Protection of Log Records by Homomorphic on Cloud Environment

R.Madhusoothanan[1] S.Chockalingam[2] B.Ayyappan[3]

[1,2]U.G. Student (Final Year) [3]Assistant Professor

[1,2,3]Department of Information Technology

[1,2,3]Jeppiaar Engineering College Chennai-119, Tamilnadu, India

Abstract— *Integrity of the log files and that of the logging process need to be ensured at all time. Besides, a log files are consists of much helpful information regarding activities of the system and network. These logs are made-up of event which has been done by users on system or in network. To address privacy concerns current implementation allows access to log records that are indirectly identified by upload tag values. We are going to propose a homomorphic encryption technique that will allow encryption of log records in such a way that logging cloud can execute some queries on the encrypted logs without breaching confidential or privacy and also it greatly reduces the communication overhead between a log monitor and logging cloud.*

Keywords— Cloud, Homomorphic, LogFiles, Security

I. INTRODUCTION

A log is composed of logging records each contain information related to a particular event that has occurred within the system. The logs which contain records related to computer security. Log management is essential to ensuring that computer security records are stored in sufficient detail for an appropriate period of time. Routing log analysis is beneficial for identifying security incident, policy violation fraudulent activity and operational problem.

In this way first target of attacker is to have access to the log files therefore after having access to log file the first thing that attacker want to do is to damage the files and threaten to the confidentiality and second thing is that to discontinue the logging service to mix up the loggers. Furthermore there are chances of outsourcing the sensitive information to others in this way violating the security. Example is that when user faulty put his password in the username filed at that time when he logged into system then logging program take password as username in this way breaches the privacy. From this scenario it is necessary that logging should be done in secure manner and log files are sufficiently protected for long amount of time.

In the above observations, it is important that logging be provided securely and that the log records are adequately protected for a predetermined amount of time (maybe even indefinitely). Traditional logging system that are based on syslog [4] have not been designed with such security measures. Security enhancement newly designed, such as reliable delivery of syslog [5], forward integrity for audit logs [6], syslog-ng [7], and syslog-sign [10], often provide either weekly protection. Besides, log management requires storage and processing ability. The log service must be able to store data in an organized manner and provide a fast and useful access facility. Log records may often need to be made available to outside auditors who are not related to the particular organization. Deploying a secure logging scheme to meet all these challenges entails significant infrastructural support and capital expenses that many organizations may find overwhelming.

Our proposal is to encrypt data before sending it to the cloud provider, but to execute the calculations the data should be decrypted every time we need to process it. Until now it was not to encrypt data and to trust a third party to keep them safe. So to allow the Cloud provider to perform the operations on encrypted data without decrypting them requires using the cryptosystems based on Homomorphic Encryption.

II. LITERATURE SURVEY

In this survey we have collected some of the secure logging methods for our reference.

Sr. No	Scientist's Name	Proposed Models	Disadvantages
1	C Lonvick, Aug2001	Syslog Protocol	Uses UDP protocol so unreliable delivery and it can't protect log records in transit.
2	Balabit, 2011	Syslog-ng	It can't protect against log data modification when it stored in system.
3	J. Kelsey & J. Callas, May 2010	Syslog-sign	It doesn't provide data confidentiality and privacy during transit of data.
4	U. Flegel, Oct 2002	Syslog-pseudo	This protocol doesn't ensure exactness of logs.
5	D. New & M. Rose, Nov 2001	Reliable-syslog	Not prevent against confidentiality and privacy of data.
6	M. Bellare and Yee, Nov 1997	Forward Integrity	This protocol requires online trusted servers.
7	D. Ma and Tsudik, March 2009	Forward secure sequential authentication	Competent method but requires more capital.
8	Indrajit Ray & K.Belyaev, June 2013	Secure Logging As A Service-Delegating Log Management to the Cloud	Most competent And secured method but loosely coupled architecture.

Table 1: Various methods for secure logging system

A. Existing Protocol

1) BSD System Log Protocol

The BSD Syslog Protocol [1][2] defines a number of service associated options and also relate to inseminating event information. This information also describes the two mappings of the syslog protocol to TCP connections, both which are useful for transmitting trustworthy delivery of event information. The administers a trivial mapping maximizing backward compatibility and also helps in supplying a more entire mapping. Both of the technique provides a degree of sturdiness and security in message delivery that is engaged to the usual UDP-based syslog process, by establishing encryption and authentication over a connection-oriented protocol.

2) New Approach to Secure Log

The necessity for secure logging is well understood by the security teams, both researchers and practitioners. The ability to validate all log entries is more essential to any purpose handling methods. We start by identifying the designs in secure logging and recognize some troubles inborn to systems based on trusted third-party servers. They suggest a dissimilar approach to secure logging based upon newly developed Forward-Secure Sequential Aggregate authentication techniques.

3) Public Key Security Protocol

The use of public key encryption was to offer a secure network communication which has established as attention. The public key encryption is frequently useful in providing against the passive eavesdroppers, who frequently try to strike the data's and try to decode the message. It has been pointed out, that an inappropriate designed procedure could be susceptible to an active behavior, one who may imitate other user or modify the message being transmitted. Numerous designs have been prepared in which the security of protocols are discussed accurately.

III. PROPOSED SYSTEM

The increasing popularity of cloud-based data and mobile devices has led to the appearance of a number of latest information services to meet customer's satisfaction. At the right time, there is an increasing attentiveness of the problem of personal information becoming public and of the require to be proficient to use personal data while keeping it more secure. This paper is introducing the concept of homomorphic encryption technique. Homomorphic encryption means it is a type of encryption which allows particular types of computations to be takes place on cipher text and produce an encrypted datas or information which, when decrypted, compare the result of operations carried out on the plaintext in this way security achieved is more.

A. Homomorphic Encryption

Homomorphic encryption technique is a type of encryption methods that allows computations to be carried out on ciphertext, thus generating an encrypted result which, when decrypted, matches the result of operations performed on the plaintext.

This is sometimes a desirable feature in modern communication system model. Homomorphic encryption would allow the linking together of different services without exposing the data to each of these services, i.e example a link of different services from different companies could 1) calculate the tax 2) the currency exchange rate 3) during shipping, without exposing the unencrypted data to each of those services. Homomorphic encryption schemes are, ductile by design. That enables their use in cloud computing infrastructure for ensuring the confidentiality of processed data. Besides the homomorphic encryption property of various cryptosystems can be used to create many other secure systems, to example secure voting systems model, collision-resistant hash functions scheme, access schemes, and many more.

1) Partially Homomorphic Cryptosystem

In the examples, the symbols $\sum x$ is used to denote the encryption of the message x.

2) Unpadded RSA

If the RSA public key is modulus m and exponent e, then the encryption of a message x is given by $\sum(x) = x^e \bmod m$. The homomorphic property is then

$$\sum(x_1).\sum(x_2) = x_1^e x_2^e \bmod m = (x_1 x_2)^e \bmod m = \sum(x_1 x_2)$$

3) ElGamal

In the ElGamal cryptosystem, in a cyclic group G of order q with generator g, if the public key is (G, q, g, h), where $h = g^x$, and x is the secret key, then the encryption of a message m is $\sum(m) = (g^r, m.h^r)$, for some random $r \in \{0, \ldots, q-1\}$. The homomorphic property is then

$$\sum(x_1).\sum(x_2) = (g^{r1}, x_1.h^{r1})(g^{r2}, x_2.h^{r2}) = (g^{r1+r2}, (x_1.x_2)h^{r1+r2})$$

4) Goldwasser–Micali[

In the Goldwasser–Micali cryptographic system, as the public key is the mod m and quadratic non-residue x, then the encryption of a bit b is $\sum(b) = (x^b r^2 \bmod m)$, for some random $r \in \{0, \ldots, m-1\}$. The homomorphic property is then

$$\sum(b_1).\sum(b_2) = x^{b1} r_1^2 x^{b2} r_2^2 = x^{b1+b2}(r_1 r_2)^2 = \sum(b_1 \oplus b_2)$$

5) Benaloh

In the Benaloh cryptosystem, if the public key is the modulus m and the base g with a block size of c, then the encryption of a message x is $\sum(x) = g^x r^c \bmod m$, for some random $r \in \{0, \ldots, m-1\}$. The homomorphic property is then

$$\sum(x_1).\sum(x_2) = (g^{x1} r_1^c)(g^{x2} r_2^c) = g^{x1+x2}(r_1 r_2)^c = \sum(x_1 + x_2 \bmod c)$$

6) Paillier

In the Paillier cryptosystem, if the public key is the modulus m and the base g, then the encryption of a message x is $\sum(x) = g^x r^m \bmod m^2$, for some random $r \in \{0, \ldots, m-1\}$. The homomorphic property is then

$$\sum(x_1).\sum(x_2) = (g^{x1} r_1^m)(g^{x2} r_2^m) = g^{x1+x2}(r_1 r_2)^m = \sum(x_1 + x_2 \bmod m^2)$$

IV. CONCLUSION

In this paper the system to securely contract out log records to a cloud environment. The homomorphic encryption method to encrypt the log records for maintenance purposes. The logging cloud can execute some queries on the encrypted logs without breeching confidentiality. However, securely maintaining log record over extended period of time cost efficient. We designed a protocol which simultaneously provides the need for security and privacy features. It reduces the communication overhead to a significant level and promises a low cost opportunity for

maintaining log records. With the help of these techniques we assure that access log files only authorized users. The implementation of the logging client is loosely coupled with the operating system based logging.

REFERENCES

[1] M. Bell are and B. S. Yee, ―Forward integrity for secure audit logs,‖ Dept. Computer. Sci., Univ. California, San Diego, Tech. Rep., Nov. 1997.
[2] Bala Bit IT Security (2011, Sep.). Syslogng—Multiplatform Syslog Server and Logging Daemon [Online]. Available: http://www.balabit. Com /network-security / syslog-ng
[3] M. Bellare and B. S. Yee, "Forward integrity for secure audit logs," Dept. Comput. Sci., Univ. California, San Diego, Tech. Rep., Nov. 1997.
[4] BalaBit IT Security (2011, Sep.). Syslog-ng—Multiplatform Syslog Server and Logging Daemon [Online]. Available: http://www.balabit.com/network-security/syslog-g
[5] C. Eckert and A. Pircher, "Internet anonymity: Problems and solutions," in Proc. 16th IFIP TC-11 Int. Conf. Inform. Security, 2001, pp. 35–50.
[6] U. Flegel, "Pseudonymizing unix log file," in Proc. Int. Conf. Infrastruture Security, LNCS 2437. Oct. 2002, pp. 162–179.
[7] Indrajit Ray, Kirill Belyaev, Mikhail Strizhov, DieudonneMulamba, MariappanRajaram "Secure Logging As a Service—Delegating Log Management to the Cloud" IEEE SYSTEMS JOURNAL, VOL. 7, NO. 2, JUNE 2013
[8] J. Kelsey, J. Callas, and A. Clemm, Signed Syslog Messages, Request for Comment RFC 5848, Internet Engineering Task Force, Network Working Group, May 2010.
[9] C. Lonvick, the BSD Syslog Protocol, Request for Comment RFC 3164, Internet Engineering Task Force, Network Working Group, Aug. 2001.
[10] D. Ma and G. Tsudik, "A new approach to secure logging," ACM Trans. Storage, vol. 5, no. 1, pp. 2:1–2:21, Mar. 2009.
[11] D. New and M. Rose, Reliable Delivery for Syslog, Request for Comment RFC 3195, Internet Engineering Task Force, Network Working Group, Nov. 2001.
[12] D. New and M. Rose, Reliable Delivery for Syslog, Request for Comment RFC 3195, Internet Engineering Task Force, Network Working Group, Nov. 2001.
[13] M. Rose, The Blocks Extensible Exchange Protocol Core, Request for Comment RFC 3080, Internet Engineering Task Force, Network Working Group, Mar. 2001.