Providing Tamper-Resistant Audit Trails with Distributed Ledger based Solutions for Forensics of IoT Systems using Cloud Resources

Magnus Westerlund
Department of Business Management and Analytics
Arcada University of Applied Sciences
Jan-Magnus Janssons plats 1, 00550 Helsinki, Finland
magnus.westerlund@arcada.fi

Mats Neovius
Faculty of Science and Engineering
Åbo Akademi University
Axelia, Piispankatu 8, 20500 Turku, Finland
mneovius@abo.fi

Göran Pulkkis
Department of Business Management and Analytics
Arcada University of Applied Sciences
Jan-Magnus Janssons plats 1, 00550 Helsinki, Finland
goran.pulkkis@arcada.fi

***Abstract*—Network and information security are often more challenging for current IoT systems than for traditional networks. Cloud computing resources used by most IoT systems are publicly accessible and thereby, through this availability, increase the risk of intrusion. The increase in the processing of sensitive data in IoT systems makes security challenges more noteworthy, particularly in light of legal issues around cross-border transfers and data protection. Technologies preventing intrusion are effective, yet not perfect. Once a system is compromised, the intruder may start to delete and to modify audit trails and system log files for covering-up the intrusion. Complete and untampered audit trails and log files are essential for the legitimate owner of an IoT system using cloud resources to estimate the losses, to reconstruct the data, to detect the origin of the intrusion attack, and eventually in a court of law be able to prosecute the attacker. Due to this, improved methods for performing forensics in IoT systems are desperately needed. IoT forensics is mostly cloud forensics, since most IoT data is currently stored in the cloud. Therefore, cloud forensics is a key component in IoT forensics. The baseline for any forensic investigation is assured data availability and integrity. In this paper, we outline how forensic evidence data can be created for IoT systems using distributed cloud resources and how the availability and integrity of this forensic data can be assured by applying distributed ledger based solutions for storing audit trails and log files securely. Given this approach, an attacker can neither delete, nor modify past trails or logs but merely stop generating new data into log files. The approach presented here is novel, yet light enough for practical use.**

Keywords-forensics; IoT; cloud computing; distributed ledger; blockchain; distributed clouds; security; computer forensics.

I. INTRODUCTION

This paper outlines a distributed ledger approach for storing the audit trail data of IoT systems using distributed cloud resources. It extends its original conference paper [1] by an elaborated outline of audit trail creation, accountability principles for IoT service providers, and a discussion. For a definition and elaboration of the distributed cloud, we direct interested readers to Westerlund and Kratzke [2].

Academic research in network and computer forensics has a long history. A systematic literature review about digital forensics investigation is presented by Alharbi et al. [3]. In this review, a forensic investigation has a proactive and a reactive phase. The proactive phase consists of

- collection of pre-defined data according to priority and volatility,
- setting of a triggering function for hypothetical suspicious events,
- preservation of data related to suspicious events, and
- preliminary analysis of data and preliminary reporting related to the adopted hypothesis about suspicious events.

The reactive phase is triggered by a suspicious event. It consists of identifying, preserving, collecting, and analysing evidence data and generation of a final report. The collected evidence data is active and passive. Active evidence data is live or dynamic evidence that exists just after a detected suspicious event, for example processes running in a computing device. Reactive evidence data is static, for example a hard drive image.

Forensic investigations can be counter-acted by anti-forensics methods which try to [4]

- prevent collection of evidence data,
- increase the time of forensic investigations,
- create misleading evidence for forensic investigations, and
- prevent digital crimes from detection.

Evidence data for forensic investigations needs therefore protection. This was considered already by Schneier and Kelsey [5] who suggest a solution for keeping an audit log on insecure servers by offering a tamper-proof forensic scheme that stored and maintained log entries. However,

with the emerging Internet of Things (IoT) technology and the shift to cloud computing, the complexity and importance of keeping a secure audit trail have drastically increased. The building blocks of an IoT device is defined to contain an entity with an energy source and a processing module which has a storage module and interfaces for sensing, actuation, and communication [6].

To secure every IoT system is an utmost challenge. Currently, embedded security solutions, middleware, and cloud security solutions are being developed for IoT security. The goal of these efforts is detection of security threats and prevention of security attacks. No single solution is hitherto known for protection of IoT systems against all types of security attacks. The forensics discussed in this paper address the means of verifiable logs for carrying evidence of source and means as well as for restoring the compromised system to a working state. IoT forensics is defined by Zawoad and Hasan [7] as one of the digital forensic branches where the main investigation process must suit the IoT infrastructure. IoT forensics has therefore a key role in its part to investigate security breaches found in the IoT infrastructure. IoT forensics is a way to reconstruct the sequential steps performed by the attacker during the attack process; providing valuable information in constructing ever more secure systems. The sequential steps are identified by collecting data from different sources such as devices, logs, applications and networks used at the time of attack.

The paper's layout is as follows: in the following section, we discuss the motivation for accountable IoT service providers. Section III provides an overview of how audit trails for IoT forensics can be obtained, the role of cloud forensics, and some case studies. Section IV presents distributed ledger-based solutions of blockchain type for tamper-resistant protected storage of audit trails. The use of distributed ledger technology (DLT) is discussed in Section V. Finally, conclusions and proposals for future work are presented in Section VI. DLT is briefly described in an Appendix with the emphasis on the blockchain.

II. ACCOUNTABILITY FOR IOT SERVICE PROVIDERS

A motivation for a shift in how organizations prioritize resource allocation and consequently the importance of how system security is perceived, has been provided by the introduction of the EU General Data Protection Regulation (GDPR) [8]. As the GDPR has a long reaching implication for service providers anywhere in the world, as long residents of the EU may use such a service, it means that the GDPR has effectively set a default and minimum requirement for such systems that handle personal data on a global scale [9]. The GDPR provides rather strict guidelines for data security, but it also requires appropriate system security so that data does not seep into unauthorized use. Duncan [10] highlights that the 72h rule for reporting security incidents to appropriate parties would have been more effective if the rule had been formulated as "after they occur", opposed to the finalized wording of the GDPR "after they are detected". Still, the accountability principles requires a company after they become aware of a security breach to inform whom this breach includes and what particular personal data has been compromised.

The accountability principles are based on several measures that a company can take to achieve compliance with the GDPR. A core principle to achieve such compliance is to adopt and implement data protection policies. For IT-systems this refers to both the development process of IT-systems and to the maintenance processes. Any changes to a system that handles personal data (data that directly or indirectly identifies a natural person) over the system lifetime must comply with this principle continuously over time. Through such an approach we can consider that data protection is by design and default. For legacy systems that have not been designed with data protection as default, it may become difficult to show that a new version of the same system has incorporated data protection by design. For distributed IoT-systems this will likely become an even bigger challenge to show using conventional methods such as using centralized logs for collection of forensic data.

The GDPR also requires that organizations define through contract such processing that is performed by a third party with the controller's permission. The controller is also obligated to maintain the original consent contract given by the data subject (owner of said personal data). Documentation is also required of any activities the controller takes in processing personal data. This may mean the storing of facial images obtained from cameras in an IoT-network, processing said images for the purpose of identifying faces, and may in some cases mean the intended future use of any derivative products from such processing. The ability for an IoT service provider to define transactional records on a granularity of an individual user will likely become necessary. As earlier mentioned for storing forensic data, using centralized storage to achieve compliance for documentation of processing and consent may become difficult. In designing a distributed IoT-network and to maintain centralized provisions for such collection efforts will not necessarily be enough. Rather a distributed transaction database, with an immutable ledger that is not susceptible to common network attacks such as Denial of Service (DoS) would be much preferable.

The accountability principles also include organizational measures that need to be taken into account. Such measures include performing data protection impact assessments for detecting solutions with high risk to data subjects' privacy. A recommended (and in certain cases required) approach is that this work is led by an independent data protection officer, with a mandate to object the development or use of particularly dangerous practices or solutions. Organizations that develop a privacy management framework and continuously follow it within all processes involving the processing of personal data, may be considered accountable and can apply for a certification scheme that should indicate a notion of trust to potential users.

For distributed technologies this may be more challenging than for centralized, because once software is deployed to the distributed nodes the service provider may lose control of said software. Due to this nature of distributed software a recommended approach is to automate both data

security and appropriate system security measures. This includes previously stated accountability principles, incl. future security updates of the complete system. In the following section we discuss in-depth the use of IoT forensics and the creation of audit trails, to better understand how to continuously monitor delivered systems.

We should also note that United States currently provides some cybersecurity provisions that requires any contractor providing Internet connected devices to US federal government to also provide written certification that the device:

- does not contain any hardware, software, or firmware component with any known security vulnerabilities or defects (some exceptions exist),
- relies on software or firmware components capable of accepting properly authenticated and trusted updates from the vendor,
- uses only non-deprecated industry-standard protocols and technologies for functions such as communication, encryption, and intercommunication with other devices, and
- does not include any fixed or hard-coded credentials used for remote administration, the delivery of updates, or communication [11].

These provisions require contractors while under contract to notify purchasing party of security vulnerabilities, to maintain software that can be updated, and to provide timely updates.

III. IoT Forensics And Audit Trails

The ability to perform forensic activities in an IoT infrastructure is a challenging task. The existence of audit trails that can be reviewed is often a missing component. Still, as has been shown for cloud computing, detecting misuse is often dependent on the ability to scan various types of logs, both on system and application level.

The creation of an audit trail for forensic investigations of IoT systems is affected by the differences between IoT forensics and traditional digital forensics. Following differences are listed by Oriwoh et al. [12]:

- Evidence sources include IoT devices such as dish washers, pressing irons, refrigerators and wearable devices.
- The number of devices for evidence retrieval is much larger since there can be thousands of devices in an IoT system.
- The quantity of evidence data is much larger and the evidence format is different because of the multitude of different devices in an IoT system.
- The location of evidence data is much more distributed including multiple IoT devices and evidence related to IoT data stored in cloud resources implemented by micro-services.
- Flexible boundary lines between networks with connected devices from which evidence data is retrieved, since a Body Area Network with connected wearable device moves with the related person between different connection networks.

The audit trail for forensic investigations of IoT systems consists of evidence sources which are categorized in related research [12] [13] [14] as

1. Evidence collected from IoT devices and sensors
2. Evidence collected from wired, wireless, and mobile network communication between IoT devices and the external world
3. Evidence collected from network perimeter devices such as firewalls, AAA servers, NAT servers, and Intrusion Detection Systems (IDS)
4. Evidence collected from hardware and software outside the network under investigation. This category includes cloud, web, social networks, ISPs and mobile network providers.

Based on this classification a 1-2-3 Zones approach to IoT forensics is proposed in [12]. Zone 1 uses evidence of category 1, Zone 2 uses evidence of categories 2 and 3, and Zone 3 uses evidence of category 4.

A proactive and reactive phase are outlined by Zulkipli et al. [15] for the creation of an audit trail for forensic investigations of IoT systems. The proactive phase is a pre-investigation phase for preparation of the IoT forensic readiness. The reactive phase a real-time phase triggered by a detected security incident. The IoT forensic readiness is divided into management readiness and technical readiness. Management readiness includes

- an investigation plan for handling an incident,
- preparation of tools, techniques, and operations to support the investigation,
- monitoring the IoT system and obtaining support for authorization, and
- preparation of investigation skills of the investigators

For technical readiness is needed a scoping plan which defines the knowledge requirements of the investigators:

- What should be identified?
- What data should be collected?
- How should the potential evidence be identified?
- How should the potential evidence be collected?
- How should the collected evidence be preserved?

In the real-time phase tree concurrent tasks are started: scanning and identification, collection, and preservation. The scanning and identification task registers IP and MAC addresses, network port numbers, URLs, and data packet sizes. The collection task collects logs, history activity traces, time stamps, and user names with related passwords. The preservation task triggers snapshots of IoT device memories, creates hashes and encryptions of the collected data and the snapshots, and sends the hashes and encryptions to a secure storage.

Models for IoT forensics audit trail creation are proposed in [7] [16] [17] [18]. These models are described in a subsection.

IoT forensics after security breaches on data integrity, confidentiality, and availability is mostly cloud forensics, since most IoT data is already being stored or will be stored in the cloud. Therefore, cloud forensics is a key component in IoT forensics and also the most challenging component in

IoT forensics in the creation of a secure audit trail for forensic investigations [14].

A. IoT Forensics Models for Audit Trail Creation.

Zawoad and Hasan proposed a conceptual model of IoT forensics [7]. A secure Evidence Preservation Module monitors how all registered IoT devices store evidence data such as network logs registry logs, sensor data, etc. in a an evidence repository database. To ensure handling of a very large evidence dataset the Hadoop Distributed File System (HDFS) [19] is proposed to be used for the stored evidence data. The integrity and confidentiality of the stored evidence data is protected by public key cryptography. The private encryption key is accessible to forensic investigators for viewing the stored evidence data. A secure Provenance Module preserves the access history of the data stored in the evidence repository database in a provenance database. The Provenance Aware File System (PASS) [20] is used for the data stored in the provenance database. The Provenance Module applies secure provenance chaining [21] to protect the data stored in the provenance database against malicious tampering. Only forensic investigators can access a Representational State Transfer (REST) [22] based web API to the evidence repository and provenance databases. Using retrieved provenance records evidence data can be fetched.

An application-specific forensics investigative model in IoT is proposed by Zia et al. [16]. The model consists of three components: Application-Specific Forensics, Digital Forensics, and Forensics process. Unique application-specific forensics issues are handled by the Application-Specific Forensics module. The 10 most popular IoT applications are ranked from high to low popularity as Smart City, Connected Industry, Connected Building, Connected Car, Smart Energy, Other, Connected Health, Smart Supply Chain, Smart Agriculture, and Smart Retail [23]. Data is extracted from IoT devices and transferred to a network or to a cloud service. Thus the data flows to the Digital Forensics Module, which consists of 3 functions IoT Forensics, Network Forensics and Cloud Forensics. The functions create logs and store trends and logs of the data flow from the Application-Specific Forensics module. The Forensics Process collects evidence from the Digital Forensics module, examines and analyses the collected evidence and creates reports.

An IoT forensic investigation model based on a top-down forensic approach methodology is proposed by Perumal et al. [17]. If a forensic investigation should be planned, the investigator should obtain a warrant and authorization to access all necessary data. The investigation start with base device identification, which refers to device-to-device communication implemented by protocols such as 3G, 4G, LTE, Wi-Fi, Ethernet, and Power Line Communication (PLC). To locate a malicious medium that has communicated with an IoT device a triage examination is carried out to retrieve evidence data. This examination deals with platforms such as router, gateway, cloud, and fog. The investigation continues with identification of the chain of custody of retrieved evidence data, analysis of all data, and storage, presentation, and proof of analysis results.

An IoT forensics model called an IoT Digital Forensic Framework is proposed by Kebande and Ray [18]. The framework consists of three modules: a proactive process, IoT forensics, and a reactive process. The proactive process implements a pre-investigation phase in the creation of an audit trail for forensic investigations of IoT systems and the reactive process, which is triggered by a security incident, implements the real time phase [15]. The IoT forensics module consists of device level forensics, network forensics, and cloud forensics in correspondence with 1-2-3 Zones approach to IoT forensics [12].

B. Cloud Forensics and Audit Trails

The last decade has entailed a transition from onsite to cloud computing. Cloud computing provides access to a pool of interconnected resources enabled by the Internet. It abstracts the hardware from the client and has a “pay-per-use” business model. In cloud computing, the resources are elastically provisioned with storage space, service, computing platforms as virtual machines [24], and networking infrastructures obtained upon request [25] [26]. Hence, cloud computing is “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” [25]. Three basic cloud computing service models are Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Contemporary cloud-based software engineering directs towards Cloud Native Applications (CNA). A CNA is a service specifically designed to run in the cloud. CNAs are often deployed as self-contained units (containers) that are designed to scale horizontally. A CNA is often implemented as micro-services [27]. Kratzke and Quint [28] have described the technicalities in detail. In addition, the availability of cloud computing resources is augmented by the Intercloud initiative [29], envisioned as the “cloud of clouds”. Hence, the Intercloud then provides virtually unlimited resources to any connected device. In this paper, we refer to connected devices as all devices that are connected to the Internet. Such devices have given rise to the Mobile cloud computing [30] and Internet-of-Things (IoT) [31]. As a mobile device may utilise or contribute to the data mass, an IoT device frequently merely contributes to the cloud relying on the service provider in administering the security and privacy of the data.

Cloud forensics has been defined as “the application of digital forensics in cloud computing as a subset of network forensics” [32] and as “to reconstruct past cloud computing events through identification, collection, preservation, examination, interpretation and reporting of digital evidence” [33]. As the former definition suggests forensics to be restricted to the network access, the latter definition includes the audit trail as a means to reconstruct events, as well as interpretation and reporting of evidence. Cloud forensics, therefore, requires audit trails to be stored in a manner with assured availability and integrity where no changes may occur. Audit trails for cloud forensics consist of collected log

data of network traffic and data processing activities of computing devices. As such data is generated it is processed by an Intrusion Detection System (IDS) that extracts features from collected log data and analyses these. State of the art IDSs provide an active network security component using machine learning techniques to determine when anomalies occur and to detect intrusions in near real-time [34]. In a SaaS or FaaS (Function as a Service) setting the cloud service provider (CSP) has the sole ability to generate system wide IDS data. However, depending on the service model, the point of responsibility deviates. A framework for cloud forensics is proposed in [35], see Fig. 1.

Log data for audit trails can be scattered and stored in different locations due to the characteristics of the cloud. In the cloud, the level of access is divided between the cloud service user and the CSP. The level of access in the basic cloud service models is shown in Fig. 2. This significantly complicates the data acquisition process. For example in the SaaS and PaaS models, only application related logs can be accessed by the cloud service user. Though in PaaS, a cloud service user can develop an application to be able to get some additional forensics data whereas, in SaaS, this is not possible. In the IaaS model, cloud service users can move to the operating system layer for acquiring forensic data. In all service models, the forensic investigators are dependent on the CSP to ensure that needed audit trail data has been collected. This is currently thus a trust issue since the availability and integrity of the data that may be affected are not transparent. Only when both parties are fully contributing to an immutable audit trail can it provide the required transparency needed for continued investigation and legal measures.

Verifiable audit trails are essential in forensic investigations to reconstruct and rigorously examine intrusions in the cloud. The reconstruction is central to find out what damage the intrusion has caused and discover sources and origins of intrusion attacks. When an attack has occurred, the cloud service user must engage a cloud forensics investigation to analyse the audit trail related to the attacked service in order to find forensic evidence. For this,

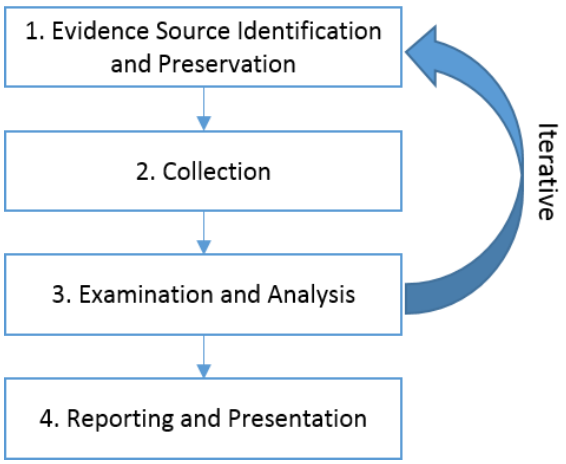

Figure 1. Cloud forensics framework proposal [35].

the audit trail is fundamental in meeting with the EU General Data Protection Regulation (GDPR), requiring enterprises to report security breaches within 72 hours after detection. Moreover, it should be possible for a CSP to present evidence on its own behalf that the source of the intrusion was external.

Traditionally, in digital forensics investigators take control of the affected physical device and perform forensic investigations on these by searching for evidence of malicious activity. As cloud computing is inherently dynamic, often the methods used traditionally in digital forensics render themselves impractical [36]. Different cloud service users may virtually share physical resources through the hypervisor and thus, to isolate the scene for forensics is next to impossible. This leads to issues that must be addressed by the forensic investigation, namely, it must be proven that any data extracted is not mixed with some other customer's data and that the availability, privacy, and integrity of the other user's data must be maintained.

Cloud forensics challenges are mostly related to architectural, data collection, and legal issues [33] [37], as well as in composing provenance data. Provenance data is the "metadata that provides details of the origins (history) of a data object" [38]. That is, provenance data is metadata tracing the history of data objects starting from original source data [39]. Complete provenance of all data stored in the cloud, all distributed computations, all data exchanges, and all transactions would enable identification of exact sources of cloud intrusion attacks and detect insider attacks in forensic investigations [40].

C. Case Studies for Reconstructing Forensic Data

Acquisition of forensic data from a network accessible smartwatch is outlined in [14] as an IoT device forensics case study. The studied smartwatch has several sensors (accelerometer, gyroscope, heart-rate sensor, and ambient light sensor), supports SMS messaging and email, can be paired with a smartphone and has following installed applications: Health App, Nike Plus App, Heartbeat App, Messages, and Maps App. Forensic data can be collected from a paired smartphone executing Cellebrite UFED forensic software [41] and by manual swipe through the smartwatch. Forensic investigators collect GPS data, heart-rate data, timestamps, MAC address, paired devices, text messages and emails, call log, contact data, etc.

The possibilities to carry out a forensic investigation on a smart TV are presented in [42]. Smart TV platforms converge traditional TV technology and computer technology and they have Internet connectivity. A smart TV device using a flash memory storage was chosen for collection and analysis of forensic data. The memory chip was removed from the motherboard of the smart TV and an image of the chip was created with the NFI Memory Toolkit II [43]. Elevated privileges, which are required for data extraction from the user space memory and for full access to the file system, were obtained for the flash memory image with a rooting procedure. Digital traces such as

- system settings: device name, connected devices, network information, and smart functions,

- use of apps: Facebook, Twitter, YouTube, etc.,
- use of web: visited web sites, search traces, etc.,
- image and multimedia files
- connected external devices: USB flash drive, hard disc, etc.,
- e-mail messages and appointments,
- use of cloud services: Dropbox, OneDrive, etc., and
- viewed TV channels

are forensically studied.

Extraction of forensic data from IoT devices in a Z-Wave [44] network is described in [45]. Z-Wave is a frequently used protocol stack in Home Area Network implementations. A typical Z-Wave network consists of controllers, sensors, and Z-Wave devices. A Z-Wave device is an IoT device (thermostat, light switch, smart locker, water valve, etc.) connected to a controller, which acts as a gateway between a Z-Wave network and Internet. Z-Wave devices can any time enter and leave a Z-Wave network. The controller assigns a unique Node ID to each Z-Wave device entering a Z-Wave network. Data extraction from a frequently used chipset with external EEPROM (Electrically Erasable Programmable Read Only Memory) on a motherboard of a Z-Wave device is described. Analysis of an event table in the EEPROM reveals which Z-wave devices worked during a specific timeframe.

IV. PROTECTION SOLUTIONS FOR AUDIT TRAIL DATA

Audit trail data for IoT system forensics requires secure protection against corruption by accidental faults and malicious forgery [46]. Protection must repel accidental corruption and all malicious anti-forensics attacks by ensuring both integrity and availability of the data.

A reasonable first choice for storage of audit trails for IoT forensics is an append-only (immutable) conventional database installation where read rights are assigned only to carefully selected set of agents. Existing implementations of immutable databases include configured conventional ones. In its most secure installation, it is hosted in-house with no means of external access and restricted physical access. Every access point (let these be logical or physical) weaken assurance of integrity. In-house installations are, however, not pragmatic for IoT systems using cloud resources; nor are the IoT systems remote installations. On this challenge, purpose-built databases and file systems are being developed, e.g., Datomic [47]. Implementation details of an immutable database for cloud audit trail are reported by Duncan and Whittington in [48].

Another attempt is the InterPlanetary File System (IPFS) [49]. The IPFS is fundamentally a protocol inspired by the Bitcoin blockchain protocol. It tries to make the web a digital resemblance to printed paper in documenting data, i.e., something that is permanent, unalterable and controllable. IPFS has a name service called InterPlanetary Name System (IPNS), which is a global namespace based on PKI [50]. IPNS serves to build trust chains and is compatible with other name services. The name services DNS, .onion, .bit, etc. can be mapped to IPNS.

The secure provenance scheme described in [21] encrypts sequences of new data, hashes the resulting datasets and provenance record, and digitally signs chains of hashed provenance records. Forensic auditors are offered access provenance data with their private keys in public key cryptography. The scheme ensures integrity and confidentiality against malicious disclosure and tampering attempts. Malicious deletion of data in the scheme is detected, but the consequence is inaccessibility to provenance data since there is no replication in the scheme.

A distributed and replicated append-only storage usually provides stronger tamper resistance than a centralized one. A distributed ledger is a replicated database, which is shared by nodes in a peer-to-peer network. Consensus algorithms are required to ensure replication and insertion across network nodes. In a truly distributed ledger, there is no central administrative node or centralized data storage. Therefore, it is considered in [51] [52] that a distributed ledger storage for audit trails typically has stronger tamper resistance than any centralized immutable database implementation.

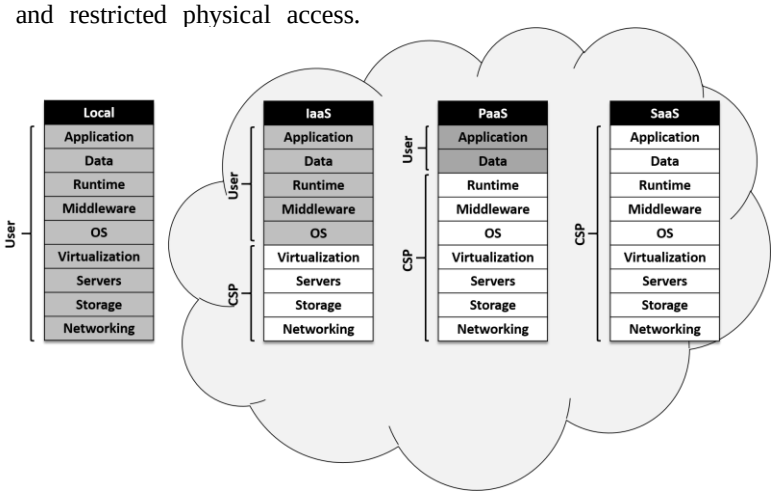

Figure 2. Access control to basic cloud service models in comparison to a local system.

The sub-sections discusses requirements for distributed ledger based solutions to protect audit trails for forensic investigations of IoT systems and presents some blockchain based solution proposals. In Section IV D, we present a novel architecture for automating and securing forensic data in distributed IoT networks. Distributed ledger technology (DLT) with the focus on blockchain technology is further described in an Appendix.

A. *Requirements for Distributed Ledger based Solutions*

In a traditional IoT architecture IoT devices are network nodes which transmit their payload data to a data store through some proxy or application programming interface. IoT device management is manual and potential device logs may often remain locally stored on the devices. Device users have credentials for authentication. Only authenticated users are authorized to access IoT devices and to update device firmware from device deliverers' databases. If a system log is stored on a respective node it would require device access for collection (pull) of data. Storage space is often very limited so only the most resent activities may be stored on the device, hence continuous collection to a centralised data store is required for ensured retention. An improved solution for a traditional architecture is presented in Fig. 3, i.e., automatically pushing log data from each node. From an accountability perspective new updates to the nodes must continuously be provided, something that often requires a manual process by a system administrator. New firmware security updates should also be provided by the manufacturer for the lifetime of said IoT devices. For this process to be complete, traditional IoT systems require many manual process steps that are often not possible to ensure in today's environment. Hence, we find it motivated to propose a new type of architecture better suited to a distributed network topology. Our proposal is presented in Section IV D.

Usage of a distributed ledger for protection of IoT forensics data is possible only if three fundamental requirements are fulfilled. First, a sufficiently large network of nodes must be available for storing replicated copies of the distributed ledger. Secondly, each network node must have sufficient storage and processing resources for management of a distributed ledger replication. Thirdly, it must be possible to extend the distributed ledger with devices producing new data at the data rate needed (i.e., throughput and scalability).

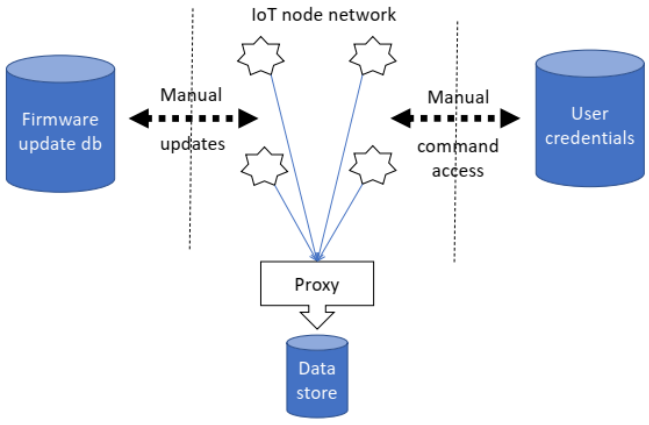

Figure 3. An improved traditional IoT architecture.

B. *Existing Distributed Ledger Based Solutions*

Applying the blockchain and distributed ledger technologies in various domains is currently a hot research and business development topic. These technologies have been proposed for many financial technology solutions with extensions assuring programmatic smart contracts, to preserve (and control) privacy and personal data, provide transparency on transactions, and in the industrial IoT to keep track of logistic chains. These are all very intriguing applications, but we concentrate on ones that are directly relevant to the distributed audit trail data. Further, we focus on forensic data in the cloud computing environment, since current IoT systems usually store generated data in the cloud and we consider this area to be among the most challenging problems for distributed ledgers.

The integrity of forensic data can be ensured by Public Key Infrastructure (PKI) signatures which depend on a certificate authority. This is not a feasible solution for IoT systems using distributed cloud resources since cloud infrastructure is inherently decentralized. An alternative to PKI signatures is keyless signatures implemented by a blockchain based distributed Keyless Signature Infrastructure [53] [54].

A blockchain based data provenance architecture, the ProvChain, is described and evaluated in [55]. ProvChain has been designed for collection and verification of cloud computing users' provenance data. ProvChain can use the global Bitcoin blockchain since the collected provenance data is restricted to metadata records of cloud service users' operations on data files stored in the cloud. Recorded metadata attributes are RecordID, Date and Time, UserID, Filename, AffectedUser, and FileOperation. A FileOperation is file creation, file modification, file copy, file share, or file delete. UserID attributes are hashed to protect cloud users' privacy. Provenance auditors can, therefore, access cloud users' provenance metadata but cannot correlate the metadata to users owning the metadata. Only the Cloud Service Provider (CSP) can relate provenance data to cloud service users owning the data. Provenance metadata records are published in blocks of a blockchain implemented by a blockchain network consisting of globally participating nodes. Several metadata records can be stored in one blockchain transaction. Each metadata record is extended with a hash and a Merkle hash tree [56] is constructed for the metadata records in a block. The Merkle root is stored as a block header attribute. ProvChain is built on the top of the open source cloud computing application ownCloud [57]. The Tierion Data API [58], is used to publish provenance metadata records in the blockchain. Tierion generates for each transaction a blockchain receipt based on the Chainpoint standard [59]. The Merkle hash tree included in this blockchain receipt proves that the provenance metadata records were recorded at a specific time. A provenance auditor can request a blockchain receipt via Tierion Data API, access the related blockchain block with Blockchain

Explorer [60], and validate the provenance metadata records in the block with the Merkle hash tree in the receipt. Measured ProvChain overhead for retrieval of provenance metadata of one file operation is about 0.7…0.8 s in an ownCloud test application [55].

Blockchain-based tamper-resistant registration of provenance data related to accessing medical data records in cloud storage is outlined in [61] [62]. The provenance data stored in the blockchain is available for auditing and in forensic investigations to detect privacy violations of medical data record owners. The outlined solution for protection of provenance data is applicable also to other types of personal data records.

C. Various Proposals for Distributed Ledger based Solutions

An ideal solution would be a global network of nodes fulfilling all three requirements in Section IV A. The global Bitcoin blockchain fulfils the two first requirements, but this blockchain cannot be extended with new blocks at a rate needed. Computationally it is not possible that even for a small cloud computing environment all the audit trail data for forensic investigations would be stored in the Bitcoin blockchain. The reason is the current blockchain size in combination with the throughput constrained Proof-of-Work (PoW) consensus algorithm.

However, other possible solutions may be engineered that circumvent this issue. One possible solution is a network of distributed ledger nodes, for example, blockchain nodes maintained by a CSP or preferably by several cooperating CSPs. As of the second requirement in Section IV A, all cloud computing users cannot be nodes in a distributed ledger network since also resource-constrained mobile devices and IoT devices can use cloud computing services. Moreover, a faster consensus algorithm than PoW must be implemented for the used distributed ledger.

Hashgraph is a DLT with a Byzantine consensus algorithm using a gossip protocol [63] [64]. While Bitcoins PoW implementation limits the throughput 7 transaction/s, the Hashgraph consensus algorithm can process even tens of thousands transactions/s [65]. The Archive Database proposed in [48] to be used as an immutable database for cloud audit trails could be implemented by a network of Hashgraph nodes maintained by a CSP or several cooperating CSPs. Each time when the database audit trail plugin stores log data the same data is transmitted to a preferably randomly chosen Hashgraph node. Reception of the log data creates a signed time-stamped event including a transaction storing the log data. An immutable record of all stored events is - due to the high event processing rate of a Hashgraph network – almost immediately available in each Hashgraph node. The Hashgraph fulfils all requirements in Section IV A. However, at the time of writing it is deployed in permissioned environments and is, therefore, a permissioned DLT. Still, a federated decentralized installation maintained by several cooperating CSPs or other service providers may offer an alternative to a public distributed ledger.

There are also other proposals that address the need for high throughput distributed ledgers. Off-chain state agreement solutions commonly referred to as state channel technology, have been developed for handling many small transactions. A use case for the development of state channel technology has been to handle micro-transactions, which in addition to needing a high throughput also require a minuscule transaction cost for the clearance of each transaction [66]. Other solutions propose to split the processing and recording of transactions into sub-chains, a technology often referred to as sharding [2].

D. Distributed IoT Architecture Proposal

In our distributed IoT architecture proposal, which is shown in Fig. 4, IoT nodes transmit their data to a distributed and replicated data store. The data store is run outside the limited nodes and utilise a Peer-to-Peer (P2P) protocol. Various data stores can be utilised that depending on requirements, such as scalability, speed, or post-processing, can be used. A suitable solution may be IPFS, a proprietary P2P data transfer protocol, or a data and analytics marketplace such as Streamr [67]. Smart contracts executed on top of a DLT implementation may authorize IoT devices and may further offer device management, e.g., issuing management commands. IoT device firmware updates may be automatized in a similar fashion. Storing the latest version of a binary update file in IPFS, and in a smart contract store an IPNS static address that allows node to query correct IPFS file and the firmware signature to confirm file integrity. This tells the IoT node how to access IPFS files and how to perform verification of the needed update.

We also consider the possibility important that a manufacturer may want to offer a service contract to any IoT node maintainer (owner). Currently, a significant problem is that IoT nodes are not provided with long-term support as the manufacturer often fails to get financial compensation for updating firmware once the product enters a maintenance/archival phase. This business model could however be implemented through a smart contract, that provides the manufacturer with a decentralised platform for selling firmware updates. An automated update function and contract resolution can be provided to any IoT node maintainer, either on a node basis (number of nodes) or on a network basis (maintaining organisation).

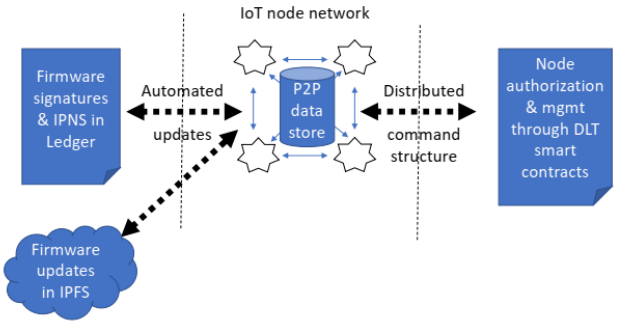

Figure 4. Proposed distributed IoT architecture.

V. DISCUSSION

A challenge for the field is that distributed ledger technology lacks a formal definition and standardisation. This may be due to the fact that it is an ensemble of technologies that in combination offer a mechanism for chaining blocks of records together. This holds the key for its disruptiveness, where centralised management of a system is impossible and the system starts living a life of its own with the help of computing resources allocated to it from any participant. A contemporary impact can be found in the financial industry due to an application, crypto currencies, where the traditionally regulated industry is disconnecting from the central governance of central banks. However, at the core DLT offer basic functionality for

- trustless interaction between two/more parties,
- third-party validation of transactions,
- distributed storage of transactions,
- some DLTs may offer a contract resolution mechanism through smart contracts.

Without a centralised authority, authenticating and validating the data is ever more important. This is fundamental for forensic evidence to hold up in a court of law. For this, DLT provides court-level forensics. The technology is developed in the wake of the financial industry with an obvious application domain being the IoT technology as this is, or will become, ubiquitous.

In the financial industry, the transitioning into cloud computing has inflicted a minimal transformation on the operational side, i.e., the cloud system do serve the end user as did centralised ones, but now in a manner scaling virtually infinitely. Yet, a cloud system runs the same databases, use storage space and encryption in the same way as would be done in a centralised system. Hence, the distributed ledger technology may enable, at time of writing, mainly for transaction storage space, independence from a centralised point of administration. Such an approach would obviously require a shared will among its participants. Comparing this with the financial industry, it may enable the creation of a global sharing economy of commodity swapping. This transformation would truly be disruptive on the global scale. Therefore, we believe DLT holds vast potential in catalysing new solutions and solving problems in existing applications.

VI. CONCLUSIONS

This paper outlines approaches for creation of audit trails from IoT systems using distributed cloud resources and for applying distributed ledger based solutions to securely store these audit trails. The security features of the distributed ledger assure the integrity of the audit trails which is essential for trustable IoT system forensics. The challenge is timely as the EU GDPR became enforced from May 2018. Moreover, the recent advancements in distributed ledgers, blockchains (cryptocurrencies) and their various spinoffs set the scene for applying this new technology by novel means. Implementation of hitherto proposed distributed ledger based solutions for protection of forensic audit trails of IoT systems using cloud resources is an important area of future research and development work. This paper lays the ground for future research into distributed ledger technology in terms of IoT system forensics.

ACKNOWLEDGMENT

This research was partly funded by the Fund for Technical Education and Research (TUF) of The Arcada Foundation and Academy of Finland project LibDat, decision number 309495.

APPENDIX

A. Distributed Ledger Technology

The most deployed distributed ledger type is a blockchain, which extends the shared database with a sequence of blocks storing transactional data. Blocks are chronologically and cryptographically linked to each another. Other distributed ledger types are the Tangle Network and Hashgraph. For the Tangle network, a Directed Acyclic graph-based network is used instead of a replicated linked chain of blocks in blockchain network nodes [68].

A Hashgraph network consists of nodes, which create context dependent events and communicate with each other using a gossip protocol. An event is a timestamped and digitally signed data structure consisting of one or several transactions and two hashes. One hash is extracted from the latest event on the node from which the latest gossip was received and the other hash is extracted from the preceding event created on the same node. A created event is sent as gossip to another randomly selected Hashgraph node together with all events still not known by the selected node. As event creation and gossip transmission continue in all Hashgraph nodes, all created events are immutably stored in each Hashgraph node. A Byzantine consensus on the order of events is achieved with probability 1 using a virtual voting procedure if more than 2n/3 nodes are uncorrupt where n is the number of nodes in the Hashgraph network. The details of the gossip protocol, the virtual voting, and the Byzantine consensus algorithm are presented in [69] and [64].

The blockchain technology is at the time of writing the best-known solution for implementing distributed ledgers and we, therefore, choose to focus on it. Findings concerning distributed ledgers, in general, should be transferable to other solutions such as the hashgraph and the Tangle network, once they become widely validated as secure.

Blockchain technology was introduced in 2008 as the Bitcoin cryptocurrency platform [70]. A blockchain implements a distributed database where a list of records called blocks is stored. New blocks can always be appended to the list but stored blocks are neither removed nor changed. The distributed database is replicated in nodes of a peer-to-peer blockchain network. A complete database copy is therefore stored in each node. The blockchain topology is a chain, since after the first block each additional block contains a hash link to the preceding block, see Fig. 5. The first block is called Genesis Block. Each block is also time stamped, however not necessarily to a universal time server.

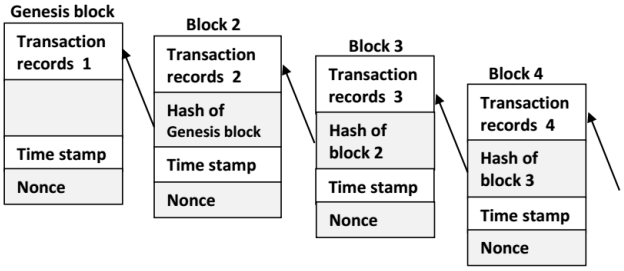

Figure 5. Basic blockchain structure.

A blockchain network node is owned by a blockchain user for execution of blockchain operations. A unique key pair of public key cryptography must also be owned by a blockchain user. The public key represents the identity of a blockchain user. A blockchain user executes a blockchain operation by initiating a transaction, which transfers some asset, for example, a cryptocurrency amount or a data object, to another blockchain user. A transaction creates a record, which is signed by the initiator of the transaction and transmitted to all nodes in the blockchain network. Each blockchain network node tries to validate a received transaction record with the transaction initiator's public key. A transaction record, which does not become validated by all blockchain network nodes, is discarded as invalid. Validated transaction records are collected by so-called mining nodes in the blockchain network and stored as lists in candidate blocks, which are time stamped. Each mining node executes a computation called mining on its candidate block. The candidate block of the mining node which first achieves a predefined mining goal is linked to the blockchain and all other mining nodes' candidate blocks are discarded. Several mining implementations for blockchains exist. Bitcoin blockchain mining uses PoW, where each mining node repeats hashing the concatenation of the last block in the blockchain and a new randomly chosen value. The mining goal is to create a hash of required difficulty.

There are public, permissioned, and private blockchains. A public blockchain, for example, Bitcoin, can be used by anyone. A public blockchain user copies the entire blockchain and installs the blockchain software on a personal node, which joins the blockchain network. Any blockchain user can also install the mining software on their own blockchain network node. Only a public blockchain can be trusted to fulfil the distributed ledger definition, as permission and private blockchains often maintain a centralized control node.

Recent blockchain implementations with extended functionality are denoted as Blockchain 2.0 for which an interesting feature is the smart contract introduced in [71]. A smart contract is a software component encompassing contractual terms and conditions enabling the verification, negotiation, or enforcement of a contract. A blockchain platform supporting smart contracts is Ethereum [72].

Blockchain security relies on the hash links between successive blocks combined with the replication of the entire blockchain to all blockchain network nodes. A public blockchain is therefore practically tamper-proof because a block cannot be changed without changing all the subsequent blocks and participation of all blockchain network nodes to validate and register the change. As the public blockchain is not managed by any centralized authority that could be a target of attacks it is less sensitive to some attack types such as DOS attacks, because full blockchain replicas are stored in many blockchain network nodes. However, an intrusion into a sufficient number of blockchain network nodes including some mining nodes can cause data losses and/or insertion of corrupt data in the attacked blockchain [73].

The tamper resistance of a blockchain does not exclude security vulnerabilities. Security attacks against blockchains are described and evaluated in [74] [75] [76] [77].

REFERENCES

[1] M. Neovius, M. Westerlund, J. Karlsson, and G. Pulkkis, "Providing Tamper-Resistant Audit Trails for Cloud Forensics with Distributed Ledger based Solutions," Proc. International Conference on Cloud Computing, IARIA, Feb. 2018, pp. 19-24, ISSN: 2308-4294, ISBN: 978-1-61208-607-1

[2] M. Westerlund and N. Kratzke, "Towards Distributed Clouds," Proc. 16th International Conference on High Performance Computing & Simulation (HPCS), IEEE Press, July 2018, pp. 655-663, doi:10.1109/HPCS.2018.00108.

[3] S. Alharbi, J. Weber-Jahnke, and I. Traore, "The Proactive and Reactive Digital Forensics Investigation Process: A Systematic Literature Review," International Journal of Security and Its Applications, vol. 5, no. 4, pp. 59-72, Oct. 2011.

[4] S. Garfinkel, "Anti-Forensics: Techniques, Detection and Countermeasures," Proc. 2nd International Conference on Information Warfare and Security, Mar. 2007, pp. 77-84.

[5] B. Schneier, and J. Kelsey, "Secure audit logs to support computer forensics," ACM Transactions on Information and System Security, vol. 2, iss. 2, pp. 159-176, May 1999, doi:10.1145/317087.317089.

[6] M. Aigner, "Security in the Internet of Things," in Cryptology and Information Security Series, vol. 4, Y. Li and J. Zhou, Eds. Amsterdam: IOS Press, pp. 109–124, 2010.

[7] S. Zawoad and R. Hasan, "FAIoT: Towards Building a Forensics Aware Eco System for the Internet of Things," Proc. 2015 IEEE International Conference on Services Computing (SCC 2015), IEEE Press, Aug. 2015, pp. 279–284, doi:10.1109/SCC.2015.46.

[8] EUR-Lex Regulation [EU] 2016/679. *General Data Protection Regulation (GDPR)*. [Online]. Available from: http://eur-lex.europa.eu/eli/reg/2016/679/oj 2018.11.26

[9] M. Westerlund, "A study of EU data protection regulation and appropriate security for digital services and platforms," Doctoral Dissertation, Åbo Akademi University, Åbo, Finland, 2018. [Online]. Available from: http://urn.fi/URN:ISBN:978-952-12-3694-5 2018.11.26

[10] B. Duncan, "Can EU General Data Protection Regulation Compliance be Achieved When Using Cloud Computing?," Proc. Ninth International Conference on Cloud Computing, GRIDs, and Virtualization, IARIA, Feb. 2018, pp. 1–6, ISSN: 2308-4294, ISBN: 978-1-61208-607-1

[11] S.1691 - Internet of Things (IoT) Cybersecurity Improvement Act of 2017. Bill, Senate - Homeland Security and Governmental Affairs, USA, 2017. [Online]. Available from https://www.congress.gov/bill/115th-congress/senate-bill/1691 2018.11.26

[12] E. Oriwoh, D. Jazani, G. Epiphaniou, and P. Sant, "Internet of Things Forensics: Challenges and Approaches," Proc. 9th

IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, IEEE Press, Oct. 2013, pp. 608-615, doi:10.4108/icst.collaboratecom.2013.254159.

[13] U. Salama, "Smart Forensics for the Internet of Things (IoT)," 2017. [Online]. Available from: https://securityintelligence.com/smart-forensics-for-the-internet-of-things-iot 2018.11.26

[14] S. Alabdulsalam, K. Schaefer, T. Kechadi, and N.-A. LeKhac, "Internet of Things Forensics: Challenge and Case Study," arXiv:1801.10391v1 [cs.CR], 2018. [Online]. Available from: https://arxiv.org/abs/1801.10391 2018.11.26

[15] N. Zulkipli, A. Alenezi, and G. Wills,"IoT Forensic: Bridging the Challenges in Digital Forensic and the Internet of Things," Proc. 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS), vol. 1, SciTePress, 2017, pp. 315-324, doi:10.5220/0006308703150324.

[16] T. Zia, P. Liu, and W. Han, "Application-Specific Digital Forensics Investigative Model in Internet of Things (IoT)," Proc. 12th International Conference on Availability, Reliability and Security (ARES'17), ACM Press, 2017, pp. 55.1-55.7, doi:10.1145/3098954.3104052.

[17] S. Perumal, N. Norwawi, and V. Raman, "Internet of Things (IoT) Digital Forensic Investigation Model: Top-down Forensic Approach Methodology," Proc. 5th International Conference on Digital Information Processing and Communications (ICDIPC), IEEE Press, Nov. 2015, pp. 19-23, doi:10.1109/ICDIPC.2015.7323000.

[18] V. R. Kebande and I. Ray, "A Generic Digital Forensic Investigation Framework for Internet of Things (IoT)," Proc. 4th International Conference on Future Internet of Things and Cloud (FiCloud), IEEE Press, 2016, pp. 356-362, doi:10.1109/FiCloud.2016.57.

[19] HDFS Architecture Guide. 2013. [Online]. Available from: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html 2018.11.26

[20] K. Muniswamy-Reddy, D. Holland, U. Braun, and M. Seltzer, "Provenance-aware storage systems," Proc. 2006 USENIX Annual Technical Conference, USENIX Association, 2006, pp. 43–56.

[21] R. Hasan, R. Sion, and M. Winslett, "The case of the fake Picasso: Preventing history forgery with secure provenance," Proc. 7th USENIX Conference on File and Storage Technologies (FAST´09), USENIX Association, 2009, pp. 1–12.

[22] R. H. Fielding. "Architectural Styles and the Design of Network-based Software Architectures," Doctoral Dissertation, University of California, Irvine, USA, 2000. [Online]. Available from: https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm 2018.11.26

[23] The Top 10 IoT Segments in 2018 – based on 1,600 real IoT projects, IoT Analytics, 2018. [Online]. Available from https://iot-analytics.com/top-10-iot-segments-2018-real-iot-projects/ 2018.11.26

[24] D. A. B. Fernandes, L. F. B. Soares, J. V. Gomes, M. M. Freire, and P. R. M. Inácio, "Security issues in cloud environments: a survey," International Journal of Information Security, vol. 13, iss. 2, pp. 113-170, Apr. 2014, doi:10.1007/s10207-013-0208-7.

[25] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," Special Publication 800-145, National Institute of Standards and Technology, U.S. Dept. Commerce, 2011. [Online]. Available from: https://csrc.nist.gov/publications/detail/sp/800-145/final 2018.11.26

[26] J. Köhler, K. Jünemann, and H. Hartenstein, "Confidential database-as-a-service approaches: taxonomy and survey," J. Cloud Computing: Advances, Systems and Applications, vol. 4, no. 1, 2015, doi:10.1186/s13677-014-0025-1.

[27] N. Dragoni, S. Giallorenzo, A. L. Lafuente, M. Mazzara, F. Montesi, R. Mustafin, and L. Safina, "Microservices: yesterday, today, and tomorrow," April 2017. [Online]. Available from: https://arxiv.org/pdf/1606.04036.pdf 2018.11.26

[28] N. Kratzke and P.-C. Quint, "Understanding cloud-native applications after 10 years of cloud computing - A systematic mapping study," J. Systems and Software, vol. 126, pp. 1-16, April 2017, doi:10.1016/j.jss.2017.01.001.

[29] D. Bernstein, E. Ludvigson, K. Sankar, S Diamond, and M. Morrow, "Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability," Proc. Fourth International Conference on Internet and Web Applications and Services (ICIW'09), IEEE Press, June 2009, pp.328-336, doi:10.1109/ICIW.2009.55.

[30] M. Shiraz, A. Gani, R. H. Khokhar, and R. Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," IEEE Communications Surveys and Tutorials, vol. 15, no. 3, pp. 1294–1313, 2013, doi:10.1109/SURV.2012.111412.00045.

[31] L. Jiang et al., "An IoT-Oriented Data Storage Framework in Cloud Computing Platform," IEEE Transactions on Industrial Informatics, vol. 10, no. 2, pp. 1443-1451, May 2014, doi:10.1109/TII.2014.2306384.

[32] K. Ruan, J. Carthy, T. Kechadi, and M. Crosbie, "Cloud Forensics," in Advances in Digital Forensics VII. DigitalForensics 2011. IFIP Advances in Information and Communication Technology, vol 361, G. Peterson and S. Shenoi, Eds. Berlin, Heidelberg: Springer, pp. 35–46, 2011.

[33] NIST Cloud Computing Forensic Science Challenges, Draft NISTIR 8006, National Institute of Standards and Technology, U.S. Department of Commerce, June 2014. [Online]. Available from: https://csrc.nist.gov/csrc/media/publications/nistir/8006/draft/documents/draft_nistir_8006.pdf 2018.11.26

[34] K. Grahn, M. Westerlund, and G. Pulkkis, "Analytics for Network Security: A Survey and Taxonomy," in Information Fusion for Cyber-Security Analytics. Studies in Computational Intelligence, vol. 691, I. Alsmadi, G. Karabatis, and A. Aleroud, Eds. Springer, Cham, pp. 175-193, 2017.

[35] B. Martini and K.-K. R. Choo, "An integrated conceptual digital forensic framework for cloud computing," Digital Investigation, vol. 9, iss. 2, pp. 71–80, Nov. 2012, doi:10.1016/j.diin.2012.07.001.

[36] V. M. Katilu, V. N. L. Franqueira, and O. Angelopoulou, "Challenges of Data Provenance for Cloud Forensic Investigations," Proc. 10th Int. Conf. on Availability, Reliability and Security, IEEE Press, Aug. 2015, pp. 312-317, doi:10.1109/ARES.2015.54.

[37] M. E. Alex and R. Kishore, "Forensics Framework for Cloud computing," J. Computers and Electrical Engineering, vol. 60, iss. C, pp. 193-205, May 2017, doi:10.1016/j.compeleceng.2017.02.006.

[38] K.-K. Muniswamy-Reddy and M. Seltzer, "Provenance as first class cloud data," ACM SIGOPS Operating Systems Review, vol. 43, no. 4, pp. 11-16, Jan. 2009, doi:10.1145/1713254.1713258.

[39] Y. L. Simmhan, B. Plale, and D. Gannon, "A survey of data provenance in e-science," ACM Sigmod Record, vol. 34, no. 3, pp. 31–36, Sep. 2005, doi:10.1145/1084805.1084812.

[40] D. K. Tosh et al., "Security Implications of Blockchain Cloud with Analysis of Block Withholding Attack," Proc. 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, IEEE Press, May 2017, pp. 458-467, doi:10.1109/CCGRID.2017.111.

[41] Extract & decode, 2018. [Online]. Available from: https://www.cellebrite.com/en/product/solutions/extract-decode/ 2018.11.26

[42] A. Boztas, A. R. J. Riethoven, and M. Roeloffs, “Smart TV forensics: Digital traces on televisions,” Digital Investigation, vol 12, supp. 1, pp. S72-S80, Mar. 2015, doi:10.1016/j.diin.2015.01.012.

[43] The NFI Memory Toolkit II. Netherlands Forensic Institute, 2011. [Online]. Available from: https://www.forensicinstitute.nl/documents/publications/2017/03/06/brochure-memory-toolkit 2018.11.26

[44] Z-Wave Alliance. *About Z-Wave Technology*. [Online]. Available from: https://z-wavealliance.org/about_z-wave_technology 2018.11.26

[45] A. C. Shin, P. Chandok, R. Liu, S. J. Nielson, and T. R. Leschke, “Potential Forensic Analysis of IoT Data: An Overview of the State-of-the-Art and Future Possibilities,” Proc. 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE Press, June 2017, pp. 705-710, doi:10.1109/iThings-GreenCom-CPSCom-SmartData.2017.182.

[46] B. Lee, A. Awad, and M. Awad, “Towards secure provenance in the cloud: A survey,” Proc. 8th International Conference on Utility and Cloud Computing (UCC), IEEE Press, Dec. 2015, pp. 577–582, doi:10.1109/UCC.2015.102.

[47] Cognitect, Inc. *Datomic Cloud. A transactional database with a flexible data model, elastic scaling, and rich queries*. [Online]. Available from: http://www.datomic.com/ 2018.11.26

[48] B. Duncan and M. Whittington, “Creating an Immutable Database for Secure Cloud Audit Trail and System Logging,” Eighth International Conference on Cloud Computing, GRIDs, and Virtualization, Athens: IARIA, Feb. 2017, pp. 54–59, ISSN: 2308-4294, ISBN: 978-1-61208-529-6

[49] J. Benet, “IPFS – Content Addressed, Versioned, P2P File System (DRAFT 3),” 2017. [Online]. Available from: https://github.com/ipfs/ipfs/blob/master/papers/ipfs-cap2pfs/ipfs-p2p-file-system.pdf 2018.11.26

[50] IPFS is the Distributed Web, 2018. [Online]. Available from: https://github.com/ipfs/ipfs/blob/master/README.md 2018.11.26

[51] Distributed Ledger Technology: beyond blockchain, 2016. [Online]. Available from: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf 2018.11.26

[52] D. Mills, K. Wang, B. Malone, A. Ravi, J. Marquardt, C. Chen, A. Badev, T. Brezinski, L. Fahy, K. Liao, V. Kargenian, M. Ellithorpe, W. Ng, and M. Baird, “Distributed ledger technology in payments, clearing, and settlement,” Finance and Economics Discussion Series 2016-095, Washington: Board of Governors of the Federal Reserve System, doi:10.17016/FEDS.2016.095.

[53] A. Buldas, A. Kroonmaa, and R. Laanoja, “Keyless Signatures´ Infrastructure: How to Build Global Distributed Hash-Trees,” in Secure IT Systems. NordSec 2013. Lecture Notes in Computer Science, vol. 8208, R. Nielson and D. Gollmann, Eds. Berlin, Heidelberg: Springer, pp. 313–320, 2013.

[54] Guardtime. *Cloud Assurance with Blockchains*, 2017. [Online]. Available from: https://guardtime.com/solutions/cloud 2018.11.26

[55] X. Liang, et al., “ProvChain: A Blockchain-based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability,” Proc. 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, IEEE Press, May 2017, pp. 468-477, doi:10.1109/CCGRID.2017.8.

[56] R. C. Merkle, "A Digital Signature Based on a Conventional Encryption Function," in Advances in Cryptology – CRYPTO '87, C. Pomerance, Ed. Berlin, Heidelberg: Springer, pp. 369-378, 1988.

[57] ownCloud, 2017. [Online]. Available from: https://owncloud.org/ 2018.11.26

[58] Tierion Documentation, 2017. [Online]. Available from: https://tierion.com/docs 2018.11.26

[59] Chainpoint, 2017. [Online]. Available from: https://chainpoint.org/ 2018.11.26

[60] BTC.com, 2017. [Online]. Available from: https://btc.com/

[61] Q. Xia, E. B. Sifah, A. Smahi, S. Amofa, and X. Zhang, ‘‘BBDS: Blockchain-Based Data Sharing for Electronic Medical Records in Cloud Environments,’’ Information 2017, vol. 8, iss. 2, pp. 1-16, Apr. 2017, doi:10.3390/info8020044.

[62] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, “MeDShare: Trust-Less Medical Data Sharing among Cloud Service Providers via Blockchain,” IEEE Access, vol. 5, pp. 14757-14767, July 2017, doi:10.1109/ACCESS.2017.2730843.

[63] G. Kingslay, “Hashgraph vs. Blockchain Is the end of Bitcoin and Ethereum near?” [Online]. Available from: https://coincodex.com/article/1151/hashgraph-vs-blockchain-is-the-end-of-bitcoin-and-ethereum-near/ 2018.11.26

[64] L. Baird, “The Swirlds Hashgraph Consensus Algorithm: Fair, Fast, Byzantine Fault Tolerance,” Swirlds Tech Report Swirlds-TR-2016-01, May 31, 2016. [Online]. Available from: http://www.swirlds.com/downloads/SWIRLDS-TR-2016-01.pdf 2018.11.26

[65] Digital Bazaar, Inc. *Blockchain Technologies Feature Analysis*, 2016. [Online]. Available from: https://lists.w3.org/Archives/Public/public-blockchain/2016Oct/att-0004/BlockchainTechnologiesFeatureAnalysis.html 2018.11.26

[66] Z. Hess, Y. Malahov, and J. Pettersson, “Æternity blockchain”, 2017. [Online]. Available from: https://blockchain.aeternity.com/aeternity-blockchain-whitepaper.pdf 2018.11.26

[67] The Streamr Platform. 2018. [Online]. Available from: https://www.streamr.com/#streamrSystem 2018.11.26

[68] S. Popov, “The Tangle,” White Paper, 2017. [Online]. Available from: https://iota.org/IOTA_Whitepaper.pdf 2018.11.26

[69] L. Baird, “Hashgraph Consensus: Detailed Examples,” Swirlds Tech Report Swirlds-TR-2016-02, Dec 11, 2016. [Online]. Available from: http://www.swirlds.com/downloads/SWIRLDS-TR-2016-02.pdf 2018.11.26

[70] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008. [Online]. Available from: https://bitcoin.org/bitcoin.pdf 2018.11.26

[71] N. Szabo, “The Idea of Smart Contracts,” 1997. [Online]. Available from: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html 2018.11.26

[72] Ethereum Blockchain App Platform. [Online]. Available from: https://www.ethereum.org/ 2018.11.26

[73] M. Conoscenti, A. Vetro, J. C. de Martin, “Blockchain for the Internet of Things: a Systematic Literature Review,” Proc. 13th International Conference on Computer Systems and Applications (AICCSA), IEEE Press, Dec. 2016, pp. 1-6, doi:10.1109/AICCSA.2016.7945805.

[74] Eyal, I and E. G. Sirer, "Majority is not Enough: Bitcoin Mining is Vulnerable," arXiv:1311.0243v5 [cs.CR], Nov. 2013. [Online]. Available from: https://arxiv.org/pdf/1311.0243v5.pdf 2018.11.26

[75] S. Barber, X. Boyen, E. Shi, and E. Uzun, "Bitter to Better – How to Make Bitcoin a Better Currency," in Financial Cryptography and Data Security, A. D. Keromytis, Ed. Berlin, Heidelberg: Springer, pp. 399-414, 2012.

[76] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse Attacks on Bitcoin's Peer-to-Peer Network," Proc. 24th USENIX Security Symposium, Washington: USENIX Association, 2015, pp. 129-144, ISBN: 978-1-931971-232

[77] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," Proc. 2016 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE Press, Mar. 2016, pp. 305-320, doi:10.1109/EuroSP.2016.32.