

Article

Blockchain-IoT Sensor (BIO-TS): A Solution to IoT-Ecosystems Security Issues

Carlos Gonzalez-Amarillo ^{1,*}, Cristian Cardenas-Garcia ², Miguel Mendoza-Moreno ²,
Gustavo Ramirez-Gonzalez ^{1,*} and Juan Carlos Corrales ¹

¹ Departamento de Telemática, Universidad del Cauca, Cauca 5 N° 4-70, Popayán 190002, Colombia; jcorral@unicauca.edu.co

² Telematics Research Group, Universidad Pedagógica y Tecnológica de Colombia, Tunja 150002, Colombia; cristianleonardo.cardenas@uptc.edu.co (C.C.-G.); miguel.mendoza@uptc.edu.co (M.M.-M.)

* Correspondence: amarillo@unicauca.edu.co (C.G.-A.); gramirez@unicauca.edu.co (G.R.-G.); Tel.: +57-313-236-3829 (C.G.-A.)

Abstract: Sensor devices that act in the IoT architecture perception layer are characterized by low data processing and storage capacity. These reduced capabilities make the system ubiquitous and lightweight, but considerably reduce its security. The IoT-based Food Traceability Systems (FTS), aimed at ensuring food safety and quality, serve as a motivating scenario for BIO-TS development and deployment; therefore, security challenges and gaps related with data integrity are analyzed from this perspective. This paper proposes the BIO-TS hardware design that contains some modules built-in VHDL (SHA-256, PoW, and SD-Memory) and other peripheral electronic devices to provide capabilities to the perception layer by implementing the blockchain architecture's security requirements in an IoT device. The proposed hardware is implemented on FPGA Altera DE0-Nano. BIO-TS can participate as a miner in the blockchain network through Smart Contracts and solve security issues related to data integrity and data traceability in an Blockchain-IoT system. Blockchain algorithms implemented in IoT hardware opens a path to IoT devices' security and ensures participation in data validation inside a food certification process.

Keywords: IoT-Device; Blockchain-IoT ecosystem; hardware development; VHDL; food traceability



Citation: Gonzalez-Amarillo, C.; Cardenas-Garcia, C.; Mendoza-Moreno, M.; Ramirez-Gonzalez, G.; Corrales, J.C. Blockchain-IoT Sensor (BIO-TS): A Solution to IoT-Ecosystems Security Issues. *Sensors* **2021**, *21*, 4388. <https://doi.org/10.3390/s21134388>

Academic Editor: Thomas K. Dasaklis

Received: 13 March 2021

Accepted: 26 April 2021

Published: 26 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Internet of Things (IoT) has become an adaptable technology to any context for collecting and transmitting data, analyzing, or informing the stakeholders [1]. In all cases where IoT systems are implemented, the sensors are the basis of the system. The sensors should ensure data transparency on the communication process, from the data collection to the transmission. Nevertheless, security issues on IoT systems are mainly due to sensors' disability for providing security on an information system.

Transparency and data integrity is a critical security issue within IoT ecosystems. Improving the communication process between peer devices might focus on solving data management from several technological schemes (fog or edge computing), but the information transparency recorded is not guaranteed. A current IoT system contains various devices with embedded sensors characterized by low power, reduced memory capacity, and limited processing capabilities. These features allow identifying the origin of significant security problems. Nevertheless, any solution concerning IoT devices' security capabilities is designed from the software that governs the management data, generally in the system's management layer. For this reason, new challenges arise today in the management of informatic security in communication systems.

Blockchain technology can be defined as a phenomenon from two viewpoints; financial and technological. Financial because the cooperation in a business process imposes cryptocurrency and breaks the standard currency scheme [2–4]. Finally, technological because

it also breaks the centralized systems concept and sets distributed systems to solve the critical issues of access and security [5–7].

Although BIoTS can act in any IoT ecosystem, we study its impact on food traceability in this paper. The traceability of Supply Chains (SC's) is defined as the ability to track food movement through specific stages of production, transformation, and distribution [8]. Scenarios such as SC's have found in IoT a powerful tool to trace and track products or raw materials in any production stage because it is highly interoperable, scalable, open, and ubiquitous. Moreover, its primary purpose is to keep the customer informed about the product's harmlessness and quality with transparent data [9,10]. The main aim of traceability is to identify the origin of foods, the manufacturing process, the used ingredients, and, most significantly, to discover the responsible party or parties whenever a product is in some way faulty [11].

SC's traceability systems enable us to locate, record, and trace products in the manufacture, processing, and distribution through platforms that offer access to users [12], for instance, IoT platforms [13]. Such media promote production quality facilitates problem identification and improves the communication capacity between stakeholders. Food security demands new concepts of trust, and blockchain is an obvious choice for further development in this regard [14].

Blockchain-IoT (BloT) can be implemented on an SC in two ways. First, from software usage, it allows management of the network resources to share the information and reach the participation of the actors of the SC through the smart contracts [15,16]. The second way involves developing hardware immersed in an IoT sensor to enhance storage and processing capabilities. Implementing a memory module and the cryptography (SHA-256) and consensus (Proof of Work) algorithms in an FPGA is intended to provide the IoT sensor with the ability to act as a miner in the mining process in the blockchain construction [3,16,17].

This work aims to identify critical security issues concerning data integrity and transparency in IoT ecosystems' communication process dedicated to food traceability. Besides, it proposes hardware specific to the blockchain architecture and includes it in an IoT sensor's architecture. This device is called BIoTS (Blockchain-IoT-Sensor) with blockchain and IoT architectural features, capable of solving security issues inside an IoT network.

This article is organized as follows: Section 1 introduces the research context of this work. Section 2 presents the related works review. Section 3 details the IoT ecosystem security issues, the Blockchain-IoT on food safety context, and BIoTS-Blockchain-IoT interaction context. Section 4 presents the BIoTS architecture and a description of how a blockchain network works. Section 5 presents the BIoTS implementation details, and the cryptographic and consensus algorithms description. Section 6 presents the results and evaluation. Finally, conclusions are offered.

2. Related Works

According to the most frequent security vulnerabilities, in [18], they were identified and classified IoT security issues. Summarizing the security threats of IoT-Systems, some works rank these threats as challenges in the security field and propose a hierarchy of security issues; (i) low-level security issues highlight the insecure initialization, insecure physical interface, or jamming adversaries. (ii) Intermediate-level security issues highlight the insecurity of network connectivity between devices, authentication, non-secure communication on end-to-end transport-level security, and privacy violation on cloud-based IoT. (iii) High-level security issues highlight CoAP safety with the Internet, insecure interfaces, insecure Software/firmware, and middleware security. Some emerging technologies, like blockchain, can solve the majority of security problems present on IoT ecosystems, a fact that makes a new Blockchain-IoT (BloT) concept possible.

Generally, networks based on IoT may suffer identity violation and information privacy issues, such as the services related to cloud, storage, transmission, or processing [19]. Security systems on the Internet about Constrained Application Protocol (CoAP) suffer

security attacks from the application layer [20,21], this fact makes the web, mobile, and cloud interfaces vulnerable, as those indicated in (OWASP IoT top 10).

Security problems like jamming attacks are considered minor problems. Nevertheless, the message collisions and errors on the sending of the packages are solved in [22], where they measure the signal to extract the noise, then compare these measures with customized threshold measurements and detect the attack. Other solutions against a jamming attack use cryptographic functions to help correct errors [23]. Others suggest avoiding the jamming attack with encoded packages through a division of the message into blocks or changing channel frequencies for the communication flow to be successful [24].

Table 1 presents work related to solving security issues for each architecture layer of an IoT system-based traceability systems. This table presents solution alternatives with a similar approach to BIoTS for some security issues identified from the problem map in Figure 1. However, it is intended to highlight the novelty that BIoTS means in the application of blockchain in scenarios such as this one. This table describes the IoT ecosystem's technical and technological features that BIoTS will perform to ensure product and data traceability.

Table 1. Security issues, threats, and technologies.

Architecture Layer	Threats in Security	Weaknesses	Related Works	Attacks
Application Layer Internet	Confidentiality	Access Centralization	[25–29]	Phishing, Malware
Transport Layer Wireless	Rogue access points, Misconfiguration	Hacking, Signal lost	[30,30–33]	DoS, War dialing, protocol tunneling; man-in-the-middle
Perception Layer Sensor Nodes	DoS, Exhaustion, Unfairness, Sybil	Flooding, Routing Protocols	[30,34–39]	Jamming, Tampering, Collisions

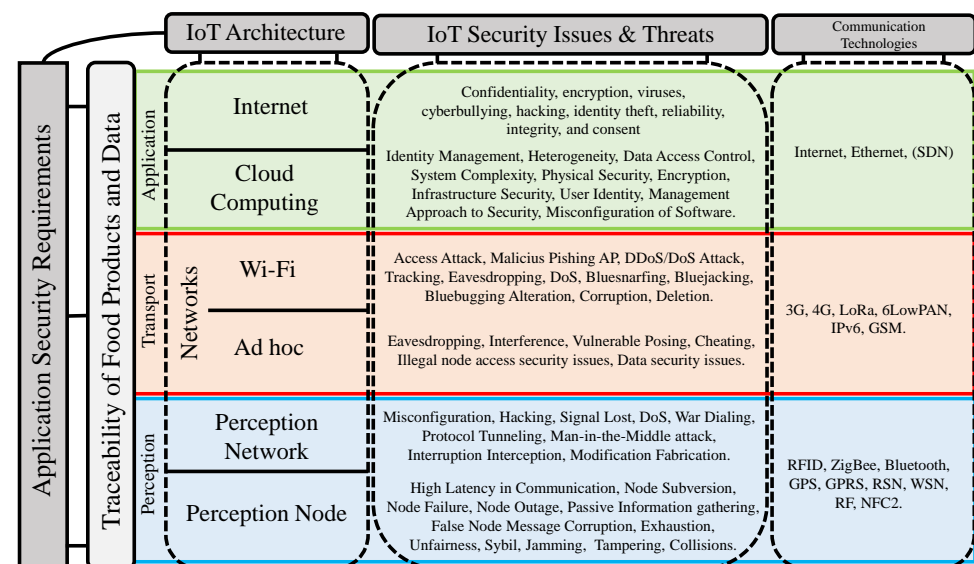


Figure 1. IoT security issues and threats.

The works listed in the table above define the roadmap for working in agricultural product traceability security, and helps to relate security vulnerabilities, scientific work, and threats. Technologically and conceptually, we focus on the problem and evaluate alternative solutions [40,41]. In the application layer, work is related to the deployment of a blockchain network, especially those that address security issues that impact food traceability.

Some remarkable findings for BIoTS hardware design are; a) most IoT systems' sensors will have some features, such as; interoperability, energy, size, position, and communication.

These features make the ubiquity term possible and make the system a lightweight system for they can adapt secure form between them to deploy a service in any context [42]. b) Biosensors, capable of identifying pathogens in contaminated products in the food processing industry, have gained relevance. For this reason, this proposal attempts to focus on the construction of a sensor-equipped with a technology (Blockchain) capable of guaranteeing data integrity and transparency in the transmission of information [43,44].

3. Foodchain Traceability Using Blockchain

3.1. IoT Security Issues and Challenges

Given the range of services provided by objects, persons, or machines in the IoT networks, it is mandatory to equip both networks and devices with security features.

The standard communication protocols define the rules and security techniques in IoT networks. Fields such as health, financial security, or food safety handle processes with sensitive data that require transparency and integrity in their handling. However, the adverse factor is that as long as the design of the IoT network is done on the LLN (Low-Power and Lossy Network) network scheme, security will have that measure; that is, the device immersed in the IoT network will not have security properties beyond those possible by its capabilities.

3.2. Overview of Security Issues on IoT

Identifying security problems on IoT is so extensive that it generally is made from the field of application. Furthermore, the application field imposes safety criteria focused on the user and the system architecture. This hierarchy in the identification of security problems helps to identify comprehensive solutions and technologies.

Due to the technology's capacity with which the build of BIoTS device proposed in this work, it is possible to identify the problems attending the two perspectives (Application and architecture), because of its implementation answers integrally to the IoT security problems. For this reason, this section presents the findings of some security problems, keeping a mixed approach between the security requirements of the application field and the IoT architecture.

The scheme in Figure 1 shows an IoT security issue map, which identifies in a general way the problems that affect to the IoT-based food traceability systems and focused on food safety and quality. This scheme is based on three works that propose taxonomies for identifying security issues on IoT [18,45–47].

- Application Field Perspective: On the left of the scheme, food safety as an application field, establishes the three layers of IoT architecture as a channel to guarantee data and product traceability in the communication process throughout the IoT system.
- Architecture Perspective: In the upper part, the IoT architecture (Perception, Transport, and Application) establish as the central axis in the identification and classification of security problems.

The implementation of BIoTS aims to solve the security problems described here. Each layer in the IoT architecture has issues that the technology on which BIoTS solves.

1. Perception Layer: In this layer, where the devices that interact with the medium are hosted, we list and describe some security issues that BIoTS can solve in two scenarios; (a) in sensor nodes and (b) in sensor gateways. (a) In the sensor nodes case, to make the process of sensing and interconnecting with other nodes possible, they have these components; a controller, a transmitter (for communication), a memory where the device stores the program (code), a power source, and the hardware that obtains the sensed data. At this level, you are prone to security problems such as; node subversion, node failure, node outage, passive information gathering, false node message corruption, exhaustion, unfairness, Sybil, jamming, tampering, and collisions. (b) For sensor gateways, the collection of information on WSN represents a problem because the wireless communication channel involving radio communication

- and its possible appears problems such as; misconfiguration, hacking, signal loss, DoS, war dialing, protocol tunneling, man-in-the-middle attack, interruption interception, and modification fabrication. As you can see, in both cases, all security problems are aimed at attacking the trust, privacy, and integrity of the transmitted data.
2. **Transport Layer:** To solve some security vulnerabilities associated with the network type BIoTS acting in a blockchain-based IoT network. The networks generally used for food traceability systems are two; (a) WiFi-centric network and (b) ad hoc non-centric network. For this reason, BIoTs and your ecosystem pretend to solve some security issues as: (a) In a WiFi network, attacks such as access attacks, malicious phishing AP, and DDos/Dos attacks. (b) In IoT, an unfocused ad hoc network is a Peer-to-Peer network. The traditional problems in this nature's networks have to do with the communication channel's vulnerability—attacks such as Eavesdropping, interference, vulnerable posing, cheating, Man-in-the-Middle (MitM).
 3. **Application Layer:** In the food safety scenario, millions of users are expected to access sensitive information on edible products. Data confidentiality and traceability is the anticipated contribution of BIoTS in the network deployed for its operation. The ecosystem is expected to contribute to security issues associated with authentication and access authorization. Besides, process safety management within a supply chain based on certification through blockchain Smart Contracts is expected.

The BIoTS sensor's design requires scanning security issues that it can solve in the IoT context. However, the sensor can be used in a variety of fields. Next, the technological ecosystem in which BIoTS deploys its function is analyzed, i.e., the adaptability with an IoT environment, where there are also security gaps to overcome, is determined. The security problems map designed from the two points of view described in Figure 1, the design of BIoTS is focused on the issues that affect the perception layer of the IoT architecture depicted in Figure 2. Once the solutions proposed by each layer of the architecture in Table 1 are identified, the BIoTS architectural characteristics are defined according to IoT and blockchain technologies' security requirements. Finally, the algorithms that will be designed in VHDL language for hardware construction are described.

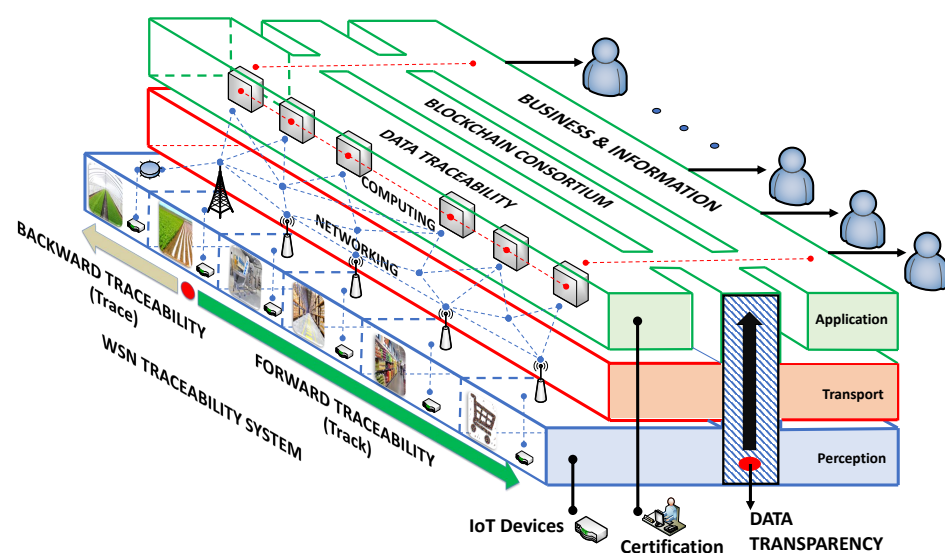


Figure 2. The Blockchain-IoT-based food traceability systems.

3.3. Blockchain-IoT on Food Safety Context

IoT can trace or track food products, and store and process critical data for recording the process; nonetheless, it is not entirely safe in terms of data transparency held. Blockchain, beyond the crypto-currencies and the business, can certify processes transparently through data traceability.

The ability to certify the processes in food supply chains through the data integrity collected from end-to-end connection points and contained in the Blockchain-IoT architecture reveals BIoTS device's aim. End-to-end connection points in the IoT architecture indicate the sensor's path to the cloud service (Vertical). Simultaneously, the path from the first stage in the traceability system to the last location (Horizontal). See Figure 2.

Combining two disruptive technologies involves identifying the problems they face in applying them in food security and then finding the architectural features that make the coupling possible.

3.4. The Blockchain-IoT-Based Food Traceability Systems

Figure 2 shows the architectonic scheme of the Blockchain-IoT-based traceability system. The vertical levels represent three layers of IoT architecture (Perception, Transport, and Application). The six horizontal stages show the flow of agricultural products throughout the supply chain, from field to consumer. The dotted red on the Vertical levels represents the flow of data in any case. The dotted red in the horizontal stages describes the orientation of the traceability; (i) tracking the product throughout forwarding traceability, and (ii) tracing the product, aiming to know the product origin.

Blockchain technology is defined as a disruptive technology that imposes a new paradigm that can be connecting securely to the world throughout the network. Blockchain technology can be described as a platform where the transactions and the information recorded are safeguarding through cryptography algorithms in a distributed ledger to all participants of the network [48–50].

IoT represents an opportunity to apply blockchain technology as a support to guarantee security in some respects [51,52]. As we can see, blockchain technology is called to resolve significant problems of connecting, support, security, business, and stakeholders participating in food traceability systems (Supply Chains or Value Chains).

3.5. Interaction between BIoTS and Blockchain-IoT-Based Traceability System Architecture

Figure 3 shows how the BIoTS device and blockchain network interact to certify processes that depend on the information collected by BIoTS in the supply chain stages. Once the smart contract is programmed with the requirements to certify a process, in this case, humidity and temperature data, the BIoTS devices act as miners to propose a transaction and validate it within the blockchain network. In this way, the collected data will enjoy the security privileges of a blockchain system.

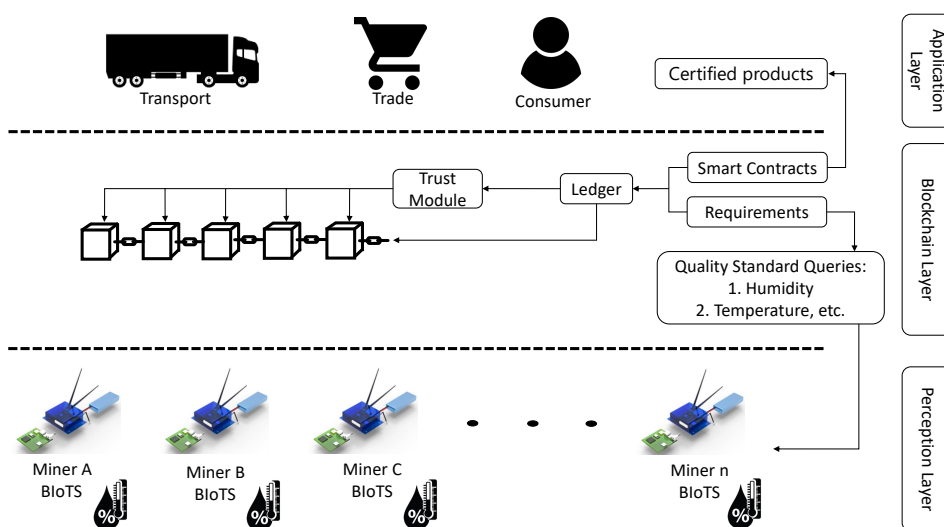


Figure 3. BIoTS System Operation.

4. BIoTS Architecture

A sensor is an embedded device capable of acquiring information, processing it, analyzing it, storing it, and transmitting it to a repository. It also can coordinate with other networked devices. Under this concept, we describe BIoTS-Sensor architecture features that allow us to define the necessities of functioning to design a Blockchain-IoT system. Then, we make a description of some architectural modules.

Most sensors immerse in the IoT ecosystem, further measure some variables, and have some capacities to provide security. Nonetheless, it does not guarantee specific security requirements for low processing, energy, and storage capabilities.

These reduced capacities make the lightweight of the IoT systems and guarantee the ubiquitous characteristic of the system. However, reducing the size of the devices (sensors) present in the IoT ecosystems conflicts with the entire system's security capabilities. For this reason, the leading security solutions presented by scientific research propose solutions on the transport layer, or (fog), and on the application layer, or (cloud), to manage security. However, the possibility of adapting the sensor hardware to an IoT architecture based on blockchain to provide the security system has not been studied so far.

Exploring this possibility has technical implications at the level of architecture and resources, for example; the system is no longer light, but there are fields of application (food and health) where the robustness of the system is worth the cost, especially if the integrity and transparency of the information are guaranteed. The BIoTS-device architecture description is made from the functional features of two technologies; IoT and blockchain. In the assembly and synchronization of both architectural blocks, it is necessary to identify the security requirements, both for software and hardware [53,54].

Block A in Figure 4 is the architectural approach of an Ethereum's blockchain. This block describes the structural layers that form a security system. For this proposal, we focus on the Miners layer's study and analysis. In this layer, we found the physical devices (Computers) that interact in the Network to make the blockchain valid. The fundamental elements in this layer are two; storage and processing capacity.

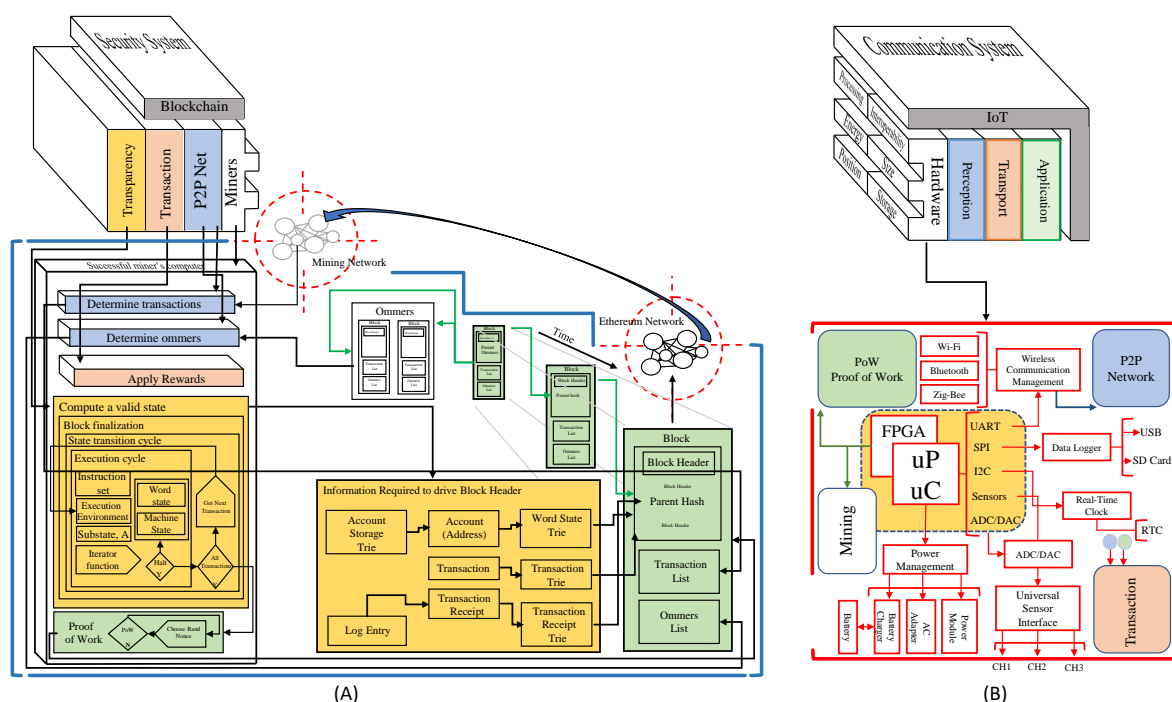


Figure 4. Blockchain-IoT Architecture Matching. (A) Blockchain Ethereum Architecture Approach by Lee Thomas based on [55]. (B) IoT-Sensor Architecture.

As we can see in Figure 4, block B represents IoT as a communication system where a service is deployed through some architectural layers Figure 2. In this block B, We focus on designing and improving the hardware capabilities in the perception layer to splice this device with block B's blockchain software technology. The primary device features that act in this layer are; (i) interoperability, (ii) processing, (iii) energy, (iv) size, (v) position, (vi) storage capacity, and (vii) security. Moreover, it can evaluate the hardware device quality involved in an IoT ecosystem according to these features. For this reason, they are the ones we take into account to develop the new BIoTS capabilities.

The block B, we can see the module that ought added to the sensor. Each module responds to the need created by the blockchain system. The modules are related by color; thus, the blue module of the P2P Network is designed to make possible the P2P Network in which the sensor acts as Miner. The green module of the Proof of Work (PoW) algorithm makes transaction validation possible and guarantees the block's information's immutability. The yellow module subject to the Mining process is designed to calculate hashes in the communication's cryptographic function. Finally, the orange module is designed to store the records for each validation through the Merkle tree.

As we can see, the junction of these blocks, the IoT as an oriented communication system, and the blockchain as a security system, together represent a communication system with a high-security level. The agro-food traceability application field, where this solution is considered, means the traceability of both product and information.

Figure 5 shows the conventional path1 used in the information flow in an IoT system and establishes the data flow and the interaction between devices within the IoT ecosystem. Path2 (Called BIoTS-Paths) sets a direct action in the information transmission, avoiding any intermediary. The BIoTS block (B) aims to make IoT hardware a miner actor in the blockchain architecture. As we can see, the objective from the hardware point of view is to make the coupling of the physical layer of the two architectures possible.

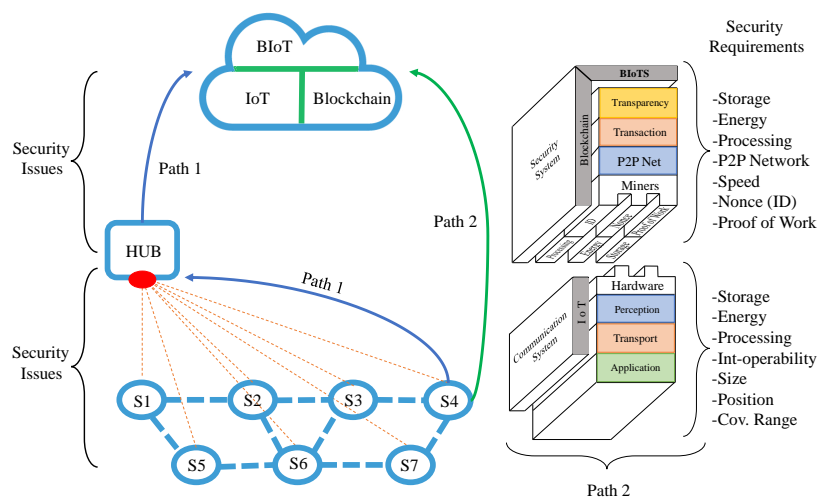


Figure 5. Path 1: conventional data transmission in an IoT system. Path 2: architecture and transmission path proposed by (BIoTS-Paths).

The architecture adaptation deserves to describe the hardware blocks that make the coupling possible. The following describes the Blockchain Ethereum that will govern the system and the development of the sensor module that allows it to operate in this network.

Blockchain

BIoTS has as a challenge to adapt all its architectural modules to the functional requirements of blockchain. In this case, it is necessary to adapt two algorithms at the hardware level; (i) the SHA-256 algorithm responsible for cryptography in the communications

process and (ii) the Proof of Work (PoW) algorithm responsible for the consensus process in the network.

As its name indicates, blockchain is a chain of blocks that systematically stores information in a decentralized network. Each node acts as a miner, and these, generate validation through their processing capabilities. The information contained in each block is interconnected with the previous block employing a hash, making it impossible to reverse or modify data in each block. That's where blockchain's security comes from Figure 6.

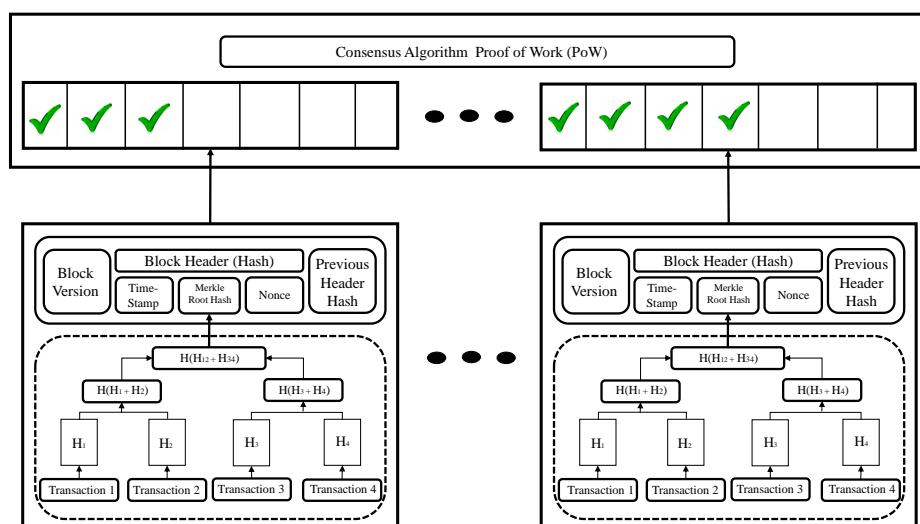


Figure 6. Blockchain system architecture and transaction validation mechanism.

Figure 6 shows the principal modules contained in each block of the blockchain. The block header module contains the hash (identification on the blockchain system) for making possible interchange transactions on the network. Hash together with the nonce module; they make up a firm, part of a public and private key to transactions on the network. The block version module contains the block number (series of consecutive numbers) throughout the chain of blocks; this module serves as an identifier to know their position in the chain. The Time-Stamp module guarantees the distributed temporal database contained on each miner in the network. This module assists part of security in the system because the proof of work algorithm reads and processes it to reach the consensus. Merkle Root Hash module allows us to know the origin and history of hash blocks; this feature makes it impossible to decipher the hashes' chain for obtaining the address or the contend any block. The nonce module assigns a zeros-chain before of hash in the block header module; thanks to this feature, each block into the blockchain has unique identification to transactions (Represents other security behavior of blockchain). Finally, the Previous Header Block module is responsible for saving the previous header hash for adding to the new hash in the new block; this module serves to form the Merkle root hash module [14].

5. Implementation Details

5.1. BloTS Cryptographic Algorithm

A hash function can convert an input message with a specific length into an alphanumeric array on the output called a digest. A hash function has the following characteristics [56].

- The reverse process of reconstructing the message from the hash is almost impossible.
- A minor change in the input message completely changes the output.
- The algorithm can compress any extension of the input message for arranging the output. It is impossible to find the same hash for two different input messages.

The SHA-256 algorithm has two modules; (i) Message Block schedule and (ii) Compression function. Below is a brief description of the modules. In the message schedule module, an N-bit message gets added with bit 1, followed by zero bits until the following, Equation (1).

$$N + 1 + k = 448 \text{ mod } 512 \quad (1)$$

is satisfied, where k indicates the number of zero bits to be added. The value N is then converted to its 64-bit binary representation and further added to the 448-bit intermediate value to get the 512-bit message block. This formed block is further subdivided into sixteen 32-bit word sub-blocks that input the compression function.

Compression function involves 8 registers, a, b, c, d, e, f, g, h , and 6 logical functions $Ch, Maj, \Sigma_0, \Sigma_1, \delta_0, \delta_1$. There are another set of eight registers, $H_0, H_1, H_2, H_3, H_4, H_5$, and H_6, H_7 , to store 32-bit hash values, which are updated Mtimes if there are M 512-bit message blocks. These registers are initialized with 32-bit constant values obtained by considering only the fractional part of the first eight prime numbers after taking the square root. Logical functions comprise XOR, right rotation, and right shift operations. This complex operations are performed on 32-bit words for 64 rounds.

Following functions are computed on each round, and the registers are updated:

1. Calculate $Maj(a, b, c), Ch(e, f, g), \Sigma_0(a), \Sigma_1(e), \delta_0(a), \delta_1(e)$.
2. Words are prepared for each round using the below equation: For the first 16 rounds,

$$W_n = Message_n^i \quad (2)$$

where n ranges from 0 to 15 and i indicates number of message blocks. For the other rounds,

$$W_n = \delta_1(W_{n-2}) + W_{n-7} + \delta_1(W_{n-15}) + W_{n-16} \quad (3)$$

3. Six registers b, c, d, f, g , and h are updated with the previous registers value, i.e., a, b, c, e, f, g , respectively, after each round of operation. While register $a = T_1 + T_2$ and register $e = d + T_1$.
4. T_1 and T_2 have the following equations:

$$T_1 = h + \Sigma_1(e) + Ch + W_n + K_n, \quad (4)$$

K are a set of 64 constant words.

$$T_2 = h + \Sigma_0(a) + Maj \quad (5)$$

After 64 rounds of operation, registers H_1 to H_7 are updated for i ranging from 1 to M as follows:

$$\begin{aligned} H_0^i &= H_0^{i-1} + a \\ H_1^i &= H_1^{i-1} + b \\ H_2^i &= H_2^{i-2} + c \\ H_3^i &= H_3^{i-3} + d \\ H_4^i &= H_4^{i-4} + e \\ H_5^i &= H_5^{i-5} + f \\ H_6^i &= H_6^{i-6} + g \\ H_7^i &= H_7^{i-7} + h \end{aligned}$$

A final 256-bit hash value is obtained by concatenating 32-bit values H_0^M to H_7^M .
 $Hashdigest = H_0^M H_1^M H_2^M H_3^M H_4^M H_5^M H_6^M H_7^M$.

The SHA-256 pseudo-code Algorithm 1 in the algorithm allows for a more comfortable hardware design. Generating hashes and encrypting data will enable us to understand the software structure division and translate the functions' flow for the hardware.

Algorithm 1 SHA-256.

```

1: for Compression Function do
2:   Message Schedule module (Equation (1))
3:   Words are prepared for each round  $M_{aj}$  (Equation (2))
4:   for First 16 rounds  $W_n$  do (Equation (3))
5:     Six registers  $b,c,d,f,g,h$  are updated with the previous registers
6:      $K$  are a set of 64 constant words (Equations (4) and (5))
7:     After 64 rounds of operation  $H_1$  to  $H_7$ 
8:     if then  $H_7^i = H_7^{i-7}$ 
9:       Final 256-bit Hash value is obtained by concatenating 32-bit values
10:    end if
11:    Hash digest =  $H_0^M$  to  $H_1^M$ 
12:  end for
13: end for

```

5.2. Consensus Algorithm Analysis for BIoTS

The consensus algorithm establishes the mining agents' computational effort to solve the mathematical puzzle that validates transactions within a blockchain network. Consensus algorithms can be categorized into two groups; proof-based consensus and vote-based consensus. In the first case, the node wishing to join the network must demonstrate higher processing and storage capabilities than the rest of the network. In the second, each node in the network is asked to propose or validate a transaction block that will be part of the validation in the rest of the network. The final decision is made only after considering the majority's results. Thus, some algorithms were analyzed theoretically and based on [17,57,58] to select the BIoTS algorithm, some voting-based consensus algorithms; Proof of Vote (PoV), Ripple, Delegated Byzantine Fault Tolerance (DBFT), and Proof of Trust (PoT). Furthermore, two proof-based; Implicit consensus and Proof of Work (PoW).

The most common consensus algorithms in Blockchain (Proof of Stake (PoS), Delegated Proof of Stake (DPoS), and Practical Byzantine Fault Tolerance (PBFT)) limit the BIoTS ecosystem for the following reasons:

- PoS: It is based on the concept of the age of the coin, this age being known as its value multiplied by the period after its creation. In other words, the longer a node has a currency, the more privileges it will obtain in the network. For this reason, the BIoTS ecosystem for food traceability systems does not require concepts of this type.
- DPoS: It is based on the fact that each node in the network can select tokens according to their participation. These selected tokens create new blocks one by one as assigned and get a reward. Throughout the network, the n best witnesses who has participated in the transaction's validation and has obtained the highest number of votes are entitled to the benefit. Blockchain using DPoS is more efficient and saves more energy than PoW and PoS. However, in the Blockchain-IoT ecosystem where BIoTS is deployed, it is not expected to have enough witness nodes to validate the data BIoTS collects. A BIoTS P2P network is expected to operate with consensual data sharing.
- PBFT: Designed to solve transmission problems and improved to avoid exponential operations. Regarding BIoTS, it is not convenient to use it as it requires a master server to execute the validation throughout all supply chain stages.

Most Blockchain networks are decentralized, with synchronous or asynchronous communication models, and are implemented in networks of nodes where mining agents are processor-based; consensus algorithms' behavior is subject to factors such as; Blockchain type, transaction rate, scalability, adversary tolerance model, experimental setup, latency, throughput, bandwidth, communication model, communication complexity, security attacks, energy consumption, mining, consensus category, and consensus finality. Here, we analyze some of these.

Table 2 shows some characteristics of the consensus algorithms studied for implementation in the BIoTS device. As can be seen, physical experimentation on hardware to find the performance parameters does not yet exist. However, theoretically, it is possible to establish the suitability of some of them according to the Blockchain network design.

Table 2. Generic Features Analysis of Consensus Algorithms (based on [17,57]).

Consensus Algorithm	Blockchain Type	Mining	Consensus Category	Reference	Experiment Setup	Communication Model	Energy Consumption
PoW	Permission-less	Based on computational power	Proof-based	[59]	Real implementation	Asynchronous	538 KWh
Implicit Consensus	Permissioned	Proof based mining	Proof-based	[60]	Theoretically evaluated	Asynchronous	Unknow
PoV	Consortium	Vote-based mining	Vote-based	[61]	Simulation, Single machine	-	Unknow
Ripple	Permissioned	Vote-based mining	Vote-based	[62]	Simulation, Single machine	Asynchronous	Unknow
DBFT	Permissioned	Non-proof of work based mining	Vote-based	[63]	Proposed solution is not validated through experiments	Asynchronous	Unknow
PoT	Permission-based consortium	Probability and vote based mining	Vote-based	[64]	Simulation, Single machine	Asynchronous	Unknow

After this analysis, it is concluded that some voting-based consensus algorithms can be very relevant for BIoTS performance in a Blockchain-IoT ecosystem. However, the assembly and comparison of the consensus algorithms on FPGA are required, and this work does not have that scope. The BIoTS architecture was based on PoW because this algorithm is robust (Used in Bitcoin). The hardware built is expected to participate in an open blockchain network with significant security challenges.

5.3. BIoTS Consensus Algorithm (PoW)

Proof of Work consensus algorithm is a mechanism that allows users or machines to coordinate in a distributed network. This algorithm ensures that all agents in the system can agree on a single truth source, even if some agents fail. In other words, a system with PoW is tolerant of security failures.

The process of verifying the block's transactions to be added, organizing these transactions in chronological order in the block, and announcing the newly mined block to the entire network does not take much energy and time. The energy-consuming part solves the "hard mathematical problem" to link the new block to the last block in the valid blockchain. When a miner finally finds the right solution, the node broadcasts it to the whole network simultaneously, receiving a cryptocurrency prize (the reward) provided by the PoW protocol. We show the PoW pseudo-code Algorithm 2 [14].

Algorithm 2 Proof of Work.

```

 $r \leftarrow ab$  ▷ Define variable to answer
1: for Loop from 1 to n do
2:   var  $x = n$ ;
3:   for n major 1  $x_0 * y_0 + n$  do
4:     var added = 0;
5:     for (var i = 0; i minor  $Math.abs(a)$  ; i++) added += answer; do
6:       answer = added (n-);
7:     end for
8:   end for
9: end for

```

Implementing the PoW in hardware determines, among other things, the difficulty in block validation (time invested in identifying the block legitimacy), and the active participation of blockchain network miners. To the hardware block of the SHA-256 algorithm, it is necessary to adapt a function capable of adding in each round of the cryptographic encoding a string of zeros (from 4 to 18 maximum, depending on the difficulty of the algorithm) that will act as an identifier of block and the transaction in the whole blockchain.

Figure 7 describes the process carried out to add the PoW to the SHA-256 algorithm in hardware. The four stages are:

1. **Generate Random Nonce:** once the clock for the hash and PoW is configured and synchronized, a 32-bit register and bus are generated. The first thing is to introduce a nonce every six cycles with a feedback delay of 12 cycles.
2. **Build Block:** since this is not straightforward, we use a register to track the hash through the bus. To follow the nonce, we pass the least significant 8 bits of the 32-bit register and then pair the remaining 24-bit.
3. **Difficulty:** in the second of two rounds of hashing of 64 bytes each, the header of the 80-byte block is the encrypted data space. The first round gives us the average state to insert the nonce at the beginning of the record of the second 64 bytes in the correct position.
4. **Broadcast Block:** subsequently, the internal hash transformation is performed to complete the register. It is still not the full SHA-256 because it involves multiple rounds. Nevertheless, this process is iterative. Here, the VHDL code in the DE10-Nano is split into phases to discriminate the SHA-256 transformations, and then unified into one block intended to do the complete hash transmission.

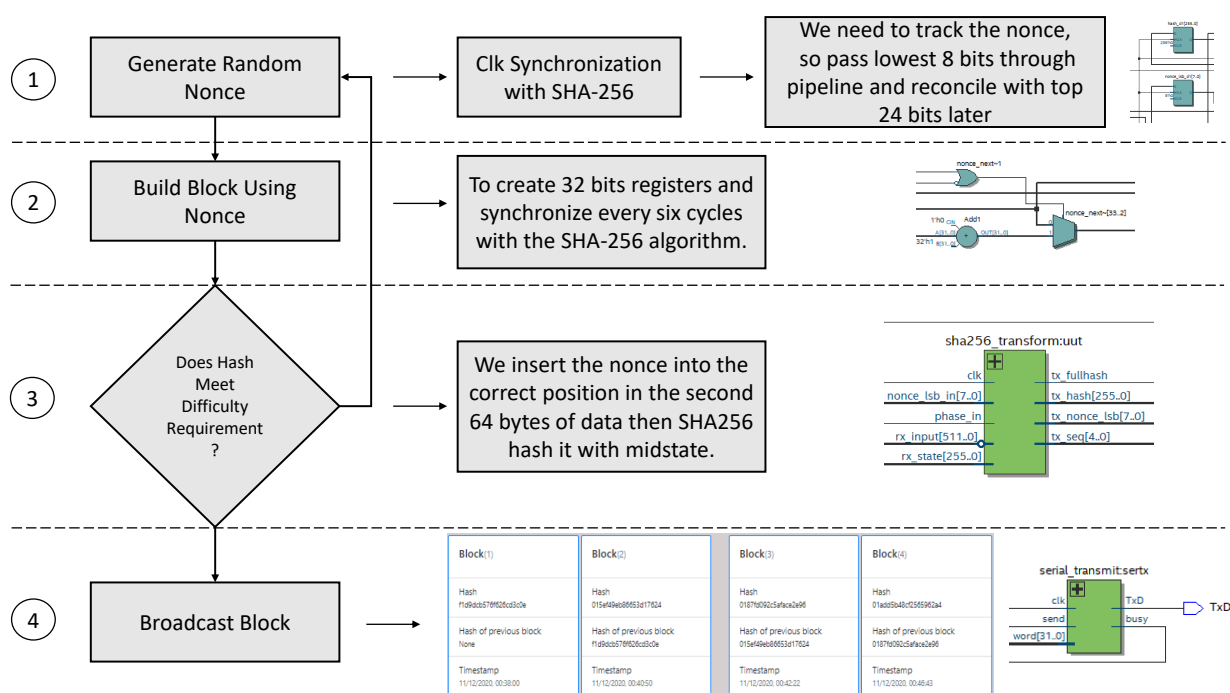


Figure 7. Proof of work implementation on hardware.

When the PoW algorithm checks zeros' existence in the hash encoded in base 64, the average and the maximum number of hashes is known to calculate the order of difficulty that increases exponentially by the expression (6). The nonces included in hashes are pseudo-random, and this feature extends the capacity of the PoW.

$$h(\rho) = \alpha^\rho \tag{6}$$

where, $h(\rho)$ = the average number of hashes required to find a valid solution α = the number of characters used in the encoding ρ = the arbitrary difficulty order.

$$h(\rho) = 64^\rho$$

$$h(3) = 64^3 = 262,144$$

Thus, typically 262,144 hashes or less are required to mine each block while testing this algorithm. However, the difficulty is arbitrarily adjusted by modifying the ρ variable. Changing ρ changes h exponentially and could be used to maintain a consistent network block generation rate despite exponentially increasing computational power.

6. Results and Evaluation

The BIoTS device comprises a platform of peripheral analog electronics connected to the digital module designed in a reconfigurable FPGA. The digital module contains several sections; cryptographic and consensus algorithms (SHA-256 and PoW) and SD storage hardware structure. The BIoTS hardware structure contains two modules (yellow and green) and one software blue module (Build of Blockchain on Python). Figure 8 shows on the right the modules designed on the DE0-Nano FPGA; these hardware modules are the ones that make the parity and interoperability of BIoTS with a blockchain network possible. This module contains, among others, the SHA-256 Algorithm, PoW Algorithm, I2C module, and SD-CARD module. These modules will be briefly described below. The next green module shows the peripherals in analog electronics for BIoTS can interact with the media, store information collected, and become a blockchain network node. The blue area contains the software development module for the blockchain network in Python, as described in the previous section.

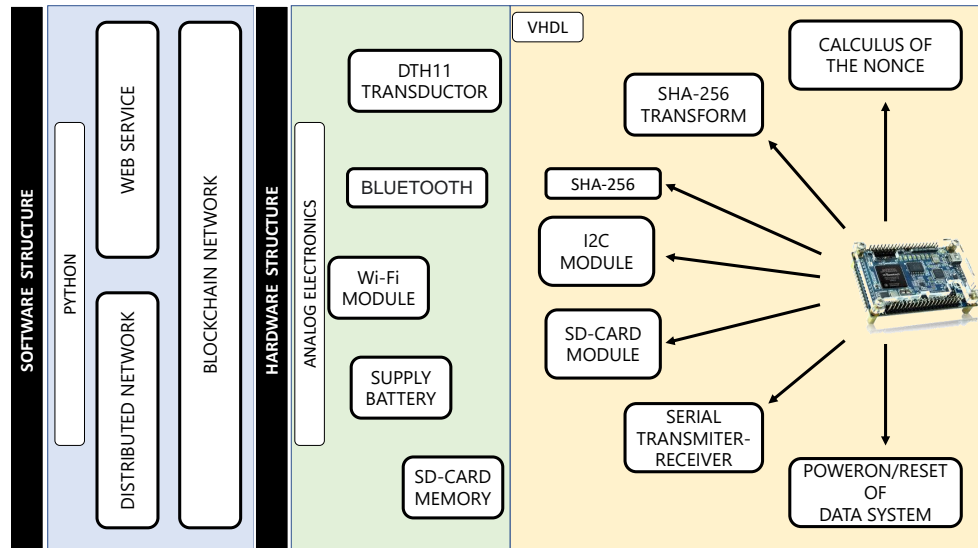


Figure 8. Structure of architectural development.

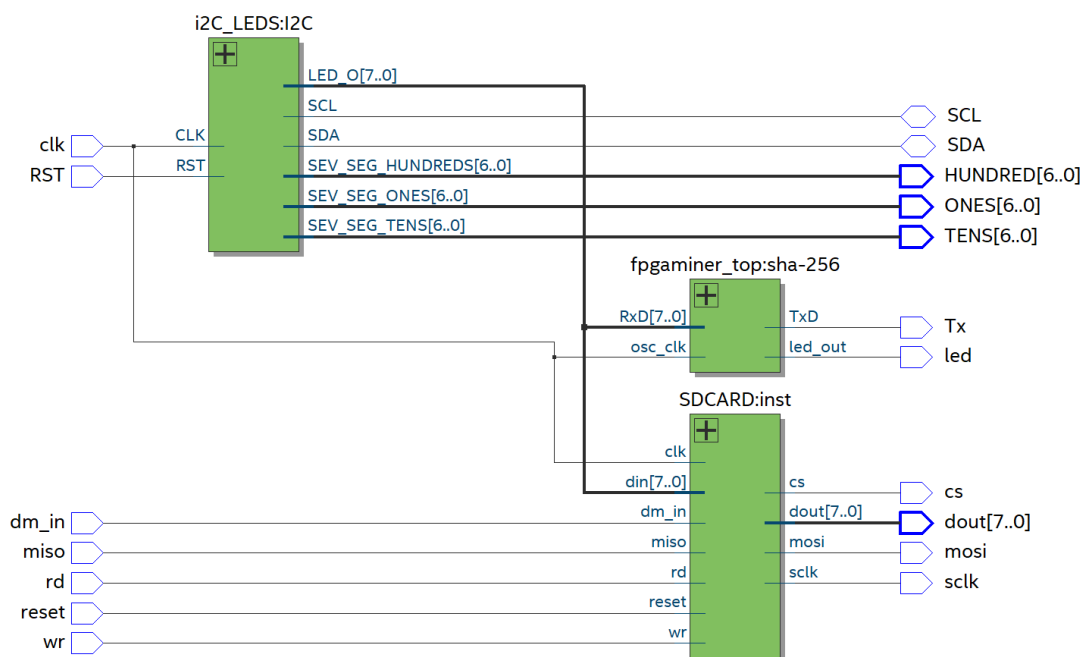
The three blocks mentioned above occupy 47 percent of the available logic units in the DE0-Nano FPGA, barely sufficient resources for implementation [65]. In Table 3, we describe the resources available and used by the three modules.

Figure 9 shows the block diagram that summarizes the deployment of the BIoTS architecture in VHDL. As we can see, there are three blocks; I2C, SHA-256 (That contains the PoW algorithm), and the SD-CARD architecture. BIoTS need these three blocks to process two complex algorithms and store a distributed database. Here, we show the relationship between modules and how the functions and records interact. Each of these blocks contains the configuration of the logical elements that make the assigned tasks possible.

Table 3. Logic elements used on DE0-Nano FPGA.

FPGA	Total Logic Elements	Percentage Available
DE0-Nano	22,320	100%
Block	Total Logic Elements	Percentage Used
SHA-256 and PoW	10,347	46%
I2C-Master	168	≤1%
I2C-Slave	114	≤1%
SD-CARD	289	1%
Total Area Used	10,556	47%

This SHA-256 algorithm design contains the Proof of Work algorithm development, necessary to calculate the nonce and make participation in the consensus process possible. The I2C master component for single master buses, written in VHDL for use in FPGAs, has component reads from and writes to user logic over a parallel interface. It was designed using Quartus II, version 18.0. Resource requirements depend on the implementation. A design incorporating this I2C master to create an SPI to I2C Bridge is available.

**Figure 9.** Diagram block of BioTS.

The SD interfaceable with FPGA is implemented from VHDL code. Here, we implement the standard size, but electrically, all sizes work the same way. Let us focus on SD card standard size, since that is conveniently popular nowadays. To this proposal, we install an SD card of 32 GB.

BioTS analog and digital electronics' energy consumption is calculated based on the evaluation scenario's functional performance. The elements that consume the most energy are; the peripheral elements: the Wi-Fi module and Bluetooth. For the digital elements: SHA-256 cryptography algorithm, PoW consensus algorithm, and the SD memory block. The BioTS power source is a three-cell LI-PO type battery at 11.2 Volts, 20–30 c discharge, and 5000 mA/h. The BioTS analog module consumes 220 mA/h transmitting data at a 1-min interval. The power consumption of the FPGA depends on the optimization of the configured algorithms and the read and write speed of the SD memory (Sequential Read = 90 MB/s and Sequential Write = 40 MB/s).

Under these conditions, the battery life is 45 min. However, as transactions occur, the complexity algorithm kicks in and demands more processing power from the FPGA. Thus, the power consumption is dynamic, and the FPGA performance is proportional to the blockchain network's activity.

Figure 10 shows the physical design in its prototype version once it is moved to integrated circuit hardware in a professional Beta version. This BIoTS prototype is the result of the invention described throughout this work.

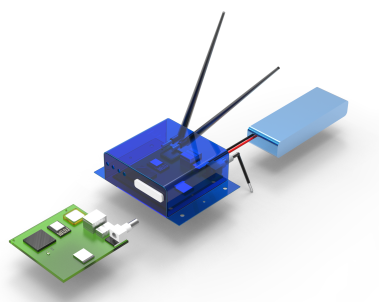


Figure 10. Full-scale modeling of BIoTS prototype.

Evaluation Scenario

The BIoTS performance is evaluated according to the configuration shows in Figure 11. As we can see, the private blockchain network in which the sensor tested consists of three nodes, two computers, and the BIoTS. The computers can propose simple transactions such as submitting a humidity and temperature value, only for the BIoTS to validate them as miners in the network. However, the importance lies in the transaction proposed by the BIoTS. The computers will validate this by comparing the humidity and temperature values indicated by an internet application. Suppose the humidity and temperature value is in the right proximity range. In that case, the transaction is validated, and the functions of BIoTS as a miner in a blockchain network are satisfied.

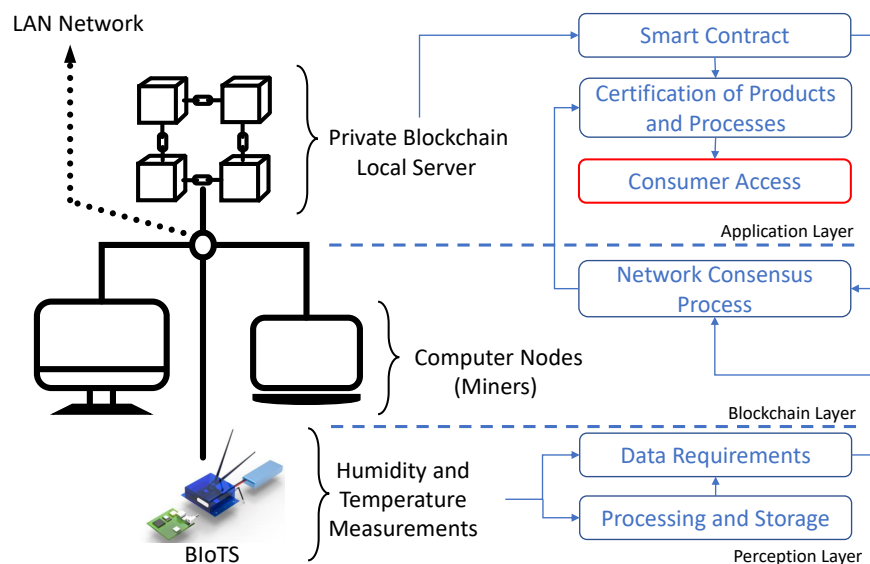


Figure 11. Evaluation scenario

The difficulty level setting of the consensus algorithm is low. We do not need to process a large number of network nodes for this demonstrative purpose, nor do we need to process a large data volume. However, we expect future work to demonstrate a Blockchain-IoT network configuration where a BIoTS sensors network interacts directly.

Some values related to data transmission from BIoTS to the blockchain designed for the use case are shown in Table 4. Each time, the cycle has an estimated amount of

transactions per second (tps) that the device and the network can support. In this case, the data size sent humidity and temperature information that grows in the stored data in the network's distributed database. Finally, there is a latency associated with each information transmitted and validation process for each transaction.

These data, as shown in Table 4, show the technical behavior of BIoTS in eight successive transactions. BIoTS proposed a block in the network eight times, with humidity and temperature data validated by the nodes of the network described in Figure 11. As we can see, as the size of the data sent increases, the latency in the transmission process increases. This behavior is attributed to the consensus algorithm's performance in this type's network.

Table 4. Evaluated parameters.

Step-Time (seconds)	Transaction Rate (tps)	Data Send (bytes)	Latency (seconds)
0.00	0	0	0
0.16	0.43	14	0.03
0.31	0.54	30	0.05
0.46	0.69	46	0.06
0.52	0.68	62	0.07
0.71	0.87	78	0.09
0.8	0.93	94	0.13
0.9	0.99	110	0.103

The graph in Figure 12 shows the performance of BIoTS in transmitting a data packet concerning the time it takes to propose a transaction on the Blockchain network. We observe that the time overhead in BIoTS transactions is one second; this is when it takes for the algorithm to encode and decode the accumulated data from seven humidity and temperature readings. The behavior of the transaction rate is linear, while the difficulty of the consensus algorithm grows exponentially. Directly, the size of the data packet sent in each transaction increases. This linear behavior may change to saturation lapses when BIoTS is subjected to the work of a blockchain network where multiple BIoTS nodes participate. However, at the same time, the network will work at the execution rate of the algorithms on FPGA. The transactions per second will surely increase, and the difference in network performance compared to a processor-based network can be determined.

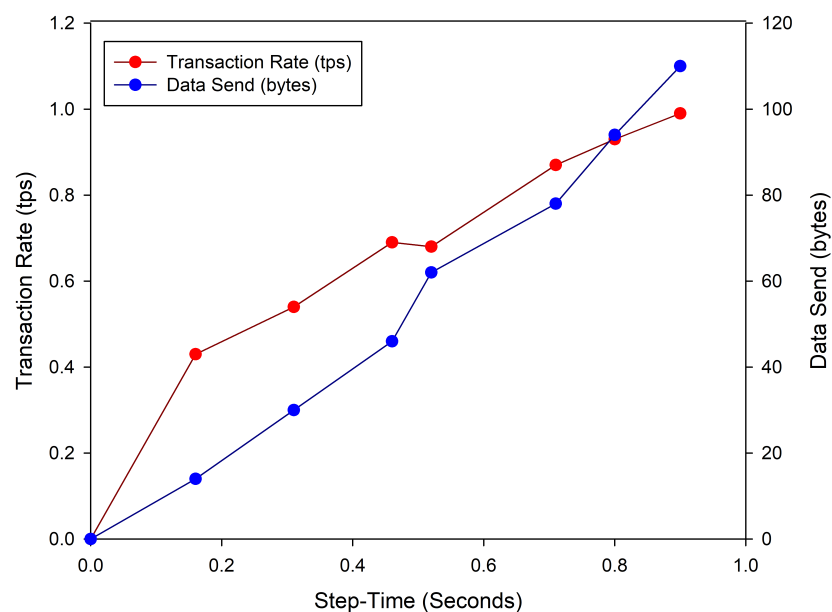


Figure 12. Transaction rate and data size sent by BIoTS.

The data shown in remix.ethereum.org (Figure 13) is the response to the programming of a Smart Contract in Solidity language; it is designed to read the humidity and temperature data of BIoTS every so often. These data are sent from BIoTS generating a new transaction, or if it is the case, another device proposes the transaction, and BIoTS can corroborate this information. The data shown in Figure 13 shows the behavior of BIoTS acting as a miner within the Blockchain-IoT network.

The screenshot displays the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar shows the account '0x5B3...eddC4', gas limit '3000000', and value '0 Wei'. The central code editor shows Solidity code for a 'BIoTSContract'. The right panel shows transaction details for a pending transaction, including the transaction hash, contract address, gas used, and transaction cost. Below the transaction details, there are fields for 'decoded input' and 'decoded output'. Two callout boxes are present: one pointing to the 'transaction cost' field and another pointing to the 'decoded input' and 'decoded output' fields.

Ethereum gas is a unit that measures the computational effort required to execute certain operations. Miners are paid an amount in Ether, which is the equivalent of the total amount of gas they had to use to execute a complete operation.

As we can see, BIoTS can generate encoded and decoded input and output data to operate as a Miner within a Blockchain network.

Figure 13. Transaction made by BIoTS in Blockchain Ethereum.

For the evaluation scenario, the transmitted humidity and temperature data from BIoTS are correctly encrypted and recorded in the blockchain network. Since the computational overhead increases as transactions (Measurements) and blocks grow, it is only possible to know the energy performance of BIoTS when hundreds or thousands of reads are accumulated. However, future work is expected to implement a BIoTS network and determine its performance as a Blockchain-IoT ecosystem.

Table 5 summarizes some of the IoT security issues that BIoTS has the potential to solve. As we can see, with this device, the resistance to specific security attacks is high and some moderate, considering the raids in a Blockchain-IoT network.

To identify the security flaws where BIoTS is a potential solution, we study a causality and effect correlation between the nature of the security attack in an IoT ecosystem and the Blockchain Hardware's architectural characteristics implemented in BIoTS [3,14,53]. The scale is weighted according to the architecture's characteristics. For example:

- "Sensor Tampering": the attack on BIoTS is unlikely because the sensor data is hosted in the SHA-256 and PoW hardware algorithms; after this process, the information is encrypted.
- "Sensor Feed Modification": this attack is possible with BIoTS; however, the resistance is high because the BIoTS firmware is almost null, and almost all elements are hardware.
- "Sybil Attack": this attack is unlikely in a network where BIoTS acts because it has the same blockchain network's resilience. However, it all depends on the network configuration (complexity level of PoW, etc.).

- “DoS, Protocol tunneling, and man-in-the-middle”: these attacks are unlikely due to the blockchain network’s nature. The communication channel by cryptography and the algorithms carried in hardware is immune to external intervention.
- “Jamming and Collisions”: these attacks are possible in a BIoTS network. The resistance to the attack is moderate because it can identify the hash’s inputs and outputs to reproduce copies.

Table 5. BIoTS-Security behavior.

Attack	Description	Attack likelihood	Resistance to Attack
Sensor Tampering	Manipulate sensors to acquire data readings	Unlikely	High
Sensor Feed Modification	Modify the sensor feed and firmware during communications process	Possible	High
Sybil Attack	Creates multiple identities and manipulates the device’s reputation.	Unlikely	High
DoS, Protocol tunneling; man-in-the-middle	Shut down a machine or network and The attacker sets up rogue hardware pretending to be a trusted network as Wi-Fi	Unlikely	High
Jamming, Collisions	Is an attempt to find two input strings of a hash function that produce the same hash result	Possible	Moderate

7. Conclusions

This paper proposes the hardware architecture of blockchain (Cryptography and Consensus Algorithms) to enrich an IoT device and provide it with remarkable capabilities to address security issues in the IoT ecosystem. This paper proposes a secure and trusted blockchain hardware architecture thanks to the reconfigurability of an FPGA. Our goal was to realize the implementation of Blockchain in hardware adapted to an IoT device to improve physical and virtual security mechanisms on devices in a network. This proof of concept works well and can positively impact the field of IoT-based food traceability systems. Finally, IoT devices immersed in ecosystems dedicated to the security and quality of supply chains are destined to be critical actors in products’ quality and safety through complete confidence in their performance in collecting, transmission, and storing information. This concept may be the future of primary IoT devices’ capabilities in food traceability systems or other application fields. The adaptation of emerging technologies in the IT security applied to various application fields opens a path to the evolution of Blockchain-IoT systems.

Author Contributions: C.G.-A., C.C.-G., G.R.-G., and J.C.C. proposed the concept of this research. M.M.-M. and C.C.-G. contributed for the state of art and final paper draft revisions. C.G.-A., G.R.-G., J.C.C., and C.C.-G. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Colciencias Doctoral scholarship 779-2017 for the Ph.D. in Telematic Engineering at the Universidad del Cauca, Popayán, Colombia and by the Universidad del Cauca (501100005682).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FTS	Food Traceability System
BioTS	Blockchain-IoT Sensor
SC	Supply Chain
PoW	Proof of Work
PoV	Proof of Vote
PoT	Proof of Trust
DBFT	Delegated Byzantine Fault Tolerance

References

1. Qiu, T.; Xiao, H.; Zhou, P. Framework and case studies of intelligence monitoring platform in facility agriculture ecosystem. In Proceedings of the 2013 Second International Conference on Agro-Geoinformatics (Agro-Geoinformatics), Fairfax, VA, USA, 12–16 August 2013; pp. 522–525. [\[CrossRef\]](#)
2. Singh, M.; Singh, A.; Kim, S. Blockchain: A game changer for securing IoT data. In Proceedings of the 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), Singapore, 5–8 February 2018; pp. 51–55. [\[CrossRef\]](#)
3. Dittmann, G.; Jelitto, J. A Blockchain Proxy for Lightweight IoT Devices. In Proceedings of the 2019 Crypto Valley Conference on Blockchain Technology (CVCBT), Rotkreuz, Switzerland, 24–26 June 2019; pp. 82–85. [\[CrossRef\]](#)
4. Niya, S.R.; Schiller, E.; Cepilov, I.; Maddaloni, F.; Aydinli, K.; Surbeck, T.; Bocek, T.; Stiller, B. Adaptation of Proof-of-Stake-based Blockchains for IoT Data Streams. In Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Seoul, Korea, 14–17 May 2019; pp. 15–16. [\[CrossRef\]](#)
5. Yu, S.; Lv, K.; Shao, Z.; Guo, Y.; Zou, J.; Zhang, B. A High Performance Blockchain Platform for Intelligent Devices. In Proceedings of the 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN), Shenzhen, China, 15–17 August 2018; pp. 260–261. [\[CrossRef\]](#)
6. Fakhri, D.; Mutijarsa, K. Secure IoT Communication using Blockchain Technology. In Proceedings of the 2018 International Symposium on Electronics and Smart Devices (ISESD), Bandung, Indonesia, 23–24 October 2018; pp. 1–6. [\[CrossRef\]](#)
7. Mezquita, Y. Internet of things platforms based on blockchain technology: A literature review. *Adv. Intell. Syst. Comput.* **2020**, *1004*, 205–208. [\[CrossRef\]](#)
8. International Trade Centre. *Traceability in Food and Agricultural Products*, 91st ed.; ITC: Geneva, Switzerland, 2015; Volume 91.
9. Suci, G.; Istrate, C.; Dițu, M. Secure smart agriculture monitoring technique through isolation. In Proceedings of the 2019 Global IoT Summit (GIoTS), Aarhus, Denmark, 17–21 June 2019; pp. 1–5. [\[CrossRef\]](#)
10. FuBing. Research on the agriculture intelligent system based on IOT. In Proceedings of the 2012 International Conference on Image Analysis and Signal Processing, Agadir, Morocco, 28–30 June 2012; pp. 1–4. [\[CrossRef\]](#)
11. Gomiero, T. Food quality assessment in organic vs. conventional agricultural produce: Findings and issues. *Appl. Soil Ecol.* **2018**, *123*, 714–728. [\[CrossRef\]](#)
12. Al, G.; Rhee, J.; Ahn, H.; Lee, J.; Farooq, U.; Fazal, M.; Syaekhoni, M.A. Integration of RFID, wireless sensor networks, and data mining in an e-pedigree food traceability system. *J. Food Eng.* **2017**, *212*, 65–75.
13. Tian, F. A Supply Chain Traceability System for Food Safety Based on HACCP, Blockchain & Internet of Things. In Proceedings of the International Conference on Services Systems and Services Management, ICSSSM, Dalian, China, 16–18 June 2017; pp. 1–6. [\[CrossRef\]](#)
14. Malik, S.; Dedeoglu, V.; Kanhere, S.S.; Jurdak, R. TrustChain: Trust Management in Blockchain and IoT Supported Supply Chains. In Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14–17 July 2019; pp. 184–193. [\[CrossRef\]](#)
15. Choudhary, T.; Virmani, C.; Juneja, D. Convergence of Blockchain and IoT: An Edge Over Technologies. *Stud. Comput. Intell.* **2020**, *846*, 299–316. [\[CrossRef\]](#)
16. Dabbagh, M.; Kakavand, M.; Tahir, M. Towards integration of blockchain and iot: A bibliometric analysis of state-of-the-art. *Adv. Intell. Syst. Comput.* **2020**, *1010*, 27–35. [\[CrossRef\]](#)
17. Mingxiao, D.; Xiaofeng, M.; Zhe, Z.; Xiangwei, W.; Qijun, C. A review on consensus algorithm of blockchain. In Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 5–8 October 2017; pp. 2567–2572. [\[CrossRef\]](#)
18. Khan, M.; Salah, K. IoT security: Review, blockchain solutions, and open challenges. *Future Gener. Comput. Syst.* **2018**, *82*, 395–411. [\[CrossRef\]](#)
19. Zhou, J.; Cao, Z.; Dong, X.; Vasilakos, A.V. Security and Privacy for Cloud-Based IoT: Challenges. *IEEE Commun. Mag.* **2017**, *55*, 26–33. [\[CrossRef\]](#)
20. Brachmann, M.; Keoh, S.L.; Morchon, O.G.; Kumar, S.S. End-to-End Transport Security in the IP-Based Internet of Things. In Proceedings of the 2012 21st International Conference on Computer Communications and Networks (ICCCN), Munich, Germany, 30 July–2 August 2012; pp. 1–5. [\[CrossRef\]](#)

21. Sethi, M.; Arkko, J.; Keränen, A. End-to-end security for sleepy smart object networks. In Proceedings of the 37th Annual IEEE Conference on Local Computer Networks-Workshops, Clearwater, FL, USA, 22–25 October 2012; pp. 964–972. [\[CrossRef\]](#)
22. Young, M.; Boutaba, R. Overcoming Adversaries in Sensor Networks: A Survey of Theoretical Models and Algorithmic Approaches for Tolerating Malicious Interference. *IEEE Commun. Surv. Tutor.* **2011**, *13*, 617–641. [\[CrossRef\]](#)
23. Noubir, G.; Lin, G. Low-power DoS Attacks in Data Wireless LANs and Countermeasures. *SIGMOBILE Mob. Comput. Commun. Rev.* **2003**, *7*, 29–30. [\[CrossRef\]](#)
24. Xu, W.; Wood, T.; Trappe, W.; Zhang, Y. Channel Surfing and Spatial Retreats: Defenses against Wireless Denial of Service. In *Proceedings of the 3rd ACM Workshop on Wireless Security, WiSe '04*; ACM: New York, NY, USA, 2004; pp. 80–89. [\[CrossRef\]](#)
25. Tsang, Y.P.; Choy, K.L.; Wu, C.H.; Ho, G.T.S.; Lam, H.Y. Blockchain-Driven IoT for Food Traceability with an Integrated Consensus Mechanism. *IEEE Access* **2019**, *7*, 129000–129017. [\[CrossRef\]](#)
26. Román, V.; Ordieres-Meré, J. WiP IoT Blockchain Technologies for Smart Sensors Based on Raspberry Pi. In Proceedings of the 2018 IEEE 11th Conference on Service-Oriented Computing and Applications (SOCA), Paris, France, 20–22 November 2018; pp. 216–220.
27. Urien, P. Blockchain IoT (BioT): A New Direction for Solving Internet of Things Security and Trust Issues. In Proceedings of the 2018 3rd Cloudification of the Internet of Things (CIoT), Paris, France, 20–22 November 2018; pp. 1–4. [\[CrossRef\]](#)
28. d. La Beaujardiere, J.; Mital, R.; Mital, R. Blockchain Application Within A Multi-Sensor Satellite Architecture. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 5293–5296.
29. She, W.; Liu, Q.; Tian, Z.; Chen, J.; Wang, B.; Liu, W. Blockchain Trust Model for Malicious Node Detection in Wireless Sensor Networks. *IEEE Access* **2019**, *7*, 38947–38956. [\[CrossRef\]](#)
30. Liu, Y.; Wei, B.; Du, Y.; Xiao, F.; Deng, Y. Identifying influential spreaders by weight degree centrality in complex networks. *Chaos Solitons Fractals* **2016**, *86*, 1–7. [\[CrossRef\]](#)
31. Zhao, G.; Guo, Y.; Sun, X.; Wang, X. A system for pesticide residues detection and agricultural products traceability based on acetylcholinesterase biosensor and internet of things. *Int. J. Electrochem. Sci.* **2015**, *10*, 3387–3399.
32. Zou, Z.; Chen, Q.; Uysal, I.; Zheng, L. Radio frequency identification enabled wireless sensing for intelligent food logistics. *Philos. Trans. R. Soc.* **2014**, *10*, 1237–1245.
33. Seo, S.M.; Kim, S.W.; Jeon, J.W.; Kim, J.H.; Kim, H.S.; Cho, J.H.; Lee, W.H.; Paek, S.H. Food contamination monitoring via internet of things, exemplified by using pocket-sized immunosensor as terminal unit. *Sens. Actuators B Chem.* **2016**, *233*, 148–156. [\[CrossRef\]](#)
34. Badia-Melis, R.; Ruiz-Garcia, L. 11—Real-Time Tracking and Remote Monitoring in Food Traceability. In *Advances in Food Traceability Techniques and Technologies*; Woodhead Publishing Series in Food Science, Technology and Nutrition; Woodhead Publishing: Cambridge, UK, 2016; pp. 209–224. [\[CrossRef\]](#)
35. Badia-Melis, R.; Carthy, U.M.; Ruiz-Garcia, L.; Garcia-Hierro, J.; Villalba, J.R. New trends in cold chain monitoring applications—A review. *Food Control* **2018**, *86*, 170–182. [\[CrossRef\]](#)
36. Chen, R.Y. Autonomous tracing system for backward design in food supply chain. *Food Control* **2015**, *51*, 70–84. [\[CrossRef\]](#)
37. Hu, T.; Zheng, M.; Zhu, L. Research Application of the Internet of Things Monitor Platform in Meat Processing Industry. In *Proceedings of the International Conference on Human-Centric Computing 2011 and Embedded and Multimedia Computing 2011*; Park, J., Jin, H., Liao, X., Zheng, R., Eds.; Springer: Dordrecht, The Netherlands, 2011; pp. 165–172.
38. Markovic, M.; Edwards, P.; Kollingbaum, M.; Rowe, A. Modelling Provenance of Sensor Data for Food Safety Compliance Checking. In *Provenance and Annotation of Data and Processes*; Mattoso, M., Glavic, B., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 134–145.
39. Tervonen, J. Experiment of the quality control of vegetable storage based on the Internet-of-Things. *Procedia Comput. Sci.* **2018**, *130*, 440–447. [\[CrossRef\]](#)
40. Hossain, M.M.; Fotouhi, M.; Hasan, R. Towards an Analysis of Security Issues, Challenges, and Open Problems in the Internet of Things. In Proceedings of the 2015 IEEE World Congress on Services, New York, NY, USA, 27 June–2 July 2015; pp. 21–28.
41. Shi, X.; An, X.; Zhao, Q.; Liu, H.; Xia, L.; Sun, X.; Guo, Y. State-of-the-Art Internet of Things in Protected Agriculture. *Sensors* **2019**, *19*, 1833. [\[CrossRef\]](#) [\[PubMed\]](#)
42. González-Amarillo, C.A.; Corrales-Muñoz, J.C.; Mendoza-Moreno, M.Á.; m. González Amarillo, A.; Hussein, A.F.; Arunkumar, N.; Ramirez-González, G. An IoT-Based Traceability System for Greenhouse Seedling Crops. *IEEE Access* **2018**, *6*, 67528–67535. [\[CrossRef\]](#)
43. Sidorov, M.; Khor, J.H.; Nhut, P.V.; Matsumoto, Y.; Ohmura, R. A Public Blockchain-Enabled Wireless LoRa Sensor Node for Easy Continuous Unattended Health Monitoring of Bolted Joints: Implementation and Evaluation. *IEEE Sens. J.* **2020**, *20*, 13057–13065. [\[CrossRef\]](#)
44. Hameed, S.; Shah, S.A.; Saeed, Q.S.; Siddiqui, S.; Ali, I.; Vedeshin, A.; Draheim, D. A Scalable Key and Trust Management Solution for IoT Sensors Using SDN and Blockchain Technology. *IEEE Sens. J.* **2021**. [\[CrossRef\]](#)
45. Alaba, F.A.; Othman, M.; Hashem, I.A.T.; Alotaibi, F. Internet of Things security: A survey. *J. Netw. Comput. Appl.* **2017**, *88*, 10–28. [\[CrossRef\]](#)
46. Jing, Q.; Vasilakos, A.V.; Wan, J.; Lu, J.; Qiu, D. Security of the Internet of Things: Perspectives and challenges. *Wirel. Netw.* **2014**, *20*, 1572–8196. [\[CrossRef\]](#)

47. Astill, J.; Dara, R.A.; Campbell, M.; Farber, J.M.; Fraser, E.D.; Sharif, S.; Yada, R.Y. Transparency in food supply chains: A review of enabling technology solutions. *Trends Food Sci. Technol.* **2019**, *91*, 240–247. [[CrossRef](#)]
48. Hong, W.; Cai, Y.; Yu, Z.; Yu, X. An Agri-product Traceability System Based on IoT and Blockchain Technology. In Proceedings of the 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN), Shenzhen, China, 15–17 August 2018; pp. 254–255. [[CrossRef](#)]
49. Danzi, P.; Kalør, A.E.; Stefanović, Č.; Popovski, P. Delay and Communication Tradeoffs for Blockchain Systems With Lightweight IoT Clients. *IEEE Internet Things J.* **2019**, *6*, 2354–2365. [[CrossRef](#)]
50. Kushch, S.; Prieto-Castrillo, F. Blockchain for Dynamic Nodes in a Smart City. In Proceedings of the 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 15–18 April 2019; pp. 29–34. [[CrossRef](#)]
51. Huh, S.; Cho, S.; Kim, S. Managing IoT devices using blockchain platform. In Proceedings of the 2017 19th International Conference on Advanced Communication Technology (ICACT), PyeongChang, Korea, 19–22 February 2017; pp. 464–467. [[CrossRef](#)]
52. Jemal, J.; Kornegay, K.T. Security Assessment of Blockchains in Heterogenous IoT Networks: Invited Presentation. In Proceedings of the 2019 53rd Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 20–22 March 2019; pp. 1–4. [[CrossRef](#)]
53. Xu, J.; Meng, X.; Liang, W.; Zhou, H.; Li, K.C. A secure mutual authentication scheme of blockchain-based in WBANs. *China Commun.* **2020**, *17*, 34–49. [[CrossRef](#)]
54. Tian, Y.; Wang, Z.; Xiong, J.; Ma, J. A Blockchain-Based Secure Key Management Scheme With Trustworthiness in DWSNs. *IEEE Trans. Ind. Inform.* **2020**, *16*, 6193–6202. [[CrossRef](#)]
55. Wood, D. Ethereum: A Secure Decentralised Generalised Transaction Ledger. Available online: <https://ethereum.github.io/yellowpaper/paper.pdf> (accessed on 26 April 2021).
56. Duong, L.V.T.; Thuy, N.T.T.; Khai, L.D. A fast approach for bitcoin blockchain cryptocurrency mining system. *Integration* **2020**, *74*, 107–114. [[CrossRef](#)]
57. Chaudhry, N.; Yousaf, M. Consensus Algorithms in Blockchain: Comparative Analysis, Challenges and Opportunities. In Proceedings of the 2018 12th International Conference on Open Source Systems and Technologies (ICOSST), Lahore, Pakistan, 19–21 December 2018; pp. 54–63. [[CrossRef](#)]
58. Wang, W.; Dinh Thai, H.; Hu, P.; Xiong, Z.; Niyato, D.; Wang, P.; Wen, Y.; Kim, D.I. A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks. *IEEE Access* **2019**, *30*, 22328–22370. [[CrossRef](#)]
59. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: <https://bitcoin.org/bitcoin.pdf> (accessed on 26 April 2021).
60. Ren, Z.; Cong, K.; Pouwelse, J.; Zekeriya, E. Implicit Consensus: Blockchain with Unbounded Throughput. *arXiv* **2017**, arXiv:1705.11046.
61. Li, K.; Li, H.; Hou, H.; Li, K.; Chen, Y. Proof of Vote: A High-Performance Consensus Protocol Based on Vote Mechanism Consortium Blockchain. In Proceedings of the 2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Bangkok, Thailand, 18–20 December 2017; pp. 466–473. [[CrossRef](#)]
62. Christodoulou, K.; Iosif, E.; Inglezakis, A.; Themistocleous, M. Consensus Crash Testing: Exploring Ripple’s Decentralization Degree in Adversarial Environments. *Future Internet* **2020**, *12*, 53. [[CrossRef](#)]
63. Jeon, S.; Doh, I.; Chae, K. RMBC: Randomized mesh blockchain using DBFT consensus algorithm. In Proceedings of the 2018 International Conference on Information Networking (ICOIN), Chiang Mai, Thailand, 10–12 January 2018; pp. 712–717. [[CrossRef](#)]
64. Zou, J.; Ye, B.; Qu, L.; Wang, Y.; Orgun, M.A.; Li, L. A Proof-of-Trust Consensus Protocol for Enhancing Accountability in Crowdsourcing Services. *IEEE Trans. Serv. Comput.* **2019**, *12*, 429–445. [[CrossRef](#)]
65. Ruiz-Rosero, J.; Ramirez-Gonzalez, G.; Khanna, R. Field Programmable Gate Array Applications—A Scientometric Review. *Computation* **2019**, *7*, 63. [[CrossRef](#)]