**International Journal of Emerging Engineering Research and Technology**
**Volume 2, Issue 9, December 2014, PP 1-7**
**ISSN 2349-4395 (Print) & ISSN 2349-4409 (Online)**

# Programmatically Signing of JAR Files

**S. Nageswara Rao[#1], G. Varaprasada Rao[#2]**

#1CSE Dept., Nova College of Engineering & Technology, Vegavaram,Jangareddy Gudem,
#2CSE Dept., Msc, Mphil, M-Tech, Associate Professor, Nova College of Engineering & Technology, Vegavaram. Jangareddy Gudem

**Abstract:** *Distributed computing empowers exceedingly versatile administrations to be effectively expended over the Internet on an as-required premise. A major gimmick of the cloud administrations is that clients' information are typically prepared remotely in obscure machines that clients don't claim or work. While getting a charge out of the accommodation brought by this new developing innovation, clients' apprehensions of losing control of their own information (especially, monetary and wellbeing information) can turn into a noteworthy hindrance to the wide appropriation of cloud administrations. To address this issue, in this paper, we propose a novel very decentralized data responsibility structure to stay informed concerning the real utilization of the clients' information in the cloud. Specifically, we propose an item focused approach that empowers encasing our logging instrument together with clients' information and arrangements. We influence the JAR programmable capacities to both make a dynamic and voyaging article, and to guarantee that any right to gain entrance to clients' information will trigger confirmation and robotized logging neighborhood to the Jars. To fortify client's control, we additionally give conveyed examining systems. We give far reaching exploratory studies that exhibit the effectiveness and adequacy of the proposed methodologies.*

**Index Terms:** *Cloud computing, accountability, data sharing*

## 1. INTRODUCTION

Distributed computing shows another approach to supplement the current utilization and conveyance model for IT administrations focused around the Internet, by accommodating progressively adaptable and regularly virtualized assets as an administration over the Web. To date, there are various striking business also individual distributed computing administrations, including Amazon, Google, Microsoft, Yahoo, and Salesforce [19]. Points of interest of the administrations gave are preoccupied from the clients who no more need to be masters of engineering base. In addition, clients may not know the machines which really process and host their information. While getting a charge out of the comfort brought by this new engineering, clients additionally begin stressing about losing control of their own information. The information transformed on mists are regularly outsourced, prompting various issues identified with responsibility, including the treatment of by and by identifiable data. Such apprehensions are getting to be a huge hindrance to the wide selection of cloud administrations [30]. To mollify clients' worries, it is key to give an compelling instrument for clients to screen the utilization of their information in the cloud. For instance, clients need to have the capacity to guarantee that their information are taken care of as indicated by the service level understandings set aside a few minutes they sign on for administrations in the cloud. Routine access control approaches produced for shut spaces, for example, databases and working frameworks, or methodologies utilizing a concentrated server in disseminated situations, are not suitable, because of the taking after peculiarities describing cloud situations. First and foremost, information taking care of can be outsourced by the immediate cloud administration supplier (CSP) to different elements in the cloud and theories elements can likewise appoint the undertakings to others, et cetera. Second, substances are permitted to join and leave the cloud in a adaptable way. Thus, information taking care of in the cloud goes through a mind boggling and element various leveled aministration chain which does not exist in routine situations. To defeat the above issues, we propose a novel approach, to be specific Cloud Information Accountability (CIA) schema, in light of the thought of data sponsibility[44]. Dissimilar to protection assurance innovations which are fabricated on the conceal it-or-lose-it point of view, data responsibility concentrates on keeping the information use straightforward and trackable. Our proposed CIA schema gives end-toend responsibility in an exceedingly appropriated design. One of the fundamental creative gimmicks of the CIA structure lies in its capacity of keeping

up lightweight and capable responsibility that consolidates parts of access control, utilization control what's more validation. By method for the CIA, information managers can track not just whether the administration level assentions are being respected, additionally authorize access and utilization control controls as required. Connected with the responsibility characteristic, we likewise create two different modes for reviewing: push mode what's more draw mode. The push mode alludes to logs being intermittently sent to the information holder or stakeholder while the pull mode alludes to an option approach hereby the client (on the other hand an alternate approved gathering) can recover the logs as required. The configuration of the CIA skeleton presents significant difficulties, including exceptionally recognizing Csps, guaranteeing the dependability of the log, adjusting to a profoundly decentralized foundation, and so on. Our essential methodology to tending to these issues is to power and amplify the programmable ability of JAR (Java Archives) documents to naturally log the use of the clients' information by any element in the cloud. Clients will send their information alongside any approaches, for example, access control strategies and logging approaches that they need to. implement, encased in JAR records, to cloud administration suppliers. Any right to gain entrance to the information will trigger a robotized and verified logging instrument nearby to the Jars. We allude to this sort of implementation as "solid tying" subsequent to the approaches and the logging instrument go with the information. This solid tying exists actually when duplicates of the Jars are made; in this manner, the client will have control over his information at any area. Such decentralized logging component meets the dynamic nature of the cloud additionally forces challenges on guaranteeing the trustworthiness of the logging. To adapt to this issue, we furnish the Jars with an essential issue of contact which structures a connection in the middle of them and the client. It records the slip adjustment data sent by the Jars, which permits it to screen the loss of any logs from any of the Jars. Also, on the off chance that a JAR is not ready to contact its main issue, any right to gain entrance to its encased information will be denied. As of now, we concentrate on picture records since pictures speak to an exceptionally basic substance sort for end clients and associations (as is demonstrated by the fame of Flickr [14]) and are progressively facilitated in the cloud as a major aspect of the stockpiling administrations offered by the utility figuring standard emphasized by distributed computing. Further, pictures frequently uncover social and individual propensities of clients, or are utilized for chronicling imperative records from associations. Furthermore, our methodology can handle individual identifiable data gave they are put away as picture records (they contain a picture of any literary content, for instance, the SSN put away as a .jpg record). We tried our CIA system in a cloud testbed, the Emulab testbed [42], with Eucalyptus as middleware [41]. Our analyses exhibit the productivity, adaptability and granularity of our methodology. Furthermore, we likewise give a point by point security investigation and examine the unwavering quality and quality of our structural planning even with different nontrivial assaults, dispatched by pernicious clients or because of bargained Java Running Environment (JRE). In rundown, our principle commitments are as per the following: . We propose a novel programmed and enforceable logging system in the cloud. As far as anyone is concerned, this is the first run through a precise methodology to information responsibility through the novel utilization of JAR records is proposed. . Our proposed structural planning is stage free furthermore exceptionally decentralized, in that it doesn't require any committed confirmation or capacity framework in place. . We go past conventional access control in that we give a certain level of use control for the secured information after these are conveyed to the recipient. . We lead investigates a true cloud testbed. The results show the effectiveness, versatility, also granularity of our methodology. We likewise give a nitty gritty security investigation and talk about the unwavering quality also quality of our structural engineering. This paper is an expansion of our past gathering paper [40]. We have made the accompanying new commitments. First and foremost, we coordinated honesty checks and careless hashing (Gracious) strategy to our framework with a specific end goal to fortify the constancy of our framework if there should be an occurrence of bargained JRE. We additionally overhauled the log records structure to give extra ensures of honesty and genuineness. Second, we augmented the security examination to cover more conceivable assault situations. Third, we report the aftereffects of new analyzes and give an intensive assessment of the framework execution. Fourth, we have included a point by point discourse on related attempts to get ready perusers with a finer understanding of foundation learning. At last, we have enhanced the presentation by including more samples and representation charts. Whatever remains of the paper is composed as takes after: Section 2 examines related work. Segment 3 lays out our issue explanation. Segment 4 displays our proposed Cloud Information Responsibility schema, and Sections 5 and 6 portray the point by point calculations for mechanized logging system furthermore evaluating

methodologies, separately. Segment 7 exhibits a security investigation of our skeleton, took after by a trial examine in Section 8. At last, Section 9 closes the paper and diagrams future exploration bearings.

## 2. EXISTING APPROACH

In this area, we first audit related works tending to the security and security issues in the cloud. At that point, we quickly examine meets expectations which receive comparable procedures as our approach however fill for diverse needs. Distributed computing has raised a scope of paramount protection furthermore security issues [19], [25], [30]. Such issues are because of the certainty that, in the cloud, clients' information and applications live in any event for a certain measure of time—on the cloud bunch which is possessed and kept up by an outsider. Concerns emerge following in the cloud it is not generally clear to people why their individual data is asked for or how it will be utilized or passed on to different gatherings. To date, little work has been carried out in this space, specifically with appreciation to responsibility. Pearson et al. have proposed responsibility instruments to address security concerns of end clients [30] and after that create a security supervisor [31]. Their fundamental thought is that the client's private information are sent to the cloud in an encoded structure, and the transforming is carried out on the encoded information. The yield of the preparing is deobfuscated by the security director to uncover the right result. On the other hand, the security supervisor gives just constrained emphasizes in that it doesn't promise insurance once the information are consistently revealed. In [7], the creators introduce a layered construction modeling for tending to the end-to-end trust administration and responsibility issue in combined frameworks. The creators' center is altogether different from our own, in that they essentially power trust connections for responsibility, alongside verification and aberrance identification. Further, their answer obliges outsider administrations to complete the checking and concentrates on lower level checking of framework assets. Scientists have researched responsibility basically as a provable property through cryptographic components, especially in the connection of electronic business [10], [21]. Arepresentative work here is given by [9]. The creators propose the use of strategies joined to the information and present a rationale for responsibility information in appropriated settings Thus, Jagadeesan et al. as of late proposed a rationale for planning responsibility based dispersed frameworks [20]. In [10], Crispo and Ruffo proposed an intriguing methodology. identified with responsibility if there should be an occurrence of assignment. Designation is correlative to our work, in that we don't go for controlling the data work process in the mists. In a synopsis, all these works stay at a hypothetical level and do exclude any calculation for errands like obligatory logging. To the best of our insight, the main work proposing a conveyed methodology to responsibility is from Lee and partners [22].

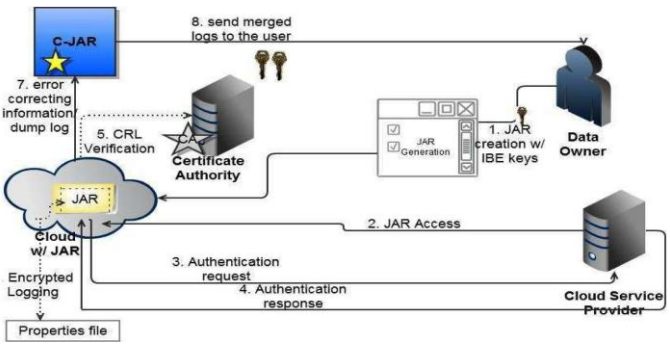


**Fig. 1.** *Overview of the cloud information accountability framework*

## 3. PROPOSED APPROACH

We start this segment by considering an illustrative illustration which serves as the premise of our issue proclamation and will be utilized all through the paper to show the principle peculiarities of our framework. In this area, we exhibit a review of the Cloud Data Accountability schema and talk about how the CIA schema meets the configuration necessities examined in the past area. The Cloud Information Accountability schema proposed in this work behaviors robotized logging and circulated examining of significant access performed by any substance,done anytime of time at any cloud administration supplier. It has two noteworthy parts: lumberjack and log harmonizer. There are two

noteworthy segments of the CIA, the first being the lumberjack, and the second being the log harmonizer. The lumberjack is the segment which is firmly coupled with the client's information, with the goal that it is downloaded when the information are gotten to, and is replicated at whatever point the information are duplicated. It handles a specific case or duplicate of the client's information and is in charge of logging access to that occurrence or duplicate. The log harmonizer structures the focal segment which permits the client access to the log records. The lumberjack is unequivocally coupled with client's information (either single or different information things). Its fundamental errands incorporate consequently logging access to information things that it contains, encoding the log record utilizing general society key of the content manager, and occasionally sending them to the log harmonizer. It might likewise be designed to guarantee that get to furthermore utilization control arrangements connected with the information are regarded. For instance, an information holder can determine that client X is just permitted to view however not to adjust the information. The lumberjack will control the information get to significantly after it is downloaded by client X.the lumberjack requires just negligible backing from the server (e.g., a legitimate Java virtual machine introduced) in place to be sent. The tight coupling in the middle of information and lumberjack, brings about an exceedingly appropriated logging framework, accordingly meeting our first plan prerequisite. Moreover, since the lumberjack does not have to be introduced on any framework or require any uncommon backing from the server, it is not exceptionally meddling in its activities, therefore fulfilling our fifth prerequisite. At last, the lumberjack is likewise in charge of producing the mistake revision data for each one log record and send the same to the log harmonizer onizer are both actualized as lightweight and convenient JAR records. The JAR record usage gives programmed logging capacities, which meets the second outline prerequisite.

**Require:** $size$: maximum size of the log file specified by the data owner, $time$: maximum time allowed to elapse before the log file is dumped, $tbeg$: timestamp at which the last dump occurred, $log$: current log file, $pull$: indicates whether a command from the data owner is received.
1: Let TS(NTP) be the network time protocol timestamp
2: $pull = 0$
3: $rec := \langle UID, OID, AccessType, Result, Time, Loc \rangle$
4: $curtime := TS(NTP)$
5: $lsize := sizeof(log)$ //current size of the log
6: **if** $((cutime - tbeg) < time) \&\& (lsize < size) \&\& (pull == 0)$ **then**
7: $log := log + ENCRYPT(rec)$ // ENCRYPT is the encryption function used to encrypt the record
8: $PING$ to CJAR //send a PING to the harmonizer to check if it is alive
9: **if** $PING$-CJAR **then**
10: PUSH $RS(rec)$ // write the error correcting bits
11: **else**
12: EXIT(1) // error if no PING is received
13: **end if**
14: **end if**
15: **if** $((cutime - tbeg) > time) || (lsize >= size) || (pull \neq 0)$ **then**
16: // Check if PING is received
17: **if** $PING$-CJAR **then**
18: PUSH $log$ //write the log file to the harmonizer
19: $RS(log) := NULL$ // reset the error correction records
20: $tbeg := TS(NTP)$ // reset the $tbeg$ variable
21: $pull := 0$
22: **else**
23: EXIT(1) // error if no PING is received
24: **end if**
25: **end if**

Push and pull PureLog mode.

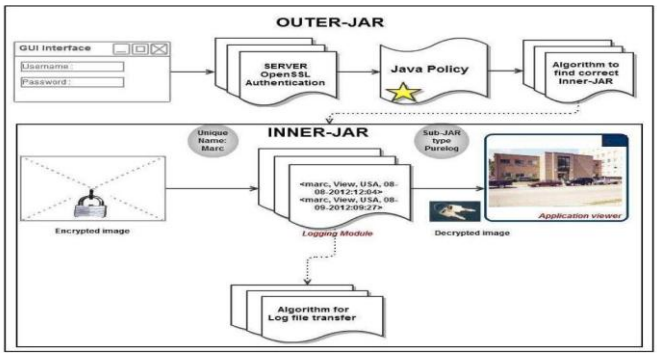


**Fig 2.** *The structure of the JAR file*

## 4. PERFORMANCE STUDY

In this section, we first introduce the settings of the test environment and then present the performance study of our system. We tested our CIA framework by setting up a small cloud using the Emulab testbed [42]. In particular, the test environment consists of several OpenSSL-enabled servers: one head node which is the certificate authority, and several computing nodes. Each of the servers is installed with Eucalyptus [41]. Eucalyptus is an open source cloud implementation for Linux-based systems. It is loosely basedon Amazon EC2, therefore bringing the powerful functionalities of Amazon EC2 into the open source domain. We used Linux-based servers running Fedora 10 OS. Each server has a 64-bit Intel Quad Core Xeon E5530 processor,4 GB RAM, and a 500 GB Hard Drive. Each of the servers is equipped to run the OpenJDK runtime environment with IcedTea6 1.8.2.

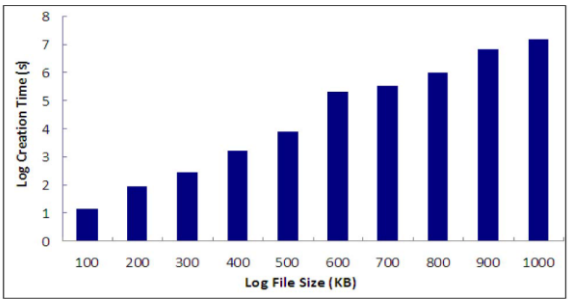


**Fig.3.** *Time to create log files of different sizes*

## 5. EXPERIMENTAL RESULTS

In the examinations, we first analyze the time taken to make a log document and afterward measure the overhead in the framework. Regarding time, the overhead can happen at three focuses: amid the confirmation, amid encryption of a log record, and amid the fusing of the logs. Additionally, with admiration to capacity overhead, we perceive that our construction modeling is exceptionally lightweight, in that the main information to be put away are given by the real documents and the related logs. Further, Container go about as an issue of the documents that it handles. In specific, as presented in Section 3, numerous documents can be taken care of by the same lumberjack part. To this degree, we explore whether a solitary lumberjack part, used to handle more than one document, brings about capacity overhead.

**Log Creation Time:** In the first round of tests, we are intrigued by discovering the time taken to make a log document when there are elements constantly getting to the information, bringing on ceaseless logging. Results are indicated in Fig. 5. It is not amazing to see that the time to make a log document increments straightly with the size of the log document. Particularly, the time to make a 100 Kb record is around 114.5 ms while the time to make a 1 MB document midpoints at 731 ms. With this test as the pattern, one can choose the measure of time to be defined between dumps, keeping different variables like space demands or system activity in mind authentication Time The following point that the overhead can happen is amid the verification of a CSP. On the off chance that the time taken for this verification is excessively long, it may turn into a bottleneck for getting to the encased information. To assess this, the head hub issued Openssl authentications for the registering hubs and we measured the aggregate time for the Openssl authenticationto be finished and the authentication

disavowal to be checked. Thinking of one as access at the time, we find that the validation time midpoints around 920 ms which demonstrates that not all that much overhead is included amid this stage. As of present, the validation happens each one time the CSP necessities to get to the information. The execution can be further enhanced by reserving the testaments. The time for verifying an end client is about the same when we consider just the activities needed by the JAR, viz. getting a SAML testament and after that assessing it. This is since both the Openssl and the SAML declarations are taken care of in a comparable manner by the JAR. When we consider the client activities (i.e., submitting his sername to the JAR), it midpoints at 1.2 minutes.

## 6. CONCLUSION AND FUTURE RESEARCH

We proposed inventive methodologies for consequently logging any right to gain entrance to the information in the cloud together with a reviewing system. Our methodology permits the information manager to review his substance as well as implement solid back-end insurance if necessary. Besides, one of the primary gimmicks of our work is that it empowers the information manager to review even those duplicates of its information that were made without his insight. Later on, we want to refine our methodology to confirm the respectability of the JRE and the confirmation of Jars [23]. For sample, wewill explore whether it is conceivable to power the thought of a protected JVM [18] being created by IBM. This examination is gone for giving programming alter safety tojava applications. In the long haul, we want to outline a thorough and more bland item arranged methodology to encourage self-ruling security of voyaging substance. We might want to help an assortment of security approaches, as indexing approaches for content documents, utilization control for executables, also non specific responsibility and provenance controls.

## 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/tc home.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 within Delegation,” Proc. Third Int’l Conf. Information and Comm. Security (ICICS), pp. 251-260, 2001.

[11] Y. Chen et al., “Oblivious Hashing: A Stealthy Software Integrity Verification Primitive,” Proc. Int’l Workshop Information Hiding, F. Petitcolas, ed., pp. 400-414, 2003.

[12] S. Etalle and W.H. Winsborough, “A Posteriori Compliance Control,” SACMAT ’07: Proc. 12th ACM Symp. Access Control Models and Technologies, pp. 11-20, 2007.

[13] X. Feng, Z. Ni, Z. Shao, and Y. Guo, “An Open Framework for Foundational Proof-Carrying Code,” Proc. ACM SIGPLAN Int’l Workshop Types in Languages Design and Implementation, pp. 67-78, 2007.

[14] Flickr, http://www.flickr.com/, 2012.

[15] R. Hasan, R. Sion, and M. Winslett, "The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance," Proc. Seventh Conf. File and Storage Technologies, pp. 1-14, 2009.

[16] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," Computer, vol. 34, no. 8, pp. 57-66, Aug. 2001.

[17] 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.

[18] Trusted Java Virtual Machine IBM, http://www.almaden.ibm.com/cs/projects/jvm/, 2012.

[19] 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.

[20] R. Jagadeesan, A. Jeffrey, C. Pitcher, and J. Riely, "Towards a Theory of Accountability and Audit," Proc. 14th European Conf. Research in Computer Security (ESORICS), pp. 152-167, 2009.

[21] R. Kailar, "Accountability in Electronic Commerce Protocols," IEEE Trans. Software Eng., vol. 22, no. 5, pp. 313-328, May 1996.

[22] W. Lee, A. Cinzia Squicciarini, and E. Bertino, "The Design and Evaluation of Accountable Grid Computing System," Proc. 29th IEEE Int'l Conf. Distributed Computing Systems (ICDCS '09), pp. 145-154, 2009.