International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
NATIONAL CONFERENCE on Developments, Advances & Trends in Engineering Sciences
(NCDATES- 09th & 10th January 2015)

RESEARCH ARTICLE OPEN ACCESS

Shared Authority Based Privacy-preserving Authentication Protocol in Cloud Computing

*Mr.Kulakarni Harish, ** Mr.Ravi kumar chandu,
'M.Tech student, CMR Engineering College, Medchal, Hyderabad
Assoc.Prof, Dept. of CSE,CMR Engineering College, Hyderabad.

Abstract:
Cloud computing is emerging as a prevalent data interactive paradigm to realize users' data remotely stored in an online cloud server. Cloud services provide great conveniences for the users to enjoy the on-demand cloud applications without considering the local infrastructure limitations. During the data accessing, different users may be in a collaborative relationship, and thus data sharing becomes significant to achieve productive benefits. The existing security solutions mainly focus on the authentication to realize that a user's privative data cannot be unauthorized accessed, but neglect a subtle privacy issue during a user challenging the cloud server to request other users for data sharing. The challenged access request itself may reveal the user's privacy no matter whether or not it can obtain the data access permissions. In this paper, we propose a shared authority based privacy-preserving authentication protocol (SAPA) to address above privacy issue for cloud storage. In the SAPA, 1) shared access authority is achieved by anonymous access request matching mechanism with security and privacy considerations (e.g., authentication, data anonymity, user privacy, and forward security); 2) attribute based access control is adopted to realize that the user can only access its own data fields; 3) proxy re-encryption is applied by the cloud server to provide data sharing among the multiple users. Meanwhile, universal compos ability (UC) model is established to prove that the SAPA theoretically has the design correctness. It indicates that the proposed protocol realizing privacy-preserving data access authority sharing, is attractive for multi-user collaborative cloud applications.

Existing System:

However, most previous researches focus on the authentication to realize that only a legal user can access its authorized data, which ignores the case that different users may want to access and share each other's authorized data fields to achieve productive benefits. When a user challenges the cloud server to request other users for data sharing, the access request itself may reveal the user's privacy no matter whether or not it can obtain the data access permissions. In this work, we aim to address a user's sensitive access desire related privacy during data sharing in the cloud environments, and it is significant to design a humanistic security scheme to simultaneously achieve data access control, access authority sharing, and privacy preservation.

Disadvantage:

Previous System does not have the option of granting/revoking data access

Proposed System:

In this paper, we address the aforementioned privacy issue to propose a shared authority based privacy preserving authentication protocol (SAPA) for the cloud data storage, which realizes authentication and authorization without compromising a user's private information.

The main contributions are as follows. 1) Identify a new privacy challenge in cloud storage, and address a subtle privacy issue during a user challenging the cloud server for data sharing, in which the challenged request itself cannot reveal the user's privacy no matter whether or not it can obtain the access authority. 2) Propose an authentication protocol to enhance a user's access request related privacy, and the shared access authority is achieved by anonymous access request matching mechanism. 3) Apply cipher text-policy attribute based access control to realize that a user can reliably access its own data fields, and adopt the proxy re-encryption to provide temp authorized data sharing among multiple users

Advantage:

Here we proposed the secured system and data owner can decide whether the user can access the system or not.

PROBLEM STATEMENT:

In our model, privacy is accomplished by encrypting the data it can prevent the un authorized access.

Scope:

We are going to raise the privacy level of the data owner and the confidentiality of the data by providing access to users

Architecture:

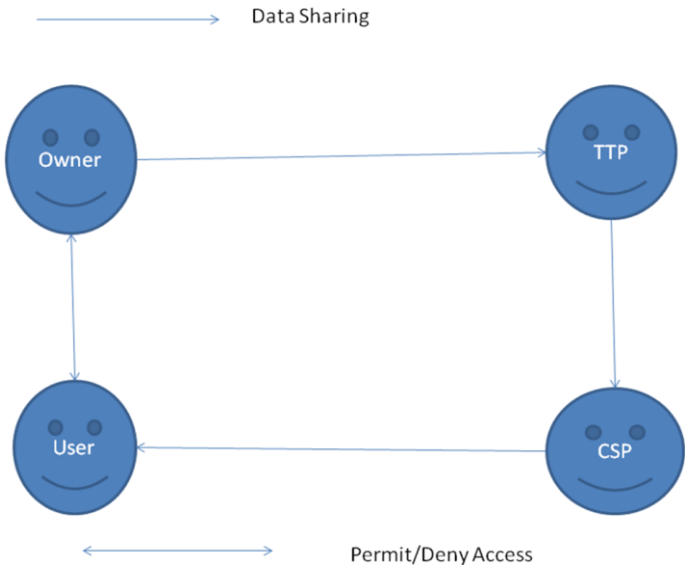

fig1: problem statement

Modules :

1. **Owner**
2. **User**
3. **Access Control**
4. **Cloud Service Provider**
5. **Encryption & Decryption**
6. **File Download**
7. **Trusted Third Party**

Modules Description

Owner Registration:

In this module an owner has to upload its files in a cloud server, he/she should register first. Then only he/she can be able to do it. For that he needs to fill the details in the registration form. These details are maintained in a database.

Owner Login:

In this module,any of the above mentioned person have to login,they should login by giving their emailid and password .

User Registration:

In this module if a user wants to access the data which is stored in a cloud, he/she should register their details first. These details are maintained in a Database.

User Login:

If the user is an authorized user, he/she can download the file by using file id which has been stored by data owner when it was uploading.

Access Control:

Owner can permit access or deny access for accessing the data. So users can able to access his/her account by the corresponding data owner. If owner does not allow, user can't able to get the data.

Encryption & Decryption:

Here we are using this aes_encrypt & aes_decrypt for encryption and decryption. The file we have uploaded which has to be in encrypted form and decrypt it

File Upload:

In this module Owner uploads the file(along with meta data) into database, with the help of this metadata and its contents, the end user has to download the file. The uploaded file was in encrypted form, only registered user can decrypt it.

File Download:

The Authorized users can download the file from cloud database.

Cloud Service Provider Registration:

In this module , if a cloud service provider(maintainer of cloud) wants to do some cloud offer , they should register first.

Cloud Service Provider Login:

After Cloud provider gets logged in, He/ She can see Cloud provider can view the files uploaded by their clients. Also upload this file into separate Cloud Database

TTP (TRUSTED THIRD PARTY) LOGIN:

In this module TTP has monitors the data owners file by verifying the data owner's file and stored the file in a database .Also ttp checks the CSP(CLOUD SERVICE PROVIDER),and find out whether the csp is authorized one or not.

Literature survey:

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things r satisfied, ten next steps are to determine which

operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

Implementation:

We have implemented our basic approach on Amazon S3 which is a popular cloud based storage service. The content management consists of two tasks. First, the Owner encrypts the data item sets based on the access control policies and uploads the encrypted sets along with some meta-data. Then, authorized users download the encrypted data items sets and meta-data from the Cloud , and decrypt the data item sets using the secrets they have. Now we illustrate the interactions of the Owner with Amazon S3 as the Cloud . In our implementation, we have used the REST API to communicate with Amazon S3. Figure 2 shows the overall involvement of the Owner in the user and content management process when uploading the data item sets to Amazon S3. While the fine-grained access control is enforced by encrypting using the keys generated through the AB-GKM scheme, it is important to limit the access to even the encrypted data item sets in order to minimize the bandwidth utilization. We associate a hash-based message authentication code (HMAC) with each encrypted data item sets such that only the users having valid identity attributes can produce matching HMACs.Initially the Owner creates a bucket , which is a logical container in S3, to store encrypted data item sets as objects . Subsequently, the Owner executes the following steps:1. The Owner generates the symmetric keys using the AB-GKM's KeyGen algorithm and instantiates an encryption client. Note that the Owner generates a unique symmetric key for each policy configuration

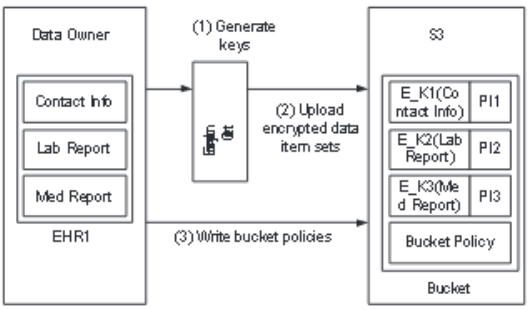

Fig.2 implementation details

Comparison & approaches:

In this section we compare ABE-based existing approaches as a whole and the two AB-GKM based approaches presented earlier. A common characteristic of all these approaches is that they support secure attribute based group communication.

Table 1: Comparison of Approaches

Table 1: Comparison of Approaches

Property	ABE	SLE	TLE
Cryptosystem	Asymmetric	Symmetric	Symmetric
Secure attribute based group communication	Yes	Yes	Yes
Efficient revocation	No	Yes	Yes
Delegation of access control	No	No	Yes

As shown in Table 1, while ABE-based approaches rely on asymmetric cryptography, our two approaches rely only on symmetric cryptography which is more efficient than the asymmetric cryptography. A key issue in the ABE-based approaches is that they do not support efficient user revocations unless they use additional attributes [2]. Our schemes address the revocation issue. It should be noted that the ABE based approaches and our SLE approach follows the conventional data outsourcing scenario by which the data owner manages all users and data before uploading the encrypted data to the cloud, whereas the TLE based approach provides the advantage of partial management of users and data in the cloud itself while assuring confidentiality of the data and privacy of users. With ever increasing user base and large amount of data, while such delegation of user management and access control is becoming very important, it also has trade offs in terms of privacy. Compared to the SLE approach, in the TLE approach, the data owner has to reveal partial access control policies to the cloud which may allow the cloud to infer some details about the identity attributes of users. It is an interesting topic to investigate how to construct symmetric key based practical solutions to hide the access control policies from the cloud while utilizing the benefits of delegation of control.

Conclusion:

In this work, we have identified a new privacy challenge during data accessing in the cloud computing to achieve privacy-preserving access authority sharing. Authentication is established to guarantee data confidentiality and data integrity. Data anonymity is achieved since the wrapped values are exchanged during transmission. User privacy is enhanced by anonymous access requests to privately inform the cloud server about the users' access desires. Forward security is realized by the session identifiers to prevent the session correlation.

It indicates that the proposed scheme is possibly applied for enhanced privacy preservation in cloud applications.

References:

[1] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In SP 2007: Proceedings of the 28th IEEE Symposium on Security and Privacy, pages 321–334, 2007.

[2] J. Camenisch, M. Dubovitskaya, R. R. Enderlein, and G. Neven. Oblivious transfer with hidden access control from attribute-based encryption. In SCN 2012: Proceedings of the 8th International Conference on Security and Cryptography for Networks, pages 559–579, 2012.

[3] D. Halevy and A. Shamir. The LSD broadcast encryption scheme. In CRYPTO 2001: Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology, pages 47–60, 2002.

[4] J. Li and N. Li. OACerts: Oblivious attribute certificates. IEEE Transactions on Dependable and Secure Computing, 3(4):340–352, 2006.

[5] M. Nabeel and E. Bertino. Towards attribute based group key management. In CCS 2011: Proceedings of the 18th ACM conference on Computer and communications security, 2011.

[6] M. Nabeel, N. Shang, and E. Bertino. Privacy preserving policy based content sharing in public clouds. IEEE Transactions on Knowledge and Data Engineering, 99, 2012.

[7] OpenID. http://openid.net/ [Last accessed: Oct. 14, 2012].

[8] T. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In CRYPTO 1991: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology, pages 129–140, 1992.

[9] N. Shang, M. Nabeel, F. Paci, and E. Bertino. A privacy-preserving approach to policy-based content dissemination. In ICDE 2010: Proceedings of the 2010 IEEE 26th International Conference on Data Engineering, 2010.

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

[11] S. Yu, C. Wang, K. Ren, and W. Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In INFOCOM 2010: Proceedings of the 29th conference on Information communications, pages 534–542, 2010.