Tamper Proof JAR Client for Distributed Accountability for Data Sharing in the Cloud

S.Jaya Prakash #1, Dr K.Subramanyam#2,U.D.S.V. Prasad #3,
#1 Student, K.L.University,Vaddeswaram,Guntur(dt),
#2 Professor, K.L.University,Vaddeswaram,Guntur(dt),
#3 Student, K.L.University,Vaddeswaram,Guntur(dt),
#1prakashsunkavalli@gmail.com,#2 smkodukula@kluniversity.in, #3 prasadudsv@gmail.com .

Abstract: Consuming on-demand highly scalable services is an essential feature of cloud computing that liberates cloud consumer from unnecessary software or hardware dependencies. While enjoying the convenience brought by cloud technology, user's fear of losing control of their own data (particularly, financial and health data) can become a significant barrier to the wide adoption of cloud services. Remote storage of users data in cloud opens up new challenges such as lack of control over data and security. To address this problem, previously a novel approach, namely Cloud Information Accountability (CIA) framework, based on the notion of information accountability to keep track of the actual usage of the users' data in the cloud was proposed. In particular, this framework uses object-entered approach that enables enclosing clients logging mechanism together with user's data and policies. It leverages the said so by developing a JAR client that has ability to ensure that any access to users' data will trigger authentication and automated logging local to the JARs. For further user's control, distributed auditing mechanisms were provided. By means of the CIA, data owners can track not only whether or not the service-level agreements are being honored, but also enforce access and usage control rules as needed. However the JAR client itself is not tamper proof which is still an open issue. So we propose to use JAR signing approach to prevent tampering. Security is controlled by the security policy that's in force at runtime. Cloud provider configures the policy to grant security privileges to JAR clients. An additional element is the certificate that the signer includes in a signed JAR file. It is a digitally signed statement from a recognized certification authority that indicates who owns a particular public key. A certification authority is entities (generally commercial bodies specialized in digital security) that are trusted throughout the industry to sign and issue certificates for keys and their owners. In the case of signed JAR files, the certificate indicates who owns the public key contained in the JAR file thus increasing accountability in cloud architecture. We provide extensive experimental studies that demonstrate the efficiency and effectiveness of the proposed approaches.

I INTRODUCTION

Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable, confidential and accountable infrastructure for application, data and file storage. With the advent of this technology, the cost of computation, application hosting, content storage and delivery is reduced significantly.

Cloud computing is the style of computing where massively scaled IT related capabilities are provided as a service across the internet to multiple external customers and are billed by consumption. Many cloud computing providers have popped up and there is a considerable growth in the usage of this service. Google, Microsoft, Yahoo, IBM and Amazon have started providing cloud computing services. Amazon is the pioneer in this field. Smaller companies like SmugMug, which is an online photo hosting site, has used cloud services for the storing all the data and doing some of its services.

Cloud computing provides the facility to access shared resources and common infrastructure, offering services on demand over the network to perform operations that meet changing business needs. The location of physical resources and devices being accessed are typically not known to the end user. It also provides facilities for users to develop, deploy and manage their applications „on the cloud‟, which entails virtualization of resources that maintains and manages itself.

Existing system uses web applications driven with cloud architecture. Uses SAAS based cloud computing applications over web that acts as a cloud server. The client is definitely user's browser. Users need to be able to ensure that their data are handled according to the service level agreements made at the time they sign on for services in the cloud which is a manual process or at times not possible at all. 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.

- First, data handling can be outsourced by the direct cloud service provider (CSP) to other entities in the cloud and theses entities can also delegate the tasks to others, and so on.
- Second, entities are allowed to join and leave the cloud in a flexible manner. As a result, data handling in the cloud goes through a complex and dynamic hierarchical service chain which does not exist in conventional environments.

In existing system, data usage is not transparent and not trackable in the user's hierarchy. So, A better system is required for an optimized saas cloud performance for content sharing in a client's perspective.

Later, System uses web applications driven with cloud architecture. They also uses SAAS based cloud computing applications over web that acts as a cloud server. They propose a novel approach, namely Cloud Information Accountability (CIA) framework, based on the notion of information accountability. The client is definitely JAR(Java Archive) application to enable client side access policy sharing. Information accountability focuses on keeping the data usage transparent and trackable. CIA framework lies in its ability of maintaining lightweight and powerful accountability that combines the following aspects

- Access control
- Usage control
- Authentication.

By means of the CIA, data owners can track not only whether or not the service-level agreements are being honored, but also enforce access and usage control rules as needed.

In this paper, we propose to use JAR signing approach to prevent tampering. Cloud provider configures the policy to grant security privileges to JAR clients. Digitally signed JAR's ensure further security and accountability. Security is controlled by the security policy that's in force at runtime. Cloud provider configures the policy to grant security privileges to JAR clients. An additional element is the certificate that the signer includes in a signed JAR file. It is a digitally signed statement from a recognized certification authority that indicates who owns a particular public key. A certification authority is entities (generally commercial bodies specialized in digital security) that are trusted throughout the industry to sign and issue certificates for keys and their owners. In the case of signed JAR files, the certificate indicates who owns the public key contained in the JAR file thus increasing accountability in cloud architecture.

II RELATED WORK

Chen and Wang have a team looking at "accountability as a service" for the cloud [1]. Their work presented a prototype which enforces accountability of service providers whose services are deployed in the cloud. This is achieved by making the service providers responsible for faulty services and a technique which allows identification of the cause of faults in binding Web services.

R. Corin et al. [2] gives a language which permits to serve data with policies by agent; agent should prove their action and authorization to use particular data. In this logic data owner attach Policies with data, which contain a description of which actions are allowed with which data, but there is the problem of Continuous auditing of agent, but they provide solution that incorrect behavior. Should monitor and agent should give justification for their action, after that authority will check the justification.

Muniswamy-Reddy et al. [3] discuss the main challenges of provenance adoption for cloud computing and suggest four properties (i.e. data coupling, multi-object casual ordering, dataindependent persistence, and efficient querying) that make provenance systems truly useful.

Buneman et al. [4] consider the notion of data provenance in the context of data management systems and propose a sub-classification into why and where-provenance. Why-provenance captures why a data item is in a query result, while where provenance explains where the data item came from.

S. Pearson et al. [5] describes privacy manager mechanism in which user's data is safe on cloud , in this technique the user's data is in encrypted form in cloud and evaluating is done on encrypted data, the privacy manager make readable data from result of evaluation manager to get the correct result.

Squicciarini et al. [6] gives a three layer architecture which protect information leakage from cloud, it provides three layer to protect data, in first layer the service provider should not view confidential data in second layer service provider should not do the indexing of data, in third layer user specify use of his data and indexing in policies, so policies always travel with data.

Chun and Bavier et al. [7] present accountability in federated system to achieve trust management. The trust towards use of resources is accomplished through accountability so to resolve problem for trust management in federated system they have given three layers architecture, in first layer is authentication and authorization in this authentication does using public key cryptography. Second layer is accountability which perform monitoring and logging. The third layer is anomaly detection which detects misuse of resources. This mechanism requires third party services to observe network resources.

III CLOUD TYPES

Public cloud: A large organization owns the cloud infrastructure and sells cloud services to industries or public. Public cloud applications, storage, and other resources are made available to the general public by a service provider. These services are free or offered on a pay-per-use model.

Community cloud: Several organizations that have similar polices, objectives, aims and concerns share the cloud infrastructure. The common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third-party and hosted internally or externally.

Hybrid cloud: Hybrid cloud is a composition of two or more clouds (private, community or public) that remain unique entities but are bound together, offering the benefits of multiple deployment models. It enables data and application probability.

Private cloud: The cloud infrastructure is owned or leased by a single organization and is operated solely for that organization Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party and hosted internally or externally.

IV CLOUD ACCOUNTABILITY

A classification of the different phases may also help researchers to focus on specific research sub-problems of the large cloud accountability problem. These phases are collectively known as the Cloud Accountability which consists of the following seven phases:

Policy Planning: CSPs have to decide what information to log and which events to log on-the-fly. It is not the focus of this paper to claim or provide an exhaustive list of recommended data to be logged. However, in our observation, there are generally four important groups of data that must be logged: (1) Event data – a sequence of activities and relevant information, (2) Actor Data – the person or computer component (e.g. worm) which trigger the event, (3) Timestamp Data – the time and date the event took place, and (4) Location Data – both virtual and physical (network, memory, etc) server addresses at which the event took place.

Sense and Trace: The main aim of this phase is to act as a sensor and to trigger logging whenever an expected phenomenon occurs in the CSP’s cloud (in real time). Accountability tools need to be able to track from the lowest-level system read/write calls all the way to the irregularities of high-level workflows hosted in virtual machines in disparate physical servers and locations.

Logging: File-centric perspective logging is performed on both virtual and physical layers in the cloud. Considerations include the lifespan of the logs within the cloud, the detail of data to be logged and the location of storage of the logs. It may in some cases be necessary to pseudonymise or anonymize private data before it is recorded in logs.

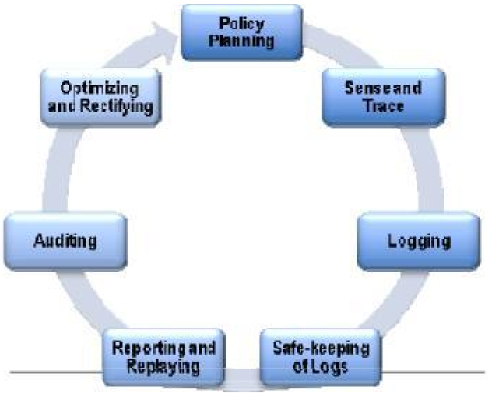

Fig 1: Cloud Accountability [1]

Safe-keeping of Logs: After logging is done, we need to protect the integrity of the logs to prevent unauthorized access and ensure that they are tamperfree. Encryption may be applied to protect the logs. There should also be mechanisms to ensure proper backing up of logs and prevent loss or corruption of logs. Pseudonymisation of sensitive data within the

logs may in some cases be appropriate. **Reporting and Replaying:** Reporting tools generate from logs file-centric summaries and reports of the audit trails, access history of files and the life cycle of files in the cloud. Suspected irregularities are also flagged to the end-user. Reports may cover a large scope, for example recording virtual and physical server histories within the cloud; from OS-level read/write operations of sensitive data, or high-level workflow audit trails.

Auditing: Logs and reports are checked and potential irregularities highlighted. The checking can be performed by auditors or stakeholders. If automated, the process of auditing will become 'enforcement'. Automated enforcement is very feasible for the massive cloud environment, enabling cloud system administrators to detect irregularities more efficiently.

Optimizing and Rectifying: Problem areas and security loopholes in the cloud are removed or rectified and control and governance of the cloud processes are improved.

V BACKGROUP WORK

JAR Generation:

The JAR file includes a set of simple access control rules specifying whether and how the cloud servers and possibly other data stakeholders (users, companies) are authorized to access the content itself. It is a digitally signed statement from a recognized certification authority that indicates who owns a particular public key. Depending on the configuration settings defined at the time of creation, the JAR will provide usage control associated with logging, or will provide only logging functionality.

Provable Data Possession:

Provable data possession (PDP) (or proofs of retrievability (POR)) is such a probabilistic proof technique for a storage provider to prove the integrity and ownership of clients' data without downloading data.

Logger Creation:

We leverage the programmable capability of JARs to conduct automated logging. A logger component is a nested Java JAR file which stores a user's data items and corresponding log Files. The main responsibility of the outer JAR is to handle authentication of entities which want to access the data stored in the JAR file. In our context, the data owners may not know the exact CSPs that are going to handle the data. Hence, authentication is specified according to the servers'. Functionality (which we assume to be known through a lookup service), rather than the server's URL or identity. The data owner can specify the permissions in usercentric terms as

opposed to the usual code-centric security offered by Java, using Java Authentication and Authorization Services.

VI PROPOSED SYSTEM

(i) The Logging Mechanism with Record Generation:

The main responsibility of the outer JAR is to handle authentication of entities which want to access the data stored in the JAR file. In our context, the data owners may not know the exact CSPs that are going to handle the data. Hence, authentication is specified according to the servers' functionality (which we assume to be known through a lookup service), rather than the server's URL or identity. Log records are generated by the logger component. Logging occurs at any access to the data in the JAR, and new log entries are appended sequentially, in order of creation. Each record ri is encrypted individually and appended to the log file.

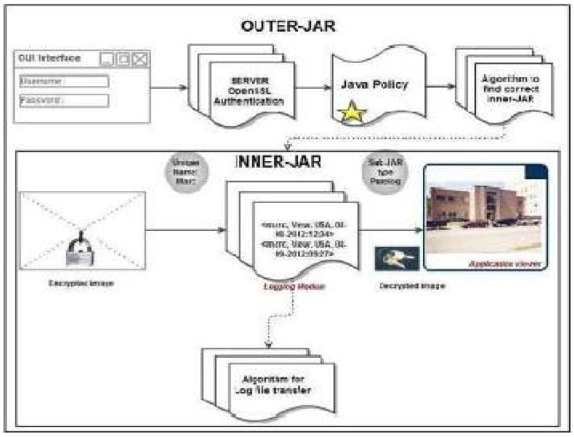

Fig 2: Structure of JAR File[2]

(ii) Data Flow:

The overall CIA framework, combining data, users, logger and harmonizer. At the beginning, each user creates a pair of public and private keys based on Identity-Based Encryption. This Identity-Based Encryption scheme is a Weil- pairing-based Identity-Based Encryption scheme, which protects us against one of the most prevalent attacks to our architecture. Using the generated key, the user will create a logger component which is a JAR file, to store its data items.

The JAR file includes a set of simple access control rules specifying whether and how the cloud servers and possibly other data stakeholders (users, companies) are authorized to access the content itself. Then, he sends the JAR file to the cloud service provider that he subscribes to. To authenticate the CSP to the JAR, we use OpenSSL-based certificates, wherein a trusted certificate authority certifies the CSP.

Once the authentication succeeds, the service provider (or the user) will be allowed to access the data enclosed in the JAR. As for the logging, each time there is an access to the data the JAR will automatically generate a log record, encrypt it using the public key distributed by the data owner, and store it along with the data. The encryption of the log file prevents unauthorized changes to the file by attackers.

VII PERFORMANCE

***Auditability*–** The relative ease of auditing a system or an environment. Poor auditability means that the system has poorly-maintained (or non-existent) records and systems that enable efficient auditing of processes within the cloud. Auditability is also an enabler of (retrospective) accountability: It allows an action to be reviewed against a Pre-determined policy to decide if the action was compliant and if it was not, to hold accountable the person or organization responsible for the action. By using our proposed system, we can achieve the efficient auditing and trust.

***Copying Attack*-** The most intuitive attack is that the attacker copies entire JAR files. The adversary may assume that doing so allows accessing the data in the JAR file without being noticed by the data owner. However, such attack will be detected by our auditing mechanism.

***Accountability*-** "the obligation and/ or willingness to demonstrate and take responsibility for performance in light of agreed-upon expectations", accountability goes beyond responsibility by obligating an organization to be answerable for its actions. Accountability has legal responsibility upon an organization that uses personally identifiable information (PII) to ensure that contracted partners to whom it supplies the PII are compliant to privacy guidelines, wherever in the world they may be. Along with these, cloud provides number of benefits like:

***Self Healing**: Any application or any service running in a cloud computing environment has the property of self healing. In case of failure of the application, there is always a hot backup of the application ready to take over without disruption. There are multiple copies of the same application - each copy updating itself regularly so that at times of failure there is at least one copy of the application which can take over without even the slightest change in its running state.

Linearly Scalable: Cloud computing services are linearly scalable. The system is able to break down the workloads into pieces and service it across the infrastructure. An exact idea of linear scalability can be obtained from the fact that if one server is able to process say 1000 transactions per second, then two servers can process 2000 transactions per second.

Reduced Cost: There are a number of reasons to attribute Cloud technology with lower costs. The billing model is pay as per usage; the infrastructure is not purchased thus lowering maintenance. Initial expense and recurring expenses are much lower than traditional computing.

Increased Storage: With the massive Infrastructure that is offered by Cloud providers today, storage &

maintenance of large volumes of data is a reality. Sudden workload spikes are also managed effectively & efficiently, since the cloud can scale dynamically.

Virtualized: The applications in cloud computing are fully decoupled from the underlying hardware. The cloud computing environment is a fully virtualized environment.

Flexible: Another feature of the cloud computing services is that they are flexible. They can be used to serve a large variety of workload types – varying from small loads of a small consumer application to very heavy loads of a commercial application. This is an extremely important characteristic. With enterprises having to adapt, even more rapidly, to changing business conditions, speed to deliver is critical. Cloud computing stresses on getting applications to market very quickly, by using the most appropriate building blocks necessary for deployment.

VIII CONCLUSION

While enjoying the convenience brought by cloud technology, user's fear of losing control of their own data (particularly, financial and health data) can become a significant barrier to the wide adoption of cloud services. Accountability is used to keep track of the genuine usage of the user's information in the cloud. In this paper, we propose to use JAR signing approach to prevent tampering. Cloud provider configures the policy to grant security privileges to JAR clients. Digitally signed JAR's ensure further security and accountability. Security is controlled by the security policy that's in force at runtime. Cloud provider configures the policy to grant security privileges to JAR clients. An additional element is the certificate that the signer includes in a signed JAR file. It is a digitally signed statement from a recognized certification authority that indicates who owns a particular public key. A certification authority is entities (generally commercial bodies specialized in digital security) that are trusted throughout the industry to sign and issue certificates for keys and their owners. In the case of signed JAR files, the certificate indicates who owns the public key contained in the JAR file thus increasing accountability in cloud architecture.

IX REFERENCES

[1] . Chen and C. Wang, "Accountability as a Service for the Cloud: From Concept to Implementation with BPEL," Proc. 6th IEEE World Congress on Services (SERVICES-1), IEEE, 2010, pp. 91-98.

[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]K.K. Muniswamy-Reddy, P. Macko and M. Seltzer, "Provenance for the Cloud," Proc. Proceedings of the 8th USENIX Conference on File and Storage Technologies, USENIX Association, 2010, pp. 197-210

[4] P. Buneman, S. Khanna and T. Wang-Chiew, "Why and where: A characterization of data provenance," Database Theory—ICDT 2001. pp. 316-330.

[5] S. Pearson , Y. Shen, and M. Mowbray," A privacy Manager for Cloud Computing," Proc. Int'l Conf. Cloud Computing (cloudcom), pp.90 - 106,2009.

[6] A. Squicciarini , S. Sundareswaran and D. Lin, " Preventing Information Leakage from Indexing in the Cloud," *Proc. IEEE Int'l* Conf. Cloud Computing, 2010.

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

[8] Ensuring Distributed Accountability for Data Sharing in the Cloud Author, Smitha Sundareswaran, Anna C.Squicciarini, Member, IEEE, and Dan Lin, IEEE Transactions on Dependable and Secure Computing ,VOL 9, NO,4 July/August 2012.

[9] Provable Data Possession for Integrity Verification in Multi-Cloud Storage Author Van Zhu, Hongxin Hu, Gail - Joon Ahn, Senior Member, IEEE , Mengyang Yu.

[10] S. Pearson and A. Charlesworth, "Accountability as a Way Forward for Privacy Protection in the Cloud, " Proc First Int'l conf. Cloud Computing, 2009.