

# Blockchain in the Internet of Things: Architectures and Implementation

Oscar Delgado-Mohatar, Ruben Tolosana, Julian Fierrez and Aythami Morales  
School of Engineering, Universidad Autonoma de Madrid, Spain

{oscar.delgado, ruben.tolosana, julian.fierrez, aythami.morales}@uam.es

**Abstract**—The world is becoming more interconnected every day. With the high technological evolution and the increasing deployment of it in our society, scenarios based on the Internet of Things (IoT) can be considered a reality nowadays. However, and before some predictions become true (75 billion devices are expected to be interconnected in the next few years), many efforts must be carried out in terms of scalability and security.

In this study we propose and evaluate a new approach based on the incorporation of Blockchain into current IoT scenarios. The main contributions of this study are as follows: *i)* an in-depth analysis of the different possibilities for the integration of Blockchain into IoT scenarios, focusing on the limited processing capabilities and storage space of most IoT devices, and the economic cost and performance of current Blockchain technologies; *ii)* a new method based on a novel module named BIoT Gateway that allows both unidirectional and bidirectional communications with IoT devices on real scenarios, allowing to exchange any kind of data; and *iii)* the proposed method has been fully implemented and validated on two different real-life IoT scenarios, extracting very interesting findings in terms of economic cost and execution time. The source code of our implementation is publicly available in the Ethereum testnet.

**Keywords**—Blockchain, Internet of Things, Security, Privacy

## I. INTRODUCTION

The Internet of Things (IoT) has the potential to change the world, just as the Internet did [1]. This term is referred to the set of objects, sensors, and everyday items that are equipped with computing capability and network connectivity to send/receive data through the Internet [2]. As a result, IoT devices can generate and manage an autonomous ecosystem without any human intervention or supervision.

Scenarios based on the IoT can already be considered a reality nowadays, for example: *i)* smart homes where the electric light, heating, and kitchen equipment such as the fridge or washing machine are automatically operating, reporting continuously to the User/Client, and *ii)* autonomous electric vehicles searching for a charging station so that as soon as the car is running out of battery, it automatically drives to the cheapest or nearest point, and starts the charging process. Once completed, the car conducts the payment [3]. These are just some of the many applications of the IoT. With the increasing evolution and deployment of the technology in our lives, it is estimated that between 50 and 75 billion devices will be interconnected by 2025 [4], [5].

Undoubtedly, and before this comes true, a lot of efforts must be carried out in order to manage such volume of

information in a scalable and secure way. Many recent studies have focused on the IoT security [2]. Additionally, key aspects of these low-cost IoT devices such as the limited processing capability and storage space must be further studied [6]. Also, most IoT devices do not usually include protection against physical attacks, so they can be compromised easily. This is exacerbated by the fact that IoT devices almost never have maintenance/upgrade capabilities to reduce production costs.

In order to mitigate these problems, different approaches have been proposed in the literature such as lightweight cryptography [7], reinforcement of the perimeter security through the use of firewalls [8], and zero-trust approaches [9]. Furthermore, recent studies have put their eyes on other technologies such as Blockchain to overcome some of the limitations existing in IoT scenarios [10].

Blockchain is essentially a decentralized public ledger of all data and transactions that have ever been executed in the system [11]. These transactions are recorded in blocks that are created and added to the Blockchain in a linear, chronological order (immutable). Each node has the task of relaying transactions, and has a copy of the Blockchain. Other nodes, called miners, are also in charge of validating transactions, performing an expensive computational process, for which they are economically rewarded.

Blockchain was originally created and applied as an auxiliary technology of Bitcoin [12], providing a secure record of the economic transactions between users of the system. Nevertheless, a Blockchain could store any kind of digital information, providing its certification and guaranteeing its authenticity and integrity. As a result, from its origin up to now, Blockchain has been deployed in many different scenarios such as: biometrics, certification of documentation, mortgages, securities and any other official documents, assets and intelligent objects that can make decisions based on the information stored in the Blockchain, distributed markets without central authority, deposit and custody services that can resolve disputes between customers and merchants, savings accounts, voting systems, and improvements in the distribution chain for all kind of products [11], [13]–[15].

However, and despite these opportunities, the current Blockchain technology suffers from some potential limitations that must be carefully studied and characterized before the adequate integration of Blockchain into IoT scenarios.

In this study we propose and evaluate a new method for

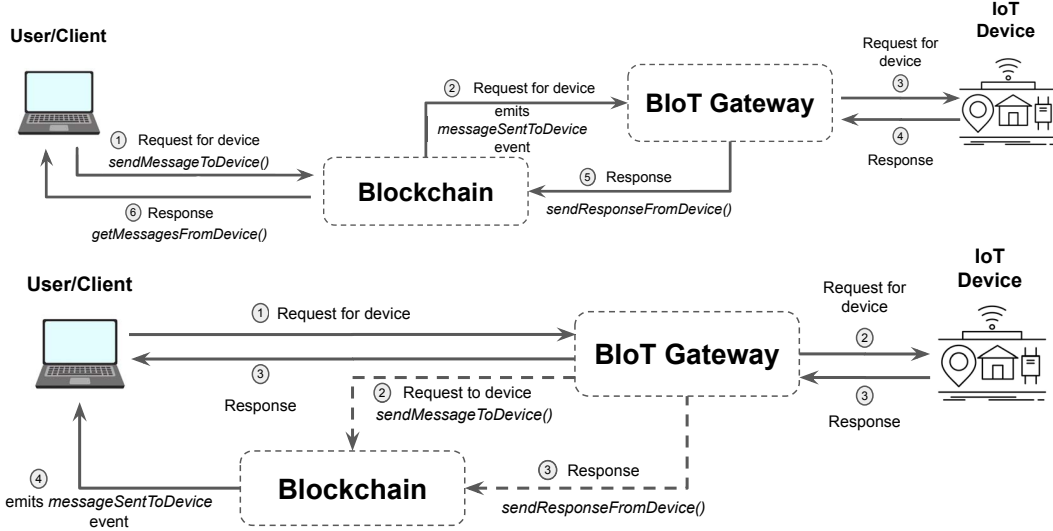


Figure 1: Description of our two proposed architectures to incorporate Blockchain into IoT scenarios. Two different configurations are evaluated regarding which module is reached first from the User/Client: (top) Client-Blockchain-Gateway (CBG), and (bottom) Client-Gateway-Blockchain (CGB). The functions and events implemented in our smart contract are detailed in Table I for a complete understanding of the framework. Dashed lines indicate optional operations.

incorporating Blockchain into current IoT scenarios. As a first approximation, Blockchain could provide IoT scenarios with some desirable properties such as immutability, accountability, availability, and universal access. These properties enabled by Blockchain may be very useful for the IoT, among other things, to improve security [16], transaction reliability [17], transparency [13] or privacy levels [17].

The main contributions of this study are as follows:

- An in-depth analysis of the different possibilities for the integration of Blockchain into current IoT scenarios, focusing on: *i*) the limited processing capabilities and storage space of most IoT devices, and *ii*) the economic cost and performance of using a Blockchain.
- A novel method based on a new module named BioT Gateway that allows both unidirectional and bidirectional communications with IoT devices on real scenarios, allowing to exchange any kind of data.
- The proposed method has been implemented and validated on two different real-life IoT scenarios, extracting very interesting findings in terms of economic cost and execution time. The source code of our implementation is publicly available in the Ethereum testnet<sup>1</sup>, and can be verified using explorers such as Etherscan.

The remainder of the paper is organized as follows. Sec. II describes our proposed methods for the incorporation of Blockchain into IoT scenarios. Sec. III describes the experimental setup. Sec. IV describes the experimental results achieved using our proposed approach. Finally, Sec. V draws the final conclusions and points out future research.

<sup>1</sup>0x89f04bFE1c8dbbdA7c2A7b7815A4A3b229989f8

## II. PROPOSED METHODS

Fig. 1 shows a graphical representation of our two proposed architectures. They comprise four main modules: User/Client, Blockchain, our proposed BioT Gateway, and the IoT Device. These two proposed architectures differ in which module (Blockchain or BioT Gateway) is reached first from the User/Client: *i*) Client-Blockchain-Gateway (CBG), and *ii*) Client-Gateway-Blockchain (CGB). It is important to highlight that our proposed architectures allow unidirectional and bidirectional communications between both the User/Client and the IoT Device, expanding their use to many different practical scenarios. We describe in detail each module involved in the proposed method.

### A. Modules

1) **User/Client:** It can take any form, typically a web or mobile application that serves as the interface with the final user, e.g., to turn on/off a smart light. This Client App is typically in charge of sending commands to the IoT devices, and receiving responses or readings from them. In our proposed architectures, the User/Client App can communicate with the IoT Device through both the BioT Gateway or the Blockchain, depending on the selected configuration (Sec. II-B).

2) **Blockchain:** We assume a Blockchain capable of storing data and running code through smart contracts, a well-known concept inside the cryptographic community [18]. A smart contract is, essentially, a piece of code executed in a secure environment that controls digital assets. This concept has not been popular until its inclusion in the

Table I: Description of all the functions implemented in our smart contract to perform read and store operations. Some functions can be called only by the administrator of the platform (●), and others only by the authorized IoT Device (⦿).

Function	Caller	Input arguments	Output arguments	Restriction	Description
<i>registerGateway()</i>	Administrator	gatewayAddr	-	●	It registers a new BIoT Gateway in the system, using its Ethereum address (gatewayAddr). Only the system Administrator can call this function.
<i>registerDevice()</i>	Administrator	deviceID, gatewayAddr	-	●	It registers a new IoT Device (deviceID) for a specific gateway (gatewayAddr). Messages from other IoT devices will be discarded. Only the system Administrator can call this function.
<i>sendMessageToDevice()</i>	User/Client	deviceID, message	-	⦿	It sends a message to the IoT Device (deviceID). It also emits an event <i>messageSentToDevice</i> that can be captured by the BIoT Gateway or the Client App.
<i>sendResponseFromDevice()</i>	BIoT Gateway	deviceID, message	-	⦿	It stores the response from the IoT Device (deviceID). It also emits an event <i>responseSentFromDevice</i> .
<i>getMessagesFromDevice()</i>	User/Client	deviceID	Array with messages generated from device (deviceID)	⦿	It retrieves all messages generated by the device (deviceID) since the last time this function was called.

Ethereum Blockchain platform [19]. In essence, Ethereum could be seen as a distributed computer, with capability to execute programs written in Turing-complete, high-level programming languages.

In our proposed method, we have developed a smart contract in order to enable a secure two-way communication, in which the integrity of the exchanged data is guaranteed by the smart contract and the underlying Blockchain. As a result, we keep track of the incoming and outgoing messages between the User/Client and the IoT Device, and also control the devices registered in the BIoT Gateway to prevent unauthorized use.

Essentially, two operations are implemented in the smart contract to read and store the data coming from or going to the IoT Device. Data can have any meaning, such as control commands, readings from the sensors or any other type of digital data. All the supported functions needed to perform the read and store operations are described in Table I, including a description of each of them, input/output arguments, and permission restrictions.

3) **BIoT Gateway**: One of the main modules of our proposed architectures is the novel BIoT Gateway. It acts as an interface between all the modules of the architecture, allowing three possible communications: *i*) the interface with the User/Client, usually through an API REST (web/mobile application); *ii*) the interface with the Blockchain (smart contract); and *iii*) the interface with the IoT Device module, for example, a MQTT broker or any other protocol such as HTTPS, as long as it provides a secure communication, and the authentication of both ends. It is important to remark that the inclusion of the BIoT Gateway breaks the intrinsic distributed nature of the Blockchain. Nevertheless, this approach remains valid and secure in most scenarios. In fact, if the same entity manages both the BIoT Gateway and the IoT Device, they form a logical unit from the point of view of trust, so the security obtained would be equivalent to place the Client directly on the Device.

4) **IoT Device**: The last module of the proposed architecture is the IoT Device. It is important to remark that no

especial hardware or software capabilities of the IoT Device are needed when considering our proposed method, making it feasible for any low-cost device. This is one of the main advantages of our proposed methods.

### B. Configurations

1) **Client-Blockchain-Gateway (CBG)**: The User/Client first communicates with the Blockchain module, which broadcasts an event that is captured by the BIoT Gateway, and then moved forward to the IoT Device. Later on, the command/data is sent back to the BIoT Gateway from the IoT Device, storing the command/data into the Blockchain. Finally, it is sent back to the User/Client.

This is the most secure configuration as the User/Client can always prove that one message has been sent, even if the BIoT Gateway refuses to process it (non-repudiation). In return, a considerable delay can be introduced under some circumstances. Therefore, this configuration is more suitable for scenarios without strong latency requirements such as in the certification of information.

2) **Client-Gateway-Blockchain (CGB)**: In this case, the communication between the User/Client and the IoT Device occurs first through the BIoT Gateway. Once the command/data is received by the BIoT Gateway, it is moved forward to the IoT Device, and optionally, to the Blockchain to assure its integrity as soon as possible. Later on, the command/data is first sent back to the BIoT Gateway and then to the User/Client and optionally, again to the Blockchain, to assure its integrity as soon as possible.

This approach reduces the overhead due to the optional use of the Blockchain, but in return, security can be affected. Therefore, this configuration should only be considered in scenarios where real-time communications are needed, such as a smart-home scenario as the security of the messages exchanged is in theory not critical.

## III. EXPERIMENTAL SETUP

### A. IoT Scenarios

Our proposed method has been implemented and validated in a real environment. Two different IoT scenarios are

recreated, considering both unidirectional and bidirectional configurations:

- **Refrigerated container:** the IoT Device periodically sends its temperature and other metadata (24bytes) to the BIoT Gateway, once per minute. A unidirectional communication is considered in this scenario.
- **Smart light:** the IoT Device can receive and send simple commands (20 per day) to turn on/off the light (24 bytes), simulated with a LED matrix. A bidirectional communication is considered in this scenario.

### B. Implementation Details

For the implementation of each scenario, the following details are considered in our experiments:

- **User/Client:** a personal computer is used to receive and send commands to the IoT Device.
- **Blockchain:** a smart contract<sup>2</sup> has been developed in Solidity language, and deployed in Ethereum Ropsten testnet<sup>3</sup>. This platform is functionally identical to the main platform, but allows development and testing of applications without economic cost.
- **BIoT Gateway:** it has been implemented using a Raspberry Pi 4 [20], running the official Ethereum (Geth), and connected to the Ropsten testnet in light mode.
- **IoT Device:** we consider the Wemos D1 mini [21], a popular low-cost microcontroller based on the ESP8266 platform, connected to a temperature sensor and a LED matrix. All the elements are operated with a battery.

In our laboratory setup, both the BIoT Gateway and the IoT Device are connected through the MQTT protocol, although, as stated before, any other secure protocol may be used. Of course, this protocol must allow the mutual authentication of both elements, because they form a single “logical unit” in terms of trust. In our case, both BIoT Gateway and IoT Device are issued with a x509 certificate by a common CA. The fingerprint of the certificate of the BIoT Gateway is hard-coded (certificate pinning), to avoid man-in-the-middle (MITM) attacks.

### C. Blockchain Storage Schemes

One of the main potential limitations for the integration of both IoT and Blockchain technologies is the economic cost of running an IoT system (totally or partially) in Blockchain. It is therefore crucial to properly estimate and minimize the cost that, to a large extent, is due to the storage of data. In our experimental framework we analyze the three different storage schemes proposed in [22], which can be ordered in terms of complexity (from lower to higher), and economic cost (from higher to lower) as follows:

- **Full on-chain storage:** all data is stored, as-is, in the Blockchain.

<sup>2</sup>It is a basic contract that should be considered only for research purposes.

<sup>3</sup>0x89f04bFE1c8dbbdB7c2A7b7815A4A3b229989f8

- **Data hashing:** the Blockchain only stores a hash of the data that guarantees its immutability. The data itself is stored off-chain in another system: distributed (e.g., IPFS [23]), cloud or even existing traditional databases.
- **Merkle trees:** data is also stored off-chain, but it is preprocessed by constructing a Merkle tree structure, reducing storage costs and increasing the bandwidth.

These alternatives are discussed in terms of economic cost and execution time in the next section.

## IV. EXPERIMENTAL RESULTS

Our proposed method has been evaluated according to the following measures:

- **Economic Cost:** associated to the use of Blockchain, due to the data storage and smart contract execution. Both CBG and CGB configurations are analyzed.
- **Performance:** associated to the execution time of the smart contract. Transmission times are not included as they are negligible with respect to the smart contract.

Table II shows the economic costs and performance of each Blockchain storage scheme (i.e., full on-chain, data hashing, and Merkle trees) and architecture configuration (i.e., CBG and CGB). We also include in the last two rows of the table the economic cost and performance results obtained in our two IoT scenarios studied (i.e., refrigerated container and smart light). The economic cost is described in terms of units of gas and US dollars at the time of writing (January, 2020). Performance is described in terms of seconds.

1) **Economic Cost:** we first analyze how the different storage schemes affect the feasibility of our proposed approach in terms of economic cost. In general, the results depicted in Table II remark that Merkle trees seem to be the only viable Blockchain storage scheme. The remaining storage schemes would quickly become prohibitive for the volume of data typically exchanged in a real environment. We analyze in detail each Blockchain alternative.

The full on-chain or direct storage scheme is specially expensive (e.g., \$11.52 per day for the refrigerated container scenario) as all data is stored, as-is, in the Blockchain. The reason of this high economic cost is due to the pricing storage in Blockchain, which is intentionally discouraged to minimize its uncontrolled growth. For example, protecting the security of one million messages with this approach would cost between \$8,756 and \$22,565 for messages of 8 and 128 bytes, respectively. Depending on the final scenario and the value of the protected information, this could be reasonable but, in general, these figures are not affordable for general purpose applications.

The next storage scheme considered is data hashing. This approach slightly improves the economic cost figures compared with the full on-chain storage scheme, but only when the size of the messages is bigger than 32 bytes (for smaller messages the direct storage is still cheaper). Despite

Table II: Experimental results in terms of economic cost and performance for each Blockchain storage scheme (i.e., full on-chain, data hashing, and Merkle trees) and architecture (i.e., CBG and CGB). The economic cost and performance results achieved in our two IoT scenarios studied (i.e., refrigerated container and smart light) are included in the last two rows of the table. We have considered a gas price of 1 gwei (1 gwei =  $10^{-9}$  ETH), and 1 ETH = \$168 (January 2020).

Configuration	Operation	Economic Cost			Performance
		Full on-chain (per message)	Data hashing (per message)	Merkle trees (for any amount of data)	Execution time (average)
-	Smart contract deployment	866,212 gas (\$0.145)			10 secs.
-	<i>registerGateway()</i>	43,702 gas (\$0.007)			15 secs.
CBG	<i>sendMessageToDevice()</i>	52,132 gas for $m_{size} = 16$ 382,119 gas for $m_{size} = 1204$ (\$0.008 for $m_{size}=16$ ) (\$0.06 for $m_{size}=1024$ )	72,433 gas (\$0.012)	72,433 gas (\$0.012)	12 secs.
	<i>receiveMessagesFromDevice()</i>	-	-	-	-
CGB	<i>sendMessageToDevice()</i>	52,132 gas for $m_{size} = 16$ 382,119 gas for $m_{size} = 1204$ (\$0.008 for $m_{size}=16$ ) (\$0.06 for $m_{size}=1024$ )	72,433 gas (\$0.012)	72,433 gas (\$0.012)	13 secs.
	<i>receiveMessagesFromDevice()</i>	-	-	-	-
<b>Scenario</b>	<b>Operation</b>	<i>(per 24h of operation and device)</i> <i>(8 data bytes + 16 device ID bytes)</i>			(Per interaction)
Refrigerated container	The temperature is sent every minute to the BIoT Gateway (unidirectional)	\$11.52	\$17.28	\$0.012	BG: 15 secs GB: 0 secs
Smart light	The light is turned on/off 20 times a day (bidirectional).	\$0.16	\$0.24		BG: 30 secs GB: 0 secs

the improvement, this storage scheme is still prohibitive in terms of economic cost (e.g., \$17.28 per day for the refrigerated container scenario).

The last storage scheme studied is Merkle trees. In this case, all messages and data received during a period of time are grouped under a single tree. As a result, only the root of the tree must be secured in the Blockchain. Therefore, an arbitrarily large volume of data can be secured at the cost of only 256 bits, having a fixed cost of \$0.0122 per day and IoT Device (in our experimental setup), being this storage scheme the only one viable in real environments. The exact duration of this period (window) of time in which data is grouped under a single tree must be determined taking into account the volume of messages processed, their importance, and the cost that can be assumed. Depending on these parameters, the duration of the window can range from a few minutes to several hours or days. Finally, despite the economic cost advantages of this storage scheme, it is important to remark that if the BIoT Gateway, for whatever reason, is lost or corrupted before the root of the tree can be secured in the Blockchain, then the security of the previous messages is lost. Also, this scheme slightly complicates the verification of data integrity, as it is necessary to save a cryptographic proof that allows its reconstruction.

Finally, and although the economic cost of this last approach is very low for a single IoT Device (\$0.0122), it could be too high in a realistic IoT environment composed of potentially millions of devices. To mitigate this aspect, a Merkle tree meta-structure could be generated by aggregat-

ing the corresponding trees to the individual BIoT Gateways. This way, it would be possible to authenticate and process an arbitrarily large volume of data at a very low fixed cost.

2) **Performance:** We now analyze how the architecture configuration selected (i.e., CBG and CGB) affects the feasibility of our proposed approach in terms of execution time (last column of Table II).

In general, the experiments show that this hybrid approach based on our proposed BIoT Gateway is also viable. As can be seen, the execution time is slightly higher than 10 seconds for the *sendMessageToDevice()* operation, which is used in both CBG and CGB configurations.

This time delay could be acceptable or not depending on the final IoT scenario. For the refrigerated container scenario considered in our experimental setup (unidirectional communication), where the container periodically sends its temperature to the BIoT Gateway, it seems feasible a time delay between 10 and 15 seconds as no hard time constraints are needed in this specific scenario. Therefore, the CBG configuration should be chosen to increase the security.

For the smart light scenario considered (bidirectional communication), in which the IoT Device can receive and send simple commands to turn on/off the light, time delay seems much more sensitive due to usability reasons. Therefore, in this specific scenario, the CGB configuration should be considered as it does not add virtually delays. However, the messages exchanged are not secured in the Blockchain until the end of the window period configured for the system.

Finally, the message retrieval operation, i.e., *receiveMes-*

*sagesFromDevice()*, is free of charge, as it is a read-only operation and does not include/modify anything of the Blockchain. Also, this operation can be considered immediate in terms of execution time, since the request is processed by the local Ethereum node, and does not reach the network.

## V. CONCLUSIONS

In this study we have explored new architectures for incorporating Blockchain into current IoT scenarios. In particular, we have proposed new methods based on a novel module named BIoT Gateway that allows both unidirectional and bidirectional communications with IoT devices on real scenarios, allowing to exchange any kind of data.

Our proposed methods have been fully implemented and validated on two different real-life IoT scenarios: *i*) a refrigerated container with a unidirectional communication, and *ii*) a smart light with a bidirectional communication. Also, three different Blockchain storage schemes are evaluated in order to minimize the economic cost of data storage.

The results achieved prove that straightforward schemes such as the direct storage of the IoT templates on-chain, or direct data hashing, are not feasible for practical IoT scenarios. Nevertheless, when the Merkle tree scheme is included as an intermediate data structure, the economic cost is significantly reduced and also fixed regardless of the volume of data to store. Regarding the performance, times between 10-20 seconds are obtained for store operations whereas for read operations, they are usually free of cost and very fast to run as they are processed locally. These figures prove the viability of our proposed approach on current IoT scenarios, overcoming some limitations of most IoT devices such as the limited processing capabilities and storage space.

## ACKNOWLEDGMENTS

This work is supported by projects: PRIMA (ITN-2019-860315), TRESPASS-ETN (ITN-2019-860813), BIBECA (RTI2018 MINECO/FEDER), and COPCIS (TIN2017 MINECO/FEDER). Ruben Tolosana is supported by CAM/FSE.

## REFERENCES

- [1] K. Ashton, "That 'Internet of Things' Thing," *RFID Journal*, vol. 22, no. 7, pp. 97–114, 2009.
- [2] L. Atzori *et al.*, "The Internet of Things: A Survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [3] W. Wachenfeld *et al.*, "Use Cases for Autonomous Driving," in *Autonomous Driving*. Springer, 2016, pp. 9–37.
- [4] Statista, "Internet of Things (IoT) Connected Devices Installed base Worldwide from 2015 to 2025," 2020. [Online]. Available: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>
- [5] Ericsson, "CEO to Shareholders: 50 Billion Connections 2020," 2010. [Online]. Available: <http://www.ericsson.com/thecompany/press/releases/2010/04/1403231>
- [6] Oscar Garcia-Morchon *et al.*, "Internet of Things (IoT) Security: State of the Art and Challenges," *RFC*, vol. 8576, pp. 1–50, 2019.
- [7] I.K. Dutta *et al.*, "Lightweight Cryptography for Internet of Insecure Things: A Survey," in *Proc. IEEE Computing and Communication Workshop and Conference*, 2019.
- [8] A. Sari, "Countering the IoT-Powered Volumetric Cyberattacks with Next-Generation Cyber-Firewall: Seddulbahir," in *Security, Privacy and Trust in the IoT Environment*. Springer, 2019, pp. 83–96.
- [9] M. Samaniego and R. Deters, "Zero-Trust Hierarchical Management in IoT," in *Proc. IEEE International Congress on Internet of Things*, 2018.
- [10] H.N. Dai *et al.*, "Blockchain for Internet of Things: A Survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.
- [11] M. Swan, *Blockchain: Blueprint for a New Economy*. O'Reilly, 2015.
- [12] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," Manubot, Tech. Rep., 2019.
- [13] A. Reyna *et al.*, "On Blockchain and Its Integration with IoT. Challenges and Opportunities," *Future Generation Computer Systems*, vol. 88, pp. 173–190, 2018.
- [14] O. Delgado-Mohatar *et al.*, "Blockchain and Biometrics: A First Look into Opportunities and Challenges," in *Proc. International Congress on Blockchain and Applications*, 2019.
- [15] N. Buchmann *et al.*, "Enhancing Breeder Document Long-Term Security Using Blockchain Technology," in *Proc. Computer Software and Applications Conference*, 2017.
- [16] T.T. Huynh *et al.*, "A Survey on Security and Privacy Issues of Blockchain Technology," in *Proc. International Conference on System Science and Engineering*, 2019.
- [17] K. Wüst and A. Gervais, "Do you Need a Blockchain?" in *Proc. Crypto Valley Conf. on Blockchain Technology*, 2018.
- [18] N. Szabo, "Smart Contracts: Building Blocks for Digital Markets," 1996.
- [19] C. Dannen, *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*. Apress, 2017.
- [20] R. P. Foundation, *Raspberry Pi 4 Tech Specs*, 2020. [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/>
- [21] W. Electronics, *Wemos D1 mini*, 2020. [Online]. Available: [https://wiki.wemos.cc/products:d1:d1\\_mini](https://wiki.wemos.cc/products:d1:d1_mini)
- [22] O. Delgado-Mohatar *et al.*, "Biometric Template Storage with Blockchain: A First Look into Cost and Performance Tradeoffs," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.
- [23] J. Benet, "IPFS - Content Addressed, Versioned, P2P File System," *arXiv preprint arXiv:1407.3561*, 2014.