Asian Journal of Engineering and Applied Technology
ISSN: 2249-068X Vol. 9 No. 1, 2020, pp.13-17

Cloud Based Secured Log Retrieval Using Fully Homomorphic Encryption

B. Rasina Begum[1] and P.Chithra[2]
[1]Research Scholar, Department of Computer Science and Engineering, Mohamed Sathak Engineering College, Kilakarai, Ramanathapuram, Tamil Nadu, India
[2]Professor, Department of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai, Tamil Nadu India
E-mail: rasinayousuf@gmail.com, pccse@tce.edu

Abstract **- Cloud computing provides a scalable platform for growing amount of data and processes that work on various applications and services by means of on-demand service providing. The storage services offered by clouds have become a new profit growth by providing a comparably cheapest, scalable, location-independent platform for managing users' data. Client uses the cloud storage and enjoys the high-quality applications and services from a shared pool of configurable computing resources using cloud services. It reduces the trouble of local data storage and maintenance. But it gives severe security threats toward users' outsourced data. Maintaining the log record securely over extended period is very crucial to the proper functioning of any organization. Log holds the record of system events specially user activities. It is the main target for malicious attackers. An attacker, breaking into a system would not try to leave traces of his or her activities behind. This proposed work mainly concentrates on the problem of log security in cloud computing system and using fully homomorphism encryption algorithm in the cloud computing log data.**
Keywords: **Cloud Security; Logging; Fully Homomorphic Encryption; Decipherment**

I. INTRODUCTION

Organization need to rapidly increase their businesses, which may enforce them to enlarge the IT infrastructure by adding storage devices, new servers, network bandwidth, etc. Mission Critical business data must be protected, and it should be available to the intended user, which in turn requires data security and disaster infrastructure. As the capital expenditure rises to fulfill the requirements, the risk associated with the investment also increases. For small and medium sized businesses, this may be a big problem, which eventually affect their business to grow. As an individual, it may not be affordable every time to purchase new applications if they are required only for a short period. Instead of purchasing needed resources, Cloud resources are hired based on pay-per-use without involving any CAPEX. CSP provide on-demand network access configurable computing resources such as servers, network, storage and applications. Users can scale up or scale down the demand of computing resources with minimal management effort or CSP interaction. Users can influence cloud service provider's expertise to store, process, protect, backup and replicate the data empowered by the most advanced technology, which otherwise would cost more. Although the cloud computing has become a good service model, and have a large commercial, cloud computing is still facing many issues. It brings a lot of security threats to the outsourced data. As a result, the correctness of the information within the cloud is place with risks. The threats in cloud side occurs from time to time. The remedy to preserve the data is to encrypt the data before uploading the data to the cloud and user must send only the encrypted query for retrieval.

Log is a record of all system events [1]. For example, Log files in web server have all requests submitted to the server. With the help of this tool, it is possible to identify the outsiders who used the system and all their activities. When system has severe security issues, Log files are used to identify the critical incidents, policy violation, unauthorized activities, and all operational problems. This kind of tools are highly helpful for auditing and all forms of forensic analysis, preparing baselines and find long term security problems. Log files have many types namely Request Log file, Manager Log file and internal concurrent manager log file. Again, there are two types called event log and transaction log. Event log records all events during the execution to provide audit trail. This audit trail helps to understand the system activities for diagnosing the problems. Database system must maintain the transaction log. This is the history of actions executed by Database Management system to check the ACID properties.

Originally logs were used mainly for troubleshooting problems[7], but logs now serves many applications in many organizations, such as optimizing the system and network performance, recording the actions of the users, and providing the data useful for investigating malicious activity. Due to advancements in security, nowadays log contains information related to many different types of events related to networks and systems. Within an business enterprise, many logs used to have records related to system security; common examples of these computer security logs are audit logs that contains user authentication attempts and security device logs that record possible attacks. Basic problem with log management that occurs in many organizations is effectively balancing a limited quantity of

log management resources with providing the log data continuously. Generation of logs and storing them became complicated by following factors, including a higher number of log sources; inconsistent content of log, formats, and timestamps among sources; and increasingly large volume of log data.

Organizations need to protect their logs. Many logs have maximum size as holding the thousands of most recent events or keeping hundred megabytes of log data. When the threshold is reached, the log might overwrite the old data with the new data or stop logging altogether, both of which would cause a loss of log data availability. To satisfy data retention requirements, organization are required need to keep copies of log files for the longer period than the original log sources can support, which necessitates establishing log storing system. Because of the increase in size and number of logs, it will be preferable in some cases to reduce the logs by filtering out the entities that are not required to archived. The confidentiality and Integrity of the archived log file need to be protected by storing the log in encrypted form.

II. THE THREAT MODEL

A log entry consists of a date (time) and description of events. A well experienced attacker will try to compromise log data pertaining to the past during breaking into the system: he wishes to alter, or erase, any entries pertaining to his current or past login attempts. In many of the real-world applications, the log files are used to generated and stored on an untrusted machine which is not physically secure enough to guarantee that it cannot be compromised. Most of the system component responsible for logging-is not totally error-free, which is unfortunately always the case. In systems using remote logging, if the server is unavailable, then the log has to be buffered and stored temporarily at the local machine once an attacker obtains the current secret key of the log machine, he can modify the post-compromise data at will, in this case, the favourable step is to forward integrity: how to ensure that pre-compromise data cannot be modified. That is, even if the attacker obtains the current secret key, he must be unable to modify the audit data generated before intrusion. An attacker who gains access to the system naturally wishes to remove the trace of his presence to hide attack details or frame from innocent users.

In fact, first target of the well experienced attacker would often be the logging system. To make the log secure, one must prevent the attacker from modifying log data. Secure versions of log should be designed to defend against tampering.

The capabilities of attackers [3] are

1. He can intercept any messages sent over the internet.
2. He can synthesize, replicate, and replay messages in his possession.
3. He can be the legitimate participant of the network or can try to impersonate legitimate hosts.

III. SYSTEM ARCHITECTURE

The architecture of the cloud based secure log management system [4] is shown in fig. 1.
There are four major functional components namely

A. Log Generators

It is used for generation of the log data. Every organization that uses the cloud-based log management service has a few log generators. These generators are built with logging capability. The log files generated by the host are not stored locally, but they are pushed to the logging client.

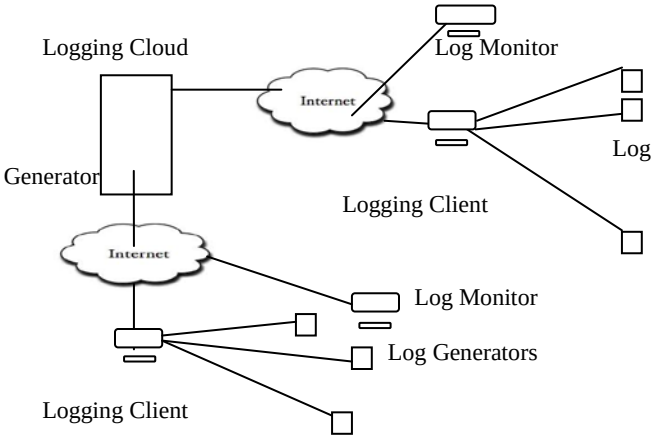

Fig 1 System architecture for cloud based secure logging

B. Logging Client or Logging Relay

The logging client is a collector that receives set of log records generated by one or more number of log generators, and produces the log data so that it could be pushed to the cloud for storing them for a long term. The log data is transferred from the generators to the client in batches, either on schedule, or as and when needed depending on the amount of log data waiting to be transferred.

C. Logging Cloud

The logging cloud provides the long-term storage and maintenance service to the log data received from different logging clients belongs to several organizations. The logging cloud is maintained by a cloud service provider. The organizations that have subscribed to the logging cloud services can upload their log data to the cloud. The cloud, on request from an organization can also delete log data and perform log rotation. logging cloud requires a proof from the requester that the latter is authorized to make such a request as a previous step to deleting or rotating log data. The logging client generates such a proof. However, the proof can be given by logging client to any that it wants to authorize.

D. Log Monitor

Log monitor is used to monitor and review log data. It is used to generate queries to retrieve log data from the cloud. Based on the log data retrieved, these will perform further analysis as needed. It will also query the log cloud to delete log data permanently, or to rotate logs.

IV. PROPERTIES OF SECURE LOG

The following are the properties hold by secure log [4].

A. Verifiability

It ensures that the content of log has not been modified by unauthorized way and available as it is. All entries must have the sufficient information for its verification.

B. Confidentiality

Log records usually hold sensitive information. It is essential to prevent sensitive information from reaching the wrong people, while making sure that the right people can in fact get it: Access must be restricted to those authorized to view the data.

C. Tamper Resistance

A secure log should be tampering resistant in such a way that no valid entries can be introduced by anyone other than the creator. In addition, once those entries are created, they cannot be manipulated without detection. No one can prevent an attacker, who has compromised the logging system from altering what that system will put in future log entries.

D. Correctness

Log record should reflect the true history of the system rather than wrong information. Once the log record is stored in storage, it will not be edited or modified by the attackers.

E. Privacy

Log records should not be casually traceable or linkable to other sources during transmission and in storage.

V. PROPOSED SCHEME

A. Privacy Protection

User transmits and save their log to the cloud by encrypted form. Both ensure the security of log data in process of transmission and ensure safe storage of data. Although the cloud computing service provider handle, they can't easily obtain information of plain text.

B. Data Processing

Fully Homomorphic encryption mechanism enables users or the trusted third-party process cipher text data directly, instead of the original log data. Users can obtain arithmetic results to decrypt to get good data.

C. The Cipher Text Retrieval

Fully Homomorphic encryption technology based on cipher text retrieval method can search directly on the cipher text data. It is not only ensuring the query privacy and improve the efficiency of retrieval, the retrieval data can be added and multiply without changing the corresponding plaintext.

VI. HOMOMORPHIC ENCRYPTION

In both private and public cloud, data should be encrypted to ensure security [5]. Data can be transmitted from local machine to cloud data store through secured way. The stored data being encrypted and secured key exchange is also possible. But performing computations, the data stored in the cloud should be decrypted first. This makes the sensitive data available to the Cloud service provider. To achieve the security completely, it is necessary for the crypto system which is based on Homomorphic Encryption.

Homomorphic encryption is one of the theoretical advancements in science to keep secrets in data. It is a cryptosystem that compute data without decrypting it. It allows complex mathematical operations to be performed on cipher text without decryption. This Homomorphic encryption plays crucial role in cloud computing that allows companies to store the encrypted data in a public Cloud. It has two types namely Fully homomorphic encryption (FHE) and Somewhat Homomorphic Encryption (SHE).

Fully Homomorphic encryption [2] is an encryption scheme that allows arbitrarily complex programs on cipher text. It allows one to perform arbitrary computation over encrypted data. This scheme can perform addition and multiplication on cipher text. But Somewhat Homomorphic Encryption (SHE) supports one of the two operation.

There are two types of fully homomorphic encryption algorithm namely multiplicative homomorphic encryption algorithm and additive homomorphic encryption algorithm. Additive Homomorphic encryption algorithm supports only addition homomorphism and multiplicative homomorphic encryption supports multiplication homomorphism only. This paper suggests fully homomorphic algorithm to find an encryption, which can be any number of addition algorithm and multiplication algorithm in the encrypted information.

For example, Cloud user wants to store the encrypted form of data in cloud storage. So, Cloud user has to do the following for enforcing security.

1. User has very important data set ABCD that consists of the numbers 10 and 20. For encrypting the data, Cloud user must simply multiply the data by 2. After multiplication, the new data set numbers are 20 and 40.
2. Now send the cipher text to the cloud storage. Now the data are available in cloud storage in encrypted form.
3. After some months, Cloud user requests the Cloud provider to perform the addition and multiplication of those two data. Cloud service provider performs the addition and multiplication of encrypted data 20 and 40 without the knowledge of original data 10 and 20.
4. Cloud service provider sends the addition result 60 and multiplication result of 800 to the user. Now he performs decryption on 60 and 800 and gets the result 30 and 200.
   This kind of processing is called blind processing because the cloud provider does not know about original data.

A. Encryption and Decryption Algorithm

1. Encryption Algorithm

Consider the encryption parameters p, q and s, where p is positive odd number, q is large positive integer, p and q determined in the generation phase, p is an encryption key, and s is a random number encrypted when selected.
For the text M, calculation

$$C=M+2s+pq$$

Then you can get the cipher text.

2. Decipherment Algorithm

To plain text,

$$M= (C \bmod p) \bmod 2$$

Because the p×q is much less than 2s+M, then

$$(C \bmod p) \bmod 2= (2s+M) \bmod 2=M$$

B. Homomorphism Verification

1) The Homomorphism Addictive Property Verification

Suppose there are two groups of the plain text M1 and M2, to encrypt them become the cipher text,

$$C_1=M_1+2s_1+pq_1$$
$$C_2=M_2+2s_2+pq_2$$

To plaintext

$$M_3=M_1+M_2$$

Due to $C_3=C_1+C_2 = (M_1+M_2) + 2(s_1+s_2) + p (q_1+q_2)$

As long as

$(M_1+M_2) + 2(s_1+s_2)$ is much less than p,

$$C_3= (C_1+C_2) \bmod p= (M_1+M_2) + 2(s_1+s_2)$$

This algorithm satisfies the additive homomorphism conditions.

2) The Homomorphic Multiplicative Property Verification

To plaintext $M_4=M_1 \times M_2$

Due to

$C_4=C_1 \times C_2 = (M_1+2s_1+pq_1) \times (M_2+2s_2+pq_2) =$
$M1M2+2s_2M_1+M_1pq_2+2M_2s_1+4s_1s_2+2s_1pq_2+M_2pq_1+2pq_1s_2+p^2q_1q_2=M_1M_2+2(2s_1s_2+s_1M_2+s_2M_1) +$
$p^2q_1q_2+q_2(M_1p+2s_1p)+ \quad q_1(M_2p+2s_2p)$
$= M_1M_2+2(2s_1s_2+s_1M_2+s_2M_1) + p[pq_1q_2+q_2(M_1+2s_1)+$
$q_1(M_2+2s_2)]$
As long as $M_1M_2+2(2s_1s_2+s_1M_2+s_2M_1)$ is less than p

Then,

$$C_4 = (C_1 \times C_2) \bmod p= M_1M_2+2(2s_1s_2+s_1M_2+s_2M_1)$$

This algorithm satisfies the multiplicative homomorphism conditions.

C. Log Retrieval Using Fully Homomorphic Encryption

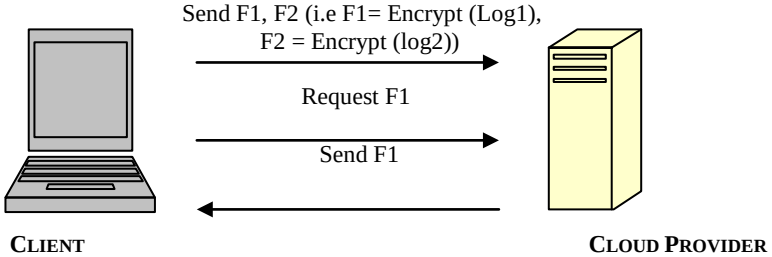

Fig.2 Secured log transmission between client and cloud provider

The log transmission using fully homomorphic encryption [6] is depicted in Fig.2. Instead of storing the original data Log1 and log2, Client has to encrypt the log data and transmit it to Cloud. When the client wants to retrieve the

data, he must send the query in encrypted form and the cloud provider has to transmit the required data in encrypted form. Client will decrypt it and get the original log.

VII. CONCLUSION AND FUTURE WORK

Cloud computing does not process the data on the user's computer, but in the cloud. So, the data has to be encrypted first and then uploaded to the cloud for process. This Fully Homomorphic encryption algorithm helps to attain confidentiality because of the calculations done in encrypted form without knowing the original form. It is very important for the data safety of the cloud computing platform because the data is invisible to the third party and can be processed by the cloud itself.

This paper suggests the fully homomorphic encryption mechanism for processing log data to propose a cloud computing data security scheme. This scheme ensures the transmission of log between cloud and the user safely. Even in cloud storage, their data is safe. At present, fully homomorphic encryption scheme has high computational problem. Future work will be eliminated that overhead.

REFERENCES

[1] K. Kent and M. Souppaya, "*Guide to Computer Security Log Management*", NIST Special Publication.

[2] Feng Zhao, Chao Li and Chun Feng Liu, "A cloud computing security solution based on Fully homomorphic encryption", *ICACT2014*, February 2014

[3] V. Varadharajan and U.Tupakula , "Security as a Service Model for Cloud Environment", *IEEE transactions on Network and Service Management*, Vol. 11, No 1, March 2014

[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] Shahank Bajpal and Padmija Srivastava, "A Fully Homomorphic Encryption Implementation on cloud Computing", *International Journal of Information & Computation Technology*, Vol. 4, No. 8, 2014

[6] Maha Tebaa, Saïd El Hajji and Abdellatif El Ghazi, "Homomorphic Encryption Applied to the Cloud Computing Security", *World Congress on Engineering*, 2012

[7] Badugu Rajeswari and Badugu Suneel Kumar, "CLASS: Cloud Log Assuring Soundness and Secrecy scheme for cloud Forensics", *International Journal of Engineering Technology Science and Research*, Vol. 6, No. 7, July 2019.