Sustainable Computing For Cloud Logs For Cloud Forensics

Ajay H C[1], Swathi C[2], Rabiya taj N[3]
Department of Computer Science Engineering
[2,3] S J C Institute of technology, Karnataka, India
[1] Assistant Professor S J C Institute of technology, Karnataka, India

Abstract- *User undertaking logs which contain important data in cloud investigate crimes; therefore, confirming and providing trust and honesty of those logs is needed. In current compounds for protecting logs are planned for ordinary structure rather than the difficulty of a cloud surroundings. In this, we initiate sustainable computing of cloud logs for cloud forensics as a different plan for protecting logs in cloud environment. Sustainable Computing of cloud logs for cloud forensics, logs are converted to a cipher text utilizing the single user's public key so that the user is able to convert it into a plain text. 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.*

Keywords- logs, cloud forensics, proof of past log, and sustainable computing

I. INTRODUCTION

CLOUD storage, security and privacy are fairly established research areas 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 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 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".
Due to the inherent nature of cloud technologies, conventional digital forensic procedures and tools need to be updated to retain the same usefulness and applicability in a cloud environment. Unlike a conventional client device, cloud virtual machines (VMs) can be supported by hardware that might be located remotely and thus would not be physically accessible

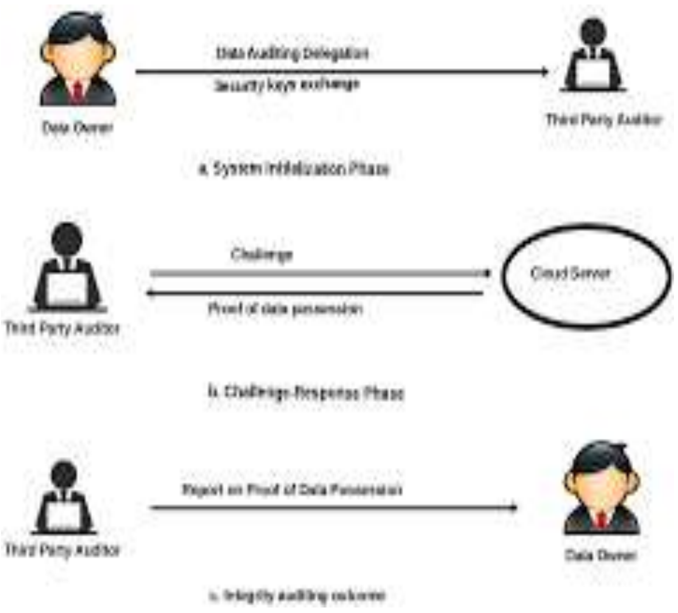

Figure 1.Overview of sustainable computing of cloud logs for cloud forensics.

(e.g. out of the jurisdictional territory) to an investigator.

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 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 .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.

Extending SecLaas, we propose Sustainable computing of cloud logs for cloud forensics designed to ensure CSP accountability (i.e. writing the correct information to the log) and preserve the user's privacy–i.e. our contribution in this paper. 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 using Shamir's or Blakley's 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 sustainable computing of cloud logs for cloud forensics in Open Stack and evaluate its performance the threat model and secure logging system requirements in the SecLaaS scheme is discussed and reviewed in Our proposed scheme is presented in An evaluation of how the proposed scheme meets the security properties a complexity evaluation is covered in Performance evaluation is covered in to include a discussion on implementation and setup.

II .THREAT MODEL & SECURITY PROPERTIES

In this section, we will describe some definitions required to understand our scheme, the threat model , an attacker's capability, possible attacks and the standard security properties that a secure cloud logging system must possess. A summary of notations used in this paper is shown in Table 1.

Table -1: SUMMARY OF NOTATIONS

Log	Log can be network log, process log, registry log, application log or any customized text that meets the requirement of being stored for investigation purpose.
Log Chain (LC)	LC is a small piece of information that co-exists with its corresponding log in order to maintain the integrity and to prevent any modification of the log (such as addition, modification, deletion, and reordering).
Proof of Past Lo g (PPL)	PPL is a signature or information about the actual log that will be available publicly for forwarding secrecy [22]. That means if the system is compromised, an attacker cannot change the log without detection. PPL can be used to establish log veracity.
Cloud Service Provid er (CSP)	CSP is a cloud service provider in which a user can rent and use computing and storage resources. We assume that a CSP is honest but curious [30]. That means it will serve according to contract agreement but has a curiosity about client activity. We design our (SUSTAINABLE COMPUTING OF CLOUD LOGS FOR CLOUD FORENSICS) scheme to include features to prevent a dishonest CSP.
User	User is a CSP client.
Investigator	An investigator is an individual or entity with legal authority to conduct investigative activities in response to some event. These activities include accessing and assessing the contents of log files supplied by a CSP. It is possible for an investigator to collude with a malicious user or CSP to manipulate the perception of an event.
Content	CC is a strategy that helps to withstand against privacy breaches that are the result of collusion between a malicious cloud employee and an investigator.
Content Co ncealing Key	CC key is a pair of the private-public key that is used for concealing log content.CC key (the private key not the public key) should be shared by Shamir's [17] or Blakley's [18] secret sharing approach among some trusted entities.

Auditor	An auditor is an individual or entity who is authorized to verify the integrity of log entries, typically through techniques such

2.1Threat Model

Our plan is designed based on the "trusted person". Any party among the CSP, investigator, and user, should be capable of protecting its own security and privacy against another party or collusion between other parties. Potential challenges to designing forensic enabled cloud logging have been discussed in a number of previous studies. For example, in an insecure cloud logging model, only the CSP can write to a log. An investigator or user can collude with the CSP to modify a log before or after publishing PPL. Thus, if a CSP falsely alters a log, whether in collusion with a malicious user or investigator or not, it can hinder the investigative process and conceal the truth of an event. This could result in an attacker failing to be identified or, more dangerously, attributing the attack to the wrong entity.

Conversely, as the cloud is the host of multiple users, a malicious user can repudiate the log under investigation as his/her own log which can lead the criminal lawsuit to be dismissed. On the other hand, the log contains secretive data of the user and the user's privacy may be vulnerable due to this fact. A malicious investigator can alter the log before presenting to court authorities. Moreover, in collaboration with dishonest CSP or CSP employee(s), the investigator can violate the privacy of the user. Based on the above discussion, possible attacks on secure cloud log are given below:

2.1.1Modification of Log:

A dishonest CSP can modify the log before or after publishing its proof (PPL) upon or beyond collusion with the user or investigator. A malicious investigator may alter the log before presenting to court to save a dishonest user or to frame an honest user. Modification of a log can be of many forms, such as insertion of invalid entries, removal of the crucial entries, changing existing entries, reordering log entries to mislead the investigation and to hide malicious activities.

2.1.2Privacy Violation:

Leakage of a log file can reveal information that is able to be directly linked to a users' identity or is able to aggregate in such a way as to create such a link. Even with cryptographic security, cloud employees can transfer the log to an entity that has the key to decrypt (i.e. an investigator) and thus privacy violation may take place.

2.1.3Repudiation of Ownership of Log:

Cloud servers host many users. This presents the possibility for a malicious cloud user to repudiate that the log files under investigation represent the activity of another user. On the other hand, a CSP can repudiate that it did not write the log under investigation. Likewise in SecLaaS, the CSP writes a log for every user and the user has no visibility regarding log entries. This may raise user suspicions regarding log veracity and credibility.

2.2 Security property:

Sustainable computing of cloud log for cloud forensics seeks to achieve three properties of cryptography, namely: confidentiality, integrity, and authenticity, in terms of the following criteria.

Correctness:

Cloud logs should reflect the correct history of a system's event with the occurring time. Any distortion to it is considered a violation of the correctness property.

Tamper Resistance:

No one except real logger can introduce an invalid log entry as a valid one. Any sort of contamination such as the addition of new log entries, modification or deletion of existing log entries or even reordering of log entries requires prevention. At a minimum a tamper resistant scheme prevents an attacker from modification of logs without detection.

Verifiability:

Verification should be possible by both the user whose activity is represented in the log and the investigating entity. The auditor or any other party involved in the related litigation need to be able to establish log veracity.

Confidentiality:

Log data contains sensitive user information and requires privacy protection. For example, if a user mistakenly puts their password into the username field, the system will record this as a failed sign-in attempt and store the password as username in the log. This illustrates the need for confidentiality for all logged data in addition to data more traditionally viewed as requiring privacy protections.
Admissibility: A secure cloud log should be maintained in a way that allows it to be admissible in a court of law for criminal prosecution. The features of log integrity (correctness

and tamper resistance), a chain of custody, and forward secrecy all help to achieve such admissibility.

proOPOSED SCHEME: Sustainable Computing of cloud logs for cloud forensics

In this section, we improve on SecLaaS and sustainable computing of cloud logs for cloud forensics present scheme. We are assuming that in a cloud infrastructure, no party is trusted, that means an attack can come from any party: a CSP, user, or investigator. We are also assuming that cryptographic primitives work properly (i.e. if someone encrypts a message, then nobody can decrypt it without knowing the secret key).

System Overview

A dishonest cloud user can attack a system outside the cloud. They can also attack any application deployed in the same cloud or an attack can be launched against a node controller which controls all the cloud activities. For a virtual machine (VM), sustainable computing of cloud logs for cloud forensics scheme (Fig. 1) takes the log from the node controller (NC), hides its content, and stores it in a database. This allows logs to become available for further investigation despite VM shutdown. Moreover, sustainable computing of cloud logs for cloud forensics publishes its proof so that log integrity can be protected and admissibility ensured.

System Details

Before a detailed examination of the proposed scheme, the following definitions and notations are provided:

- M1||M2: concatenation between two messages M1 and M2.
- H(m): collision-resistance one-way hash function of message m.
- •EPK(m): encryption of message m with the public key (PK).
- SignatureSK(m): signature of message m with private key SK.
- EK(m): encryption of message m with symmetric key K.

We assume that the cloud service provider (CSP), law enforcement agency (LEA), and user set up their necessary public/private key pairs and publish public keys. PKC and SKC are the public and private keys of the CSP respectively and PKA and SKA are the public and private keys of the LEA (or auditor). PKU and SKU are the public and private keys of a particular cloud user. PKU and SKU are titled together as CC-key because they are used to conceal log content of a particular user. During a user's subscription time, the CSP and user mutually generate a CC-key for the user and the CSP only keeps the public portion of the CC- key. The private key (SKU) of the CC-key is shared among multiple clouds using Rabin‘s or Blakley's secret sharing scheme.

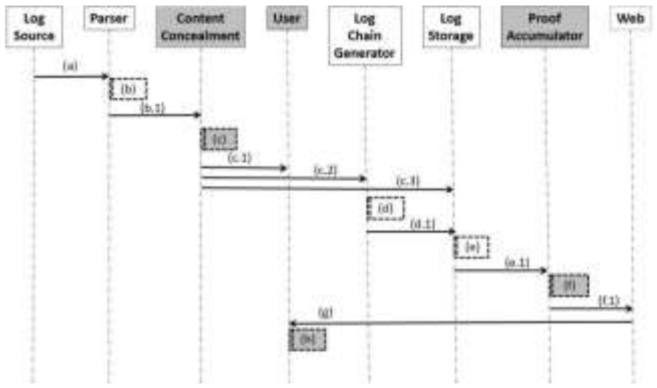

Fig. 2. Proposed sustainable computing of cloud logs for cloud computing scheme (new algorithms shaded

Preservation of Log & Its Proof

Fig. 2 depicts details of sustainable computing of cloud logs for cloud forensics from log retrieval to proof of past log (PPL) publication. Modified portions are shaded in grey. For illustration, we have chosen the network log but it can be any log of the system. Details of our scheme are given in the following step-by-step list:

a. Parser collects the log from log source. For example, we collect log from log file stored in a specific directory. 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. For the time being there is no standard format of a log and this is another challenge of cloud forensics . The how, when, where, and who, aspects of logs are not yet standardized. 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. For our example, we choose originating IP (FromIP), destination IP (ToIP), user id (UserID) and log time (TL). With this information, the parser generates log entry (LE).
LE = <FromIP, ToIP, UserID, TL >

b. Asymmetric encryption of log with individual user's public key PKU is computed to conceal user's content.. If we use public key encryption with investigator's public key PKA and store the log database in CSP's custody then neither cloud employee nor investigator can violate the privacy of user alone. But if a malicious cloud employee provides a portion or full copy of the log database to an investigator then the user's privacy is in jeopardy. Asymmetric encryption with (individual) user's private key addresses this problem, though in turn leads to a potential concern of recovery of the log in the case of an investigation because privacy is provided with user's secret key. This problem can be solved using Shamir or Blackley's secret key sharing scheme and is discussed in the following "secret key sharing" section .After content concealment, the Sustainable Computing of cloud logs for cloud forensicsscheme will generate an encrypted log entry (ELE) and this ELE will be available to the user so that user can cross check if CSP is writing a correct log entry. Because searching through encrypted data is expensive, we keep some data in plain text format for filtering purposes
   ELE = <EPKu(ToIP || UserID), FromIP, TL>

c. After that, log chain (LC) is created in order to protect the integrity of the log and prevent potential manipulation. Sustainable Computing of cloud logs for cloud forensics creates LC using a hash function with current ELE and previous LC:
   LC = <H(ELE || LCPrevious) >

d. At this stage, the payload is ready to be stored in the database. Sustainable Computing of cloud logs for cloud forensics generates database log entry (DBLE) with ELE and LC and stores it in the log database.
   DBLE = < ELE, LC >

e. For each DBLE, the Sustainable Computing of cloud logs for cloud forensics scheme requires the generation of proof of past log (PPL) which is then made publicly available. Sustainable Computing of cloud logs for cloud forensics generates the PPL in a manner that is designed to minimize the usage of memory space. In Sustainable Computing of cloud logs for cloud forensics, we propose to generate PPL in a batch of the logs of a certain period or a certain amount of logs (e.g. n number of logs). At the end of each epoch, for each DBLE in a batch Sustainable Computing of cloud logs for cloud forensics concatenates each of the logs in a chain of logs in chronological order, derives the fingerprint, FP using Rabin's fingerprint [19]. Then the Sustainable Computing of cloud logs for cloud forensics scheme constructs an accumulator entry (AE) which is bloom filter membership information of the fingerprint FP. For each static IP, Sustainable Computing of cloud logs for cloud forensics retrieves accumulator entry (AE), epoch time TE, a signature using CSP's private key and concatenates AE, TE, with its signature, generating PPL.

f. Finally, the sustainable computing of cloud logs for cloud forensics scheme publishes PPL to the web via restful API or RSS feed:

   FP = FP (LC1 || LC2 || … … … || LCN) AE = BloomFilter(FP)

   PPL = <AE, TE, SingatureSKc(AE, TE)>
   After PPL is publicly available, it can be shared among multiple CSPs to build a trust model. This mitigates the potential for forgery so long as a single CSP is honest.

   After the publishing of PPL, the parser can verify their log, which is readily available in each epoch. If the user finds any discrepancy then the user can take steps accordingly. A backward check is to verify if logs accurately depict user activity and a forward check is to verify if the cloud server.
   Analyze and understand all the provided review comments thoroughly. Now make the required amendments in your paper. If you are not confident about any review comment, then don't forget to get clarity about that comment. And in some publishes the correct PPL for the corresponding logs. Sustainable Computing of cloud logs for cloud forensics

algorithms can be categorized into two major groups: One for Log Preservation (see Algorithm 1) and one for Proof Accumulation (see Algorithm 2). The Log Preservation algorithm can take log entries individually or in a batch and performs processing prior to storage in a log database. This algorithm encrypts for secrecy and generates hash digest for consistency. The Proof Accumulator algorithm performs daily processing of all log entries corresponding to an IP address to prepare and publish proof of past log (PPL). Pseudocode for both of these algorithms is provided below.

```
 LogPreservation( log entries LEs)
$for\ i \leftarrow 1$ to size( LEs )
```

```
        encrypted_log$_i$ = $encryp$( 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;
```

Algorithm 1. LogPreservation pseudocode for processing log entries

```
  ProofAccumulation( log entries LEs)
      $chronological\_concatinate\_LEs = LE_1 \| LE_2 \| \ldots \| LE_n$;
      $finger_{print} = FingerPrint( chronological\_concatinate\_LEs )$;
      $accumulator\_entry = BloomFilter( finger\_print )$;
      $signature = Signature( acuumulator\_entry, time)$;
      $Publish$ < $accumulator_{entry}, time, signature$ >;
e$nd$;
```

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

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 be used. 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).

Sustainable computing of cloud logs for cloud forensics proposes a bloom filter based PPL that computes membership information of (Rabin's) fingerprint [19] of chronologically ordered log chain of an epoch. It requires one single bloom filter for an entire batch of logs. The fingerprint has a total reflection of the entire log chain in a particular epoch. For example, if there are L logs in an epoch and log chains of these logs in chronological order are LC1, LC2, LC3, ..., LCL. Fingerprint FP of these log chains is,

FP = FingerPrint(LC1 || LC2 || LC3 || || LCL)
Then a bloom filter with membership information of FP is accumulator entry (AE) for this epoch of log. If the hash function for generating bloom filter information are: hash1, hash2, hash3, hashn.
v1 = hash1(FP) v2 = hash2(FP) v3 = hash3(FP)
... ...vn = hashn(FP)

A bloom filter with 1 at v1th, v2th, v3th, , vnth positions and 0 at rest of the positions represents accumulator entry, AE. Consequently, PPL for this epoch of log is a combination of accumulator entry (AE), epoch time (TE) and signature of this information by CSP.

PPL = < AE, TE, SignatureCSP(AE, TE) >

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.

Correct log:

In the Sustainable Computing of cloud logs for cloud forensics scheme, after symmetric encryption, encrypted log (ELE) gets available to the user via some secure channels. Then the user can decrypt and determine the authenticity of log entries. Though it should not be presumed that a user will have an understanding of low-level information within a log, the user may be able to verify high-level information such as sites visited, operations launched and so on. In this way, a user is able to establish log veracity.

PPL verification:

This verification is required for forwarding secrecy so that the CSP can't modify logs after publishing proof of log to the public. Any party with sufficient credentials, e.g. user or investigator or auditor, can verify PPL. Once the log is available, the user (or auditor) collects encrypted log (ELE) for a specific epoch, computes its PPL exactly in the same way the system generated the PPL and cross-matches with published PPL to verify.

III. CONCLUSION

In this paper, we proposed a secure logging scheme Sustainable Computing of cloud logs for cloud forensics for cloud computing with features that facilitate the preservation of user privacy and that mitigate the damaging effects of collusion among other parties. V Sustainable Computing of cloud logs for cloud forensics 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 Sustainable Computing of cloud logs for cloud forensics scheme, particularly in verification phase. Potential future extensions include the following:

1. Normally logs are low-level data and hard for the common user to understand what exactly those logs signify. Thus, we will explore leveraging big data techniques to facilitate user retrieval and visualization of information from log data. Standardization of log format is also an associated research area.

2. To ease searching, we kept some crucial and sensitive information in plaintext format. This makes them vulnerable to be exposure. Thus, designing secure and efficient searchable encryption would extend this work.

3. There is also the need for an online credibility system designed to develop trust and credibility of a cloud user so that the CSP can enable stricter auditing policies for low- trust users in comparison to high-trust users. Designing and implementing a prototype of the proposed scheme in collaboration with a real-world CSP, with the aim of evaluating its utility (e.g. performance and scalability) in a real-world environment.

REFERENCES

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] C. Esposito, A. Castiglione, F. Pop, and K.-K. R. Choo, "Challenges of Connecting Edge and Cloud Computing: A Security and Forensic Perspective," IEEE Cloud Computing, vol. 4, pp. 13-17, 2017.

[8] Z. Qi, C. Xiang, R. Ma, J. Li, H. Guan, and D. S. Wei, "ForenVisor: A tool for acquiring and preserving reliable data in cloud live forensics," IEEE Transactions on Cloud Computing, vol. 5, pp. 443-456,20

[9] C. Hooper, B. Martini, and K.-K. R. Choo, "Cloud computing and its implications for cybercrime investigations in Australia," Computer Law & Security Review, vol. 29, pp. 152-163, 2013.

[10] O. Osanaiye, K.-K. R. Choo, and M. Dlodlo, "Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework," Journal of Network and Computer Applications, vol. 67, pp. 147-165, 2016.

[11] N. H. Ab Rahman, W. B. Glisson, Y. Yang, and K.-K. R. Choo, "Forensic-by- design framework for cyber-physical cloud systems," IEEE. Cloud Computing, vol. 3, pp. 50-59, 2016.

[12] B. Martini and K.-K. R. Choo, "An integrated conceptual digital forensic framework for cloud computing," Digital Investigation, vol. 9, pp. 71-80, 2012.

[13] S. Khan, A. Gani, A. W. A. Wahab, M. A. Bagiwa, M. Shiraz, S. U. Khan, et al., "Cloud log forensics: foundations, state of the art, and future directions," ACM Computing Surveys (CSUR), vol. 49, p. 7, 2016.