huang, X. Chen, S. S. M. Chow, D. S. Wong, and D. Xie, "Multi-authority ciphertext-policy attribute-based encryption with accountability," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '11. New York, NY, USA: ACM, 2011, pp. 386–390.

[14]. S. M¨uller, S. Katzenbeisser, and C. Eckert, "Distributed attribute-based encryption," *Information Security and Cryptology*, pp. 20–36, 2009.

[15] M. Chase and S. Chow, "Improving privacy and security in multiauthority attribute-based encryption," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 121–130.

BIOGRAPHIES:

Uriti Bhagya Latha , is student in M.Tech(CSE) in Sarada Institute of Science Technology and Management, Srikakulam. She has received her B.Tech(C.S.E) from GMRIT, Rajam ,Srikakulam. she is interesting areas are datamining and network security.

Behara Vineela is working as Asst.professorin Sarada Institute of Science, Technology And Management , Srikakulam, Andhra Pradesh. She received her M.Tech (CSE)from AITAM ,Tekkali,Srikakulam, AndhraPradesh. JNTU Kakinada Andhra Pradesh.Her research areas include Network Security