Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

GROUP BASED USER REVOCATION USING KEY GENERATION

VASANTHAPRIYA.P[1], T.K.P.RAJAGOPAL[2]
[1]*Computer Science and Engineering, Kathir College of Engineering*
[2]*Computer Science and Engineering, Kathir College of Engineering*

Abstract-In the cloud with the help of data storage and sharing services, users can easily modify and share data as a group. In many cases the employees inside the organization itself is not trustful for their concern.So many threads are happening due to employees inside the organisation. In order to overcome this problem, we propose a new decentralized access control scheme for secure data storage in clouds that supports anonymous authentication. The cloud verifies the authenticity of the series without knowing the user's identity before storing data. Employee groups will be created, each group will be provided with a group key and each employee will be provided with a personal key. All the uploaded data will be stores in the centralized server. Our scheme also has the added feature of access control in which only valid users are able to decrypt the stored information. The scheme prevents data stored in the cloud from anonymous users. The revocation process will modify the key if those users must not have the ability to access data, even if they possess matching set of attributes. For this reason, the owners should change the stored data key and send updated information to other users.In order to access the file system the employee need to use their both group key and personal key. Both key will be get authenticated in the cloud server for data access and data transfer. The communication, computation, and storage overheads are comparable to centralized approaches. So that key will be changed in case of resigning of new users or introduction of new user. User resigned from the group cant able to create any threads to the existing groups.
Keywords-Anonymous, Computation, Data storage, User identity, User revocation.

I. INTRODUCTION

Nowadays security is less in group oriented applications. So to secure the confidential data we are going to use key transfer protocol.Key transfer protocol is fully trusted on KGC, where it will generate the key and pass it to all the group members in a safe and secure way.Admin will create a user, whenever user is created a 16 bit Alpha numeric key will generated along with user name and password and the details about these things will send to user's mail. So Admin will act as a KGC. Once that user is created, Admin only had a rights to allot the group for that user.Admin only had a rights to view the group details and to edit or update the user details.If any malfunction happens in a group, Admin can get the warning message by using IDS, so Admin had a right to block the sending rights of the user, so data transfer done in a secured way. A member in one group can send a file to a member in a same group or to another group. A file can be sent from one Group to another group so all the group member belongs to both the group can view the file. A safe group Communication will be done.

The user had a column like hackers list by using that they can find whether some body is tried to hack their profile, by clicking on that hackers list they can view hackers IP address and these details can be viewed by admin also. Admin can view all the details like sending details and receiving details of all the group members. If users have any request like adding any new member in a group or changing the user from one group to another means they can mail to admin. All the users in every group will be provided by a 16 bit key, by using this key the user can view their received

files. To send data from one group to another group they have a group key, by using that key they can send the data to all the group members in a safe and secure manner.

Users have rights to change their password. Users can view their files and download it. Whenever group member leave the group, the group key is regenerated dynamically so that that member cannot rejoin the group. Whenever a new member enters in to the group he/she can't able to retrieve the previous messages. Whenever a new member leaves the group he/she can't able to retrieve the messages by using key, because the key will regenerate dynamically.

II. OBJECTIVE

- The primary objective of this project is to create a cloud environment for Public auditing mechanism for integrity of shared data with user revocation in the cloud.
- The cloud will re sign the blocks signed by the revoked user.Creating Data Storage and Sharing services for public auditing
- The involved actors are Cloud, Public verifiers and users.
- Public Verifiers are the Client or third party auditor correctly checks for the integrity and verification of shared data.
- And Finally the cloud re signs blocks with a re signing when an user is revoked
- Uploaded data can be modified by other users, and when the user revoked the modified data get resigned by the cloud.

III. EXISTING SYSTEM

In the existing system the Panda concept is used for the integrity of shared data with efficient user revocation in the cloud.In our mechanism, by utilizing the idea of revoking process, once a user in the group is revoked, the cloud is able to resign the blocks of the revoked user.It is very important for the cloud to securely store and manage the datas of the group, so that the cloud can correctly and successfully convert datas from a revoked user to an existing user when it is necessary.As a result, the efficiency of user revocation can be significantly improved, and computation and communication resources of existing users can be easily saved.The mechanism is scalable, which indicates it is not only able to efficiently support a large number of users to share data.It also able to handle multiple auditing tasks simultaneously with batch auditing.

DISADVANTAGES OF EXISTING SYSTEM

- Huge amount of communication is available.
- Time will be delayed while verifying each blocks.
- It is tough to maintain correctness when number of re-signed blocks are large.
- More number of intruders and hackers are trying to collapse many networks.
- The employees inside the organization itself is not trustful for their concern. So many threads are happening due to employees inside the organisation.

IV. PROPOSED SYSTEM

We propose a new decentralized access control scheme for secure data storage in clouds that supports anonymous authentication. The cloud verifies the authenticity of the series without knowing the user's identity before storing data. file systems are used according to the employee rights, which can be decided by the admin. Employee groups will be created, each group will be provided with a group key and each employee will be provided with a personal key. All the uploaded data will be stores in the centralized server. Our scheme also has the added feature of access control in which only valid users are able to decrypt the stored information. The scheme prevents data stored in the cloud from anonymous users. These methods can be implemented using KDC (Key Distribution and certification) methods.In order to access the file system the employee need to use both group key and personal key. For data access and data transfer both key will be get authenticated in the cloud server. In case if any user is revoked from the group or new user is created the key will get changed.

ADVANTAGES OF PROPOSED SYSTEM

- A file can be sent from one Group to another group so all the group member belongs to both the group can view the file. A safe group Communication will be done.
- The user had a column like hackers list by using that they can find whether some body is tried to hack their profile, by clicking on that hackers list they can view hackers IP address and these details also can be viewed by admin.
- It increase the efficiency
- It impoves reliability

V. SYSTEM MODEL

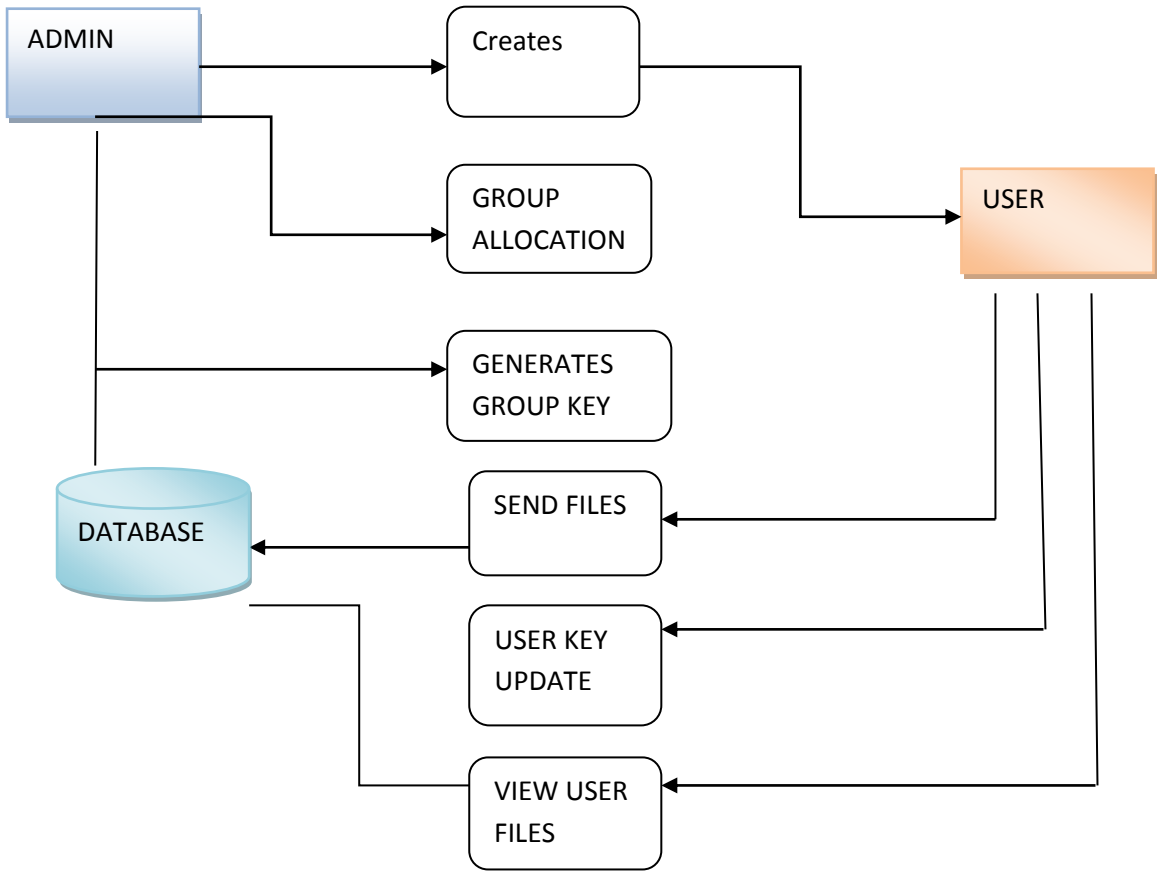

Fig 1:SYSTEM MODEL

VI METHODOLOGIES

6.1 HYBRID CRYPTOGRAPHY

A Computer Network is an interconnected group of autonomous computing nodes, which use a well defined, mutually agreed set of rules and conventions known as protocols, to interact with one-another meaningfully and allow resource sharing preferably in a predictable and controllable manner. Communication has a major impact on today's business. It is desired to communicate data with high security. With the rapid development of network technology, internet attacks are also versatile, the traditional encryption algorithms (single data encryption) is not enough for today's information security over internet, so we propose this hybrid Cryptograph Algorithm.

AES

Advanced Encryption Standard (AES) is based on a design principle known as a substitution-permutation network. It is fast in both software and hardware. Unlike its predecessor, DES, AES does not use a Feistel network . AES has a fixed block size of 128 bit and a key size of 128, 192, or 256 bit, whereas Rijndael has specified with block and key sizes in multiples of 32 bit, with a minimum of 128 bit. The block size has a maximum of 256 bit but the key size has no theoretical maximum AES operates on a 4×4 column-major order matrix of bytes, termed the state (versions of Rijndael with a larger block size have additional columns in the state). Most AES calculations are down in a special finite field.

The name "Hybrid" is used to show that the algorithm has built-in features that are inherited from data hiding techniques or "Steganography." The distinctive features of this algorithm are as follows:

• Key length is variable: the key length can be varied from 16 up to any larger value depending on the security level required.

• Word length is variable: the block size can be varied between 1 to 16 bit or 1 to 32 and so on. That is, encryption can be performed on 16 or 32 or 64 bit blocks. This, in turn, can be used on different processor architectures employing 16, 32, or 64 bit registers.

• The algorithm, therefore, provides variable degrees of security. However, this increased security level will be at the cost of increased size of the cipher-text.

• The number of rounds is variable: the whole process can be repeated r times using the same key. The method is quite suitable for hardware implementation employing Field programmable gate arrays (FPGA).

Encryption Process

$$k_{ij} \in \{1,2,3,4,5,6,7,8\} \begin{cases} \forall i = 1,...,L\ ; \mathrm{L} \geq 16 \\ \forall j = 1,2 \end{cases}$$

The method is reasonably simple. We have a key matrix KLx2 where,

This key is known only to the sender and receiver. When the first party wants to send a message M to the second party, he/she determines the key 2 L K×and every character from the message is replaced by a binary value. An eight-bit octet is generated randomly and set in a temporary vector V. the bits in the vector V from position K [1,1] to position K[1,2] are replaced by bits from the secret message. Then the resulting vector V is stored in a file. As long as the message file has not reached its end yet, we move to the next row of the key matrix and another octet is generated randomly and the replacement is performed repeatedly and the resulting vector is stored in the file. The previous procedure is repeated over and over again pending the end the message. The resulting file is sent to the receiver who beforehand has the key matrix. If the key length is not enough to cover the whole message during the encryption process, the key will be reapplied over and over again until the encryption of the whole message is completed. This formal algorithm is shown next.

The Decryption Process

For decrypting the received encrypted file the following steps are taken. An octet is read from the encrypted binary plain text message EBPM file, then it is set in a temporary vector V, from this vector, bits are extracted from position K(1,1) to position K(1,2) and set in a BPM file. Since the EBPM file is nonetheless not empty, the next octet is read from the EBPM file and then it is set in a temporary vector V. From this vector, bits are extracted from position K (2, 1) to position K (2, 2) and added to the binary plain text message BPM file. The above steps are repeated over and over again until the EBPM file becomes empty. Every octet form the BPM file is transformed to the corresponding character, and then is put in the plaintext file. When the EPBM is empty the plaintext file becomes the message. In case that the key length is not enough to cover the whole message

during the decryption process, the key will be reapplied over and over again till the decryption of the whole message is completed.

Key Length

Now we will show the number of possible keys, i.e., the key space when the key length is 16. The probability of replacing a string of bits whose length ranges from 1 to 8 bit in an octet is 1/64. Consequently, if the key length is 16 there are 64(16) = 7.9x10(28)possible keys. So we can say that if the attacker has a cipher text and he knows that the key length is 16, there are 7.9x10(28)attempts to find the correct key, i.e. , there are 7.9x10(28)attempts to find the correct plaintext or secret message.

Assuming that a supercomputer working in parallel is able to try 10(12) attempts per second, it will take 2.5x10(9) years to find the secret message. Note that the universe is only 10(10)years old. This eliminates brute force attack; however other types of attacks will be discussed in future work.

Algorithms Analysis

The worst case, regarding storage requirements, occurs when replacing one bit only from message to the V vector. Hence, cipher text equal eight times the size of the plain text. We have analyzed worst case running times for our encryption algorithm and found that it has linear complexity of O(n). Moreover, we have studied the following:

- Key length is variable: the key length can be varied from 16 up to any larger value depending on the security level required.
- Word length is variable: the block size can be varied between 1 to 16 bits or 1 to 32 bits and so on. That is, encryption can be performed on 16, 32 or 64 bit blocks. This, in turn, can be used on different processor architectures employing 16, 32, or 64 bit registers.
- The algorithm, therefore, provides variable degrees of security. However, this improved security levels will be at the cost of increased size of the cipher text.
- The number of rounds is variable: the whole process can be repeated r times using the same key

VII. CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

In the cloud with the help of data storage and sharing services (such as Dropbox and Google Drive), people can easily work together as a group by sharing data with each other. With shared data, once a user modifies a block, she also needs to compute a new signature for the modified block. Due to the modifications from different users, different blocks are signed by different users. For security reasons, when a user leaves the group or misbehaves, this user must be revoked from the group

In this paper, we propose a hybrid cryptography for key generation. When a new user is created simultaneously two keys will be generated automatically. One is Group key and other key is personal key. Only by using these key we can transfer the data .When a user in the group is revoked, Group key automatically changes and it is available only for users existing in the Group. This method increases the security and data confidentiality among the organisation.

7.2 FUTURE ENHANCEMENT

Logging plays a very important role in the proper operation of an organization’s information processing system. However, maintaining logs securely over long periods of time is difficult and expensive in terms of the resources needed. The emerging paradigm of cloud computing promises a more economical alternative. In this paper, we proposed a complete system to securely outsource log records to a cloud provider. We reviewed existing solutions and identified problems in the current operating system based logging services such as sys log and practical difficulties in some of the existing secure logging techniques. One of the unique challenges is the problem of log privacy that arises when we outsourced log management to the cloud. Log information in this case should not be casually linkable or traceable to their sources during storage, retrieval and deletion. We provided anonymous up-load, retrieve and delete protocols on log records in the cloud using the Tor network.

The protocols that we developed for this purpose have potential for usage in many different areas including anonymous publish-subscribe. Current implementation of the logging client is loosely coupled with the operating system based logging. In the future, we plan to refine the log client implementation so that it is tightly integrated with the OS to replace current log process. In addition, to address privacy concerns current implementation allows access to log records that are indirectly identified by upload-tag values. We plan to investigate practical homomorphism encryption schemes that will allow encryption of log records in such a way that the logging cloud can execute some queries on the encrypted logs without breaching confidentiality or privacy. This will greatly reduce the communication overhead between a log monitor and the logging cloud needed to answer queries on logs.

REFERENCES

[1] B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," in the Proceedings of IEEE Cloud 2012, 2012, pp. 295–302.

[2] B. Wang, B. Li, and H. Li, "Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud," in the Proceedings of ACNS 2012, June 2012, pp. 507–525.

[3] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," in the Proceedings of IEEE INFOCOM 2010, 2010, pp. 525–533.

[4] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing," in the Proceedings of ESORICS 2009. Springer-Verlag, 2009, pp. 355–370.

[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," in the Proceedings of ACM CCS 2007, 2007, pp. 598–610.

[6] H. Shacham and B. Waters, "Compact Proofs of Retrievability," in the Proceedings of ASIACRYPT 2008. Springer-Verlag,2008,pp. 90–107.

[7] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," in the Proceedings of ACM/IEEE IWQoS 2009, 2009, pp. 1–9.

[8] Y. Zhu, H.Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Dynamic Audit Services for Integrity Verification of Outsourced Storage in Clouds," in the Proceedings of ACM SAC 2011, 2011, pp. 1550–1557.

