

# Blockchain Solutions for Forensic Evidence Preservation in IoT Environments

Sotirios Brotsis\*, Nicholas Kolokotronis\*, Konstantinos Limniotis\*, Stavros Shiaeles†, Dimitris Kavallieros‡, Emanuele Bellini§, and Clément Pavué¶

\*University of Peloponnese, Greece. Email: brotsis@uop.gr, nkolok@uop.gr, klimn@uop.gr

†Plymouth University, UK. Email: stavros.shiaeles@plymouth.ac.uk

‡Center for Security Studies, Greece. Email: d.kavallieros@kemea-research.gr

§Mathema s.r.l., Italy; Khalifa University, UAE. Email: emanuele.bellini@mathema.com

¶Scorechain S.A., Luxembourg. Email: clement.pavue@scorechain.com

**Abstract**—The technological evolution brought by the *Internet of things* (IoT) comes with new forms of cyber-attacks exploiting the complexity and heterogeneity of IoT networks, as well as, the existence of many vulnerabilities in IoT devices. The detection of compromised devices, as well as the collection and preservation of evidence regarding alleged malicious behavior in IoT networks, emerge as areas of high priority. This paper presents a blockchain-based solution, which is designed for the smart home domain, dealing with the collection and preservation of digital forensic evidence. The system utilizes a private forensic evidence database, where the captured evidence is stored, along with a permissioned blockchain that allows providing security services like integrity, authentication, and non-repudiation, so that the evidence can be used in a court of law. The blockchain stores evidences' metadata, which are critical for providing the aforementioned services, and interacts via smart contracts with the different entities involved in an investigation process, including Internet service providers, law enforcement agencies and prosecutors. A high-level architecture of the blockchain-based solution is presented that allows tackling the unique challenges posed by the need for digitally handling forensic evidence collected from IoT networks.

**Index Terms**—Blockchain, Cyber-security, Forensic evidence, Intrusion detection, Internet of things.

## I. INTRODUCTION

The *Internet of things* (IoT) ecosystem is comprised of a vast number of interconnected devices that collect, process, generate, and share huge amounts of (possibly sensitive and critical) information [1]. To a large extent, these devices are highly resource-constrained, like sensors and legacy embedded systems, therefore devoting most of their computational power and storage/memory capacity to delivering their core functionality. Strong security controls that are typically found in today's personal computers cannot be adopted, since they are more resource-demanding, hence leading to the usage of lightweight and often insecure protection mechanisms (if any) for the data stored or transmitted. This fact, if combined with

the complexity and heterogeneity of IoT networks that make the design and provisioning of security solutions a challenging task [2], allows cyber-attackers to easily compromise IoT devices and use them as the means for launching other advanced attacks, such as the *distributed denial of service* (DDoS) attack against Dyn that was attributed to Mirai malware [3].

The collection of forensic evidence from the attacked IoT devices and networks, along with the storage, preservation and analysis of the evidence constitute major challenges [4], primarily due to the fact that IoT devices are designed to work autonomously and in many cases, there is no reliable method to assemble residual evidence [5]. The utilization of *intrusion detection systems* (IDS) in the collection process is important towards identifying cyber-criminals and preventing future occurrences of attacks [6]. In an IoT environment, the identification of a crime scene's boundaries and its preservation are quite hard to accomplish while interactions continuously occur at real-time. Since the majority of IoT devices are sensors and monitors that record user's personal information, privacy is an important issue to consider in a digital forensics investigation.

This paper aims at addressing the challenges in the forensic evidence collection, preservation and investigation process, for IoT environments in the smart home domain, by exploiting the advanced intrusion detection and *distributed ledger technology* (DLT) solutions that are being developed in the context of the Cyber-Trust project (<https://cyber-trust.eu/>). More precisely, a number of mechanisms installed at a smart home's gateway, like profiling, monitoring, and anomaly detection, allow to monitor the state and behavior of IoT devices, significantly enhancing the detection of known threats and zero-day vulnerabilities, as well as to immediately collect forensic evidence for detected malicious interactions. The collected data are stored at the *evidence database* (evDB), hosted by the *Internet service provider* (ISP), along with the metadata needed in order to allow the correlation and further investigation of an attack's generated events. The metadata are published on a blockchain, which is maintained by the ISPs, maintaining the chronological ordering of attacks' evidences at a global scale, thus providing the means to *law enforcement agencies* (LEA) to effectively trace back an attack to its source. The proposed solution, referred to as *Cyber-Trust blockchain* (CTB), allows



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 786698. The work reflects only the authors' view and the Agency is not responsible for any use that may be made of the information it contains.

the entities involved in the investigation process, such as LEAs and prosecutors, to access and handle the digital evidence, therefore realizing the *chain-of-custody* (CoC) by recording and preserving the chronological history of handling the digital evidence. The CTB solution relies on HyperLedger Fabric and constitutes a permissioned blockchain in order to meet privacy requirements.

The remainder of the paper is structured as follows. Section II presents the current state-of-the-art and related work, while the forensic evidence collection process is described in Section III. Section IV provides the architecture of the CTB solution whereas concluding remarks are given in Section V.

## II. BACKGROUND AND RELATED WORK

This section presents the current state-of-the-art in the areas of intrusion detection and forensic evidence collection for IoT environments, along with the blockchain solutions that have been proposed.

### A. IoT intrusion detection

Intrusion detection systems typically utilize signature-based and anomaly-based techniques for identifying possible threats in a network, where the latter relies on the monitoring of a network's devices for any abnormal behavioral patterns [6]. In order to detect compromised IoT devices, the framework that is proposed by Nguyen, *et al.* autonomously identifies anomalies in an IoT network [7]; this is achieved by employing a self-learning framework to classify devices according to their types and generate normal profiles that are subsequently used for the detection of deviations. A privacy-preserving architecture, called Siotome, was proposed in [8] to provide security in smart home environments against distributed network attacks by malicious IoT devices; the system is able to monitor, detect and analyze IoT-based threats, but also to provide an effective defense framework by utilizing machine learning methods to establish optimal operational configurations.

Smart phones are a particular type of devices within a smart home environment, since they are mostly used for personal and sensitive tasks, thus becoming extremely beneficial and easy targets for adversaries. Smart phones, which are vulnerable to attacks (e.g. viruses, Trojans, worms, etc.) common in personal computers, but lack the capabilities to execute highly advanced algorithms for detecting malicious activities. Due to this fact, IDS solutions that are often proposed to regularly perform in-depth analysis and observe any misbehavior are either cloud-based [9], or are performed remotely at a central server [10], allowing optimal actions to be taken for thwarting the attack in both architectures.

### B. IoT forensics

The wide adoption of smart devices, which can provide a wealth of forensic evidence on malicious activities during an investigation process, necessitated the advancement of tools and techniques for collecting residual evidences. A forensics edge management system for the smart home environment was introduced in [11] to gather digital evidence and deal with any

security issues; it provides intelligence, flexibility, automated detection, and advanced data logging capabilities. The authors in [12] proposed a forensic investigation architecture to ensure the collection, preservation and storage of digital evidences, while they validated their approach in a real-world smart home environment. Focusing on a smart home's IoT devices, a physical analyzer called *universal forensic extraction device* has been proposed for conducting forensic investigation on smart phones [13], which has been tested on Android devices. A comparative analysis of digital forensics tools for Android smart phones was carried out in [14], where it was illustrated that the choice of the tool to be used plays a crucial role in the quality of the forensic evidence that is extracted from the devices. In contrast to [14], a method for acquiring forensic evidence from Android smart phones without using specialized commercial forensics tools, i.e. by only relying on open source software, was proposed in [15]. In all the above works, it was shown that the collection of information from smart phones so that it can be used as evidence in a court of law still remains a challenging task.

### C. Blockchain solutions

Blockchain solutions have recently been proposed for both intrusion detection and forensic evidence applications, since in both cases blockchain can solve issues pertaining to trust, integrity, transparency, accountability, and secure data sharing. Addressing the issue of trust management, Alexopoulos, *et al.* [16] applied blockchain in collaborative intrusion detection networks to deal with insider threats but also enhance the security of the information shared among the participating IDS nodes. More precisely, the authors proposed to store the generated (raw) alerts of the network as transactions in a permissioned blockchain. Meng, *et al.* [17] in addition to the dimension of trust between the IDS nodes, refer to issues that pertain to privacy when collaborating nodes belong to different trust domains, as shared data may have sensitive information linked to individuals or organizations, e.g., IP addresses and packet payloads. Methods for exchanging encrypted content, or only hashed data rather than raw, are considered.

In forensic investigations, it is important that the evidence is not modified while passing from one entity to another. The blockchain can be used in order to certify the authenticity and legitimacy of the procedures used to gather, store and transfer digital evidence, as well as, to provide a comprehensive view of all the interactions in the CoC. In a blockchain-based CoC, it is crucial to assure that members, having read/write access to the distributed ledger, are authenticated and the evidences are verified via a consensus algorithm. Towards that direction, Lone, *et al.* propose a private blockchain that can be used in digital forensics to ensure the integrity of evidences [18]; the authors also aim at recording the actions taken by each entity when interacting with the evidence. On the other hand, Probe-IoT uses a blockchain to discover criminal events, which can be used as evidence, by collecting interactions between IoT devices and verify their authenticity [19].

### III. FORENSIC EVIDENCE COLLECTION ARCHITECTURE

The primary goal of Cyber-Trust in the smart home domain, or in general in *small office/home office* (SOHO) network, is to accurately detect the local network's compromised and/or infected IoT devices to apply the appropriate countermeasures, e.g. to isolate the devices from the rest of the network and to proceed with the application of proper remediation measures. The intrusion detection mechanisms that are being employed are operating both at the device- and network-level to facilitate the collection and subsequent correlation of forensic evidence from various independent sources.

To combat cyber-attacks and assist the evidence collection, IoT devices' critical information is recorded on the blockchain so that it can be later queried when e.g. a verification of proper functioning is needed, or parts of the system's software have to be updated or patched reliably. This implies that properties, like a device's firmware, configuration files, etc. are registered into the Cyber-Trust blockchain, at the beginning of system's operation, and verified if needed against a history of previously valid states, in order to ensure that they have not been tampered with. This approach fits well within the practices of software distributors that publish hashes of software binaries to allow verifying their authenticity.

#### A. Adversarial model

The adversary is a typical IoT malware botnet that actively scans for vulnerable Linux-based IoT devices in the SOHO network, like smart watches, home surveillance systems, smart phones, etc., and infects the discovered vulnerable devices by uploading and executing malware code of an unknown bot on the compromised devices; once infected, the IoT devices may take a variety of malicious actions. Typically, the phases of a botnet, prior to performing attacks in a coordinated manner, are the following.

1) *Propagation*: If having been infected with malware, a smart home's device updates its configuration and downloads further exploits. The bot replicates itself in the SOHO network using telnet/FTP/SSH default credentials and attacks nearby devices with firmware vulnerabilities.

2) *Rallying*: The bot contacts a *command & control* (C&C) server, queries for instructions, and also downloads the main configuration files. The bot and the bot-master share a seeded pseudorandom generator that computes the domain names.

3) *Interaction*: Bot-masters use a pull approach, in which the bot should initiate contact with the C&C server, and then poll for updates regularly. Obfuscation techniques are used, by hiding communications in regular web traffic, hence allowing perimeter controls to be bypassed.

As seen from above, the bot is listening for commands via the HTTP and HTTPS protocols (utilizing ports 80 and 443) and is assumed to execute three types of attacks, namely *man-in-the-middle* (MiTM), DDoS, and spamming.

#### B. Architectural elements

In the sequel, we describe the high-level design of the smart home environment's security elements, as illustrated in Fig. 1.

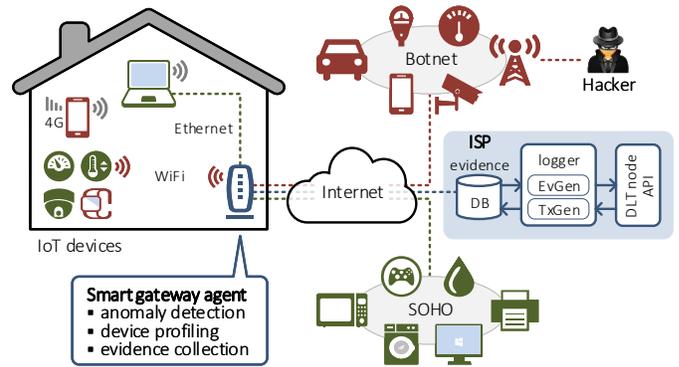


Fig. 1. An overview of Cyber-Trust's forensic evidence collection process; it is assumed that the red-colored devices in the smart home have been attacked and this is detected by the SGA that collects the evidence.

The *smart gateway agent* (SGA) is the core component that is responsible for the smart home's network security by utilizing advanced intrusion detection methods, monitoring its health status and profiling the IoT devices' behavior, as well as the collection of network information including forensic evidence; the SGA is the main link with the core platform components running at the ISP layer (only those relevant to the evidence collection process are depicted in Fig. 1). When a new device is registered, the SGA performs *device fingerprinting* in order to extract the device's behavioral patterns based on network flows — assuming that the device is initially in a clean state. In addition, the SGA actively monitors the communication of connected devices to detect abnormal behavior by employing a lightweight IDS which transfers any suspicious traffic to the platform's back-end for *deep packet inspection* (DPI). Further to the above, the SGA uses *manufacturer's usage description* (MUD) to deliver device-focused network profiling to support accurate feature-set extraction for the anomaly detection.

More capable IoT devices, e.g. smart phones, have a *smart device agent* (SDA) installed that allows the direct acquisition of information (including evidence) from end-user IoT devices. The SDA operates in a more restrictive manner as it is mainly responsible for monitoring the device's usage, critical files and security — firmware integrity, patching status, vulnerabilities. Information on run-time processes and the hardware resources used is regularly synchronized with the Cyber-Trust platform's back-end, and more precisely the *profiling service* (PS).

#### C. Evidence collection

When suspicious network traffic and (resp. device activity) is detected by the SGA (resp. SDA), the necessary evidence is collected and sent to the ISP so as to be stored to the evDB. The evidence is comprised of IP packets (amongst other data) in the case of network attacks, whereas for device-level attacks it might include the entire device's image. At a minimum, the whole process is designed to achieve the following objectives: (a) ensure the confidentiality and integrity of forensic evidence during transmission and storage; (b) ensure that the evidence is collected from and destined to secure systems, which have

established a trust relationship via an attestation protocol to authenticate the hardware/software configuration of the remote device (such as the BIOS, MBR, firmware); and (c) compute a non-repudiated proof of existence (along with other properties) of the acquired forensic evidence.

As shown in Fig. 1, the latter property is achieved by means of the CTB. The logger generates *evidence log events*, denoted by the EvGen function, at the time that new evidence material is being inserted into the evidence DB, and signs these events. To achieve this step, the logger needs to have generated a key pair for use with digital signature algorithms, something that requires a *certificate authority* (CA) — HyperLedger Fabric’s CA is used for that purpose. When a new signed evidence *ev* is inserted in the evDB, a new identifier *id* is created as

$$id = \text{Hash}(ev \parallel \text{nonce})$$

where the value *nonce* is chosen uniformly at random to ensure the uniqueness of the evidence’s identifier. Note that *id* serves the purpose of the signed evidence log event’s integrity proof that can be verified by means of a cryptographic hash function; the evidence identifier, and the nonce used, are also stored in the evDB along with the actual data.

After computing the integrity proofs of the signed evidence log events, each proof is written to the CTB through a series of transactions, which is denoted by the function TxGen, for subsequent generation of the next block in CTB blockchain. The blockchain explorer can then be used for retrieving the immutable record of integrity proofs on the blockchain and validate forensic evidences’ properties.

#### IV. FORENSIC EVIDENCE BLOCKCHAIN

In the course of digital forensic investigations, the evidence examination needs to be carried out by authenticated entities, while ensuring privacy requirements. Due to this fact, only the forensic evidences’ metadata are stored in the CTB, which is a permissioned distributed ledger build on HyperLedger Fabric, to provide auditing and integrity services on evidence gathered from a smart home environment. To realize the CoC and allow the entities involved to access the digital evidence, information about the chronological history of handling the evidence has to be recorded. The authenticated entities that may obtain the ownership of a forensic evidence, issue new transactions and create blocks (that contain change of ownership information), are classified as (also referred to as participants):

- *Internet service provider*. Collects the evidence regarding a security incident from the smart home environment as described in Section III-C. As the creator of the evidence, only the ISP is able to permanently delete it, regardless who the current owner is.
- *Law enforcement agency*. Can access the evidentiary data about a particular *id*, IoT device, or attack that are stored in the CTB when conducting an investigation. LEAs can also issue new transactions to transfer ownership.
- *Prosecutor*. Considered to be the final owner of the digital forensic evidence in the course of an investigation.

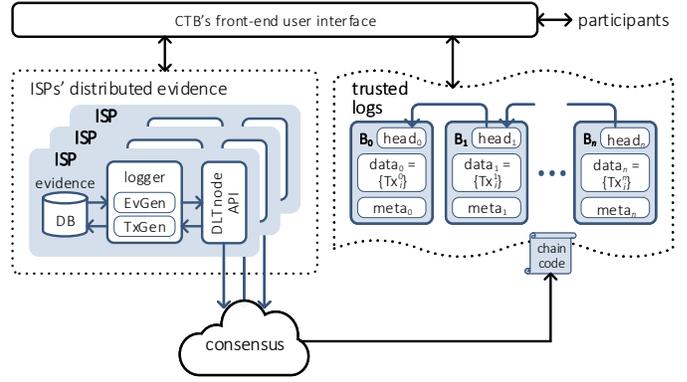


Fig. 2. High-level architecture of the Cyber-Trust’s blockchain.

In the high-level architecture of CTB, that is illustrated in Fig. 2, the transactions stored are about the actions performed by the involved entities and also record the ownership transfer of the digital evidence from the moment of its collection until it reaches the prosecutor. The CTB is comprised of the following core components: (a) the front-end *user interface* (UI), (b) the blockchain node, (c) the trusted transaction logs, and (d) the forensic evidence DB. More precisely:

- *Front-end UI*. Interface allowing the participants to view, invoke, or query blocks, transactions, chaincodes, etc., in the CTB; it is based on Fabric’s blockchain explorer.
- *Blockchain node*. This component ensures that authorized participants can communicate with the CTB network.
- *Trusted logs*. Implements the blockchain and stores the historical record of facts about when evidence was created and how its ownership was transferred from one entity to another so as to arrive at the current system state.
- *Forensic evidence DBs*. The off-chain databases, in which the current owner of an evidence has access to, where the raw evidentiary material is stored.

Note that there are several forensic evidence DBs, one for each ISP, and therefore, upon request of a particular evidence, the front-end UI delegates the request for access to the appropriate ISP. The design of the CTB provides main function allowing the participants to create, transfer, erase or view the evidences stored in the evidence DB. Each function, if properly invoked, issues and broadcasts a new transaction to the network.

CreateEvidence(*id*, *dsc*). This function submits a new block to the CTB with the identifier *id* and the description *dsc* of the new evidence as input. The function’s role is not just to create a new evidence, but also checks if an evidence with the same *id* has already been created. Another functionality is to set the first owner of the evidence, which by default is evidence’s creator (i.e. the ISP).

GetEvidence(*id*). Given as input an evidence identifier, the function displays/retrieves the evidence after having first checked that the evidence indeed exists and the requesting participant is its current owner.

EraseEvidence(*id*). The function checks if the evidence with

identifier  $id$  has already been stored in the CTB and if the invoking participant is the ISP that created the evidence. It is evident that forensic evidences' metadata cannot actually be erased from the CTB, as this would imply that the entire blockchain would have to be reformed. The function just deletes the evidence from the  $evDB$  and then issues a new transaction declaring that the evidence no longer exists.

$TransferOwnership(id, own)$ . Given an evidence identifier  $id$  and a participant address  $own$ , the function checks various conditions. First, the evidence must exist in the CTB and the participant invoking the function has to be the current owner of the evidence. Then, the function checks if  $own$ , where the evidence will be transferred to, is authorized to access the evidence. If all conditions are true, the function transfers ownership of the evidence to the new owner  $own$ , and the address of the new owner is added to the CTB.

New evidence is defined as a transaction having the following metadata: the evidence identifier  $id$ , the address creator of the ISP having collected the evidence, the description  $dsc$  of the security incident (initialized by the creator, and later updated by other participants) and a timestamp time of its occurrence, the current  $own$  (resp. previous  $own'$ ) owner of the evidence, the type (type) of the attacked IoT device, as well as, the list of time records  $\{\tau_i\}_{i=1,2,\dots}$  that each owner had the evidence at his possession. The form of each transaction stored in the CTB is the following

$$Tx = id || creator || dsc || time || own || own' || type || \{\tau_i\}.$$

Let us note that, in the context of HyperLedger Fabric, only a transaction's *proposal* field is shown above, which encodes the input parameters to the chaincode for creating the proposed ledger update; trivial fields, such as a transaction's header and signature, are omitted for simplicity. Since the security of CTB is of utmost importance, a number of fundamental properties need to hold [20], the analysis of which is outside the scope of this work, such as persistence, liveness, chain quality property, and common prefix property. If all true, they considerably limit the ability of adversaries to alter CTB's evidentiary metadata.

## V. CONCLUSIONS

The Cyber-Trust platform relies on advanced intrusion detection tools to identify malicious activities and enhance the security of IoT environments by inspecting compromised devices and collecting forensic evidence so as to determine the source of cyber-attacks. The evidentiary information is safely stored as raw data in an off-chain database, while the hashes and metadata of the evidence are stored on the blockchain. The CTB is a permissioned distributed ledger, which is build on top of HyperLedger Fabric. The Cyber-Trust's blockchain-based solution dematerializes the CoC process of recording and preserving a chronological history of digital evidences.

## REFERENCES

[1] F.-C. Cheng, "Automatic and secure wi-fi connection mechanisms for iot end-devices and gateways," in *Emerging Technologies in Computing*, M. H. Miraz, P. Excell, A. Ware, S. Soomro, and M. Ali, Eds. Springer International Publishing, 2018, pp. 98–106.

[2] K. Zhao and L. Ge, "A survey on the internet of things security," in *Proceedings of the 2013 Ninth International Conference on Computational Intelligence and Security*. Washington, DC, USA: IEEE Computer Society, 2013, pp. 663–667.

[3] C. Koliadis, G. Kambourakis, A. Stavrou, and J. Voas, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[4] A. MacDermott, T. Baker, and Q. Shi, "Iot forensics: Challenges for the iot era," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Feb 2018, pp. 1–5.

[5] C. J. D'Orazio, K. R. Choo, and L. T. Yang, "Data exfiltration from internet of things devices: ios devices as case studies," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 524–535, April 2017.

[6] C. J. Fung, O. Baysal, J. Zhang, I. Aib, and R. Boutaba, "Trust management for host-based collaborative intrusion detection," in *Managing Large-Scale Service Deployment*, F. De Turck, W. Kellerer, and G. Kormentzas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 109–122.

[7] T. D. Nguyen, S. Marchal, M. Miettinen, M. H. Dang, N. Asokan, and A. Sadeghi, "Diot: A crowdsourced self-learning approach for detecting compromised iot devices," *CoRR*, vol. abs/1804.07474, 2018.

[8] H. Haddadi, V. Christophides, R. Teixeira, K. Cho, S. Suzuki, and A. Perrig, "Siotome: An edge-isp collaborative architecture for iot security," in *Proceedings of International Workshop on Security and Privacy for the Internet-of-Things (IoTSec) 2018*. ETH Zürich, 2018, 1st International Workshop on Security and Privacy for the Internet-of-Things (IoTSec); Conference Location: Orlando, Florida, USA; Conference Date: April 17, 2018.

[9] A. Houmansadr, S. A. Zonouz, and R. Berthier, "A cloud-based intrusion detection and response system for mobile phones," in *2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W)*, June 2011, pp. 31–32.

[10] A.-D. Schmidt, F. Peters, F. Lamour, C. Scheel, S. A. Çamtepe, and Ş. Albayrak, "Monitoring smartphones for anomaly detection," *Mobile Networks and Applications*, vol. 14, no. 1, pp. 92–106, Feb 2009.

[11] E. Oriwoh and P. Sant, "The forensics edge management system: A concept and design," in *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing*, Dec 2013, pp. 544–550.

[12] A. Goudbeek, K. R. Choo, and N. Le-Khac, "A forensic investigation framework for smart home environment," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, Aug 2018, pp. 1446–1451.

[13] M. Faheem, N.-A. Le-Khac, and T. Kechadi, "Smartphone forensic analysis: A case study for obtaining root access of an android samsung s3 device and analyse the image without an expensive commercial tool," *Journal of Information Security*, vol. 5, pp. 83–90, 01 2014.

[14] M. Raji, H. Wimmer, and R. J. Haddad, "Analyzing data from an android smartphone while comparing between two forensic tools," *SoutheastCon 2018*, pp. 1–6, 2018.

[15] P. Andriotis, G. Oikonomou, and T. Tryfonas, "Forensic analysis of wireless networking evidence of android smartphones," in *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec 2012, pp. 109–114.

[16] N. Alexopoulos, E. Vasilomanolakis, N. R. Ivánkó, and M. Mühlhäuser, "Towards blockchain-based collaborative intrusion detection systems," in *Critical Information Infrastructures Security*, G. D'Agostino and A. Scala, Eds. Cham: Springer International Publishing, 2018, pp. 107–118.

[17] W. Meng, E. W. Tischhauser, Q. Wang, Y. Wang, and J. Han, "When intrusion detection meets blockchain technology: A review," *IEEE Access*, vol. 6, pp. 10 179–10 188, 2018.

[18] A. Lone and R. Mir, "Forensic-chain: Blockchain based digital forensics chain of custody with poc in hyperledger composer," *Digital Investigation*, vol. 28, 03 2019.

[19] M. Hossain, R. Hasan, and S. Zawoad, "Probe-iot: A public digital ledger based forensic investigation framework for iot," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2018, pp. 1–2.

[20] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Advances in Cryptology — CRYPTO 2017*. Springer, 2017, pp. 357–388.