Smart Log Management using Secured Cloud Rendering System

Priti C. Darange[1], Prof.N.R.Wankhade[2]
Department of Computer Engineering
Late G.N. Sapkal College of Engineering
Nashik, Maharashtra, India
[1]darangepriti@gmail.com, [2]nileshrw_2000@yahoo.com

Abstract
Secure maintenance of log records for the long duration is a need of several organizations for their smooth and desirable performance. Logging process & Integrity of various log files requires assurance at every period of time. Most of log files are associated with significant information therefore to maintain their security, privacy and secrecy is very important. But structures of most of the existing logging system are complex and costlier, due to which deployment of such logging system is not affordable for every organization. To address this issue, here in this paper we are proposing a cloud rendered log management system with novel framework that not only provides high level security to data but also provides cost effective solution to the stated problem.

Keywords-Log Management; Integrity; Security; Cryptographic protocols

I. INTRODUCTION

A record of the event is called as log. Logs are created through log entries. Each log entry consists of details regarding a certain event which is appeared inside the systems network. The records that are deals with secrecy of computers are there in logs of organization. There are number of sources that generate security logs like software associated with security of computer or software deals with detection of intrusion [1].

The computer security is originated due to large extent of
Security logs which has been increased drastically over the years. The process of log management includes generation, transmission, storage, analysis and disposal of security log data. The log management ensures the security of log records for expected duration of time. Doing log analysis routinely is essential to identify security related incidents such as violations of policy, deceitful activities and problems associated with operation [2]. Logs plays vital role in performing analysis of audit and forensic, support to investigations as well to identify trends in operations.

To balance limited quantity of resources required for log management is problematic with constant log data supply. Generation of logs and storage are affected by many parameters such as increasing log sources; log data volumes; irregular contents of log, their size and timestamps. Most of log files are associated with significant information therefore to maintain their security, privacy and secrecy is very important. One more problem is assurance of effective log data analysis is done by network administrators [2].

Files of log contain activities of user which is the main task of different attackers. Generally every attacker takes critical care that his/her activities should not be tracked during or after execution of unauthorized activity [3]. An attacker often does is to interrupt the logging services or damage log files. Furthermore log files having sensitive information add secrecy braches. The example of this is when logs has database of transactions. To obtain unauthenticated systems access, information of log is useful for attackers [4].

The logs are categorized into various classes as per specific interest such as security logs consist of information about computer security whereas application logs consist of other security related information [5].

1.1 Usefulness of Logs:

Logs are categorized as per the information content in it. Log records are useful to identify various attacks, scam as well as unsuitable usage. As per the situations, certain logs consist of complete information regarding the detail activity whereas few logs contain less information and are useful in the condition where events correlations are stored in the types of primary logs. E.g. system of intrusion detection stores commands given to server from outside host; which might be first origin of attack. All logs are reviewed by incident handler to supervise for another attempt to connection from the similar IP address; which might be second origin of attack on the information.

1.2 Cloud Computing

The low cost solution is provided by cloud computing for storage and management of log records in sequential order. The organizations can also demand or facilitated with long duration storage of their log records and related data on cloud.

The service provider of cloud not only provides maintenance but also security to log records due to which they become single point of service for several organizations which ultimately benefits to them on large economic scale. A hurdle in providing

storage and maintenance to the log records is pushing these records on the cloud. In case if service provider of cloud is loyal but curious which means they will get secret information easily from log records also link of activities related to log records on their source. As per our survey till date there is no any specific protocol has been designed which resolves all these challenges [2].

II. LITERATURE SURVEY

Most of the proposed methodologies for logging the information in computing systems are based on syslog which is gold standard for network wide logging protocol. User datagram protocol is used by syslog protocol for transmission of log information on log server due to this reliability of accurate data transmission is very less. In addition, syslog protocol does not provide the protection to log records while transmission is in process or when data reached at the end points. The author in [2] presented Syslog-ng which uses Transmission control protocol while transferring log messages. Syslog-ng also supports IPv6 in which log contents are filtered using regular expressions. Syslog-ng prescribes log record encryption using SSL during transmission to provide confidentiality and integrity to the data. But at end-point it does not protect against log data modifications.

Syslog-pseudo [2] proposes a logging architecture to pseudonymize log files. Before archiving the pseudonymizer processes log records and filters the features from specific fields in the log record and insert them with cautiously crafted pseudonyms. Therefore the correctness of logs is not assured. The generated log records are different from the log records that are stored. One more problem with this technique is when the protocol anonymizes each log record individually it does not protect log records from attacks which tries to correlate a number of anonymized records. In addition, the issues such as entering the incorrect login credentials are not addressed or identifying information available in fields that are not anonymized is also not addressed. Similar anonymization of identifying information is performed by anonymous log file which is achieved through substitution of default values or more coarse values. The detail investigation needs original values which can't be restored.

III. SYSTEM DESIGN

3.1 System Overview:

The overview of the system design is shown in figure1.

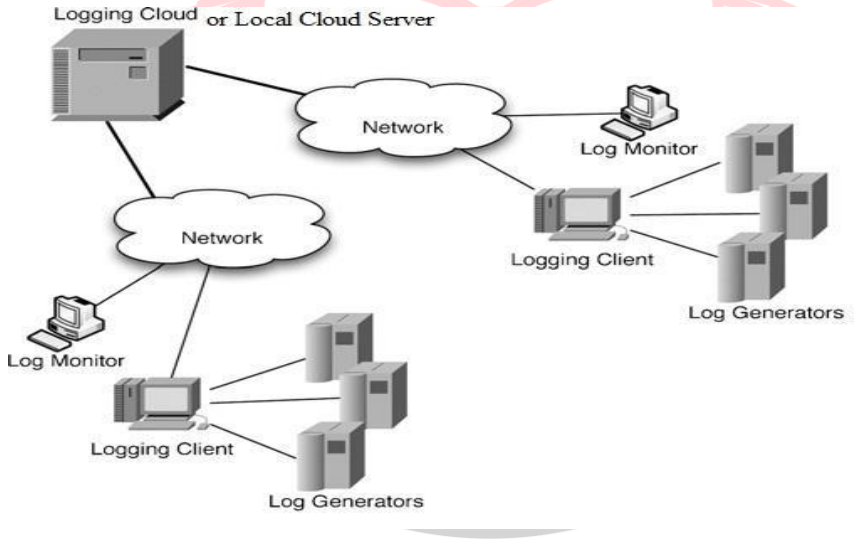

Figure 1. System Architecture

The major functional components in this system are as follows [1].

3.2 Log Generator:

To generate the log data log generator is used which an computing device. Any organization that has cloud-based log management needs many log generators which are having logging capability.

3.3 Logging Cloud:

The log data collected by logging clients from various organizations are maintained and stored by logging cloud. Maintenance of logging cloud is the responsibility of cloud service providers. Access to upload the data on the cloud is provided only to the subscribed organizations.

3.4 Log Monitor:

The task of log monitor is to monitor and review the log data. Generation of queries and retrieval of the log data from the cloud is done by host log monitor. The log monitor performs analysis based on retrieved log data. Log monitor also has right to request the log cloud for the rotation as well permanent deletion of logs.

3.5 Breakdown Structure:

The details of all modules of breakdown structure are described in this section whereas represeted in figure 2. Following areas are focused by the breakdown structure:

- Module 1: Log File Preparation for Secure Storage
  - Log Aggregation and Encryption Module
  - MAC computation and its aggregation
- Module 2: This module handles Secret Sharing
- Module 3. Uploading, Retrieving and Deleting the Log Data are covered in this module. Three sub modules under this module are mentioned below:
  - Upload Tag generation and storage Module
  - Delete Tag generation Module
  - Log Retrieval Module

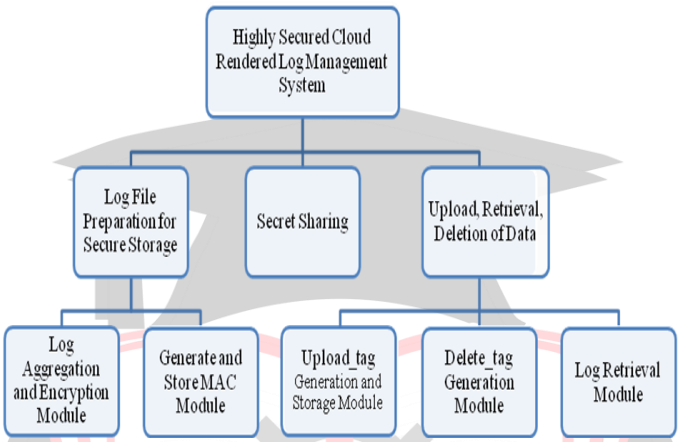

Figure 2.Work Breakdown Structure

IV. MODELLING

4.1 Mathematical Modeling

Whenever solving problems, consider the difficulty level of problem. There are mainly three types of classes provided for that. These are as follows:

1) P Class
2) NP-hard Class
3) NP-Complete Class

This Secured Cloud rendered Log Management System is of P Class because:

1. In polynomial time problem can be solved.
2. Strong results is been produces the system.

4.2 Set Theory

Let S be the set of Inputs, Functions and Outputs S = I, F, O where

I represents input i.e. log file and encryption keys which is input to log files; F represents the set of functions that are performed on the input. O is the Set of output.

Inputs are:

I1= Log File
I2=Encryption Keys

Functions are:

F1= Log File Preparation for Secure Storage
F2= Secret Sharing Module
F3= Upload, Retrieval and Deletion Log Data

Output is:

O1= Retrieve Encrypted File

Sets are:

I = I1, I2
F= F1, F2, F3
O=O1

Mapping is carried out here. Input is mapped to output which is shown in the following Venn diagram:

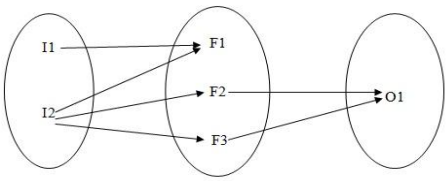

Figure 3: Venn diagram

The comparative analysis of DES, AES and Blowfish algorithm has been depicted in table I.

TABLE I: Comparative analysis of DES, AES and Blowfish Algorithm

V. PROPOSED ALGORITHM

5.1 Implementation Steps

The Implementation includes Encryption and Secure storage. The below section describes the steps in details. As the first module is aggregation of log and encryption module and we are using Blowfish Algorithm. We must store the input keys into Cloud server.

For this here performing some few steps as given below.

1. Generation of logs, the log generators is been used. Log generators task is proves for authentication and perform his duties.

	Symmetric Encryption Algorithms		
	DES	**AES**	**BLOWFISH**
Block Size	64 bit	128 bit	64 bit
Key Size	56 bit	128,192,256 bit	32-448 bit
Created By	IBM in1975	Joan Daemon in 1998	Bruce Schneier in 1998
Algorithm structure	Fiestel Network	Substitution Permutation Network	Fiestel Network
Rounds	16	9,11,13	16
Attacks	Brute Force Attack	Side Channel Attacks	Not Yet

2. A group of log records generated by log generators is receives by logging client.
3. Logging Client will prepares the log data so that it can be pushed to the cloud for long term storage.
4. The transformation of log data from the generators to the client in batches; the logging client incorporates security protection i.e. encryption on batches of accumulated log data and pushes each batch to the logging cloud.

5.2 Blowfish Algorithm:

Blowfish algorithm is a symmetric block cipher that can be effectively used for the purpose of encryption and safeguarding of data. From 32 bits to 448 bits such variable length key is use by it, making it ideal for securing data. Blowfish Algorithm is a Feistel Network, iterating a simple encryption function 16 times. The size of block is 64 bits, and the key can be any length up to 448 bits. Although there is a complex initialization phase required before any encryption can take place, the actual encryption of data is very efficient on large microprocessors.

Blowfish algorithm is a variable-length key block cipher. It is suitable for many applications where the key does not change often, like a communications link or an automatic file encryption when implemented on 32-bit microprocessors with large data caches it is significantly faster than most encryption algorithms.

Basic Steps of Blowfish Algorithm is as follows.

1. Divide input x into two 32-bit halves: xL, xR.
2. Then, for i = 1 to 16:
   xL = xL XOR Pi
   xR = F(xL) XOR xR
   Swap xL and xR
3. After the sixteenth round, swap xL and xR again to undo the last swap.
4. Then, xR = xR XOR P17 and xL = xL XOR P18.
5. Finally, recombine xL and xR to get the ciphertext.

VI. RESULTS AND DISCUSSION

The detail discussion regarding results obtained for the cloud management are depicted in following subsequent figures. In Figure 6.1 logging process has been shown where user will enter into the system by a logging process that is entering the username and password.

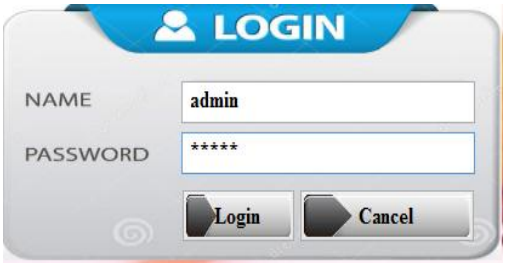

Figure 6.1: Login

After entering the login credentials, the two random input files have been displayed. For authentication purpose we have applied MAC. The designed cloud rendered log management system has been shown in figure 6.2.

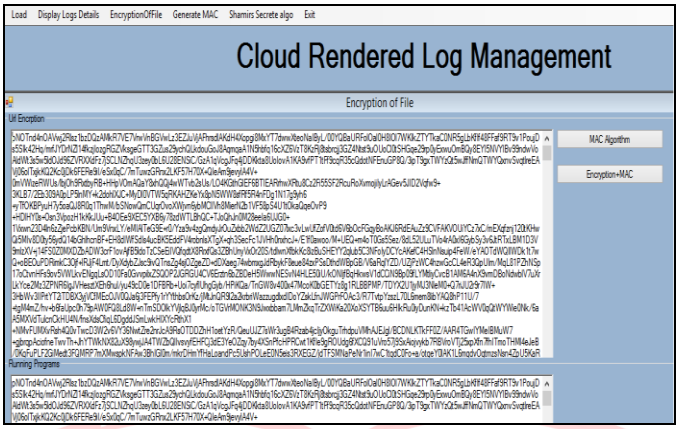

Figure 6.2: Cloud rendered log management

In figure 6.3 generating the senders and receivers MAC address have been carried out. After that MAC has been added.

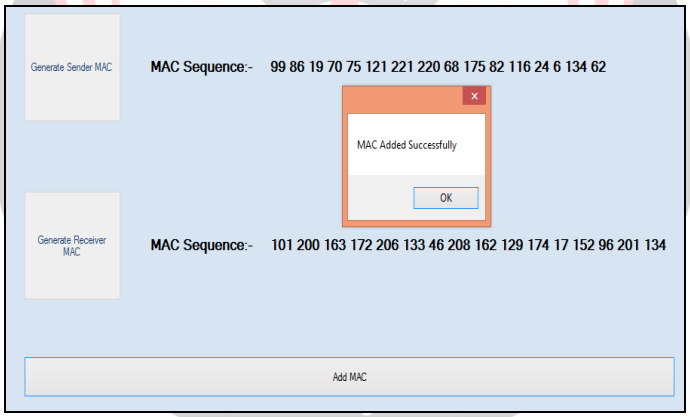

Figure 6.3: MAC

In fig 6.4 here the actual encryption is done with having MAC.

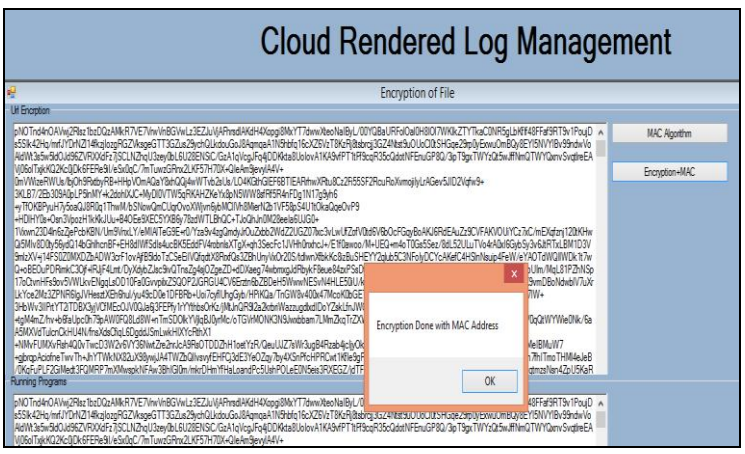

Figure 6.4: Encryption and MAC

In figure 6.5 secret Shamir key algorithm is been applied to stores the data in various location for security purpose.

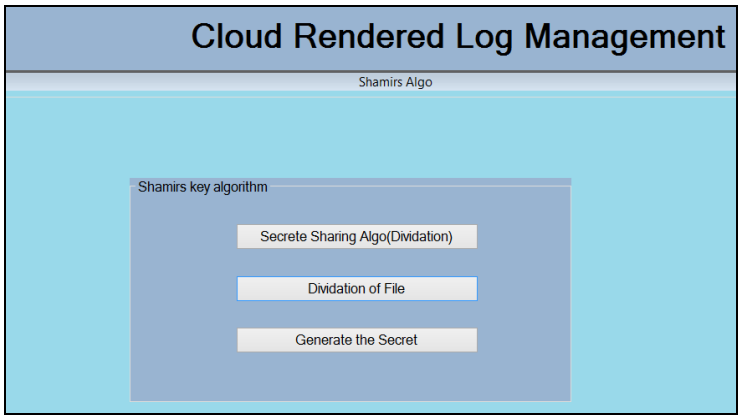

Figure 6.5: Secret Sharing Algorithm

VII. Conclusion

In this paper we are have proposed a cloud rendered log management system with novel framework that not only provides high level security to data but also provides cost effective solution to the stated problem. This work will be helpful for numerous organizations to maintain their log records securely. Secure maintenance of log records for the long duration is a need of several organizations for their smooth and desirable performance. Logging process & Integrity of various log files requires assurance at every period of time which will be provided by this proposed frame work.

References

[1] Indrajit Ray, Kirill Belyaev, Mikhail Strizhov, Dieudonne Mulamba, Mariappan-Rajaram \Secure Logging As a Service Delegating Log Management to the Cloud" in IEEE SYSTEMS JOURNAL, VOL. 7, NO. 2, JUNE 2017.

[2] Karen Kent ,Murugiah Souppaya \Guide to Computer Security Log Management" in NIST Special Publication 800-92J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[3] C. Lonvick, The BSD Syslog Protocol, Request for Comment RFC 3164, Internet Engineering Task Force, Network Group, Aug. 2015.

[4] D. New and M. Rose, Reliable Delivery for Syslog, Request for Comment RFC 3195, Internet Engineering Task Force, Network Working Group, Nov. 2015

[5] M. Bellare and B. S. Yee, \Forward integrity for secure audit logs", Dept. Comput. Sci., Univ. California, San Diego, Tech. rep, Nov. 2013.

[6] J. E. Holt, \Logcrypt: Forward security and public verification for secure audit logs", in Proc. 4th Australasian Inform. Security Workshop, 2006, pp. 203211.

[7] B. Schneier and J. Kelsey, \Security audit logs to support computer forensics", in ACM Trans. Inform. Syst. Security, vol. 2, no. 2, pp. 159176, May 2003.

[8] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, \Proactive secret sharing or: How to cope with perpetual leakage", in Proc. 15th Ann. Int. Cryptology Conf., Aug. 2002, pp. 339352.

