iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** IV **Month of publication:** April 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

An Open Framework for Foundational Security and Public Verification for Secure Audit Logs

K.N.Jawahar[1], R.Nallakumar[2], R.Shylendar[3]
P.G. Student, Department of CSE, Anna University Regional Centre, Coimbatore, Tamil Nadu, India[1]
Teaching Fellow, Department of CSE, Anna University Regional Centre, Coimbatore, Tamil Nadu, India[2]
P.G. Student, Department of CSE, Anna University Regional Centre, Coimbatore, Tamil Nadu, India[3]

Abstract— Data handling and Security in cloud goes through a dynamic and complex hierarchical service chain. Secure Logging is a service that helps to maintain and store the log records more securely in the server operating in cloud based environment. Log is a record of events occurring within an organization's system or network. Log file usually contains more sensitive and confidential information so it should be maintained more securely. In this project we discuss the security issues involved in log management for a Secure Logging as a service and present a design and implementation of a prototype delegating secure log manager. Here we secure the log records by encrypting it in a JAR file. The JAR file here will include a group of access management rule specifying whether or not and the way cloud servers and most likely different knowledge stakeholder's square measures approved to access the contents itself. When the authentication succeeds, the service providers are allowed to access the information present within the JAR. Reckoning based on configuration settings that is set during the time of creation, the JAR will offer usage management related to the work, or can offer the solely work practicality. As for the work, every time there's correlate access to the information, the JAR can automatically generate a log records. In this paper, to provide the data security in the cloud and to address integrity and confidentiality issues while storing, maintaining, and querying log records JAR file encryption is used.
Keywords: JAR creation, log Record generation, AES, Push and Pull Mode.

I. INTRODUCTION

Organizations use the Cloud during a sort of completely different service models (SaaS, PaaS, IaaS) and preparation models (Private, Public, and Hybrid). There square measure variety of security problems/concerns related to cloud computing however these problems represent two broad categories: Security issues round-faced by cloud suppliers and security problems round-faced by their customer. In most of the cases, the suppliers should make sure that their infrastructure is secure in which their clients' knowledge and applications are protected whereas the client should make sure that their suppliers has taken the right and best security measures to safeguard their data. The intensive use of the virtualization in implementing cloud infrastructure brings distinctive security consideration for customers or tenants in public cloud services.

Log is a record of events occurring within an organization's system or network. Log file usually contains more sensitive and confidential information so it should be maintained more securely. In real world applications sensitive information are kept in log files in an untreated machine. The event that when the attacker captures this system we would guarantee that the attacker will gain a little or no information from the log file and to bound his ability to hack the log file.

JAR file is a compressed file format, where you can store many files. JAR (Java Archive) file format is used to distribute a sets of java classes. JAR file helps to reduce file size and can collect many files in one by compressing it. You can make these jar files executable by collecting many class file of java applications in it. A jar file can execute from java (Java Web Start). JAR file also contain some simple access control rules which will provide authorization for the users to access the content, A JAR file can be classified into two: Inner JAR and Outer JAR. inner JAR file contains the encrypted data, class files to facilitate the retrieval of log files and will display enclosed data in a suitable format, and a log file for every encrypted item. Outer JAR file will authenticate user and guide them to the specific inner JAR. As Log files contains more secure information , it can be secured by encrypting it using AES algorithm and storing it inside a jar file,

Advanced Encryption Standard algorithm is a Symmetric block cipher used to protect classified information by Encrypting and Decrypting it. AES Algorithm uses a key (cipher key) whose length is 128, 192 or 256 bits. The cipher secret is distended to into 10, 12, or fourteen spherical keys, severally, exploitation the "Key Expansion" algorithmic program, where each spherical secret is of 128 bits. As AES uses Larger key sizes which makes it more secure

©IJRASET 2015: All Rights are Reserved

II. LITERATURE SURVEY

In this section, we have a tendency to 1st review connected works addressing the privacy and security problems within the cloud. Then, we briefly discuss works that adopt similar techniques as our approach however serves for various functions.

A. Forward Integriy For Secure Audit Logs

In this paper, they define the forward integrity security properties, inspire its appropriateness as system security requirement, and demonstrate design that can achieve this property. Application includes secure audit logs for intrusion detection or accountability, communication security, and for authenticating partial result of calculation for mobile agents. They discuss about secure audit log applications in detail and also prove security theorems on forward integrity messages authentication schemes.

B. Guide to Computer Security Log Management

Here they provide a practical, real-world guidance for developing, implementing, and maintaining effective log management. This publication covers several topics such as how to establish log management infrastructures, and developing and maintaining robust log management process throughout an organization. The publication also present log management technology from a high-level viewpoint, and the process involved in log management technologies.

C. Secure Logging As a Service—Delegating Log Management to the Cloud

In this paper, they identify the challenges present in a secure cloud based log management services and propose a framework for it. For proper functioning of an organization its Log Records should be securely maintained for a long period. Integrity of the log files and the logging process should be ensured at all the time. In addition, the log files usually contain sensitive information, so confidentiality and the privacy of log records are important. However, for deploying a secure logging infrastructure we need substantial capital expenses so many organizations may find it overwhelming. Delegating log management to the cloud will appear to be a possible cost saving measure.

D. Reliable Delivery and Filtering for Syslog

This paper explains about the features that allow the device to be customized for receiving of messages in Syslog. Using Blocks Extensible Exchange Protocol this feature provides secure and reliable delivery for syslog messages. Also, it allow multiple session for a single logging host independent of underlying transport methods, and provide the filtering mechanism called message discriminator. This module also describes the function of the Reliable Delivery and Filtering of Syslog feature and how to configure it in a network.

III.EXISTING SYSTEM

Data handling is outsourced directly by the cloud service supplier to the alternative entities within the cloud and these entities may delegate the task to others, and so on. Secondly, entities square measure allowed affixing and leaving the cloud in an exceedingly flexible manner. As a result, the knowledge handling within the cloud goes through a posh and dynamic class-conscious service chain that doesn't exist in typical environments. Ordinary internet framework Uses internet services for request and responses.

Traditional cryptographic primitives for the purpose of the data security cannot be directly adopted due to users loss of control of data in Cloud Computing. Therefore, the verification of proper data storage in the cloud should be done without explicit knowledge of whole data. As each user stores various kinds of data in the cloud and demand for the long term continuous assurance for their data safety, and also the user may often modify the data by deleting or updating it, so the problem for verifying correctness of the data stored in the cloud becomes more challenging.

A. Limitations

1) No security for user's knowledge. No authentication or security provided
2) High resource prices required for the implementation.
3) Not appropriate for tiny and medium level storage users.

IV.PROPOSED SYSTEM

In this paper, we have a tendency to propose a comprehensive answer for storing and maintaining log records in an exceedingly server operational in an cloud-based surroundings. We have a tendency to address security and integrity problems scarcely throughout the log generation part, however conjointly throughout alternative stages within the log management method, as well

©IJRASET 2015: All Rights are Reserved

as log assortment, transmission, storage, and retrieval. The key contributions of this paper are as follows . we have a tendency to propose design for the varied parts of the system and develop scientific discipline protocols to handle integrity and confidentiality problems , saving, and query log records at properly , however curious cloud supplier and in transit.
It provides a close security analysis and discusses the dependableness and strength of our design within the face of varied nontrivial attacks by implementing Java Running atmosphere. Timer mechanism for limiting the accessing time for security purpose Secure JVM for creating software system tamper resistance capabilities to JAR file. It provides integrity, confidentiality to JAR.

AES Algorithm provide security to the data by encrypting it. As AES algorithm uses lager key sizes which makes it more secure. Usually the Log file contains more confidential data so it should be stored securely. So to provide more security to the Log file, we encrypt it using the AES algorithm and store it in a jar file.

A. Advantages

1) One of the most innovative options of CIA framework lies in its ability of maintaining the light-weight and powerful responsible that mixes aspects of the access management, usage management and authentication.
2) Providing defences against man in middle attack, lexicon attack, Disassembling Attack, Compromised JVM Attack
3) It's appropriate for restricted and enormous variety of storages.
4) Provides High security.

V. SYSTEM DESIGN

To maintain the Log records more securely we encrypt the files using the AES algorithm and store it inside the JAR file. The overall architecture of the cloud based secure log management system is shown in Fig. 2. There are four major functional components in this system Log Generators, Logging Client, Logging Cloud and Log Monitor.

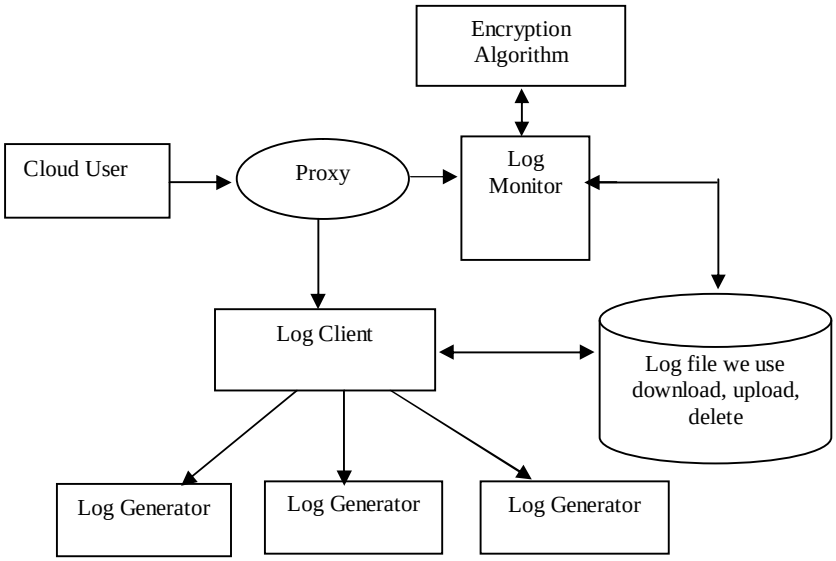


Fig. 2, System Architecture for Secured Log Maintenance

Log Generators are the computing devices that generate log data. The log files generated by these hosts are not stored locally except temporarily till such time as they are pushed to the logging client. Logging client is a collector that receives groups of log records generated by one or more log generators, and prepares the log data so that it can be pushed to the cloud for long term storage. The logging client incorporates security protection on batches of accumulated log data and pushes each batch to the logging cloud. When the logging client pushes log data to the cloud it acts as a logging relay. The logging cloud provides long term storage and maintenance service to log data received from different logging clients belonging to different organizations. The Logging cloud is maintained by a cloud service provider. Only those organizations that have subscribed to the logging cloud's services can upload data to the cloud. The cloud, on request from an organization can also delete log data and perform log rotation. Before the logging cloud will delete or rotate log data it needs a proof from the requester that the latter is authorized to make such a request. The logging client generates such a proof. However, the proof can be given by the logging client to any entity that it wants to authorize. Log Monitor are hosts that are used to monitor and review log data. They can

©IJRASET 2015: All Rights are Reserved

generate queries to retrieve log data from the cloud. Based on the log data retrieved, these monitors will perform further analysis as needed. They can also ask the log cloud to delete log data permanently, or rotate logs.

A. AES Algorithm

Advanced Encryption Standard is a symmetric key block cipher algorithm with a block length of 128 bits. In AES algorithm same key is used for both encryption and decryption. In AES we can use 3 different key lengths: 128 bits, 192 bits, or 256 bits. Encryption uses 10 rounds of processing for 128 bit keys, 12 rounds of processing for 192 bit keys and 14 rounds of processing for 256 bit keys. All rounds are identical except the last round in each case. Each rounds of processing involves one single-byte based substitution step, a row-wise permutation step, a column-wise mixing step, and the addition of the round key. The order in which these four steps are executed is different for encryption and decryption.

B. AES Encryption and Decryption

During Encryption, four steps are involved during each round. Four steps involved are: 1) Substitute bytes, 2) Shift rows, 3) Mix columns, 4) Add round key. The last step consists of XOR'ing the output of the previous three steps with four words from the key schedule.

The four steps involved during the Decryption are: 1) Inverse shift rows, 2) Inverse substitute bytes, 3) Add round key, and 4) Inverse mix columns. The third step consists of XOR'ing the output of the previous two steps with four words from the key schedule. During last round for encryption “Mix columns” step is not involved and during last round for decryption “Inverse mix columns” step is not involved.

VI. EXPERIMENTAL RESULTS

Our approach permits the information owner to not solely audit his content however additionally enforces sturdy back-end protection if required. Moreover, one amongst the most options of our work is that it permits owner to audit even those copies of its data that were created while not his information. Within the future, we have a tendency to decide to refine the approach to verify integrity of the JRE and also the authentication of JARs.

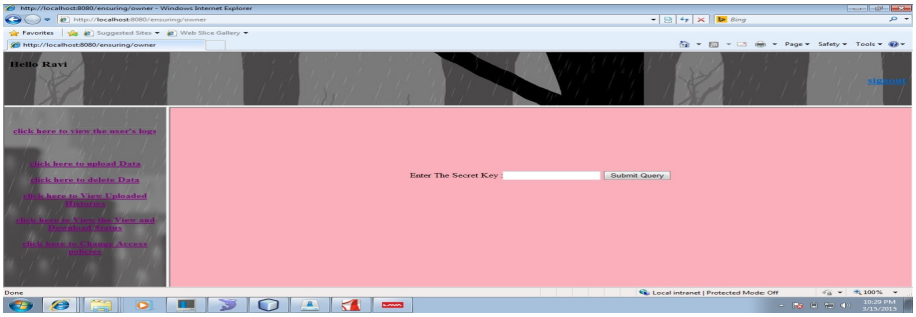


Fig 3, Request for Secured AES Key to Decrypt the Log Data

As Log Data are encrypted using the AES algorithm, so if the user wants to view the Log data they must enter the AES secret key for Decryption. Request for Secured AES Key to Decrypt the Log Data is shown in the Fig 3.

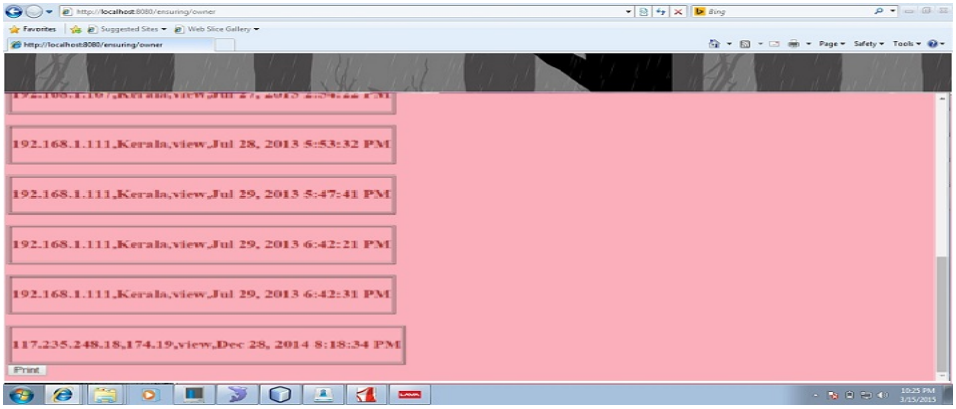


Fig 4, Users Log Records

Users Log Records will be decrypted after entering the AES Secret Key and it is shown in the Fig 4.

©IJRASET 2015: All Rights are Reserved

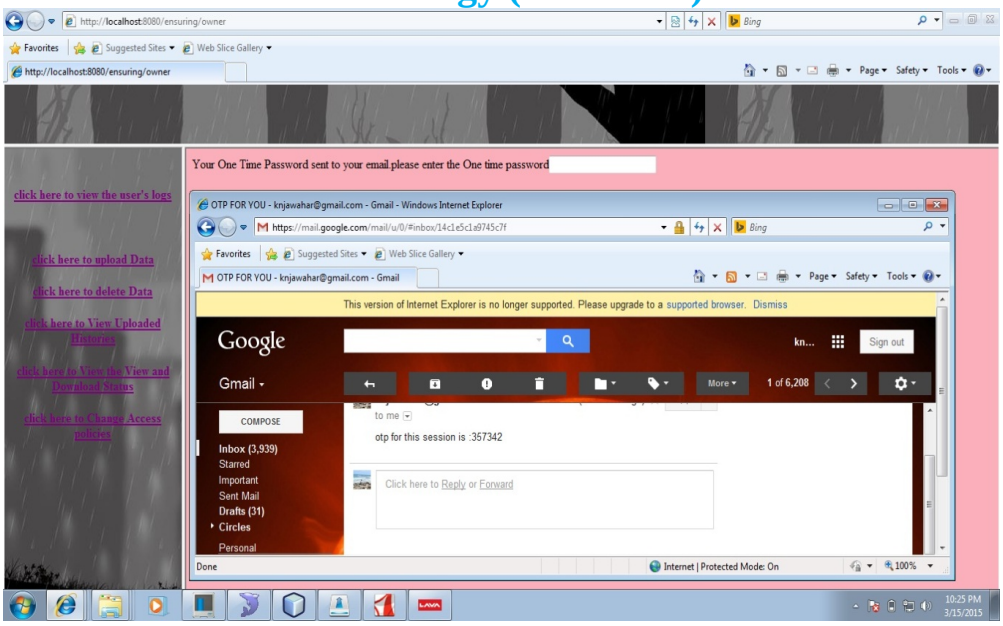


Fig 3, Request for OTP Password while uploading data.

Here we enhance the security by generating the One Time Password(OTP). So When the user tries to upload or download data OTP will be sent to his registered mail id, only after entering the correct OTP the user can upload or delete his data. Fig 3 shows the request for One Time Password while uploading data.

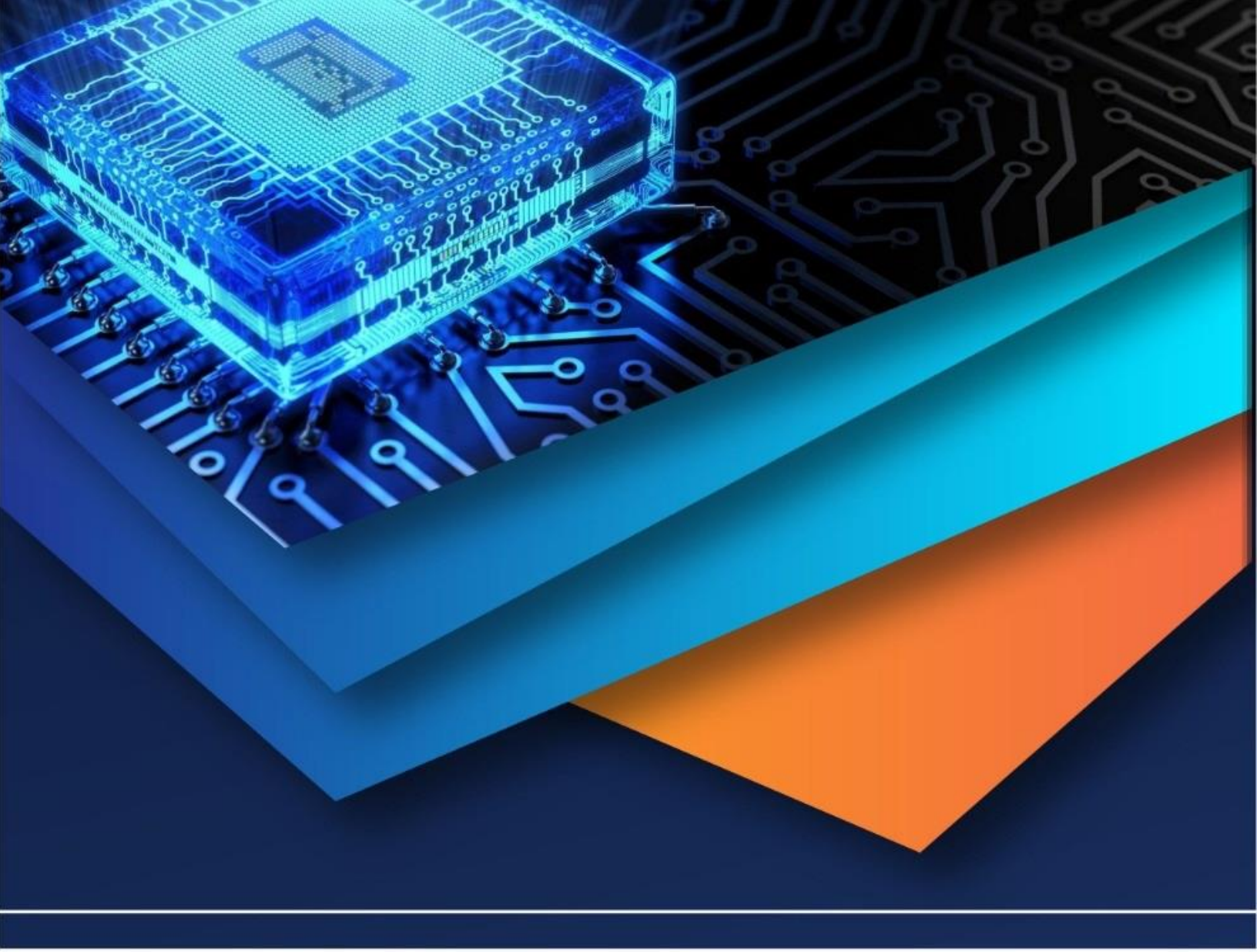
VII. CONCLUSION

Cloud computing permits extraordinarily scalable services to be merely consumed over World Wide Web on associate as-needed basis. Various challenges that arise during log storage and maintenance in the cloud are addressed in this paper by proposing a comprehensive solution. Log files mostly contains more confidential and sensitive information so it is maintained securely by addressing integrity and security issues not only during the log generation phase but also during the other phases in the log management process, including log collection, transmission, storage, and retrieval.. In this paper we propose a secure solution by encrypting the log records and storing it inside the JAR File. We also increase the authentication security by Introducing OTP (One Time Password) during each user Operation.

REFERENCES

[1] Mihir Bellare_ Bennet S. Yeey, "Forward Integrity For Secure Audit Logs, ", November 23, 1997.
[2] Karen Kent Scarfone and Murugiah P. Souppaya (2006). "Guide to Computer Security Log Management," NIST special Publication 800-92 [Online].
[3] D. New and M. Rose, " Reliable Delivery and Filtering for Syslog," Request for Comment RFC 3195, Internet Engineering Task Force, Network Working Group, Nov. 2001.
[4] Indrajit Ray, Kirill Belyaev, Mikhail Strizhov, Dieudonne Mulamba, and Mariappan Rajaram, " Secure Logging As a Service—Delegating Log Management to the Cloud," IEEE SYSTEMS JOURNAL, VOL. 7, NO. 2, JUNE 2013.
[5] Sebastian Schmerl, Michael Vogel, René Rietz, and HartmutKönig, 'Explorative Visualization of Log Data to support Forensic Analysis and Signature Development' J. Computer Networks and Communication Systems Group Brandenburg University of Technology, Cottbus, Germany.
[6] A Sreekanth, Sri K Ishthaq Ahamed, " Assuring Data Sharing and Culpability as a Service in Cloud, ", International Journal of Computer Trends and Technology , ISSN 2231-2803, 2013.
[7] JAYAPRAKASH, " Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing, ", IEEE, Oct 22, 2011.
[8] Sundareswaran, S., Squicciarini, A.C. ; Lin, D. , " Ensuring Distributed Accountability for Data Sharing in the Cloud, ", IEEE, 06 March 2012.
[9] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Int'l Cryptology Conf. Advances in Cryptology, pp. 213-229, 2001.
[10] 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.

©IJRASET 2015: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)