Quest Journals
Journal of Software Engineering and Simulation
Volume 8 ~ Issue 7 (2022) pp: 77-84
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

Ensuring Forensic Integrity in Cloud Storage with Fingerprint

S. Ahalya
Department of Computer Science and Engineering
CSI Institute Of Technology, Thovalai

Dr. T. Saju Raj ME.,PH.d.
Department of Computer Science and Engineering
CSI Institute of Technology, Thovalai

Abstract- *To expedite the forensic investigation process in the cloud, excessive and yet volatile data need to be acquired, transmitted, and analyzed in a timely manner. A common assumption for most existing forensic systems is that credible data can always be collected from a cloud infrastructure, which might be susceptible to various exploits. The proposed system presents the design, implementation, and evaluation of fingerprint based secure systems, that enforces a trustworthy forensic data acquisition and transmission process in the cloud, whose computer platforms' integrity has been verified. To ensure the integrity of the data stored in the cloud, many data integrity auditing schemes have been proposed. In most, if not all, of the existing schemes, a user needs to employ his private key to generate the data authenticators for realizing the data integrity. Thus, the user has to possess a hardware token (e.g. USB token, smart card) to store his private key and memorize a password to activate this private key. If this hardware token is lost or this password is forgotten, most of the current data integrity auditing schemes would be unable to work. In order to overcome this problem, the proposed system implements a new paradigm called data integrity auditing without private key storage and design such a scheme. In this scheme, it use biometric data (e.g. iris scan, fingerprint) as the user's fuzzy private key to avoid using the hardware token. Meanwhile, the scheme can still effectively complete the data integrity auditing. It utilizes a linear sketch with coding and error correction processes to confirm the identity of the user to transfer the data. In addition, a new signature scheme is designed which not only supports blockless verifiability, but also is compatible with the linear sketch. The security proof and the performance analysis show that our proposed scheme achieves desirable security and efficiency.*

Index Terms: *Cloud storage, Data Integrity auditing, Data Security, Biometric Data*

R*eceived 06 July, 2022;* ***R****evised 18 July, 2022;* ***A****ccepted 20 July, 2022 © The author(s) 2022.*
P*ublished with open access at* ***www.questjournals.org***

I. INTRODUCTION

Throughout computer science history, numerous attempts have been made to disengage users from computer hardware needs, from time-sharing utilities envisioned in the 1960s, network computers of the 1990s, to the commercial grid systems of more recent years. This abstraction is steadily becoming a reality as a number of academic and business leaders in this field of science are spiraling towards cloud computing. Cloud computing is an innovative Information System (IS) architecture, visualized as what may be the future of computing, a driving force demanding from its audience to rethink their understanding of operating systems, client server architectures, and browsers. Cloud computing has leveraged users from hardware requirements, while reducing overall client side requirements and complexity.

As cloud computing is achieving increased popularity, concerns are being voiced about the security issues introduced through the adoption of this new model. The effectiveness and efficiency of traditional protection mechanisms are being reconsidered, as the characteristics of this innovative deployment model differ widely from those of traditional architectures.In this paper we attempt to demystify the unique security challenges introduced in a cloud environment and clarify issues from a security perspective. The notion of trust and security is investigated and specific security requirements are documented. With the rapid development of cloud computing technology and the popularization of mobile smart terminals, biometric authentication based on cloud computing has gradually penetrated every corner of people’s daily lives. A simple method of identifying people, biological characteristics, or biological behavior characteristics is receiving more attention for their convenience. However, while biometric authentication provides convenience to people's lives, it also brings about privacy concerns. In 2019, the VpnMentor team attacked the security platform BioStar 2, leaving the fingerprint records and facial recognition information of more than one million users at risk of leakage. Once the biometric data is leaked, it cannot be revoked or replaced, and the same biometrics will be used in different systems [5]. Therefore, it is important to establish appropriate security and privacy protection mechanisms to prevent fingerprint data leakage or abuse [6]. Many proposed biometric systems have been proposed using custom encryption and processing methods, leading to security risks. Considering the system’s security and efficiency requirements, the privacy protection of online fingerprint authentication is still a challenging task. In response to this problem, researchers have proposed a variety of solutions to solve it. In 1994, Bodo linked biometrics and cryptography for identification. To solve the contradiction between the cryptographic system's accuracy and the ambiguity of biometrics, the Fuzzy Vault scheme and the BioHashing scheme propose corresponding solutions based on the error correction code and Hamming distance matching methods. The BioHashing scheme has an obvious flaw. If an attacker pretends to be a legitimate user for identity authentication after obtaining the orthogonal matrix, there is a high probability that he can cheat the authentication system.

The Fuzzy Vault scheme has two serious security flaws:

(1) The data of the original feature point template can be obtained by cross-comparing multiple fuzzy vaults;

(2) Once the key is stolen, the attacker can replace part of the random hash value with another value, then can pass system verification by impersonating a legitimate user by verifying these values.

The proposed system analyzes the e-Finga scheme's security and designs a more secure and efficient secure e-finger scheme. First, it analyzes the three security risks of the literature:

- Deterministic encryption algorithms encrypt temporary fingerprints,
- All the users use the same secret parameters,
- The authorization data package does not authenticate the identity of the authorized object in the Encrypted Template Authorization phase.

The proposed system shows a temporary fingerprint attack to get user’s fingerprints in response to the above risks. The experiments show that it can get the fingerprint characteristics and some secret parameters in eavesdropping on the user’s temporary fingerprint ciphertext. Second, it proposes a secure online fingerprint authentication scheme Secure e-finger scheme. It use hard samples on the lattice cryptography to encrypt temporary fingerprints in the Authentication Query Generation phase. This approach has two benefits: on the one hand, the temporary fingerprint feature distribution information can be hidden; on the other hand, in the Fingerprint Matching phase, the Euclidean distance between the template and the temporary fingerprint feature can be calculated without affecting the matching result. Secondly, different secret parameters are generated for different users to prevent collusion between corrupt users and servers who want to obtain secret parameters and legitimate user's template information. Finally, in the Encrypted Template Authorization phase, the authorization object information is added to the authorization data package. The method can prevent attackers from pretending to be authorized objects and pretending to be the user’s fingerprint template data, then protecting the template data from abuse. Finally, we have compared the proposed scheme and the e-Finga scheme from the aspects of the client running time, computational complexity, space cost, and communication cost, and the analysis showed that the security of the scheme could be improved without affecting the system operating efficiency. This paper analyzes and discusses the literature's security issues and introduces the samples of LWE (Learning with errors) to encrypt the temporary fingerprint feature. It proposes a new security privacy protection online fingerprint authentication scheme based on cloud computing, called Secure e-finger, based on BGN homomorphic encryption scheme. In the scheme, the user’s fingerprint can be outsourced to different servers with user’s authorization, and it can provide a secure, accurate authentication service without the leakage of fingerprint information. The computation and the communication cost of our scheme are low.

Four deployment models have been identified for cloud architecture solutions, described below:

- Private cloud. The cloud infrastructure is operated for a private organization. It may be managed by the organization or a third party, and may exist on premise or off premise.
- Community cloud. The cloud infrastructure is shared by several organizations and supports a specific community that has communal concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party, and may exist on premise or off premise.
- Public cloud. The cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services.

● Hybrid cloud. The cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology, that enables data and application portability (e.g., cloud bursting for load-balancing between clouds).

II. RELATED WORK

This related work section will be followed by the to expedite the forensic investigation process in the cloud, excessive and yet volatile data need to be acquired, transmitted, and analyzed methods used in the previous papers. A common assumption for most existing forensic systems is that credible data can always be collected from a cloud infrastructure, which might be susceptible to various exploits. [1] presents the design, implementation, and evaluation of LiveForen, a system that enforces a trustworthy forensic data acquisition and transmission process in the cloud, whose computer platforms' integrity has been verified. To fulfill this objective, it uses two secure protocols that verify the fingerprints of the computer platforms, as well as the attributes of the human agents, by taking advantage of the trusted platform module and the attribute-based encryption. To transmit forensic data as a data stream and verify its integrity at the same time, a unique fragile watermark is embedded into the data stream without altering the data itself. The watermark allows not only the data integrity to be verified but also any malicious data manipulation to be localized, with minimum communication overhead. [2] which stores virtual machines' logs and provides access to forensic investigators ensuring the confidentiality of the cloud users. Additionally, SeclaaS preserves proofs of past log and thus protects the integrity of the logs from dishonest investigators or cloud providers.

[9] propose a novel fragile watermarking algorithm which verifies the integrity of streaming data at the application layer. The data are divided into groups based on synchronization points, so each group can be synchronized and any modifications made to one group only affect up to two groups. A unique watermark is embedded directly into each group to save communications bandwidth. The embedded watermark can detect as well as locate any modifications made to a data stream. To ensure the completeness of the data stream, watermarks are chained across groups so that no matter how many data are deleted, these deletions can be correctly detected. Security analysis and experimental results show that the proposed scheme can efficiently detect and locate modifications and ensure the completeness of data streams. [4] supports an extensible set of forensic objectives, including the future addition of other data preservation, discovery, real-time monitoring, metrics, auditing, and acquisition capabilities. Today, cloud computing environments lack trustworthy capabilities for the cloud customer or forensic investigator to perform incident response and forensic investigation. Consequently, customers of public cloud services are at the mercy of their cloud provider to assist in an investigation. [4] based on the paper, for instance, the evidence can be tampered with when it is under the control of the privileged but malicious human agent who owns the secret key. The encrypted data can still be breached during the transmission phase. Moreover, live forensics requires evidence to be resumed when it is needed. Finally, because the physical TPM could become the computational bottleneck, the number of computationally expensive TPM-related operations should be limited. As companies turn to cloud services to reduce costs compared to their internally managed Information Technology (IT) systems, a fundamental shift is occurring in the way IT and computing services are delivered and purchased. With this shift towards utility computing , new trust relationships arise that force the parties to reconsider the way we handle and manage information in the cloud. [10]reduces the overhead of key management and improves the performance of the distributed protocols employed. To demonstrate that Excalibur is practical, we incorporated it in the Eucalyptus open-source cloud platform. Policy-sealed data can provide greater confidence to Eucalyptus customers that their data is not being mismanaged.

III. SYSTEM DESIGN

As shown in fig 3.1 The proposed system employs biometric data as a fuzzy private key to perform data integrity auditing, and proposes a new paradigm called data integrity auditing without private key storage. In such a scheme, a user utilizes biometric data as his fuzzy private key for confirming his identity. A practical data integrity auditing scheme without private key storage for secure cloud storage. In this scheme, two fuzzy private keys (biometric data) are extracted from the user in the phase of registration and the phase of signature generation. In order to confirm the user's identity, it compares these two fuzzy private keys by removing the "noise" from two sketches. If the two biometric data are sufficiently close, itcan confirm that they are extracted from the same user; otherwise, from different users.

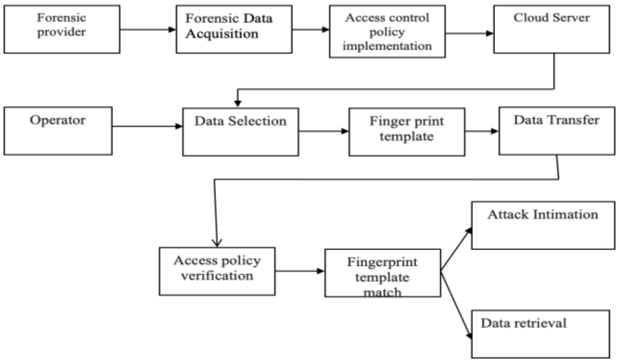

Fig 3.1 System Design

IV. GROUNDWORKS

In this section it gives a brief explanation about the groundwork done for this paper. The groundwork means a preprocessing or preliminary work that is done for the project.

[1] To expedite the forensic investigation process in the cloud, excessive and yet volatile data need to be acquired, transmitted, and analyzed in a timely manner. To achieve this it uses two secure protocols that verify the fingerprints of the computer platforms, as well as the attributes of the human agents, by taking advantage of the trusted platform module and the attribute-based encryption. [5] Safety and reliability are important in the cloud computing environment. This is especially true today as distributed denial-of-service (DDoS) attacks constitute one of the largest threats faced by Internet users and cloud computing services. DDoS attacks target the resources of these services, lowering their ability to provide optimum usage of the network infrastructure. Due to the nature of cloud computing, the methodologies for preventing or stopping DDoS attacks are quite different compared to those used in traditional networks. In this paper, we investigate the effect of DDoS attacks on cloud resources and recommend practical defense mechanisms against different types of DDoS attacks in the cloud environment.

● Setup: This algorithm takes as input a fuzzy key setting FKS and a security parameter k. It outputs the public parameter pp'.

● KeyGen (pp', y): This algorithm takes as input the public parameter pp' and the biometric data y€ Rn. It generates pk as his public key, which includes a sketch c and a verification key vk.

● Sign Gen (y', F) This algorithm takes as input the biometric data y € Rn and the file F. It outputs a signature α which includes the verification key vk', the sketch c' and the set of authenticators α.

V. PROPOSED MODEL

The system model involves three types of entities: the user, the cloud, and the TPA. The cloud provides enormous data storage space to the user. The user has a large number of files to be uploaded to the cloud. The TPA is a public verifier who is delegated by the user to verify the integrity of the data stored in the cloud. In the phase of user registration, the biometric data (e.g. fingerprint) is extracted from the user who wants to use the cloud storage service. When a data owner would like to upload data to the cloud, he firstly extracts biometric data as his fuzzy private key and randomly generates a signing key. Then, this data owner computes authenticators for data blocks with his signing key. Finally, he uploads these data blocks along with the authenticator set to the cloud and deletes these messages from the local storage. In the phase of data integrity auditing, the TPA verifies whether the cloud truly keeps the user's data intact or not by executing the challenge-response protocol with the cloud. A data integrity auditing scheme without private key storage consists of the following five algorithms: Setup, KeyGen, Sign-Gen, Proof Gen and Proof Verify. Specifically, these algorithms are described as follows:

Fig 5.1 Setup model.

This Fig 5.1 explains about the Setup model, in that model data owners provide fingerprints. From the finger print it will take Finger Key Settings and Secret Key(sk) as a input and generates Public parameter (pp)

The cloud provides enormous data storage space to the user. The user has a large number of files to be uploaded to the cloud. The TPA is a public verifier who is delegated by the user to verify the integrity of the data stored in the cloud. In the phase of user registration, the biometric data (e.g. fingerprint) is extracted from the user who wants to use the cloud storage service. When a data owner would like to upload data to the cloud, he firstly extracts biometric data as his fuzzy private key and randomly generates a signing key. Then, this data owner computes authenticators for data blocks with his signing key. Finally, he uploads these data blocks along with the authenticator set to the cloud and deletes these messages from the local storage.

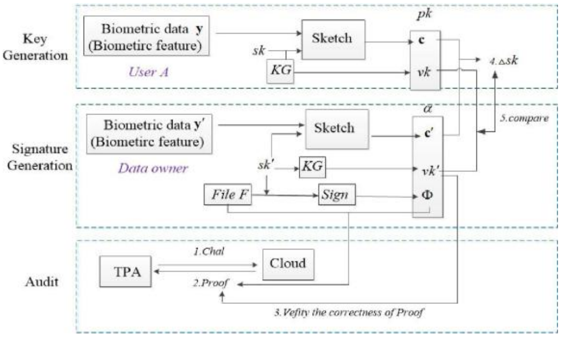

Fig 5.2 Signature Generation

Fig 5.2 explains the concept of fuzzy signature uses biometric data as a private key, such as iris scan and fingerprint, to generate the signature. The biometric data y is a feature vector which is defined as an n-dimensional vector. In a fuzzy signature scheme, the key generation algorithm KeyGen takes the biometric data y as input, and generates a verification key vk. The signature generation algorithm SigGen takes as input the biometric data y’ and a data block mi, and generates the signature of mi. The verification algorithm Verify takes as input the verification key vk, the data block mi and the signature and verifies whether the signature is valid or not. If the biometric data y’ is sufficiently close to the biometric data y, it means that y0 and y are extracted from the same user. Thus, the signature is valid; otherwise, it is invalid.

- Setup: This algorithm takes as input a fuzzy key setting FKS and a security parameter k. It outputs the public parameter pp’.
- KeyGen (pp’, y): This algorithm takes as input the public parameter pp’ and the biometric data y€ Rn. It generates pk as his public key, which includes a sketch c and a verification key vk.
- Sign Gen (y’, F) This algorithm takes as input the biometric data y € Rn and the file F. It outputs a signature α which includes the verification key vk’, the sketch c’ and the set of authenticators α.
- Proof Gen (F,chal) This algorithm takes as input the file F, the corresponding authenticator set and the auditing challenge chal. It outputs an auditing proof P that proves the cloud indeed keeps this file.
- Proof Verify (pk, chal, P, vk’, c’) This algorithm takes as input the user’s public key pk, the auditing challenge chal, the auditing proof P, the verification key vk’ and the sketch c’. The TPA verifies the correctness of proof P.

Our proposed scheme implements the fuzzy private key to realize data integrity auditing without storing private key. The first practical data integrity auditing scheme without private key storage for secure cloud storage. In the proposed scheme, it utilizes biometric data (e.g. fingerprint, iris scan) as a user's fuzzy private key to achieve data integrity auditing without private key storage.

1. Setup

This algorithm takes as input a fuzzy key setting FKS and a security parameter k. It outputs the public parameter pp’.

Algorithm 1: Setup (Initialization)

Input: (FKS,Sk): A security parameter Sk and a Fuzzy key System.
Output: pp’ a public parameter.

2. ***KeyGen:***

This algorithm takes as input the public parameter pp’ and the biometric data y€ Rn. It generates pk as his public key, which includes a sketch c and a verification key vk.

Algorithm 2: Key Generation (KeyGen)

Input: (pp’, y): The public parameter pp’ and the biometric data y€ Rn
Output: pk a public key ← c, vk.

c → sketch of biometric input
vk → verification key

Algorithm 3: Signature Generation (SignGen)

1 Biometric data (y) as a private key.
y = (y1…………yn) € Rn.
2 y’ and mi as input (y’ - Biometric Data and data block) → Signature for mi
3 Randomly x € Zw as his private key sk
4 Generates a sketch c used to code and correct the error of biometric data.

5 c’ = $(\mathrm{CRT}^{-1}{}_{w}(x') + F_{w}(y'))\mathrm{mod}\ w$.
6 Biometric data y0 = (y01…………yn) € Rn
7 Choose x’ € Zw as sk’ → Vk

2. Signature Generation

The concept of fuzzy signature uses biometric data as a private key, such as iris scan and fingerprint, to generate the signature. The biometric data y is a feature vector which is defined as an n-dimensional vector. In a fuzzy signature scheme, the key generation algorithm KeyGen takes the biometric data y as input, and generates a verification key vk. The signature generation algorithm SigGen takes as input the biometric data y’ and a data block mi, and generates the signature of mi. The verification algorithm Verify takes as input the verification key vk, the data block mi and the signature and verifies whether the signature is valid or not. If the biometric data y’ is sufficiently close to the biometric data y, it means that y0 and y are extracted from the same user. Thus, the signature is valid; otherwise, it is invalid.

The data integrity auditing scheme consists of the following three procedures: Key Generation, Signature Generation and Audit. Key Generation. It includes Setup and Key Gen algorithms. Firstly, the public global parameter pp’ is generated in the Setup algorithm. In the Key Gen algorithm, the user A, who wants to store his data in the cloud, extracts biometric data y in the phase of registration. Next, this user randomly generates a key pair (sk, vk). Finally, this user generates a sketch c of private key sk using y, which is used to code and correct the error of biometric data. The public key pk of our proposed scheme includes (vk, c). Signature Generation. It consists of the Sign Gen algorithm. The data owner generates the signature of the file F, and uploads this file along with its signature to the cloud. KeyGen (pp’, y) In the phase of registration, biometric data y = (y1…. yn) € Rn is extracted from the user A who wants to use the cloud storage service. The user A randomly selects x € Zw as his private key sk, and generates his verification key vk. Next, the user A generates a sketch c used to code and correct the error of biometric data as follows:

$$c' = (\mathrm{CRT}^{-1}{}_{w}(x') + F_{w}(y'))\bmod w.$$

The data owner divides his outsourcing file F into s blocks. In the phase of signature generation, biometric data y0 = (y01…………yn) € Rn is extracted from the data owner. The data owner randomly chooses x’ € Zw as his signing key sk’, and calculates his verification key vk’. Next, the data owner generates the authenticator of the data block mi with his signing key x’. Finally, the data owner sends the file to the cloud, and deletes the file F and its corresponding signature from the local storage.

4. Signature Verification

Specifically, the data owner randomly generates a signing key sk’ and its corresponding verification key vk’, where sk’ is used to generate the sketch and the authenticators. Then the data owner generates a sketch c’ of signing key sk’ using the biometric data y’ extracted from him. He generates a data authenticator set for file F with signing key sk’. The signature of file F is (ϕ, vk’, c’). The data owner sends files to the cloud, and deletes them from the local storage. The Proof Gen algorithm and Proof Verify algorithm are executed in this phase. In the Proof Gen algorithm, the TPA sends an auditing challenge chal to the cloud. Upon receiving the chal, the cloud returns an auditing proof P to the TPA. In the Proof Verify algorithm, the TPA firstly checks the correctness of the proof P using the verification key vk’. And then, in order to confirm the identity of the data owner, the TPA recovers sk from c and c’ by using the technique of coding and error correction. Finally, the TPA verifies whether the difference between vk and vk’ truly corresponds to sk. If it does, the data owner is the user A; otherwise, he is not.

5. Auditing

The Proof Gen algorithm and Proof Verify algorithm are executed in this phase. In the Proof Gen algorithm, the TPA sends an auditing challenge chal to the cloud. Upon receiving the chal, the cloud returns an auditing proof P to the TPA. In the Proof Verify algorithm, the TPA firstly checks the correctness of the proof P using the verification key vk’. And then, in order to confirm the identity of the data owner, the TPA recovers sk from c and c’ by using the technique of coding and error correction. Finally, the TPA verifies whether the difference between vk and vk’ truly corresponds to sk. If it does, the data owner is the user A; otherwise, he is not. To verify the integrity of cloud data, the TPA randomly selects a c-element subset I of set [1; s], and generates a random for each i € I. The TPA ends the auditing challenge chal = fi; to the cloud. Upon receiving the auditing challenge, the cloud computes a linear combination of data blocks and an aggregated data authenticator. Then, the cloud returns an auditing proof to the TPA. The TPA checks whether the following equation holds.

$$E(\sigma,g) = e\ (\Pi_{2\in I}\, H(name\|i)^{\beta i}\ .u^{\mu}{}_{,}vk')$$

If the equation does not hold, then the data stored in the cloud is corrupted; otherwise, the TPA does as follows: The TPA recovers Δx from c and c by computing Pw(c – c’) = Δx and CRTw(Δx) = Δx, which can confirm the identity of the user. Then, he verifies whether the following equation holds or not: (vk)zΔx = vk’. If the equation holds, then the data stored in the cloud is intact; otherwise, it is corrupted.

VI. CONCLUSION

The proposed system implements the fuzzy private key to realize data integrity auditing without storing private key. The first practical data integrity auditing scheme without private key storage for secure cloud storage. In the proposed scheme, it utilizes biometric data (e.g. fingerprint, iris scan) as a user's fuzzy private key to achieve data integrity auditing without private key storage. In addition, it designs a signature scheme supporting blockless verifiability and the compatibility with the linear sketch. The formal analysis shows that the proposed scheme is provably secure and efficient.

The proposed system can be further enhanced with advanced encryption using fuzzy logic and secret sharing scheme.

REFERENCE

[1]. Liu, Anyi, Huirong Fu, Yuan Hong, Jigang Liu, and Yingjiu Li. "$ LiveForen $: Ensuring Live Forensic Integrity in the Cloud." *IEEE Transactions on Information Forensics and Security* 14, no. 10 (2019): 2749-2764.

[2]. Zawoad, Shams, Amit Kumar Dutta, and Ragib Hasan. "SecLaaS: secure logging-as-a-service for cloud forensics." In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pp. 219-230. 2013.

[3]. Bethencourt, John, Amit Sahai, and Brent Waters. "Ciphertext-policy attribute-based encryption." In *2007 IEEE symposium on security and privacy (SP'07)*, pp. 321-334. IEEE, 2007.

[4]. Krautheim, F. John, Dhananjay S. Phatak, and Alan T. Sherman. "Introducing the trusted virtual environment module: a new mechanism for rooting trust in cloud computing." In *International conference on trust and trustworthy computing*, pp. 211-227. Springer, Berlin, Heidelberg, 2010.

[5]. Darwish, Marwan, Abdelkader Ouda, and Luiz Fernando Capretz. "Cloud-based DDoS attacks and defenses." In *International Conference on Information Society (i-Society 2013)*, pp. 67-71. IEEE, 2013.

[6]. Kotla, Ramakrishna, Tom Rodeheffer, Indrajit Roy, Patrick Stuedi, and Benjamin Wester. "Pasture: Secure offline data access using commodity trusted hardware." In *10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12)*, pp. 321-334. 2012.

[7]. Chen, Chen, Himanshu Raj, Stefan Saroiu, and Alec Wolman. "{cTPM}: A Cloud {TPM} for {Cross-Device} Trusted Applications." In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*. 2014.

[8]. Dykstra, Josiah, and A. T. Sherman. "Design and implementation of FROST: Digital forensic tools for the OpenStack cloud computing platform. Digit. Invest." *S87–S95* (2013).

[9]. Guo, Huiping, Yingjiu Li, and Sushil Jajodia. "Chaining watermarks for detecting malicious modifications to streaming data." *Information Sciences* 177, no. 1 (2007): 281-298.

[10]. Santos, Nuno, Rodrigo Rodrigues, Krishna P. Gummadi, and Stefan Saroiu. "{Policy-Sealed} Data: A New Abstraction for Building Trusted Cloud Services." In *21st USENIX Security Symposium (USENIX Security 12)*, pp. 175-188. 2012.

[11]. King, Samuel T., and Peter M. Chen. "SubVirt: Implementing malware with virtual machines." In *2006 IEEE Symposium on Security and Privacy (S&P'06)*, pp. 14-pp. IEEE, 2006.

[12]. V. Ciancaglini, M. Balduzzi, R. McArdle, and M. Rösler. (2015). Below the Surface: Exploring the Deep Web. [Online]. Available: https://www. trendmicro.com/cloud-content/us/pdfs/security-intelligence/whitepapers/ wp_below_the_surface.pdf

[13]. Symantec. Avoiding the Hidden Costs of the Cloud. Accessed: Aug. 1, 2018. [Online]. Available:https://www.symantec.com/content/ en/us/about/media/pdfs/b-state-of-cloud-global-results-2013.en-us.pdf [4] R. Samani and F. Paget. (2013). Cybercrime Exposed: Cybercrime-as-a-Service.[Online].Available:http://www.mcafee.com/jp/resources/whitepapers/wp-cybercrime-exposed.pdf

[14]. D. Goodin. ZeusBot Found Using Amazon’s EC2 as C&C Server. [Online]. Available: ttp://www.theregister.co.uk/2009/12/09/amazon_ec2_bot_control_channel/

[15]. M. D. Ryan, “Cloud computing security: The scientific challenge, and a survey of solutions,” J. Syst. Softw., vol. 86, no. 9, pp. 2263–2268, Sep. 2013.

[16]. D. Zissis and D. Lekkas, “Addressing cloud computing security issues,” Future Gener. Comput. Syst., vol. 28, no. 3, pp. 583–592, Mar. 2012.

[17]. NIST Information Technology Laboratory. (2013). NIST Cloud Computing Forensic Science Challenges. [Online]. Available: http://csrc.nist. gov/publications/drafts/nistir-8006/draft_nistir_8006.pdf

[18]. U. S. Congress. (2018). Cloud Act. [Online]. Available:https://www.congress.gov/bill/115th-congress/house-bill/4943

[19]. (2017). Forensics at the OJ Simpson Trial. [Online].Available:https://www.crimemuseum.org/crime-library/famous-murders/forensicinvestigation- of-the-oj-simpson-trial/

[20]. Regional Computer Forensics Laboratory. (2016). RCFL Annual Reports FY2016. [Online]. Available: https://www.rcfl.gov/downloads

[21]. Y. Fu and Z. Lin, “Space traveling across VM: Automatically bridging the semantic gap in virtual machine introspection via online kernel data redirection,” in Proc. IEEE Symp. Secur. Privacy, May 2012, pp. 586–600.

[22]. J. Dykstra and A. T. Sherman, “Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques,” Digit. Invest., vol. 9, pp. 90–98, Aug. 2012.

[23]. S. Zawoad and R. Hasan, “I have the proof: Providing proofs of past data possession in cloud forensics,” in Proc. Int. Conf. Cyber Security., Dec. 2012, pp. 75–82.