A Survey on CLASS: Cloud Log Assuring Soundness And Secrecy Scheme For Cloud Forensics

Fazal Shama

PG Scholar, Department of Computer Science and Engineering, KBN College of Engineering, Kalaburagi

Abstract—Activity logs of users can be an important source of information in investigation of cloud. Therefore, it is critical to ensure the reliability and integrity of such logs. Most of the existing solutions for secure logging are proposed for conventional systems instead of the complexity of a cloud environment. An alternative scheme for the securing of logs name as Cloud Log Assuring Soundness And Secrecy (CLASS) process is design. In CLASS, logs are encrypted using the user's public key so that only the user is able to decrypt the content. The proof of past log (PPL) is generated using Rabin's fingerprint and Bloom filter. This approach reduces verification time significantly. Deploy the CLASS in Open Stack to demonstrate the utility of CLASS in a real-world context.

Keywords: Cloud forensics, Cloud log, Cloud log assuring soundness and secrecy, Cloud security, Proof of past log, Sustainable computing

1. INTRODUCTION

Cloud storage, security and privacy are research areas which is not unexpected to consider the extensive adoption of cloud services and the ability for criminal slavery (e.g. compromising cloud accounts and servers for the stealing of sensitive data) but cloud forensics is a relatively less understood topic. The event in which cloud service, cloud server, or client device has been damaged or involved in malicious cyber activity (e.g. used to host illegal contents, or conduct distributed denial of service (DDoS) attacks) then forensic investigation must be conduct by investigators to answer the questions of an incident – what, why, how, who, when, and where.

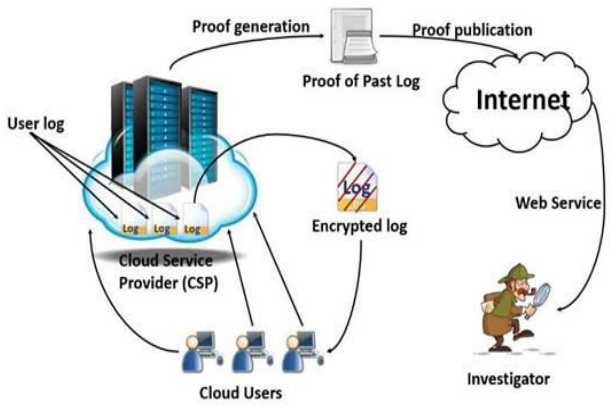

Fig. Overview of CLASS scheme process

Because of the inherent nature of cloud technologies, regular forensic procedures and tools need to be updated to remain the useful and applicable in a cloud environment. Cloud virtual machines (VMs) can be supported by hardware that might be located remotely and thus would not be physically accessible to an investigator. Morever, VMs can be placed across multiple physical devices in a clustered environment. Data preserved in a VM may be modified and could be lost once the power is off or the VM terminates. Hence, CSP plays a important role in the collecting the evidential data. For example, the CSP writes
the activity log for each user. Such data must be kept confidential to ensure the privacy and enable the investigation activities.

The notations used in this scheme is present in below table.

SUMMARY OF NOTATIONS

Log Chain (LC)	LC is a small piece of information that coexists with its corresponding log.
Proof of Past Log (PPL)	PPL is a signature or information about the actual log that will be available publicly for forwarding secrecy.
Cloud Service Provider (CSP)	CSP is a cloud service provider in which a user can rent and use computing and storage resources.

2. LITERATURE SURVEY

Earlier, Zawoad et al. design a secure logging services called "secLaas" [1] to collect data from many log sources, parse the data and then store that data in persistent storage to reduces the risks of data volatility. Before to store the data, it encrypts the log and produce the log chain to ensure integrity. This service encrypts the logs using investigating agency's public key and stores in cloud server. In this scheme, it is difficult to ensure or verify that CSP is writing correct information to the log, or that any information require for investigation is deleted or changed. This scheme possess some weakness, namely combining of cloud data, volatility , and malicious loggers.

Anwar et al. [2] try to address some challenges such as acquisition, integrity, forward secrecy by identifying the possibility of Syslog or snort log to help in the detection of cloud attacks. he conducted cloud forensic investigations based on the logs produced by Eucalyptus. They identified attacking device IP address by generating their own dataset by imitate a DDoS attack on Eucalyptus . Security, access control, and verification of log were not considered.

Bellare and Lee proposes forward integrity [3] as a system property to reduce an attacker's capability to damage a logging system without detection. Forward integrity is established using a one-way hash function (i.e. HMAC) and a secret key. i.e, every successive log entry has an associated hash key that is dependent upon the previous entry.

Log management scheme based on forward integrity is designed by Schneier and Kelsey [4].The forward integrity property in this approach is ensured using a secret key which is the initial point of a one-way hash chain and message authentication code. Such schemes [3,4] require an online trusted server to maintain the secret key and to ensure its integrity.

Holt proposed "Logcrypt" using public key cryptography [5] to avoid the necessity of an online trusted server. In this scheme, instead of using the one-way hash function, he used a digital signature storing in the trusted server. Drawback of the public key cryptography is the associated computational overheads and these approaches do not consider the privacy protection.

Ma and Tsudik proposed the new concept of secure logging[6] to address the truncation and delayed detection challenge. In this approach, two tags are associated with each log entry, namely: one is for the semi-trusted log accumulator and other is for the trusted verifier. Using these tags they were able to ensure the integrity of forward secure stream rather than forwarding security. Also, the last tag of an epoch can `testify' the entire chain of log entries up to that epoch. However, this approach suffers from additional computation costs during the verification phase.

3. A STUDY OF THE ALGORITHM

The algorithm use in this scheme can be divided into two main groups: One for LogPreservation and one for ProofAccumulation [7].

```
LogPreservation( log entries LEs)
    for i ← 1 to size( LEs )
        encrypted log_i = encrypt( log entry_i )
        log chain_i = hash( encrypted log_i || log chain_{i-1} );

        Database_log_entry_i = < encrypted_log_i, log _chain_i >;
        store database log entry_i into log database;
    end
    for;
end;
```

Algorithm 1. LogPreservation pseudocode for processing log Entries

```
ProofAccumulation( log entries LEs)
    chronological concatinate LEs = LE_1 || LE_2 || ... || LE_n;
    finger_print = FingerPrint( chronological concatinate LEs );
    accumulator entry = BloomFilter( finger print );
signature=Signature(acuumulator entrytime);

Publish < accumulator_entry. time. signature >; end;
```

Algorithm 2. ProofAccumulation pseudocode to generate and publish proof of past log (PPL)

- Parser collects the log from log source.
- Retrieving log from log source, the parser parses the

log and gets the necessary information.

- Asymmetric encryption of log with individual user's public key is computed to conceal user's content.
- After that, log chain is created in order to protect the integrity of the log and prevent potential manipulation.
- At this stage, the payload is ready to be stored in the databases.
- For each database log entry, the CLASS scheme requires the generation of proof of past log (PPL) which is then made publicly available.
- After the publishing of PPL, the parser cant verify their log, which is readily available in each epoch.

4. HOW IS THIS ALGORITHM USEFUL IN THE SCHEME

The Algorithm here defines the new activity which occurs in the cloud, the log file will changes (i.e new line append). Then Logpreservation algorithm triggers the parser to check the change and to start processing the new log. This algorithm can take log entries either individually or in a batch and performs processing before to store in a log database. This algorithm encrypts the logs with user's public key for the secrecy which enable the user to verify the accuracy of their log and generates hash digest for consistency. The goal is to keep the log content secure provided by the parser.

After the log entries are stored in the database, the Proofaccumulator algorithm performs daily processing of all log entries corresponding to an IP address to prepare and publish proof of past log (PPL) using Rabin's fingerprint and bloom filter. Since the Rabin's fingerprint has faster computation time. The accumulator of this CLASS scheme generate the PPL in a manner that is designed to minimize the usage of memory space. The PPL is made publically available so that user can check whether CSP is writing correct information. The forensics investigator is also benefited by the proof of past log (PPL) to investigate the malicious behavior.

5. CONCLUSION

The secure logging scheme CLASS for cloud computing facilitate the preservation of user privacy and reduces the damaging effects that arises by collusion among other parties. This scheme preserves the privacy of cloud users by encrypting cloud logs with a public key of the respective user while also facilitating log retrieval in the event of an investigation. It allows the user to identify the log modification hence ensures accountability of the cloud server. It has the additional effect of preventing a user from repudiating entries in his own log once its PPL established.

REFERENCES

[1] S. Zawoad, A. K. Dutta, and R. Hasan, "Towards building forensics enabled cloud through secure logging-as-a-service," IEEE Transactions on Dependable and Secure Computing, vol. 13, pp. 148-162, 2016.

[2] F.Anwar and Z.Anwar, "Digital forensics foreucalyptus," in Frontiers of Information Technology (FIT), 2011, 2011, pp. 110-116

[3] M. Bellare and B. Yee, "Forward integrity for secure audit logs," Technical report, Computer Science and Engineering Department, University of California at San Diego1997.

[4] B. Schneier and J. Kelsey, "Secure audit logs to support computer forensics," ACM Transactions on Information and System Security (TISSEC), vol. 2, pp. 159-176, 1999.

[5] J. E. Holt, "Logcrypt: forward security and public verification for secure audit logs," in Proceedings of the 2006 Australasian workshops on Grid computing and e-researchVolume 54, 2006, pp. 203-211.

[6] D. Ma and G. Tsudik, "A new approach to secure logging," ACM Transactions on Storage (TOS), vol. 5, p. 2, 2009.

[7] Ahsan, M.A.M.; Wahab, A.W.A.; Idris, M.Y.I.; khan, S.; Bachura, E.; Choo, K.K.R, "CLASS: Cloud Log Assuring Soundness and Secrecy Scheme for Cloud Forensics". IEEE Trans. Sustain. Comput. 2018, 1–15.