International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Implementation of Highly Secured Log Management System over Cloud

Rajebhosale Sagar S.[1], Pawar Anil B.[2]

[1]PG Student, Department of Computer Engineering, SRES's College of Engineering, Kopargaon, India

[2]Assistant Professor, Department of Computer Engineering, SRES's College of Engineering, Kopargaon, India

Abstract: ***A log is an assortment of record of events that happens at an organization containing systems and network. These logs are very important for any organization, as a result of log file can in a position to record all user activities. As this log files plays important role and also it contains sensitive data, it ought to be maintained highly secure. So, management and firmly maintenance of log records are terribly tedious task. However, for privacy and high security of these log records, it may increase the overhead and additionally it needs further value to deploying such system for an organization. Several techniques have been design up to now for security of log records. Another solution is to maintaining log records over a cloud. Log files over cloud, results in challenges regarding integrity, confidentiality and privacy of log files. During this paper, we propose log management system over cloud in highly secured manner and additionally for dealing the problems to access a cloud based storage, some cryptographic algorithms is used. To overcome the drawbacks of cloud based log management system, this can be the strong work as complete solution.***

Keywords: Cloud Computing, Privacy, Confidentiality, Integrity, Cryptographic algorithms.

1. Introduction

A Log file is the record of detailed information of each and every event of a system, network or application running in associate organization [1]. Some different operational issues occurred in a system, Log file provides useful knowledge to resolve that issues. To distinguish the policy violations, inaccurate activities, protection incidents, log files are extremely helpful. There need to be some extra protection from malicious attacker to log files because Log contains of each and every activity in organization which is very sensitive information about organization. Since log files contain record of most system events furthermore as user activities, so, malicious attackers choose that files as significant target for attack on organization [2]. Associate attacker typically would try to not leave traces of his or her activities behind as they were breaking into a system.

1.1 Generation of log files

To generating logs several protocols are supported syslog[3]. Syslog-ng [4], Reliable delivery of syslog [5], syslog-pseudo [6], forward integrity for audit logs [7], and syslog-sign [8] are some security extensions projected to syslog. They were not able to firmly secure the log files as they provide either partial protection, or dont shield the log records from malicious attacks. On different side the format, size and count of security logs have increased quickly, that needs of some extra features to log management like different tactic for generating, transmitting, storing, analyzing, and eliminating security log data.

1.2 Maintenance of log files

Organizations facing a serious disadvantage with log management are to effectively effort a restricted quantity of log management resources with a continuing supply of log information [1]. For any organization, maintenance of log are typically difficult by several factors, additionally a high range of log sources; inconsistent log formats, content and timestamps among sources; and more and more huge volumes of log information [2]. Log management additionally must to deliver the goods some properties like confidentiality, integrity, and availability of logs. Deploying secure work information to meet all the above challenges cloud storage is best economical different.

1.3 Why Cloud Computing?

In recent years, the emerging cloud-computing paradigm is rapidly gaining momentum as an alternative to traditional information technology. Cloud computing provides a scalability environment for growing amounts of data and processes that work on various applications and services by means of on demand self-services. One fundamental aspect of this paradigm shifting is that data are being centralized and outsourced into clouds. This kind of outsourced storage services in clouds have become a new profit growth point by providing a comparably low-cost, scalable, location-independent platform for managing client's data.

1.4 Organization of paper:

The remainder of paper is organized as follows. Section 2 explains existing protocols as literature survey of this project. In Section 3, present the objective set of this project. Section 4 describes proposed system model of secure logging as protocol. Section 5 describes about algorithms used in this project. Section 6 describes experimental results and concludes paper in section 7.

2. Literature Survey

For storing log information, several approaches are projected. Most of these protocols supported a protocol referred to as syslog. By using UDP (User Datagram

Protocol) Syslog protocol transfers the log information to syslog server. As it uses UDP, the transmission of log messages aren't reliable in syslog. And furthermore it doesn't defend the log information from end-point attacks [3].

For reliable transmission of log records, Syslog - ng uses TCP (Transmission Control Protocol) and SSL (Secure Socket Layer) to supply integrity and confidentiality throughout transit. Syslog ng prescribes log record encryption using SSL during transmission so as to protect the data from confidentiality and integrity breaches while in transit. But it doesn't protect log record modifications at end-points [4].

Syslog - sign which provides integrity to log message and detect missing message using signature block and certificate block, but this protocol has various privacy and confidentiality issues. As like others, Syslog - sign do not afford privacy to log records during transmission and end points [8].

Syslog - pseudo protocol uses the pseudonymizer filters out identifying features from specific fields in the log record and substitutes them with carefully crafted pseudonyms. On the other hand, it doesn't assured correctness of log records. Once log records are substituted by some values they cannot be retrieve back [6].

Reliable - syslog protocol provides trivial mapping backward compatibility. It is built on top of the blocks extensible exchange protocol (BEEP) which runs over TCP to provide the required reliable delivery service. The Reliable-syslog protocol allows device authentication and incorporates mechanisms to protect the integrity of log messages and protect against replay attacks of log data; still it does not defend the log information from privacy and confidentiality breaches [5].

Schneier and Kelsey proposed a logging scheme for cloud that relays on forward integrity and assures it. This scheme is mainly based upon the forward-secure message authentication codes and one-way hash chaining similar to that suggested by the Bellare-Yee protocol [7] i.e., if the trusted server is being attacked or being compromised, it breaks the security of the logging scheme [8]

Holt improves the Schneier-Kelsey protocol by merging the public verifiability log records. However, this scheme being a public-key based scheme, the overhead is found to be significantly more. None of these three schemes consider the privacy concerns of storing and retrieving log records [9].

In addition to all, these three schemes suffer equally from truncation attacks where an attacker deletes the contiguous subset of log records from the very end it is being kept.

2.1 Pros and Cons

Due to usefulness of log files, Management and security to logs has become a necessity. Log management is a continuously evolving discipline and ever changing tactics to commit security. Table 1. shows that several approaches of log management is used which has some limitations in real time. In case of the existing system the log management is done but it doesnt assure about security of log files. To overcome these drawbacks the System uses cryptographic techniques to prohibit the attack on log files. This method provides a way to prevent provokers from reaching to log files, rather than providing way to securely storing and maintaining log files.

Security algorithms selected for the log management was based on the effectiveness and strengths of the algorithm. The system concentrates on improving the log management systems security and effectiveness. The rising paradigm of cloud computing guarantees a low cost opportunity for organizations to store and manage log records in a proper manner. Organizations can outsource the long-term storage requirements of log files to the cloud. The challenges of storing and maintaining the log records become a concern of the cloud provider [1]. Since the cloud provider is providing a single service to many organizations that it will benefit from economies of scale.

Table 1: Secured Logging Techniques Review

SR No.	*Researcher's Name*	*Proposed Work*	*Drawbacks*
1	M. Bellare & B.S.Yee,Nov 1997	Forward integrity of log records	Requires online trusted servers to maintain secret keys, and it can be attacked.
2	C. Lonvick, Aug 2001	Syslog	It uses UDP protocol so no reliable delivery of log message.
3	D.New & M.Rose, Nov 2001	Reliable Syslog	It does not prevent against confidentiality breaches.
4	U.Flegel, Oct 2002	Syslog-pseudo	The protocol does not ensure correctness of logs.
5	J.E.Holt,2006	Logcrypt	Suffers from Truncation Attack
6	D.Ma & G. Tusdik, March 2009	Forward Secure sequential aggregate authentication	Efficient but very expensive method
7	J.Kelsey & J.Callas, May 2010	Syslog-sign	No privacy or confidentiality during transmission of data.
8	Balabit IT Security, Sept 2011	Syslog-ng	No protection against log log data modification.
9	Indrajit Ray & K. Belyaev, June 2013	Secure Logging As A Service-Delegating Log Management to the Cloud	Most efficient and secured technique but loosely coupled architecture.

3. Objective Set

General Objective of this system is

1. To provide comprehensive solution to maintain security (confidentiality, integrity and privacy) of organizations log records.
2. To develop cryptographic protocols to address integrity and confidentiality issues with storing, maintaining, and querying log records.

3. To maintain the anonymity of the organization whose log records are being stored on the cloud.

4. System Model

The overall proposed system architecture, breakdown structure and mathematical modeling details are covered in this section.

4.1 System Architecture

The following Fig.1 shows the system architecture for the proposed dissertation work based on introduced dissertation idea in introduction Section. The breakdown structure mainly focuses on following modules and their details are explained in subsequent section.

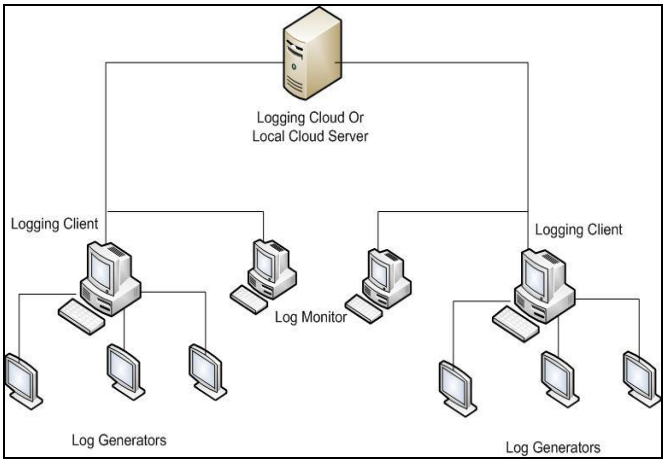

Figure 1: System Architecture

The complete system architecture is shown in figure 1. Major components of the projected system are as follows.

- Log Generator: Log generators are computing devices which generates log file. Log generators are the main target of attacker. So, Logs of these hosts not stored locally. For security purpose, it is pushed to logging client for storing the logs over cloud.
- Logging Client: The logging client is receives groups of log records generated by log generators. Logging Client prepares the log data using cryptographic algorithms so that it can be securely store to the cloud. The log data is transferred from the generators to the client in batches, either on a schedule, or as and when required depending on the amount of log data waiting to be transferred.
- Logging Cloud or Local Cloud Server: For maintaining log records over long term, Logging Cloud or Local Cloud Server is used. Logging cloud is maintained by cloud provider. Instead of it, we use Local cloud server in our project. It can delete log records only after the authenticate request from logging client.
- Log Monitor: Log Monitors are simply hosts that are used to monitor and review log data. They can analyze log records based on log records retrieved by authenticate request of log retrieval. Log Monitor and Logging Client may be the same host. The log monitor can be maintained by the same organization or can be a separate entity.

4.2 Break Down Structure

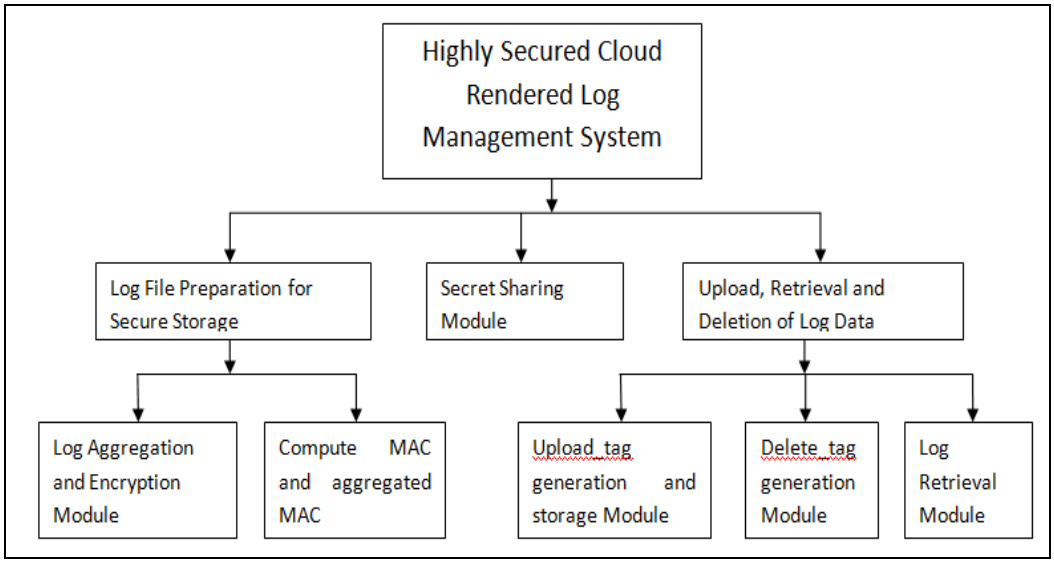

Figure 2: Work Breakdown Structure

The Break down Structure includes 3 different modules. Main Modules further divided into sub modules.

1) Log file preparation for secure storage
   a. Log Aggregation and Encryption Module.
   b. Compute MAC Module
2) Secret Sharing Module
3) Upload, Retrieval, Deletion of log Module.
   a. Upload_tag Generation and Storage
   b. Delete_tag Generation Module
   c. Log Retrieval Module

The following section describes the steps in details.

1) Log file preparation for secure storage: As the first module contains aggregation of log and encryption module and we are using Blowfish Algorithm. We must store the input

keys into local cloud server. For preparing and storing the log files we are performing some steps as given below.

i. Log generator proves authentication and performing his duties.
ii. The logging client is receives groups of log records generated by log generators.
iii. Logging Client prepares the log data so that it can be pushed to the cloud for long term storage.
iv. The log data is transferred from the generators to the client in batches; the logging client incorporates security protection i.e. encryption on batches of accumulated log data using Blowfish algorithm and then create MAC of the data using SHA-2 algorithm. After creating it, each batch of MAC pushes to the logging cloud.

2) Secret sharing module: We use secret key cryptosystem to provide confidentiality and integrity. To distribute the keys across several hosts, we use proactive secret sharing scheme. The idea behind this scheme is that at the end of a fixed period of time, the shares stored at each host change although the original secret stays the same [11]. The idea is that given a secret S, and n and q two non-negative integers such that 0 ¡ q¡ n, we would like n entities to share the secret S such that

i. No single entity holds the complete secret;
ii. Any subgroup of entities of size q can collectively recreate or recover the secret S;
iii. No subgroup of entities of size t ¡ q can re-create or recover the secret.

Secret sharing schemes were used to protect the secrets by distributing them over different locations. In particular k out of n threshold schemes, security is assured if the entire life time is secret and therefore its adversary and is restricted to compromise less than k of the n location for long lived and sensitive secret protection is insufficient .Here they propose an efficient proactive secret sharing scheme, where the shares are periodically renewed in such a way that information gained by the adversary in one period of time is useless for attacking the secret after the shares have been renewed [12].

3) Upload, Retrieval, Deletion of log Module: The logging client uploads data in batches where each batch is delimited by a start-of-log record and an end-of log record. The cloud provider will accept log records only from its authorized clients. Thus, during upload a logging client has to authenticate to the logging cloud to prove that the client had obtained prior authorization from the logging cloud to use the latter services. After proving authentication, system should generate upload tag for uploading the data over cloud and upload tag can be work as unique identifier of that particular log file. No one can recognize or reuse of upload tag.

To view or retrieval of log file, the entity that needs to download log data (most of the time the log monitor), sends a retrieve request together with the upload-tag corresponding to the desired log data. The logging cloud gets the data from its storage and sends it to the requester. The cloud provider does not have to authenticate the requester. This is because, by virtue of the log batches being encrypted, the retrieved data is useful only to those who have the valid decryption keys. To delete log data, the delete requester sends an appropriate delete message to the logging cloud. In response the logging cloud throws a challenge to the requester. The requester proves authorization to delete by presenting a correct delete tag.

5. Algorithm used

Blowfish Algorithm

Blowfish is a symmetric block cipher that can be effectively used for encryption and safeguarding of data. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for securing data. Blowfish algorithm is a Feistel Network, iterating a simple encryption function 16 times. The block size 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.

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.

Implementations of Blowfish that require the fastest speeds should unroll the loop and ensure that all subkeys are stored in cache.

Secure Hash Algorithm-2

SHA-2 is a set of cryptographic hash functions. SHA stands for Secure Hash Algorithm. This hashing algorithm has four variants SHA-224, SHA-256, SHA-384, and SHA- 512 which are named according to the number of bits in their outputs. By comparing the computed “hash” to a known and expected hash value, a person can determine the data’s integrity [14]. For example, computing the hash of a downloaded file and comparing the result to a previously published hash result can show whether the download has been modified or tampered with. A key aspect of cryptographic hash functions is their one-way nature: given only a computed hash value, it is generally impossible to derive the original data [15]. To store log files securely, here we use SHA-2. After encryption, MAC will be generated. Hence Log files become more secure.

Proactive Secret Sharing Algorithm

Secret sharing schemes protect secrets by distributing them over different locations (shareholders). In particular, in k out of R threshold schemes, security is assured if throughout the entire life-time of the secret the adversary is restricted to compromise less than k of the n locations. For long-lived and sensitive secrets this protection may be insufficient. We propose an efficient proactive secret sharing scheme, where shares are periodically renewed (without changing the secret)

in such a way that information gained by the adversary in one time period is useless for attacking the secret after the shares are renewed. Hence, the adversary willing to learn the secret needs to break to all k locations during the same time period (e.g., one day, a week, etc.). Furthermore, in order to guarantee the availability and integrity of the secret, we provide mechanisms to detect maliciously (or accidentally) corrupted shares, as well as mechanisms to secretly recover the correct shares when modification is detected [12].

A proactive secret sharing system must be able to check whether a share of each participating server has been corrupted (or lost), and restore the correct share if necessary. Otherwise, **an** adversary could cause the loss of the secret by gradually destroying **n** - ***k*** shares. Algorithm presents the necessary mechanisms for detection and recovery of corrupted shares.

6. Result Analysis

This section highlights initial module development of the proposed system where developed GUI is explained in next subsection and further actual result analysis for this module are outlined in later part.

The development environment is selected for the proposed system development is Operating System: Windows XP and Above with Front-End: C#.NET and Back- End: MY SQL SERVER 2005 on single computer system with minimum 1GB RAM and enough storage space.

6.1 Input File

Figure 3 shows the input file in notepad. After loading log file, this input file will be open. For opening the log file system should ask for proper application to open specific file and after selecting it i.e. Notepad, System should open Log file which contains all the details about events and activities performed by user on system.

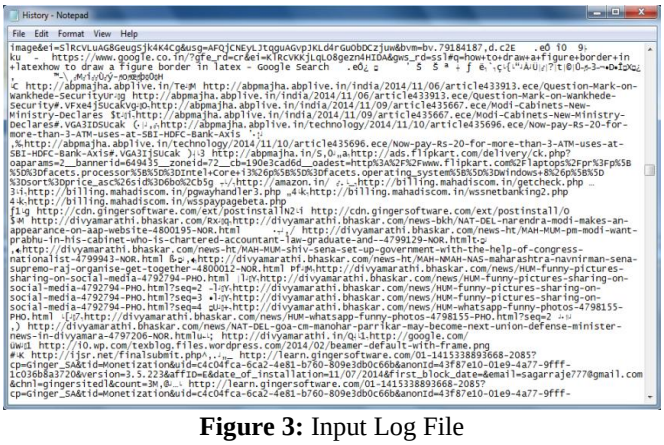

Figure 3: Input Log File

6.2 Provided User Interface

Fig. 4 shows the initial window developed to input the data contents for the proposed system. As stated earlier, with the help of provided provision, user can upload standard log file contents as well as current running applications data contents by browsing the desired storage location on the terminal.

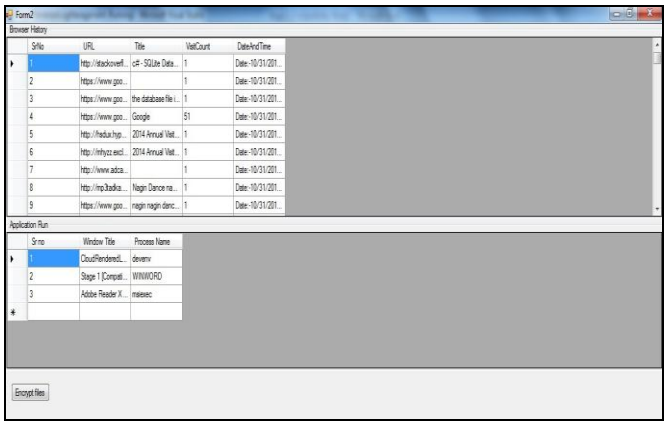

Figure 4: Contents of log file

Once, the original contents are loaded, it cannot be used as it is in further computation and hence, required preprocessing, so the system provides processing facility as shown in Figure 5, after that, by clicking on View tab, it is possible to get secure log file.

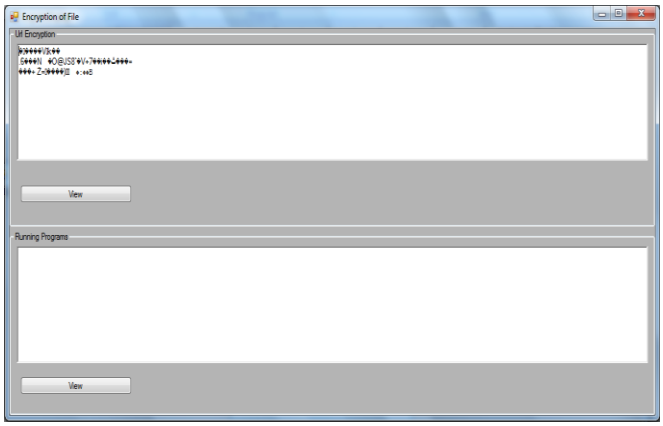

Figure 5: Processing on log file

6.3 Result of Implemented Module

Following Figure 6 shows the contents of processed log file after first module execution. Shows the output file which is in encrypted form. So, It is in non-readable form. Blowfish algorithm uses 16 rounds. So, Encrypted file will be more secure and then save it locally.

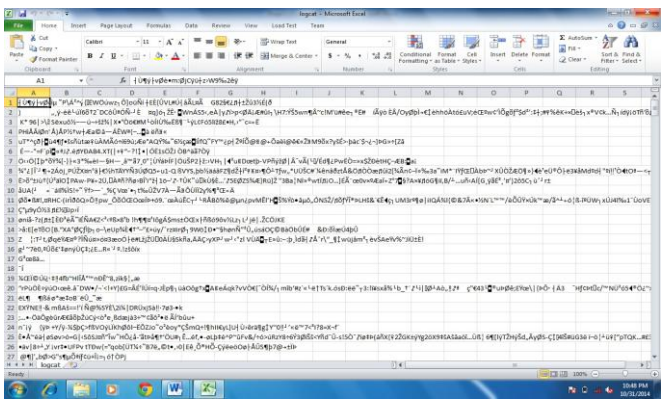

Figure 6: Result of implemented module

7. Conclusion

In this project, an entire system is planned to firmly delegate log information to a cloud. It observes existing protocols and system with their problems in various situations and provides complete log management service which is extremely secured using cloud. It does summarize that encryption techniques used affects the overall performance of system. Then projected a comprehensive scheme that addresses all security issues like integrity, privacy, confidentiality and not simply throughout log generation phase, however collectively throughout all different stages in log management process. We addressed the construction of an efficient audit service for data integrity in clouds. From initial results, it is proven that log records are much secured due to encryption technique used in this project. The initial steps of the system have performed with successful results and with test cases. Future work of this project will be implementation of advanced cryptographic algorithms which able to give high security to log records. Simultaneously, work on automation of log aggregation and upload to cloud server. It'll affect to less overhead of projected system.

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 2013
[2] D. Ma and G. Tsudik, "A new approach to secure logging" ACM TRANS. STORAGE, VOL. 5, NO. 1, MAR. 2009.
[3] Karen Kent ,Murugiah Souppaya "Guide to Computer Security Log Management" in NIST Special Publication 92
[4] C. Lonvick, The "BSD Syslog Protocol", Internet Engineering Task Force, Network Working Group, Aug. 2001.
[5] D. New and M. Rose, "Reliable Delivery for Syslog", Internet Engineering Task Force, Network Working Group, Nov. 2001
[6] U. Flegel, "Pseudonymizing unix log file", in Proc. Int. Conf. Infrastruture Security, LNCS 2437. Oct. 2002, pp. 162-179
[7] M. Bellare and B. S. Yee, "Forward integrity for secure audit logs", Dept. Computer Sci., Univ. California, San Diego, Tech. Rep., Nov. 1997.
[8] J. Kelsey, J. Callas, and A. Clemm, "Signed Syslog Messages", Internet Engineering Task Force, Network Working Group, May 2010.
[9] B. Schneier and J. Kelsey, "Security audit logs to support computer forensics",in ACM Trans. Inform. Syst. Security, vol. 2, no. 2, pp. 159- 176, May 1999.
[10] Aderemi A. Atayero, Oluwaseyi Feyisetan "Security Issues in Cloud Computing: The Potentials of Homomorphic Encryption" vol. 2, no. 10, October 2011
[11] A. Shamir, "How to share a secret", Commun. ACM, vol. 22, no. 11, pp. 612-613, Nov. 1979.
[12] 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. 1995, pp. 339-352
[13] Vinod Vaikuntanathan "Homomorphic and General encryption" Madars Virza 6.892 Computing on Encrypted Data September 09, 2013.
[14] NIST, "Descriptions of SHA-256, 384 and 512"[Online] available:http://csrc.nist.gov/groups/STM/cavp/documents/shs/sha256- 384-512.pdf
[15] Federal Information Processing Standards Publication, "SECURE HASH STANDARD", OCT 2008, [Online] Available: http://csrc.nist.gov/publications/fips/fips180-3/fips180-3 final.pdf
[16] Pawar Anil B., Rajebhosale Sagar S., "Development of Highly Secured Cloud Rendered Log Management System, in IJCA, VOL. 108, No.16, DEC 2014. [Online] Available:http://research.ijcaonline.org/volume108/number16/pxc3900448.pdf

Author Profile

Mr. Rajebhosale Sagar S. received the B.E. degree in Information Technology from University of Pune, in 2012. Currently he is pursuing Master's degree in Computer Engineering from Sanjiwani Rural Education Society's College of Engineering, Kopargaon under University of Pune. His areas of interest are networking, network Security and cloud computing. He is currently working in the field of computer security.

Prof. Pawar Anil B. received the B.E. degree in Information Technology from University of Pune. He completed his M.E (CSE) from Government college of Engineering, Aurangabad. He is currently pursuing the Ph.D. degree in computer science. He is presently working as Assistant Professor in Dept. of Computer Engineering in Sanjivani Rural Education Society's College of Engineering, Kopargaon, India. His current research interests include computer and network security, database security, security and trust models, privacy, and computer forensics.