

A Survey of Secure Internet of Things in Relation to Blockchain

Morteza Alizadeh^{1*}, Karl Andersson¹, and Olov Schelen²

¹Pervasive and Mobile Computing and Laboratory

Luleå University of Technology, Skellefteå, Sweden

{morteza.alizadeh, karl.andersson}@ltu.se

²Pervasive and Mobile Computing Laboratory, Luleå University of Technology, Luleå, Sweden
olov.schelen@ltu.se

Abstract

Distributed ledgers and blockchain technologies can improve system security and trustworthiness by providing immutable replicated histories of data. Blockchain is a linked list of blocks containing digitally signed transactions, a cryptographic hash of the previous block, and a timestamp stored in a decentralized and distributed network. The Internet of Things (IoT) is one of the application domains in which security based on blockchain is discussed. In this article, we review the structure and architectures of distributed IoT systems and explain the motivations, challenges, and needs of blockchain to secure such systems. However, there are substantial threats and attacks to blockchain that must be understood, as well as suitable approaches to mitigate them. We, therefore, survey the most common attacks to blockchain systems and the solutions to mitigate them, with the objective of assessing how malicious these attacks are in the IoT context.

Keywords: Distributed Systems, Blockchain, Internet of Things, IoT Architectures, Security, Attacks

1 Introduction

Most devices around us use the Internet for their communications. Mobile and wireless technology, such as 5G in combination with security innovations, will help to increase the use of the Internet of Things (IoT) [31].

IoT systems need to use service architectures that are decentralized or distributed to increase their performance when encountering numerous sensors, users, and computational elements [51, 73]. Researchers are investigating new solutions to address scalability.

It has been discovered that malware, botnets, and several types of attacks exist that target millions of IoT devices, such as smart cameras and routers. The security problem is a significant issue in IoT systems [70]. As an example, research has found cybersecurity issues in network protocols, such as LoRaWAN [37], that could put network users at risk of attack by bringing greater numbers of unauthorized devices into the enterprise with potentially serious consequences. Thus, IoT developers should consider the possibility of attacks. Service providers such as Google and Amazon provide a variety of capabilities, such as transaction analysis and secure storage management. They help deploy IoT systems and applications in distributed centralized networks. Researchers are trying to find a distributed decentralized solution to avoid the centralized ownership of data.

Blockchain is an invention defining immutable transactions and distributed consensus [56] that can be used as a security solution for IoT systems. The immutability and consensus for each transaction do

Journal of Internet Services and Information Security, volume: 10, number: 3 (August 2020), pp. 47-75

DOI: 10.22667/JISIS.2020.08.31.047

*Corresponding author: Pervasive and Mobile Computing and Laboratory, Luleå University of Technology, S-93187 Skellefteå, Sweden. Tel: +46910585368

not rely on centralized authorities. IoT applications can execute transactions securely in noneditable and distributed environments with the help of the blockchain system [56]. There are many various unknown attacks in blockchain, which can potentially make it insecure. Thus, there is a need to assess these threats and solutions.

This article focuses on the IoT and blockchain, including architectures and security problems. We discuss the use of blockchain as a tool for improving security in IoT services and networks. We show that blockchain is a secure technology for decentralized distributed systems. Additionally, blockchain is highly fault-tolerant and does not require human calculations. These are properties that can help blockchain remain secure from malicious attackers, i.e., by making it too expensive to attack the system.

The rest of this article is organized as follows. Section 2 explains distributed and decentralized service and network architectures. Section 3 discusses the definition of blockchain, how it can solve the transaction storage problem in distributed networks, and its purpose. We explain three types of blockchain that are common: public, private, and consortium. We present common consensus algorithms that are popular to use in these systems. Section 4 introduces IoT architecture and security layers. Section 5 discusses challenges existing in the IoT system in combination with blockchain. Section 6 describes possible attacks on the blockchain system and the strategies for detecting and preventing these attacks. We provide a comparison between them. Section 8 concludes the article.

2 Service and Network Architectures

In this section, we explain the decentralized and distributed systems as two types of essential architectures in blockchain.

There are several topics considered in IoT systems. IoT architectures [53, 57], software problems [43, 65, 84], and privacy [69] have been studied. There are studies that have investigated blockchain security [87, 52], which is a popular topic in IoT research. Security levels are typically higher in blockchain-based systems than in systems without blockchain in a large-scale network [21, 42, 71]. There are many surveys and review papers that discuss the challenges modeled by privacy issues. Many of them deal with cryptocurrency [50, 23, 10], and in some cases, they also review the security of the blockchain system [48, 50, 12]. Other articles discuss the flexibility of using blockchain [61] and how to use blockchain in industry [2].

2.1 Architecture Categories

Decentralized systems should have no single point of failure and should not be dependent on one server or central party. Normally, such a system has multiple authoritative parties, each of which serves a subset of the workload. Although the security risk potentially rises by increasing the number of nodes, this architecture may solve many problems, such as scaling, the independence of central servers, and the extensive use of network traffic.

The distributed system is typically fully connected, where parties are located in different places. There is no significant difference between parties. They communicate and coordinate their actions by sending and receiving messages directly to other parties [39]. A distributed system can consist of many devices, such as applications, users, workstations, and IoT devices. Many features can represent the fundamental ideas of distributed systems, such as resource sharing, concurrency, scalability, and transparency [39].

2.1.1 Network Architecture Analyzing Views

The physical view and the development view are two aspects by which to analyze system architecture. Amazon and Google are two popular services that are widely distributed when we look at them through a physical or deployment viewpoint. On the other hand, they are centralized by the cloud service provider when we look at them through the aspect of development or management control. Therefore, clouds are a distributed centralized system. The distributed system has an entirely technical meaning. It is separated geographically in terms of storage or processing for partitioning, dispersing, part ownership of storage or computational capacity. Decentralized systems are still distributed in a technical sense, and no single member owns a decentralized system; otherwise, it would not be decentralized [44]. This means that most systems are centralized similar to clouds.

2.1.2 Network and Service Architecture Definition

The network architecture is a structure for the specification of a network's physical components and functional organization and configuration. Peer-to-peer (P2P) is an example of a network architecture. A P2P network includes a group of devices with equal power that perform the same tasks and can store and share files. A P2P sometimes refers to cryptocurrencies via a distributed network in the combination of financial technology [62].

Service architecture is a helpful method for developing software that continues to work when something fails. In other words, a service is software that delivers automated duties to hardware events or listens for data requests from other components without user interaction. Therefore, it is known as software architecture in many references. The cloud is an example of services. The cloud is used to describe data centers available to many users over the Internet. There are three ways to use the cloud as a service, namely, information as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS), all of which refer to a service-oriented architecture [80]. The meaning of cloud computing is to access resources owned and managed by a third party via the Internet. It is a technology used to deliver computing resources based on long-existing technologies such as server virtualization. The cloud involves the transfer, storage, and processing of information on the provider's infrastructure, without direct active management by the user [9].

2.1.3 Network and Service Architecture in Relation to Blockchain

Digital money can be transferred from one user to another through a P2P network with the help of a distributed ledger called blockchain. The P2P architecture offers decentralization, and most trades are executed without the need for mediators. Therefore, the P2P architecture is a solution for use cases with sharing programs. Today, P2P networks are at the core of distributed computing applications, cryptocurrencies, and blockchain.

The relation among services and blockchain concerns storing data in various applications, such as recording transaction details, banking, contracting, and IoT systems. Regularly, blockchains store information in the cloud, which makes the information more secure and robust. The information stored in the blockchain system depends on the services and the applications. It could be used in a P2P file system in distributed databases such as Apache Cassandra or a cloud file storage system such as Ethereum swarm. [40].

2.2 Service and Network Design Goals

The system should work without interruption all the time, even during equipment failure and system overloads. The system should be compatible with various applications and provide reasonable response

times from P2P devices. The system's environment should be secure. It should protect both the data transmitted over it and the data stored on the devices connected to it. Additionally, the system should be simple to modify and update, such that obtaining and solving problems should not be too time-consuming. Many different parameters are related to the system design. These parameters turn into five major design goals [66].

The first goal is availability. Availability is about delivering consistent services with reliable performance that work without interruptions or the quality of being present or ready for immediate use [13]. For example, a peer-to-peer system may replicate files on devices. However, some of them might be turned off and then turned on at some later time. The availability of the hosts is dependent on time. The availability of the file is a function of time. In addition, the failure of a single link or piece of equipment should not introduce a significantly negative impact on network performance.

The second goal is that of scalability. Scalability is the feature in which a system handles more tasks after adding more resources. There are two types of scaling: horizontal scaling, which means that the system's size changes by adding more separate resources in parallel into the network or the system, and vertical scaling, which means that the system resizes itself by adding more resources such as CPU and RAM to an existing node. As a general example, a horizontal and vertical scalable network model means that a network with multiclient support can increase the number of clients who can store information on the server, given increased resources such as hard drives, processors, and memory on the server side [15, 36]. Scalable service designs can accommodate new user groups and remote places and can support new applications without changing services. An excellent scalable distributed system is similar to a system that should follow the horizontally scalable goal by adding many resources and nodes.

The third goal is security. Security is a feature that should add to the system when it is working. Designing the best secure location of devices, filters, firewalls, hardware, and embedded server software aspects are critical sections in network resources.

The fourth goal is affordability or cost-effectiveness. Most clients have a goal of affordability, and it is partly a business-oriented goal. Network design should carry the maximum amount of traffic for a given financial cost. Financial costs include nonrecurring equipment costs and recurring network operational costs.

The last goal is manageability. The available network parties must support the network and manage it. Effectiveness and efficiency cannot be achieved by having a network design that is too complicated. Developers need to know that the devices within the systems are designed in a layered approach at the second stage of the network design. For example, there are three underlying layers in the hierarchical network design model, namely, the core layer that connects the distribution layer devices, the distribution layer that interconnects the smaller local networks, and the access layer that provides connectivity for network hosts and end devices.

Network performance is often defined through a service level. It is the primary purpose of network design. Other requirements are about determining the features and functions required to meet the needs identified in the previous stages, such as performing a network-readiness assessment and creating a project plan.

2.3 Service and Network Discussion

Blockchain has a conflict with a centralized network when we look at it from a physical viewpoint. As an example, data distributed over the cloud are stored on one company's centralized set of data centers, but data in the blockchain system are stored over a group of servers. The blockchain system is not just about decentralization; it is a distributed system technology [30].

In conclusion, for using blockchain technology such as Bitcoin, a system must be decentralized when we consider it through the physical viewpoint, which means that separated yet connected parties

on a large scale within the network are parts of the system's association. Decisions will be made after consultation and consensus with all members. Additionally, the developer must consider the five abovementioned design goals for their systems' design.

3 Overview of Blockchain Technologies

The distributed ledger is information that is published across several nodes or networked devices. All nodes replicate and preserve the same copy of the ledger. The distributed ledger updates itself individually. The agreed-upon version of the ledger by other members of the network is stored on each node separately. Distributed ledger technologies reduce the cost of trust. The architectures and structures of distributed ledgers can help us decrease our necessity of governments, agents, clerks, and assent officers.

Blockchain is known as a new invention that shares an immutable digital ledger of transactions as digital information to a distributed system. This distributed ledger provides immutability, trust, and data security [46]. This technology is helpful for cryptocurrencies, record keeping, digital notary, and smart contracts. Blockchain stores transactions on distributed networks. It was used initially for digital currency [23] and secure distributed storage systems [90]. Bitcoin is a famous example of cryptocurrency. It uses decentralized digital currency by removing a central management or administrator such as a bank [85]. Increasing trustworthiness is a result of using cryptography. It makes it easy to transfer different kinds of assets P2P over the Internet [64].

Blockchain contains a set of blocks where each block is a data package. The blockchain length increases when new blocks are added and thus represents a complete ledger of the transaction history. Other parties can take part in a new block validation process to gain rewards. The data in each block contain a timestamp and a hash code, whereas each block knows the previous hash code, similar to parents and children. It is possible to track back to the first block or the genesis block by following this process. This theory secures the integrity of the whole blockchain. The hash value is unique for each data point. The blockchain's block will be added when most of the parties in the network agree by a consensus mechanism. Hence, new transactions are not automatically added to the ledger. The stored information in the blockchain is unchanged.

Sections 3.1 and 3.2 discuss the categories of blockchain based on consensus algorithms such as permissioned and permissionless blockchain systems. Sections 3.3, 3.4, and 3.5 discuss three types of blockchain: public, private, and consortium. Section 3.6 explains the main differences between them. At the end of this section, section 3.7 presents components in blockchain, the hash, smart ledger, and smart contracts.

3.1 Consensus in Blockchain

Consensus algorithms are the central entity of blockchain systems. There are many works that show consensus algorithms in distributed systems [88]. Consensus algorithms try to make a secure environment for transactions or replicas by use of consensus protocols to ensure all that replicas of the shared agreement at any given time are secure. The time for validating a new transaction or producing the result of a consensus is known as a main issue to be improved, and it is essential for blockchain systems, no matter whether that transaction is correct and secure.

3.2 Permissioned and Permissionless Blockchain

Several methods exist for consensus that we can divide into two broad categories: permissioned and permissionless blockchain systems. It is necessary to know which one is a better solution for the specific system targeted.

3.2.1 Permissioned

In this type of system, a limited number of known trusted participants carry a copy of the blockchain's ledger. Researchers examine this type of system in private and consortium blockchains. It is faster and does not need costly functions for publishing a new block. For example, there is no risk of a Sybil attack (see Section 6.15 for details). Researchers believe that permissioned blockchains have a significantly higher performance level compared to permissionless blockchains [5]. Due to the conditional and smaller creation of permissioned blockchains such as banks and shipping firms, these blockchains tend to be more secure and faster. Additionally, they are often more centralized than permissionless blockchains.

As mentioned above, private and consortium blockchains allow only a few known participants to keep and store a copy of the entire blockchain or ledger. Furthermore, miners are perceived by others, so there is no risk of an attack such as the Sybil attack [67]. Therefore, voting and averaging are used as mechanisms to achieve consensus. Practical Byzantine fault tolerance [38] works similar to voting as a famous consensus algorithm in a private permissioned blockchain.

3.2.2 Permissionless

Public blockchains are termed permissionless in that they permit anyone to keep a copy of the blockchain and to be involved in the validation process to publish new blocks. There are famous networks in the world that use permissionless blockchain, and they usually work on digital currencies. Therefore, all nodes have the right to act as a simple member on the system or employ the mining protocols to help verify transactions. A consensus algorithm is designed to convince members that a new block added to the blockchain is unique and the only version of the truth that is accepted by all the members in the blockchain.

There are at least five types of consensus algorithms that exist in public and permissionless blockchain systems [5, 23]: 1) proof of work (PoW), 2) proof of stake (PoS), 3) proof of X, 4) proof of activity, and 5) proof of elapsed time. We will explain proof of work and proof of stake as two consensus algorithms, such that PoW is compatible for the public and PoS is better for private blockchain.

PoW is a consensus process for validation. The base is that all nodes can join in publishing new blocks to the blockchain as they want to compete for a prize, which is referred to as mining. All the time, miners are trying to compete with others in a race for increased mining. Finally, the winner adds the block to the chain and broadcasts their block to the network. Then, the network allows all the participating nodes to know the new block published by the miner [5].

PoS is similar to the PoW algorithm, but we have validators instead of miners. One of the validators is enough to publish a block over the blockchain by a specific policy. One validator is chosen randomly or by policy, with the equal probability of being selected in the network. A new block is chosen randomly but, depending on its wealth, it is also defined as the stake size, which means that blocks with a larger size of the stake have a better chance to be selected. Table 1 illustrates a simple comparison between blockchains by considering parameters such as consensus, immutability, efficiency, identity, and transaction speed [91]. All nodes can participate in consensus decisions, whereas the voting power depends on the amount of money, which is a disadvantage in comparison with PoW; however, PoS blockchains are faster compared to PoW blockchains [5] since there is no need to make competition between validators and no need to choose a specific validator.

3.3 Public Blockchain

Public blockchains are fully distributed and decentralized, such that all nodes can participate in publishing new blocks and accessing blockchain contents. Public blockchains are termed permissionless in that they allow anyone to maintain a copy of the blockchain and join in validating new blocks. A public

blockchain network is entirely open, and anyone can join the network. Public blockchain technology can be a way to store and share rewards. The network mechanism encourages more participants to join the network. They are great platforms where all users should be treated equally. Most public blockchains are currently used to support cryptocurrencies. There are many public blockchains that work with a cryptocurrency, such as Bitcoin, Litecoin, and Ethereum. In these blockchains, anyone can own some money or rewards to trade with others and thus take part in the competition for rewards. This leads to an open permissionless model with full transparency. Public blockchains resist being hacked since it would be too costly to tamper with the contents [91].

3.4 Private Blockchain

Private blockchain is distributed and permissioned. Certified users of a single organization can join the network. A private blockchain can provide a high level of security, privacy, compliance, performance, and many of the properties related to being either open-sourced, consortium, or privately developed. There are many options for a private blockchain, and the most common ones are Hyperledger and Quorum. Selected nodes can process transactions in the blockchain. Transactions are not publicly transparent in the blockchain, and only chosen nodes can access the ledger. The rest of the blockchain does not participate. Private blockchains do not require currency or tokens to function, and there are no processing fees included in their transactions. Since blocks are published by delegated nodes within the network, a private blockchain is not as tamper-resistant as a public blockchain, and the organization may choose to roll back their blockchain to any point in the past.

As an overview of a private blockchain, such a system functions as a distributed centralized system, where all things are returned to how the network's private owner wants them to be and another party is needed to approve and manage everything [91, 47]. Therefore, the degree of distribution is low, as shown in Table 1. This means that all transactions or communications are dependent on some parties in the network.

3.5 Consortium Blockchain

Consortium blockchains or federated blockchains are similar to private blockchains in the sense that it functions as a permissioned network. It is a distributed and partially decentralized system, and the responsibility is shared among the members of the consortium. Consortium networks span multiple organizations and help maintain transparency among the involved parties.

In other words, the consortium blockchain combines elements from both public and private blockchains. The main difference regards the consensus. Instead of a public blockchain where anyone can validate blocks and a private blockchain where a closed single entity appoints block producers, a consortium chain considers some of the equally powerful parties or organizations to act as validators, such as multiple organizations operating in the same industry.

A consortium blockchain is used as an auditable and synchronized database that keeps track of data exchanges taking place between the participating consortium members [74]. The consortium blockchain does not involve processing fees, and it is not computationally expensive to publish new blocks. While it does provide lower latency in transaction processing, it is not entirely decentralized and distributed [6].

3.6 Comparison of Public, Consortium, and Private Blockchains

There is a comparison among public, private, and consortium blockchain, as shown in Table 1. There are six properties used for this comparison. The consortium blockchain decreases the risks of a private blockchain by removing centralized control, and a smaller number of nodes in consensus generally allows

them to be much faster than a public chain. It can be set up to be a more public consortium blockchain or a more private consortium blockchain. It depends on whether the system needs to be more open or closed. In other words, a consortium blockchain is designed to depend on individual requirements. When the system needs to have a high level of distribution and nonfree transactions, it is better to use a public blockchain. When one needs a system with a high security level, needs to know all the parties on the network and there is no need to pay for services, it is better to use a private blockchain. In addition, one needs to consider a consortium blockchain when it is necessary to customize the system to be more closed or more public and have multiple centralized parties.

Table 1: Comparison of public blockchains, consortium blockchains, and private blockchains

Criteria	Public Blockchain	Consortium Blockchain	Private Blockchain
Consensus Determination	All	Multiple Organization	One Organization
Read permission	Public	Can be Public or Restricted	Less Public or High Restricted
Immutability	Nearly Impossible to Tamper	Can be Tampered	Can be Tampered
Identity	Pseudo-Anonymous	Approved Participants	Approved Participants
Distribution	High	Medium	Low
Consensus	Permissionless	Permissioned	Permissioned

3.7 Main Blockchain Procedures

There are several significant procedures that exist in the blockchain. The issue of how to chain the blocks with the ability to keep the data immutable against changing is one of them. This procedure can be used when a new block of data is coming to connect the previous blocks. Hashing in the blockchain is another part. This procedure can be useful when we use it to encrypt the data as a unique form. The last one is a smart contract. This procedure is similar to a contract, but it includes lines of code that are stored on a blockchain and are automatically executed when two parties in the network want to make a contract after negotiating a transaction.

3.7.1 Basic Mathematics Instruction

As we mentioned earlier, a blockchain is a group of chained data that contains parts of information such as the block's name, the block's hash code, and the previous block's hash code. Thus, all the blocks chain to each other by knowing the previous block's hash code. For example, data in a block in a blockchain can be contract or transaction information, such as the amount of money transferring from one person to another. Every hash code is unique and specific for each block; thus, we can consider it as a fingerprint for blocks [85, 23]. However, hash codes have many applications, such as encryption, lookup tables, and databases. A hash code is another form of a string of characters changed into a usually shorter fixed-length value or key that represents the original string. This code is used to index and reclaim items in a database because it is faster to find the problem by using the shorter hashed key rather than find it by using the original long value.

Another form is that of a distributed hash table (DHT). This is a distributed system that has a lookup service, and it uses a hash table where (key, value) pairs are stored in a DHT, and any participating

node can efficiently recover the value linked with a given key. All the nodes can be added/removed at a minimum time just by redistributing the keys [55]. This system can help blockchains be faster.

A hash code is a numeric value that helps in the identification of a thing during identity testing, and it can work for object indexing. It is better to say that it is too difficult to change the value of a hash code. The purpose of the hash code is to help in the effective lookup and insertion of data collections that are based on a distributed hash table. A block consists of two main sections, the block header and the block body.

The block header includes the block version and the parent block hash value that is shown to the system that is chained to the previous block as a chain of hash codes. The Merkle tree [5] includes all of the transactions in the block and the timestamps.

The block body is made of at least two parts, a transaction counter and transaction data. The maximum number of transactions that can be handled by a block depends on the block size and the size of each transaction. Blockchain uses an asymmetric cryptography mechanism such as digitalization to validate the authentication of transactions. Digital signature is a main method based on asymmetric cryptography used in an untrustworthy environment with private and public keys. This method can be added to an electronically transmitted document or contract to verify its content and the sender's identity [86].

3.7.2 Smart Contracts

Smart contracts provide the connection of credible transactions without third parties. These transactions are trackable and immutable. Usual forms of using smart contracts are exchanging money, making investments, trading shares, or many other use cases in e-commerce. In this approach, an asset or currency is transferred by a program, and the asset should be paid to another party or forwarded to another person, or else it should be paid to the person who sent it. These systems store their contracts as data.

4 IoT ARCHITECTURE

This section explains a theoretical realization of IoT architectures in recent studies. The networks need to be supported by flexible and expandable infrastructures and protocols.

There are many sets of related components that build the IoT. Sensors are working in the first outer layer, referred to as things. Other devices are part of the IoT infrastructure, such as gateways and access points. A considerable number of devices with unique identifiers are necessary due to the large scale of their deployments. Communication elements are needed to connect the heterogeneous devices that provide fast services and fast communication foundations. They use radio frequency identification (RFID), wireless sensor networks (WSNs), and protocols such as IEEE 802.11 (WiFi), IEEE 802.15 (Bluetooth), and IETF low-power wireless personal area networks (6LoWPAN) [73] for their communications.

4.1 Architecture Layers

There are different ideas concerning the number of layers in the IoT. The IoT mostly operates on three layers [54], termed perception (sensors and actuators), network (communications), and application layers. Each layer has security issues that are correlated to it. Additionally, other approaches consider four or five layers for the IoT [60]. The first layer is the perception layer, whose duty is to receive data from sensors (collect big data that is generated by IoTs) and then transfer and process the information to the next layer called the object abstraction layer. Cloud and fog computing and data management processes are in the second layer. The cloud layer refers to the software in the background of the IoT solution. IoT cloud providers are expected to provide secure cloud services to protect from significant data gaps or solution downtime issues. Sending the data generated in the first layer to the third layer through secure

methods is a primary duty of this layer. The third layer's duties are about finding services with their requester by having addresses, names, and processes data. Service management is a layer that consists of services with delivering ability beyond the network. The application layer, known as the fourth layer, provides services requested by customers. It can provide services according to the needs of the customer. The fifth and final layer is called the business layer, and it manages the overall IoT system. The last layer is essential for development and used for changing the system. There are many duties, such as design, analysis, monitoring, and decision-making, based on big data. Privacy control of the users is also handled in this layer [81].

4.2 Security and Privacy

The IoT includes several prevalent components, such as trust, access control, data integrity, physical tampering, user privacy, and security. The risk of attacks in an unsecured environment with less privacy is high. Security and privacy are significant problems and are known as the main requirements for systems. IoT devices' companies have found themselves at the center of user privacy scandals. Recent reports claim that IoT applications send an excessive amount of personally identifiable data to third parties [69]. When a third party is present in the network, it means that there is an external actor who is monitoring and managing all activities all the time.

There exist many security issues, such as physical security, data at rest, chip security, secure booting, device authentication, and device identity, which are the device layer's components. Additionally, there are many issues such as access control, firewall/IPS/IDS, and end-to-end encryption components in the communications layer. Among all the layers, the application layer should support the authenticity, integrity, and confidentiality of the data.

The categorization of different attacks in different clusters, such as the access level, location, devices, and protocols, is considerable. Most attacks are in areas related to data management and commit transfers, such as data exchanges, generate fake messages, send false messages, and create routing loops between nodes to be successful. Social media, such as Facebook, is not entirely secure in regard to these attacks.

4.3 IoT and Big Data

Data from IoT devices are always multiform or unstructured. The data need to be analyzed after being received by the servers. Additionally, there are many challenges that still need to be addressed in this area. The analysis of big data generated by the IoT is not simple. The primary constraints are energy, time, and resources. As a result, big data and IoT systems need to develop together in parallel [19].

5 Summary of IoT with Blockchain

This section discusses the major problems of the IoT and discusses the intersection of the IoT with blockchain.

5.1 IoT Systems Concerns

Peer-to-peer communications beyond devices have security problems. The ultimate IoT goal is to obtain data securely at the right place, at the right time, and in the right format. Protecting sensitive data in peer-to-peer messaging, distributed file sharing, and autonomous device coordination are significant gaps in the IoT.

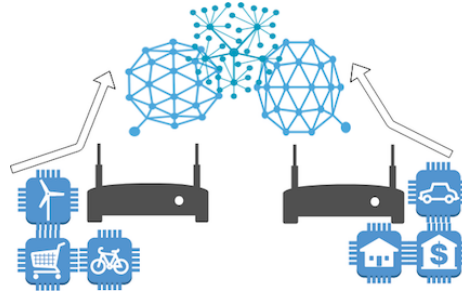


Figure 1: Gateway devices as end points

5.2 Intersection of IoT with Blockchain

Today, IoT users need a system for controlling their data and information, such that every party in the network has its own version of the current state. Blockchain can be used in IoT systems with distributed ledgers. Therefore, every trusted party in the IoT system will have access to the same information, and they can send their responses to the network, which is agreed or disagreed on in their consensus [22]. Members who are identified can only access records that are shared and secured at the same time, specifically in private blockchains. Additionally, managers can access data by combining hash code advantages in the large scale of IoT systems [11].

There are many uses of blockchain in the industry, for example, building a trust environment between devices that are connected and solar panels that can buy and sell energy. They record their outputs on the blockchain and sell it to other users in the area via smart contracts.

5.2.1 Decentralized Distributed Blockchain Gateway Level

In a decentralized distributed blockchain integration scheme, all communications go through the blockchain, while the IoT gateways act as end-points to the blockchain network [3, 49, 25]. In this case, the IoT devices will be registered to the gateway device, and the gateway performs transactions to the blockchain on behalf of the IoT devices. This approach enables the traceability of all communications involving a specific IoT gateway and IoT services. This integration scheme can also be used to authenticate communications between devices connected to separate blockchain enabled gateways.

In this approach, the rest of the transferred data needs to be stored on the blockchain. The blockchain itself can be used as a control mechanism, with smart contracts acting as programmable logic, while data can be stored over P2P technologies such as BitTorrent and IPFS [4]. Fig. 1 is an illustration of this approach.

5.2.2 Decentralized Distributed Blockchain IoT Level

Interconnected edge devices as end-points to the blockchain are similar to the previous approach, and all IoT interaction events are logged into the blockchain for secure accountability [49, 25]. In this approach, IoT devices can be provided with cryptographic functionality. The trade-off here is a higher degree of autonomy of IoT devices and applications versus increased computational complexity of the IoT hardware, as shown in Fig. 2.

5.2.3 Distributed Blockchain Edge Devices Level

In this approach, IoT gateways and devices issue transactions to the blockchain and can communicate with each other [49, 76, 1], as shown in Fig. 3. This approach ensures low latency between the IoT



Figure 2: IoT devices as end points



Figure 3: Edge devices as end points

devices and chooses specific interactions on the blockchain.

5.2.4 Cloud Blockchain Hybrid with the IoT Edge

This approach is an extension to the previous integration scheme, whereas the IoT users have a choice to use the blockchain for certain IoT interaction events, and the remaining events occur directly between IoT devices [49]. This approach leverages the benefits of decentralized record keeping through blockchains as well as real-time IoT communication. Fig. 4 is an illustration of this hybrid integration schema. The challenge posed by this approach is to optimize the split between the interactions that occur in real-time and those that go through the blockchain. Hybrid approaches can utilize cloud computing to overcome the limitations of blockchain-based IoT networks such as storage. Which integration schema to implement depends upon the requirements of the IoT application.

5.3 Blockchain Concerns

Blockchain is secure, but there are some problems remaining in this area. The mining process is very complicated [17], and IoT devices typically work with limited resources [7]. The mining time of blocks is long [32], and it is a problem where time is highly considered [26].

The risk of attacks from threats grows exponentially. The goal of the IoT-based blockchain is to automate functions while keeping protection focused on the threat of a different range of security attacks. We can divide security issues into four sections: centralized, decentralized, cloud-based, and hybrid architecture of the IoT system. In the IoT network, nodes at the edge are at risk of failure. A set of

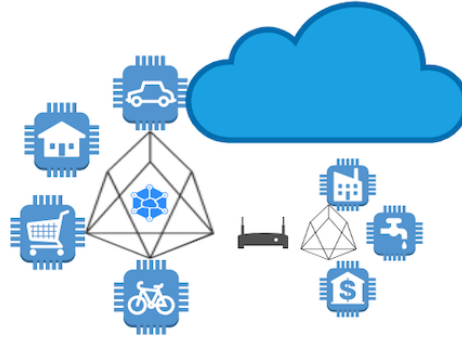


Figure 4: Hybrid with IoT-Edge

corrupted nodes and devices can be a problem that fails the IoT service provision within the IoT edge. Blockchains have the potential to increase the security level of infrastructure for the IoT by providing a secure public key that is more fault-tolerant than centralized solutions.

Increased energy usage is another problem of blockchain [61], as it should deal with system priorities and system needs. After that, we can decide which one is better, as is it essential to know the correlation and contradiction of energy and security in each system and how much energy can be used to keep systems secure.

5.4 Discussion

Recording all IoT interaction events on a blockchain will increase bandwidth and storage requirements. Security is a well-known research challenge towards the integration of the blockchain and the IoT. Most of the abovementioned integration schemas would be more suitable to scenarios where interactions are much more frequent and high throughput, low latency, and secure data are required. The last presented model is favored when immutable record-keeping and the lower number of interactions are the main requirements.

The model presented in section 5.2.4 shows the IoT in combination with blockchain. This combination solves many problems, such as scalability and security, that are still being considered. There are many parameters for comparing these architectures, as shown in Table 2. Blockchain use can prevent most security attacks by developing immutable records and a distributed consensus, which are two features of blockchains. They make the cryptocurrency network system more secure because these options can protect such systems against double spending attacks, which is a major issue in cryptocurrency. We will explain possible attacks to the blockchain in Section 6.

6 Attacks

This section discusses fundamental problems around the security of the blockchain and its mechanisms. There are many types of attacks on blockchains, such as a massive range of wallet attacks, network attacks, and miners' attacks. These attacks can be destructive for the blockchain and affect the whole system.

Table 2: Comparison of IoT architectures in relation to blockchain

Criteria	Decentralized Distributed Blockchain Gateway level	Decentralized Distributed Blockchain IoT Level	Distributed Blockchain Edge Devices Level	Cloud Blockchain hybrid with the IoT edge
Requirements	Gateways should be registered	IoTs need to register	Should all party register on network	No need all party to register, just those are in blockchain
Traceability	Low	Low	High	High
Security	Very High	High	High	High
Privacy	Low	Medium	Medium	Medium

6.1 Race Attack

The race attack is a kind of double-spending attack, meaning spending the same money multiple times. The double-spending attack is prevented through block validation.

This type of malicious behavior by a user is termed double spending, as shown in Fig. 5. The attacker sends the same transaction to multiple users and waits for victims to accept the requests.

A race attack can occur in public blockchains, but it is difficult in private blockchains because of their central management units. Storing multiple copies of the ledger in networks is known as a solution in the system, but it has a privacy problem. All the users act as parties to store and keep a copy of the ledger in a public blockchain. For example, a user may send many different transactions with the same set of blocks to many receivers at the same time, but it requires much power to change the agreement.

The probability of a race attack is low because of the validation process. The common solution for this attack is to wait for the validation process to finish completely [23].

6.2 Finney Attack

A Finney attack is a variation to the double-spending attack, where the attacker’s purpose is to participate in mining to destroy a mined block that is approved by consensus.

Generally, such an attack occurs when a merchant accepts unconfirmed transactions [34]. The attacker abuses the verification time, which is a time gap to ensure that everyone agrees that a transaction payment is valid. The attacker is mining blocks all the time. In each block, the attacker includes a transaction to A, which he controls. When the attacker succeeds in mining the block, he does not broadcast it but rather sends another transaction to victim B and victim C. There will be a waiting period before the transaction agreement with the other parties arrives. When the victim does not hear anything from the network for a long time, the victim will either send the goods to the attacker or cancel the transaction. The attacker will be happy if the victim accepts and sends the goods. Then, the attacker broadcasts his private block and sends the money to himself. The attack ends with the attacker having both the goods and the currency.

The probability of a Finney attack is low because the nodes must wait for multiconfirmations for transactions to finish [23].

6.3 Brute-Force Attack

A brute-force attack is yet another form of double-spending attack; it is an advanced version of the Finney attack that causes the same effect [68].

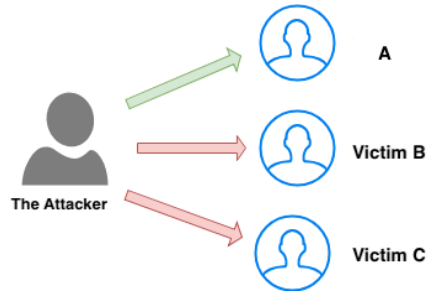


Figure 5: Double spending or race attack

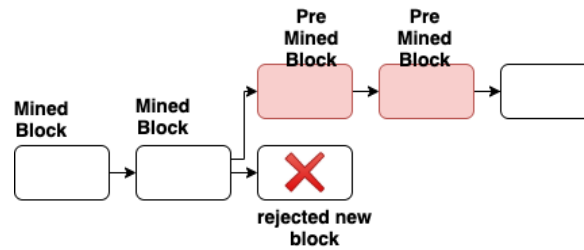


Figure 6: Premined block along with the recently mined block

In contrast to the Finney attack, which has one node for each attacker in the network, a brute-force attacker has multiple nodes, and they work in a chain of blocks parallel to the main chain. An attacker performs a double-spending transaction in a block while working on the size of his or her private blockchain at the same time. A merchant waits for the next block in the chain as a confirmation from the network's parties before accepting a transaction as usual. In the next step, the merchant sends the product to the customer. Then, the attacker updates the public chain with a larger number of blocks using contradicting confirmations [23].

The probability of a brute-force attack is low because most of the nodes in the blockchain know that they should wait for the completion of transaction validation before sending their products. The common solution for this attack is waiting for the validation process to finish completely [23] and define a punishment for the attacker, such as locking the attacker's account [58].

6.4 Vector 76 or One-Confirmation Attack

The Vector 76 attack is specially designed to attack Bitcoin exchange networks. The attacker creates the privately mined block to perform a double-spending attack on the exchanges [72].

An attacker creates a transaction with a large amount of money and saves it to a block in private. Then, the attacker waits for someone else to mine another block. The attacker sends his block to the service before the other block. The service accepts the transaction with a large amount of money, which will have one confirmation. The rest of the network simultaneously accepts the other block. After that, the attacker immediately requests a withdrawal, and the service sends a large amount of money to the attacker. The network carries out the blockchain fork. Even though the attacker's deposit will soon be invalidated, the withdrawal will be considered valid. Many parties in the network recognize the transaction as valid, as shown in Fig. 6 [14].

Those nodes that want to be safe have to wait for multiconfirmations for transactions to finish. The probability of a one-confirmation attack is low as long as nodes wait for confirmation [23].

6.5 51% Hash Power or Goldfinger Attack

In a 51% hash power or Goldfinger attack, a mining pool or organized group of miners cooperate. The group forces and shares the rewards from transactions. A mining pool with many members and great computational power can easily change the consensus. This means that the attacker has more than 50% of the computing resources of the network under the control of a single miner or a pool of miners. This enables a double-spending attack called a 51% attack.

An attacker controls more than half of the network's power, which means most of the transactions are in their control and other parties are prevented from winning competitions [89].

The attacker alters the transaction after sending it, and it appears they still had the money they just spent. However, an attacker or group of attackers can prevent the process of recording new blocks by controlling the majority of the computational power on the network. They prevent other miners from completing blocks, which allows them to dominate the mining of new blocks and earn all of the rewards.

Many solutions exist for this attack by modifying the consensus protocols. The changes will accept that ignoring the longest chain rule must be explored to mitigate the 51% attack effectively. Delayed proof of work (DPoW) is one of the policies used to protect against the double-spending problem. It applies to cryptocurrency systems that work based on an unspent transaction output (UTXO). The participants must wait for an explicit amount of time for the notarization process to be completed. The risk of attack with this security technique is low [78].

6.6 Selfish Mining

The chances of winning a reward for a mining pool depend on hash power [75]. In selfish mining, the miner receives more than the other miner's power after the attack [75, 23].

Miners are in competition to earn the reward associated with each block, and they have to follow some rules; in addition, they can behave selfishly. Selfish mining refers to malicious miner behavior, such as attacking in an unfair way.

All miners can create one block separately at a time. After creating a single block, it has to be published, and every miner works to produce a block that will follow the previous one. However, in selfish mining, the miner does not follow this normal process of making a block. In selfish mining, the dishonest miner tries to prevent a block from being broadcast to the rest of the mining pool network. Then, they continue to mine the next block. Fig. 7 illustrates the keeping of the mined block(s). The attackers do this to cause trouble between other miners and to gain more proof-of-work than other miners in the mining pool. Thus, the miner receives more than the specified amount of mining power from the attack.

According to the theory of selfish mining, a miner promotes other miners to continue working on blocks that lead to a dead end instead of adding them to the longest chain. An attacker is only giving himself a priority in mining the next block by hiding the mined block from the network. For attackers to be successful in performing and disrupting any transactions, they must be more powerful than all other network members together.

Zeroblock is a solution that uses timestamps and makes all honest miners reject a block that a selfish miner keeps private over a period of time. Moreover, a selfish mining pool cannot achieve more than its expected reward. Zeroblock has decreased the high probability of selfish mining attacks to a low probability [83].

6.7 Stalker Miner Attack

The stalker miner attack is similar to the selfish mining attack [38]. It involves a malicious miner trying to prevent a specific miner from publishing their blocks on the main chain. The selfish attackers try

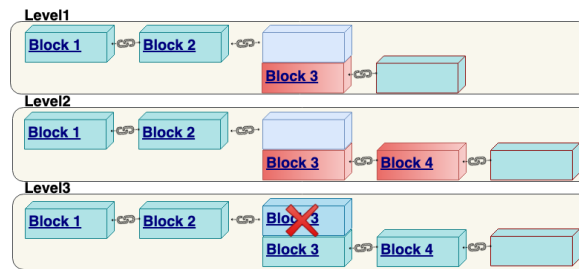


Figure 7: Selfish mining attack

to increase their relative revenue and try to dismiss the network goals, while the stalker miners do not worry about how to earn rewards. Their objective is to target honest nodes and decrease their revenue by preventing a node from being able to have its block published in the chain.

This attack only consists of a few functions involving modifying, waiting, and publishing. Then, all honest miners follow them, and they show a block immediately after mining it. All parties accept the longest chain and mine on top of it [38]. The main issue in this system is confidence because the attackers should have power and confidence. If the miners waste resources without rewards, they will leave work, and then the confidence will be low. The attacker has to spend a lot of money to be successful and to buy particular resources in big networks, such as Bitcoin and Ethereum. In these systems that use proof of work, the attacker must control the substantial network energy and spend a lot of money to become successful.

This type of attack is predictable by finding the height of forks when the blockchain suffers from selfish mining and stalker attacks. There is a simple heuristic method to detect the behavior of a malicious miner by monitoring the fork height. Fork heights greater than two are suspected to be a possible attack [20]. The probability of this attack is low.

6.8 Block Withholding (BWH) Attack

The block withholding (BWH) attack is very similar to the selfish mining attack. The BWH attack works on mining pools, and this is the main difference [79]. In the BWH attack, blocks are discarded, and the attacker tries to infiltrate another pool's miner. However, in selfish mining, blocks are only kept hidden until the right time to publish them, and other miners are treated as covered and closed pools.

A pool consists of many honest miners who work under the control of the pool manager. A standard pool uses all its honest miners for mining on its behalf. The attacker infiltrates a victim pool by registering as a regular miner while the other miners work as usual. Then, the attacker receives mining tasks from the victim pool and sends them to some of the honest miners to perform in the pool. The attacker spends the mining power on the victim's tasks, which is called the infiltration rate, and thus infiltrates the miners. When the attacker receives partial proof of work (PPoW) from the infiltrated miners, he or she sends them to the victim pool, which estimates their true mining power. When the attacker receives full proof of work (FPoW) from the infiltrating miners, he or she keeps them hidden and does not contribute to the victim's revenue; therefore, they achieve opportunities by diverting these miners. The victim pool thinks that it is doing effective mining and shares its revenue with the attacking pool. The attacker distributes the revenue from the infiltration among all the honest miners, but the attacker receives extra revenue through the infiltration of the other pool [28].

The Miners Dilemma is a game with multiple pools as opponents that is used to model this type of attack [28]. Each iteration of the game is a case of the Prisoners Dilemma. If pool A chooses to attack pool B, then pool A gains revenue while pool B loses revenue. Pool B can later retaliate by attacking

pool A to gain revenue. Therefore, attacking in each iteration can be effective; if both pool A and pool B attack each other, there will be a Nash equilibrium [59]. Both will earn less than they would have, but both of them will be safe. However, if neither of them is attacked by the other pool, then they could increase their revenue by attacking completely different pools.

Because of the similarity with the selfish mining attack, ZeroBlock is also a solution for the BWH attack that is mentioned in the selfish mining section 6.7. It can reduce the probability of intentional forks that are a result of block-withholding attacks. This solution can turn a high probability of a BWH attack risk to a low probability [83].

6.9 Fork after Withholding Attack

The fork after Withholding attack (FAW) is a combination of the BWH attack with intentional forks. The FAW attack also presents better rewards compared to those of the BWH attack.

Such an attack is expected to be seen among mining pools. It increases the reward by simultaneously attacking multiple pools. The attacker withholds FPoWs in multiple target pools simultaneously. If other honest miners are observed propagating a block outside of those pools, the attackers submit all their FPoWs immediately and create forks with multiple branches [45].

J. Ke et al. [41] define the differences as follows: "the attacker can generate a fork through the pool if he/she found and withheld an FPoW before the honest miner (neither the attacker nor someone within the victim pool) publishes a block". The attacker takes all the rewards if the main pool is comprised of the attacker only [18]. If two pools both used the FAW attack against each other, then we have an FAW attack game, where the larger pool will win. However, unlike in the BWH attack, the larger pool always earns the extra reward. Therefore, the Nash equilibrium for the FAW attack game in which two pools decide whether to attack may be Pareto optimal [45].

The probability of FAW attacks is high. The systems need an anti-withholding reward system (AWRS) to reduce the risk of this attack. An AWRS works as the pool manager to reduce adoption overhead and support backward compatibility. It performs these processes by providing greater rewards to the block submissions. It reduces the optimal attacker's behavior to honest mining and makes withholding-based threats irrelevant for rational miners [77].

6.10 The Pool-Hopping Attack

The pool-hopping attack target is the blockchain mining pool, and it focuses on creating an environment to catch more rewards in the future. A miner takes part in different mining pools. He or she takes part in the pool when the reward of the pool is high and then leaves when the reward is low. This behavior is the main difference from an FAW attack. These malicious miners can still be rewarded even after leaving. There is no reason for an honest miner to stay in a pool when the system does not perform any checks for this attack.

The honest miner will lose money and leave the blockchain mining pool, and only individual miners will remain in the network when most nodes perform pool hopping. The remaining miners will not participate in mining blockchain blocks because that requires high computational resources, and many of them cannot afford it. Finally, the remaining miners cannot guarantee the integrity of the blockchain [82, 70]. It might be more profitable for the opponent to switch to another pool or mine alone [23].

The probability of the pool-hopping attack is high. A smart contract-based pool-hopping attack prevention model is one solution to prevent pool-hopping attacks. The main objective of this solution is to maintain the proportional relationship between the miners by pushing them to contribute to mine a block with their computational power successfully. The pool manager keeps a record of each miner's behavior before joining the mining pool. This approach prevents the most frequent mine hoppers from abandoning

the mining pool by calculating the exact escrow amount for each miner and defining punishment for those who commit a pool-hopping attack [82].

6.11 Bribery Attack

Through the bribery attack, an attacker with high hash power may receive the rest of the computing resources for a short duration of time. Typically, the attacker forks a new branch and bribes the other participants to work on his/her branch instead of the other branch.

There are three ways to perform a bribery attack. The first is the out-of-band payment, where the attacker pays directly to the seller from the wallet. The second is the negative-fee mining pool, where the attacker makes a pool by paying a higher share of the reward. The third is in-band payment by forking. In this case, the attacker attempts to bribe through the blockchain itself by creating a fork containing bribe money, which is freely available to any miner adopting the fork [16]. This means that with power and money, attackers can perform more attacks. The miners who took the bribes will get benefits for a short time, but these short-term benefits might be labeled as losses under another type of attack in the long term due to double-spending, Sybil, and Goldfinger attacks or through an exchange rate crash.

The probability of this type of attack is high. There are some strategies for making it difficult to succeed. The simple strategy is extra confirmation for large transactions. The attackers must spend a lot more money to be successful. The attacker's bribery costs increase linearly in this model [16].

6.12 Eclipse Attack

The eclipse attacker concentrates on a specific node's connection rather than attack the whole network. This is a particular type of attack that is designed for a distributed and decentralized network [35]. It is impossible for a node to connect to all the other nodes on the decentralized network, such as a peer-to-peer (P2P) network. A node usually connects to a selection in the network. For example, a node has at least eight outgoing connections. The eclipse attack proposes to cut nodes and miners' power to manage all the connections.

Such attacks disrupt the normal routing of the network and end up being forwarded towards the wrong side in the network. It is known as network hijacking. It works by rerouting Internet protocol (IP) addresses without permission from the nodes. IP hijacking can occur intentionally or by accident in several ways. It can be accomplished by a malicious autonomous system (AS) [29]. An autonomous system is a network or group of networks managed by an organization, and they are under a single external routing policy. The malicious AS performs a man-in-the-middle attack, and its purpose is to isolate the other nodes. It controls all the sent and received messages to the victim node. Additionally, it uses corrupted or newly added nodes to communicate with them, as shown in Fig. 8.

In summary, a malicious node tries to capture all of the node's connections. Generally, an attacker can control the host nodes' connections with each IP address of the neighboring nodes of an intended victim by trial and error. The next time, the victim node loses the connections and then rejoins the network by resetting their connections and finds a new group of nodes to connect with. The attacker has a good chance to be the administrator of all the victim's connections. The cost of destroying an eclipse attack is high when each peer is always communicating with other peers, as the hacker needs to control the entire network to hack the P2P system.

The probability of this attack is high. There are several policies to help systems prevent this kind of attack, such as increasing the number of outgoing connections for nodes in the network and random node selection [35].

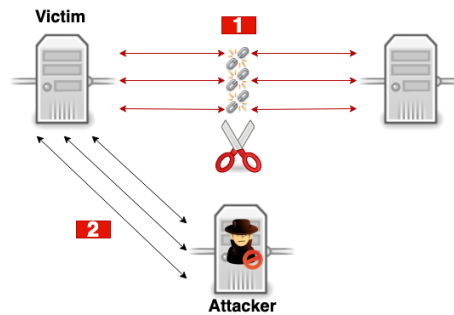


Figure 8: The Eclipse attack

6.13 Balance Attack

The balance attack goal is to destroy the proof-of-work (PoW) systems by delaying network communication. It performs a delay between groups of nodes that have a balanced level of mining power [23]. It is specially designed for a public blockchain. Initially, the attacker starts to cause some destructive effect on a transaction, such as introducing a delay between groups of miners in the same network with similar mining power. This attack can be a double-spend on PoW consensus. An attacker can use its limited hashing power to delay messages over the blockchain with a small amount of the total hash power.

In summary, after the attacker begins to cause delays between subgroups with similar mining power, he or she publishes transactions in one subgroup. Later, he or she mines many blocks in another subgroup to outweigh the transaction subgroup rather than the subtree of another subgroup. Although the transactions are committed, the attacker can rewrite the blocks with the outweighing transactions that the subtree contains this transaction [63, 78].

SmartPool is a novel mining pool system implemented in the form of a smart contract. The miner returns the completed shares to the SmartPool client. When the number of completed shares reaches a specific amount, they will be sent to the SmartPool contract. The SmartPool contract will verify the shares and deliver rewards to the client. SmartPool is more efficient, and it can prevent the attacker from resubmitting shares in different batches. Moreover, SmartPool can guarantee that honest miners will gain the expected rewards, even if there are malicious miners in the pool [48]. The probability of this attack is low.

6.14 Liveness Attack

The liveness attack is a type of attack that introduces delay, such as the transaction confirmation time of the target transaction. The liveness attack works in three steps: 1) preparation, 2) transaction denial, and 3) blockchain retardation [48, 33].

The preparation step is similar to a selfish mining attack. An attacker builds a group of honest miners before the transaction is broadcast to all nodes of the network. The attacker makes the chain private, and the length of the chain is longer than the other chains. In the transaction's denial step, the attacker anonymously keeps or holds the block that contains the target transaction, thus preventing it from joining the public chain. The final step is the blockchain retarder. In the process of building the chain, the attacker will publish the private chain to the public chain to slow down the growth rate of the public chain. In some blockchain systems, when the depth of the block that includes chains of transactions is longer than normal, chains of transactions will be considered valid [33]. Accordingly, the attacker will continue building a private chain to create an influence attack over the public chain. Later, the attacker will publish its hidden blocks into the public chain at the same time to slow down the growth rate of the public chain. The liveness attack will end when chains of transactions are verified as valid in the public chain.

There is no well-known solution for this type of attack, but SmartPool can be a solution [48, 33]. The probability of a liveness attack is not high.

6.15 Sybil Attack

The Sybil attack is known as a security threat on the network where a node tries to attack by creating multiple accounts, nodes, or computers in the network [27]. The attacker creates as many identities as possible and behaves as multiple peers in the network. This attacker tries to increase their number of peers to increase their voting power. A Sybil attacker tries to find identities by finding the routers that are connected to victim nodes. As an example, in the network, they perform the flood fill algorithm to routers for finding nodes. Flood fill is an algorithm that "determines the area connected to a given node in a multidimensional array" [24]. One node is enough to destroy a resource, and this attack is easier to execute with the newly created network. Additionally, it becomes more challenging to complete the attack with a large network size.

There is no way to detect Sybil attacks. The Sybil attacker can perform the attack in a blockchain network when somebody runs multiple nodes in the network. The attacker can create fake identities to change or turn voting results to his/her favor while others are busy with the blocking process. The Sybil attack can be useful as a special component used in other attacks, such as an eclipse attack, in the blockchain. It can be used when the attacker controls enough numbers of identities.

TrustChain is a solution that creates trusted transactions among strangers without central control. TrustChain is a permissionless tamper-proof data structure and is an immutable chain for storing transaction records [67]. The probability of a Sybil attack in the blockchain is low because of costly resources for the attacker to participate in proof of work or proof of stake. The blockchain network is distributed and uses different consensus algorithms to help defend itself against attacks.

6.16 BGP Hijacking Attack

A BGP hijack attack is a routing attack in which an Internet service provider (ISP) redirects Internet traffic by distributing fake BGP messages in the Internet routing system towards incorrect addresses. These attacks already affect the Bitcoin network today [8]. An attacker can use routing attacks to divide the network into two or more groups.

In one scenario, the nodes are divided into a left and a right group. By modifying the BGP routing, the connections are cut between the two groups. The nodes inside each group maintain their communication with the nodes inside their related group. In the next step, all the Bitcoin traffic between the groups is intercepted by the attacker. In another scenario, the attacker usually changes the parameters or information of the communication by introducing a delay and making nodes entirely hidden [8].

The victim node is unaware of the numerous newly mined blocks and the same transactions during this delay. The attacker builds up the new parallel blockchain by preventing nodes inside a group from communicating with nodes outside of it.

The impact of such an attack will be severe in two ways. Merchants will suffer double-spending attacks. Miners will suffer from wasting their energy and/or will be unable to contribute to the network. After the attack is over, blocks mined in a group will be discarded, along with all the included transactions and the miners' revenue. Anti-prefix filtering is a common solution for this type of attack. It has several methods to detect and prevent BGP attacks. For example, encrypted Bitcoin communication and/or Adopt MAC and Request a Block on Multiple Connections are two of them [8]. The probability of a BGP hijacking attack is low because only ISPs that run BGP routing can perform such an attack.

Table 3: Blockchain attacks and solutions to protect systems

Attacks	Blockchain			
	Permissioned		Permissionless	
	Probability	Solution	Probability	Solution
Race	Low	Validation Process	Low	Validation Process [23]
Stalker Miner	Low	Decreasing Delay Time	Low	Height of Forks [20]
BGP Hijacking	Low	Anti-Prefix Filtering	Low	Anti-Prefix Filtering [8]
Sybil	Low	Trust Chain	Low	TrustChain [67]
Liveness	Low	SmartPool	Medium	Decreasing Delay Time & SmartPool [33, 48]
Balance	Low	Design policies	Low	SmartPool [48]
Eclipse	High	Increase Number of Connections Random Node Selection	High	Increase Number of Connections Random Node Selection [35]
Bribery	High	Extra Confirmation Based Preventing	High	Extra Confirmation Based Preventing [16]
Pool Hopping	Low	Hopping Prevention Algorithm	Medium	Smart Contract Based Preventing [82]
Fork After Withholding	Low	Controlling Reward Rate	High	Anti-Withholding Reward System [77]
Block withholding	Low	Controlling Hash Rate	High	ZeroBlock [83]
Selfish Mining	Low	Controlling Hash Rate	High	ZeroBlock [83]
51% Hash Power	High	Controlling Higher Hash Rate	High	Delayed Proof of Work [78]
Vector 76	Low	Wait for Multiconfirmations For Transactions	Low	Wait for Multiconfirmations For Transactions [23]
Brute Force	Low	Improve Identification System	Low	Locking Accounts & Wait for Confirmations [58]
Finney	Low	Defines limitation	Low	Decreasing Block Generation Time Wait For Multiconfirmations Transactions [23]

6.17 Summary of Attacks and Solutions

We explained the most common attacks in blockchain, and we showed how serious they are. We illustrated them in Table 3 by dividing them among permissionless and permissioned blockchains. Then, we explained each attack's probability of occurring in three levels (low, medium, high) and referred to the most common solutions for attack prevention. Therefore, we categorized the attacks based on their effect level.

Most attacks in the permissionless blockchains' reasons are due to a lack of privacy, designed consensus algorithms that are not good enough, and having no centralized party in the network to manage and control all the activities. These issues motivate attackers to attack this type of blockchain.

Most solutions in the permissioned blockchain in Table 3 are not the best solutions; they are just suggested policies. The solutions depend on central management decisions, but they can practice permissionless blockchain solutions instead.

A successful attack is hard but possible to instigate in a blockchain. The immutable history prevents

them from performing their attack efficiently in both public and private blockchains. In the permissioned system, we can easily track back and check the failure because there is a central manager who knows all the parties in the network and controls them. The risk of attack to the central manager is the primary security issue in these systems. The probability of attacks in the blockchain is lower than that in other types of networks. It is difficult to compare permissioned and permissionless attacks. They have specific attacks because of different schemes. Today, the newly designed architectures with new policies keep the system more secure against most of these attacks.

7 Research Challenges

IoT systems typically generate huge amounts of data. Security and transaction ordering are important issues in such systems. These issues can in many cases be resolved by using the blockchain, but we have identified some remaining research challenges related to security and privacy. Moreover, manageability and traceability are significant problems in IoT. Accordingly, finding the best method of blockchain-based consensus algorithms is considered as one of the challenges. Also, since blockchain faces issues around scalability and storage of huge amounts of data, architectures that use blockchain in concert with scalable storage technologies to support large scale IoT systems should be researched further.

8 Conclusion

It is crucial to design and produce a secure blockchain-based IoT system that satisfies the digital world's expected requirements. The IoT system's future should be cooperative with present IoT technologies so that the transformation from traditional architecture to a distributed and decentralized system is economically feasible. Furthermore, performance features such as privacy should also be given attention, in parallel to security problems.

In this article, we presented a survey of networks and service architectures and how it is possible to use blockchain with them. Then, the properties of the blockchain were discussed. We started the article by explaining how the blockchain works, and we covered its essential internal functions. Then, we illustrated the meaning of privacy and security problems in systems. Furthermore, we discussed how the blockchain solves security problems by comparing the types of blockchains. We demonstrated three types of blockchains that are specially designed for purposes such as industry, business, social applications, and other networks. We defined consensus algorithms that are used explicitly for each of them, such as permissionless and permissioned blockchains. We explained the IoT and how smart devices communicate and connect with and without third parties. Next, this article illustrated a variety of architectures in IoT systems and showed the blockchain encounter with the IoT and the resulting behavior.

Finally, we explained most of the frequent types of attacks on blockchains as a primary challenge in IoT related to using blockchain in the systems. Although blockchain is a secure method and there are many attack-prevention methods available, the risk of attacks remains. This means that one must have a highly secure system to detect most types of attacks. It is impossible to know all of the possible types of attacks and find a solution to them at their initial step. Nevertheless, we can protect our systems with highly secure methods against most of the current attacks.

Acknowledgement

This work has been supported by the Kolarctic CBC under grant KO4096 and VINNOVA under grant 2019-02836.

References

- [1] M. Aazam and E.-N. Huh. Fog computing and smart gateway based communication for cloud of Things. In *Proc. of the 2014 International Conference on Future Internet of Things and Cloud (FiCloud'14), Barcelona, Spain*, pages 464–470. IEEE, August 2014.
- [2] J. Al-Jaroodi and N. Mohamed. Blockchain in industries: A survey. *IEEE Access*, 7:36500–36515, March 2019.
- [3] S. A. Al-Qaseemi, H. A. Almulhim, M. F. Almulhim, and S. R. Chaudhry. IoT architecture challenges and issues: Lack of standardization. In *Proc. of the 2016 Future Technologies Conference (FTC'16), San Francisco, California, USA*, pages 731–738. IEEE, December 2016.
- [4] M. S. Ali, K. Dolui, and F. Antonelli. IoT data privacy via blockchains and IPFS. In *Proc. of the 7th International Conference on the Internet of Things (IoT'17), Linz, Austria*, pages 1–7. ACM, October 2017.
- [5] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani. Applications of blockchains in the Internet of Things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 21(2):1676–1717, December 2018.
- [6] A. Alkhalil and R. A. Ramadan. IoT data provenance implementation challenges. *Procedia Computer Science*, 109:1134–1139, June 2017.
- [7] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng. Fog computing for the Internet of Things: Security and privacy issues. *IEEE Internet Computing*, 21(2):34–42, March 2017.
- [8] M. Apostolaki, A. Zohar, and L. Vanbever. Hijacking bitcoin: Routing attacks on cryptocurrencies. In *Proc. of the 2017 IEEE Symposium on Security and Privacy (SP'17), San Jose, California, USA*, pages 375–392. IEEE, May 2017.
- [9] R. Arora and A. Parashar. Secure user data in cloud computing using encryption algorithms. *International Journal of Engineering Research and Applications*, 3(4):1922–1926, July-August 2013.
- [10] Y. N. Aung and T. Tantidham. Review of Ethereum: Smart home case study. In *Proc. of the second International Conference on Information Technology (INCIT'17), Nakhonpathom, Thailand*, pages 1–4. IEEE, November 2017.
- [11] A. Banafa. IoT and blockchain convergence: benefits and challenges. <https://iot.ieee.org/newsletter/january-2017/iot-and-blockchain-convergence-benefits-and-challenges.html> [Online; accessed on August 20, 2020], January 2017. IEEE Internet of Things.
- [12] M. Banerjee, J. Lee, and K.-K. R. Choo. A blockchain future for Internet of Things security: A position paper. *Digital Communications and Networks*, 4(3):149–160, August 2018.
- [13] R. Bhagwan, S. Savage, and G. M. Voelker. Understanding availability. In *Proc. of the second International Workshop on Peer-to-Peer Systems (IPTPS'03), Berkeley, California, USA*, volume 2735 of *Lecture Notes in Computer Science*, pages 256–267. Springer, February 2003.
- [14] A. Biryukov and I. Pustogarov. Bitcoin over Tor isn't a good idea. In *Proc. of the 2015 IEEE Symposium on Security and Privacy (SP'15), San Jose, California, USA*, pages 122–134, May 2015.
- [15] A. B. Bondi. Characteristics of scalability and their impact on performance. In *Proc. of the second International workshop on Software and Performance (WOSP'00), Ottawa, Ontario, Canada*, pages 195–203. ACM, September 2000.
- [16] J. Bonneau. Why buy when you can rent? Bribery attacks on bitcoin-style consensus. In *Proc. of the 2016 International Conference on Financial Cryptography and Data Security (FC'2016), Christ Church, Barbados*, volume 9604 of *Lecture Notes in Computer Science*, pages 19–26. Springer, February 2016.
- [17] B. Cao, Z. Zhang, D. Feng, S. Zhang, L. Zhang, M. Peng, and Y. Li. Performance analysis and comparison of PoW, PoS and DAG based blockchains. *Digital Communications and Networks*, January 2020.

- [18] S.-Y. Chang and Y. Park. Silent timestamping for blockchain mining pool security. In *Proc. of the 2019 International Conference on Computing, Networking and Communications (ICNC'19), Honolulu, Hawaii, USA*, pages 1–5. IEEE, February 2019.
- [19] M. Chen, S. Mao, and Y. Liu. Big data: A survey. *Mobile Networks and Applications*, 19(2):171–209, April 2014.
- [20] V. Chicarino, C. Albuquerque, E. Jesus, and A. Rocha. On the detection of selfish mining and stalker attacks in blockchain networks. *Annals of Telecommunications*, 75:143–152, February 2020.
- [21] J. Choi, Y. In, C. Park, S. Seok, H. Seo, and H. Kim. Secure IoT framework and 2D architecture for End-To-End security. *The Journal of Supercomputing*, 74(8):3521–3535, August 2018.
- [22] K. Christidis and M. Devetsikiotis. Blockchains and smart contracts for the Internet of Things. *IEEE Access*, 4:2292–2303, May 2016.
- [23] M. Conti, E. S. Kumar, C. Lal, and S. Ruj. A survey on security and privacy issues of bitcoin. *IEEE Communications Surveys & Tutorials*, 20(4):3416–3452, May 2018.
- [24] G. Danezis, C. Lesniewski-Laas, M. F. Kaashoek, and R. Anderson. Sybil-resistant DHT routing. In *Proc. of the 10th European Symposium on Research in Computer Security (ESORICS'05), Milan, Italy*, volume 3679 of *Lecture Notes in Computer Science*, pages 305–318. Springer, September 2005.
- [25] P. Danzi, A. E. Kalor, C. Stefanovic, and P. Popovski. Analysis of the communication traffic for blockchain synchronization of IoT devices. In *Proc. of the 2018 IEEE International Conference on Communications (ICC'18), Kansas City, Missouri, USA*, pages 1–7. IEEE, May 2018.
- [26] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak. Blockchain: A distributed solution to automotive security and privacy. *IEEE Communications Magazine*, 55(12):119–125, December 2017.
- [27] J. R. Douceur. The Sybil Attack. In *Proc. of the first International Workshop on Peer-to-Peer Systems (IPTPS'02), Cambridge, Massachusetts, USA*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer, March 2002.
- [28] I. Eyal. The miner's dilemma. In *Proc. of the 2015 IEEE Symposium on Security and Privacy (SP'15), San Jose, California, USA*, pages 89–103. IEEE, May 2015.
- [29] S. Feld, M. Schönfeld, and M. Werner. Analyzing the deployment of bitcoin's P2P network under an AS-level perspective. *Procedia Computer Science*, 32:1121–1126, December 2014.
- [30] K. Garimella. Decentralized vs Distributed systems Part I. <https://medium.com/@koreprotocol/decentralized-vs-distributed-systems-part-i-1a16891b28a1> [Online; accessed on August 15, 2020], 2019.
- [31] M. Geller and P. Nair. 5G security innovation with Cisco. *Whitepaper Cisco Public*, pages 1–29, May 2018.
- [32] J. Göbel, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor. Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay. *Performance Evaluation*, 104:23–41, October 2016.
- [33] N. Gupta. Security and privacy issues of blockchain technology. In *Advanced Applications of Blockchain Technology*, pages 207–226. Springer, February 2020.
- [34] S. Hameed and S. Farooq. The art of crypto currencies: A comprehensive analysis of popular crypto currencies. *International Journal of Advanced Computer Science and Applications*, 7(12), November 2016.
- [35] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg. Eclipse attacks on bitcoin's peer-to-peer network. In *Proc. of the 24th USENIX Security Symposium (USENIX Security'15), Washington, D.C., USA*, pages 129–144. USENIX, August 2015.
- [36] M. D. Hill. What is scalability? *ACM SIGARCH Computer Architecture News*, 18(4):18–21, December 1990.
- [37] M. Ingham, J. Marchang, and D. Bhowmik. IoT security vulnerabilities and predictive signal jamming attack analysis in lorawan. *IET Information Security*, 14(4), June 2020.
- [38] E. F. Jesus, V. R. Chicarino, C. V. de Albuquerque, and A. A. d. A. Rocha. A survey of how to use blockchain to secure Internet of Things and the stalker attack. *Security and Communication Networks*, 2018, April 2018.
- [39] W. Jia and W. Zhou. *Distributed Network Systems: From Concepts to Implementations*, volume 15. Springer Science & Business Media, December 2004.
- [40] A. P. Joshi, M. Han, and Y. Wang. A survey on security and privacy issues of blockchain technology.

- Mathematical Foundations of Computing*, 1(2):121–147, May 2018.
- [41] J. Ke, H. Jiang, X. Song, S. Zhao, H. Wang, and Q. Xu. Analysis on the block reward of fork after withholding (FAW). In *Proc. of the 12th International Conference on Network and System Security (NSS'18)*, Hong Kong, China, volume 11058 of *Lecture Notes in Computer Science*, pages 16–31. Springer, August 2018.
- [42] S. Kim and I. Lee. IoT device security based on proxy re-encryption. *Journal of Ambient Intelligence and Humanized Computing*, 9(4):1267–1273, August 2018.
- [43] E. Ko, T. Kim, and H. Kim. Management platform of threats information in IoT environment. *Journal of Ambient Intelligence and Humanized Computing*, 9(4):1167–1176, August 2018.
- [44] S. Kozlovski. A thorough introduction to distributed systems, 2018.
- [45] Y. Kwon, D. Kim, Y. Son, E. Vasserman, and Y. Kim. Be selfish and avoid dilemmas: Fork after withholding (FAW) attacks on bitcoin. In *Proc. of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS'17)*, Dallas, Texas, USA, pages 195–209. ACM, October 2017.
- [46] V. L. Lemieux. Blockchain and distributed ledgers as trusted recordkeeping systems: An archival theoretic evaluation framework. In *Proc. of the 2017 Future Technologies Conference (FTC'17)*, Vancouver, British Columbia, Canada, volume 2017, pages 1–11, November 2017.
- [47] W. Li, A. Sforzin, S. Fedorov, and G. O. Karame. Towards scalable and private industrial blockchains. In *Proc. of the 2017 ACM Workshop on Blockchain, Cryptocurrencies and Contracts (BCC'17)*, Abu Dhabi, UAE, pages 9–14. ACM, April 2017.
- [48] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen. A survey on the security of blockchain systems. *Future Generation Computer Systems*, 107, June 2017.
- [49] C.-F. Liao, S.-W. Bao, C.-J. Cheng, and K. Chen. On design issues and architectural styles for blockchain-driven IoT services. In *Proc. of the 2017 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW'17)*, Taipei, Taiwan, pages 351–352. IEEE, June 2017.
- [50] I.-C. Lin and T.-C. Liao. A survey of blockchain security issues and challenges. *International Journal of Network Security*, 19(5):653–659, September 2017.
- [51] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao. A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet of Things Journal*, 4(5):1125–1142, October 2017.
- [52] H. Liu, C. Li, X. Jin, J. Li, Y. Zhang, and D. Gu. Smart solution, poor protection: An empirical study of security and privacy issues in developing and deploying smart home devices. In *Proc. of the 2017 Workshop on Internet of Things Security and Privacy (IoTS&P'17)*, Dallas, Texas, USA, pages 13–18. ACM, November 2017.
- [53] B. Lu, L. Wang, J. Liu, W. Zhou, L. Guo, M.-H. Jeong, S. Wang, and G. Han. LaSa: Location aware wireless security access control for IoT systems. *Mobile Networks and Applications*, 24(3):748–760, June 2019.
- [54] Y. Lu and L. Da Xu. Internet of Things (IoT) cybersecurity research: A review of current research topics. *IEEE Internet of Things Journal*, 6(2):2103–2115, September 2018.
- [55] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys & Tutorials*, 7(2):72–93, April 2005.
- [56] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni. Blockchain’s adoption in IoT: The challenges, and a way forward. *Journal of Network and Computer Applications*, 125:251–279, January 2019.
- [57] J. Maktoubian and K. Ansari. An IoT architecture for preventive maintenance of medical devices in healthcare organizations. *Health and Technology*, 9(3):233–243, May 2019.
- [58] N. Manthey and J. Heusser. SATcoin–Bitcoin mining via SAT. *Proceedings of SAT Competition 2018: Solver and Benchmark Descriptions*, B-2018-1:67, July 2018.
- [59] E. Maskin. *The theory of implementation in Nash equilibrium: A survey*. Cambridge, Mass.: Dept. of Economics, Massachusetts Institute of Technology, October 1983.
- [60] D. Minoli, K. Sohraby, and J. Kouns. IoT security (IoTSec) considerations, requirements, and architectures. In *Proc. of the 14th IEEE Annual Consumer Communications & Networking Conference (CCNC'17)*, Las Vegas, Nevada, USA, pages 1006–1007. IEEE, January 2017.
- [61] A. A. Monrat, O. Schelén, and K. Andersson. A survey of blockchain from the perspectives of applications,

- challenges, and opportunities. *IEEE Access*, 7:117134–117151, August 2019.
- [62] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. <http://bitcoin.org/bitcoin.pdf> [Online; accessed on August 15, 2020], October 2008.
- [63] C. Natoli and V. Gramoli. The balance attack or why forkable blockchains are ill-suited for consortium. In *Proc. of the 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'17), Denver, Colorado, USA*, pages 579–590. IEEE, June 2017.
- [64] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck. Blockchain. *Business & Information Systems Engineering*, 59:183–187, March 2017.
- [65] A. Onasanya, S. Lakkis, and M. Elshakankiri. Implementing IoT/WSN based smart Saskatchewan healthcare system. *Wireless Networks*, 25:3999–4020, January 2019.
- [66] P. Oppenheimer. *Top-down network design*. Pearson Education India, December 1999.
- [67] P. Otte, M. de Vos, and J. Pouwelse. TrustChain: A Sybil-resistant scalable blockchain. *Future Generation Computer Systems*, 107:770–780, June 2020.
- [68] P. S. Pannu and R. Mathew. Review on security problems of bitcoin. In *Proc. of the 2018 International Conference on Computer Networks, Big data and IoT (ICCBT'18), Madurai, India*, pages 180–184. Springer, December 2018.
- [69] S. Pape and K. Rannenber. Applying privacy patterns to the Internet of Things' (IoT) architecture. *Mobile Networks and Applications*, 24:925–933, October 2019.
- [70] J. H. Park and J. H. Park. Blockchain security in cloud computing: Use cases, challenges, and solutions. *Symmetry*, 9(8):164, August 2017.
- [71] S. Patranabis, D. B. Roy, A. Chakraborty, N. Nagar, A. Singh, D. Mukhopadhyay, and S. Ghosh. Lightweight design-for-security strategies for combined countermeasures against side channel and fault analysis in IoT applications. *Journal of Hardware and Systems Security*, 3:103–131, September 2018.
- [72] N. Rathod and D. Motwani. Security threats on blockchain and its countermeasures. *International Research Journal of Engineering and Technology*, 5(11):1636–1642, November 2018.
- [73] P. P. Ray. A survey on Internet of Things architectures. *Journal of King Saud University-Computer and Information Sciences*, 30(3):291–319, July 2018.
- [74] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz. On blockchain and its integration with IoT. challenges and opportunities. *Future Generation Computer Systems*, 88:173–190, November 2018.
- [75] M. Saad, L. Njilla, C. Kamhoua, and A. Mohaisen. Countering selfish mining in blockchains. In *Proc. of the 2019 International Conference on Computing, Networking and Communications (ICNC'19), Honolulu, Hawaii, USA*, pages 360–364. IEEE, February 2019.
- [76] M. Samaniego and R. Deters. Hosting virtual IoT resources on edge-hosts with blockchain. In *Proc. of the 2016 IEEE International Conference on Computer and Information Technology (CIT'16), Nadi, Fiji*, pages 116–119. IEEE, December 2016.
- [77] A. Sarker, S. Wuthier, and S.-Y. Chang. Anti-withholding reward system to secure blockchain mining pools. In *Proc. of the 2019 Crypto Valley Conference on Blockchain Technology (CVCBT'19), Rotkreuz, Switzerland*, pages 43–46. IEEE, June 2019.
- [78] S. Sayeed and H. Marco-Gisbert. Assessing blockchain consensus and security mechanisms against the 51% attack. *Applied Sciences*, 9(9):1788:1–17, April 2019.
- [79] S. Shalini and H. Santhi. A survey on various attacks in bitcoin and cryptocurrency. In *Proc. of the 2019 International Conference on Communication and Signal Processing (ICCSP'19), Chennai, India*, pages 220–224. IEEE, April 2019.
- [80] Z. Shen, L. Li, F. Yan, and X. Wu. Cloud computing system based on trusted computing platform. In *Proc. of the 2010 International Conference on Intelligent Computation Technology and Automation (ICICTA'10), Changsha, China*, pages 942–945. IEEE, May 2010.
- [81] B. N. Silva, M. Khan, and K. Han. Internet of Things: A comprehensive review of enabling technologies, architecture, and challenges. *IETE Technical Review*, 35(2):205–220, February 2018.
- [82] S. K. Singh, M. M. Salim, M. Cho, J. Cha, Y. Pan, and J. H. Park. Smart contract-based pool hopping attack prevention for blockchain networks. *Symmetry*, 11(7):941:1–19, July 2019.

- [83] S. Solat and M. Potop-Butucaru. Brief announcement: Zeroblock: Timestamp-free prevention of block-withholding attack in bitcoin. In *Proc. of the 19th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'17), Boston, Massachusetts, USA*, volume 10616 of *Lecture Notes in Computer Science*, pages 356–360. Springer, November 2017.
- [84] M. Talal, A. Zaidan, B. Zaidan, A. Albahri, A. Alamoodi, O. Albahri, M. Alsalem, C. Lim, K. L. Tan, W. Shir, et al. Smart home-based IoT for real-time and secure remote health monitoring of triage and priority system using body sensors: Multi-driven systematic review. *Journal of Medical Systems*, 43(42), January 2019.
- [85] F. Tschorsch and B. Scheuermann. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys & Tutorials*, 18(3):2084–2123, March 2016.
- [86] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen. A survey on learning to hash. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):769–790, May 2017.
- [87] P. Wang, F. Ye, and X. Chen. A smart home gateway platform for data collection and awareness. *IEEE Communications Magazine*, 56(9):87–93, September 2018.
- [88] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim. A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE Access*, 7:22328–22370, January 2019.
- [89] X. Wang, X. Zha, W. Ni, R. P. Liu, Y. J. Guo, X. Niu, and K. Zheng. Survey on blockchain for Internet of Things. *Computer Communications*, 136:10–29, February 2019.
- [90] Q. Xu, K. M. M. Aung, Y. Zhu, and K. L. Yong. A blockchain-based storage system for data analytics in the Internet of Things. In *New Advances in the Internet of Things*, pages 119–138. Springer, June 2018.
- [91] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, 14(4):352–375, October 2018.
-

Author Biography



Morteza Alizadeh has a master’s degree in computer science (artificial intelligence) from Qazvin Islamic Azad University, Iran (2013). Morteza is currently continuing his education as a Ph.D. candidate, and he is researching blockchain and IoT systems in the Pervasive and Mobile Computing group of Luleå University of Technology, Luleå, Sweden (2018). His interests focus on blockchain, smart contracts, machine learning, multiagent systems, and optimization algorithms.



Karl Andersson (Senior Member of IEEE) has an M.Sc. degree in computer science and technology from the Royal Institute of Technology, Stockholm, Sweden and a Ph.D. degree in mobile systems from at Luleå University of Technology, Sweden. After being a postdoctoral research fellow at the Internet Real-time Laboratory at Columbia University, New York, USA and a JSPS fellow with the National Institute of Information and Communications Technology, Tokyo, Japan, he is now an associate professor of pervasive and mobile computing at Luleå University of Technology, Sweden. His research interests include mobile computing, the Internet of Things, cloud technologies, and information security.



Olov Schelen (Member of IEEE) is a full professor at Luleå University of Technology and CEO at Xarepo AB. His research interests include mobile and distributed systems, software orchestration, computer networking, artificial intelligence and blockchain. He has a PhD in computer networking from Luleå University of Technology, and he also has more than 20 years of experience in industry and academia.