ISSN: 2278-4632
Vol-12 Issue-01 No.01: 2022

PRIVACY PRESERVING CLOUD LOGGING SCHEME WITH PROOF OF PAST LOG (PPL) SCHEME IN CLOUD

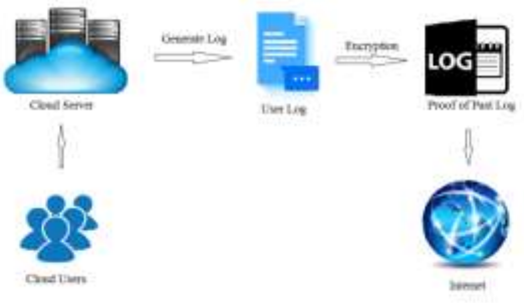
KALAKONDA LAKSHMI MOUNANKA, Student, M.Tech (CSE), VIKAS GROUP OF INSTITUTIONS, A.P., India.

KISHORE DASARI, Assistant Professor, Dept. of Computer Science & Engineering, VIKAS GROUP OF INSTITUTIONS, A.P., India.

ABSTRACT: In this paper, propose the Cloud Log Assuring Soundness and Secrecy (CLASS) process as an alternative scheme for the securing of logs in a cloud environment. In CLASS, logs are encrypted using the individual user's public key so that only the user is able to decrypt the content. In order to prevent unauthorized modification of the log, we generate proof of past log (PPL) using Rabin's fingerprint and Bloom filter. Such an approach reduces verification time significantly. User activity logs can be a valuable source of information in cloud forensic investigations; hence, ensuring the reliability and integrity of such logs is crucial.

INTRODUCTION:

C LOUD storage, security and privacy are fairly established research areas [1-7], which is not surprising considering the widespread adoption of cloud services and the potential for criminal exploitation (e.g. compromising cloud accounts and servers for the stealing of sensitive data). Interestingly though, cloud forensics [8- 10] is a relatively less understood topic. In the event that a cloud service, cloud server, or client device has been compromised or involved in malicious cyber activity (e.g. used to host illegal contents such as radicalization materials, or conduct distributed denial of service (DDoS) attacks) [11, 12], investigators need to be able to conduct forensic analysis in order to "answer the six key questions of an incident – what, why, how, who, when, and where" [13]. In addition, VMs can be distributed across multiple physical devices in a clustered environment or they can exist within a pool of VMs on the same physical components. Therefore, seizing the machine for forensic analysis is not viable in most investigations. Furthermore, data residing in a VM may be volatile and could be lost once the power is off or the VM terminates. Hence, the cloud service provider (CSP) plays a crucial role in the collection of evidential data (e.g. cloud user's activity log from the log). For example, the CSP writes the activity log (cloud log) for each user. Thus, preventing modification of the logs, maintaining a proper chain of custody and ensuring data privacy is crucial [15]. This research considers "activity log data" as any recorded computer event that corresponds to a specific user. Such data must be maintained confidentially to preserver user privacy and to facilitate potential investigative activities.



LITERATURE SURVEY

In 2016, Zawoad et al. proposed a secure logging service called "SecLaaS" [16] that is designed to collect data from one or more log sources, parse the data and then store the parsed data in persistent storage in order to mitigate the risk associated with data volatility. Prior to the storing of data, it encrypts the log and generates a log chain to achieve confidentiality and integrity respectively. SecLaaS encrypts the log(s) using the investigating agency's public key and stores the encrypted log(s) in a cloud server. This ensures privacy and confidentiality of the cloud user, unless the particular user is subject to an investigation (e.g. via a court order). To facilitate log integrity, SecLaaS generates proof of past log (PPL) with the log chain and publishes it publicly after each predefined epoch. A trust model was also suggested that stores the PPL in other clouds to minimize the risk of a malicious cloud entity altering the log. However, in SecLaaS, it is difficult to ensure or verify that the CSP is writing the correct information to the log, or that any information pertinent to the investigation is not omitted or modified. Specifically, SecLaaS does not provide the user the ability to verify the accuracy of the log (since the log is encrypted with the agency's public key). In other words, SecLaaS has limitations in addressing accountability and transparency enforced, especially from the perspective of the user.

PROPOSED APPROACH:

SecLaaS encrypts the log(s) using the investigating agency's public key and stores the encrypted log(s) in a cloud server. This ensures privacy and confidentiality of the cloud user, unless the particular user is subject to an investigation (e.g. via a court order). To facilitate log integrity, SecLaaS generates proof of past log (PPL) with the log chain and publishes it publicly after each predefined epoch. A trust model was also suggested that stores the PPL in other clouds to minimize the risk of a malicious cloud entity altering the log. However, in SecLaaS, it is difficult to ensure or verify that the CSP is writing the correct information to the log, or that any information pertinent to the investigation is not omitted or modified. Specifically, SecLaaS does not provide the user the ability to verify the accuracy of the log (since the log is encrypted with the agency's public key).

PROPOSED SYSTEM:

Extending SecLaaS, we propose a secure cloud logging scheme, Cloud Log Assuring Soundness and Secrecy (CLASS), designed to ensure CSP accountability (i.e. writing the correct information to the log) and preserve the user's privacy. Specifically, we include the capability for the user to verify the accuracy of their log. To do this, the log will be encrypted using the user's public key (rather than the agency's public key). To avoid introducing unnecessary delays to the forensic investigation, during user registration with the cloud service, both the CSP and the user will collectively choose a public/private key pair referred to as content concealing key (CC-key) for the user. The corresponding (content concealing) private key will be shared with other CSPs secret sharing schemes. This would allow the private key to be regenerated whenever necessary. We also demonstrate how we can leverage Rabin's fingerprint and bloom filter in PPL generation to establish log veracity. We then implement CLASS in OpenStack and evaluate its performance.

MODULES DESCRIPTION:

1. **Preservation Of Log &Its Proof**

   Parser collects the log from log source. When a log file changes (i.e. new lines append) it triggers the parser to check the change and to start processing new log.Retrieving log from log source, the parser parses the log and gets the necessary information.Our goal is to keep log content secure given a parser that will provide the system a log message in string format, regardless of content. The format of the log is out of the scope of this work.

2. **Accumulator Design**

Bloom filter as a proof of past data possession, which is fails to account for Bloom filter's inherent potential for false positives. When using a Bloom filter technique, there is a trade-off between the number of false positives and the size of the filter. To mitigate this problem, a cryptographic one-way accumulator could beused. However, this requires significant computational overhead. In SecLaaS, they used their own data structure Bloom Tree that reduced the number of false positive incidents but requires an increased number of instances of logs and significant computational resources at verification time. This is true regardless of the number of entries being verified. In addition, it still remains vulnerable to false positives (albeit reduced).

3. Verification

Only a verification process that establishes authenticity will be able to determine the presence of log contamination. There are two types of verifications in our approach. First is verification where the user checks if the CSP is writing correct log entries or not. Second is verification by any party: user, investigator, law enforcement authority (LEA) or court of law to verify PPL to check for log modification. In both cases, the CSP can provide a small utility verification software or the user/investigator can buy it from individual software vendor (ISV) to verify.

4. Secret Key Sharing

We propose, in CLASS, to encrypt the log with the user's private key (CC-key). In recognition that this might lead to permanent loss of log data (even for investigative entities), as the private key of a user's CC-key is known only to the user, we propose to share individual user's private key according to Shamir's or Blackley's secret key sharing strategy among multiple CSPs. This sharing scheme requires standardization. We can build sharing clouds for such a purpose when a user subscribes to a cloud service. The CSP and user together choose a pair of public/private key that is called the content concealing key (or CC-key) because it is used to hide user's log content.

SAMPLE RESULTS

CLASS Cloud Log Assuring Soundness and Secrecy Scheme for Cloud Forensics
CLASS: CLOUD LOG ASSURING SOUNDNESS AND SECRECY SCHEME FOR CLOUD FORENSICS

CLASS Cloud Log Assuring Soundness and Secrecy Scheme for Cloud Forensics
Welcome Investigator or Auditor
Menu

CLASS Cloud Log Assuring Soundness and Secrecy Scheme for Cloud Forensics
Search Ratio
Menu

CONCLUSION:

In this paper, we proposed a secure logging scheme (CLASS) for cloud computing with features that facilitate the preservation of user privacy and that mitigate the damaging effects of collusion among other parties. CLASS 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. Moreover, it ensures accountability of the cloud server by allowing the user to identify any log modification. This has the additional effect of preventing a user from repudiating entries in his own log once the log has had its PPL established. Our implementation on Open Stack demonstrates the feasibility and practicality of the proposed scheme. The experimental results show an improvement in efficiency thanks to the features of the CLASS scheme, particularly in verification phase.

REFERENCES:

[1]X. Liu, R. H. Deng, K.-K. R. Choo, and J. Weng, "An efficient privacy-preserving outsourced calculation toolkit with multiple keys," IEEE Transactions on Information Forensics and Security, vol. 11, pp. 2401-2414, 2016.

[2]Y. Mansouri, A. N. Toosi, and R. Buyya, "Data storage management in cloud environments: Taxonomy, survey, and future directions," ACM Computing Surveys (CSUR), vol. 50, p. 91, 2017.

[3]M. Tao, J. Zuo, Z. Liu, A. Castiglione, and F. Palmieri, "Multi-layer cloud architectural model and ontology-based security service framework for IoT-based smart homes," Future Generation Computer Systems, vol. 78, pp. 1040-1051, 2018.

[4]Z. Xia, Y. Zhu, X. Sun, Z. Qin, and K. Ren, "Towards privacy-preserving content-based image retrieval in cloud computing," IEEE Transactions on Cloud Computing, pp. 276-286, 2018.

[5]L. Zhou, Y. Zhu, and A. Castiglione, "Efficient k-NN query over encrypted data incloud with limited key-disclosure and offline data owner," Computers & Security, vol. 69, pp. 84-96, 2017.

[6]Q. Alam, S. U. Malik, A. Akhunzada, K.-K. R. Choo, S. Tabbasum, and M. Alam, "A Cross Tenant Access Control (CTAC) Model for Cloud Computing: Formal Specification and Verification," IEEE Transactions on Information Forensics and Security, vol. 12, pp. 1259-1268,2017.

[7]L. Li, R. Lu, K.-K. R. Choo, A. Datta, and J. Shao, "Privacy-preserving-outsourced association rule mining on vertically partitioned databases," IEEE Transactions on Information Forensics and Security, vol. 11, pp. 1847-1861, 2016.

[8]K.-K. R. Choo, M. Herman, M. Iorga, and B. Martini, "Cloud forensics: State-of-the-art and future directions," Digital Investigation, pp. 77-78, 2016.