



# On the suitability of blockchain platforms for IoT applications: Architectures, security, privacy, and performance<sup>☆</sup>

Sotirios Brotsis<sup>a,\*</sup>, Konstantinos Limniotis<sup>a,b</sup>, Gueltoum Bendiab<sup>c</sup>, Nicholas Kolokotronis<sup>a</sup>, Stavros Shiales<sup>c</sup>

<sup>a</sup> University of Peloponnese, Department of Informatics and Telecommunications, 22131 Tripolis, Greece

<sup>b</sup> Hellenic Data Protection Authority, 11523 Athens, Greece

<sup>c</sup> University of Portsmouth, Cyber Security Research Group, PO1 2UP, Portsmouth, UK

## ARTICLE INFO

### Keywords:

Blockchain  
Consensus protocols  
Cyber-attacks  
Fault tolerance  
Internet of things  
Security  
Smart contracts  
Smart homes  
Performance  
Privacy

## ABSTRACT

Blockchain and distributed ledger technologies have received significant interest in various areas beyond the financial sector, with profound applications in the Internet of Things (IoT), providing the means for creating truly trustless and secure solutions for IoT applications. Taking into account the weak security defences that the majority of IoT devices have, it is critical that a blockchain-based solution targeting the IoT is not only capable of addressing the many challenges IoT is facing, but also does not introduce other defects, e.g. in terms of performance, making its adoption hard to achieve. This paper aims at addressing the above needs by providing a comprehensive and coherent review of the available blockchain solutions to determine their ability to meet the requirements and tackle the challenges of the IoT, using the smart home as the reference domain. Key architectural aspects of blockchain solutions, like the platforms' software and network setups, the consensus protocols used, as well as smart contracts, are examined in terms of their ability to withstand various types of common IoT and blockchain attacks, deliver enhanced privacy features, and assure adequate performance levels while processing large amounts of transactions being generated in an IoT environment. The analysis carried out identified that the defences currently provided by blockchain platforms are not sufficient to thwart all the prominent attacks against blockchains, with blockchain 1.0 and 2.0 platforms being susceptible to the majority of them. On the other side, privacy related mechanisms are being supported, to varying degrees, by all platforms investigated; however, each of them tackles specific only privacy aspects, thus rendering the overall privacy evaluation a challenging task which needs to be considered in an ad-hoc basis. If the underlying consensus protocols' performance and fault tolerance is also considered, then only a small number of platforms meet the requirements of our reference IoT domain.

## 1. Introduction

The impact of blockchain technology has already begun to be widely known and this may just be the tip of the iceberg. Since the evolution of Bitcoin, the virtual currency created by Satoshi Nakamoto [1], the blockchain technology has met growing interest in the last years as a novel technology facilitating the degree of decentralisation required by modern applications and services in an efficient and robust way. In simple words, blockchain is a distributed database of records, or shared ledger of all the transactions or digital events having been executed and exchanged among a number of parties. The key characteristic of this distributed database is the fact that each transaction in the

ledger has been verified by the majority of the network's peers (via a consensus protocol) and is immutable, therefore inherently supporting transparency and considerably high security level.

Apart from the financial sector and the development of decentralised currency systems, the IoT constitutes a prominent application domain of blockchain with a large number of frameworks, architectures, and solutions having been proposed so far to improve the operational characteristics and overcome the limitations of the current systems [2,3]. Application domains range from smart grid, smart cities and smart homes to IoT cloud and IoT edge following the recent

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

\* Corresponding author.

E-mail addresses: [brotsis@uop.gr](mailto:brotsis@uop.gr) (S. Brotsis), [klimn@uop.gr](mailto:klimn@uop.gr) (K. Limniotis), [gueltoum.bendiab@port.ac.uk](mailto:gueltoum.bendiab@port.ac.uk) (G. Bendiab), [nkolok@uop.gr](mailto:nkolok@uop.gr) (N. Kolokotronis), [stavros.shiales@port.ac.uk](mailto:stavros.shiales@port.ac.uk) (S. Shiales).

<https://doi.org/10.1016/j.comnet.2021.108005>

Received 15 September 2020; Received in revised form 24 January 2021; Accepted 11 March 2021

Available online 19 March 2021

1389-1286/© 2021 Elsevier B.V. All rights reserved.

advancements in telecommunications to improve latency and other network performance metrics. In particular, the adoption of blockchain in the IoT has already led to powerful decentralised information systems, allowing IoT devices to act autonomously and carry out transactions by executing smart contracts [4]. At the same time, the technological evolution brought by the IoT gives birth to new forms of cyber-attacks that take advantage of the heterogeneity and complexity of the IoT ecosystem. Current centralised security defences, apart from the fact that they often rely on trusted third parties, they are inadequate to deal with sophisticated attacks that exploit the existence of various vulnerabilities in IoT devices and platforms. Therefore, applications of blockchain towards enhancing the security and privacy of the IoT have received great attention due to its capacity in providing a fundamentally different approach as a distributed, transparent and accountable system [5–7]. Examples of application areas include identity security [8], intrusion detection systems and mitigation of distributed denial-of-service attacks [9,10], data security and digital forensic evidence preservation [11,12], as well as privacy enhancement and management [13,14]. Specifically for the IoT, the security through transparency approach in public blockchains has in many cases clear advantages compared to the usual security through obscurity model. However, there are major concerns when integrating blockchain with the IoT [15–17]. The blockchain network needs to: (a) support an increasingly huge number of IoT devices, which could join or leave the network at any time; (b) withstand adversarial behaviour and mitigate the attacks occurring in an IoT environment; (c) be explicitly designed for the IoT ecosystem, taking into account the limited resources (computing power, storage, memory) available in IoT devices; (d) achieve high performance; and (e) satisfy the needs for increased user privacy.

In this article we perform a systematic literature review of a large number of dominant blockchain platforms and their consensus protocols to define their ability to meet the requirements and tackle the challenges imposed by IoT applications. In particular, starting from our preliminary results in [18], which considered the security aspects of a limited number of consensus protocols, we considerably extend our study in terms of both the blockchain platforms and consensus protocols investigated but also on the criteria against which these are evaluated. Our reference IoT domain is that of a smart home or *small-office, home-office* (SOHO), due to the fact that it constitutes a prominent application domain of the IoT platforms and services, thus extracting the requirements that the blockchain solutions should meet. Compared to previous relative surveys of blockchain-enabled IoT applications [2,19,20], in which a limited number of blockchain protocols and/or platforms is compared, little or no focus on security and privacy is given, whereas the suitability of blockchain implementations to the IoT is investigated with unclear evaluation criteria, we provide a holistic approach with a large number of factors considered jointly on a common ground and a reference IoT use case (i.e. the smart home ecosystem). More precisely, we utilise seven critical factors to evaluate in a quantitative manner (instead of a qualitative one), following an extensive analysis, the suitability of 32 well-known blockchain platforms and the underlying 16 consensus protocols for use in IoT applications along with their ability to address various IoT challenges; these factors fall into the broad areas of security, privacy, and performance. The high-level contributions of this work are summarised as follows:

- Presentation, analysis, and comparative evaluation of a *large number of dominant* blockchain platforms and consensus protocols under a common framework and working assumptions linked to a concrete IoT domain with realistic key performance indicators;
- Evaluation is *quantitative* covering the broad areas of security, privacy, and performance (they constitute the grand challenges of the IoT) and is supported by an *extensive* analysis of the relative aspects to allow identifying an optimal balance between the above;

- Consideration of security issues at *all layers* of a blockchain architecture instead of the consensus protocols only — a detailed investigation is carried out to identify the availability of defences against state-of-the-art high-impact attacks towards blockchains;
- Identification of the prominent and emerging privacy-enhancing techniques in blockchains and investigation of the extent to which privacy would be safeguarded in the blockchain platforms and consensus protocols of interest if integrated in IoT applications;
- Analysis of the performance and security trade-off in a large number of blockchain consensus protocols with the goal of supporting Byzantine fault tolerance due to the adversarial nature of the IoT ecosystem.

In terms of security, our analysis indicates that none of the blockchain platforms considered in this work has proved to be successful in addressing all the cyber-attacks; to be more precise, most of the platforms provide security defences to mitigate only some of the attacks, with blockchain 1.0 and 2.0 platforms being susceptible to the majority of them (thus making their use in security-critical IoT applications rather questionable). Hedera [21] was an exception, as it provides protection against most of the considered attacks. In terms of privacy, it was seen that all of the blockchain platforms considered provide some privacy preservation mechanisms, at a minimum supporting the features of pseudonymisation and off-chain storage (in case of private permissioned blockchain). Depending on the intended usage (related to cross-industry usage), some platforms such as Ethereum, Hyperledger Fabric, Quorum, have shown to provide an increased support for privacy-enhancing mechanisms compared to the others. However, we show that none privacy-enhancing mechanism should be considered as a panacea per se. Finally, it was shown that if platforms' security and privacy is jointly considered with the underlying consensus protocols' performance and fault tolerance, then only a small number of cases meet the requirements of our reference IoT domain. New and emerging protocols, such as BFT-SMaRt, PoET, VBFT, Exonum, and DAG-based consensus, seem to satisfy these requirements if implemented in a secure, permissioned and high performance blockchain platform.

The rest of the paper is organised as follows. Section 2 provides an overview on blockchain technology and Internet of Things (IoT) challenges. In Section 3, we investigate a prominent IoT domain, namely the case of smart homes, and how various proposed solutions utilising blockchain are being used to tackle security, privacy, and other IoT challenges, while the requirements imposed on blockchain solutions by the domain under consideration are discussed in Section 4. Section 5 provides a brief introduction to fault tolerance mechanisms that are commonly used in distributed systems, whereas their incorporation into various blockchain platforms' consensus protocols (16 protocol families) is further analysed in detail in Section 6, in conjunction with their ability to meet the requirements identified for the IoT. A thorough investigation of the security properties of a number of well-known (32 in total) blockchain platforms is being provided in Section 7, focusing on their ability to thwart common types of attacks on blockchains (covering areas ranging from network attacks to cryptographic attacks), while the smart contracts' considerations, security and verification are treated in Section 8; likewise, an extensive analysis on privacy aspects and mechanisms provided to comply with the *General Data Protection Directive* (GDPR) is given in Section 9. Finally, a comparative evaluation of blockchain platforms is carried out in Section 10 to determine the degree of their suitability in IoT (smart home) applications; the conclusions of the paper are provided in Section 11.

## 2. Background and related work

### 2.1. IoT considerations and challenges

The proliferation of the wireless communication in digital technologies has led to the ongoing technological revolution of the IoT.

The cyber-physical devices, such as smart watches, laptops, tablets, smart televisions, patients or infants monitoring systems and medical implants, have become an indispensable part of our today's society, with the IoT technology being implemented into various fields. However, these IoT devices appeared to be an easy victim for the attackers due to their flaws, mainly stemming from the inherent limitations of the IoT devices. More precisely, device manufacturers put emphasis on the devices' cost, size and usability, without taking into account the requirements for power consumption and computational capabilities that are needed to implement robust security mechanisms. Hence, lightweight security practices are being usually adopted, only because a possible exploitation of some of their IoT products might result to the loss of the company's reputation.

Therefore, to counter system flaws, cloud computing has been proposed to enhance the IoT applications through providing significant security and privacy preservation mechanisms [8,22]. Unfortunately, these implementations are based on centralised systems that are associated with cloud servers and a *trusted-third-party* (TTP) to address the necessary requirements of an IoT ecosystem. Although such solutions might be applicable for the time being, various barriers still exist. With the significantly increasing number of the IoT devices, there is a need for more decentralised networks to cope with the aforementioned issues. Therefore, a synopsis of the most prominent challenges that the IoT implementations face is being provided below:

- **Security:** The design and provisioning of a secure solution is a challenging task. Flawed designs and implementations allow cyber-attackers to easily compromise IoT devices and use them as the means for launching other advanced attacks, such as the distributed denial of service (DDoS) attack against Dyn that was attributed to Mirai malware. Current implementations, in which a private cloud acts as single point of failure, can wreak havoc to the network's security, if it is compromised. Even if a single device is attacked, malicious messages and malware can be sent to other devices, which in turn can result to the exploitation of the device or even to the manipulation of the entire information distributed within the IoT ecosystem.
- **Privacy:** With the ever increasing number of the IoT devices and the huge amounts of the generated data, privacy has become a critical concern in the IoT ecosystem and therefore, the device owner needs to be assured that there will be no leakage when its private information is being processed. Unfortunately, many companies allow access to third parties (such as internet service providers) without providing adequate or precise information to their customers of what has been disclosed and for which purposes [23]. Possible leakage or misuse of personal data in an IoT environment can result to significant privacy violations. Therefore, secure mechanisms for ensuring lawful, fair and transparent personal data processing, without disclosing data to unauthorised entities or using the data for other purposes than the original one, are necessary.
- **Encryption:** Before the transmission of the IoT data to the storage mechanism, it is required to be encrypted. The heterogeneity and the limited computation power that the IoT devices possess make the encryption process a significant challenge, since most of the secure cryptographic algorithms (such as the ECDSA) need a significant amount of computational power and energy. Some devices may employ security network protocols, such as IPsec, SSL/TLS and DTL. Due to the existing inherent limitations, such protocols can only be implemented in devices like the gateways that do not suffer, at this level, from such limitations. To this end, the use of hardware to accelerate the cryptographic operations can prevent the network from computational overhead.
- **Performance:** Due to the reliance on real-time data shared by most of the IoT ecosystems, such as smart grids, smart cities, smart homes, *Wireless Sensors Networks* (WSN) etc., the system's

performance can be considered as significant as its security. To secure the future of efficient IoT ecosystems against failures, it is imperative these systems to be self-managed and self-regulated. However, a possible increase in the system's performance should not result in lowering its security. Moreover, the number of IoT devices is constantly increased and this will result in the generation of more IoT data, estimated to be 79.4 zettabytes until 2025. Therefore, it is compulsory that the corresponding IoT ecosystems will be able to accommodate any possible expansion of their network and handle an abundant number of messages with high performance metrics.

The current threat spectrum necessitates the use of multiple sophisticated security mechanisms for the IoT ecosystem. Various researchers envisage the blockchain technology as the silver bullet to strengthen the security of the IoT. Therefore, it is important to get familiarised with this technology, before proceeding further.

## 2.2. Blockchain technology characteristics

The blockchain is an innovative technology, which has proven to provide considerable advancements in several areas by inherently providing interoperability, decentralisation, security, auditability, privacy, persistence and sustainability; it can be considered as an autonomous root-of-trust, to transfer information from one entity to another, by forging not only a distributed sequence of records but also a linear and chronological order of blocks.

This technology basically realises a distributed service without depending on any central authority to enforce trust between the participating entities. It is meant to provide an immutable service for preserving records between a set of mutually distrustful entities. The security properties that are fulfilled by the blockchain technology are unprecedented really enthusiastic. The pioneering work of Nakamoto [1] has tremendously affected the digital payments around the world. Therefore, due to the unique and desirable properties of blockchain, this technology along with the applications that are built on top of it, are envisioned to revolutionise not only the majority of the financial services but also various other areas.

From its infancy, the blockchain technology empowered segments of executable code to define operations that have been acknowledged by all the participating entities, i.e. the smart contracts. The smart contracts are computer programs which are embedded in the blockchain to digitally establish, verify and enforce that a contractual agreement is executed on a given condition. Therefore, each node can run the contract locally, propose the result to the network and then, cooperate with it to decide which result is going to be included into the blockchain [24]. Smart contracts however can provide features only as far as the blockchain technology can, and test the limits of our modern lives beyond crypto-currency. Therefore, as a revolutionised technology of our today's society, the smart contracts can be adopted in many business areas, such as the supply chain, the intellectual property, the digital asset exchange and the IoT ecosystem.

Blockchain networks are categorised into different settings. In contrast to permissionless, in which each node can be part of the network, the permissioned settings can be either public or private and are defined by the fact that the participating entities, who regulate and amend the blockchain, are identified and can be held accountable for their actions. The key insight behind this concept is to fulfil the necessary flavours of privacy and other requirements of the blockchain applications, as well as to provide beneficial access control mechanisms along with some means of identification.

Between the various components that are included in a distributed blockchain network, the consensus protocol is the key process that facilitates the network's security and most importantly ensures that all the participating entities agree upon the total order of the transactions. Therefore, various design choices of the protocol can significantly impact the network's security (i.e., the fault tolerance) and its performance (i.e., the transactions' throughput, latency and the system's scalability).

**Table 1**

Related work regarding the surveyed factors for the adoption of the blockchain technology to the IoT ecosystem.

References	Challenges in the IoT	Consensus' security	Platform's security	Smart contracts' security	Performance metrics	Privacy mechanisms	IoT suitability
[4]	✓✓	✓✓	–	✓✓	–	–	–
[5]	✓✓	–	–	–	–	–	–
[25]	✓✓	✓	–	–	✓	–	/
[26]	✓✓	✓✓	–	–	–	✓	–
[27]	✓✓	–	–	–	–	–	–
[2]	✓✓	✓✓	–	–	✓	–	–
[28]	–	✓	✓	✓✓	–	✓	–
[29]	✓✓	✓✓	✓	–	✓	–	/
[30]	✓✓	✓	–	–	✓	–	/
[16]	✓✓	✓	–	–	–	–	–
[20]	✓✓	✓✓	–	–	×	–	/
[31]	✓✓	✓✓	–	–	✓✓	–	–
[32]	✓✓	✓✓	✓	✓	✓✓	–	–
Our work	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓	✓✓

✓✓ Denotes that the factor is fully examined.

✓ Denotes that the factor is surveyed, but not fully examined.

– Denotes that the factor is not surveyed.

× Denotes that the performance metrics are given in a qualitative way.

/ Denotes that the IoT suitability is undefined.

### 2.3. Related work

To date, several surveys [2,4,5,16,20,25–33] have been conducted regarding the blockchain technology and its integration with the IoT ecosystem. However, there is no promising work that evaluates, in a quantitative manner, this integration. Several of the aforementioned surveys are focused on general applications of the blockchain technology or discuss various technical aspects that concern the digital cryptocurrencies. Some of these surveys are focused only on the consensus' security and provide qualitative information – instead of quantitative – about the performance metrics, without taking into account other important aspects, such as the smart contracts, the platform's security and the privacy preservation mechanisms that have been implemented.

The authors in [4] provide an intuition into the use of smart contracts for the IoT. The paper focuses on the use cases of a blockchain-enabled IoT ecosystem and discusses the requirements of this integration. These requirements include the high transactions' throughput and the low block confirmation time in consensus protocol, as well as privacy and legal matters that are correlated with the smart contracts. The authors in [5] proposed a lightweight model for the integration of the blockchain technology to a smart home, focusing only on the limitations of the proof-of-work consensus protocol and presenting solutions to avert issues that may occur from the computation overhead and the block confirmation time, as well as scalability issues.

In [26] and [16], the authors conducted a comprehensive review of how to integrate the blockchain technology to the IoT. However, these papers discuss only the challenges, the recommendations and all the possible optimisations that can affect the architecture and the development of this integration. In [25], the authors examined the requirements of the IoT and proposed a framework that identifies suitable blockchain solutions for different IoT applications.

Khan et al. [27] conducted a survey regarding the security issues that might arise in an IoT ecosystem and outlined only how these issues can be addressed by the blockchain technology. Lao et al. [2] surveyed a number of consensus protocols and blockchain platforms for the IoT ecosystem and compared them in terms of the given transaction's throughput and the consensus security. The paper focuses only on these two factors in order to evaluate whether the blockchain technology can be implemented to the IoT. In a recent paper [28], the authors partially examined the security and privacy risks of the most prominent blockchains in all the layers of the blockchain architecture. This survey paves the way to further consider more aspects regarding the security issues that might arise in a blockchain implementation. Perez et al. [30] proposed an overview of a small number of consensus algorithms and examined their suitability to an IoT ecosystem.

In another work, [29], the authors compared a number of consensus protocols and few blockchain platforms and, based on the consensus' security and the performance factors, made an attempt to define a suitable technology for the IoT. However, this paper does not focus on the privacy issues, the platform's security or the smart contracts. In [33], the prerequisites and the issues that might occur when the blockchain technology is implemented in a smart home ecosystem is being discussed, highlighting the design practices that eventually will lead to better integration of a smart home ecosystem with the blockchain technology.

In yet another work [31], a survey on consensus protocols and blockchain platforms for IoT applications is given. The authors highlighted the advantages, the drawbacks and the key challenges of the blockchain technology in IoT applications.

In a similar work to ours [32], a taxonomy of the most prominent consensus protocols for the IoT was proposed. The comparison is based on the performance and adversarial tolerance of the surveyed consensus protocols, as well as, on the platforms' type. However, several other factors for IoT suitability are not mentioned in this paper, such as the privacy preservation mechanisms that have been implemented in each blockchain platform and the security of the smart contracts.

To the best of our knowledge, the only works closest to ours are [20,29,31,32]. In a recent survey [20], Salimitari et al. examined a number of consensus protocols and blockchain platforms for evaluating their suitability to the IoT ecosystem. However, the corresponding suitability of each consensus protocol for the needs of the IoT seems to be undefined and without strictly defined criteria. The performance metrics and all the critical factors are presented in a qualitative manner without considering other important aspects for the blockchain technology integration, such as the platform's security, the security of the smart contracts and the privacy preservation mechanisms.

Therefore, to cover the gaps in the literature regarding the blockchain's integration in the IoT and particularly in a smart home ecosystem, there is a need to carry out a comprehensive survey to find out how existing blockchain solutions integrate with the IoT ecosystem and how suitable these solutions are in a quantitative manner. For this reason, a methodical review for the available blockchain solutions to determine their ability to meet the requirements and address the challenges of the smart home ecosystem, based on certain efficiency, security and privacy factors, and an evaluation of their suitability for our reference IoT ecosystem is necessary. Therefore, there are many factors that distinguish our work from previous relative work and they are presented in Table 1.



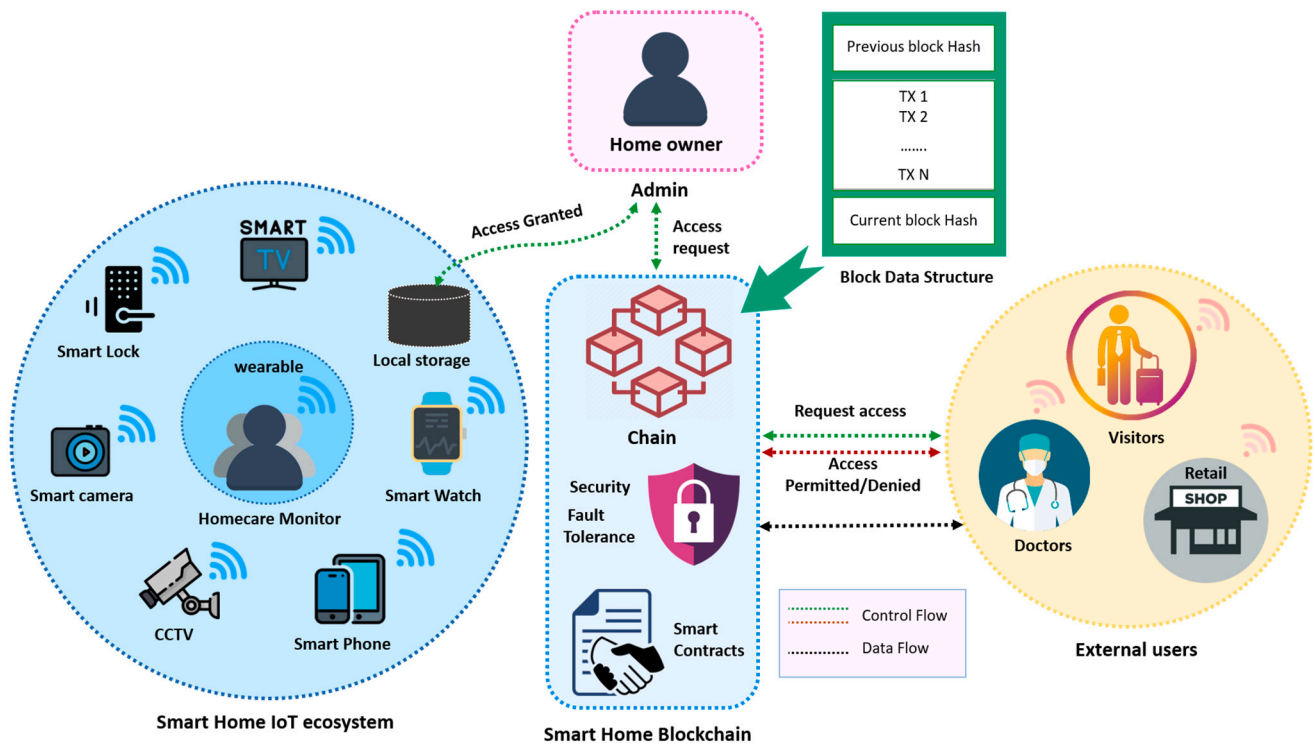


Fig. 1. Blockchain integration to the smart home ecosystem.

### 3. Blockchain architectures for smart homes

Smart home, the most popular and promising use case of the IoT, has already become a significant part of the aforementioned digital revolution. This smart residence encompasses a network of appliances and devices that belong to various application areas, such as lighting control, appliance control, entertainment, assisted living, climate control and safety systems [34]. The combination of these appliances to a particular use case can aid to create innovative, automated, and smart services for residents through distributed and collaborative operations. However, these smart devices pose potential risks to privacy, since the IoT data involved in smart homes is mostly personal. In this context, many research studies have been demonstrated that most smart home appliances lack basic security features and can be easily compromised [33,35]. Moreover, most current smart homes, depend upon centralised cloud services with a single access point for data storing, processing and management. Such architectures create data transparency, trust and privacy-related issues, since the home residents have no control over their own data [34,36].

The emergence of blockchain as a secure and transparent structure for protecting data has inspired fresh thinking about the benefits of applying this technology to the area of the smart home. The functionalities provided by the blockchain technology can significantly (a) enhance the smart home's security, (b) satisfy the necessary flavours of the home-owner's privacy and (c) resolve integrity issues [15]. In addition, a blockchain-enabled smart home ecosystem can significantly improve the security level of this domain by providing trusted, transparent, and secure sharing services and therefore, assure the authenticity and traceability of the personal data exchanged amongst the IoT devices. Another benefit that is brought by this integration is the development of a trusted and distributed authentication network, in which the source of data can be uniquely identified at any time and remain immutable [16]. All these benefits, amongst many others, demonstrate that the blockchain technology can face the security and privacy requirements in a smart home ecosystem (Fig. 1).

When the blockchain technology is implemented in a smart home, three possible scenarios can occur [16]. In the first scenario, the interactions between the IoT devices are stored on an external database and only a part of the associated data is stored on-chain. Whereas, in the second scenarios, all the interactions are stored on the blockchain along with the associated data, keeping an immutable record of all the interactions that occur in the network. The third scenario is a hybrid design, in which only a part of the IoT data and the corresponding interactions are recorded.

The first scenario is the optimum case in terms of latency, due to its off-chain implementation. However, it can only be applied in cases where the IoT data is reliable and the interactions between devices and the appliances occur almost instantaneously [16]. The second scenario is more applicable, but in this case all the interactions are stored on the blockchain. In this case, bandwidth and performance issues can easily arise. The last case is gaining more attention from academia and industry, but choosing which interactions should be recorded is considered to be another challenging task. In this article, we present the most relevant implementations that have been developed by individual researchers and/or in the context of research projects.

#### 3.1. Cyber-Trust blockchain

Cyber-Trust Blockchain (CTB) is a private blockchain platform, built on-top of Hyperledger Fabric and designed to facilitate a secure collection and preservation mechanism of forensic evidence for a smart home ecosystem [12]. The system implements a private forensic evidence database, where the captured evidence is stored, so that it can be used in a court of law. The blockchain platform stores evidences' metadata, which are critical for providing the important services; and interacts via smart contracts with the different entities involved in an investigation process, including law enforcement agencies, internet service providers and prosecutors. The proposed blockchain platform has been developed in the context of the Cyber-Trust project [37] that aims to develop an advanced detection and mitigation platform for trusted smart home ecosystem. The main idea of this mechanism is that if an IoT device is

compromised, an *Intrusion Detection System* (IDS) is initiated to collect relevant information that can be used as digital forensic evidence during the investigation process. The IDS can operate not only at the device's, but also at the network's level to facilitate the collection of evidence from various independent sources. During pilot tests in this blockchain-enabled SOHO ecosystem [12], it was found that when each SOHO is under attack, it creates 50 alerts per second, with each alert to be on average 1.5 kB. Having  $10^4$  SOHOs, it was found that more than 2000 TPS were required for a sustainable implementation. This mechanism also follows a combination of the first and the third integration scenarios, in which only the critical information that has been captured by the detection system is recorded off-chain and the metadata are recorded on-chain.

### 3.2. IoTex blockchain

IoTex [38] is a decentralised blockchain solution that aims to empower the IoT ecosystem by an end-to-end trust, where all IoT devices, humans, machines and businesses can securely exchange data at a global scale. This system provides a well-balanced distributed network that addresses both privacy and scalability issues. A fast consensus protocol with low-cost transactions is combined with a lightweight cryptography mechanism to conserve energy, storage and computation resources. The main idea of this project [38] is analogous to that of the Internet, but in this case, the IoTex blockchain locally supervise a specific group of IoT nodes, which are part of the smart home ecosystem. This implementation can provide various benefits to the network, but the main goal is to safeguard the home-owner's privacy.

### 3.3. GHOST blockchain

The GHOST protocol [14] aims to deploy an efficient security framework by implementing a private Ethereum network in a smart home ecosystem. The blockchain implementation follows the second integration scenario, which enables each IoT device to act as a regular blockchain peer and as a smart home gateway middle-ware, at the same time. The computational limitations are addressed by assigning in each node a different work to perform; therefore, the nodes with limited computational capabilities act as lightweight nodes and only devices with adequate computation power (i.e. the full nodes) participate in the consensus process, thus making the GHOST protocol a functional implementation.

### 3.4. Atonomi blockchain

The key idea behind the Atonomi project [39] is to provide an innovative embedded solution to secure IoT devices by rooting their identity and their reputation on an immutable distributed ledger. This is accomplished by creating and deploying a network that maintains decentralised consensus for the transactions of the IoT devices on the Atonomi network. The Atonomi architecture, which is built on-top of the Ethereum platform, is fully extensible by programmers across the IoT verticals and capable to safeguard the vast realm of the IoT, ranging from the smart home ecosystem to the wide-industrial IoT landscape. In the context of the smart home, the home-owners can register their IoT devices to the Atonomi network and secure them from various attacks, but also from other devices that might have been compromised.

Various other individual researchers have also introduced intriguing blockchain-based smart home applications [5,40–45]. In this context, a novel instantiation of blockchains for smart homes is proposed in [5]. The main goal of this work is to preserve home-owner's security and safeguard its privacy by using a local and private blockchain that provide secure access control to the devices and to the user's IoT data. The proposed architecture involves three core tiers, namely: the overlay network, the smart home, and the cloud storage layer. In this solution, a highly resource device centrally manages all the communication that

occurs in its smart home ecosystem, either internal or external. The shared overlay network has been proposed to inhibit multiple smart home-owners in a single blockchain network and reduce their operating costs. For these challenges to be achieved, the nodes are gathered into clusters, with each cluster to appoint a Cluster Head to be the leader. This architecture follows the second integration scenario, in which a local blockchain mechanism is used to keep track of all the transactions that occur in the smart home, along with the local and the cloud storage solutions that store and share the IoT data.

## 4. Requirements for a blockchain-enabled smart home ecosystem

This section studies the major issues needed to be addressed for the integration of the blockchain technology in a smart home. This integration is not trivial, since blockchains have been designed for a use case scenario with capable computers, which is far from the smart home reality [16]. Currently, various commercial projects provide to the smart home-owners the ability to purchase their own blockchain-based solutions. However, some of these solutions lack in terms of security, performance and privacy protection, with no explicit instructions to be included for their integration. The primary challenges concern the implementation of blockchain applications that better suit to these requirements. Therefore, the feasibility of distributed applications in smart homes, needs to be fully realised under real world deployments. For these reasons, we explore and categorise the prerequisites that are needed to be taken into account before the blockchain technology is integrated in a smart home ecosystem. Our methodology, the requirements taken into consideration and the sections that they are being discusses in order to evaluate the suitability of the blockchain technology to a smart home ecosystem, are presented in Fig. 2.

### 4.1. Performance requirements

In a smart home ecosystem, the IoT devices can be categorised as data generators that create data traffic, and data consumers that respond to a generated event. For instance, a smart thermometer regulates the temperature to create or maintain a comfortable environment for the home-owner by disseminating instructions to the smart air-conditioning devices. The implementation of a blockchain mechanism in this case, might hamper responsiveness by its performance with the generated IoT data to be consumed at a later time than the expected. If the confirmation time of blockchain transactions that record IoT operations is stalled, significant delays might occur in the future control and communication with the IoT devices. Therefore, the IoT data should be immediately stored on a high performance shared ledger without impeding the network's efficiency.

These requirements, in terms of performance, are studied in many research papers [2,46–67], and are categorised as: (1) Scalability, which is the ability of the network to support an amount of users and any possible increase of its size, without degrading its performance. (2) Transactions' throughput, which defines how many *transactions per second* (TPS) can be processed during the consensus process, and (3) Latency or *block confirmation time* (BCT), which defines how much time is required for a transaction to be confirmed and included to the distributed ledger.

In a smart home ecosystem, the IoT devices might need to join or leave the network on-the-fly for various application scenarios. Scalability is often defined as a broader concept of the network's performance and it is the qualification of the blockchain network to maintain throughput when the consensus nodes are increased. For instance, some protocols can only work well in a limited and fixed-size network, in which the number of the network's nodes cannot be more than a few tens and easily modified. Therefore, deploying a huge blockchain network in a smart home, without the nodes being able to leave or join the network, can lead to its performance degradation and compromise the network's security.

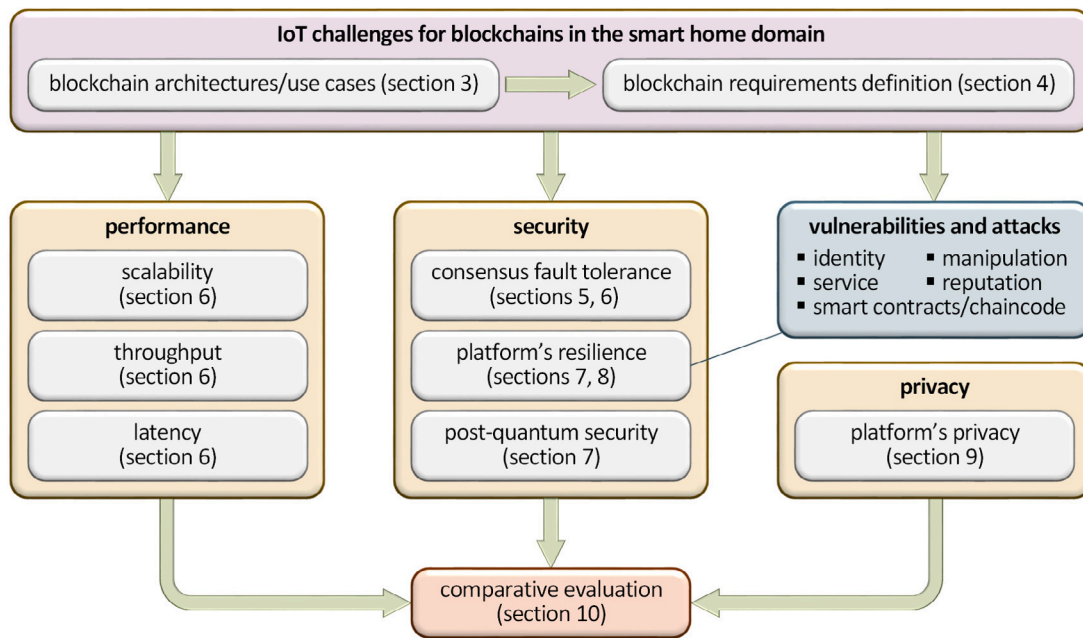


Fig. 2. Blockchain requirements and paper's methodology.

Apart from scalability, the transactions' throughput and latency are some of the most critical factors to be considered in a blockchain-enabled smart home deployment, since it is required that a transaction is being broadcast and appended into a block within seconds or even less [20,33]. Therefore, traditional implementations (such as, Bitcoin's) and consensus protocols (such as the Nakamoto's consensus) might not be applicable for delay-delegate smart home applications, in which small networks delays can cause severe consequences.

In our work, we evaluate the transactions' throughput as *high*, if more than 2000 TPS can be processed, and the transactions' latency also as *high*, if the time that is required for each transaction to be confirmed is more than 10 s. The thresholds derivation occurred based on the researched blockchain smart home ecosystems and the expected volume of events/transactions being generated in a typical smart home ecosystem, according to the approaches presented in Section 3. For example, in [68] and [69], blockchain-enabled smart home ecosystems were proposed based on the Ethereum platform. In both of these implementations, the authors mentioned that their systems may undergo a challenge in time-delegate conditions, since it takes enough time for each transaction to be confirmed [68,69]. Moniruzzaman et al. [33] even claims that the BCT of 3.9 s needs to be further reduced for a sustainable blockchain-enabled smart home ecosystem. Therefore, BCT greater than 10 s might not be so suitable for time-sensitive applications. On the other hand, the required transactions' efficiency is rather based on the integration scenarios. For instance, much higher throughput is required when all the interactions that occur in the smart home are stored on the blockchain than in a scenario, in which only the critical alerts are being stored on the blockchain. However, in our work, different values have been given for the different performance metrics for both the transactions' throughput and confirmation time. An acceptable example, in terms of performance, is proposed in [70], in which the achieved throughput is greater than 2000 TPS and the BCT is less than 1 s.

#### 4.2. Storage requirements

A smart home includes always connected IoT devices that communicate and target services. These devices do not require any tremendous computation power and storage area. Therefore, their limited/non-existing memory or storage capacity make the preservation of the IoT

data a significant issue. The integration of a huge processing mechanism, such as a blockchain platform, into a resource-constraint network might not be seen as a suitable solution. However there are ways, in which these limitations could be alleviated or even avoided.

Currently, in a smart home ecosystem, a huge amount of IoT data is generated, whereas only a limited part of it is useful to extract knowledge and to generate actions. Furthermore, several techniques have been suggested to filter and compress the data derived from the IoT devices and address the storage limitations. Therefore, savings in the amount of the stored data of the smart home can be achieved. Another possible solution to this issue is the separation of the actual data from the metadata. Having the data stored in a secure database and the hashes of this data on the blockchain, the storage issues can be easily solved, whilst the system's security is being preserved [12]. Throughout the paper, we consider storage requirements to be strongly linked to the maximum expected amount of transactions generated by a blockchain platform; thus, TPS also serves to measure the storage needs for a typical transaction size of  $\approx 1$  kB (we are *only* interested in the *order of magnitude* here), and this is eventually reflected in a platform's scalability; (see Section 10 for the details).

#### 4.3. Privacy requirements

The right to privacy has been recognised as a fundamental human right by the United Nations Declaration of Human Rights, the International Covenant on Civil and Political Rights, the Charter of Fundamental Rights in European Union and other international treaties. Privacy is strongly related with the personal data protection; as stated in the Charter of Fundamental Rights, personal data (i.e. data relating to an identified or identifiable natural person) must be processed fairly for specified purposes and on the basis of the consent of the person concerned or some other legitimate basis laid down by law under Article 8(2) of the Charter. The Charter furthermore provides that everyone has a right to access personal data relating to them, including a right to have such data rectified or deleted [71].

The *General Data Protection Regulation* (GDPR), being applicable since May 2018, constitutes the main legal instrument for personal data protection in Europe, establishing a detailed framework that harmonises data protection across the European Union. Most importantly, GDPR also applies to the processing of personal data of individuals

being in the European Union by organisations not established in the Union, in case that the processing activities are related to either the offering of goods or services or the monitoring of their behaviour (as far as their behaviour takes place within the European Union). The GDPR codifies the basic principles that need to be present when personal data are being collected or further processed, in order to promote the fundamental right of personal data protection.

In a smart home ecosystem, many IoT devices process confidential personal information, which poses risks for the rights and freedoms of individuals; for instance, if an IoT device is linked to a smart home-owner and is related with health data, such as in an e-health scenario [72], then an improper use of this information may harm the individual even if the device is not attacked. Therefore, it is essential to ensure each user's essential right to privacy. In general, the necessary flavours of privacy cannot be easily guaranteed when data derived from IoT devices are being processed and, thus, a careful design approach should be followed [13]. Such privacy issues entail more effort, since they start from the data collection process and extend to the data preservation mechanisms.

The question of compatibility between blockchains (not only in IoT applications but in any case) and GDPR is not always an easy answer. For example, blockchains are presented as the ideal solution to counter the identity management issue in a smart home ecosystem [5] or when the home-owners want to allow, refuse and withdraw access to their personal data according to different cases of potential use [73]. However, in general, the inherent properties of blockchains may also give rise to privacy threats. For instance, the transaction's linkability is a privacy threat, in which the data (transactions) that are stored on the blockchain can be linked and allow a reconstruction of the user's profile (which in turn can be used for other purposes, without the concerned user being aware of this). This is an obvious privacy threat in permissionless and public settings, where the entire blockchain is visible to an adversary. As stated in [71], the aforementioned compatibility issue can only ever be determined on a case-by-case basis that accounts for the respective technical and contextual factors (such as the governance framework).

Therefore, for securing the sensitive IoT data from unauthorised entities, it is required the implementation of permissioned and private settings. Moreover, to tackle with other risks of malicious access to the users data (which is both a privacy as well as a security issue), appropriate cryptographic primitives may be adopted by the resource constrained IoT devices. In addition, depending on the specific data protection requirements induced by the context of the application, several privacy enhancing technologies relying on advanced cryptographic techniques can be implemented (i.e. in cases where the data in a blockchain corresponding to a specific individual need to be indistinguishable from the relevant stored data corresponding to other individuals, in order to mitigate the risk of illegitimate profiling). Such technologies are being subsequently discussed in Section 9, in relation with the relevant blockchain platforms that support them.

In our work, privacy is mapped as *medium*, and *high* in order to satisfy the user's privacy requirements in a blockchain-enabled smart home ecosystem. Privacy is defined, based to the nature of the blockchain platforms: permissionless; denoted in the text as "PL" and permissioned, which is denoted as "P". Towards this categorisation, privacy preservation mechanisms have been implemented in various blockchain platforms, even in Bitcoin, to serve various purposes. Nevertheless, permissioned blockchain platforms are more capable to satisfy the necessary flavours of privacy in this particular ecosystem and therefore, privacy is mapped in this way.

Last but not least, a smart home owner may exercise its data rights on the processing of its personal data — including the case that the personal data are stored on a blockchain. Due to the immutability that the blockchain technology provides, the information that it is stored on a distributed ledger cannot be easily erased. This inherent

property of blockchains is another privacy issue related with the so-called "the right to be forgotten" in the GDPR. Therefore, privacy laws that concern the user's privacy with respect to the right of deletion of data constitute another significant challenge to be tackled in a smart home environment and will be an even greater if they are used in integration with the blockchain technology.

#### 4.4. Security requirements

The increasing number of IoT devices comes with new cyber attacks that have serious effects, thus necessitating the development of smart home ecosystems with more sophisticated security. The IoT devices interact with each other through a gateway using various wireless communication protocols, which usually paves the way for cyber-attackers to perform eavesdropping and several other attacks. The IoT home devices (such as web-cameras, smartphones, etc.) may be compromised and provide unauthorised access or allow the attacker to collect sensitive information. The key security issues in smart home ecosystem are related with weaknesses in authentication and authorisation mechanisms, vulnerabilities in the access control system and configuration issues. The traditional smart home ecosystems systems are mostly centralised and linked with a cloud server; hence if the central server is compromised, the overall security collapses. Therefore, to overcome these issues, various solutions have been proposed that include additional security layers in existing smart home ecosystems [74].

Blockchain can be seen as the key technology to provide the fundamental security enhancements to secure the smart home ecosystem and diminish the security concerns regarding the authorisation and authentication, integrity and confidentiality of IoT data, as well as, the single point of failure. However, an essential challenge for the integration of the blockchain technology to the smart home ecosystem is the reliability and the trustworthiness of the IoT data [7,10]. Due to the immutability of blockchains, if the IoT data that are being stored to the blockchain are already corrupted, then they will remain corrupted. Such situations can arise not only from malicious behaviour, but also from several other possible failures related with the environment or the participants. Therefore, the IoT devices should be exhaustively tested before this integration and appropriate security techniques should be adopted to detect possible device failures.

Despite the fact that most of the blockchain platforms can resist traditional attacks in an optimum way, the adversaries create each day new ways to mount successful attacks [6]. Various blockchain platforms and consensus protocols have been thoroughly analysed in terms of security and several vulnerabilities and security issues have been discovered. One of the most common attacks in distributed systems is the double spending, in which the adversary manipulates the consensus protocol used and attempts to insert malicious transactions to the shared ledger. Therefore, (1) the consensus protocol that is going to be implemented in a blockchain-enabled smart home ecosystem should be able to address any possible malicious behaviour; (2) the blockchain platform should not be susceptible to cyber-security attacks that aim at breaking its integrity and availability, and (3) the smart contracts that execute security operations (such as authentication and access control) should be secure, in such a way that there will be no backdoors for an adversary to compromise the network.

In this paper, we evaluate the consensus' and the platform's security by mapping them to a three-valued scale, namely *low*, *medium*, and *high*. For the consensus protocols that can tolerate crash failures and malicious behaviour (resp. only crashes) security is considered to be high (resp. medium), whereas for the centralised consensus protocols that are destined for developing testing and therefore, the security mechanisms that they possess are considered to be low. On the other hand, each platform's security is based on its capability to address some of the most common attacks that can occur in a blockchain-enabled smart home ecosystem.



## 5. Fault tolerance in consensus protocols

Distributed systems are usually disciplined by a set of clients and services, where each service utilises one or more servers to extract information or execute operations that are requested from the clients. Using a central server is the easiest way to fulfil the necessary needs and the implementation of a service, but it possess a major concern about the security of a smart home ecosystem, due to the fact that it becomes a *single point of failure* (SPoF). Thus, to avoid centralised faults, multiple servers should be deployed to implement a Fault Tolerant service with *state machine replication* (SMR) [75].

The pioneering work of Lamport [76], who introduced the Byzantine Agreement, triggered the research of developing algorithms in order to exploit and construct resilient distributed systems. Nonetheless, several blockchain systems deviate from the classical SMR in crucial ways. Many distributed applications run simultaneously and can be deployed at any time, even if the embedded application code is untrusted or occasionally malicious. The key idea to provide security in blockchains is to reach agreement on a single request from a client, which is the core functionality of a consensus protocol. In the context of blockchains, a consensus protocol or as it is commonly known as an “atomic broadcast”, provides a total order of the disseminated messages and propagates them to the network’s peers/nodes.

### 5.1. Crash fault tolerance

An atomic broadcast certifies that all the legitimate peers output or deliver the identical array of messages by means of the deliver event. Accurately, considering a set of  $n$  peers in the network, it certifies that the properties validity, integrity and total order are fulfilled [77].

The way to achieve consensus (i.e. to realise atomic broadcast) in distributed systems that are vulnerable to  $t < n/2$  node crashes is to adopt consensus protocols known as the *viewstamped replication* (VSR) [78] and Paxos [79] family of protocols. To provide security, this family is characterised by the same rules. In each round a leader is elected or voted to create a new block and if the ongoing leader crashes or even if the peers in the network suspect that the leader has crashed, the leader is reinstated by proceeding to the next round. This family of protocols is known today as *crash fault tolerant* (CFT) consensus protocols and it guaranties that a set of failing nodes  $t < n/2$  does not impact the system.

### 5.2. Byzantine fault tolerance

Consensus protocols with the purpose of tolerating byzantine nodes, which are subverted by a malicious actor and avert the common goal of reaching agreement, have recently emerged. In the *Byzantine fault tolerant* (BFT) consensus protocols family, the most common protocol is the *Practical Byzantine Fault Tolerant* (PBFT) [80], which can be displayed as a blossom of the VSR/Paxos [78,79]. In a network comprised of a set of  $n$  nodes, the PBFT consensus protocol [80] can tolerate  $f < n/3$  subverted nodes using a progression of rounds with a unique leader within each round. Under the assumption that the BFT protocols seem to be more secure than CFT, vast research work has focused on the improvement of PBFT with BFT-SMaRt [81] to be considered one of the most advanced and scalable BFT consensus protocols.

## 6. Consensus protocols in blockchains

The consensus protocols are mechanisms that define a set of rules to provide truthful services among participants without mutual trust to collectively maintain a synchronised state. Due to its nature, a consensus protocol provides incentives for the participating nodes to be honest and create or add new blocks to the blockchain. Such consensus mechanisms are described in this section and are mostly used in cryptocurrency-based applications, where the need to behave

honest and not deviate from the predefined process might cost significant losses. Moreover, some of these blockchain solutions provide a desirable, suitable and high performance network required by the smart home ecosystem only if, the monetary reward that incentivises the consensus nodes to behave honestly is replaced by IoT-based criteria.

### 6.1. Proof of work (PoW) and its variants

The Nakamoto’s consensus [1] is the key technology that revolutionised the digital currency exchange, since it is implemented on several other cryptocurrency-based ledgers, such as Litecoin [82], Monero [83], Ethereum [84], etc.. Several other hybrid variations of the protocol also exist, in which the peers use their computational power to elect a leader, e.g. Bitcoin-NG [85] and Elastico [86]. Nakamoto’s consensus forces each node to calculate a huge amount of hashes and mine a block only if the node finds an appropriate value that satisfies a specific property which is difficult to be achieved. This mathematical problem serves as a means to establish consensus and verify the validity of blocks on the network before accepting it. Moreover, the validation of blocks necessitates the assistance of other principles as well, such as (1) the longest chain rule, where only one chain is commonly adopted by network, and (2) the incentive rule, where only a single node will gain the reward by finding the next block in the chain. PoW can tolerate at most 50% of its network’s total computational power to be used for malicious purposes towards solving the aforementioned puzzle. Therefore, each node is required to spent a vast amount of energy to adhere to the protocol. Therefore, the spent computational power is the necessary regulation for each node to adhere to the protocol, by building trust between all the participating nodes.

As to the terms of performance, even in the same consensus protocol, slightly changing some parameters might result in different throughput and BCT. Bitcoin can provide 7 TPS processing rate, with each transaction to be included in a block in 10 min. Similarly, Litecoin can achieve 56 TPS and BCT of 2.5 min, while Ethereum and Monero seem to be better solutions with 25 and 1700 TPS, as well as, BCT of 10–15 s and 2 min, respectively. However, several other variations of the Nakamoto’s consensus have been proposed to reduce the computational requirements and provide increased performance. Bitcoin-NG [87] can achieve 200 TPS and BCT of 10 min, while Elastico [86] adds to its distributed ledger 16 blocks every 110 s.

### 6.2. Proof of stake (PoS) and its variants

To address the limitations with respect to the computational power that the Nakamoto’s consensus introduces, a different type of consensus has been proposed. The key idea of PoS is the concept of “stake”, where the nodes that participate in the consensus process, lock into an escrow account, which is a specific amount of coins. The concept of the stake serves as a guarantee that each node will behave in accordance with the protocol rules and will not deviate from it. Therefore, the users that possess more stake are more incentivised – with stake’s loss – to safeguard the system’s reliability, since it is less possible to be turned and become malicious nodes. PoS can tolerate at most 50% of its network’s total stake to be used for malicious purposes. Despite of this, the classical PoS protocols cannot achieve high performance metrics; Snow-White [88] and Ouroboros [89] are quite popular protocols, but in terms of performance Snow-White achieves at average 125 TPS and BCT of 10 min, Ouroboros can reach 256 TPS, with each transaction being confirmed in 2 min. To counter the low performance of PoS protocols, several other variations have been proposed with Algorand [90] to be built on top of byzantine agreement. Algorand, a platform that aims to solve the “decentralisation, scalability, and security trilemma”, attaches a cryptographic proof to each new block and checks the eligibility of the block proposed to be chosen, while this probability is directly proportional to its stake. In terms of security and performance, Algorand can tolerate malicious behaviour and provide high scalability with more than 1k TPS and each transaction to be confirmed in 5 s.

### 6.3. Delegated proof-of-stake (DPoS)

Implemented in Bitshares [51], EOSIO [52], Lisk [91], Tezos [62] and Tron [63], DPoS [92] is a reputation-based consensus protocol, considered to be a more efficient and democratic version of PoS. In a similar way, the protocol proceeds in rounds and within each round, a leader is selected (voted) according to the blockchain platform that the protocol is being implemented to. Although it derives from PoS, there are many differences that make DPoS exceptional. The network's users vote with reputation scores to choose a class of delegates that will be authorised to participate in the consensus process. Amongst a set of delegates, a leader is elected and incentivised to adhere to the protocol and create honest blocks. With the creation of a new block, the leader is either rewarded, if the block is honest, or penalised and blacklisted in any other case. The protocol can withstand 33% of the delegates to be malicious. With the ambition to be elected, the delegates can promise various levels of rewards to the token holders. A token holder can entrust a delegate with its own stake, by casting a vote via a blockchain transaction. Then, the delegate acquires the stake power from all of its voters and behaves as their fronted in the consensus process. As mentioned, DPoS' performance metrics differ across the blockchain platforms that it is implemented, but the scalability of the protocol is similar to PBFT's. Bitshares achieves the tremendous transactions' throughput of 100 thousand TPS with each transaction to be confirmed in 1 s, EOSIO's throughput is measured to be 1–6 thousand TPS and its BCT is less than 1 s. DPoS has also achieved high performance metrics in Tron with throughput greater than 2 thousand TPS and BCT 1–3 s. Lisk and Tezos are the exception of the DPoS implementation and can achieve low throughput of 2.5 and 40 TPS, while the BCT is measured to be 6 and 30 min, respectively.

### 6.4. Kafka protocol

Kafka [93] is a distributed publish/subscribe messaging pattern that can transfer large amount of log data with significantly low latency. Kafka's key components are the producers, the topics, the consumers and the brokers. The recorded information is published from the producers to a stream of messages called the "topic", which is a partition of segments of files. The messages are stored from the brokers to the latest segment file and when the producers publish messages to the partitioned logs, only the subscribed consumers can consume these messages, sequentially, by simply making requests to the brokers. At high level, the Kafka's conceptual configuration is based to the leader-follower setting, in which the transactions are replicated from the elected leader to its followers and if the leader crashes or even if there are suspicions that the leader has crashed, then one of its followers takes command. Kafka is a CFT consensus protocol that can address less than 50% of network's crashes and it is mostly known for its implementation in Fabric [47] and FastFabric [94], in which it has been shown to achieve throughput of 3.5 and 20k TPS, respectively, with each transaction being confirmed in less than 1 s.

### 6.5. Raft protocol

The Raft consensus protocol [95] is currently implemented in Fabric [47], Quorum [96] and Corda [97]. Acquiring its security properties from Paxos [98], it is a CFT protocol based on the "leader-election" model. Raft establishes consensus by electing a leading node to obtain the incoming entries from the clients and replicate them. The protocol is separated into three phases allowing a strong and coherent leader election process. These phases are the leader election, the log replication and the safety [95]. The time in Raft proceeds in arbitrary time periods, called "terms", with each term to be defined by an increasing number. The nodes in Raft are hierarchically ranked in different states, with each node to be either a leader, a follower or a candidate. The leader is the principal entity of the protocol and it

is elected per channel with the task to interact with the clients and then replicate its entries to its synchronised followers. Raft achieves the best possible synchronisation by sending systematic "heartbeats" to its followers and if the leader crashes, then at least one of its followers will detect this divergence, cast a vote to the network and attempt to take its place. Some nodes might compete to win the election process by seeking votes from other nodes. Therefore, these nodes are considered as "candidates".

Raft ensures exclusively that a single node will become a leader, even if some nodes miss a term or split ups occur in the leader election process. In the first case, the outdated node will revise the term's number and fall into the follower's state; and in the latter, the current term will be ended without any outcome of the election process.

Raft's performance is not fully tested, but its implementation in Fabric [53], as the recommended consensus protocol since the version 1.4.1, showed that it can provide thousands of transactions in real world scenarios, being two times faster than Kafka in terms of transaction's throughput with even less latency. In the context of Corda and Quorum, Raft reaches throughput less than 200 and 650 TPS with BCT as low as 1 s and 4.5 s, respectively.

### 6.6. Practical BFT (PBFT) protocol

The PBFT [80] is one of the most well-established BFT protocols and has recently attracted much attention with the emerge of the blockchain technology. PBFT works under the assumption that less than 33% of the network's nodes are behaving maliciously. Specifically, there are three phases required to establish consensus, namely the pre-prepare, prepare and commit. Each node is considered to be a *validating replica* (VR) that votes to elect a primary node. This primary node, called here *leader validating replica* (LVR), starts the three phase consensus, after receiving a request from a client, by multicasting a pre-prepare message. When the VRs receive this message, they enter to the pre-prepared phase and disseminate a prepare message to ensure that they are on the same sequence. When the LVR accepts  $2f+1$  prepare messages, enters to the prepared phase and multicasts a commit message. Therefore, there will be at least  $2f+1$  (as  $N \geq 3f+1$ ) correct nodes in the same state after the commit phase that produce the same result.

Despite PBFT's tremendous transaction throughput of 78 000 TPS, measured in [81], the protocol has several drawbacks. The protocol lacks in terms of scalability, supporting only a few nodes and demanding several messages to be transferred to achieve consensus [99, 100].

### 6.7. Istanbul BFT (IBFT) protocol

IBFT [101] belongs to the family of *proof-of-authority* (PoA) consensus protocols and it is implemented in Hyperledger Besu [102] and Quorum [96]. The protocol leverages the potentials of smart contracts to ensure instantaneous finality and robustness in an eventually synchronous network. IBFT was initially designed as an alternative consensus protocol for the Ethereum blockchain that fits better with permissioned blockchains, in which the waste of the computational power to reach consensus is undesirable. The new version IBFT 2.0 of the protocol preserves all the original beneficial features of IBFT and addresses the security limitations, such as the security and the liveness properties. In addition, the number of the nodes that are required to reach consensus has changed and therefore, a byzantine node needs to reach the super-majority of the network, i.e. more than  $2/3$  of validators' votes, to insert a malicious transaction to the blockchain.

In terms of security, both versions of IBFT inherit their security properties from the three phase consensus. In contradiction to the similarities that IBFT and PBFT have, there are some tweaks that are needed to be made in the later to be used as a consensus protocol in

blockchains. The fact that no particular client exists in IBFT to claim results by sending requests, considers all the VRs to be recognised as such and propose new blocks when their turn comes. IBFT's performance in Quorum is not peer-reviewed [48], but it is measured to be 600–650 TPS with BCT of 4.5 s with very low scalability.

### 6.8. Clique/Aura protocols

Clique and Aura [99] are also PoA consensus protocols created to increase the platform's throughput, lower the confirmation time, but also to degrade the need for computational consumption in a permissioned setting. In the case of Clique, the time is separated into epochs, wherein the right to propose a block is given to the mining leader and other authorities only when a specific number of epochs has elapsed. In the case of Aura, each block is not committed at once, but instead it has to pass from the block acceptance round. If the participating authorities disagree on the leader's proposed block during the acceptance round, then the leader's position is under vote. Any leader can be considered malicious by any authority in the network only if: (1) it has not disseminated any block, (2) it has disseminated contrasting blocks to various authorities, or (3) it has disseminated multiple blocks. In any of the aforementioned cases, if the majority of the authorities agree, the malicious leader is removed from their list and all its proposed blocks are discarded. Clique is currently implemented in Besu [103], while Aura is implemented in Parity [104]. In terms of performance, the adoption of Clique in Besu is not yet measured, but Aura can achieve less than 50 TPS and BCT of 5 s. Despite their performance, both Clique and Aura are scalable protocols and can achieve consensus under the presence of byzantine nodes, with tolerance of  $N \geq 2f + 1$ .

### 6.9. BFT-SMaRT protocol

BFT-SMaRT [81] is a java-based library that encapsulates all the complexity of BFT algorithms in a simple API, which can be used to provide deterministic services. Apart from the modules that exist for trustworthy point-to-point communication, as a modular consensus protocol, it can provide reconfiguration modules that are completely separated from the process of agreement. These modules make the protocol vary from prior BFT networks, which are fixed-size networks that cannot be expanded or even reduced. In addition, some of its significant advantages are the features of multi-core awareness regarding the verification of signatures and the flexibility of the programming interface [61]. Compared to the existing SMR libraries, the protocol can reach better performance regarding not only the system's scalability, but also the transactions' throughput and latency, while providing the capability to withstand real-world failures that previous BFT implementations are not able to.

BFT-SMaRT is experimentally implemented in Fabric [47], Symbiont [105] and recently to SMaRTChain [49]. In Fabric's case, the protocol also implements WHEAT [61], a component that BFT-SMaRT's ordering service is relied upon to provide a powerful vote assignment scheme, low latency and fast replication among the VRs without compromising the network's security. The performance of BFT-SMaRT in Fabric is measured to be more than 10 thousand TPS with BCT of 0.5 s [61]. In the context of Symbiont, BFT-SMaRT is not fully tested, since it demands Java serialisation that is disabled by default due to security risks. Nevertheless, it can achieve the tremendous throughput of 80 thousand TPS with transaction latency even less than 1 s. The authors in [49] showed that, if the protocol is implemented in SMaRTChain, it can achieve high performance, even higher than the one achieved in Fabric – with a significant transactions' throughput of 12–13k TPS – and much lower latency, even less than 0.25 s.

### 6.10. VBFT protocol

As the core consensus protocol of the Ontology platform [58], VBFT [106] has been developed to address malicious behaviour via the combination of a PoS and a BFT protocol, while the randomness of the leader election process is achieved by means of *verifiable random functions* (VRFs). Therefore, the protocol obtains the best from all of these three worlds. In VBFT, the nodes are divided into two different types, namely, (1) the consensus nodes, which create blocks by weighting each node's stake and (2) the consensus candidates that only validate and update the consensus blocks, but they do not provide any further assistance in the consensus process. The VRF function is mainly used to select a difficult to be predicted type of work that each node will perform. The initiation of the protocol begins with each node to create and propose a new block. All the new blocks are then gathered, verified and voted by the verification nodes according to their highest priority. The results from the verification nodes are then validated from the confirmation nodes, which in their turn, establish a consensus result and when this new result is accepted, a new round begins. VBFT can withstand malicious behaviour, while implementing many features of scalability (such as, side-chains, sharding, and parallel execution). The protocol can deliver throughput more than 3k TPS and at peak performance 4 thousand TPS with BCT of 5–60 s.

### 6.11. Delegated BFT (DBFT) protocol

DBFT [107] is currently implemented in Neo [57] and in Ontology [58] and as a combination of PBFT [80] and DPoS [92], includes the advantages of both worlds. To address malicious behaviour, the protocol follows the three-phase consensus, without inheriting PBFT's undesirable drawbacks. As more nodes join in the consensus process of the PBFT protocol, the faster its performance degrades, with time complexity of  $\mathcal{O}(n^2)$  [57]. For this reason, DBFT adopts the characteristics of DPoS, in which the nodes with their voting power propose a class of delegates that will be authorised to establish consensus in the next round. Therefore, DBFT provides a fast voting scheme to elect the delegates and address any malicious behaviour. The beneficial features of DBFT regarding its scalability and its byzantine fault tolerance (as  $N \geq 3f + 1$ ), come to contradiction with its performance metrics in the Neo platform, which are less than 1000 TPS and BCT of 15–20 s.

### 6.12. Tendermint protocol

The fault tolerance properties of Tendermint [108] are established under a partially synchronous network for Hyperledger Burrow [109] and Ethermint [54], with high tolerance against double-spending attacks. The protocol relies on a leader election mechanism to maintain a constant consensus group, from which each validator's identities can be disclosed for public supervision.

Although Tendermint seems to be quite similar to the most of the BFT protocols, the voting power that the VRs use to elect a leader is defined in a different way and it is based on each VR's stake. In addition, a concept of “*polka*” or “*locks*” is added in the pre-commit phase to lock the blocks that collect more than the 2/3 of all the network's pre-votes. For various reasons, some honest blocks might fail to be included in a specific height; and therefore, to address this limitation, the protocol moves to a new round with a VR to propose a block for the new height. This block can be either a new block or a previous block, in which the proposer has been locked to from another round. The VRs can unlock from their blocks by collecting more than 2/3 of nil pre-votes. These type of votes, according on the phase that they occur, can define either that a VR has to be unlocked from a previous outdated block or that the protocol has to move to the next round. In any case, if the block is either locked (or unlocked) it has to be accompanied by a *proof-of-lock* (PoL), which is a collection of the (nil) pre-votes that it has received. Tendermint can provide



medium performance metrics in Ethermint and Burrow, but the concept of locks has given the inspiration for the creation of another and high performance protocol; the Exonum [110].

In terms of performance, Tendermint can provide 200–800 TPS and each block to be included in the blockchain in less than 1 s, while Burrow can achieve throughput more than 400 TPS and by default produces blocks every 2 s. Exonum on the other hand, shows significant results with transactions' throughput approximately at 5 thousand TPS and BCT of 0.5 s.

#### 6.13. Proof-of-elapsed time (PoET) protocol

PoET [111] is designed to remove the computational power that PoW-based consensus protocols introduce. The protocol is implemented in Sawtooth [111] and solves the problem of Byzantine agreement by adopting a lottery-based method to achieve investment, verification and fairness in the leader election process. The peers wait for an indefinite time-period to be elapsed [46]; and the peer that finishes waiting first, is elected to be the leader and create a new block. This procedure is executed in a private memory area, called *trusted executed environment* (TEE), such as the Intel's SGX [112], which is mostly used in the Sawtooth platform. Essentially, the TEE provides a cohesion proof that is obtained from a trusted function via remote attestation techniques. This trusted function, which is often called "enclave", establishes trust on the consensus peers and allows the network to tolerate malicious behaviour. Each node calls an enclave inside the SGX to produce a random delay, wait for its timer to be finished, and then declare itself to be the leader. The protocol is BFT but without the integration of the SGX hardware or if the adversary manages to compromise a set of participating nodes beyond the  $\Theta(\log \log n / \log n)$  threshold [113], the network can be manipulated. Although there is no any performance evaluation from the Hyperledger project, Ampel [46] showed that Sawtooth can achieve 2.3 TPS and latency less than 1 s.

#### 6.14. Ripple protocol consensus algorithm (RPCA)

The Ripple consensus protocol is created to provide enhanced security and a stable real-time cryptocurrency-based network for transferring remittances without the obligation to develop smart contracts. In RPCA, each node maintains a *unique node list* (UNL), which is a set of trusted validators especially configured to participate in the consensus process. In Ripple's vast network, trust does not literally means that the set belonging to each node's UNL is actually trusted, but rather that this set is going to behave honestly and that the nodes included in each node's UNL are not intended to collude with each other or attempt to break the protocol's rules or manipulate the network with positive votes on malicious transactions. Each node listens to its trusted validators, but if agreement is not reached upon a set of transactions, the node's proposals are modified according to the proposals of their trusted validators. RPCA requires than 1/5 of nodes to be faulty in each node's UNL in order to achieve overall consensus. Therefore, only the transactions that exceed a specific threshold of 80% of positive votes are processed, while the rest are either deserted or included in a candidate set of new and not yet applied transactions on the ledger. Ripple can reach 1.5 thousand TPS and BCT in 4 s.

#### 6.15. Yet another consensus (YAC) protocol

YAC [56] is based on voting for block hashes to solve the problem of inefficient message processing and strong leader election, that both occur in classical BFT consensus protocols. The novel design of YAC is created for the needs of Hyperledger Iroha to provide the simplest possible construction for the development of mobile applications [114]. Similarly to Fabric, the transactions' validation is not part of the consensus process, but it is rather determined by the platform's transaction flow. For each transaction to be included in a block, two validation

operations are executed from the peers (a stateless and a stateful). The stateless validation ensures the validity of the transaction's format and signature, while the stateful occurs in a slower rate after the transactions' ordering. Iroha's operations are limited in ordering and consensus. When a transaction has passed from the stateless validation check, it is sent to the ordering nodes and if a number of transactions are accumulated or if a specific amount of time has elapsed, a batch that contains the proposed transactions is sent from the ordering service to the peers for the stateful validation. When the peers receive this proposal, they verify its contents by performing the stateful validation and then create a block containing only valid transactions. Afterwards, this block is voted through signing and if it receives more than 2/3 of the total network's votes, the leader sends a commit message declaring that this block should be accepted to the peer's chains. In terms of performance, YAC can achieve high throughput of several thousand transactions per second, with each transaction to be confirmed in less than 3 s.

#### 6.16. Stellar consensus protocol (SCP)

SCP [115] operates as a provably secure mechanism of federated byzantine agreement to address malicious behaviour across the Stellar network. The protocol is tailored towards financial purposes using quorums and quorum slices. Quorum is the set of nodes that attempt to establish agreement on each state's validity, once a specific threshold of nodes is met; and quorum slices are subsets of the quorum associated with and trusted by a given node, meaning that each node can rely on multiple slices to collect information. The SCP consists of two primary sub-algorithms, the nomination protocol and the ballot protocol. In each consensus slot, the nomination protocol is executed first, in which the new candidate values are produced and disseminated to all the quorum nodes that can vote only for a single one of them. With the termination of this process and the values to be unanimously selected for the current slot, the ballot protocol is initiated and includes a federated voting for either accepting or discharging the generated values that were created in the nomination protocol. In scenarios, where agreement cannot be reached (such as split votes), the nomination and the ballot protocol contain some high-level details to resolve these issues. For example, the execution of a higher valued algorithm that can serve the same purposes and be considered as the execution of a new ballot protocol.

In terms of performance, SCP caps out 1000 TPS with each transaction to be confirmed in 3–5 s; and in terms of security, in a symmetric system with  $N$  nodes and any set of  $T > N/2$  nodes to constitute a quorum, a system designed to survive  $f$  failures will be comprised of  $N > 3f + 1$  nodes and a quorum size of  $T = 2f + 1$ .

### 7. Blockchain security

Blockchains are inherently secure by design, however, numerous recent security studies have proved that this new technology could be susceptible to various cyber-security attacks aiming at breaking its integrity and availability [34,116,117]. Just like any software program, blockchain possesses its own vulnerabilities that can be exploited by attackers to gain unauthorised access to the system and perform malicious operations [116,117]. However, successfully exploited vulnerabilities in blockchain can have serious impact, as they can provide unauthorised access not only to the data stored at the compromised point but also to all data recorded on the distributed ledger, which could have wider negative consequences on the state of the blockchain. Some of these vulnerabilities are specific to particular blockchain platforms, while the others are general. In this section, we focus on the attack surface of the blockchain platforms whose consensus protocols were analysed in Section 6, along with the security mechanisms that these platforms provide to address potential cyber-threats. Consensus-based attacks are not included in this evaluation as they are treated in Section 6.



**Table 2**  
Attacks on blockchains and indicative countermeasures.

	Blockchain attacks	Countermeasures
Identity	■ Replay attacks	■ Elliptic curve encryption
	■ Key attacks	■ Elliptic curve encryption
	■ Impersonation attacks	■ PKC key pairs' freshness
	■ Sybil attacks	■ Elliptic curve encryption
Service		■ Signcryption
		■ Attribute-based signatures
		■ New identity creation cost
		■ Digital certificates
	■ Refusal-to-sign	■ Separation of transactions
	■ DDoS attacks	■ ECDSA ring signatures
Manipulation		■ Block size limitation
	■ Tampering attacks	■ Attribute-based signatures
	■ Overlay attacks	■ Multi-receivers encryption
	■ MiTM attacks	■ Distributed SDN architecture
	■ Eclipse attacks	■ PKC signatures
		■ Elliptic curve encryption
Reputation	■ Whitewashing attacks	■ Timestamps for uniqueness
		■ Elliptic curve encryption
Crypto	■ Quantum attacks	■ Secure mutual authentication
	■ Brute force attacks	■ Randomness in choosing peers
		■ Secure channels
		■ Low priority to newcomers
		■ QC / PQC techniques
		■ QC / PQC techniques
		■ Cryptographic keys' sizes

As illustrated in Table 2, cyber-attacks against blockchain platforms are grouped into the following categories, identity-based attacks, manipulation-based attacks, service-based attacks, reputation-based attacks, and cryptography-based attacks, which are analysed in the following subsections.

### 7.1. Identity-based attacks

In attacks under this category, the adversary aims to forge its identity, masquerade itself as a legitimate user, gain access to the targeted system and manipulate it [116]. Such type of attacks exploit weaknesses related to the security of the cryptographic keys. Any possible leakage of the cryptographic keys, can be used by adversaries to control the peer's identities and undermine the consensus process. The most interesting attacks in this category are the key, replay, impersonation, and the Sybil attacks. The replay attacks aim at spoofing the identities of two parties, intercept their data, and replay them to their destinations [116]. Such attacks occur due to the confusion that some nodes may experience after a hard fork and often within insecure cryptocurrency-based protocols. Various blockchain platforms, such as the Exonum [110], Bitcoin [1], Ethereum [84], Elastico [86] and Parity [104] are vulnerable to security attacks that might fork the blockchain [118]. Currently, Monero does not provide any security mechanism against the replay attack [28]. Corda [97], Quorum [96], Hedera [21], Ripple [119] and Ouroboros [89] proposed mitigation mechanisms, such as secure TLS mutual authentication and *elliptic curve cryptography* (ECC) to calculate the hash functions [120].

Key attacks exploit weak key schemes. Any inappropriate use or storage of the keys can allow unauthorised users to take control of the identities of the participating nodes. Especially, with the advance of the quantum computing, cryptographic keys can be easily cracked, leading to the unauthorised control of the victim's identity [116]. Similarly, in impersonation attacks, weak or leaked private keys can be exploited by adversaries to masquerade themselves as legitimate users and perform unauthorised actions in the system. For instance, they can be impersonated as "authorised block creators" and insert fake data to the blockchain. Unfortunately, the protection measures against these attacks in most blockchain platforms are not very robust [28]. For instance, Ethereum [84], Quorum [96], Parity [104] and Corda [97] are vulnerable to key compromise attacks, since the account keys

are generated using elliptic curve cryptography, which is proved to be weak against these type of attacks [120]. To deal with these attacks, Monero uses the One-Time Private Key (OTPK), which can be used only one time in conjunction with some well-established cryptography method [121]. IOTA replaced some uses of Curl hash function, which is vulnerable to signature forgery attacks, with a new hash function named Kerl [122].

In a Sybil attack, the adversary exploits leaked keys to create a large number of fake identities that can act as authenticated nodes and perform malicious interactions to intentionally increase or decrease the reputation scores of the targeted nodes [117]. BitShare, Lisk and EOSIO [123] cannot prevent this attack as an anonymous or pseudonymous party can participate in the validation and creation of blocks [117]; Tayebbeh et al. [124] found that Elastico is vulnerable to this attack. IOTA [125] is also prone to sybil attacks, since the malicious nodes can create random transactions to select a tip and thus, process invalid transactions. In Ripple, Stellar and Quorum the authorised participants must be authenticated to avoid a Sybil attack [118]. In blockchain platforms that implement a PoW-based consensus protocol, such as Litecoin [82], Monero and Ethereum [84] the malicious node, who wants to create a fake identity should divide its mining power to the corresponding number of sibyls and therefore, reduce its possibility to find the next block. However, Ethereum [84] is still vulnerable to Sybil attacks due to weak restrictions on the node generation process. Tezos [62] and Hedera [21] implement security mechanisms to avoid Sybil attacks, however, Hedera is considered to be prone due to the anonymous individuals that can participate in the network [21].

Exonum [110] has provided increased security against these attacks by using an anchoring service that writes the hash of the current Exonum state to the Bitcoin, in a certain time interval. The anchored data is authenticated by the majority of the validators using digital signatures. Therefore, even if the attacker compromises all the validators, it is impossible to change the transaction log unnoticeable. NEO [57] supports digital certificates, which can solve the potential problem of dishonest nodes in the public blockchain. In order to enhance keys' security, *hardware security modules* (HSMs) have been proposed to securely generate, protect, and store private and public keys [126].

## 7.2. Service-based attacks

This category involves attacks that aim to disrupt the normal operation of the blockchain network, making it unavailable for the legitimate users, or forcing it to operate in a different way from its specification [116]. There are two different types of attacks in this category and are: the *refusal-to-sign*, and the *denial-of-service* (DoS) or *distributed DoS* (DDoS) attacks [116,127]. In a DDoS attack, the attacker typically leverages a number of hijacked devices to flood the network with an excessive number of requests that can disrupt the normal operation of the blockchain, degrade its efficiency, or make it unavailable for legitimate users. Most discussed platforms, including Bitcoin [1], Monero [83], Litecoin [82], Parity [104], Ethereum [84] and the Hyperledger frameworks, Fabric [47], Burrow [109], Iroha [114] and Sawtooth [111], are vulnerable to DDoS attacks, which become the most common attacks on the blockchain ecosystem [117,128]. Particularly, blockchain platforms with a centralised structure are prone to this attack [129], e.g., Quorum (i.e., the structure of Quorum slices is highly centralised) and Stellar, which is highly depended on the structure of the quorum slices [130]. In the same context, a recent study found that over 30% of the users in EOSIO correspond to botnets, with more than 300 attacks that have already been detected [118].

In IOTA [9], the DDoS attack is prevented by requiring from the devices to issue a transaction and execute PoW, a requirement that needs only a few seconds, depending on each device specs; and ensures that a given device cannot send multiple transactions in a short amount of time. Hedera [21] is also DDoS resilient, since it does not provide to any node or to any small number of nodes special rights or responsibilities in consensus process. Even if a DDoS attack floods a miner and temporarily disconnects it from the network, the community as a whole, will continue to operate normally. Liu et al. [131] proposed a combination of ring-based signatures with the ECDSA in order to secure the transfer of coins between the user's addresses. Bitcoin and Bitcoin-NG include some protection against DoS attacks by limiting each block's size and the maximum number of the attribute signatures for the transaction input, but these platforms are still vulnerable to more sophisticated DoS attacks [128].

Refusal-to-sign attacks occur when a malicious node refuses to sign an unfavourable transaction. In this context, malicious nodes, especially in Bitcoin, may pollute the blockchain with refusal to sign attacks and therefore decrease the trust in the blockchain network. None of the aforementioned blockchain platforms can provide a concrete solution to this issue. A simple way to address this attack is proposed by TrustChain [132], in which the interactions that derive from such nodes are blocked and the transactions are split into smaller amounts.

## 7.3. Manipulation-based attacks

This category incorporates attacks that illegally attempt to intercept, manipulate or destroy sensitive data in transit or at rest. Most common attacks are the overlay, the Man-in-the-Middle (MiTM), the eclipse and tampering attack [127]. Overlay attacks exploit blockchain vulnerabilities to maliciously enclose an encrypted amount to an original transaction by using the receiver's public key [116]. Protection against this attack can be guaranteed through the verification of the timestamps used to ensure uniqueness of the transactions. Therefore, the different inputs under the same trader can be identified and associated with the different transactions [116].

MiTM attacks exploit vulnerabilities like private key leakage to spoof identities of two parties and secretly intercept and even modify their communication, while they believe they communicate with each other in a secure channel [116]. Most blockchain frameworks include some form of endpoint authentication to prevent MiTM attacks. However, some of them are still vulnerable to this attack such as Ethereum and Bitcoin. For instance, Hedera, Quorum, Fabric, Corda, Sawtooth and Ouroboros platforms prevent this attack by using TLS mutual

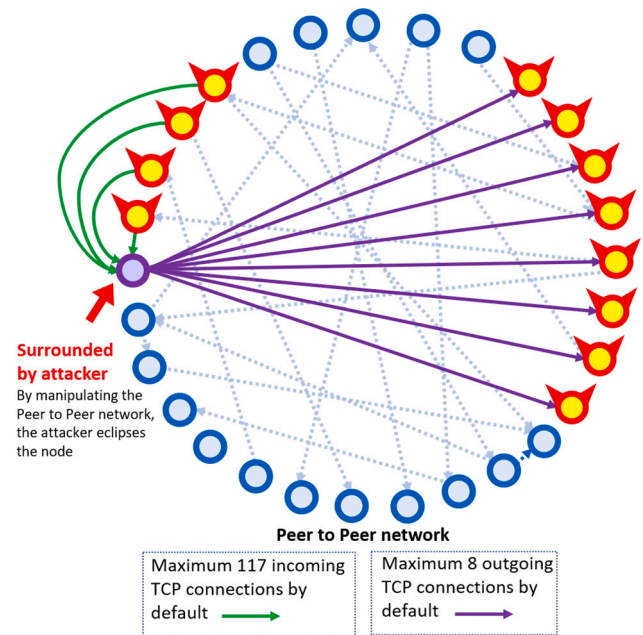


Fig. 3. Eclipsing a Bitcoin node: The nodes have a limited number of outgoing connections, therefore, the victim is forced to establish connections only with the malicious nodes that were created by the attacker [127].

authentication among the nodes. All communication among Fabric nodes is configured to use TLS, however, it is still vulnerable to MiTM attacks and SSL Stripping, which is a type of an MiTM attack, due to the centralisation of the MSP admins and the OSNs [129]. In IOTA [122], messages are transferred over a TLS-encrypted TCP/IP connection with mutual certificate controls to prevent these attacks. On the other hand, Exonum [110] includes a lightweight client that provides additional security against MiTM attacks and maliciously acting nodes. Tezos [62] also supports secure connections via TLS v1.3 and v1.2 with forward secrecy based on elliptic curve Diffie–Hellman key exchange. However, Bubberman et al. [133] found that it is possible to conduct a TLS MiTM attack on the Ripple XRP ledger to gain access to message content.

Eclipse attacks attempt to isolate the victim node by monopolising all of its incoming and outgoing connections. This allows the attacker to corrupt the victim's view of the blockchain, force it to waste computing power or leverage the victim's computing power to conduct its malicious acts. Eclipse attacks can trigger other attacks like Selfish mining, 51% attacks, etc. [28,127]. Research works [117,134–136] proved that Bitcoin (see Fig. 3), Litecoin, Bitcoin-NG, Monero and Ethereum are vulnerable to this attack. However, some security defences have been taken to deal with the eclipse attack, but these defences are rather inadequate. For instance, Bitcoin [1] enhanced the randomness by choosing IP address from the tried table. Work by Zhang et al. [137] have also demonstrated the feasibility of launching Eclipse based stake-bleeding attacks against blockchain systems that are based on PoS like Snow-White and Ouroboros. This attack is a combination of the Eclipse and the stake-bleeding attack. The combination of a virtual voting with a gossip-to-gossip protocol in Hedera is not adequate to provide resistance, but by using secure communication channels its impact can be reduced. In addition, implementing defences against Sybil attacks can bound the portion of overlay nodes controlled by the attacker [138].

Tampering attack is one of the main security risks to the blockchain platforms. In this attack scenario, the attackers attempt to modify the signed transactions being disseminated in the network, the addresses and other data, and then propagate the tampered transactions to the peer-to-peer network for validation. Bitcoin [1] is prone to this kind

**Table 3**  
Weights assigned to the attacks considered.

$w_{ik}$	Attack ( $Att_{ik}$ )	$w_{ik}$	Attack ( $Att_{ik}$ )
0.30	Replay attacks	0.20	Key attacks
0.25	Impersonation attacks	0.25	Sybil attacks
0.50	Refusal-to-sign attacks	0.50	DDoS attacks
0.20	Tampering attacks	0.20	Overlay attacks
0.40	MiTM attacks	0.20	Eclipse attacks
1.00	Whitewashing attacks		

of attacks. Most discussed platforms try to follow NIST's guidelines for using cryptographic standards to avoid weak and non-secure schemes in order to prevent such attacks. In this context, Paillier encryption scheme has been proposed to preserve the Bitcoin transactions privacy and hide the plaintext amounts in the transactions [139]. Tampering attacks could also delay the propagation of transactions and blocks to specific nodes [140]. PoW cryptocurrency-based ledgers like Bitcoin, Litecoin and Monero are more vulnerable to this attack [117]. Arthur et al. [141] found that attackers can provoke significant delays in the broadcast of Bitcoin packets by introducing congestion in the network or making a victim node busy by sending requests to all its ports.

#### 7.4. Reputation-based attacks

In this category, attackers attempt to manipulate the reputation of specific nodes in the blockchain network by intentionally decreasing or increasing their reputation. In this context, a whitewashing attack is a big concern for the blockchain community. In this attack, nodes with poor reputation, exploit some system's vulnerabilities to rejoin the blockchain with new identities and update their reputation. Currently, there is no concrete solution against this attack [127], only TrustChain [132] provides lower priorities and capabilities to nodes with new identities.

#### 7.5. Cryptanalytic attacks

The robustness of blockchain platforms highly depends on the public-key cryptography and the hash functions, such as the RSA encryption scheme and the elliptic curve cryptography. However, using mathematical encryption methods could no longer guarantee the security of these platforms, especially with the fast progress of quantum computing. This technology will allow the implementation of algorithms that can efficiently solve classes of mathematical problems that cannot be practically solved today. In fact, a quantum computer is able to break most of the current encryption algorithms. With these features, brute-force attacks will only take a few minutes to crack the encryption and easily retrieve the encryption keys. As illustrated in Table 6, some blockchain platforms like Bitcoin-PQ, IOTA and Corda have implemented quantum and post-quantum cryptographic techniques to tackle the quantum threat. Some other platforms, such as NEO, Algorand and Fabric are currently transitioning to a quantum resistant infrastructure. However, in a very recent implementation [142], a hybrid quantum-safe Fabric network has been presented, in which hybrid digital signatures are incorporated and provide security against both classical and quantum attacks. Quantum cryptography uses the principles of quantum mechanics to encrypt data and make it impenetrable, while, post-quantum cryptography refers to the security against an attack by a quantum computer.

#### 7.6. Synopsis

Next, we provide a summary of the analysis carried out in the previous subsections and derive an overall security score (see Table 6) that expresses the ability of the blockchain platforms to mitigate attacks,

in each of the aforementioned categories. In particular, the score  $sec_j$  achieved by platform  $j$  is given by the following expression

$$sec_j = 0.35R_{j1} + 0.20R_{j2} + 0.35R_{j3} + 0.10R_{j4} \quad (1)$$

where  $R_{ji}$  is the resistance score of the  $j$ th platform against the  $i$ th attack category; that is, the index  $i = 1, \dots, 4$  is associated with *identity*, *service*, *manipulation* and *reputation* based attacks. Due to the different means of performing cryptographic attacks, these are separately considered in Table 6 and the overall comparative evaluation of Section 10. The weights given in (1) were based on the estimated impact of each attack category; for example, the impact of manipulation-based attacks (e.g. tampering attacks) is high, as it affects the integrity of the data eventually stored in the shared ledger. The value of  $R_{ji}$  is obtained via

$$R_{ji} = \sum_k w_{ik} S_j(Att_{ik}), \quad i = 1, \dots, 4 \quad (2)$$

where  $w_{ik}$  is the weight of attack  $Att_{ik}$  within the  $i$ th attack category and  $S_j(\cdot)$  is a discrete-valued function quantifying the security score achieved by platform  $j$  against a specific attack. The weights assigned to each attack are summarised in Table 3 (again, excluding cryptanalytic attacks), while the possible scores given by function  $S$  are defined as

$$S_j(Att) = \begin{cases} 0.1, & \text{if platform is vulnerable to Att,} \\ 0.5, & \text{if platform has defences against} \\ & \text{Att but are rather inadequate,} \\ 1.0, & \text{if platform is robust against Att.} \end{cases} \quad (3)$$

The obtained security scores  $sec_j$  for all platforms (i.e. all  $j$ ) are presented in Table 6. However, to ease presentation, two typical examples are presented next, to clarify how each platform's overall security scores is calculated and a three-level scale *low*, *medium*, and *high* has been used; the level for each platform is computed as follows.

$$L(sec) = \begin{cases} \text{Low,} & \text{if } sec < 0.3, \\ \text{Medium,} & \text{if } 0.3 \leq sec \leq 0.7, \\ \text{High,} & \text{if } 0.7 < sec. \end{cases} \quad (4)$$

Therefore, *high* level indicates that the blockchain platform offers a complete security solution to address most of the attacks discussed; *medium* level is assigned to the platforms addressing a sufficient subset of the attacks, while *low* indicates that the blockchain under consideration is vulnerable to most of the attacks discussed here.

**Example 1.** Algorand, is not found to be robust against any of the aforementioned attacks, but vulnerable to sybil, refusal to sign and whitewashing attacks. The platform also provides defences against replay, key, impersonation, DDoS, tampering, overlay, eclipse and MiTM attacks, but these defence mechanisms are rather inadequate. Hence, we need to calculate (2) for  $j = 1$ , which corresponds to Algorand; the scores obtained for each attack category ( $i = 1, \dots, 4$ ) are

$$R_{11} = w_{11} 0.5 + w_{12} 0.5 + w_{13} 0.5 + w_{14} 0.1 = 0.4$$

$$R_{12} = w_{21} 0.1 + w_{22} 0.5 = 0.3$$

$$R_{13} = w_{31} 0.5 + w_{32} 0.5 + w_{33} 0.5 + w_{34} 0.5 = 0.5$$

$$R_{14} = w_{41} 0.1 = 0.1$$

Hence, evaluating (1) for Algorand gives the overall resilience score of

$$\begin{aligned} sec_1 &= 0.35R_{11} + 0.20R_{12} + 0.35R_{13} + 0.10R_{14} \\ &= 0.39 \end{aligned}$$

which corresponds to a medium level of security from (4) ■

**Example 2.** Fabric, on the other hand, is found to be robust against tampering and overlay attacks, as well as, vulnerable to refusal to sign, DDoS and whitewashing attacks. The platform also provides defences

**Table 4**  
A taxonomy of vulnerabilities in smart contracts [143].

	Vulnerability	Cause
Source code	Call to the unknown	The called function does not exist
	Out-of-gas send	Fallback of the callee is executed
	Exception disorder	Exception handling irregularity
	Type casts	Contract execution type check error
	Re-entrance flaw	Function re-entered before exit
	Field disclosure	Private value published by peer
Ethereum virtual machine	Immutable bug	Contract change after deployment
	Ether lost	Send ether to orphan address
	Stack overflow	Number of values in stack > 1024
Blockchain	Unpredictable state	State modified prior invocation
	Randomness bug	Seed biased by malicious peer
	Timestamp dependence	Timestamp modification by a malicious peer

against replay attacks, key attacks, impersonation, sybil, eclipse and MiTM attacks, but these defence mechanisms are rather inadequate. Hence, we need to calculate (2) for  $j = 2$ , which corresponds to Fabric; the scores obtained for each attack category ( $i = 1, \dots, 4$ ) are

$$R_{21} = w_{21} 0.5 + w_{22} 0.5 + w_{23} 0.5 + w_{24} 0.5 = 0.5$$

$$R_{22} = w_{31} 0.1 + w_{32} 0.1 = 0.1$$

$$R_{23} = w_{21} 1 + w_{22} 1 + w_{23} 0.5 + w_{24} 0.5 = 0.7$$

$$R_{24} = w_{41} 0.1 = 0.1$$

Hence, evaluating (1) for Fabric gives the overall resilience score of

$$\begin{aligned} \text{sec}_2 &= 0.35R_{21} + 0.20R_{22} + 0.35R_{23} + 0.10R_{24} \\ &= 0.45 \end{aligned}$$

which corresponds to a medium level of security from (4) ■

The analysis shows that blockchain 1.0 and 2.0 platforms (e.g. Bitcoin [1], Ethereum [84], Litecoin [82], etc.) have relatively *low* security levels as they are subject to a number of security attacks such as DDoS, sybil, replay, eclipse, MiTM, and tampering attacks. Study in [144] affirmed that several currency exchanges have been shut down as a result of DDoS attacks. In addition, the Sawtooth [111] platform lacks adequate security mechanisms against the considered attacks. Most of the surveyed blockchain platforms achieved *medium* security levels, since they adopt some level of protection mechanisms against the aforementioned attacks. However, none of these mechanisms is adequate enough to address them. For instance, the blockchain platforms that implement the DPoS consensus protocol, such as BitShares [51], Lisk [91], Tezos [62] and Tron [62] are vulnerable to centralisation, while the risk of DDoS, sybil and eclipse attacks is increased due to the strictly limited number of witnesses. Amongst the discussed platforms, Hedera [21] achieves *high* security level, as it provides protection against most of the considered attacks [145]; however, its security properties need further investigation.

## 8. Smart contracts' security

Security is a critical requirement in the context of a smart home. Within such a resource restrained environment it is excessively difficult to enforce security by increasing the key sizes with multiple cryptographic operations. The IoT devices are restricted in the form of storage capabilities and computational power. Therefore, the use of digital public key certificates can be an expensive operation for the smart home's IoT devices. The systems based on *public key infrastructures* (PKI) might demand verification via the requests sent to the cloud servers which will create the network's traffic. In this case, the privilege definition and the access control on the centralised servers will be vulnerable.

Blockchain technology can provide the functionality of executable logic via smart contracts. This functionality can be used to handle the

interactions of a smart home ecosystem as transactions, by issuing commands to the IoT devices and ensure confidentiality. The smart home IoT devices can trigger the smart contract functions when some specific conditions are satisfied and call them with addresses or prompt them as application reaction to listening events. Therefore, the smart contracts can execute security operations, like authentication and access control, as well as, to enhance the overall security of a blockchain-based smart home ecosystem. Within this ecosystem, the service provider and the IoT data owner can interact as transacting entities without a trusted third party. With the use of smart contracts, the granularity of the IoT data being shared is in the control of the data owner, who can determine exactly the amount, the type and the time-span of the data to be shared. Therefore, smart contracts can be used to grant or revoke access privileges and safeguard the home-owner's privacy.

### 8.1. Smart contracts' vulnerabilities

Upon its invocation, the smart contract code is stored on the blockchain, while the smart contracts' operations can be called at any time by any participating node. The smart contracts are often called as autonomous agents, due to the fact that they are being given their own accounts on the network and their own network addresses. Therefore, their implementation can hold ownership of tokenised assets or custody, while the engaging parties are complied with the agreed-upon conditions. Their invocation often incurs an execution fee, since each operation is considered as a transaction logged into the blockchain platform. The smart contracts can be used to execute a variety of operations within the blockchain network, such as to: (a) enable multi-signature transactions, where each transaction is only executed when the majority of the participating entities sign it; (b) enable automated transactions generated by unique events; for instance, transactions that are automatically disseminated over specific time periods or transactions that are disseminated in response to other transactions; (c) provide services to other smart contracts; for instance, in occasions where a smart contract invokes operations from another smart contract, and (d) allow storage space for application-specific information, such as Boolean states, lists, membership records, etc.

Though it is hard to modify records stored on a distributed ledger, it is possible to compromise the software systems implementing the technology; the hack of Mt. Gox, resulting in \$450 million losses, is a notable example. Another incident is related to the *decentralised autonomous organisation* (DAO), holding a large percentage of Ether; it suffered ~\$60 million in losses when a smart contract vulnerability was exploited leading to an infinite recursive calling situation that blocked the invocation of the function updating a user's balance. A summary of vulnerabilities in smart contracts is provided in Table 4. Most of these vulnerabilities apply to Solidity, which is an object-oriented, high-level language programming language supported by Ethereum [143]; this is due to the misalignment between the language semantics and programmers' intuition, and the fact that Solidity does not have a structure to deal with domain-specific attitudes in public blockchains



**Table 5**

A taxonomy of verification methods for smart contracts.

	Method	Framework	Level
Smart contract verification	Theorem proving	■ Isabelle/HOL	Bytecode
		■ K - Framework	Bytecode
		■ F* - Framework	Bytecode & Solidity
		■ FSPVM-E	Solidity
		■ EtherTrust	Bytecode
	Model checking	■ Zeus	Solidity
		■ VERISOL	Bytecode & Solidity
		■ VeriSolid	Solidity <sup>a</sup>
		■ ContractLARVA	Solidity
	Symbolic execution	■ Oyente	Bytecode
		■ teETHER	Bytecode
		■ MAIAN	Bytecode
		■ Gasper	Bytecode
		■ sCompile	Bytecode
		■ Osiris	Bytecode
■ Mythril		Bytecode	
■ Slither		Bytecode	
Abstract interpretation	■ MadMax	Bytecode	
	■ Securify	Bytecode	
	■ Vandal	Bytecode	
	■ EthIR	Bytecode	
	■ EtherTrust	Bytecode	

■ Static analysis is performed in a non-runtime environment and without executing the contract, but by examining the source code for signs of security flaws.

■ Dynamic Analysis is performed while the contract is executed by examining it and trying to manipulate it in order to discover security flaws.

<sup>a</sup>As transition system.

leaving them open to reordering or delaying. Yet another cause for the insecurity of smart contracts is the lack of a single source of documentation for known vulnerabilities, being comprehensive, self-contained and regularly updated, so as to avoid past mistakes.

## 8.2. Smart contracts' verification

As a result of the severe consequence that can occur from the insecurity of the smart-contracts, researchers have adopted several techniques and tools that detect these vulnerabilities. Theorem proving, symbolic execution, model checking and abstract interpretation can be used to improve the security aspect of smart contracts. Therefore, in this section, we investigate methods and techniques that have been used for their verification and their improvement. The characteristics of these frameworks and tools are summarised in Table 5. The majority of them choose static analysis and support EVM bytecodes analysis.

### 8.2.1. Theorem proving

*Theorem proving* is the most ordinary method to formalise smart contracts. The system is modelled mathematically, while the aimed properties to be proven are stated, with the verification to be executed on a computer by a theorem prover software. The theorem prover applies well-known axioms and simple inference rules for the verification process with each new theorem or lemma that is required for the proof can be derived from them. Theorem proving can be considered as an adaptable verification method, due to its suitability on all the systems that can be expressed in a mathematical way.

Theorem provers can be automated, interactive or hybrid [146]. The automated theorem provers perform the proving automatically, while the interactive might need some human input. Whether it is going to be used automated theorem proving or interactive, it is depended on the system's complexity, but for both, manual set up is required for the specifications and the system model. The hybrid theorem proving allows the partitioning of the system's model based on the complexity level. Then most complex parts are verified with interactive theorem proving, while the least complex parts are verified with automated. For some simple systems, statement logic or propositional logic can be adequate to verify the smart contract.

The most expressive logic for theorem proving is *higher-order logic* (HOL), which enables the use of quantifiers over sets and functions.

Several frameworks also exist in the context of the smart contracts. The authors in [147], used an existing EVM-formal model and the logical framework Isabelle/HOL to propose a logic for EVM bytecode. The authors in [148], used formal verification of smart contracts with the K-Framework for expressing the EVM bytecode using logic. They created a simple smart contract on the Ethereum platform, wrote the high-level and the EVM-level specifications, and then verified this smart contract. Ether-Trust [149] is the first static analyser, which is developed by Grishchenko et al. to detect the re-entrancy vulnerability. FSPVM-E [150], which stands for *formal symbolic process virtual machine*, verifies the security and reliability of Ethereum-based services at the source code level of smart contracts. This verification system is formulated in Coq and is comprised of four primary components: (1) the formal memory model (denoted as GERM); (2) the formal specification language for Solidity (denoted as Lolisa); (3) the execution and proof engine (denoted as FEther) and (4) the assistant tools and libraries that enhance the automation's degree and the validation efficiency of the system. F\* [151] is a high-level general-purpose programming language, which is built for formal verification. The authors in [152], made an effort to formally verify Ethereum smart contracts, by translating the contract code to F\*.

### 8.2.2. Model checking

*Model checking* is an automated technique for formal verification, applicable for systems that can be expressed by a finite-state model. The verification is performed using a model checking software, such as NuSMV. The user provides the model checker with a finite-state model of the system and a formal specification that it should possess. Then the model checks automatically if every state of the model satisfies the given specification. For the model to be formally verified for a specific property each state of the model must satisfy the specification [153]. If the specification is not satisfied, then a counter-example is given by the model checker, which shows a run of the system that violates the specification. This can help the user to identify mistakes and to correct bugs.

Model checking has been mostly used on Solidity, but it can verify contracts written in other languages, as well. ZEUS [154] is an automatic framework for formal verification by using abstract interpretation and symbolic model checking to soundly reason about program behaviour. The smart contracts, which are written in high-level languages,

are given as input to the ZEUS framework that uses the user's assistance to help generate the correctness or/and the fairness principles in a XACML-styled template [155]. *Verifier for Solidity* (VERISOL) is a formal verifier for smart contracts (written in Solidity), while it is not tied to Workbench. VERISOL encodes the semantics of the smart contracts into a low-level intermediate verification language Boogie [156] and leverages the Boogie verification pipeline not only for verification, but also for counter-example generation. VeriSolid [157] is built on top of FSolidM [158] and WebGME [159] and provides an approach for correct-by-design development of Ethereum-based smart contracts.

Model checking is also well adopted in behaviour-based verification. In [160], the authors based on a formal model checking language, proposed a novel way to model smart contracts, which was a way to model the users' behaviour. In [161], Ellul et al. proposed a runtime verification method to ensure that all execution paths followed at runtime, can satisfy the required specification, which are enclosed in the proposed method, called ContractLarva.

### 8.2.3. Symbolic execution

*Symbolic execution* is a software testing technique that helps to test data generation and proofs regarding the quality of a given program [162]. This technique is a delicate solution, in which possible execution paths are systematically explored simultaneously, without necessarily requiring concrete inputs. Instead of taking on fully specified input values, this technique represents them as symbols, in an abstract way, resorting to constraint solvers to construct actual instances that could cause property violations. Symbolic execution leads to major practical breakthroughs in a number of prominent software reliability applications and it is considered as the most prominent technique for vulnerability detection in EVM bytecode.

Several tools have been proposed that use symbolic execution. OYENTE [163] aims at timestamp dependence, mishandled exceptions, transaction-ordering dependence, as well as, re-entrancy bugs. The tool was used to check the first 1,459,999 blocks of Ethereum and flagged 8833 contracts as vulnerable, out of the 19,366, including the DAO bug. teETHER [164] automatically generates transactions that can exploit the given contracts. MAIAN [165] is a dynamic tool that analyses the invocations of smart contracts to identify three types of trace vulnerabilities: prodigality, greediness and suicidality. Chen et al. [166] developed Gasper [166], a static analysis tool that detects gas-costly patterns instead of vulnerabilities. sCompile [167] reports program paths containing monetary transactions, called critical paths; it discovered 224 unknown vulnerabilities with a false positive rate of 15.4%. Osiris [168], Mythril [169] and Slither [170] are some additional examples that use symbolic execution.

### 8.2.4. Abstract interpretation

*Abstract Interpretation* [171] is a basic static analysis technique that soundly estimates program semantics. Various analysers have used this technique for EVM bytecode, since it can examine any possible execution. MadMax [172] identifies three gas-related vulnerabilities: non-isolated external invocations, unbounded mass operations and integer overflows. For an input property and a given smart contract SECURIFY [173] checks the smart contract behaviours with respect to the property. The analyser works in two phases: first precise semantic information from the code is extracted and then the compliance and violation patterns are checked to prove if a specific property holds. Vandal [174] leverages the Soufflé language [175] to generate results for custom vulnerability queries. EthIR [176] is a static analyser for Ethereum bytecode that relies upon an extension of the Oyente tool. ZEUS [154] and EtherTrust [149], mentioned earlier, are frameworks that also use abstract interpretation.

## 9. Blockchain privacy

In principle, a public blockchain platform raises more privacy concerns than a private. Similarly, in the latter case, it is easier to cope with privacy issues in a permissioned setting than in a permissionless one [71]. However, the form of the platform is actually contingent on the specific application; in other words, each application determines, by its nature, which type is more appropriate. For example, in the discussed case of integrating the blockchain technology to a smart home ecosystem towards facilitating and enhancing security services, the home-owner's personal data (including user's device identifier data) should not be public and, moreover, any access on them should satisfy the data protection requirements (i.e. on the basis of the user's explicit informed consent or some other legal basis). Clearly, in this case, a private permissioned blockchain network takes precedence over the other approaches. On the other side, other privacy requirements occur in other cases (e.g. in cryptocurrencies offered to the public, in which anyone can contribute in adding transactions to the ledger, a private blockchain network is not an option — and, hence, this in turn poses other data protection requirements). Blockchain technology can also be used to enhance the transparency of the user's communication, via providing an audit trail of the underlying data processes; again, in such scenarios, other data protection requirements may be present.

In any case though, several privacy enhancing mechanisms exist, to alleviate privacy issues arising by the use of the blockchain technology as the means to process personal data. As already stated in Section 4.3, the necessity of whether each of these technologies should be implemented or not can be assessed only on a case-by-case basis, since each of them focuses on a specific privacy threat which may or may not be present, depending on the specific application and implementation decisions. We next present these techniques, with a discussion to the relevant blockchain platforms in which are being met.

### 9.1. Pseudonymisation

Privacy mechanisms on pseudonymisation are available in all of the blockchain platforms shown in Table 6 and these mechanisms are used to protect individuals' identities and allow them to retain some utility on their data. According to the GDPR, pseudonymisation is the processing of personal data, in such a manner that the personal data can no longer be attributed to a specific individual without the use of additional information, provided that such additional information is kept separately and is subject to technical and organisational measures to ensure that the personal data are not attributed to an identified or identifiable natural person. In typical scenarios, pseudonymisation is achieved by replacing users' identifiers by pseudonyms - i.e. specific type identifiers that do not allow, by themselves, re-identification of the individuals.

Blockchain platforms that are being used for cryptocurrencies (e.g. Bitcoin, Ethereum) employ pseudonymisation techniques by default, since each user having a wallet is uniquely associated with a random-looking address (which is generated through cryptographic mechanisms and plays the role of a pseudonym). Therefore, it can be seen that in public financial transactions, amounts of cryptocurrency being sent by one address to another, whereas there is no direct way to map the address back to an identified individual (typically, these addresses are derived by cryptographic public keys). In a similar way, any blockchain platform, for any application, may store pseudonymous data on the ledger, if this is necessary with respect to personal data protection requirements within a specific context; such a pseudonymisation may take place offline, so as to assign pseudonyms which in turn are being used in any blockchain transaction (e.g. pseudonyms may be assigned to user's devices' identifiers in a smart home ecosystem). This offline process allows for providing the additional information required to reverse the pseudonymisation, if such a pseudonymisation reversal is needed.

It should be pointed out though that pseudonymous data are still personal (and not anonymous) data. Moreover, depending on the pseudonymisation technique and other information being available, a third party may manage to identify an individual by her/his pseudonymous data. For example, in the case of the classical Bitcoin, it is known that a former federal agent managed to trace 3760 bitcoin transactions over 12 months [177], finding out the person behind a specific Bitcoin address (i.e. pseudonym). As also stated in [177], the possibility – under specific preconditions – of tracing back the Bitcoin transactions to an individual exists due to the facts that: (i) it is a public blockchain, (ii) the public address of each user is always the same (public key-based address).

### 9.2. One-time address

To alleviate some issues arising from the usage of one pseudonym per user, a user may be associated by different pseudonyms (or addresses) to prevent linking transactions corresponding to a single user. To this end, using one-time addresses will quickly result in a large number of addresses per participant. Privacy mechanisms relying on one-time address are available in Bitcoin, Ethereum, Monero, as well as, in Ripple.

Such a characteristic example in blockchain technology is the use of the so-called *stealth addresses*; this notion has been first introduced for financial blockchain platforms, aiming to protect payee's privacy (as in the case of the Cryptonote technology [178]). More precisely, by appropriately using cryptographic mechanisms, such as the Elliptic Curve Diffie-Hellman algorithm, the payer creates a one-time address for every transaction with a specific payee, in order to enhance unlinkability for the latter. This is the case of Bitcoin. Apparently, techniques of this form could be also employed in other types of blockchain platforms, if such an unlinkability is needed; Ethereum and Monero (the latter being based on the Cryptonote technology) support one-time addresses.

In the case of Ripple, the notion of the so-called blinded tags has been introduced; each such tag consists of a sequence of numbers corresponding to the recipient of a transaction that is meaningful only for the intended recipient and is random-looking for everyone else.

### 9.3. Mixing techniques

Privacy mechanisms relying on Mixing techniques are available in IOTA, Ethereum and Bitcoin; this technique allows multiple participants to shuffle multiple transactions without revealing the exact relationships between the transactions - i.e. they cannot be linked. Again, such a technique was first developed to address privacy issues in cryptocurrencies. The idea of mixing is quite old [179], having been used in several contexts. In blockchain, mixing techniques are used to conceal the history of a particular token (see, e.g. [180]), since each resulting (mixed) transaction that is stored on the blockchain corresponds to multiple senders and receivers — and, thus, rendering difficult for an adversary to single out one transaction. Mixing services (also known as tumblers) could be provided either through a centralised mixing service provider or on a peer-to-peer basis. For the Bitcoin, the Mixcoin protocol has been introduced in [181], being a centralised mixing service, whilst several other mixing services being compatible with the Bitcoin have been also proposed (a recent survey can be found in [182]).

It should be stressed though that the information (transactions) recorded in the ledger is in a readable/interpretable form, when mixing is being used and, thus, some linkability may occur in specific cases.

In the IOTA platform, there is a coin-mixing service being called TangleMixer, with the aim to hide the sources of the transactions. The Ethereum also supports mixing services, like the MixEth [183] and Möbius [184].

### 9.4. Ring signatures

Ring signature is a special type of a cryptographic digital signature; this technique is only available in Monero and provides the following property. Given a group of users (members), a third party can verify that a signature has been put by a member of this group, but she/he cannot discriminate exactly which member is the actual signer [185]. Ring signatures are being considered as a nice privacy enhancing technology in blockchains, in terms of hiding the initiator of a blockchain transaction.

Similarly though to the case of mixing, the transaction information that is stored in the ledger is still in a readable form and thus, in specific cases, this may allow singling out of a signer.

An improved version of the ring signatures yields the so-called ring confidential transaction, which obfuscates the actual amount of the transaction that is being sent from the sender to the receiver. Therefore, only the participants in the transaction are able to see its amount, whilst, at the same time, the network is able to confirm the validity of this transaction. Such a technique is being called as Multi-layered Linkable Spontaneous Anonymous Group signature (MLSAG) and is currently being used in the Monero platform [83] - which is a cryptocurrency based on the Cryptonote technology. However, it seems that MLSAG has not been used yet in the context of IoT applications.

A somehow generalisation of ring signatures is the multi-signatures, which allow multiple signers to jointly authenticate a message using a single compact signature [186]. Although the technology is not new, it has become widespread during the last years in the world of cryptocurrencies. Multi-signature technology was first implemented in Bitcoin since 2012. For example, Ethereum, IOTA, Hyperledger Iroha, Stellar, Tron, Ripple and Algorand blockchains support multi-signatures. However, it should be pointed out that multi-signatures mainly focus on enhancing security rather than privacy, since that the private keys needed to sign/authorise can be spread across multiple machines, eliminating any one of those machines as a single point of failure.

### 9.5. Confidential transactions

Several tools exist to achieve this technique in a blockchain platform; i.e. enabling participants to hide their transaction data from unauthorised third parties (either other legitimate users of blockchains or not), while still the transactions can be executed. Note that this is something different from the aforementioned ring signatures, in which the contents of the transactions are being shown to all. Privacy mechanisms relying on confidential (or private) transactions are available in Quorum, Hyperledger Besu and Sawtooth, Symbiont, Ethereum (Parity), Ouroboros Cryptsinous and Bitshares.

In Quorum, an optional feature of the so-called private transactions exists, allowing a user designate parties to privately transact with and, subsequently, the public ledger stores the hash value of the transaction whereas the original data are stored into private blockchains of the authorised participants. Therefore, all participants share a common public state which is created through public transactions, whereas they also have a local unique private state. If the transaction is private, the participant can only execute the transaction if it has the ability to access and decrypt the payload. Participants who are not involved in the transaction do not have the private payload at all. Hence, a private transactions manager serves as a off-chain privacy mechanism (related with the case of off-chain transactions, described next). Quorum communicates with the private transaction manager using HTTPS and keeps a reference to private transactions with relevant state trees on the blockchain.

In the Hyperledger Besu, a Private Transaction Manager is being used, to ensure that only involved parties can access the contents of transactions. In Hyperledger Sawtooth, there exists the concept of the

so-called private-transaction-families, enforcing a policy of access control to the ledger. As stated in [111], this solution supports encryption of information in transactions and blocks while allowing the ledger to validate the information in those transactions in all the nodes and allow them to reach a consensus on the current state of the ledger.

In the Symbiont ledger, all the data is on the ledger but encrypted such as – despite the fact that all the data is stored on all nodes – a node can only decrypt what it is allowed to see, and likewise, it can just execute transactions that it is able to decode [105].

Parity, which is an Ethereum client supporting standard Ethereum APIs and protocols Technologies, implements a private transaction and smart contract capability in which a private contract's code and state are stored encrypted inside a public contract. The private transaction capabilities sit on top of Parity's existing Parity Ethereum Client [187].

In Bitshare, there is the notion of blinded transfers, in order to hide the balance amount and the identity of the party that is currently in control of the balance. But the transactional history of that balance may reveal details of the balance that can be deduced by inference.

Another way to hide transaction information is through cryptographic commitment schemes, such as the Pedersen commitment [188]. By these means, a sender creates a commitment to a transaction information and share the commitment instead of the information (see, e.g., [189]). The sender cannot lie later on a fake value of the initial information (which remains though hidden). In financial transactions (i.e. in cryptocurrencies), such commitments may allow third parties to verify that the input and output amounts of a transaction are equivalent without revealing them. Other such cryptographic commitments are also being used - e.g. in the Ouroboros Cryptosinus blockchain [190].

Last but not least, a blockchain mechanism called *Mimblewimble* (MW) should be also mentioned, which is a way of structuring and storing transactions – utilising a Pedersen commitment – so as to ensure that all transactions resemble random data to an outsider. The transaction data is only visible to their respective participants. Therefore, MW lies also in the concept of confidential transactions. Litecoin has announced the integration of MW, for enhancing privacy.

## 9.6. Homomorphic encryption

As stated in [189], this technique may be the vehicle to share and perform operations over data without revealing their private values, hence, implementing somehow the aforementioned idea of confidential transactions. Privacy mechanisms relying on homomorphic encryption are available in Ethereum and Quorum.

Due to their inherent properties, homomorphic ciphers allow arithmetic operations over the ciphertexts without having access to the decryption key, so as to ensure that the result coincides with the encrypted version of the value that we would get if the operations were initially performed over the original values. This is the case of the AZTEC protocol in the Ethereum (and also in some implementations of Quorum, as an enterprise-focused version of Ethereum), which utilises zero-knowledge proofs (being described next) as well as homomorphic encryption to encrypt the inputs and outputs of a transaction, so as to ensure that the blockchain can still test the logical correctness of these encrypted statements.

In general, there is a high potential for the integration of blockchain based-IoT with homomorphic encryption. The main task of this encryption in the IoT domain is securing IoT-Data with high privacy in decentralised mode. The task of this description is discussed in [17].

## 9.7. Zero-knowledge proofs

Towards implementing confidential transactions, the so-called zero-knowledge proofs (ZKP) can be also used. A ZKP is a cryptographic protocol with the following purpose: an entity (the prover) proves to another entity (the verifier) that she possesses/knows an information/statement, without exposing anything about the information/statement

itself (e.g. it can be proved that she has at least 100 coins in her account, without revealing the actual amount). ZKPs are known in the cryptographic community for many decades [191]; they are being revisited though in the context of blockchains, being used by transacting parties to create confidential transactions, whose information can be verified without revealing the actual information. In this framework, the Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (known as zk-SNARK) is being mainly used: *succinct* means that the zero-knowledge proofs can be verified within a few milliseconds, whilst *non-interactive* indicates that the proof consists of a single message sent from the prover (i.e. the claimant) to the verifier (and, thus, it is not an interactive protocol). In cases that zero knowledge proofs are implemented, the blockchain reveals only that a transaction has occurred, but not which public key transferred which amount to the recipient.

It has been pointed out that zero knowledge proofs and homomorphic encryption mentioned above have the potential to solve the conflict between data minimisation and the verifiability of data between many parties [71]. It is a highly evolving field, where several improvements of zk-SNARKs are being proposed – such as the less expensive implementation in [45].

Several blockchain platforms employ ZKPs, such as the Ethereum though the so-called AZTEC protocol and Quorum through the zero-knowledge security layer (ZSL) protocol. Ouroboros Cryptosinus [190], which is built upon Ouroboros is also employing zk-SNARKs. The latest version Tron (v4.0) also introduces privacy techniques based on zk-SNARKs, by using relevant code from the well-known cryptocurrency Zcash. In Fabric, privacy-preserving authentication and transfer or certified attributes can be done using Identity Mixer (Idemix) which is a ZKP-based cryptographic protocol [192]. The developers of Stellar have introduced a zero-knowledge virtual machine being called ZkVM, allowing individuals and organisations performing safely their transactions directly on the shared ledger, instead of keeping them in siloed databases. Corda documents also refer to the intention on migrating to ZKP [193], whilst very recently the relevant idea seems to be implemented by utilising Corda in the bank sector [194]. Finally, according to the public information from the Tezos community [62], the Tezos developer community has been particularly interested in enabling private transactions by implementing zk-SNARKs, also based on the relevant implementation of the Zcash. It should be also stressed, as a last remark, that the Zerocoin project investigates how ZKPs can alleviate the privacy issues of Bitcoin.

## 9.8. Blockchain segregation

The notion of blockchain segregation can also be considered as an important privacy enhancing technology, which seems to be prominent for IoT applications; by these means, there is no shared ledger, with all the data (transactions) available to all participants; instead, each participant has access only to a subset of all transactions (let us say in a subset of the whole blockchain). Hence, unauthorised third parties cannot have any knowledge on the existence of transactions for which they should not be aware of.

Privacy mechanisms relying on blockchain segregation are available in several platforms, namely Corda, Hyperledger Fabric, Hyperledger Iroha and EOSIO. In Corda, only a set of predefined and identified participants can be part of a particular communication, while all the other participants in the network remain unaware of the transaction. However, a network service, being called notary, is being used which gets transaction information for several purposes (such as to ensure that double-spending does not occur in the case of cryptocurrency).

The Hyperledger Fabric supports two mechanisms for such a segregation. The first one rests with the so-called channels concept, which allows to separate the information between different channels [195]. The second one is the concept of the private data, which allows to



isolate data between different organisations within the same channel. The channels are associated with the network level, whereas the private data are associated with the chaincode (application) level. Such privacy mechanisms are very important in cases of providing blockchain network and applications into a consortium environment (where consortium is a number of organisations with common business goals).

In the Hyperledger Iroha, access control rules along with some encryption maintain such a segregation. Access control rules can be defined at three levels: user-level, domain-level or system-level. At the user-level privacy rules for a specific individual are defined. If access rules are determined at domain or system level, they are affecting all users in the domain. In general, a role-based access control design has been implemented, where each role has specific permissions. In a somehow similar underlying logic (although fully different in the implementations), EOSIO also supports several permission schemes for the users; each permission is linked to an authority table which contains a threshold that must be reached in order to allow the action associated with the given permission to be executed.

### 9.9. Off-chain transactions/storage

By means of off-chain transactions, we refer to the case of using off-ledger channels to perform transactions, without broadcasting each transaction to the entire network. For example, off-ledger payment channel setups are implemented on public blockchains such as Bitcoin and Ethereum (Lightning and Raiden networks, respectively). This technique can possibly be used in all the blockchain platforms with proper implementation.

Generally we may decide, regardless the underlying blockchain platform, to not store all transactional data on the blockchain itself - and this is something that is expected to occur in IoT environments, due to the personal data protection issues that arise. Rather, such data could be stored in another, off-chain database and merely linked to the blockchain through a hash. Clearly, this process has a number of advantages from a data protection perspective.

For example, in the Hedera blockchain, based on the info in [196], the personal data is encrypted before being submitted to the Hedera Consensus Service. It will not be persisted on the main net nodes and will be persisted in that encrypted form on whichever mirror network nodes keep the messages. The mirror network nodes do not have the relevant decryption keys; these are being held only by the business application network. Also in the Ontology blockchain white paper, there is an explicit reference to the fact that the network structure supports both blockchain ledger storage and off-chain decentralised storage [197]. Exonum has been also used under this principle; for example, in [198] an Exonum-based health data ecosystem is described, divided into two segments — open and closed. The medical data is stored in the closed segment, whereas each patient unique identifier is stored in open segment.

### 9.10. Editable blockchains

The inherent immutability property of the blockchains results in a data protection concern related to the so-called right to be forgotten: data cannot be deleted. Therefore, since from a data protection perspective, there generally exists the right to data erasure which should be satisfied (e.g. if the user's data have been put to the ledger based on her explicit consent, then this consent should be able to be revoked if no other legal basis for maintaining them to the ledger exists), the fulfilment of such a right becomes an important challenge in blockchain platforms. Although there exist exemptions for this right, mainly depending on the legal basis of the processing or other specific circumstances (e.g. the GDPR provides such an exemption in cases that the data are necessary for the establishment, exercise or defence of

legal claims), one should always consider how to handle the case that personal data are to be deleted, if this option may be in place.

Whenever such a data protection right is to be satisfied, a typical approach is to appropriately design the blockchain application so as to avoid storing plaintext data in the ledger but only to an off-line database (see Section 9.9), whereas the blockchain stores an irreversible commitment (e.g. a hash value) of these data. Hence, deleting the data from the off-line database does not allow any subsequent restoration, since the commitment stored in the immutable ledger is irreversible. Therefore, the aforementioned usage of off-line database in Section 9.9 also suffices to practically address this privacy concern (although, there still exist issues with respect to whether keeping only a commitment of the original data is always legally equivalent to deletion of data). However, the research community has started exploring the so-called editable blockchains (also known as redactable blockchains), using chameleon hash functions to edit, rewrite or delete data (see, e.g., [199,200]). Chameleon (or trapdoor) hash functions allow an authorised party to compute a collision with a given hash value, whilst such collisions cannot be found by anyone else if this trapdoor is not known. It has, however, been stressed [71], that since a blockchain becomes editable, its initial advantages of using this solution compared to other forms of databases may be questionable. On the other side, such a solution may be preferable in cases that it is desirable, while preserving the tampering detection property, to explicitly allow someone with higher privileged access like an appropriate authority to be able to change the data or erase it — and this could possibly be the case of a private blockchain for IoT applications [200]. In any case, this is for the moment a new evolving area, not being implemented yet by the main stakeholders in the field.

### 9.11. Synopsis

Table 6, which summarises the main properties of the discussed blockchain platforms, contains also information on the main privacy mechanisms that are being employed in each of them; each mechanism is associated therein with its corresponding number of the subsection that is being described. Note that the pseudonymisation feature (Section 9.1) is present to all platforms. Moreover, the off-chain storage (Section 9.9) is also present to any platform supporting private permissioned solutions (since we assume that any such platform may easily implement this privacy feature, if necessary); however, it should be pointed out that this feature could be possible implemented in other platforms too (see, e.g., the relevant discussion for Bitcoin and Ethereum in Section 9.9). Finally, in cases that a blockchain platform provider has announced the adoption of a privacy mechanism in the near future, this is also included in Table 6.

We should stress that the number of adopted privacy mechanisms does not necessarily yields by itself a conclusion on whether privacy is present or not. As it is discussed earlier, depending on the context that a blockchain is being used, some privacy features may be unnecessary in some cases, as well as, on the contrary, insufficient in some other cases. Therefore, Table 6 simply surveys the adopted privacy mechanisms per blockchain platform and should not be used to evaluate these platforms in terms of privacy.

Concluding, it should be pointed out that, in general, future developments of blockchains may in turn pose new privacy threats, thus rendering their overall evaluation a challenging and evolving task. For example, according to a statement on the Hedera's website (which currently employs permissioned nodes), the plan is to include permissionless nodes in the future, towards growing the network. Such an extension clearly affects the privacy requirements. Therefore, in such changes of a platform, its applicability in a SOHO environment should be considered on a new basis. In other words, towards examining the blockchain platforms, a long-term approach needs to be followed - i.e. it is essential to also take into account the planned changes of the platform.

**Table 6**

Comparison of blockchain platforms in terms of security and privacy, PQC capabilities, as well as other characteristics.

Blockchain platforms	Type	Industry focus	Smart contracts language	Platform's security	PQC	Privacy mechanisms
Bitcoin	PL	Financial	Bitcoin script, Miniscript	Low (0.1)	✓	9.1–9.3
Bitcoin-NG	PL	Financial	–	Low (0.29)	–	9.1–9.3
Litecoin	PL	Financial	C++	Low (0.29)	–	9.1, 9.5
Ethereum	PL	Cross-industry	Ethereum bytecode, LLL, Solidity	Low (0.27)	–	9.1–9.3, 9.5–9.7
Elastico	PL	Financial	–	Med (0.36)	–	9.1
Parity	P	Cross-industry	Rust	Med (0.34)	–	9.1, 9.5, 9.9
Corda	P	Financial	Kotlin, Java	Med (0.49)	✓	9.1, 9.7–9.9
Stellar	PL/P	Financial	JavaScript, Golang, Java, Ruby, Python, C#	Med (0.42)	–	9.1, 9.7, 9.9
Fabric	P	Cross-industry	Golang, Java, Node.js	Med (0.45)	/ <sup>a</sup>	9.1, 9.7–9.9
Symbiont	P	Financial	Symbiont Programming Language (SymPL)	Med (0.38)	–	9.1, 9.5, 9.9
Sawtooth	P	Cross-industry	C++, Go, Java, Solidity, Python, Rust, JavaScript	Low (0.28)	–	9.1, 9.5, 9.9
SMaRtChain	P	Cross-industry	–	Med (0.45)	–	9.1, 9.9
NEO	P	Smart economy	C#, VB.Net, Kotlin, F#, Python, JavaScript	Med (0.43)	/	9.1, 9.9
EOSIO	P	Cross-industry	C, C++, Wasm	Med (0.31)	–	9.1, 9.8, 9.9
Bitshares	PL	Financial	C++	Med (0.36)	–	9.1, 9.5
Tron	PL	Financial	Solidity	Med (0.42)	–	9.1, 9.7
Lisk	PL	Financial	JavaScript, TypeScript	Med (0.32)	–	9.1
Tezos	PL	Cross-industry	Michelson	Med (0.54)	–	9.1, 9.7
Exonum	P	Cross-industry	Java, Rust	Med (0.64)	–	9.1, 9.9
Quorum	P	Cross-industry	Solidity, Vyper	Med (0.49)	–	9.1, 9.5–9.7, 9.9
Ripple	P	Financial	Any language	Med (0.48)	–	9.1, 9.2, 9.9
Ethermint	PL/P	Financial	Ethereum smart contracts	Med (0.4)	–	9.1, 9.9
Burrow	P	Cross-industry	Ethereum smart contracts Web Assembly (WASM)	Med (0.45)	–	9.1, 9.9
Ontology	P	Cross-industry	C, Python	Med (0.42)	–	9.1, 9.9
Iroha	P	Cross-industry	Native	Med (0.39)	–	9.1, 9.8, 9.9
IOTA	PL	IoT	Qupla, Abra VM	Med (0.54)	✓	9.1, 9.3, 9.9
Hedera	P	Cross-industry	Solidity, Vyper	High (0.74)	–	9.1, 9.9
Ouroboros	PL	Financial	–	Med (0.4)	–	9.1, 9.5, 9.7
Snow-White	P	Financial	–	Med (0.39)	–	9.1, 9.9
Algorand	PL	Financial	TEAL, Clarity	Med (0.39)	/	9.1, 9.9
Monero	PL	Financial	– <sup>b</sup>	Med (0.39)	–	9.1, 9.2, 9.4
Besu	P	Cross-industry	Solidity, Vyper	Med (0.42)	–	9.1, 9.5, 9.9

✓ Denotes that techniques against cryptanalytic attacks have being implemented.

/ Denotes that the platform is transitioning to a quantum-resistant infrastructure.

– Denotes that the factor has not yet being implemented.

<sup>a</sup>PQ-Fabric [142] is not yet peer-reviewed and therefore not considered for our evaluation.<sup>b</sup>Tari [201], written in Rust, can be used as a secondary layer to provide the smart contracts' functionality in Monero.

## 10. Blockchains' comparative evaluation

The consensus protocols discussed in Section 6 have been implemented into various blockchain platforms, with different benefits, shortcomings and security levels. The adversarial tolerance of each protocol, along with its performance; i.e. scalability, transactions' efficiency and block confirmation time are considered to be the most critical factors in an IoT implementation. Therefore, it is compulsory each transaction that contains IoT data to be stored to a block in a matter of seconds or even less [33]. As already mentioned, for the transactions' throughput, we adopt a staircase model for TPS score below the threshold of 2000 TPS. A similar approach was followed for other performance metrics, like the confirmation time. As to the terms of performance, even with the same scalable and secure consensus protocol, slightly changing some parameters of its design, can result into different performance metrics. Therefore, in this section, we compare the aforementioned solutions for the needs of a blockchain-enabled smart home ecosystem.

Despite PoW's security advantages, several defects and restraints of PoW-based consensus algorithms exist and are becoming more and more evident. Such protocols have many restraints that IoT devices cannot solve. The mining process requires high computational power and high bandwidth to establish consensus, while most of the IoT devices possess low processing capabilities and elementary hardware specifications with very low storage capabilities. Thus, the resource demanding mining tasks cannot be executed on low power home devices and the requirements that are needed to store an entire blockchain mechanism cannot be satisfied.

PoS and its variants can significantly reduce the energy consumption that PoW-based consensus protocols introduce and allow IoT nodes to participate in the consensus process. Despite of this, new security

issues might arise that cannot be easily solved in a blockchain-based smart home ecosystem. The nodes with the highest amounts of stakes, have higher probability to create the next block; an issue that does not offer much in terms of decentralisation and makes the protocol somehow centralised with only a few number nodes to be able to create a new block. Although Ouroboros [89] and Snow White [88] are both PoS consensus protocols, they present interesting differences between them in the leader election process. In Ouroboros, a subset of stakeholders executes a multi-party coin tossing protocol and agrees upon a random seed that is given into a pseudo-random function that outputs a leader according to its accumulated stake. This seed is also used to define the next subset of stakeholders in the following round and the reward for the creation of the block to be distributed between all the subset's members. On the other hand, Snow White is based on a weakly synchronised clock and employs a private leader election process. Nevertheless, PoS protocols cannot provide the necessary performance requirements to be deployed in an smart home ecosystem, as shown in Table 7 and they are prone to various attacks. In a "nothing-at-stake-attack", an adversary has nothing to loose, if it starts to behave maliciously and create alternative chains. This attack could not be easily addressed in the referenced IoT ecosystem.

In the case of hybrid consensus protocols (PoW and BFT or in PoS and BA), the voting weight of each node is increased according to its block proposing capabilities. This augmentation is established either by linking the computational capabilities of each node to its opportunity of participating in the BFT consensus process (e.g. Elastico) or by weighting each nodes' vote to its stake. Algorand guarantees that the probability of forks is negligible, which is a significant advantage over the classical PoS protocols, since various attacks can be mitigated, such as the aforementioned nothing-at-stake attack. The protocol's design is such that it allows to scale to millions of users and maintain a high

**Table 7**

Comparison of blockchain consensus protocols, in terms of fault tolerance and performance requirements, also incorporating the security and privacy requirements illustrated in Table 6, for the needs of the IoT applications under consideration.

Consensus protocol	Blockchain platforms	Fault tolerance	Scalability	TPS	BCT	Overall score	SOHO suitability
PoW	Bitcoin	< 50% comput. power	Low	7	10 min	37%	
	Litecoin			56	2.5 min	34%	
	Monero			1.7k	2 min	49%	
	Ethereum			25	10–15 s	34%	
PoW & BFT	Bitcoin-NG	< 50% comput. power BFT: $4f + 1$	Low	200	10 min	36%	
	Elastico			40	110 s	35%	
PoS	Ouroboros	< 50% stake	Low	257	2 min	38%	
	Snow-White			100–150	10 min	42%	
PoS & BA	Algorand	BFT: $3f + 1^a$	High	1k	5 s	68%	
DpoS	EOSIO	BFT: $3f + 1^b$	Low	1k–6k	< 1 s	65%	
	Bitshares			100k	1 s	60%	
	Lisk			2.5	6 min	35%	
	Tron			> 2k	3 s	58%	
	Tezos			≈ 40	≈ 30 min	38%	
Kafka	Fabric	CFT: $2t + 1$	Medium	3.5k/20k	< 1 s	67%	
Raft	Corda	CFT: $2t + 1$	Medium	170	1 s	57%	
	Quorum			≈ 650	4.5 s	53%	
	Fabric			≈ 7k	< 1 s	68%	
IBFT	Quorum	BFT: $3f + 1$	Low	≈ 600	4.5 s	53%	
	Besu			350–400	2 s	52%	
Aura	Parity	BFT: $2f + 1$	High	35–45	5 s	65%	
BFT-SMaRt	Fabric	BFT: $3f + 1$	High	> 10k	0.5 s	87%	
	Symbiont			80k	< 1 s	86%	
	SMaRtChain			12k–13k	< 0.25 s	87%	
VBFT	Ontology	BFT: $3f + 1$	High	3k–4k	5 s	81%	
DBFT	NEO	BFT: $3f + 1$	High	< 1k	15–20 s	69%	
Tendermint	Ethermint	BFT: $3f + 1$	Medium	200–800	< 1 s	67%	
	Burrow			> 200	2 s	61%	
Exonum	Exonum	BFT: $3f + 1$	Medium	≈ 5k	0.5 s	80%	
PoET-SGX	Sawtooth	BFT: $\Theta(\log \log n / \log n)$	High	1k–2.3k	< 1 s	84%	
Ripple	Ripple	BFT: $5f + 1^c$	High	1.5k	4 s	79%	
YAC	Iroha	BFT: $3f + 1$	Medium	× 1k	< 3 s	74%	
SCP	Stellar	BFT: $3f + 1$	High	1k	3 s	76%	
Tangle	IOTA	< 50% comput. power	High	1.5k	10 ms	84%	
Hashgraph	Hedera	BFT: $3f + 1$	High	10k	5 s	86%	

<sup>a</sup>Of maliciously-possessed stakes.

<sup>b</sup>Of delegates.

<sup>c</sup>Of nodes in each UNL.

transaction rate, which is a significant appealing feature for any IoT implementation.

Other hybrid consensus protocols, like PoA and PoET-SGX, utilise the computational power for identity management and therefore the fault tolerance threshold is based on the percentage of the malicious identities. DAG-based, such as Tangle achieve probabilistic finality, high scalability and performance, as well as, 50% adversarial tolerance of computing power. IOTA is designed to meet the necessary requirements for the integration of the blockchain technology to the IoT.

PBFT-based consensus protocols cannot be directly implemented to public and permissionless settings. However, higher throughput and lower latency can be achieved, in contrast to protocols implemented to permissionless settings, such as PoW and PoS. The fixed-size group of VRs/delegates in PBFT and DPoS, in which less 1/3 can be malicious to establish agreement for the translations' total order, is a factor that significantly impacts their nomination for a smart home implementation, despite their high performance metrics.

IBFT provides immediate block finality, as all the deterministic protocols do, high data integrity, byzantine fault tolerance and high level of governance. Unfortunately, the protocol lacks in terms of scalability, since it does not scale well under a totally decentralised network, with

the IoT devices that participate in the consensus process to be fixed and difficult to be modified.

RPCA is designed for financial purposes and requires less than 1/5 of the nodes to be faulty in the UNL to ensure network consensus. Therefore, when a specific level of trust is established, the protocol seems to trade adversarial tolerance with better performance metrics. On the down side, the multi-round dissemination scheme between each UNL, as well as, the requirements for quick convergence of votes, demand from to protocol to be highly synchronous; a requirement does not provide much to RPCA's decentralisation.

Kafka and Raft are CFT consensus protocols, which are mostly known for their high performance metrics in Fabric. These protocols cannot withstand malicious behaviour. For this reason, Kafka and Raft are implemented in permissioned blockchain platforms (Corda and Fabric) that do not follow the order-execute design, in which the transactions are ordered first and then executed in the same order on all the nodes sequentially. On the contrary, these two platforms introduce the execute-order-validate architecture, in which the transactions are executed first; for their validity to be checked (stateless validation), ordered via the consensus mechanism and then validated again (stateful validation) with all the transactions to be included to the ledger (either valid or invalid). Such a modification to the ordinary transaction flow,

might seems to “give room” to potential adversaries to take advantage of the CFT consensus process and insert malicious transactions on the ledger. On the contrary; the validation of the transactions is occurring only in the execution and validation phase [47], while in the consensus process the nodes only establish their total order without any validation checks. In contrast to BFT protocols, in which the validation replicas already vote to elect the leader, a CFT implementation in Fabric can not only accommodate flexible trust and fault assumptions, but also to separate the trust scheme for applications from the trust scheme of the consensus process; and therefore, it can allow to store trustworthy IoT data to the distributed ledger [11]. Various other cases also exist that have implemented Kafka or Raft in Fabric to address the needs of the IoT [12,202].

PoW, PoS and their variations, as well as, DPoS, Aura, PoET, Tangle and Hashgraph are probabilistic protocols and therefore, in these protocols forks can easily arise; in contrast to the rest that are deterministic. For example, IOTA and Hedera are both currently projects using consensus protocols based on *directed acyclic graphs* (DAGs) to provide high transactions’ throughput and low BCT. The transaction’s finality in IOTA is probabilistic and it is based on a *Markov Chain Monte Carlo* (MCMC) method, while the finality in Hedera is deterministic, meaning that if forks occur (in probabilistic protocols), they cannot be easily addressed in a network with resource constrained IoT nodes. On the other hand, DAGs become more and more wide, with high performance metrics and they are not limited to linear processing, as observed in the most blockchain platforms. Therefore, DAGs can be considered to be the solution for the scalability issue in any IoT ecosystem.

Permissionless blockchain platforms are affected by the propagation speed in the network and hence provide lower performance metrics and scalability. On the other hand, permissioned blockchain platforms have much better overall performance, but implement voting mechanisms and introduce network overhead limiting the system’s scalability, with the worst case complexity to be  $\mathcal{O}(N^2)$  compared to that of permissionless, which is  $\mathcal{O}(N)$ . Therefore, the suitability of permissioned blockchain implementations to the IoT communication is limited by each protocol’s communication complexity. Via the virtues of public, anonymised access and decentralisation, permissionless blockchain platforms are more applicable to industry-wide IoT applications, while the permissioned are more applicable to enterprise-based solutions, due to their significantly higher degree of access control and permission allocating capabilities.

In contrast to other works, where the corresponding suitability of the blockchain technology to the IoT seems to be vague, we define the suitability of the consensus protocol  $i$  and the blockchain platform  $j$  to the referenced IoT ecosystem (smart home) as follows

$$\text{soho}(i, j) = w_1 \text{pft}_i + w_2 \text{scl}_i + w_3 \text{sec}_j + w_4 \text{prv}_{i,j} + w_5 \text{ppc}_j + w_6 \text{thr}_{i,j} + w_7 \text{bct}_{i,j} \quad (5)$$

where, each one of the corresponding parameters are defined subsequently. The fault tolerance  $\text{pft}_i$  of the consensus protocol  $i$  is defined as

$$\text{pft}_i = \begin{cases} 1, & \text{if } i \text{ tolerates adversarial behaviour,} \\ 0.5, & \text{if } i \text{ tolerates crashes only,} \\ 0, & \text{otherwise (i.e. if } i \text{ is centralised).} \end{cases} \quad (6)$$

Clearly, centralised consensus protocols, such as the Solo protocol in Fabric, the single node protocol in Corda, etc., are not suggested for any IoT application. Each of these protocols, creates a single point of failure and therefore, such protocols are not included in our work. The scalability  $\text{scl}_i$  of consensus protocol  $i$  equals

$$\text{scl}_i = \begin{cases} 1, & \text{if } i\text{'s scalability is High,} \\ 0.5, & \text{if } i\text{'s scalability is Medium,} \\ 0, & \text{if } i\text{'s scalability is Low} \end{cases} \quad (7)$$

whereas the security  $\text{sec}_j$  of blockchain platform  $j$  is given by (1). The privacy that the combination of the consensus protocol  $i$  and blockchain

platform  $j$  provides, is denoted by  $\text{prv}_{i,j}$ . As shown in Section 9 and Table 6, all platforms and their consensus protocols considered in this work support a sufficient number of strong privacy-preserving mechanisms. Hence, it can be seen that the type of the blockchain platform (i.e. whether  $j$  is permissioned or permissionless) should be the dominant factor for evaluating privacy; this is reflected to the following expression

$$\text{prv}_{i,j} = \begin{cases} 1, & \text{if } j \text{ is permissioned,} \\ 0.5, & \text{otherwise} \end{cases} \quad (8)$$

due to the fact that permissionless blockchain platforms have in principle higher privacy risks compared to permissioned, which are more capable to satisfy the necessary flavours of privacy in an smart home ecosystem. The capability  $\text{ppc}_j$  of the blockchain platform  $j$  to implement post quantum signatures and post-quantum cryptography (PQC) in general, leads to

$$\text{ppc}_j = \begin{cases} 1, & \text{if } j \text{ implements PQC,} \\ 0.5, & \text{otherwise} \end{cases} \quad (9)$$

since apart from the implementation of PQC schemes, other factors, like cryptographic keys’ sizes, also play an important role towards withstanding various cryptographic attacks, e.g. brute force attacks in Table 2. Based on the analysis presented in Section 4.1, the transactions’ throughput  $\text{thr}_{i,j}$  that consensus protocol  $i$  can achieve when implemented in the blockchain platform  $j$  is defined as

$$\text{thr}_{i,j} = \min(0.05 \cdot \lfloor \text{TPS}_{i,j} / 100 \rfloor, 1). \quad (10)$$

The block confirmation time  $\text{bct}_{i,j}$  that the consensus protocol  $i$  can achieve when implemented in the blockchain platform  $j$  is likewise given by the expression

$$\text{bct}_{i,j} = \max(1 - 0.1 \cdot \lfloor \text{BCT}_{i,j} \rfloor, 0). \quad (11)$$

If the  $\text{thr}_{i,j}$  is greater than 2000 TPS and the the  $\text{BCT}_{i,j}$  is controled under 10 s, then the performance requirements of most IoT applications can be met [203]. The weights given to (5) are equal to  $w_1 = w_2 = 0.2$ ,  $w_3 = w_6 = 0.15$ , and  $w_4 = w_5 = w_7 = 0.1$ , with

$$w_1 + w_2 + \dots + w_7 = 1. \quad (12)$$

The assigned weights are categorised and separated into security (55%) and performance (45%); and they are selected in a way so as to give more weight to security since any IoT ecosystem is clearly an adversarial environment. Nevertheless, for the aforementioned consensus protocols and blockchain platforms, we have provided additional information, such as, consensus finality, industry focus – monetary concepts, smart contracts’ implementations and privacy preservation mechanisms. In our work, a blockchain-enabled smart home implementation is mostly focused only on the critical features of security, privacy and performance that the blockchain technology should provide to any IoT application, if used. Therefore, the assigned weights are mostly given to the consensus protocol’s security, scalability and performance (a total weight of 65%), since, this is where the IoT nodes face the most significant challenges. The rest (35%) is given to the platform’s security and privacy.

As illustrated in Table 7, by taking into consideration all the established evaluation criteria (related to the platforms’ security and privacy, but also to the underlying protocols’ performance, fault tolerance and scalability) only six consensus protocols out of sixteen can achieve high suitability score so as to be implemented to the smart home ecosystem; these are BFT-SMaRt, PoET, VBFT, Exonum, Tangle, and Hashgraph.

**Example 3.** When BFT-SMaRt is implemented in Fabric, Symbiont or SMaRtChain, it can achieve high suitability scores of 87%, 86% and 87%, respectively. The consensus protocol is secure against malicious behaviour, scalable and when it is implemented in Fabric, high throughput and low block confirmation time can be achieved. Fabric is



a permissioned ledger and achieves a medium level of security, while techniques against cryptanalytic attacks have not yet being adopted. Therefore, for  $i = 1$ , from Eq. (5), we have

$$\begin{aligned} \text{soho}(1, 2) &= w_1 \text{pft}_1 + w_2 \text{scl}_1 + w_3 \text{sec}_2 + w_4 \text{prv}_{1,2} \\ &\quad + w_5 \text{pqc}_2 + w_6 \text{thr}_{1,2} + w_7 \text{bct}_{1,2} \\ &= 0.2 \cdot 1 + 0.2 \cdot 1 + 0.15 \cdot 0.45 + 0.1 \cdot 1 \\ &\quad + 0.1 \cdot 0.5 + 0.15 \cdot 1 + 0.1 \cdot 1 \approx 0.87 \end{aligned}$$

the overall SOHO suitability score of BFT-SMaRt, when it is implemented in Fabric, to be  $\approx 87\%$ . ■

On the other hand, as it was anticipated, the cryptocurrency based ledgers and the protocols that rely on PoW or PoS to establish consensus, receive low suitability scores. Without taking into consideration the computational power required to establish consensus, which is an important factor for any blockchain-based IoT implementation, these protocols receive scores below 50%. Therefore, the computational overhead is not the only factor that makes the traditional blockchain solutions unsuitable for the needs of the IoT. The only exception from these platforms is Algorand, which receives the overall suitability score of 68%.

**Example 4.** To be more detailed, Algorand relies on PoS integrated in a BA layout to achieve optimal resilience against faulty nodes and high scalability. Algorand is a permissionless platform and achieves a medium level of security, while techniques against cryptanalytic attacks have not yet being adopted. Algorand can also achieve throughput of 1000 TPS with each transaction to be confirmed in 5 s. Therefore, for  $i = 2$ , Algorand's suitability to a SOHO ecosystem is defined from Eq. (5), and it is

$$\begin{aligned} \text{soho}(2, 1) &= w_1 \text{pft}_2 + w_2 \text{scl}_2 + w_3 \text{sec}_1 + w_4 \text{prv}_{2,1} \\ &\quad + w_5 \text{pqc}_1 + w_6 \text{thr}_{2,1} + w_7 \text{bct}_{2,1} \\ &= 0.2 \cdot 1 + 0.2 \cdot 1 + 0.15 \cdot 0.39 + 0.1 \cdot 0.5 \\ &\quad + 0.1 \cdot 0.5 + 0.15 \cdot 0.5 + 0.1 \cdot 0.5 \approx 0.68 \end{aligned}$$

and thus the overall SOHO suitability is 68%. ■

Apart from DPoS implementation in Lisk and Tezos, the rest of the consensus protocols, in any possible implementation, achieve medium suitability scores between 50% and 80%, with Ripple to achieve the highest score among them (79%). However, our evaluation framework can be used with different weights and impose other quantitative criteria to select the best solution for different IoT applications, according to each specific use case.

## 11. Conclusions

This paper conducted a comprehensive and coherent review of the available blockchain solutions to determine their ability to meet the requirements and address the challenges of the IoT ecosystem. Having the smart home ecosystem as our reference IoT domain, a number of requirements were extracted so as to assess the strengths and shortcomings of the investigated blockchains when utilised as a building block to deliver relevant applications and services to the end users. A large number of characteristics were considered during our study, covering various architectural components of blockchain solutions, that are classified into the broad areas of

- **Security:** protocols' fault tolerance, network security, overall platforms' security including smart contracts, and cryptographic strength;
- **Privacy:** both well-known but also emerging privacy-preserving mechanisms were investigated; as well as
- **Performance:** throughput, block confirmation time/latency, and scalability.

Amongst the key findings of our work, was that the defences currently provided by blockchain platforms are not sufficient to thwart all the prominent attacks against blockchains, with blockchain 1.0 and 2.0 platforms being susceptible to the majority of them. This seems to be in contrast with privacy preservation mechanisms, which are embraced, to varying degrees, by all the platforms investigated (at a minimum supporting the features of pseudonymisation and off-chain storage). If the underlying consensus protocols' performance and fault tolerance is also considered, then only a small number of platforms meet the requirements of our reference IoT ecosystem (smart home).

This work provides a holistic approach that could however be further explored in different ways so as to yield valuable research or practical results. Such explorations could consider the accurate modelling of the blockchain platforms' consensus protocols, and other operations that play a key role into determining performance or storage criteria, so as to derive measurements based on the same underlying assumptions; this constitutes part of our ongoing research. At the practical side, this goal could also be achieved by performing in-lab measurements (for the platforms allowing to do so) using a given use case scenario, thus allowing to get results under the same operational conditions.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- [1] N. Satoshi, Bitcoin: A peer-to-peer electronic cash system, 2008, <https://bitcoin.org/bitcoin.pdf> [Online]; accessed Aug 29, 2020.
- [2] L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, Y. Yang, A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling, *ACM Comput. Surv.* 53 (1) (2020).
- [3] G.S. Ramachandran, B. Krishnamachari, Blockchain for the IoT: Opportunities and challenges, 2018, [arXiv:1805.02818](https://arxiv.org/abs/1805.02818).
- [4] K. Christidis, M. Devetsikiotis, Blockchains and smart contracts for the internet of things, *IEEE Access* 4 (2016) 2292–2303.
- [5] A. Dorri, S.S. Kanhere, R. Jurdak, P. Gauravaram, Blockchain for IoT security and privacy: The case study of a smart home, in: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 2017, pp. 618–623.
- [6] N. Kolokotronis, K. Limnitiotis, S. Shiaeles, R. Griffiths, Secured by blockchain: Safeguarding internet of things devices, *IEEE Consum. Electron. Mag.* 8 (3) (2019) 28–34.
- [7] B. Shala, U. Trick, A. Lehmann, B. Ghita, S. Shiaeles, Blockchain and trust for secure, end-user-based and decentralized IoT service provision, *IEEE Access* 8 (2020) 119961–119979.
- [8] K. Bendiab, N. Kolokotronis, S. Shiaeles, S. Boucherka, Wip: A novel blockchain-based trust model for cloud identity management, in: 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 2018, pp. 724–729.
- [9] V. Attias, L. Vigneri, V. Dimitrov, Preventing denial of service attacks in IoT networks through verifiable delay functions, 2020, [arXiv preprint arXiv:2006.01977](https://arxiv.org/abs/2006.01977).
- [10] N. Kolokotronis, S. Brotsis, G. Germanos, C. Vassilakis, S. Shiaeles, On blockchain architectures for trust-based collaborative intrusion detection, in: 2019 IEEE World Congress on Services (SERVICES), 2642–939X, 2019, pp. 21–28.
- [11] C. Agostino Ardagna, R. Asal, E. Damiani, N. El Ioini, C. Pahl, Trustworthy IoT: An evidence collection approach based on smart contracts, in: 2019 IEEE International Conference on Services Computing (SCC), 2019, pp. 46–50.
- [12] S. Brotsis, N. Kolokotronis, K. Limnitiotis, S. Shiaeles, D. Kavallieros, E. Bellini, C. Pavu  , Blockchain solutions for forensic evidence preservation in IoT environments, in: 2019 IEEE Conference on Network Softwarization (NetSoft), 2019, pp. 110–114.
- [13] O. Gkotsopoulou, E. Charalambous, K. Limnitiotis, P. Quinn, D. Kavallieros, G. Sargsyan, S. Shiaeles, N. Kolokotronis, Data protection by design for cybersecurity systems in a smart home environment, in: 2019 IEEE Conference on Network Softwarization (NetSoft), IEEE, 2019, pp. 101–109.

- [14] C.S. Kouzinopoulos, K.M. Giannoutakis, K. Votis, D. Tzavaras, A. Collen, N.A. Nijdam, D. Konstantas, G. Spathoulas, P. Pandey, S. Katsikas, Implementing a forms of consent smart contract on an IoT-based blockchain to promote user trust, in: 2018 Innovations in Intelligent Systems and Applications (INISTA), IEEE, 2018, pp. 1–6.
- [15] A. Panarello, N. Tapas, G. Merlino, F. Longo, A. Puliafito, Blockchain and iot integration: A systematic survey, *Sensors* 18 (8) (2018) 2575.
- [16] A. Reyna, C. Martín, J. Chen, E. Soler, M. Díaz, On blockchain and its integration with IoT. Challenges and opportunities, *Future Gener. Comput. Syst.* 88 (2018) 173–190.
- [17] R. Shrestha, S. Kim, Chapter ten - integration of IoT with blockchain and homomorphic encryption: Challenging issues and opportunities, *Adv. Comput.* 115 (2019) 293–331.
- [18] S. Brotsis, N. Kolokotronis, K. Limniotis, S. Shiales, On the security of permissioned blockchain solutions for IoT applications, in: 2020 6th IEEE Conference on Network Softwarization (NetSoft), 2020, pp. 465–472.
- [19] M. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, M.H. Rehmani, Applications of blockchains in the internet of things: A comprehensive survey, *IEEE Commun. Surv. Tutor.* 21 (2) (2019) 1676–1717.
- [20] M. Salimitari, M. Chatterjee, Y.P. Fallah, A survey on consensus methods in blockchain for resource-constrained IoT networks, *Internet Things* 11 (2020) 100212.
- [21] L. Baird, M. Harmon, P. Madsen, Hedera: A Governing Council & Public Hashgraph Network, Whitepaper V.1.4, 2018, <https://www.hedera.com/hh-whitepaper-v1.4-181017.pdf>.
- [22] C. Stergiou, K.E. Psannis, B.-G. Kim, B. Gupta, Secure integration of iot and Cloud Computing, *Future Gener. Comput. Syst.*, 78, 964–975.
- [23] S. Monogios, K. Limniotis, N. Kolokotronis, S. Shiales, A case study of intra-library privacy issues on android GPS navigation apps, in: E-Democracy – Safeguarding Democracy and Human Rights in the Digital Age, Springer International Publishing, 2020, pp. 34–48.
- [24] Y. Huang, Y. Bian, R. Li, J.L. Zhao, P. Shi, Smart contract security: A software lifecycle perspective, *IEEE Access* 7 (2019) 150184–150202.
- [25] M.S. Ferdous, K. Biswas, M.J.M. Chowdhury, Niaz, Chapter two - integrated platforms for blockchain enablement, in: Role of Blockchain Technology in IoT Applications, in: Advances in Computers, vol. 115, Elsevier, 2019, pp. 41–72.
- [26] T. Fernández-Caramés, P. Fraga-Lamas, A review on the use of blockchain for the internet of things, *IEEE Access* 6 (2018) 32979–33001.
- [27] M.A. Khan, K. Salah, IoT Security: Review, blockchain solutions, and open challenges, *Future Gener. Comput. Syst.* 82 (2018) 395–411.
- [28] X. Li, P. Jiang, T. Chen, X. Luo, Q. Wen, A survey on the security of blockchain systems, *Future Gener. Comput. Syst.* 107 (2020) 841–853.
- [29] I. Makhdoom, M. Abolhasan, H. Abbas, W. Ni, Blockchain's adoption in IoT: The challenges, and a way forward, *J. Netw. Comput. Appl.* 125 (2019) 251–279.
- [30] M.L. Perez, A.C. Lagman, J.B.C. Legaspi, R.D.M. De Angel, K.A.S. Awat, Suitability of IoT to blockchain network based on consensus algorithm, in: 2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), 2019, pp. 1–5.
- [31] X. Wang, X. Zha, W. Ni, R.P. Liu, Y.J. Guo, X. Niu, K. Zheng, Survey on blockchain for internet of things, *Comput. Commun.* 136 (2019) 10–29.
- [32] M. Wu, K. Wang, X. Cai, S. Guo, M. Guo, C. Rong, A comprehensive survey of blockchain: From theory to IoT applications and beyond, *IEEE Internet Things J.* 6 (5) (2019) 8114–8154.
- [33] M. Moniruzzaman, S. Khezr, A. Yassine, R. Benlamri, Blockchain for smart homes: Review of current trends and research challenges, *Comput. Electr. Eng.* 83 (2020) 106585.
- [34] C. Lee, L. Zappaterra, K. Choi, H.-A. Choi, Securing smart home: Technologies, security challenges, and security requirements, in: 2014 IEEE Conference on Communications and Network Security, IEEE, 2014, pp. 67–72.
- [35] C.M. Mathas, C. Vassilakis, N. Kolokotronis, A trust management system for the IoT domain, in: 2020 IEEE World Congress on Services (SERVICES), 2020, pp. 183–188.
- [36] P.K. Singh, R. Singh, S.K. Nandi, S. Nandi, Managing smart home appliances with proof of authority and blockchain, in: International Conference on Innovations for Community Services, Springer, 2019, pp. 221–232.
- [37] Cyber-Trust, The cyber-trust project, 2020, URL: <https://cyber-trust.eu/> [Online]; accessed Aug 29, 2020.
- [38] T.I. Team, A decentralized network for internet of things powered by a privacy-centric blockchain, 2018, URL: [https://s3.amazonaws.com/web-iotex-static/home/IoTeX\\_Whitepaper\\_1.5\\_EN.pdf](https://s3.amazonaws.com/web-iotex-static/home/IoTeX_Whitepaper_1.5_EN.pdf) [Online]; accessed Aug 29, 2020.
- [39] Atonomi, The atonomi project, 2020, URL: <https://atonomi.io/> [Online]; accessed Aug 29, 2020.
- [40] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi, K. Salah, A user authentication scheme of IoT devices using blockchain-enabled fog nodes, in: 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), 2018, pp. 1–8.
- [41] L. Axon, Privacy-Awareness in Blockchain-Based PKI, Cdt Technical Paper Series, 2015.
- [42] S. Huh, S. Cho, S. Kim, Managing IoT devices using blockchain platform, in: 2017 19th International Conference on Advanced Communication Technology (ICACT), IEEE, 2017, pp. 464–467.
- [43] F. Lombardi, L. Aniello, S. De Angelis, A. Margheri, V. Sassone, A blockchain-based infrastructure for reliable and cost-effective IoT-aided smart grids, in: Living in the Internet of Things: Cybersecurity of the IoT - 2018, 2018, pp. 1–6.
- [44] Z. Nehai, G. Guérard, Integration of the blockchain in a smart grid model, in: The 14th International Conference of Young Scientists on Energy Issues (CYSENI) 2017, 2017, pp. 127–134.
- [45] H. Wu, W. Zheng, A. Chiesa, R.A. Popa, I. Stoica, DIZK: A Distributed Zero Knowledge Proof System, Cryptology ePrint Archive, Report 2018/691, 2018, <https://eprint.iacr.org/2018/691>.
- [46] B. Ampel, M. Patton, H. Chen, Performance modeling of hyperledger sawtooth blockchain, in: 2019 IEEE Int'l Conf. on Intelligence and Security Informatics (ISI), 2019, pp. 59–61.
- [47] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S.W. Cocco, J. Yellick, Hyperledger fabric: A distributed operating system for permissioned blockchains, in: Proc. of 13th EuroSys Conf., in: EuroSys '18, 2018, pp. 30:1–30:15.
- [48] A. Baliga, I. Subhod, P. Kamat, S. Chatterjee, Performance evaluation of the quorum blockchain platform, 2018, CoRR abs/1809.03421. [arXiv:1809.03421](https://arxiv.org/abs/1809.03421).
- [49] A. Bessani, E. Alchieri, J. ao Sousa, A. Oliveira, F. Pedone, From byzantine replication to blockchain: Consensus is only the beginning, 2020, [arXiv:2004.14527](https://arxiv.org/abs/2004.14527).
- [50] Bitfury, Exonum documentation, 2020, URL: <https://exonum.com/developers> [Online]; accessed Aug 29, 2020.
- [51] Bitshares, Bitshares documentation, 2020, URL: <https://bitshares.org/technology/industrial-performance-scalability/> [Online]; accessed Aug 29, 2020.
- [52] EOSIO, EOSIO Documentation, 2018, URL: <https://eos.io/news/askblockone-what-are-the-benefits-of-the-eosio-infrastructure/> [Online]; accessed Aug 29, 2020.
- [53] C. Ferris, Does hyperledger fabric perform at scale?, 2020, URL: <https://www.ibm.com/blogs/blockchain/2019/04/does-hyperledger-fabric-perform-at-scale/>.
- [54] I. Foundation, A beginner's guide to ethermint, 2020, URL: <https://blog.cosmos.network/a-beginners-guide-to-ethermint-38ee15f8a6f4> [Online]; accessed Aug 29, 2020.
- [55] S. Mahankali, Blockchain: The Untold Story: From birth of Internet to future of Blockchain, BPB Pubs, 2019.
- [56] F. Muratov, A. Lebedev, N. Iushkevich, B. Nasrulin, M. Takemiya, YAC: BFT consensus algorithm for blockchain, 2018, CoRR abs/1809.00554. [arXiv:1809.00554](https://arxiv.org/abs/1809.00554) URL: <http://arxiv.org/abs/1809.00554>.
- [57] NEO, NEO Documentation, 2020, URL: <https://docs.neo.org/docs/en-us/basic/whitepaper.html> [Online]; accessed Aug 29, 2020.
- [58] Ontology, Ontology documentation, 2020, URL: <https://dev-docs.ont.io/#/docs-en/DeveloperGuide/02-VBFT-introduction> [Online]; accessed Aug 29, 2020.
- [59] Ripple, Ripple documentation, 2020, <https://ripple.com/xrp> [Online]; accessed Aug 29, 2020.
- [60] Soramitsu, Hyperledger iroha: Working towards standard protocols for identity and digital asset management, 2020, URL: [https://www.jpibec.or.jp/sp/topics/event/u71kba000000an5u-att/15\\_soramitsu.pdf](https://www.jpibec.or.jp/sp/topics/event/u71kba000000an5u-att/15_soramitsu.pdf) [Online]; accessed Aug 29, 2020.
- [61] J. Sousa, A. Bessani, M. Vukolic, A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform, in: 2018 48th Annual IEEE/IFIP Int'l Conf. on Dependable Systems and Networks (DSN), 2018, pp. 51–58.
- [62] Tezos, Tezos documentation, 2020, <https://blockfrye.com/tezos-xtz/> [Online]; accessed Aug 29, 2020.
- [63] Tron, Tron documentation, 2020, URL: <https://tron.network/> [Online]; accessed Aug 29, 2020.
- [64] R. Tushov, Hyperledger explorer, 2020, <https://www.hyperledger.org/category/hyperledger-explorer>, [Online]; accessed Aug 29, 2020.
- [65] G. Wang, Z.J. Shi, M. Nixon, S. Han, Sok: Sharding on blockchain, in: Proceedings of the 1st ACM Conference on Advances in Financial Technologies, 2019, pp. 41–61.
- [66] Y. Xiao, N. Zhang, W. Lou, Y.T. Hou, A survey of distributed consensus protocols for blockchain networks, *IEEE Commun. Surv. Tutor.* (2020) 1.
- [67] H. Yi, F. Wei, Research on a suitable blockchain for IoT platform, in: S. Patnaik, V. Jain (Eds.), Recent Developments in Intelligent Computing, Communication and Devices, Springer, 2019, pp. 1063–1072.
- [68] S. Arif, M.A. Khan, S.U. Rehman, M.A. Kabir, M. Imran, Investigating smart home security: Is blockchain the answer?, *IEEE Access* 8 (2020) 117802–117816.
- [69] Y. Aung, T. Tantiidham, Review of ethereum: Smart home case study, in: 2017 2nd International Conference on Information Technology (INCIT), 2017, pp. 1–4.

- [70] J. Ali, T.A. Syed, S. Musa, A. Zaharani, Towards secure iot communication with smart contracts in a blockchain infrastructure, 2020, CoRR abs/2001.01837.
- [71] E. Parliament, Blockchain and the general data protection regulation, 2020, [https://www.europarl.europa.eu/RegData/etudes/STUD/2019/634445/EPRS\\_STU\(2019\)634445\\_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/STUD/2019/634445/EPRS_STU(2019)634445_EN.pdf) [Online]; accessed Aug 29, 2020.
- [72] A. Dwivedi, G. Srivastava, S. Dhar, R. Singh, A decentralized privacy-preserving healthcare blockchain for IoT, *Sensors* 19 (2019) 326.
- [73] MyHealthMyData, Myhealthmydata, 2020, <http://www.myhealthmydata.eu/> [Online]; accessed Aug 29, 2020.
- [74] B. Janita, R.J. Kannan, N. Kumaratharan, Enhancing security in smart homes-a review, in: *Intelligent Computing in Engineering*, 2020, pp. 361–369.
- [75] F.B. Schneider, Implementing fault-tolerant services using the state machine approach: A tutorial, *ACM Comput. Surv.* 22 (4) (1990) 299–319.
- [76] L. Lamport, R. Shostak, M. Pease, The byzantine generals problem, *ACM Trans. Program. Lang. Syst.* 4 (3) (1982) 382–401.
- [77] V. Hadzilacos, S. Toueg, Fault-tolerant broadcasts and related problems, in: S. Mullender (Ed.), *Distributed Systems*, second ed., 1993, pp. 97–145.
- [78] B.M. Oki, B.H. Liskov, Viewstamped replication: A new primary copy method to support highly-available distributed systems, in: *Proc. of 7th Annual ACM Symposium on Principles of Distributed Computing*, 1988, pp. 8–17.
- [79] L. Lamport, The part-time parliament, *ACM Trans. Comput. Syst.* 16 (2) (1998) 133–169.
- [80] M. Castro, B. Liskov, Practical byzantine fault tolerance and proactive recovery, *ACM Trans. Comput. Syst.* 20 (4) (2002) 398–461.
- [81] A. Bessani, J. Sousa, E.E.P. Alchieri, State machine replication for the masses with BFT-SMART, in: 2014 44th Annual IEEE/IFIP Int'l Conf. on Dependable Systems and Networks, 2014, pp. 355–362.
- [82] Litecoin, Litecoin foundation, 2020, <https://litecoin.org> [Online]; accessed Aug 29, 2020.
- [83] S. Noether, Ring Signature Confidential Transactions for Monero, *Cryptology ePrint Archive*, Report 2015/1098, 2015, <https://eprint.iacr.org/2015/1098>.
- [84] G. Wood, et al., Ethereum: A Secure Decentralised Generalised Transaction Ledger, *Ethereum Project Yellow Paper*, 151, 2014, pp. 1–32.
- [85] I. Eyal, A.E. Gencer, E.G. Sirer, R.V. Renese, Bitcoin-NG: A scalable blockchain protocol, in: 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16), USENIX Association, 2016, pp. 45–59.
- [86] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, P. Saxena, A secure sharding protocol for open blockchains, in: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Association for Computing Machinery, 2016, pp. 17–30.
- [87] I. Eyal, A.E. Gencer, E.G. Sirer, R. Van Renesse, Bitcoin-ng: A scalable blockchain protocol, in: 13th (USENIX) Symposium on Networked Systems Design and Implementation (NSDI 16), 2016, pp. 45–59.
- [88] I. Bentov, R. Pass, E. Shi, Snow white: Provably secure proofs of stake., *IACR Cryptol. ePrint Arch.* 2016 (2016) 919.
- [89] A. Kiayias, A. Russell, B. David, R. Oliynykov, Ouroboros: A provably secure proof-of-stake blockchain protocol, in: J. Katz, H. Shacham (Eds.), *Advances in Cryptology – CRYPTO 2017*, Springer International Publishing, 2017, pp. 357–388.
- [90] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, N. Zeldovich, Algorand: Scaling byzantine agreements for cryptocurrencies, in: *Proceedings of the 26th Symposium on Operating Systems Principles*, Association for Computing Machinery, 2017, pp. 51–68.
- [91] Lisk, Lisk documentation, 2020, URL: <https://lisk.io/documentation/lisk-sdk/index.html> [Online]; accessed Aug 29, 2020.
- [92] D. Larimer, DPOS Consensus Algorithm – The Missing, White Paper, 2017, <https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-whitepaper> [Online]; accessed Aug 29, 2020.
- [93] J. Kreps, N. Narkhede, J. Rao, et al., Kafka: A distributed messaging system for log processing, in: *Proceedings of the NetDB*, Vol. 11, 2011, pp. 1–7.
- [94] C. Gorenflo, S. Lee, L. Golab, S. Keshav, Fastfabric: Scaling hyperledger fabric to 20,000 transactions per second, in: 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 2019, pp. 455–463.
- [95] D. Ongaro, J. Ousterhout, In search of an understandable consensus algorithm, in: 2014 USENIX Annual Technical Conf. (USENIX ATC 14), 2014, pp. 305–319.
- [96] Quorum, Quorum documentation, 2020, URL: <http://docs.quorum.com/en/latest/> [Online]; accessed Aug 29, 2020.
- [97] R3, R3 primer series 1, 2020, URL: <https://www.r3.com/wp-content/uploads/2019/01/R3-Quick-Facts.pdf> [Online]; accessed Aug 29, 2020.
- [98] L. Lamport, et al., Paxos made simple, *ACM Sigact News* 32 (4) (2001) 18–25.
- [99] S.D. Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, V. Sassone, PBFT Vs. proof-of-authority: applying the CAP theorem to permissioned blockchain, in: *Italian Conf. on Cyber Security*, Proc. ITASEC, 2018, pp. 1–11.
- [100] J. Liu, W. Li, G.O. Karame, N. Asokan, Scalable byzantine consensus via hardware-assisted secret sharing, *IEEE Trans. Comput.* 68 (1) (2019) 139–151.
- [101] H. Moniz, The Istanbul BFT consensus algorithm, 2020, [arXiv:2002.03613](https://arxiv.org/abs/2002.03613).
- [102] Besu, Besu documentation, 2020, <https://www.hyperledger.org/use/besu> [Online]; accessed Aug 29, 2020.
- [103] Hyperledger Besu Team, Hyperledger besu 1.5 performance enhancements, 2020, <https://www.hyperledger.org/blog/2020/08/06/hyperledger-besu-1-5-performance-enhancements> [Online]; accessed Aug 29, 2020.
- [104] Parity, Parity tech documentation, 2020, <https://wiki.parity.io/> [Online]; accessed Aug 29, 2020.
- [105] Symbiont, Symbiont documentation, 2020, <https://symbiont.io/> [Online]; accessed Aug 29, 2020.
- [106] Honglei-Cong, Vbft introduction, 2018, URL: <https://github.com/ontio/documentation/blob/master/vbft-intro/vbft-intro.md> [Online]; accessed Aug 29, 2020.
- [107] J. Zhang, Y. Rong, J. Cao, C. Rong, J. Bian, W. Wu, DBFT: A byzantine fault tolerant protocol with graceful performance degradation, in: 2019 38th Symposium on Reliable Distributed Systems (SRDS), 2019, pp. 123–12309.
- [108] J. Kwon, Tendermint: Consensus without Mining, Draft v. 0.6, fall 1, 2014.
- [109] L. Foundation, Hyperledger burrow documentation, 2020, URL: <https://github.com/hyperledger/burrow> [Online]; accessed Aug 29, 2020.
- [110] Y. Yanovich, I. Ivashchenko, A. Ostrovsky, A. Shevchenko, A. Sidorov, Exonum: Byzantine fault tolerant protocol for blockchains, *bitfury. com* (2018) 1–36.
- [111] Hyperledger, Hyperledger sawtooth documentation, 2020, <https://hyperledger.org/docs/core/releases/1.2.5/> [Online]; accessed Aug 29, 2020.
- [112] V. Costan, S. Devadas, Intel SGX explained, *IACR Cryptol. ePrint Arch.* 2016 (2016) 86.
- [113] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, W. Shi, On security analysis of proof-of-elapsed-time (poet), in: P. Spirakis, P. Tsigas (Eds.), *Stabilization, Safety, and Security of Distributed Systems*, 2017, pp. 282–297.
- [114] Iroha, Hyperledger iroha documentation, 2020, URL: <https://iroha.readthedocs.io/en/latest/overview.html> [Online]; accessed Aug 29, 2020.
- [115] D. Mazieres, The stellar consensus protocol: A federated model for internet-level consensus, 2020, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.696.93&rep=rep1&type=pdf> [Online]; accessed Aug 29, 2020.
- [116] M.A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, H. Janicke, Blockchain technologies for the internet of things: Research issues and challenges, *IEEE Internet Things J.* 6 (2) (2018) 2188–2204.
- [117] H. Hasanova, U.-j. Baek, M.-g. Shin, K. Cho, M.-S. Kim, A survey on blockchain cybersecurity vulnerabilities and possible countermeasures, *Int. J. Netw. Manag.* 29 (2) (2019) e2060.
- [118] G.A.F. Rebello, G.F. Camilo, L.C. Guimaraes, L.A.C. de Souza, O.C.M. Duarte, Security and performance analysis of quorum-based blockchain consensus protocols, 2020, <https://www.gta.ufrj.br/ftp/gta/TechReports/RCG20c.pdf> [Online]; accessed Aug 29, 2020.
- [119] D. Schwartz, N. Youngs, A. Britto, et al., The Ripple Protocol Consensus Algorithm, *Ripple Labs Inc White Paper* 5, 2014.
- [120] M.-J. Lagarde, Security assessment of authentication and authorization mechanisms in ethereum, quorum, hyperledger fabric and corda, 2019, [https://www.epfl.ch/labs/dedis/wp-content/uploads/2020/01/report-2018\\_2-marie-jeanne-security-assessment.pdf](https://www.epfl.ch/labs/dedis/wp-content/uploads/2020/01/report-2018_2-marie-jeanne-security-assessment.pdf) [Online]; accessed Aug 29, 2020.
- [121] D.A. Wijaya, J.K. Liu, R. Steinfeld, D. Liu, J. Yu, On the unforkability of monero, in: *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, 2019, pp. 621–632.
- [122] E. Heilman, Breaking the crypto used in iota, 2019, <https://ethanheilman.tumblr.com/post/165085348035/breaking-the-crypto-used-in-iota> [Online]; accessed Sept 11, 2020.
- [123] Y. Huang, H. Wang, L. Wu, G. Tyson, X. Luo, R. Zhang, X. Liu, G. Huang, X. Jiang, Understanding (mis) behavior on the EOSIO blockchain, *Proc. ACM Meas. Anal. Comput. Syst.* 4 (2) (2020) 1–28.
- [124] T. Rajab, M.H. Manshaei, M. Dakhilalian, M. Jadhwal, M.A. Rahman, On the feasibility of sybil attacks in shard-based permissionless blockchains, 2020, [arXiv preprint arXiv:2002.06531](https://arxiv.org/abs/2002.06531).
- [125] L. Ismail, H. Materwala, A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions, *Symmetry* 11 (10) (2019) 1198.
- [126] O. Boireau, Securing the blockchain against hackers, *Netw. Secur.* 2018 (1) (2018) 8–11.
- [127] D. Dasgupta, J.M. Shreiner, K.D. Gupta, A survey of blockchain from security perspective, *J. Bank. Financ. Technol.* 3 (1) (2019) 1–17.
- [128] A.A. Kochkarov, S.D. Osipovich, R.A. Kochkarov, Analysis of ddos attacks on bitcoin cryptocurrency payment system, *Rev. ESPACIOS* 41 (03) (2020).
- [129] A. Dabholkar, V. Saraswat, Ripping the fabric: Attacks and mitigations on hyperledger fabric, in: *International Conference on Applications and Techniques in Information Security*, Springer, 2019, pp. 300–311.
- [130] M. Kim, Y. Kwon, Y. Kim, Is stellar as secure as you think?, in: 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE, 2019, pp. 377–385.
- [131] Y. Liu, X. Liu, C. Tang, J. Wang, L. Zhang, Unlinkable coin mixing scheme for transaction privacy enhancement of bitcoin, *IEEE Access* 6 (2018) 23261–23270.
- [132] P. Otte, M. de Vos, J. Pouwelse, Trustchain: A sybil-resistant scalable blockchain, *Future Gener. Comput. Syst.* 107 (2020) 770–780.
- [133] W. Bubberman, S. Roos, S.P. Kumble, TLS MITM Attack on the Ripple XRP Ledger, Technical Report, Delft University of Technology, 2020.
- [134] E. Heilman, A. Kendler, A. Zohar, S. Goldberg, Eclipse attacks on bitcoin's peer-to-peer network, in: 24th (USENIX) Security Symposium (USENIX) Securi-Ty 15), 2015, pp. 129–144.



- [135] Z. Wang, J. Liu, Z. Zhang, Y. Zhang, J. Yin, H. Yu, W. Liu, A combined micro-block chain truncation attack on bitcoin-NG, in: *Australasian Conference on Information Security and Privacy*, Springer, 2019, pp. 322–339.
- [136] Y. Marcus, E. Heilman, S. Goldberg, Low-resource eclipse attacks on ethereum's peer-to-peer network., *IACR Cryptol. ePrint Arch.* 2018 (2018) 236.
- [137] S. Zhang, J.-H. Lee, Eclipse-based stake-bleeding attacks in pos blockchain systems, in: *Proceedings of the 2019 ACM International Symposium on Block-Chain and Secure Critical Infrastructure*, 2019, pp. 67–72.
- [138] A. Singh, M. Castro, P. Druschel, A. Rowstron, Defending against eclipse attacks on overlay networks, in: *Proceedings of the 11th Workshop on ACM SIGOPS European Workshop*, 2004, pp. 21–es.
- [139] Q. Wang, B. Qin, J. Hu, F. Xiao, Preserving transaction privacy in bitcoin, *Future Gener. Comput. Syst.* 107 (2020) 793–804.
- [140] M. Conti, E.S. Kumar, C. Lal, S. Ruj, A survey on security and privacy issues of bitcoin, *IEEE Commun. Surv. Tutor.* 20 (4) (2018) 3416–3452.
- [141] A. Gervais, H. Ritzdorf, G.O. Karamé, S. Capkun, Tampering with the delivery of blocks and transactions in bitcoin, in: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 692–705.
- [142] A. Holcomb, G. Pereira, B. Das, M. Mosca, Pqfabric: A permissioned blockchain secure from both classical and quantum attacks, 2020.
- [143] N. Atzei, M. Bartoletti, T. Cimoli, A survey of attacks on ethereum smart contracts (sok), in: M. Maffei, M. Ryan (Eds.), *Principles of Security and Trust*, Springer Berlin Heidelberg, 2017, pp. 164–186.
- [144] A. Feder, N. Gandal, J. Hamrick, T. Moore, The impact of ddos and other security shocks on bitcoin currency exchanges: Evidence from mt. Gox, *J. Cybersecur.* 3 (2) (2017) 137–144.
- [145] H. Hashgraph, Is hashgraph secure?, 2019, <https://hbarprice.com/is-hashgraph-secure/> [Online]; accessed Sept 14, 2020.
- [146] O. Hasan, S. Tahar, Formal verification methods, in: *Encyclopedia of Information Science and Technology*, Third Edition, IGI Global, 2015, pp. 7162–7170.
- [147] S. Amani, M. Bégel, M. Bortin, M. Staples, Towards verifying ethereum smart contract bytecode in Isabelle/HOL, in: *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs*, 2018, pp. 66–77.
- [148] M. Sotnick, Formal verification of smart contracts with the k framework, 2019, URL: <https://www.apriorit.com/dev-blog/592-formal-verification-with-k-framework> [Online]; accessed Aug 29, 2020.
- [149] I. Grishchenko, M. Maffei, C. Schneidewind, Foundations and tools for the static analysis of ethereum smart contracts, in: *Computer Aided Verification*, 2018, pp. 51–78.
- [150] Z. Yang, H. Lei, W. Qian, A hybrid formal verification system in Coq for ensuring the reliability and security of ethereum-based service smart contracts, *IEEE Access* 8 (2020) 21411–21436.
- [151] N. Swamy, C. Hrițcu, C. Keller, A. Rastogi, A. Delignat-Lavaud, S. Forest, K. Bhargavan, C. Fournet, P.-Y. Strub, M. Kohlweiss, J.-K. Zinzindohoue, S. Zanella-Béguelin, Dependent types and multi-monadic effects in  $F^*$ , *SIGPLAN Not.* 51 (1) (2016) 256–270.
- [152] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Gollamudi, G. Gonthier, N. Kobeissi, N. Kulatova, A. Rastogi, T. Sibut-Pinote, N. Swamy, S. Zanella-Béguelin, Formal verification of smart contracts: Short paper, in: *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security*, 2016, pp. 91–96.
- [153] Y. Murray, D.A. Anisi, Survey of formal verification methods for smart contracts on blockchain, in: *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2019, pp. 1–6.
- [154] M. Dhawan, Analyzing safety of smart contracts, in: *Proceedings of the Conference: Network and Distributed System Security Symposium*, San Diego, CA, USA, 2017, pp. 16–17.
- [155] R. Sinnema, Extensible access control markup language (xacml) xml media type, 2013, URL: <https://tools.ietf.org/html/rfc7061> [Online]; accessed Aug 29, 2020.
- [156] M. Barnett, B.-Y.E. Chang, R. DeLine, B. Jacobs, K.R.M. Leino, Boogie: A modular reusable verifier for object-oriented programs, in: F.S. de Boer, M.M. Bonsangue, S. Graf, W.-P. de Roever (Eds.), *Formal Methods for Components and Objects*, 2006, pp. 364–387.
- [157] A. Mavridou, A. Laszka, E. Stachtari, A. Dubey, Verisolid: Correct-by-design smart contracts for ethereum, in: *Financial Cryptography and Data Security*, Springer International Publishing, 2019, pp. 446–465.
- [158] A. Mavridou, A. Laszka, Tool demonstration: Fsolidm for designing secure ethereum smart contracts, in: *Principles of Security and Trust*, 2018, pp. 270–277.
- [159] M. Maróti, T. Kecskés, R. Kereskényi, B. Broll, P. Völgyesi, L. Jurácz, T. Levendovszky, Á. Lédeczi, Next generation (meta) modeling: web-and cloud-based collaborative tool infrastructure., *MPM@ MoDELS* 1237 (2014) 41–60.
- [160] T. Abdellatif, K. Brousmiche, Formal verification of smart contracts based on users and blockchain behaviors models, in: *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2018, pp. 1–5.
- [161] J. Ellul, G.J. Pace, Runtime verification of ethereum smart contracts, in: *2018 14th European Dependable Computing Conference (EDCC)*, 2018, pp. 158–163.
- [162] R. Baldoni, E. Coppa, D.C. D'elia, C. Demetrescu, I. Finocchi, A survey of symbolic execution techniques, *ACM Comput. Surv.* 51 (3) (2018).
- [163] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, A. Hobor, Making smart contracts smarter, in: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 254–269.
- [164] J. Krupp, C. Rossow, Teether: Gnawing at ethereum to automatically exploit smart contracts, in: *27th USENIX Security Symposium (USENIX Security 18)*, USENIX Association, 2018, pp. 1317–1333.
- [165] I. Nikolić, A. Kolluri, I. Sergey, P. Saxena, A. Hobor, Finding the greedy, prodigal, and suicidal contracts at scale, in: *Proceedings of the 34th Annual Computer Security Applications Conference*, 2018, pp. 653–663.
- [166] T. Chen, X. Li, X. Luo, X. Zhang, Under-optimized smart contracts devour your money, in: *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2017, pp. 442–446.
- [167] J. Chang, B. Gao, H. Xiao, J. Sun, Y. Cai, Z. Yang, Scompile: Critical path identification and analysis for smart contracts, in: *Formal Methods and Software Engineering*, 2019, pp. 286–304.
- [168] C.F. Torres, J. Schütte, R. State, Osiris: Hunting for integer bugs in ethereum smart contracts, in: *Proceedings of the 34th Annual Computer Security Applications Conference*, Association for Computing Machinery, 2018, pp. 664–676.
- [169] Mythril, Security analysis tool for ethereum smart contracts., 2020, <https://pypi.org/project/mythril/> [Online]; accessed Aug 29, 2020.
- [170] J. Feist, G. Grieco, A. Groce, Slither: A static analysis framework for smart contracts, in: *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, 2019, pp. 8–15.
- [171] P. Cousot, R. Cousot, Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints, in: *Proceedings of the 4th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, 1977, pp. 238–252.
- [172] N. Grech, M. Kong, A. Jurisevic, L. Brent, B. Scholz, Y. Smaragdakis, Madmax: Surviving out-of-gas conditions in ethereum smart contracts, in: *Proc. ACM Program. Lang.*, Vol. 2, 2018.
- [173] P. Tsankov, A. Dan, D. Drachler-Cohen, A. Gervais, F. Bünzli, M. Vechev, Securify: Practical security analysis of smart contracts, in: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, Association for Computing Machinery, 2018, pp. 67–82.
- [174] L. Brent, A. Jurisevic, M. Kong, E. Liu, F. Gauthier, V. Gramoli, R. Holz, B. Scholz, Vandal: A scalable security analysis framework for smart contracts, 2018, CoRR abs/1809.03981. arXiv:1809.03981.
- [175] H. Jordan, B. Scholz, P. Subotić, Soufflé: On synthesis of program analyzers, in: *Computer Aided Verification*, Springer International Publishing, 2016, pp. 422–430.
- [176] E. Albert, P. Gordillo, B. Livshits, A. Rubio, I. Sergey, Ethir: A framework for high-level analysis of ethereum bytecode, in: *Automated Technology for Verification and Analysis*, 2018, pp. 513–520.
- [177] J. Lee, Rise of anonymous cryptocurrencies: Brief introduction, *IEEE Consum. Electron. Mag.* 8 (5) (2019) 20–25.
- [178] Cryptonote Standard, Cryptonote one-time keys, 2020, <https://cryptonote.org/cns/cns006.txt> [Online]; accessed Aug 29, 2020.
- [179] D. Chaum, Untraceable electronic mail, return addresses, and digital pseudonyms, *Commun. ACM* 24 (2) (1981) 84–88.
- [180] T. Ruffing, P. Moreno-Sanchez, A. Kate, Coinshuffle: Practical decentralized coin mixing for bitcoin, in: M. Kutylowski, J. Vaidya (Eds.), *Computer Security - ESORICS 2014 - 19th European Symposium on Research in Computer Security*, Wrocław, Poland, September 7–11, 2014. *Proceedings, Part II*, in: *Lecture Notes in Computer Science*, vol. 8713, Springer, 2014, pp. 345–364.
- [181] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J.A. Kroll, E.W. Felten, Mixcoin: Anonymity for bitcoin with accountable mixes, in: N. Christin, R. Safavi-Naini (Eds.), *Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3–7, 2014, Revised Selected Papers*, in: *Lecture Notes in Computer Science*, vol. 8437, Springer, 2014, pp. 486–504.
- [182] L. Peng, W. Feng, Z. Yan, Y. Li, X. Zhou, S. Shimizu, Privacy preservation in permissionless blockchain: A survey, *Digit. Commun. Netw.* (ISSN: 2352-8648) (2020) URL: <http://www.sciencedirect.com/science/article/pii/S2352864819303827>.
- [183] I.A. Seres, D.A. Nagy, C. Buckland, P. Burcsi, MixEth: Efficient, Trustless Coin Mixing Service for Ethereum, *Cryptology ePrint Archive*, Report 2019/341, 2019, <https://eprint.iacr.org/2019/341>.
- [184] S. Meiklejohn, R. Mercer, Möbius: Trustless Tumbling for Transaction Privacy, *Cryptology ePrint Archive*, Report 2017/881, 2017, <https://eprint.iacr.org/2017/881>.
- [185] R.L. Rivest, A. Shamir, Y. Tauman, How to leak a secret, in: C. Boyd (Ed.), *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security*, Gold Coast, Australia, December 9–13, 2001, *Proceedings*, in: *Lecture Notes in Computer Science*, vol. 2248, Springer, 2001, pp. 552–565.



- [186] M. Bellare, G. Neven, Identity-based multi-signatures from RSA, in: M. Abe (Ed.), Topics in Cryptology - CT-RSA 2007, the Cryptographers' Track At the RSA Conference 2007, San Francisco, CA, USA, February 5-9, 2007, Proceedings, in: Lecture Notes in Computer Science, vol. 4377, Springer, 2007, pp. 145-162.
- [187] P. Robinson, Requirements for ethereum private sidechains, 2018, CoRR abs/1806.09834. arXiv:1806.09834 URL: <http://arxiv.org/abs/1806.09834>.
- [188] T.P. Pedersen, Non-interactive and information-theoretic secure verifiable secret sharing, in: J. Feigenbaum (Ed.), Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings, in: Lecture Notes in Computer Science, vol. 576, Springer, 1991, pp. 129-140.
- [189] J.B. Bernabé, J.L. Cánovas, J.L.H. Ramos, R.T. Moreno, A.F. Skarmeta, Privacy-preserving solutions for blockchain: Review and challenges, IEEE Access 7 (2019) 164908-164940.
- [190] T. Kerber, M. Kohlweiss, A. Kiayias, V. Zikas, Ouroboros Cryptosinus: Privacy-Preserving Proof-of-Stake, Cryptology ePrint Archive, Report 2018/1132, 2018, <https://eprint.iacr.org/2018/1132>.
- [191] S. Goldwasser, S. Micali, C. Rackoff, The knowledge complexity of interactive proof systems, SIAM J. Comput. 18 (1) (1989) 186-208.
- [192] S. Brotsis, N. Kolokotronis, K. Limniotis, G. Bendiab, S. Shiaeles, On the security and privacy of hyperledger fabric: Challenges and open issues, in: 2020 IEEE World Congress on Services (SERVICES), 2020, pp. 197-204.
- [193] R3, Fact sheet, 2020, <https://www.r3.com/wp-content/uploads/2019/01/R3-Quick-Facts.pdf> [Online]; accessed Aug 29, 2020.
- [194] T. Koenig, S. King, M. van den Bos, C. van Wijk, A. Koren, Solutions for the corda security and privacy trade-off: Having your cake and eating it, 2020, <https://www.ingwb.com/media/3024436/solutions-for-the-corda-security-and-privacy-trade-off-whitepaper.pdf> [Online]; accessed Aug 29, 2020.
- [195] E. Androulaki, C. Cachin, A.D. Caro, E. Kokoris-Kogias, Channels: Horizontal scaling and confidentiality on permissioned blockchains, in: Computer Security - 23rd European Symposium on Research in Computer Security, ESORICS 2018, Barcelona, Spain, September 3-7, 2018, Proceedings, Part I, Vol. 11098, 2018, pp. 111-131.
- [196] N. Buxton, P. Madsen, S. Bryliski, Data privacy compliance using hedera consensus service, 2020, [https://www.hedera.com/HH\\_Data-Privacy-Compliance-Paper\\_200613.pdf](https://www.hedera.com/HH_Data-Privacy-Compliance-Paper_200613.pdf) [Online]; accessed Aug 29, 2020.
- [197] Ontology, Ontology infrastructure white paper, 2020, <https://ont.io/wp/Ontology-Infrastructure-White-Paper-EN-2.0.0.pdf> [Online]; accessed Aug 29, 2020.
- [198] I. Kotsiuba, A. Velkzhani, Y. Yanovich, I.S. Bandurova, Y. Dyachenko, V. Zhygulin, Decentralized e-health architecture for boosting healthcare analytics, in: 2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), 2018, pp. 113-118.
- [199] K. Ashritha, M. Sindhu, K.V. Lakshmy, Redactable blockchain using enhanced chameleon hash function, in: 2019 5th International Conference on Advanced Computing Communication Systems (ICACCS), 2019, pp. 323-328.
- [200] D. Grigoriev, V. Shpilrain, RSA and redactable blockchains, Cryptology ePrint Archive, Report 2020/069, 2020, <https://eprint.iacr.org/2020/069>.
- [201] Tari, The protocol for digital assets, 2020, <https://www.tari.com/#what-is-tari> [Online]; accessed Aug 29, 2020.
- [202] N. Klokliang, P. Teawtim, P. Aimtongkham, C. So-In, A. Niruntasukrat, A novel IoT authorization architecture on hyperledger fabric with optimal consensus using genetic algorithm, in: 2018 Seventh ICT International Student Project Conference (ICT-ISPC), 2018, pp. 1-5.
- [203] Z. Qiu, J. Hao, Y. Guo, Y. Zhang, Dual vote confirmation based consensus design for blockchain integrated iot, in: NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium, 2020, pp. 1-7.



**Sotirios Brotsis** received the B.S. degree from the Department of Mathematics, Greece, University of Patras and the M.S. degree from the Department of Informatics and Telecommunications, Greece, University of Peloponnese. He is currently pursuing the Ph.D. degree with the same department, supervised by the Associate Professor N. Kolokotronis. His research interests lie in Blockchain technology, cryptography and IoT security. He has published his research in top venues, and he has been a TPC member and a regular reviewer for international journals and conferences.



**Konstantinos Limniotis** received the Ph.D. in cryptography from the National and Kapodistrian University of Athens in 2007. He is currently an ICT Specialist at the Hellenic Data Protection Authority. He also holds visiting positions, as an Adjunct Faculty Member, at the Open University of Cyprus and the National and Kapodistrian University of Athens, Greece. In addition, he is a research associate with the University of Peloponnese, Greece, since 2017. His main research interests include cryptography, personal data protection and information security. He has over 40 publications in these areas, whilst he is regular reviewer in international journals and conferences.



**Guletoom Bendiab** has a PhD degree in Computer Science. Her research interests are in the areas of trust and identity management, Blockchain, machine learning with applications in cyber security and most specifically in digital forensics, Intrusion Prevention Detection and Response, Social Engineering, OSINT, Insider Threats, Malware and Distributed Denial of Service Attacks.



**Nicholas Kolokotronis** is an Associate Professor and head of the Cryptography and Security Group at the Department of Informatics and Telecommunications, University of the Peloponnese. He received his B.Sc. in mathematics from the Aristotle University of Thessaloniki, Greece (1995), M.Sc. in highly efficient algorithms (1998, highest honors) and Ph.D. in cryptography (2003) from the University of Athens. He has been a cochair and member of the organising committee in many international conferences. His research interests span cryptography, security, and coding theory, areas in which he has published more than 85 papers in international scientific journals, conferences, and books.



**Stavros Shiaeles** is currently Senior Lecturer at University of Portsmouth, UK. He holds various professional certifications in Cyber Security and he is an EC-Council Accredited Instructor, delivering CEH to professionals. He has published more than 70 articles in international scientific journals, conferences and books, he has been a TPC member and a regular reviewer for a number of international journals and conferences, and he is actively involved with EU-funded and national research and development projects. His research interests span the broad areas of cybersecurity, open-source intelligence, trust, blockchain, digital forensics, and machine learning applications in cybersecurity.