ISSN:0975-9646

A New Approach for Finding the Guilt Agent in Dynamic Groups on Cloud Computing

Y. Venkateswarlu[1], K. Sowrya[2], and T. Anuradha[3]
[1] *Student, Department of Electronics and computer Engineering, K L University, India.*
[2]*Student , Department of Electronics and computer Engineering, K L University, India.*
[3]*Associate Professor, Department of Electronics and computer Engineering, K L University, India.*

***Abstract*— There is always a risk of data owner's identity and privacy when data is stored in a untrusted cloud. As the data can be easily accessed by others in the cloud, there is also a risk of data modification or data stealing. To overcome this, a method that encrypts the data before it is stored in the cloud and a technique to find the guilt agent is proposed. This paper also focuses on the dynamic groups that work in organizations that use cloud for their data storage, where a newly joined user can directly decrypt the files stored in the cloud before he joined the cloud. We are also proposing a method using which storage servers as well as unauthorized users cannot learn the content stored in the cloud.**

***Keywords*— guilt agent, secure cloud computing, dynamic groups, file log, group key.**

I. INTRODUCTION

Recently, people have started using cloud as an alternative technique to traditional way of storing the information because of its immense features and as the cost of maintenance of cloud is low. One of the significant uses of cloud computing is it reduces investment on local infrastructures. For example consider GOOGLE DRIVE the cloud service offered by the Google that can be used to store and share the files. With the help of the cloud, we are completely relieved from the local data storage and its maintenance. However, the point is whether our data privacy is preserved or not i.e. whether confidentiality is maintained as the cloud server is not the completely trusted one. So to preserve the data privacy, one standard solution available is making the data encrypted before it is uploaded into the cloud.

The important and most concerned task here is to design a scheme for storing the data as well as sharing the data which should be efficient and secure because of the following issues which are very much challenging. First and foremost is preserving the privacy of identity. Suppose you have booked a holiday trip and got ticket through email you can see a lot of ads relating to hotels and tour packages in our Gmail account. Which means our privacy is not disclosed.

The second challenging issue is maintaining multiple owners. Previously only who is the owner of the cloud is only able to store and change the data in the cloud. But every member of the cloud should be able to store and share the data in the cloud.

The third challenging issue is to maintain the groups which are dynamic in nature as a group can't be static always. There is every possibility that a new member can always join the group or an existing member can leave the group at any time. These dynamic groups can play an important role in securing the data. There can be chances of data modification in such cases we propose an efficient method to find the guilt agent and an exiling policy.

Many privacy techniques have been proposed. One such method is the data owner distributes the decryption keys to authorized users without they cannot access the data content. However adding new user or a rescinded old user service is still a challenging issue. As the new user should get the decryption key from the data owner and a rescinded old user must be denied permission for accessing the data contents. The proposed system overcomes this problem by an attribute key generation technique. Using which a new user can directly access the data stored before his time and even does not require decryption key from the data owner to access the file.

II. RELATED WORK

In [1], the authors proposed a secure data sharing schema, Mona. In Mona, a data owner can share data with others in the group without revealing their identity. Moreover, it also supports user revocation that does not involve any updation of the keys of the other users, and new users can access the files stored before their presence in the cloud.

In [2], Priyanka Barge, Pratibha Dhawale and Namrata Kolashetti proposed a data leakage detection by using fake objects in between the original records that helps to find out the guilt agent. This method overcomes the problems with watermarking. But also uses up hard disk memory by using fake records.

In [3], Kallahalla et al. defined a cryptographic storage system which provides secure file sharing. By breaking the data into groups and encrypting each group with a key, the data owner can share the file groups with others by handover of the corresponding key. But it gives an additional load for key distribution.

In [6], the file is stored along with metadata and file data. The file metadata contains the access data that relates to the collection of encrypted keys. These metadata files are encrypted with public key of authorized users. But here we need to update the private keys of others while adding a new user. This limits the application to support dynamic groups.

The re-encryption model given by Ateniese et al. [7] strengthens the cloud storage. The data encryption done by the data owners is a twostep procedure. First, encryption is done using exclusive and symmetric content keys. Second, the data is encrypted with a master public key. Cryptography is used by the server to re-encrypt the particular content keys from the master public key. On the other hand, the remote storage server can be attacked by any malicious user to find the decryption keys of all encrypted blocks.

Therefore, from the above research, we can understand how securely one can share data files in a multiple-owner concept for dynamic groups while being preserving the identity privacy from an untrusted cloud. But it still remains as a challenging issue.

III. System Design And Design Goals

A. System Design

For the proposed system there are 3 major actors. They are the group manager, the group member and the cloud. The group member can register, revoke, can upload a file and can download the file.

The group manager can add users, see the members of the entire group and their activities, file related activities and can rescind guilt agent when found that he has committed for some mischief by data modification.

The cloud is provided by the cloud service provider and is maintained by the group manager. Being an untrusted one i.e. cloud users will show less interest to join in cloud. But we consider that the group manager can be fully trusted one. As per our assumption the group manager takes care that our data should be encrypted before it is stored in cloud.

Another major action performed by group manager is finding the guilt agent. A guilt agent is one in the group who leaks the important data outside the group and may cause disturbance to the privacy of the data owner. A guilt agent is not an unauthorized user who was stopped at the login step in proposed work.

B. Design Goals

Here in this section we characterize some of the design goals that are considered as pre requisites for designing the proposed work. These include aspects relating to access, privacy, storage and efficiency.

1) *ACCESS*: A group member should register himself before his first login for obtaining an encrypted key necessary to make him an authorized user. Unauthorized users must not get access to the contents in the cloud. The details at login are verified with the one stored in server before granting access. The rescind users will not get access as their profile is updated in the server to make him remain blocked forever.

2) *PRIVACY:* The feature that a new user can access the contents stored before his time without contacting the data owner for decryption keys is achieved by the hide out of the data owner's identity. The users in group can access the files but cannot learn the owners of the data files. Thus their privacy is preserved.

3) *STORAGE:* A group member is provided with full freedom to access the files uploaded, download them and also can update them. But certain important data can be modified that may result in disputes. For example a mischief person in the department of a university may change the holiday schedule noticed by updating the actual notice without revealing his identity to others. So a log is maintained to resolve such disputes.

4) *EFFICIENCY:* The efficiency of the proposed system is outlined as follows: within the cloud a group member can store and share his data files with others in group. User's rescind can be easily achieved without updating the keys of the remaining users. Newly granted users can access the files stored before his time without contacting the data owner.

IV. Experimental setup

The system model was developed using ASP.NET and database was created using Microsoft SQL server 2008. This works on all windows versions like XP and 7.

The data is encrypted before it is uploaded to the cloud. The identity of the file owner is only known to the admin. The other members of the same group even do not know the owner of the file. Thus the identity of data owner is not revealed.

A newly joined user can now decrypt the file without contacting the file owner for the decryption keys. The owner of the file is not revealed or not known to all.

A log file is created and maintained by admin. This records all the actions that are performed by the user on a file. But here the point to be noticed is the user identity is not known. Depending upon the key generated against his account he is identified.

A. Group Member

The group member in the proposed work will do the following works. These include registration, obtaining group signature key, file upload, file download, file edit and save and also can rescind his account. The roles of the group members is shown below from fig1 to fig

B. Group Manager

The group manager in the proposed work is the fully trusted one. He has several activities that includes monitoring the group users and files. He also can rescind the guilt agent who modified data at a pace that causes controversy. He maintains a log which has several options like update, download, upload and delete. These are the functions that a user can perform in the proposed work. The exiling of a guilt agent is done up updating his group key that prevents him from getting logged in as his group key is updated so now he may not use the previous key.

1) GROUP KEY GENERATION

Before outsourcing user information M to the cloud servers, the system processes the information as follows.

1. It first divides the information M into several small data components as M= $\{m1,m_n\}$ according to the logic granularities. For example, the person record

data may be divided into {name, password, and sex, emailed}.

2. It encrypts each data component m_i with various content keys k_i (i=1, n) by using the symmetric encryption techniques.
3. For each content key K_i(i=1,…,n), the owner defines the access structure M over the universe of attributes S and then encrypts K_i under this access structure by running the encryption algorithm Encrypt.

The encryption algorithm Encrypt can be constructed as follows. It takes as inputs the public parameters PP, a set of public attribute key {PK_x}, a content key k and a LSSS access structure (M1, p). Let M1 be a l x n matrix, where l denotes the number of attributes involved in the encryption.

The function p contains rows of M1 to attributes. It first chooses a random encryption exponent s belongs to Z_p and a random vector v=(s, y2, .yn). Where $y_2....y_n$ are used to share the encryption exponent s. For i=1 to l, it computes $\lambda_i = v \cdot M_i$, where Mi is the vector corresponding to the i-th row of M1. Then it randomly chooses r_1, r_2, r_l and computes the cipher text as

CT_1	$E_{K1}(m1)$	.....	……..	CT_n	$E_{Kn}(m_n)$

Fig.4.1 group key format

2) ENTITY DESCRYPTION

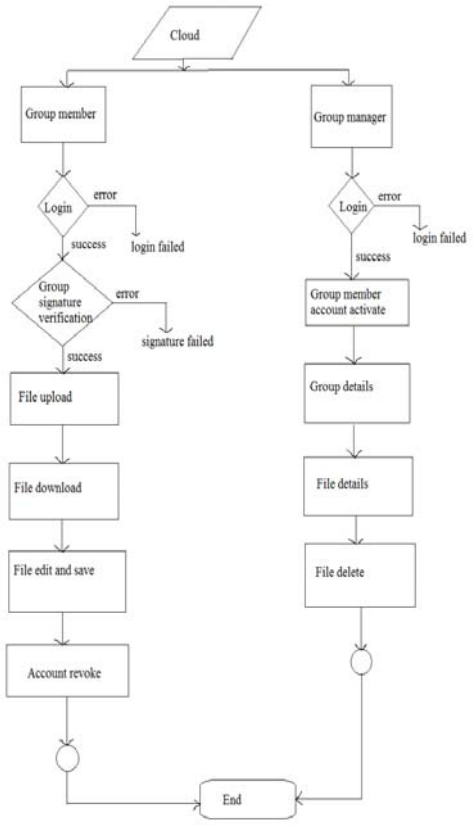

Fig.4.2 Data flow diagram representing actions of the entities

3) GUILT AGENT MODELING

The previous methods of finding guilt agent are watermarking and inserting the fake objects. Here a new method is proposed to find the guilt agent. Here we set a flag which keeps the track of all user actions on a particular file. When a new user has registered the flag is set to inactive. When the user login for the first time the flag is now set to active. When the user uploaded a file the flag will be set to upload. Similarly the flag is set to view when user views the file, set to update when user updates, sets to download when the user downloads and set to delete when user deletes the file. In this manner each and every action of the user is recorded and this data can be used to find the guilt agent who modified a file that leads to controversy.

V. PERFORMANCE EVALUATION

A) Performance:

Whenever a new user entered into the cloud he does not require to contact the file owner for decryption key rather than he would ask the group key manager for obtaining secret key. The Group manager decrypts the file with master key and encrypts with the new group key and now it can be made available to everyone in the group.

B) Security

The group member is authenticated by the key. Any outside person will not get access to the content unless he is authenticated. He can only be authenticated if he knows the group key. The rescind policy is improved by allowing the group member to communicate with the manager for revocation.

C) Improvement

Till today the data leakage detection is carried by two popular methods like watermarking and fake objects. This paper proposes a new method of keeping the tracks for the file that records all the actions done to a particular file. One disadvantage is it requires some amount of memory to store log but it is very much less when compared to other two methods watermarking and inserting fake objects.

VI. CONCLUSION

In this paper, we developed a guilt agent finding model in dynamic groups while being enhancing its security features that makes users join even with the untrusted cloud source. In this model, a member can store data on the cloud without letting others to know his identity. The group manager finds the guilt agent who modified or leaked the important data. The new user registration and exiling is made easy since the user can communicate with the group manager through the rescind policy.

VII. FUTURE SCOPE

To achieve approximate results of finding guilt agent we can model the system in such a way that the updated file can also be downloaded by this admin. This helps us to find the real guilt agent when a same file is modified by more than one at a time.

REFERENCES

[1] Xuefeng Liu, Yuqing Zhang, Member, IEEE, Boyang Wang, and Jingbo Yan Xuefeng Liu, Yuqing Zhang, Member, IEEE, Boyang Wang, and Jingbo Yan, proposed a "MONA" in IEEE transactions on parallel and distributed systems, vol. 24, no. 6, june 2013.

[2] Priyanka Barge, Pratibha Dhawale and Namrata Kolashetti A novel data leakage detection Ajmer, vol.3 feb2013.

[3] M. Kallahalla, E. Riedel, R. Swaminathan, Q.Wang, and K. Fu,"Plutus: Scalable Secure File Sharing on Untrusted Storage," Proc.USENIX Conf. File and Storage Technologies, pp. 29-42, 2003.

[4] S. Kamara and K. Lauter, "Cryptographic Cloud Storage," Proc. Int'l Conf. Financial Cryptography and Data Security (FC), pp. 136-149, Jan. 2010.

[5] Kan yang and Xioaohua Jia's security for cloud storage systems.

[6] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, "Sirius: Securing Remote Untrusted Storage," Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 131-145, 2003.

[7] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 29-43, 2005.

[8] 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 (CCS).

[9] D. Naor, M. Naor, and J.B. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers," Proc. Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 41-62, 2001.

[10] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 213-229, 2001.

[11] Yang, Yanjiang, and Youcheng Zhang. "A generic scheme for secure data sharing in cloud." In *Parallel Processing Workshops (ICPPW), 2011 40th International Conference on*, pp. 145-153. IEEE, 2011.

SCREEN SHOTS

fig.1 welcome page

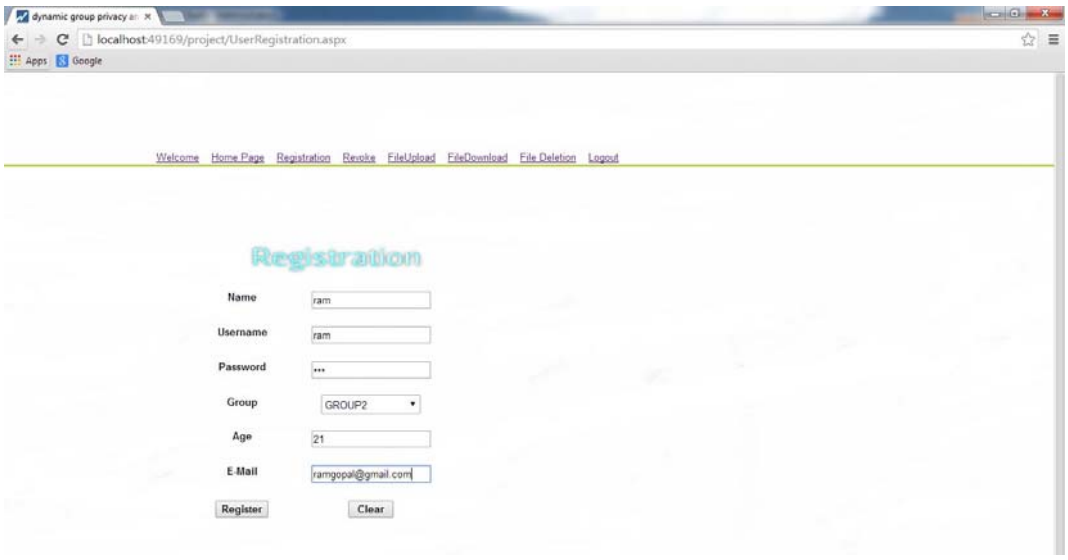

Fig 2 user registration page

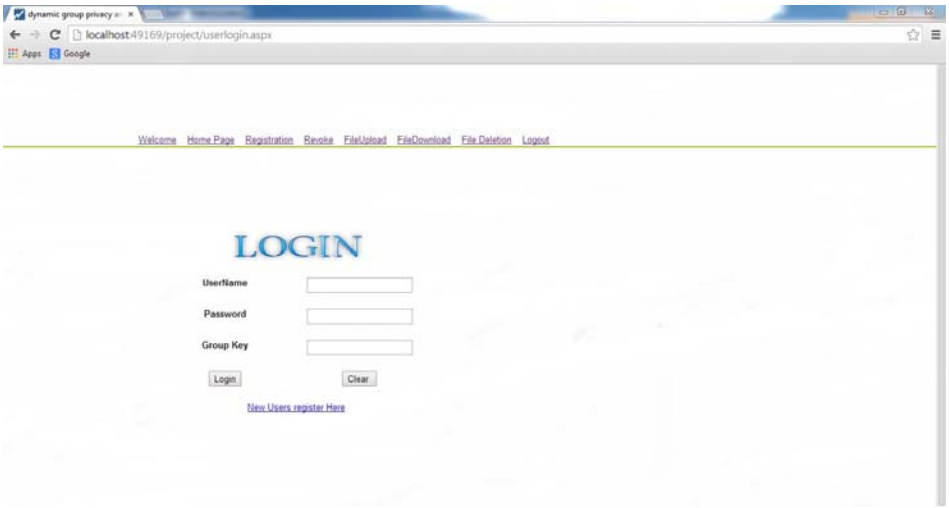

Fig 3 userlogin page

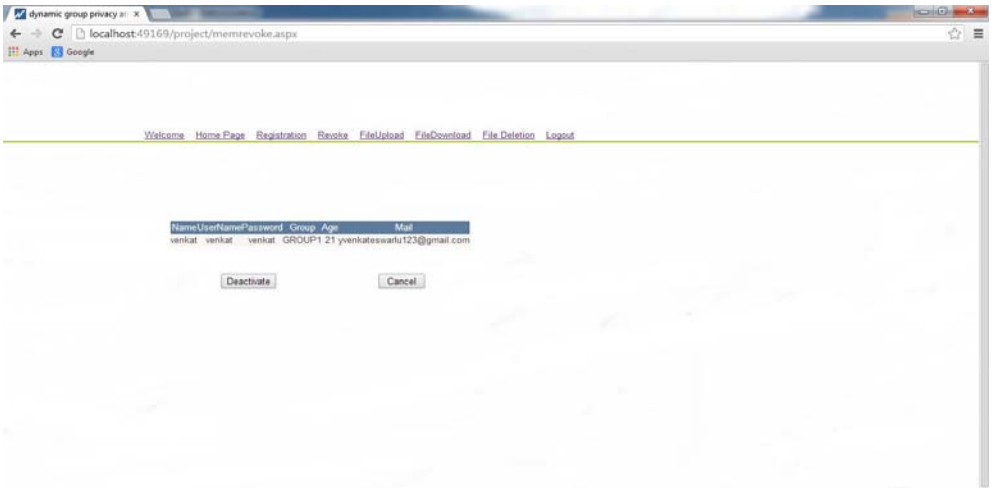

Fig 4 member deactivate page

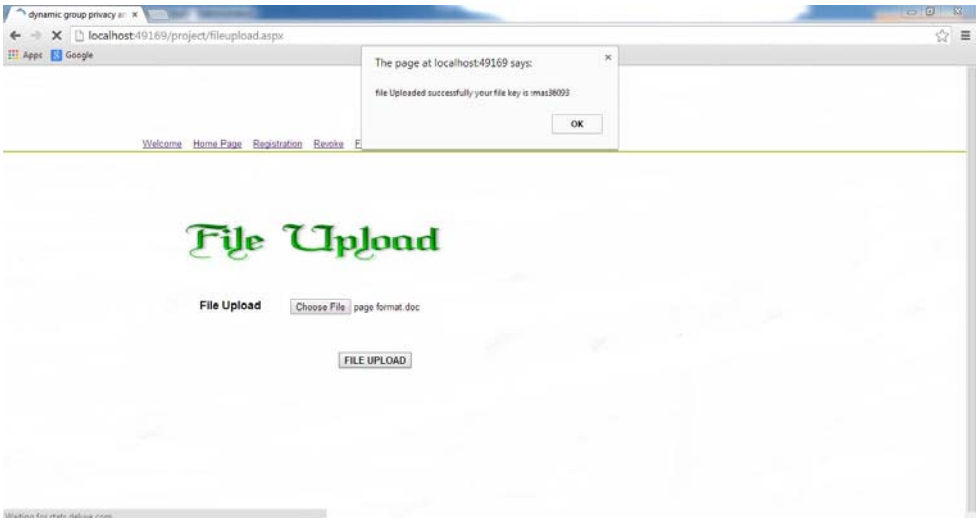

Fig 5 file upload page

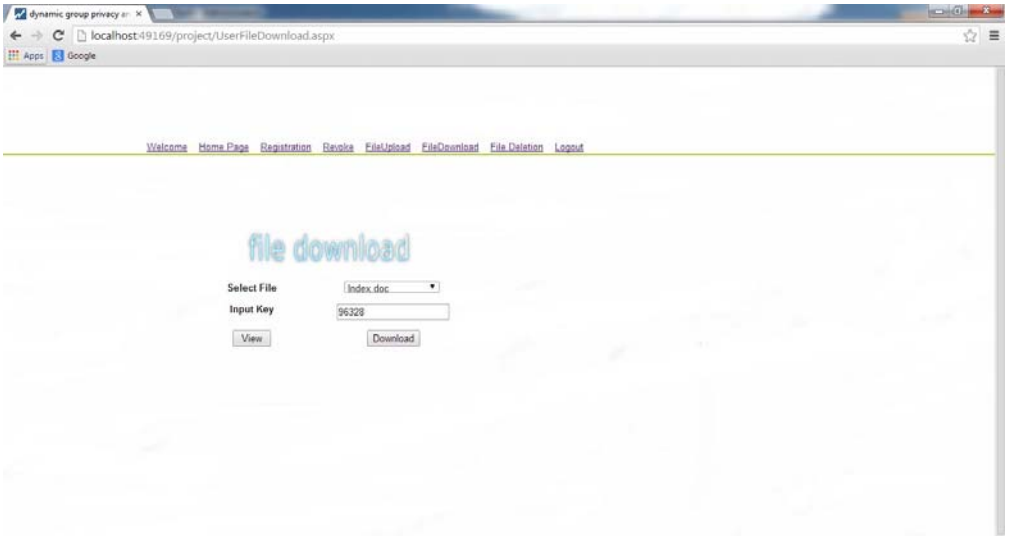

Fig 6 file download page

Fig 7 file view page

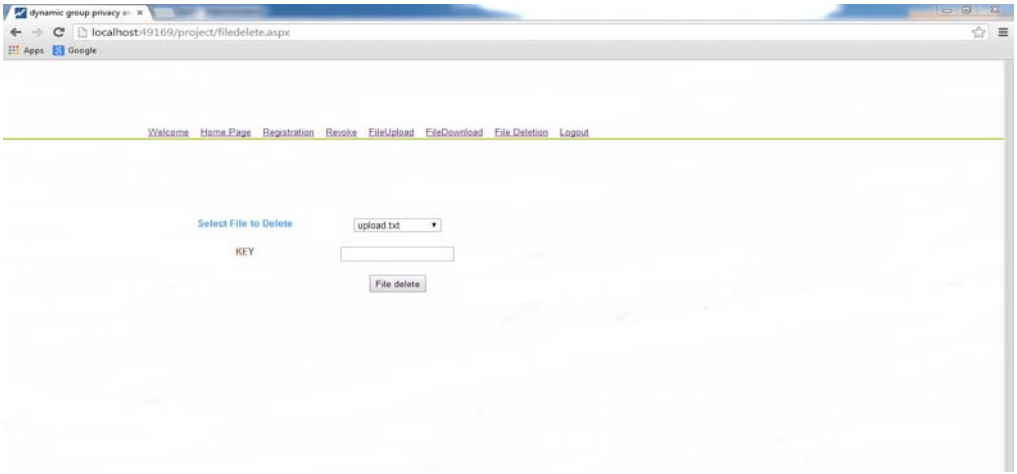

Fig 8 user file delete page

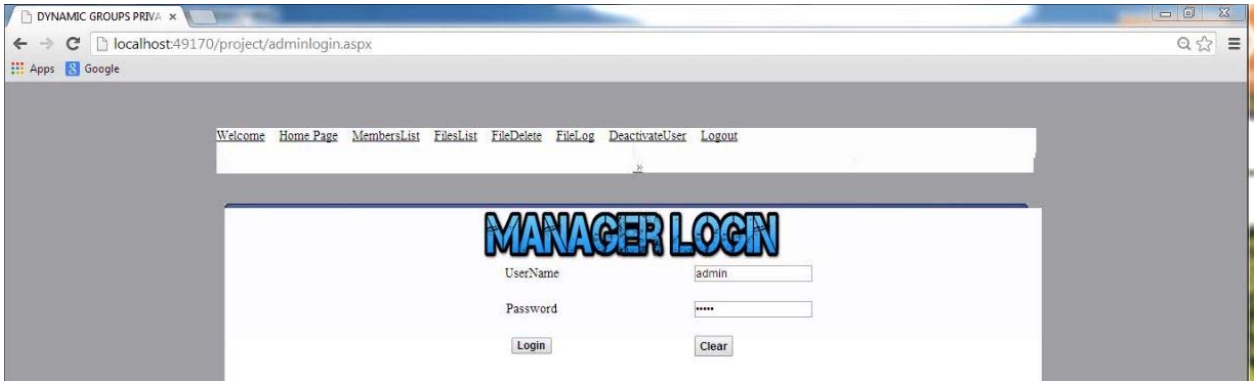

Fig 9 admin login page

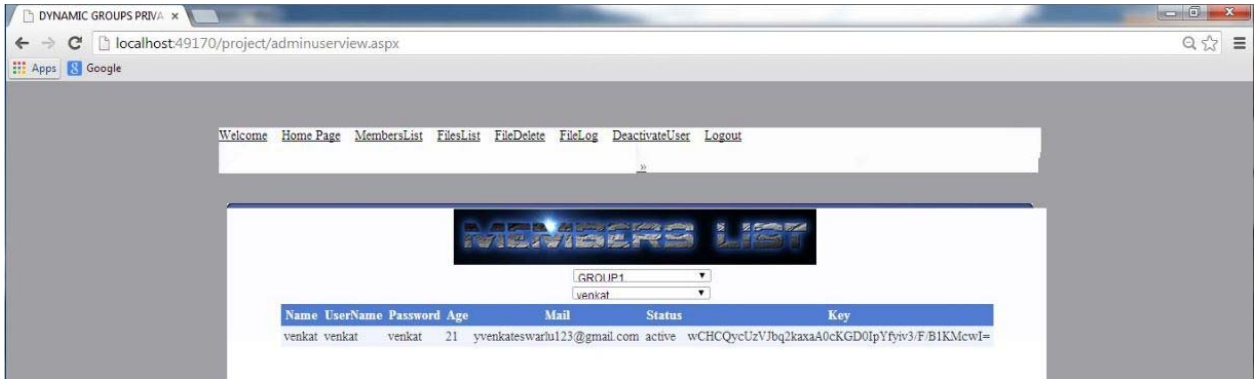

Fig 10 admin user view page

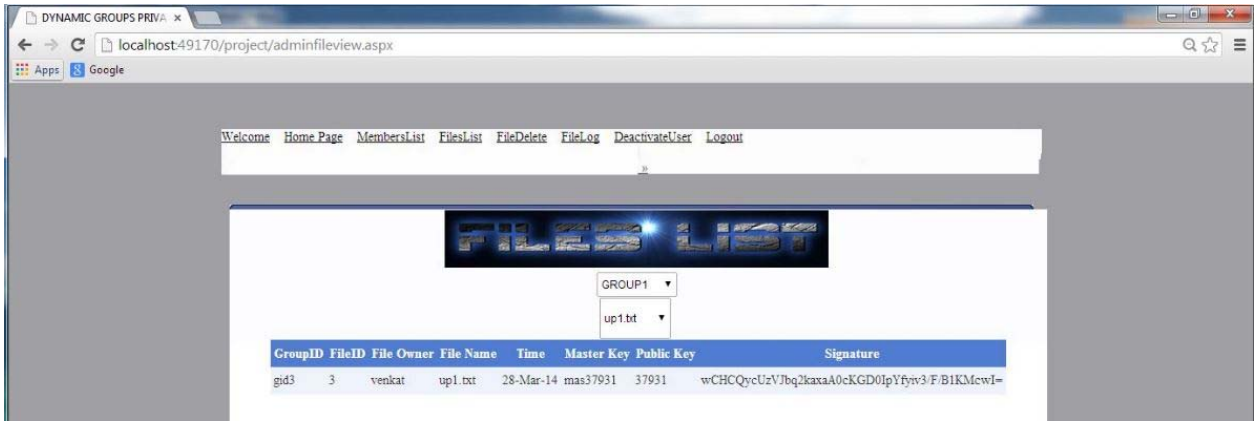

Fig 11 admin file view page

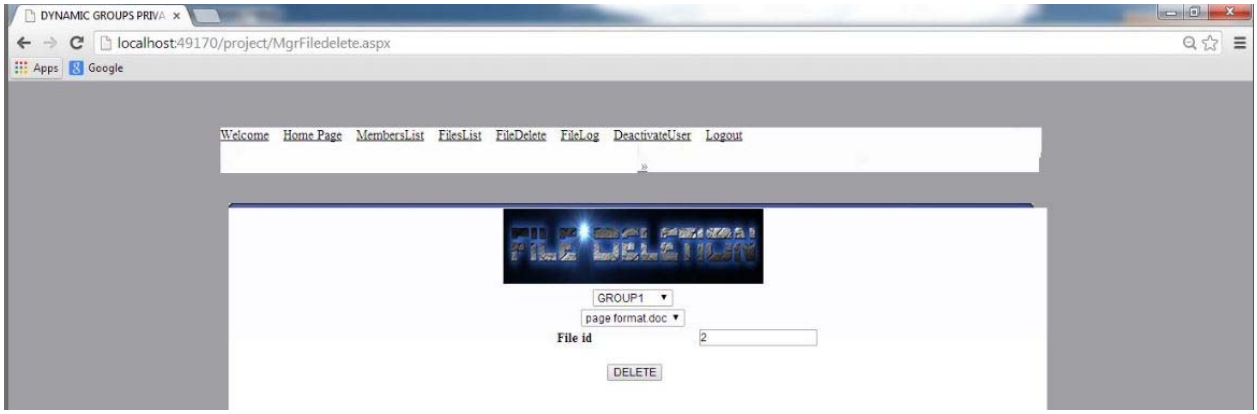

Fig 12 admin file delete page

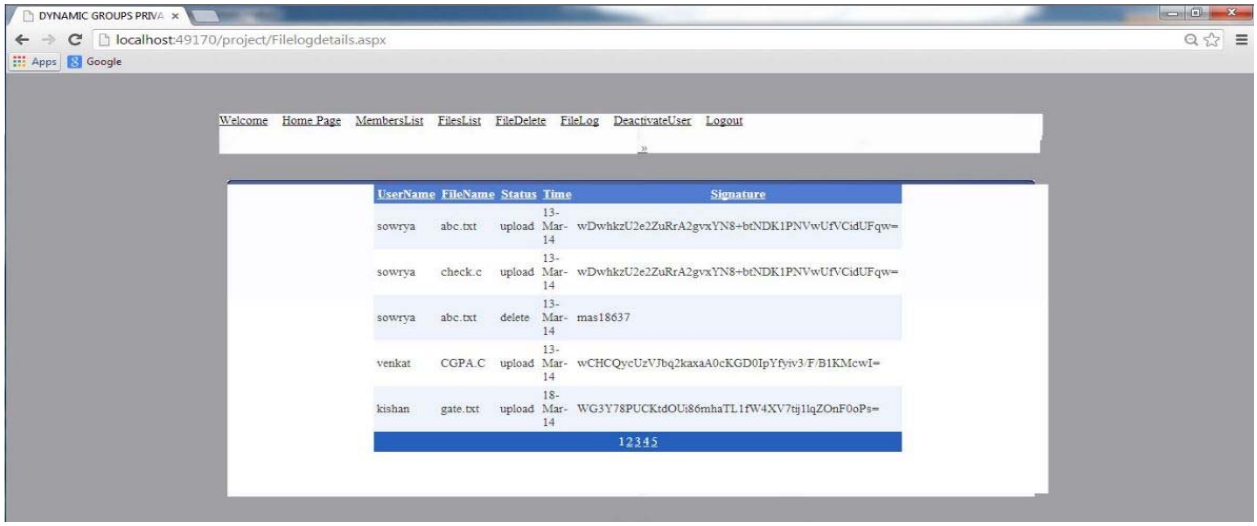

Fig 13 guilt agent detection page

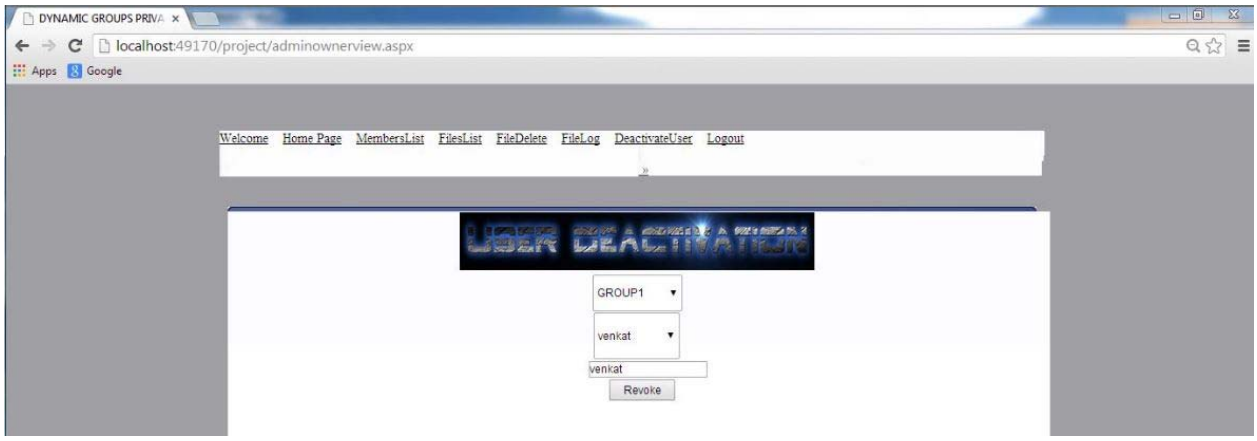

Fig 14 admin user deactivate page