International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 4, April - 2013

Shared Data Protection In Cloud Computing

Santhosh Guhan R[1], Shyamsundar D[2], Zia khan K[3],Mettildha Mary I[4]

UG Scholar, Department of Information Technology, Sri Ramakrishna Engineering College, Coimbatore, India[1,2,3]

Assistant Professor, Department of Information Technology, Sri Ramakrishna Engineering College, Coimbatore, India[4]

ABSTRACT: **The project deals with the protection of client's data residing in cloud. Client's data in cloud may be classified as (a) sensitive data (personal information such as documents) (b) Non Sensitive data (general data such as multimedia files). The sensitive data is protected by means of Blowfish algorithm. The Non sensitive data cannot be protected as the size may vary from user to user and may not contain much useful information but if this data is shared with the third party then there is a chance that the third party can insert some suspicious event tracking mechanism or some information that can be accessed secretly by the third party. In such case before the third party can access the client's data the access request is sent to the CSP who then verifies that the third party is genuine and then allows the access of the client's data. There may be n number of third parties which access the particular client's data so a data usage is maintained inside a text file which contains the name of the third parties and their IP's. The CSP can verify the third party's genuineness from the LOG file which is present on the server and it contains all the information about the third party. The client can also define the genuineness of the third party directly based on the method named "SAME IP SAME USER" if the third party's registered IP matches the IP of his system maximum number of times then the particular third party is verified as genuine. There is a concept called log files which record the activities of the client and if any user gains control over the client, since every activity is recorded in the log file the client can report it to the CSP and hence the CSP can then terminate the intruder.**

Keywords: CSP, Log files, Third party, Group policy

I. INDRODUCTION

Cloud computing is a computing platform where the users can store their data on a centralized data server which can be then accessed via internet. It is becomes a complicated task for the CSP to monitor the data on the server. The CSP must be able to provide integrity, security and reliable access of the client's data. The client is not aware that how his data is being used by him and does not have a proper record of the activities done by him on the cloud. Third parties in cloud who access the client's data have no separate record of the activities that are done on the client's data.[10] Thereby there are LOG files that contain the necessary information about the activities of the user like when was the last login and when the particular server file was accessed etc. These LOG files reside on the server side and contain information in a text format.

The proposed novel automatic and enforceable logging mechanism in the cloud. To our knowledge, this is the first time a systematic approach to data accountability through the novel usage of JAR files is proposed. Our proposed architecture is platform independent and highly decentralized, in that it does not require any dedicated authentication or storage system in place. The system goes beyond traditional access control in that the proposed system provide a certain degree of usage control for the protected data after these are delivered to the receiver. The results demonstrate the efficiency, scalability, and granularity of our approach. The detailed security analysis and discuss the reliability and strength of our the architecture

II. RELATED WORK

In this section, the first review related are addressing the privacy and security issues in the

cloud. Then, in brief the discussed works which
adopt similar techniques as our approach
serve for different purposes. Cloud computing has raised a range of important privacy and security issues [19], [25], [30]. Such issues are due to the fact that, in the cloud, users' data and applications reside—at least for a certain amount of time—on the cloud cluster which is owned and maintained by a third party.Concerns arise since in the cloud it is not always clear toindividuals why their personal information is requested or how it will be used or passed on to other parties. To date,little work has been done in this space, in particular with respect to accountability. Pearson et al. have proposed accountability mechanisms to address privacy concerns of end users [30] and then develop a privacy manager [31].Their basic idea is that the user's private data are sent to the cloud in an encrypted form, and the processing is done on
the encrypted data. The output of the processing is
deobfuscated by the privacy manager to reveal the correct result. However, the privacy manager provides only limited features in that it does not guarantee protection once the data are being disclosed. In [7], the authors present a layered architecture for addressing the end-to-end trust management and accountability problem in federated systems. The authors' focus is very different from ours, in that they mainly leverage trust relationships for accountability, along with authentication and anomaly detection.Further, their solution requires third-party services tocomplete the monitoring and focuses on lower level monitoring of system resources.Researchers have investigated accountability mostly as a provable property through cryptographic mechanisms,particularly in the context of electronic commerce [10], [21].A representative work in this area is given by [9]. The authorspropose the usage of policies attached to the data and present a logic for accountability data in distributed settings.Similarly, Jagadeesan et al. recently proposed a logic for designing accountability-based distributed systems [20]. In[10], Crispo and Ruffo proposed an interesting approach related to accountability in case of delegation. Delegation is complementary to our work, in that we do not aim at controlling the information workflow in the clouds. In a summary, all these works stay at a theoretical level and do not include any algorithm for tasks like mandatory logging. To the best of our knowledge, the only work proposing a distributed approach to accountability is from Lee and colleagues [22]. The authors have proposed an agent-based system specific to grid computing. Distributed jobs, along with the resource consumption at local machines are tracked by static software agents. The notion of accountability policies in [22] is related to ours, but it is mainly focused on resource consumption and on tracking of subjobs processed at multiple computing nodes, rather than access control.

III. PROBLEM STATEMENT

The illustration of the problem statement can be understood by the below example
Example 1. Alice, a professional photographer, plans to sell her photographs by using the SkyHigh Cloud Services.
For her business in the cloud, she has the following requirements:

- Her photographs are downloaded only by users who have paid for her services.
- Potential buyers are allowed to view her pictures first before they make the payment to obtain the download right.
- Due to the nature of some of her works, only users from certain countries can view or download some sets of photographs.
- For some of her works, users are allowed to only view them for a

limited time, so that the users cannot reproduce her work easily.[6]

- In case any dispute arises with a client, she wants to have all the access information of that client.

  She wants to ensure that the cloud service providers of SkyHigh do not share her data with other service providers, so that the accountability provided for individual users can also be expected from the cloud service providersWith the above scenarios in mind the goal of this paper is to solve all the above discrepancies. They are:

a. The first security issue to solve is the presence of the LOG files on both the server and client side. The presence of LOG file reduces the burden on pushing and pulling the data from the client.

b. The GROUP POLICY permissions which are assigned by the client on the requisition by the third party are actually verified by the CSP.[6]

c. The LOG files are either encrypted or a password protection mechanism is enabled so that the intruder cannot modify the files in order to generate the false activities done by the user

d. The use of the DATA TRACKER enables the client to know about how many people have used their data.

e. Sometime the CSP can misuse the data of the client suitable measuresto prevent the access of the client's valuable data have been taken i.e., the CSP has now to take permission to access the clients data[7].

IV. ALGORITHMS IMPLEMENTED

A. BLOWFISH ALGORITHM

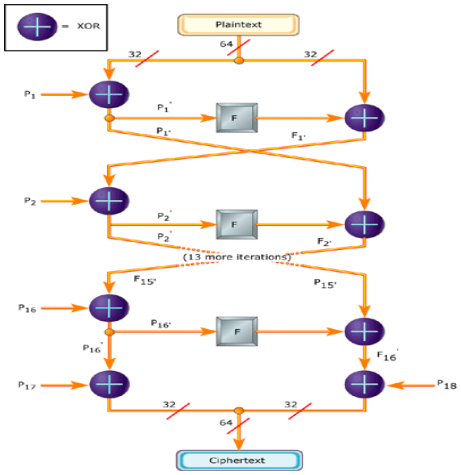

Fig.1 Blowfish algorithm

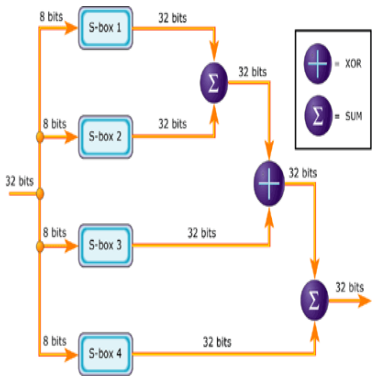

Fig.2 S-box Expansion

B. AES ALGORITHM

Input: Plain text
Output: Cipher text

1 Derive the set of round keys from the cipher key.
2 Initialize the state array with the block data (plaintext**).**
3 Add the initial round key to the starting state array.
4 Perform nine rounds of state manipulation.
5 Perform the tenth and final round of state manipulation.
6 Copy the final state array out as the encrypted data (cipher text).

V. SECURITY ISSUES

A. LOG FILE Attack

The most intuitive attack is that the attacker copies entire LOG files. The attacker may assume that doing so the client is unaware of the modification of the LOG file without being noticed by the data owner. However, such attack will be detected by our auditing mechanism. Recall that every activity is recorded in the LOG file. In particular, with the push mode, the harmonizer will send the logs to data owners periodically. That is, even if the data owner is not aware of the of the misusing third party, the LOG files will contain all the information of his activity, he will still be able to receive log files.

B. Man-in-the-Middle Attack

An attacker may intercept messages during the authentication of a service provider with the certificate authority, and reply the messages in order to masquerade as a legitimate service provider. There are two points in time that the attacker can replay the messages. One is after the actual service provider has completely disconnected and ended a session with the certificate authority. The other is when the actual service provider is disconnected but the session is not over, so the attacker may try to renegotiate the connection. The first type of attack will not succeed since the certificate typically has a time stamp which will become obsolete at the time point of reuse. The second type of attack will also fail since renegotiation is banned in the latest version of OpenSSL and cryptographic checks have been added.

C. Disassembling Attack

Another possible attack is to disassemble the LOG file of the logger and then attempt to extract useful information out of it or spoil the log records in it. Given the ease of disassembling LOG files, this attack poses one of the most serious threats to our architecture. Since we cannot prevent an attacker to gain possession of the LOG, we rely on the strength of the cryptographic schemes applied to preserve the integrity and confidentiality of the logs. Once the JAR files are disassembled, the attacker is in possession of the public IBE key used for encrypting the log files, the encrypted log file itself, and the *.class files. Therefore, the attacker has to rely on learning the private key or subverting the encryption to read the log records.

VI. PROPOSED CONCEPT

A. SENSITIVE DATA PROTECTION

The client's data resides on the cloud and hence can be misused by the CSP so in order to solve this problem the sensitive data is protected by means of BLOWFISH algorithm. The algorithm is very strong and has no known attacks so far. The algorithm offers both compression and password protection also.

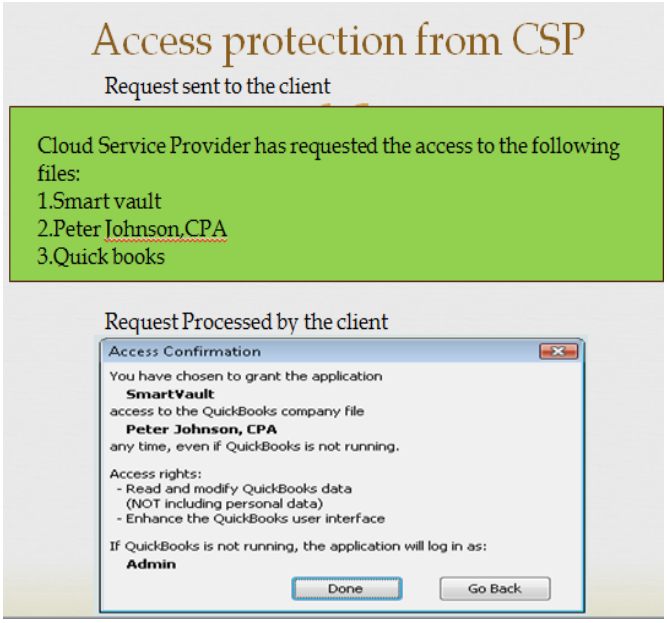

Fig.3 Sensitive data protection

B. GROUP POLICY PERMISSIONS (HYBRID CLOUD)

The policy for individual data is applied in the existing system. In group policy the

access policies are applied to the group(third parties). In case of hybrid clouds sometimes the CSP might not want the third party to access the data directly so the requests from the third parties are sent verified by the CSP. The CSP can verify or filter the third party based on LOCATION or based on IP addresses [3]. Thus the client can be very sure that the data accessed is by the genuine third party.

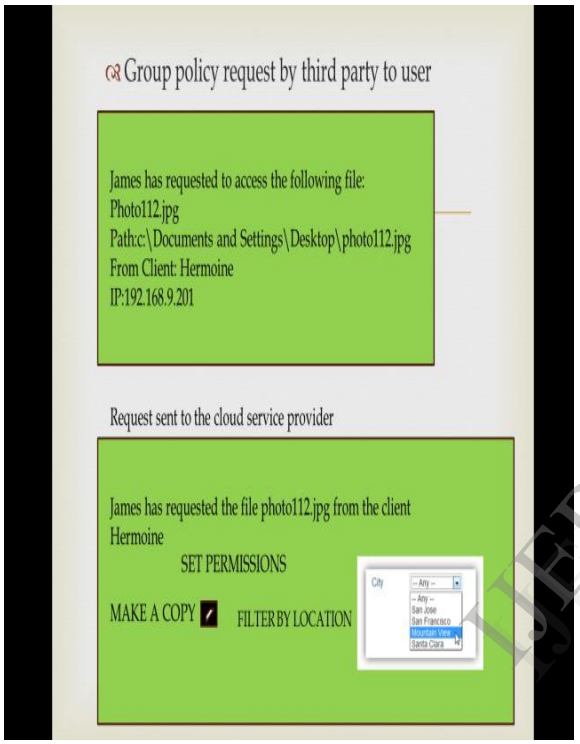

Fig. 4 Group policy permissions

C. DATA TRACKER

There may any number of third parties in the cloud and after the group policy permissions are verified and if the permission granted the data usage is recorded in a text file on the client side so that they can be aware of the third party using the data [10].

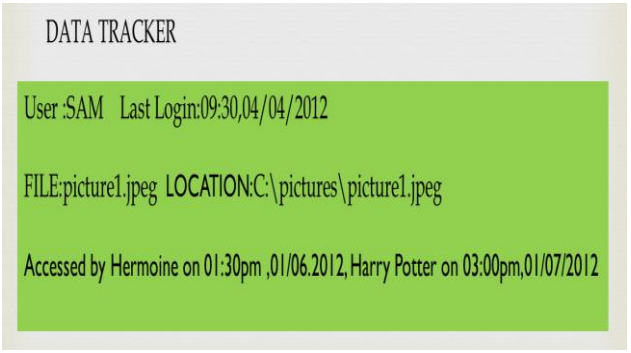

Fig. 5 Data Tracker

D. LOG FILES

The LOG files are the files that contain information of the user activities and they help the user to audit their activities and maintain a record in a text format which can then be viewed by the user as required. The LOG files are present on both the server and the client side. The following is an example a LOG file. The server side log files contains the all the details of the clients on the cloud. The log files store the login details and help to detect illegal logins

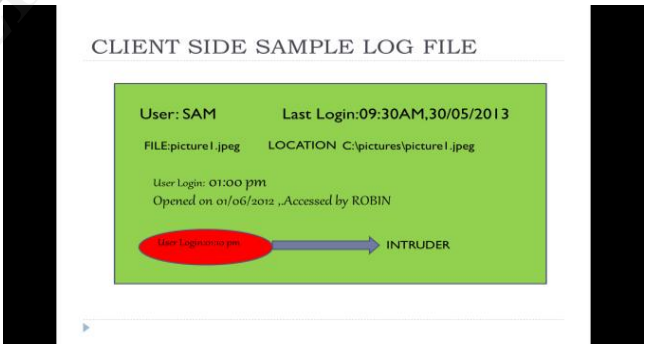

Fig. 6 Client Side Log File

Fig. 7 Server Side Log File

E. LOG FILE SECURITY

The LOG files are encrypted so that the user cannot view or modify the client's log

file. This is done so that the third party cannot gain advantage of the client's data and generate a false view of the activity of the user.. The log files though are plainly encrypted but the protection against the intruder is so strong that he/she cannot modify the log file. The AES algorithm is chosen because of the following reasons

- It is symmetric and fast
- It is highly secured
- It cannot be Brute forced
- It has a high efficiency

VII. CONCLUSION AND FUTURE RESEARCH

The proposed innovative approaches for automatically logging any access to the data in the cloud together with an auditing mechanism. The approach allows the data owner to not only audit his content but also enforce strong back-end protection if needed. Moreover, one of the main features of our work is that it enables the data owner to audit even those copies of its data that were made without his knowledge.

REFERENCES

[1] P. Ammann and S. Jajodia, "Distributed Timestamp Generation in Planar Lattice Networks," ACM Trans. Computer Systems, vol. 11, pp. 205-225, Aug. 1993.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z.Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. ACM Conf. Computer and Comm. Security, pp. 598- 609, 2007.

[3] E. Barka and A. Lakas, "Integrating Usage Control with SIP-Based Communications," J. Computer Systems, Networks, and Comm., vol. 2008, pp. 1-8, 2008.

[4] D. Boneh and M.K. Franklin, "Identity-Based Encryption from theWeil Pairing," Proc. Int'l Cryptology Conf. Advances in Cryptology,pp. 213-229, 2001.

[5] R. Bose and J. Frew, "Lineage Retrieval for Scientific Data Processing: A Survey," ACM Computing Surveys, vol. 37, pp. 1- 28, Mar. 2005.

[6] P. Buneman, A. Chapman, and J. Cheney, "Provenance Management in Curated Databases," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '06), pp. 539-550, 2006.

[7] B. Chun and A.C. Bavier, "Decentralized Trust Management and Accountability in Federated Systems," Proc. Ann. Hawaii Int'l Conf. System Sciences (HICSS), 2004.

[8] OASIS Security Services Technical Committee, "Security Assertion Markup Language (saml) 2.0," http://www.oasis-open.org/committees/tchome.php?wg abbrev=security, 2012.

[9] R. Corin, S. Etalle, J.I. den Hartog, G. Lenzini, and I. Staicu, "A Logic for Auditing Accountability in Decentralized Systems," Proc. IFIP TC1 WG1.7 Workshop Formal Aspects in Security and Trust, pp. 187-201, 2005.

[10] B. Crispo and G. Ruffo, "Reasoning about Accountability withinDelegation," Proc. Third Int'l Conf. Information and Comm. Security (ICICS), pp. 251-260, 2001.