Cloud Information Accountability Framework for Auditing the Data Usage in Cloud Environment

D.Dhivya[1,] S.CHINNADURAI [2]
1,M.E.(Cse), Srinivasan Engg College,Perambalur,Tamilnadu,India.
2,Ap/Cse, Srinivasan Engg College,Perambalur,Tamilnadu,India.

ABSTRACT:

Cloud computing enables highly scalable services to be easily consumed over the Internet on an as-needed basis. A major feature of the cloud services is that users' data are usually processed remotely in unknown machines that users do not own or operate. While enjoying the convenience brought by this new emerging technology, users' fears of losing control of their own data (particularly, financial and health data) can become a significant barrier to the wide adoption of cloud services. To address this problem, in this paper, a novel highly decentralized information accountability framework to keep track of the actual usage of the users' data in the cloud. In particular, an object-centered approach that enables enclosing our logging mechanism together with users' data and policies. Leverage the JAR programmable capabilities to both create a dynamic and traveling object, and to ensure that any access to users' data will trigger authentication and automated logging local to the JARs. To strengthen user's control, provide distributed auditing mechanisms. Extensive experimental studies that demonstrate the efficiency and effectiveness of the proposed approaches.

KEYWORDS: *Cloud computing, accountability, data sharing*

I. INTRODUCTION

Cloud computing presents a new way to supplement the current consumption and delivery model for ITservices based on the Internet, by providing for dynamically scalable and often virtualized resources as a service over the Internet. To date, there are a number of notable commercial and individual cloud computing services. Details of the services provided are abstracted from the users who no longer need to be experts of technology infrastructure. Moreover, users may not know the machines which actually process and host their data. While enjoying the convenience brought by this new technology, users also start worrying about losing control of their own data. The data processed on clouds are often out sourced, leading to a number of issues related to accountability, including the handling of personally identifiable information. Such fears are becoming significant barrier to the wide adoption of cloud services. Conventional access control approaches developed for closed domains such as databases and operating systems, or approaches using a centralized server in distributed environments, are not suitable due to the following features characterizing cloud environments. CIA framework provides end-to-end accountability in a highly distributed fashion. One of the maintaining lightweight and powerful accountability that combines aspects of access control, usage control and authentication. Two distinct modes for auditing: push mode and pull mode. The push mode refers to logs being periodically sent to the data owner or stakeholder while the pull mode refers to an alternative approach whereby the user can retrieve the logs as needed. First, Integrated integrity checks and oblivious hashing (OH) technique to our system in order to strengthen the dependability of our system in case of compromised JRE.Updated the log records structure to provide additional guarantees of integrity and authenticity. Second the security analysis to cover more possible attack scenarios. Third, the results of new experiments and provide a thorough evaluation of the system performance. Fourth, detailed discussion on related works to prepare readers with a better understanding of background knowledge. Finally, improved the presentation by adding more examples and illustration graphs.

II. RELATED WORK

2.1. Cloud Privacy and Security

Cloud computing has raised a range of important privacy and security issues. 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. Concerns arise since in the cloud it is not always clear to Individuals why their personal information is requested or it will be used or passed on to other parties. Their basic idea is that the user's private data are sent to the cloud in an encrypted form, and the processing is done other encrypted data. The output of the processing is deobfuscated by the privacy manager to reveal the correct result. The author's present 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 to complete the monitoring and focuses on lower level monitoring of system resources. The authors propose the usage of policies attached to the data and present logic for accountability data in distributed settings. Crispo and Ruffo proposed an interesting approach related to accountability in case of delegation. It is mainly focused on resource consumption and on tracking of sub jobs processed at multiple computing nodes, rather than access control.

2.2. Self-Defending Objects

Self-defending objects are an extension of the object-oriented programming paradigm, where software objects that offer sensitive functions or hold sensitive data are responsible for protecting those functions/data. Similarly, extend the concepts of object-oriented programming. The key difference in our implementations is that the authors still rely on a centralized database to maintain the access records, while the items being protected are held as separate files. Provided a Java-based approach to prevent privacy leakage from indexing, which could be integrated with the CIA framework proposed in this work since they build on related architectures.

2.3. Proof-Carrying Authentication

The PCA includes a high order logic language that allows quantification over predicates, and focuses on access control for web services. While related to ours to the extent that it helps maintaining safe, high-performance, mobile code, the PCA's goal is highly different from our research, as it focuses on validating code, rather than monitoring content. Another work is by Mont et al. who proposed an approach for strongly coupling content with access control, using Identity-Based Encryption (IBE) .We also leverage IBE techniques, but in a very different way. We do not rely on IBE to bind the content with the rules. Instead, we use it to provide strong guarantees for the encrypted content and the log files, such as protection against chosen plaintext and cipher text attacks.

III. SYSTEM MODEL

3.1. Data Owner Configuration Phase

In this module every data owner must register their details in the cloud server. And Cloud server establishes the public key and private key access policy using IBE scheme. Finally the cloud server distributes the secret key to the data owner. Cloud server stores the data owner details in the data store as a entity (it is key object model). It is persistence storage

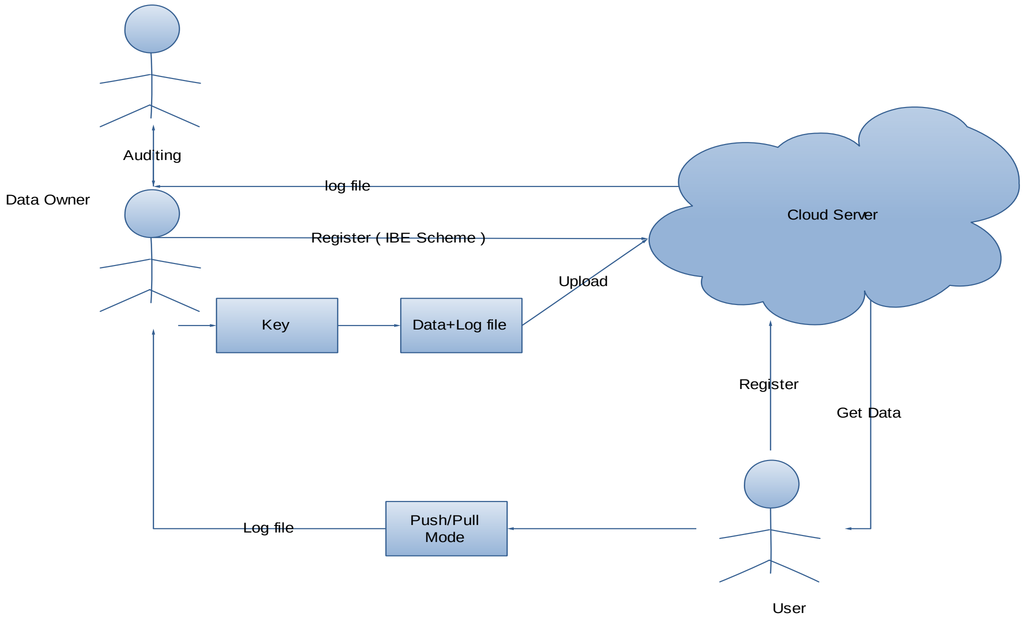

Fig 1: Overview architecture

3.2. Data Uploading in Cloud Server.

After configuration process completed data owner create the log file (it contain configuration details) and encrypt it using the secret key established by the cloud server to the particular data owner. Then load the owner data into encrypted log file. The owner data and log file is bounded or coupled together.

3.3. User Configuration Phase

In this module every user must register their details and account details in the cloud server. And Cloud server establishes the public key and private key access policy using IBE scheme. Finally the cloud server distributes the secret key to the user. Cloud server stores the user details in the data store as a entity (it is key object model). When the user request to get the data from the cloud server, the user get data with log file. This log file store the user session details and finally this log file send to data owner.

3.4. Auditing Phase.

In this module log harmonizer is used to perform the auditing work. This is maintained in the data owner. Data owner gets the log file information from the cloud server and user separately and decrypts the log files using the secret keys of data owner, and passes decrypted log files into the log harmonizer. Finally auditing process is conducted.

IV. EXPERIMENTAL RESULT

This experiment the time taken to create a log file and then measure the overhead in the system. With respect to time, the overhead can occur at three points: during the authentication, during encryption of a log record, and during the merging of the logs.

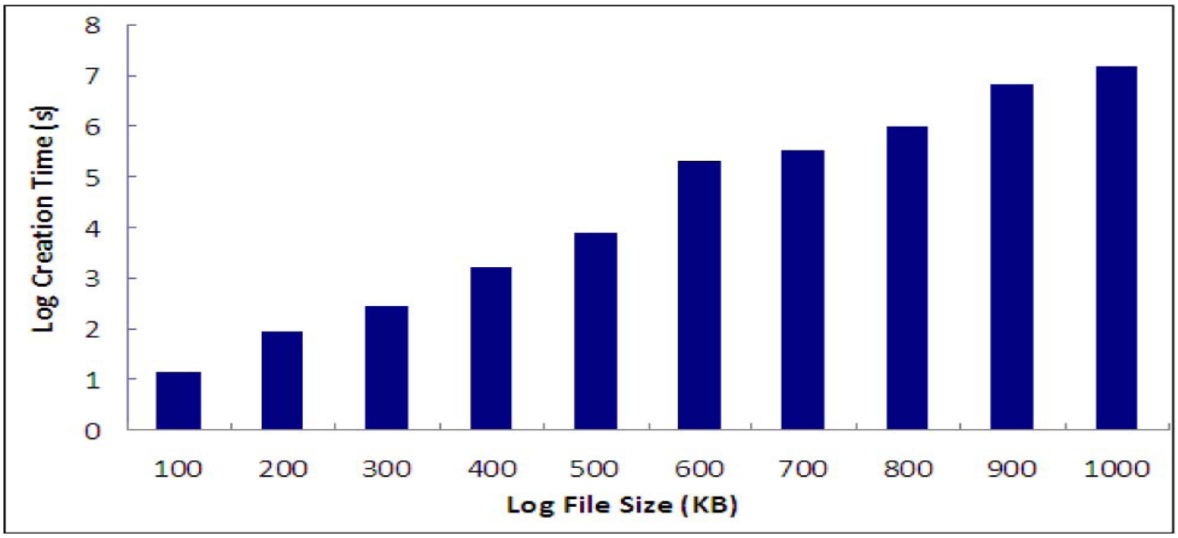

Fig 2: This result shows that time to create log files of different sizes.

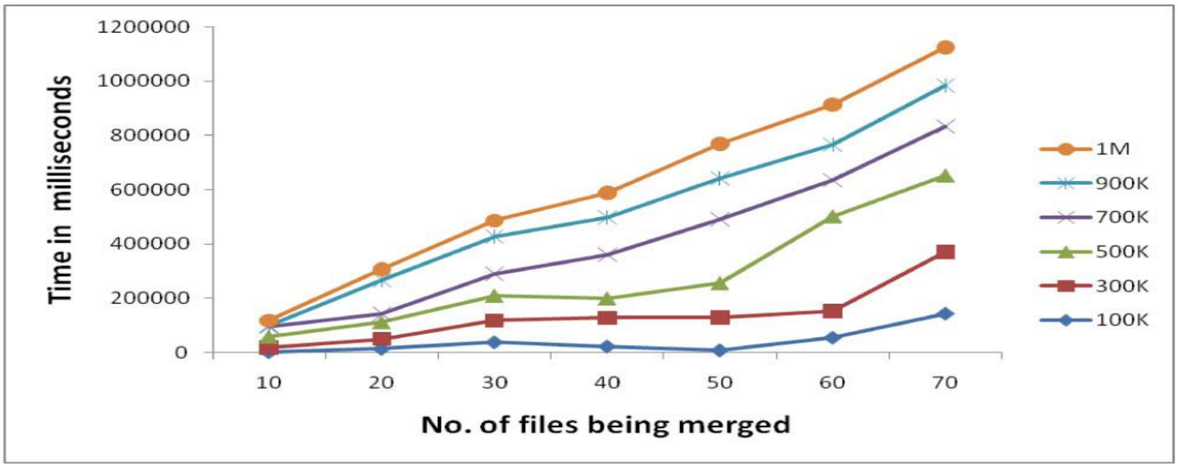

Fig 3: This result shows that time to merge log files

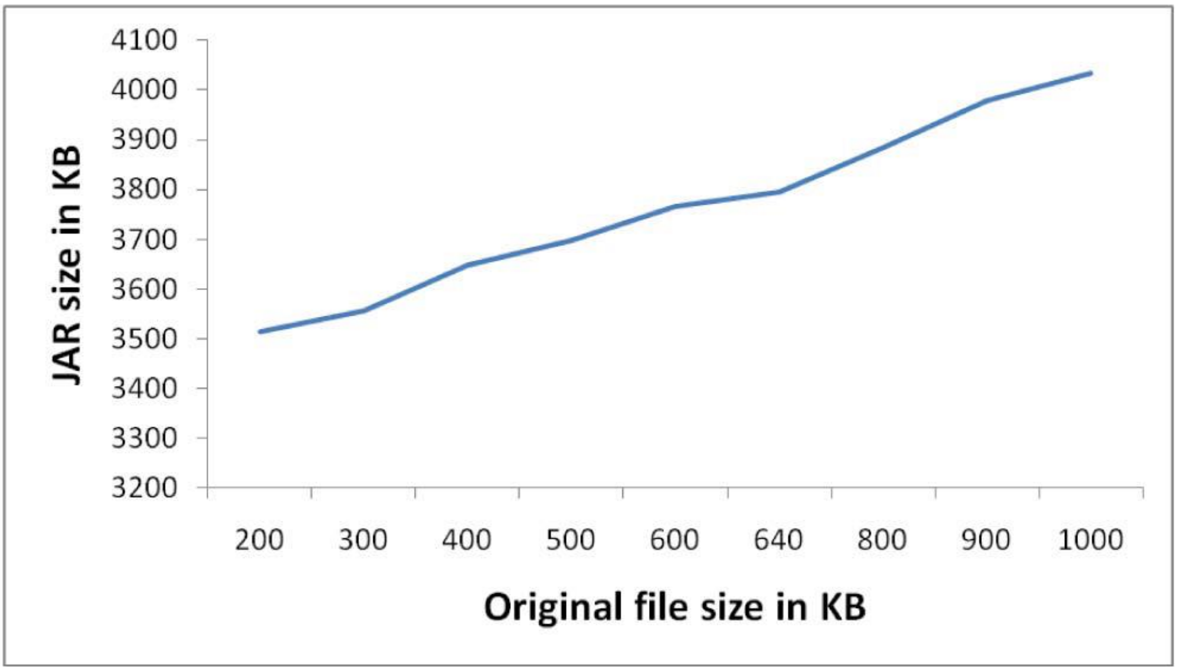

Fig 4: This result shows that size of the logger component.

V. CONCLUSION

Innovative approaches for automatically logging any access to the data in the cloud together with an auditing mechanism. 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. In future plan to refine our approach to verify the integrity of the JRE and the authentication of JARs. For example, investigate whether it is possible to leverage the notion of a secure JVM being developed by IBM. This research is aimed at providing software tamper resistance to Java applications. Design a comprehensive and more generic object-oriented approach to facilitate autonomous protection of traveling content. To support a variety of security policies, like indexing policies for text files, usage control for executables, and generic accountability and provenance controls.

ACKNOWLEDGEMENTS

This work was presented in part at the IEEE International Conference on Communications (ICC), 2012. This work was can be done in part of in our institution and support of all staff members.

REFERENCES

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

[2] 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.

[3] J.W. Holford, W.J. Caelli, and A.W. Rhodes, "Using Self-Defending Objects to Develop Security Aware Applications in Java," Proc. 27th Australasian Conf. Computer Science, vol. 26,pp. 341-349, 2004.

[4] P.T. Jaeger, J. Lin, and J.M. Grimes, "Cloud Computing and Information Policy: Computing in a Policy Cloud?," J. Information Technology and Politics, vol. 5, no. 3, pp. 269-283, 2009.

[5] J. Park and R. Sandhu, "Towards Usage Control Models: Beyond Traditional Access Control," SACMAT '02: Proc. Seventh ACM Symp. Access Control Models and Technologies, pp. 57-64, 2002.

[6] A. Pretschner, M. Hilty, and D. Basin, "Distributed Usage Control," Comm. ACM, vol. 49, no. 9, pp. 39-44, Sept. 2006.

[7] Sumitha Sundareswaran, Anna C. Squicciarini, "Ensuring Distributed Accountability for Data Sharing in the Cloud," IEEE Transactions on Dependable and secure computing, Vol 9. No.4, July/Aug 2012.

[8] S. Sundareswaran, A. Squicciarini, D. Lin, and S. Huang,"Promoting Distributed Accountability in the Cloud," Proc. IEEE Int'l Conf. Cloud Computing, 2011.

[9] D.J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigen-baum, J.Hendler, and G.J. Sussman, "Information Accountability," Comm.ACM, vol. 51, no. 6, pp. 82-87, 2008.

[10] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," Proc. European Conf. Research in Computer Security (ESORICS), pp. 355-370, 2009.

[11] M. Xu, X. Jiang, R. Sandhu, and X. Zhang, "Towards a VMMBased Usage Control Framework for OS Kernel Integrity Protection," SACMAT '07: Proc. 12th ACM Symp. Access Control Models and Technologies, pp. 71-80, 2007.

AUTHORS PROFILE

D.Dhivya received the B.E Degree computer science and engineering and now she is an M.E student in the Department of Computer Science & Engineering, Srinivasan Engineering College – Dhanalakshmi Srinivasan Group of Institutions, Perambalur, TN, India.

Her research interest includes Network Security and Cloud Computing.

S.Chinnadurai is working as Assistant Professor/CSE, Srinivasan Engineering College – Dhanalakshmi Srinivasan Group of Institutions, Perambalur, TN, India.

His research interest includes pervasive computing, Wireless Networks and Image Processing.