Multi-party Consensus-based Blockchain for Chain of Custody in IoT Forensic Investigations

Riccardo Pezzoni, Antonio Boiano, Fabio Palmese, Alessandro E. C. Redondi
DEIB, Politecnico di Milano
Milan, Italy
{*name.surname*}@polimi.it

***Abstract*—The rapid adoption of Internet of Things (IoT) devices in smart environments has led to a new era for digital forensics. As IoT devices become increasingly prevalent in homes, cities, and workplaces, they serve as silent observers of everyday human activities. Recent studies have investigated how network traces from these devices could be leveraged to support forensic investigations. However, this approach requires large-scale data collection while ensuring the confidentiality, anonymity, and integrity of the collected traces. This work advances the field of IoT forensics by introducing Chain4ensic, a blockchain-based chain-of-custody (CoC) framework designed to preserve network traces extracted from IoT devices in a secure way. The proposed framework utilizes Ethereum smart contracts and edge computing to preserve the integrity and traceability of digital evidence. Additionally, the proposed system facilitates public notifications of data access once authorized by a court of justice. The architecture is developed as an open-source solution and evaluated in a smart home scenario, demonstrating its feasibility and effectiveness in real-life applications. The results indicate low resource usage, high throughput, and small gas consumption, making it a promising approach to tackle the challenges of storing sensitive data from participants in IoT forensic investigations.**

***Index Terms*—IoT Forensics, Chain-of-Custody, Blockchain**

I. Introduction

The increasing deployment of consumer Internet of Things (IoT) devices in homes and workplaces, where people usually spend most of their time, generates new digital witnesses of human daily activities. The digital traces produced by IoT devices can be beneficial for solving forensic cases, as they serve as additional sources of evidence to establish facts or incriminate individuals engaged in crimes. For this reason, a branch of digital forensics focusing on smart devices, named IoT forensics, has become increasingly popular in current research. IoT forensics takes care of the full process, covering the collection, preservation, analysis, and presentation of the traces extracted from IoT infrastructure to investigate and establish crime scene evidence [1]. In traditional digital forensics, evidence is retrieved directly from the devices' memory. However, in IoT scenarios, devices are often constrained by limited memory capacity, making them retain only a restricted amount of data to be potentially included in investigations. Furthermore, the dynamic cloud-based nature of the IoT ecosystem makes the extraction of forensic traces very tedious and complex, given that data could be located under many different cloud servers and possibly under different jurisdictions. To overcome these limitations, recent research investigated the potential of network traffic exchanged by IoT devices as an additional source of evidence to be collected as close as possible to the originating devices. Despite traffic encryption, the state-of-the-art proved that in-place collection of network traffic with proper analysis techniques allows the reveal of extremely useful information on both the device and the user activity [2], [3]. It is indeed possible to use the traffic traces to distinguish the type of device [4] or to reveal user interactions with the devices (e.g., with smart cameras or voice assistants [5]). For these reasons, different frameworks have been recently proposed for network traffic feature extraction in IoT gateways to enhance forensic analysis [6]. However, storing the traffic for an extended period can become extremely heavy, given the amount of data transmitted by the increasing number of devices involved in the IoT paradigm. Moreover, since IoT network traffic represents potential evidence that reveals information on the user, it is fundamental to store those traces securely and build an infrastructure capable of protecting the user's privacy from malicious or unauthorized entities, while maintaining the integrity of digital evidence to be presented in the investigation process. Therefore, a secure Chain of Custody (CoC) is required to ensure that traces are protected and secure for the full process from the acquisition to the final presentation. To tackle this problem, we propose Chain4ensic, an open-source and publicly available implementation of a blockchain-based infrastructure[1] aiming to collect, secure, protect, store, and retrieve network traces from IoT devices, ultimately working as a CoC. The proposed solution is based on Ethereum, taking advantage of all its capabilities, while the storage platform leverages edge-computing technologies. This serves the scope of securing, protecting, and storing recorded traces, ensuring its preservation and immutability. As additional support to the CoC process, all data access requests are logged and made public to avoid unauthorized data access from law enforcement authorities (LEAs).

The rest of the work is organized as follows: Section II presents related work about CoCs for IoT forensics. Section III presents an architecture overview of the proposed CoC infrastructure. Section IV showcases proof of concept of the Chain4ensic architecture and presents a performance evaluation in a smart-home IoT scenario. Finally, section V concludes the work with final remarks and future research directions.

II. Related Work

Blockchain-based Chain of Custody for IoT forensics is a novel and emerging research area that addresses the challenges of collecting, preserving, and analyzing digital evidence from IoT devices and networks. Several works have proposed different architectures and frameworks to integrate blockchain technology with IoT forensics, as recently summarized in [7]. The authors of [8] remark that public blockchains are among the most decentralized architectures, permitting

[1] https://github.com/antonio-boiano/chain4ensic

979-8-3503-9281-4/25/$31.00 ©2025 IEEE

any entity to join and participate. However, this openness also means that such systems may compromise privacy and anonymity, resulting in a weaker solution with reduced chain of custody security, and limited traceability of stakeholder identities. This leads the authors to opt for a permissioned blockchain, where all nodes participating in consensus are known, and access is limited to authorized entities [9]. Mercan et al. [10] implemented a CoC architecture to overcome the costs of storing huge amounts of data in a public blockchain due to transaction fees. They exploited multiple inexpensive blockchain networks as temporary storage before the data is committed to Ethereum, finding that the framework significantly reduces the costs and it is therefore attractive for companies. In [11], Ahmad et al. employ a private Ethereum blockchain to record all transmissions about digital evidence. This framework integrates smart locks alongside the digital evidence system, where evidence is stored securely and locked. All communications within the framework occur through a peer-to-peer network, allowing the framework to maintain realistic workloads with an acceptable transaction throughput. The authors in [12] solve the problem of current digital forensic frameworks not meeting the heterogeneity and distribution characteristics of the IoT environment by proposing a permissioned Ethereum private network platform (GEth) running Proof of Work and using smart contract interfaces for evidence generation, acquisition, and report generation. Le et al. [13] propose a solution centralizing many processes in the hands of a Law Enforcement Authority that serves as public keys distributor in the registration process and as a verifier of the evidence submitted by users. Smart contracts handle basic interactions, and Bizantine Fault Tolerance consensus is used. Many papers use blockchain's inherent properties to enhance transparency, tamper-resistance, and verifiability throughout the CoC lifecycle. However, only a few works provided functioning implementations, and many frameworks are limited to a detailed explanation of the forensic interactions and the architectural choices made. The lack of standardized evaluation tools and the diversity of implementations make a direct performance comparison challenging. In contrast to the reviewed literature, we present a functional implementation of a blockchain-based chain of custody with multi-party consensus, mimicking bureaucratic behaviors for data management and enforcing transparency for both on-chain and off-chain operations. Our idea relies on the concept of independent law enforcement organisms working together while simultaneously functioning as controllers. We detail the architectural decisions and the required forensic interactions and provide a performance evaluation in a simulated smart-home environment, an additional precious contribution for a real-life deployment.

III. Chain of Custody Architecture

To address the challenges IoT forensics poses, such as secure collection, storage, and retrieval of digital traces, this work proposes a blockchain-based CoC architecture to support forensic investigations. In legal contexts, a CoC is defined as a chronological documentation that records the sequence of custody, control, transfer, analysis, and final disposition of physical and digital evidence. Ensuring an immutable and transparent CoC is essential in any forensic investigation to safeguard the integrity of evidence from its collection to its presentation in legal proceedings. In our intended use case, the digital traces are represented by the network traces generated by IoT devices. This includes raw traffic data, aggregated traffic statistics, and even user logs. These traces, once processed with proper Machine Learning algorithms, can serve as potential evidence in forensic investigations [5], [6]. Due to the digital nature of such data, maintaining its integrity and ensuring it remains tamper-proof are key requirements for its admissibility in legal contexts. At the same time, the CoC system must comply with data protection laws established by various governments. Specifically, in the European Community, two major regulations have been defined regarding data protection and access for investigations: the European Union's General Data Protection Regulation (GDPR; Regulation (EU) 2016/679) [14], which governs the lawful processing and protection of personal data, and the Law Enforcement Directive (LED; Directive (EU) 2016/680) [15], which complements the GDPR by permitting data access to competent law enforcement authorities. Notably, Article 8 of the LED requires that any interference with privacy for investigation purposes must be necessary, proportionate, and subject to adequate judicial or administrative oversight.

The CoC in Chain4ensic is built with these requirements in mind. It prevents the misuse of data by a single authority by incorporating oversight mechanisms involving multiple institutions, thereby ensuring that both the integrity of the digital evidence and the privacy rights of individuals are maintained throughout the investigative process. The proposed CoC architecture is intended for two key members:

- **User**: The owner of IoT devices that generate traffic traces. By joining the CoC system, the user acknowledges that the traces from their IoT devices could be valuable in aiding forensic investigations. Although this data may be shared for investigative purposes, robust encryption and privacy safeguards within the system ensure the user's privacy is fully protected.
- **Trusted Authority**: A legal entity authorized to participate in the investigation process, such as a law enforcement agency or a courthouse. Trusted authorities are occasionally interested in retrieving evidence stored on the blockchain for forensic purposes.

A sketch of the proposed CoC architecture is shown in Figure 1, and the design follows this logic: upon user acceptance, the IoT gateway (e.g., a smart home Access Point) authenticates and joins the CoC. The reason for leveraging the IoT gateway for network trace retrieval and storage resides in the more advanced computing capabilities of these devices. Following the CoC pipeline, the traffic traces are then encrypted and periodically published to a distributed storage server. A transaction is also logged on the blockchain, ensuring the integrity of the submitted evidence. In the case of a forensic investigation, a trusted authority is granted access to the data by a court of justice. Data retrieval is then performed by interacting with the blockchain, contingent upon the shared agreement of other legal bodies.

In the following sections, we outline the CoC architecture components and their interactions.

A. Blockchain design

Prior to making any design choice for the CoC, the decision regarding the type of blockchain and consensus mechanism is pivotal. The first step is to determine whether

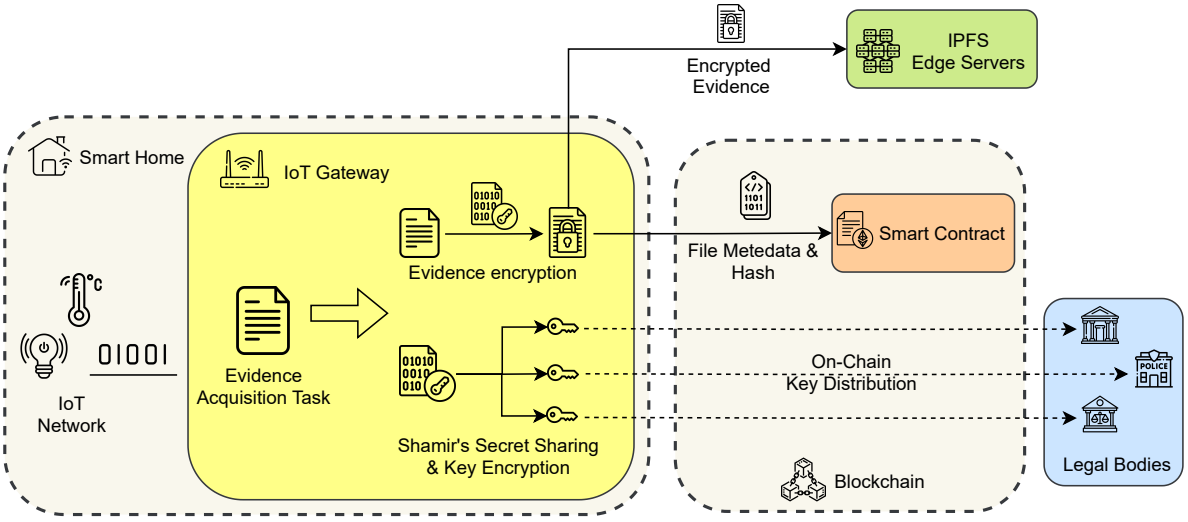

Fig. 1. Chain of custody overview in the submission process of a new evidence

the blockchain should be private or public. Both types are commonly used in the literature and offer distinct advantages and disadvantages (as detailed in Section II). In our proposal, we rely heavily on the trust of legal entities and the concept of mutual oversight among law enforcement entities. Therefore, a private blockchain is more suitable for this purpose. Nevertheless, the detailed approach presented in this work can also be applied easily to a public blockchain. The second step in any blockchain-based design involves selecting the appropriate consensus mechanism to elect the validator nodes, which are responsible for validating the correctness of transactions. In our CoC, we selected the Proof of Authority (PoA) consensus mechanism. PoA is well-suited for forensic applications because it provides efficient and lightweight consensus through trusted validator nodes, in contrast to the resource-intensive Proof of Work (PoW) mechanism that demands significant computational power from miner nodes. In PoA, validator nodes are pre-approved legal entities such as law enforcement agencies. Due to the aforementioned blockchain design choices as well as the capabilities of executing small functions and and its widespread adoption, we chose Ethereum as the underlying framework for our CoC system. Evidence management within our proposed system is conducted using Ethereum smart contracts. A smart contract is a self-executing piece of code deployed on the blockchain that automates specific actions when predefined conditions are met. Within the proposed system, smart contracts manage the evidence lifecycle, from creation to retrieval. Each user is assigned a unique smart contract upon registration, which records all actions related to the submission and management of their evidence.

B. Evidence Collection Pipeline

This section outlines the comprehensive pipeline that evidence undergoes from the acquisition to the storage phase. Once the forensic-ready IoT gateway is configured for digital evidence acquisition, it begins capturing digital traces. The collected data are then subject to feature extraction and compression processes to reduce dimensionality [5]. When the accumulated traces reach a predefined threshold determined by factors such as time intervals or data size, the system initiates the offloading of this data to the CoC. This offloading process is essential to prevent overloading the IoT gateway's memory capacity and to ensure that the traces remain admissible in legal proceedings by safeguarding against data tampering and maintaining data accessibility. This offloading process consists of the following steps:

- **Key Generation and Evidence Encryption:** An encryption key is generated to encrypt the collected evidence before uploading it to the Chain of Custody (CoC).
- **Publishing Encrypted Evidence:** The encrypted evidence is stored on a distributed file system such as the InterPlanetary File System (IPFS) or an equivalent data storage platform which can guarantee data redundancy.
- **Key Sharding Using Shamir's Secret Sharing:** The encryption key is split into multiple shards using Shamir's Secret Sharing (SSS) algorithm [16], enhancing security by ensuring that the complete key cannot be reconstructed without a quorum of shards.
- **Encrypting Shards with Legal Entities' Public Keys:** Each shard is encrypted using the public key of a different legal entity and published on the blockchain. This ensures that no single entity can access the entire encryption key, promoting mutual oversight among independent legal bodies.
- **Publishing Metadata on the Blockchain:** Alongside the encrypted shards, the metadata of the evidence file is published on the blockchain. Specifically the following information are reported: (i) The cryptographic hash (SHA256) of an uploaded evidence file, serving as a unique identifier to ensure file integrity; (ii) Evidence metadata including timestamps, user identifiers and file storage locations; (iii) encrypted key shards to reconstruct the evidence.
- **Data Validation:** A final validation step computes the hash of the file published on the storage system. If this matches the hash declared by the IoT gateway, the chain flags the file as verified.

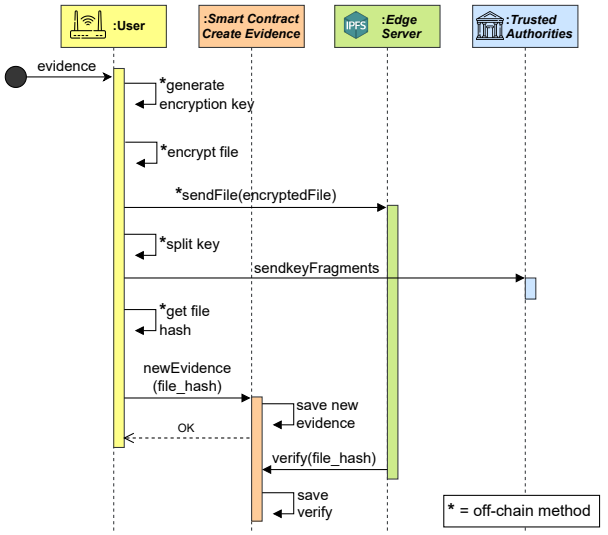

Fig. 2. Flow diagram of the Evidence Creation phase

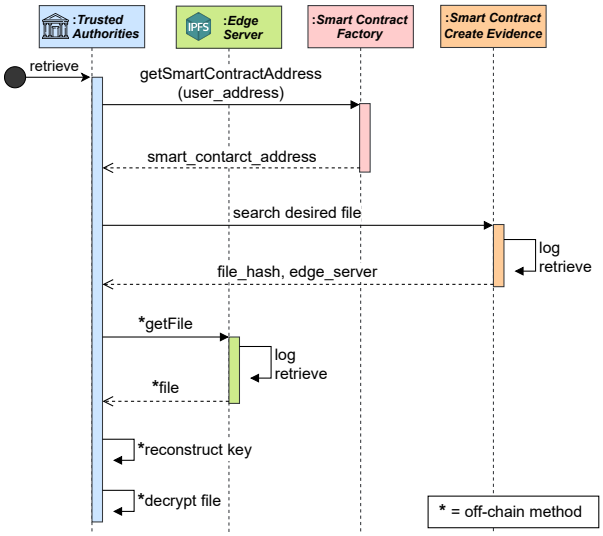

Fig. 3. Flow diagram of Evidence Retrieval from an authorized authority

C. Components Interactions

The architecture supports three key interactions between the system components: user registration, evidence creation, and evidence retrieval.

1) User Registration: The first step in interacting with the CoC system is user registration. This is a one-time process that ensures only authenticated users can submit and manage evidence in the private chain. The full registration process, works as follows:

- The IoT gateway joins the CoC and invoke the SmartContractFactory to create a user-specific smart contract. This contract will manage all future evidence submissions for that user.
- The user-specific smart contract is permanently linked to the user's public address, and all actions issued by the users (e.g., evidence submission or retrieval) are logged within this contract.

To prevent unauthorized entities from accessing the blockchain, authentication keys are pre-provisioned for each IoT gateway. Additionally, the number of allowed transactions per user is limited to one per day, further mitigating the risk of chain overload.

2) Evidence Creation: Once authenticated, users can submit evidence generated by their IoT devices. The process follows the data pipeline described in Section III-B. The evidence file is first encrypted using a unique key generated by the user's device. This key is then split into multiple shards using the SSS algorithm. This key splitting ensures that no single entity can reconstruct the key independently; instead, a threshold number of shards is required for decryption, which protects the evidence from unauthorized access. The encrypted file is uploaded to a distributed storage network, specifically the InterPlanetary File System (IPFS). IPFS provides decentralized storage, ensuring that the evidence is resilient to tampering or loss by eliminating centralized points of failure. The encrypted file hash together with the encrypted shards, are stored on the blockchain through the user's smart contract. This hash serves as a unique identifier and ensures the file's integrity. Any tampering with the file would be detectable via a hash mismatch.The shard distribution instead ensures that any attempts to reconstruct the key are visible to all participating authorities. This two-step process ensures the evidence is securely stored and verifiable while maintaining a complete chain of custody from its creation. The full process is reported as a flow diagram in Figure 2, where the main interactions are reported.

3) Evidence Retrieval: Retrieving evidence from the CoC system is a highly controlled process designed to ensure the integrity of the evidence while allowing access only to authorized entities. The retrieval process follows these steps, as outlined in the flow diagram in Figure 3: (i) When a trusted authority (e.g. law enforcement entity) requires access to evidence, they query the blockchain for the relevant user-specific smart contract. This contract holds crucial details, including the file's hash, storage location, encrypted shards, and the identities of authorities capable of decrypting them. (ii) The trusted authorities must collaborate to reconstruct the encryption key upon verification of the right to the requested data. Since no single authority possesses enough key shards to decrypt the file alone, this process mirrors the checks and balances of real-world forensic investigations, ensuring that multiple independent entities must authorize access. (iii) The file hash is used to retrieve the encrypted file from the IPFS edge server, which follows a DHT-based storage using file hashes for querying. As the encryption key is reconstructed from the shards, the evidence file can be decrypted and used for the investigation process. The blockchain is then updated to log the retrieval, ensuring a transparent and immutable record of the access events. Furthermore, the IPFS edge server, which manages the file storage, verifies the retrieval request by checking the blockchain to ensure that the request is legitimate before transmitting the encrypted data.

IV. Performance Evaluation

We implemented and tested the proposed solution locally to properly test the architecture and analyze its performance metrics. The user node is executed in a Raspberry Pi 4, chosen for its hardware characteristics aligning closely with commercial IoT gateways, reflecting potential real-world deployments. The local Ethereum node is run with a Docker

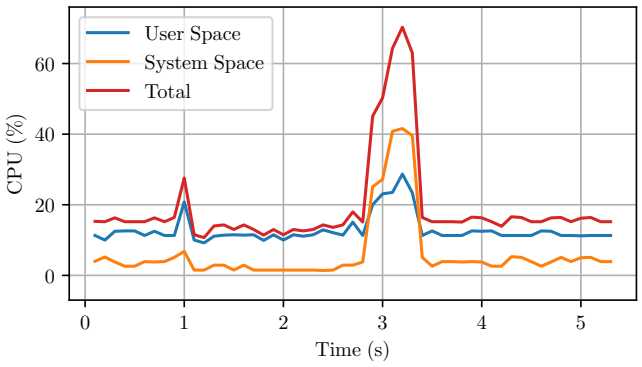

Fig. 4. CPU consumption during the single creation of an evidence

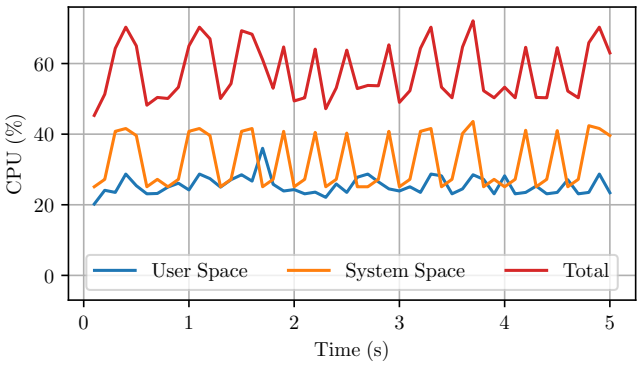

Fig. 5. CPU consumption during continuous creation of evidences

container running a Go-Ethereum (Geth) image. Geth is one of the most popular Ethereum client implementations, written in Go; it allows one to join both public and private networks and exposes a Javascript console to receive commands with the JSON-RPC-API. To join the private blockchain, the client only needs the *genesis.json* file, defining the origin block of the network, the chain identifier, and some parameters such as the type of consensus, the block size, and the pre-founded accounts. Nodes can form a blockchain together only with the same Genesis file. The configuration to execute the node is minimal and requires only a few environment variables, which are essential to setting connection details, such as the willingness of the node to be a signer and the address of the bootnode. Given that the Ethereum blockchain's capacity to handle concurrent transactions and scale effectively under a Proof of Authority (PoA) consensus has been well-established and tested in both literature and practice, we can confidently conduct all tests by simulating on an external workstation all the additional components of the architecture, such as other user nodes and signer nodes representing the trusted authorities. Our analysis will focus on the performance of the local user node updating new pieces of evidence. To assess the capabilities and limitations of the local architecture, the user node is set to record the CPU and RAM usage during its operations. Considering the deployment scenario involving numerous devices in individual homes, cost-effectiveness and energy efficiency are paramount. Therefore, a key objective of the various tests is to determine the viability of using a single-board computer like the Raspberry Pi for a real-life deployment.

A. Single Creation

Given that real-world implementations will likely involve limited evidence submissions (e.g., one per day), the initial performance evaluation focuses on generating well-spaced evidence. The user node is first allowed to run idle; then, a single evidence generation process is initiated, followed by a return to idle state. During evidence generation, the node encrypts a 100MB file, computes its hash, invokes the smart contract method to upload the evidence, and transmits the file to the edge server via a standard socket connection. Analysis of CPU and RAM usage reveals low resource consumption during idle periods. We report the CPU usage value during the evidence creation process in Figure 4. At time $t = 0$, the device starts idle, and the CPU value is low. At approximately $t = 1s$, we notice a first spike in CPU usage, corresponding to file encryption and hashing. As the file is successfully encrypted, the CPU usage returns to a low value until $t = 3s$, where a bigger spike is observed. This latter is related to establishing two socket connections: one for the Go-Ethereum client's interaction with the network and the other for the file transfer. It is important to note that smart contract execution on a node involves local bytecode execution followed by transaction signature generation. In the idle periods, the local node maintains communication with the bootnode to monitor the transaction status and log its approval by a signer node. Significantly, even when other nodes are active and publishing evidence, the monitored local node continues to exhibit low CPU usage during idle periods. This is because only signer nodes actively participate in transaction validation and processing. The results demonstrate favorable performance, low idle consumption, and CPU usage. Docker-specific data, excluding the overhead of Docker and the operating system, reveals an idle CPU usage in the 1.5-2.5% range. This low idle consumption is a key advantage, as it suggests the feasibility of deploying the architecture on existing devices without significantly impacting their overall resource usage.

B. Continuous Creation

Our second test is executed with continuous evidence generation to understand the performance in a more intense scenario. Even if this case deviates from potential real-life cases, it is useful to understand the limitations of the proposed framework. After determining the hardware's maximum transaction throughput at around 23 transactions per second (tps), a continuous creation rate of 20 tps has been selected. The monitored CPU usage is reported in Figure 5 and shows a significantly elevated and consistent CPU load, as the computationally intensive tasks of encryption, hashing, and transaction creation are continuously executed. The results indicate that, while the Raspberry Pi has good capacity for individual evidence creation, continuous generation places a substantial demand on its resources and continuously stresses its CPU. However, the values suggest that the selected hardware can properly handle a creation rate of 20 tps and could potentially support even higher transaction rates if required in specific applications.

C. Architecture Test

To evaluate the real-world viability of the architecture, we conduct additional tests on the remaining components. We aim to establish the number of trusted authorities required under various scenarios and estimate the necessary storage

capacity. Experiments using a medium-level home desktop computer (Mac Mini with M1 and 16GB RAM) as a signer node demonstrated a maximum throughput of 120 validated transactions per second (tps) without noticeable delay. However, the work in [17] demonstrated a linear relationship between computational resources and transaction throughput, from which we can conclude that server-grade hardware could easily achieve significantly higher throughput in the order of thousands of tps. Assuming a conservative estimate of one piece of evidence per day per user (i.e., transmitting daily logs), with evidence generation evenly distributed throughout the day and an average throughput of 500 tps, a reasonable estimate for the number of required validator nodes is approximately one per 40 million users. This demonstrates the architecture's efficiency and suggests that a relatively low number of trusted authorities would be needed for a real deployment. Experimentation with the Feature Sniffer network forensics tool [5] on a household with 20 IoT devices revealed a daily data collection that can be reduced to 4MB after proper feature extraction and compression. Even with this compression level, a medium-sized city of 100,000 households would generate roughly 12TB of data per month. While this storage requirement necessitates careful planning, it indicates that real-world deployment is feasible. For example, distributing edge servers with 1TB monthly storage capacity across every 8.000 households, or roughly one per village or city neighborhood, would be sufficient to deploy the system.

V. Conclusion

Common IoT devices that fill our homes, offices, and businesses can represent a treasure during investigations and trials as they continuously share data that can be used to witness our daily activity. Considering the extremely transient nature of network traffic, usually involved in IoT forensics, a secure chain of custody is required to ensure the data are safely collected and preserved during all the phases of the forensic process. This work presented Chain4ensic, a blockchain-based chain of custody covering evidence preservation in forensic investigation, with a safe storage place and a publicly verifiable log. The architecture built using a private blockchain based on Ethereum with Proof of Authority proved to satisfy all the requirements typical of the forensic scenario as being a reliable log, allowing data immutability, and forbidding unwanted entities from accessing the data. In detail, data is stored in edge servers using InterPlanetary File Systems, while any data access is handled using the blockchain. Each file is encrypted, and its hash is sent in the chain before its transmission to the server. In addition, the file encryption key is shared with different trusted authorities using Shamir secret sharing.

After presenting the framework architecture with its implementation details, we validate the proposed solution to understand its possible uses in a real deployment. We set up the testbed running the user node in a Raspberry Pi 4, and we show that the evidence creation process can be handled properly with low impact on the device processing capabilities, even with high transaction rates. We conclude with an estimation of a real-world deployment, and we show that a very limited number of validators would be required to support the solution on a worldwide scale. Using a powerful server as a validator would be enough to support more than 40 million user nodes. In future research, we plan to deploy a real network with nodes running in Wi-Fi access points and to integrate the evidence creation process within the Feature-Sniffer framework to obtain a full-functional forensic home gateway with proper distributed evidence preservation.

Acknowledgments

This work was supported by the European Union and the Italian Ministry for University and Research (MUR) through the PRIN project "COMPACT" (Mission 4, Component 1, CUP D53D23001340006) and the Extended Partnership MICS (PE00000004, CUP D43C22003120001) under the Italian National Recovery and Resilience Plan (NRRP). All projects are funded by NextGeneration EU.

References

[1] M. Stoyanova et al., "A Survey on the Internet of Things (IoT) forensics: Challenges, Approaches, and Open Issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1191–1221, 2020.

[2] J. Ren, D. J. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun, and H. Haddadi, "Information Exposure From Consumer IoT Devices: A Multidimensional, Network-Informed Measurement Approach," in *Proceedings of the Internet Measurement Conference*, 2019, pp. 267–279.

[3] A. Acar et al., "Peek-a-boo: I see your smart home activities, even encrypted!" in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2020, pp. 207–218.

[4] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Classifying IoT devices in smart environments using network traffic characteristics," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1745–1759, 2018.

[5] F. Palmese, A. E. C. Redondi, and M. Cesana, "Designing a Forensic-Ready Wi-Fi Access Point for the Internet of Things," *IEEE Internet of Things Journal*, vol. 10, no. 23, pp. 20 686–20 702, 2023.

[6] A. Boiano, A. E. Cesare Redondi, and M. Cesana, "IoTScent: Enhancing Forensic Capabilities in Internet of Things Gateways," in *2023 IEEE 9th World Forum on Internet of Things (WF-IoT)*, 2023, pp. 1–6.

[7] Sakshi et al., "A survey on blockchain based IoT forensic evidence preservation: research trends and current challenges," *Multimedia Tools and Applications*, vol. 83, no. 14, pp. 42 413–42 458, 2024.

[8] S. Chen, C. Zhao, L. Huang, J. Yuan, and M. Liu, "Study and Implementation on the Application of Blockchain in Electronic Evidence Generation," *Forensic Science International: Digital Investigation*, vol. 35, p. 301001, 2020.

[9] S. Bano et al., "SoK: Consensus in the Age of Blockchains," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, 2019, pp. 183–198.

[10] S. Mercan, M. Cebe, E. Tekiner, K. Akkaya, M. Chang, and S. Uluagac, "A cost-efficient iot forensics framework with blockchain," in *2020 IEEE international conference on blockchain and cryptocurrency (ICBC)*. IEEE, 2020, pp. 1–5.

[11] L. Ahmad, S. Khanji, F. Iqbal, and F. Kamoun, "Blockchain-based Chain of Custody: Towards Real-time Tamper-proof Evidence Management," in *15th International Conference on Availability, Reliability and Security*, 2020, pp. 1–8.

[12] J. H. Ryu, P. K. Sharma, J. H. Jo, and J. H. Park, "A blockchain-based decentralized efficient investigation framework for IoT digital forensics," *The Journal of Supercomputing*, vol. 75, pp. 4372–4387, 2019.

[13] D. Le et al., "BIFF: A blockchain-based IoT forensics framework with identity privacy," in *TENCON 2018-2018 IEEE region 10 conference*. IEEE, 2018, pp. 2372–2377.

[14] European Parliament and Council of the European Union, "Regulation (EU) 2016/679 of the European Parliament and of the Council," *OJ L 119, 4.5.2016, p. 1–88*, 2016.

[15] E. Parliament and of the Council, "Directive 2016/680 of the european parliament and of the council of 27 april 2016 on the protection of individuals with regard to the processing of personal data by competent authorities for the purposes of the prevention, investigation, detection or prosecution of criminal offences or the execution of criminal penalties, and on the free movement of such data, and repealing council framework decision 2008/977/jha," *Official Journal of the European Union*, vol. L117, no. 2016/680, pp. 1–88, 2016.

[16] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[17] K. Toyoda, K. Machi, Y. Ohtake, and A. N. Zhang, "Function-level bottleneck analysis of private proof-of-authority ethereum blockchain," *IEEE Access*, vol. 8, pp. 141 611–141 621, 2020.