electronics

MDPI

Article

Blockchain-Based Authentication Scheme for Collaborative Traffic Light Systems Using Fog Computing

Sarra Namane [1], Marwa Ahmim [1], Aron Kondoro [2] and Imed Ben Dhaou [3,4,5,*]

[1] Networks and Systems Laboratory (LRS), Department of Computer Science, Badji Mokhtar-Annaba University, Annaba 23000, Algeria
[2] Department of Computer Science and Engineering (CSE), University of Dar-es-Salaam (UDSM), Dar es Salaam P.O. Box 33335, Tanzania
[3] Department of Computer Science, Hekma School of Engineering, Computing, and Informatics, Dar Al-Hekma University, Jeddah 22246-4872, Saudi Arabia
[4] Department of Computing, University of Turku, FI-20014 Turku, Finland
[5] Department of Technology, Higher Institute of Computer Sciences and Mathematics, University of Monastir, Monastir 5000, Tunisia
* Correspondence: imed.bendhaou@utu.fi

Abstract: In the era of the Fourth Industrial Revolution, cybercriminals are targeting critical infrastructures such as traffic light systems and smart grids. A major concern is the security of such systems, which can be broken down into a number of categories, such as the authentication of data collection devices, secure data transmission, and use of the data by authorized and authenticated parties. The majority of research studies in the literature have largely focused on data integrity and user authentication. So far, no published work has addressed the security of a traffic light system from data collection to data access. Furthermore, it is evident that the conventional cloud computing architecture is incapable of analyzing and managing the massive amount of generated data. As a result, the fog computing paradigm combined with blockchain technology may be the best way to ensure data privacy in a decentralized manner while reducing overheads, latency, and maintaining security. This paper presents a blockchain-based authentication scheme named VDAS using the fog computing paradigm. The formal and informal verifications of the proposed solution are presented. The evaluation of the proposed scheme VDAS showed that it has low communication and computation costs compared to existing lightweight authentication techniques.

Keywords: NTLS; fog computing; blockchain; authentication; sensor node; collaborative traffic control; AVISPA; informal verification

check for updates

Citation: Namane, S.; Ahmim, M.; Kondor, A.; Dhaou, I.B. Blockchain-Based Authentication Scheme for Collaborative Traffic Light Systems Using Fog Computing. *Electronics* **2023**, *12*, 431. https://doi.org/10.3390/electronics12020431

Academic Editor: Martin Reisslein

Received: 2 December 2022
Revised: 4 January 2023
Accepted: 11 January 2023
Published: 13 January 2023

1. Introduction

According to a study accomplished by the United Nations (UN), it is estimated that urbanization will continue to increase in the approaching decades. Approximately one billion people will live in cities by 2050. Megacities are also expected to grow steadily. The UN has estimated that by 2030, the number of mega-cities will settle at 43, leading urban sustainability to the forefront. However, it is necessary to take into account that poor city planning and inefficient transportation infrastructure are considered as major problems of urbanization for their negative impact on congestion and mobility in cities.

As a proposed solution, the use of Traffic Light Systems (TLSs) in intersections showed efficiency in reducing accidents and traffic congestion in urban areas, conforming to international traffic accident statistics. These systems encompass several traffic signals handled by a traffic controller. Traditional traffic light systems do not deliver sufficient real-time road traffic information which helps to reduce congestion in cities, greenhouse gas emissions, and fuel consumption for vehicles. Conversely, the advanced technology of communication and sensing technologies, including Wireless Sensor Networks (WSN), as well as the emergence of recent paradigms, namely machine learning, fog computing, and blockchain technology, are potential solutions for overcoming the limitations of the existing traffic light systems.

A modern traffic light system has three key layers: data collection, processing, and exploitation. The data sensing stage enables the fusion of traffic-related data from numerous sensors, which may be of diverse sorts, such as anisotropic magnetoresistive, acoustic, and optical sensors (cameras). The magnetoresistive sensor is a non-intrusive method that operates in many environmental conditions [1]. Additionally, it can be used to classify, count the number of vehicles, and determine the speed of moving cars. Moreover, it is affordable and simple to set up [2].

Sensing data is gathered and combined into a single format that is prepared for release to traffic-related apps for additional processing. The traffic light system uses the collected data to offer multiple services, namely, the prediction of traffic-related air pollution (TRAP), vehicle routing, and congestion prevention. Recent studies on the last item use deep reinforcement learning to grant emergency vehicles priority over other vehicles and machine learning techniques to predict traffic flow [3,4].

Cloud computing is typically used by the outdated traffic light system for data analysis and decision-making. In major cities, numerous traffic light controllers must cooperate and share traffic data in order to achieve network-wide objectives. A cloud-centric traffic light system creates a lot of traffic data that needs to be transferred from many locations, which increases network latency, exposes the data to security risks, and necessitates more energy. To overcome those limitations, a fog-based computer architecture was proposed in Ref. [5]. Despite the fact that fog-IoT integration consumes less energy and has lower latency than cloud-IoT integration, data-sensing devices, also known as end nodes, are unquestionably vulnerable to a range of security threats. For example, a hacker may utilize the sensor node and fog node of an intelligent traffic system to broadcast false information about the flow and density of the traffic. At significant intersections, malicious alteration of traffic data might potentially result in tragic collisions.

An effective method for tackling security concerns is access control, which includes the phases of authentication and authorization [6,7]. It is worth mentioning that a variety of recently published papers tackled user authentication in various IoT applications but did not address the severe ramifications of leveraging unauthenticated devices. A secure data-sensing phase will surely be ensured by the secure transfer of the generated data. In fact, blockchain technology might be a better choice for handling traffic light systems' initial stage security. Immutability, decentralization, robustness, and adaptability are some of the key attributes of the blockchain. Additionally, it resolves the single point of failure problem.

Few papers have focused on security issues in traffic light systems. In 2021, Ben Dhaou [1] presented a sensor node with IoT-enabled security for the management of the traffic light system. Indeed, in the proposed solution the author concentrated his efforts on designing the node using the Zigbee communication protocol, a magnetoresistive sensor, and a microcontroller. The node is responsible for reporting the level of service at each intersection providing useful information for traffic management authorities. In addition, Ben Dhaou managed the security using the Elliptic Curve Digital Signature Algorithm (ECDSA) to sign the data generated by a sensor in one intersection. However, ensuring a good level of security while reducing computational complexity and energy savings was not the priority of the author.

All of the aforementioned issues, as well as the relevance of security in a related application field, motivated the search for a solution that permits a secure collaboration between multiple traffic light systems scattered around a city. Because of the characteristics of the system, a lightweight Vehicle Detector Authentication Scheme (VDAS) was developed to provide secure communication between neighboring traffic light systems while also accounting for IoT resource constraints. Before traffic data can be gathered, a sensor must first be identified by the system, and a constant secure connection must be established between the sensor node and the traffic light controller. Consequently, a tampered-with or malicious sensor would not disturb the operation of the network traffic light system. The authentication system (VDAS) is also coupled with blockchain technology to make use of its decentralization feature and to solve the single point of failure issue. This work's main goal is to concurrently authenticate the sensor and the controller while ensuring the secure transmission of data in a constrained environment (processing power and memory size).

This paper represents an extension of the conference paper [1]. The main contributions of this paper are the following:

- Enhance the vehicle detection and counting algorithm to incorporate multiple sensors in various locations in the lane;
- Propose a blockchain-based Vehicle Detector Authentication Scheme (VDAS) in a Fog-based architecture for networked traffic light systems;
- Present formal and informal verifications of the proposed authentication strategy and validate the suggested scheme using simulation.

The paper is structured as follows. Section 2 presents the recent related work papers. Section 3 describes the proposed architecture while giving a brief description of blockchain technology and fog computing architecture. Section 4 presents the Vehicle-Detector Authentication Scheme (VDAS) for collaborative traffic light systems. Section 5 provides the formal and informal verification of the proposed scheme. The implementation details are given in Section 6. A discussion is presented in Section 7. Finally, Section 8 concludes the paper.

2. Related Work

The use of blockchain in intelligent transportation systems is a new area of study. Blockchain has been utilized in the Internet of Vehicle (IoV) to increase security (storage and communication) and to generate a value-added service, as detailed in Ref. [8]. A slew of access-control techniques based on blockchain technology have recently been developed to safeguard IoT devices and services [7].

A blockchain-based access control scheme in a smart grid environment was presented by Zhou et al. [9]. They used an identity-based combined encryption, signature, and signcryption scheme. Besides, the authors tried to solve the key escrow problem of the untrusted third party by designing a consensus algorithm in the power system. The performance evaluation of the proposed scheme showed a lower communication and computational costs compared to existing solutions. However, the authors did not present the formal and informal verification of the proposal.

Kumari et al. [10] discussed the performance evaluation among a traditional smart grid architecture, a smart grid with cloud computing architecture, and a smart grid with cloud computing and fog layer. The authors observed that the fog layer reduced the bandwidth while ensuring data protection. Furthermore, the proposed 5G-enabled three-tier architecture reduced the end-to-end latency.

Rodriguez et al. [11] analyzed and compared two existing authentication protocols developed for wireless sensor networks (WSNs). Then, they adjusted them for the use in unmanned aerial vehicles (UAV). The examination of the offered techniques revealed that the Drone to Ground Control Station (GCS) authentication required a longer average execution time due to the usage of expensive elliptic curve operations. The authors did not present the formal and informal verification of the proposed scheme.

Malani et al. [12] designed a certificate-based device access control scheme in an IoT environment preserving anonymity and security against several mentioned attacks. The authors used the AVISPA tools, the ROR model, and informal verification to demonstrate the security strength of the proposed scheme.

Ali et al. [13] analyzed the authentication scheme proposed in Ref. [14] to ensure protection against unauthorized drone access. The authors highlight the scalability issues of this scheme and its ability to work only in one environmental flying zone. In addition, Ali et al. discovered that the Srinivas et al. protocol is vulnerable to traceability and impersonation. To overcome these issues, the authors used symmetric encryption/decryption operations and lightweight hash to improve the previously cited scheme. Performance evaluation showed that the new protocol consumes similar computational time as the Srinivas et al. scheme and is strong against several attacks.

Bera et al. [15] designed a blockchain-based access control technique for the detection and mitigation of unauthorized unmanned aerial vehicles (UAV) in the Internet of Drones (IoD) environment. The authors presented formal security verification using the AVISPA tool and the Real-Or-Random (ROR) model. Furthermore, Bera et al. performed experiments on various cryptographic primitives under both server and Raspberry PI 3 configurations using the Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL). Finally, the authors compared the computation and communication overhead of their proposed solution to those of other well-known schemes.

A blockchain-based access control protocol in an IoT-enabled smart-grid system was presented by Bera et al. [16]. The formal and informal verification of the proposed DBACP-IoTSG showed security against multiple attacks.

Kumari et al. [17] proposed a blockchain-based Secure Energy Trading System (SETS) to store and process the data generated from smart meters (SMs). The authors evaluated the communication and computation costs of the proposed framework, it appears that the solution achieves good performance compared to Traditional Energy Trading System (TETS).

Khalid et al. [18] focused on power consumption and latency issues. They proposed a lightweight decentralized blockchain-based authentication mechanism for a smart hospital environment. The proposed scheme is based on a fog computing architecture while ensuring device-fog node authentication and device-device authentication. Moreover, the authors used blockchain technology to benefit from its decentralized nature and cryptographic features. The obtained evaluation results affirm that the use of fog architecture can reduce the time required to create and send an authentication request. However, Khalid et al. did not present a formal verification of the proposed scheme.

A fog computing architecture for multiple intersections was proposed by Hossan and Nower [5]. The main objective of this paper was to reduce vehicle waiting time. The evaluation of the proposed solution showed that their approach consumes the minimum quantity of fuel in different traffic densities and guarantees the lowest waiting time compared to other algorithms. However, the proposed solution neglected the security of such a system. It is obvious that the system is not secure against sensor impersonation attacks. For instance, the data generated by a sensor node can be altered easily by an attacker and ultimately threaten human lives.

A lightweight authentication and authorization framework was presented by Tahir et al. [19]. They used a probabilistic model for blockchain-enabled IoT networks. Tahir et al. used random numbers for the authentication phase, taking into account two types of IoT devices: homogeneous and heterogeneous. In addition, they focused on a fog computing architecture to overcome the limitations of the blockchain. The suggested method was examined by the authors using the AVISPA (Automated Validation of Internet Security Protocols and Applications) tool and the Cooja simulator. However, they did not present the informal verification of the proposed scheme.

Kumari et al. [20] proposed a decentralized peer-to-peer energy trading scheme using the Ethereum blockchain. The main purpose of this solution was to reduce the grid's energy generation while increasing the profit for both prosumers and consumers. The authors evaluated the proposed scheme in terms of data transfer rate, scalability, and storage cost. The obtained results showed that the solution can be considered as effective.

In 2021, Ben Dhaou [1] focused on the design of a secure sensor node using Zigbee as a low-power communication protocol, and a magnetoresistive sensor for the detection of moving or stopped vehicles. The integrity of the message issued by the sensor node is protected using ECDSA. However, access control has not been addressed.

Recently, the authors started to combine blockchain technology and fog computing architecture in IoT environments. In Ref. [19], Al Naji and Zagrouba presented a user authentication scheme for general IoT applications. The proposed mechanism was divided into three phases, namely registration, static authentication, and continuous authentication. The authors did not present a formal verification of the proposed scheme.

Altaf Haqani et al. [21] proposed mutual authentication among users and devices in smart home environments. The paper presented both the formal and informal verification of the proposed scheme. However, the solution is based on a cloud computing paradigm, leading to latency and bandwidth challenges. Adopting a fog computing-based architecture in smart home environments can be presented as a suitable solution to deal with the mentioned issues.

A comparative analysis of the related work is presented in Table 1 using several comparison criteria, namely: the Application Domain (AD), Blockchain (BC), Fog Computing (FC), the Authentication Type (AT), the Computation Cost (CC), the Communication Cost (MC), the Formal Verification (FV) and the Informal Verification (IV). According to Table 1, it is notable that only the paper of Ben Dhaou [1] took into consideration the traffic light systems security issue. All the remaining papers directed the focus in different application domains, for instance, smart grid environment [9,10,16,17,20], internet of drones [11,13,16], smart home environment [21], and general IoT environment [12,19]. By having a decentralization property that permits to face the single point of failure problem by avoiding the need for a trusted third party, blockchain technology can be used to resolve several issues. To illustrate , numerous solutions have used blockchain in a different manner, for instance, Refs. [9,15,16,21,22] used this technology combined with their proposed authentication protocols considering the constraint nature of tiny devices, namely sensors, actuators, and smart meters that do not support costly blockchain computation. Furthermore, Refs. [17,20] proposed a blockchain-based energy management schemes in a smart grid environment. The fog computing paradigm permits to make data storage and computation more adjacent to data gathering devices, reducing the data processing cost and the network latency. According to Table 1, only Refs. [10,22] proposed a fog computing architecture. Regarding the Authentication Type (AT), it can be classified in the following categories according to the system architecture entities: user–device authentication [13,21,22], user–server authentication [9,15–17,20], and device–device authentication [11,12,15]. Multiple papers have evaluated the Computation Cost (CC) and the Communication Cost (MC) [9,11,12,15–17], whereas Refs. [21,22] solely considered the calculation cost, which is the time spent managing the authentication request. The security level of an authentication scheme can be evaluated using Formal Verification (FV) through different known tools, namely AVISPA, Scyther, and ProVerif. The two types of verification were managed in Refs. [12,13,15,16,21] while Ref. [22], presented only the Informal Verification (IV).

Table 1. Comparison of related works.

References	AD	BC	FC	AT	CC	MC	FV	IV
[1]	Traffic light systems	X	X	Node authentication	X	X	X	X
[9]	Smart grid environment	✓	X	User and power provider mutual authentication	✓	✓	X	X
[10]	Smart grid environment	X	✓	X	X	X	X	X
[11]	Unmanned Aerial Vehicles	X	X	Mutual UAV authentication	✓	✓	X	X
[12]	IoT environment	X	X	Device to device authentication	✓	✓	✓	✓
[13]	Internet of Drones	X	X	Users and drones authentication	✓	✓	✓	✓
[15]	Internet of Drones	✓	X	Drone to drone and drone to GSS authentications	✓	✓	✓	✓
[16]	Smart-grid system	✓	X	Smart meter and service provider mutual authentication	✓	✓	✓	✓
[17]	Smart grid system	✓	X	Consumers and producers authentication	✓	✓	X	X
[19]	General IoT applications	✓	✓	User authentication	✓	X	X	✓
[20]	Smart grid	✓	X	Prosumers and consumers authentication	X	X	X	X
[21]	Smart home environments	✓	X	User–device authentication	✓	X	✓	✓
Our scheme	Traffic light systems	✓	✓	Sensor authentication	✓	✓	✓	✓

X: Not supported; ✓: supported.

Thus, many papers have proposed to guarantee security in different IoT environments, and the introduction of blockchain technology permits them to solve the single point of failure issue. However, the proposed solutions did not manage all the comparison criteria cited in Table 1. In this paper, we propose a blockchain-based Vehicle Detector Authentication Scheme (VDAS). The solution is based on three layers of fog computing architecture. The combination of blockchain technology with fog computing ensures a decentralized authentication while reducing network latency. Furthermore, the proposed VDAS has lower computation and communication costs compared to the existing schemes.

3. The Proposed Architecture

After a thorough analysis of the related work, it is noticeable that the security issue of a network traffic light system (NTLS) has been neglected by recent researches and the collaboration between several traffic light systems of different regions in a city is required. Furthermore, regardless of the device used for data detection, it is vital to ensure the device's authenticity while guaranteeing that only authorized participants have access to the transmitted data. Conventionally, a traditional NTLS is connected to cloud computing services to store important data and make decisions. To provide a collaborative traffic light management system, a large quantity of data has to be transmitted from various locations in the city. For this reason, it is safe to affirm that the use of the cloud computing paradigm may be responsible for causing the latency and overhead challenges. Fog computing architecture can be used to overcome the issues mentioned previously. Moreover, blockchain technology, with its decentralized nature and cryptographic features, allows data to be stored securely and avoids the need for a third party. In this section, a brief description of blockchain technology and fog computing architecture is presented. Then, we will describe the proposed architecture that gives the role of each participant.

3.1. An Overview of Blockchain Technology

Blockchain technology offers the possibility to keep data in a distributed ledger, allowing users to read and record data in the ledger using transactions, but does not authorize data modification and deletion. This section incorporates a definition of some important terminologies related to blockchain technology. A brief description of how it works is provided as follows.

3.1.1. Annotations Related to Blockchain Technology

In this subsection, there is a definition of the prominent terms related to blockchain technology.

- Transaction: a term used to define an exchange between two parties;
- Node refers to any member of the blockchain network. The type of electronic device that maintains copies of the blockchain is nondescript. Each node has an address, manifested under the form of a string of alphanumeric characters to identify it;
- Blockchain: a chain of blocks responsible for storing information in a specific type of database and and keeping a record of each transaction carried across the network;
- Block: a data structure that contains all the necessary metadata concerned with the block header and related transactions. The first block in a blockchain is known as a genesis block, it represents a special case considering that it does not reference a previous block;
- Distributed ledger: is a ledger maintained on many nodes in the network having the function of organizing these nodes into chronological order. This ledger can be of two types: permissioned and unpermissioned;
- Smart contract: a code that gives details on the permissions and the sequence of events to manage and change the state of the ledger;
- Cryptographic hash function: is a function that seizes a random input of data (keys) and provides a string of bytes with stable length and structure (hash value);

- Consensus algorithm: an algorithm that allows all nodes of the network to agree on the shared state of the ledger. Several consensus algorithms were developed, the first one was called Proof of Work (POW) and it requires a lot of processing power. Then, a Proof of Stake (POS) was proposed based on the amount of funds on the network. The most popular consensus algorithms are as follows: Distributed Proof of Stake (DPOS), Proof Of Authority (PoA), Byzantine Fault Tolerance (BFT), Practical Byzantine Fault Tolerance (PBFT), and Delegated Byzantine Fault Tolerance (dBFT).

3.1.2. The Functionality of Blockchain Technology

From the previous definitions, it is obvious that the blockchain concept refers to storing data digitally in a secure way. On a blockchain network, nodes can exchange data using transactions. After the authenticity of these transactions is verified, a block is created. Moreover, adding this new block to the main blockchain is executed using the consensus algorithm. The main idea of this algorithm is to solve a difficult mathematical puzzle. Furthermore, regardless of the type of consensus algorithm used, it requires great computational power. The resolution of the mathematical problem means that, a hash value of the concerning block is generated. Then, the node that solved the mathematical problem is rewarded in the form of cryptocurrency. Each block has a number and a timestamp that refers to the order in which it is attached to the chain. Moreover, the hash value of each block is added to the following block. This hash value acts as a digital block signature and guarantees an extremely secure blockchain.

3.2. An Overview of the Fog Computing Paradigm

In an IoT environment, several devices collect a large amount of data that need to be treated. However, these devices, namely sensors, actuators, and trackers, are known to have reduced computational and storage capacities. Transferring the processing of these data to cloud systems with high capabilities is regarded to be an adequate solution. However, it causes long latency and security issues. The fog computing paradigm introduced a new layer, known as the fog layer, which found to be well situated between IoT devices and the cloud computing layer. The main role of the fog layer is to combine the available storage, computing, and network resources at the edge of the network to provide more efficient services [23].

In Ref. [24], the authors devised a fog-based traffic congestion monitoring system as well as a cloud-enabled traffic congestion monitoring system. A comparative study using different data sets was conducted. The results revealed that the fog computing architecture has significant benefits over the cloud computing architecture in terms of high bandwidth and low latency. The response time and bandwidth of the fog network are five times more efficient than those of the cloud. The integration of fog and cloud computing paradigms in intelligent traffic monitoring permits to overcome the drawbacks of each technology while benefiting from the advantages of each one [25].

3.3. A Detailed Description of the Proposed Architecture

A city can be divided into several regions that encompass multiple intersections managed particularly by a traffic light controller and having numerous lanes (illustrated in Figure 1).

To reduce traffic congestion, a collaboration between multiple traffic light systems belonging to the same region or to different regions is required. This paper proposes a collaborative secure networked traffic light architecture based on blockchain technology. The proposed architecture (described in Figure 2) is composed of three layers, namely the sensors layer, the fog layer and the cloud layer.

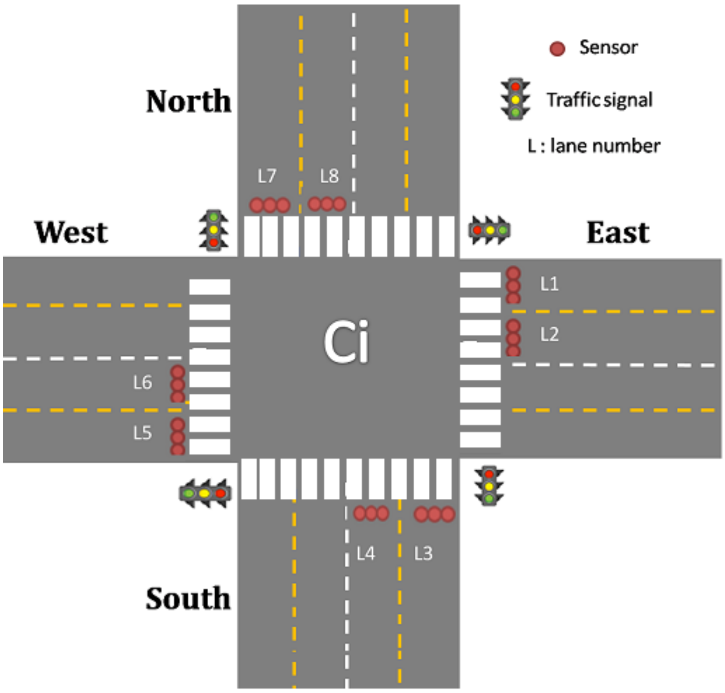


Figure 1. The intersection of a city region.

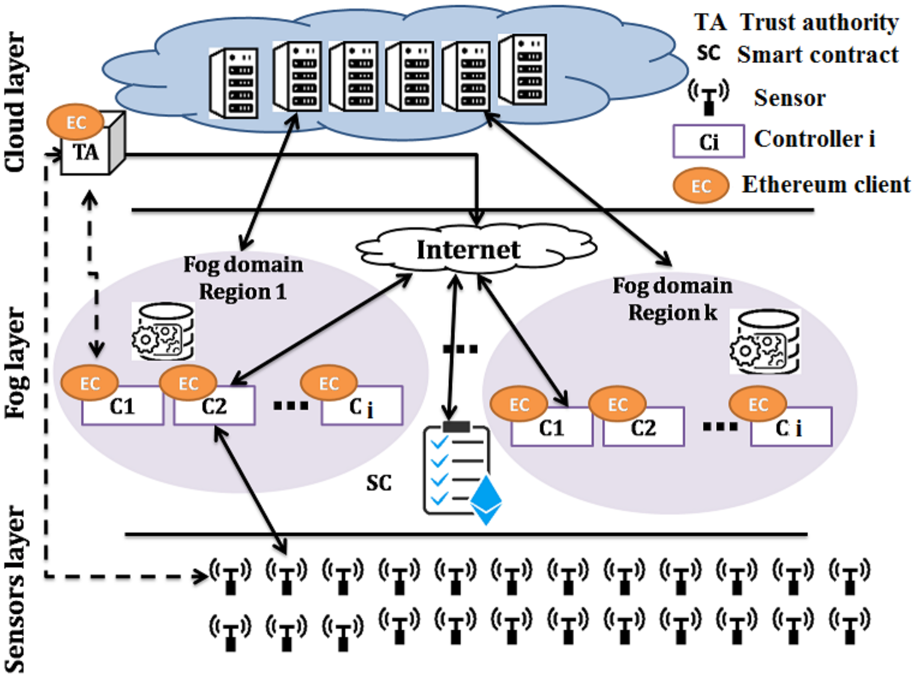


Figure 2. The proposed architecture.

3.3.1. The Sensors Layer

This layer encompasses several sensors buried in groups of three or five at different levels of the road pavement to detect and count vehicle numbers. Furthermore, the sensor is known as a constrained device with limited computational and storage capacities. It is considered to be an embedded system consisting of [1] a radio transceiver, a magnetoresistive sensor, a communication module, and a microcontroller.

The magnetoresistive sensor sends the magnetic field intensity (B_x, B_y, B_z) to the microcontroller. Then, the magnetic field is compared to a defined threshold τ that specifies if a vehicle is passing over the sensor (magnetic field larger than τ) or stopped over the sensor. A comparison between the rise time (T_r) and the fall time (T_f) is performed to determine the status of the vehicle against the sensor. In case the rise time (T_r) is found to be smaller than the fall time (T_f) by 10 s, a vehicle is detected in stop mode. When the sensor detects a vehicle in stop mode, it calculates the number of the vehicle stopped before it (presented in Algorithm 1) using the level of the sensor in the lane, the distance between two groups of sensors and the average length of a vehicle (illustrated in Figure 3).

The following algorithm shows how the sensor detects and counts the vehicle number. Sending the latter to the controller to which it belongs is a compulsory procedure. It is obvious that a fake sensor can intercept the data and change the value. Besides, the identity of a sensor can be stolen by an attacker to send erroneous information. This lack of security can have serious consequences and cost lives. For this reason an authentication protocol is required in order to guarantee the integrity of the shared data as well as the identity of the sensor.

Algorithm 1 Vehicle detection and counting algorithm.

```
procedure VDAS($B_x, B_y, B_z, \tau, Sensor_{Level}, D, L$)
    $d_x \leftarrow 0$
    $X \leftarrow \sqrt{B_x^2 + B_y^2 + B_z^2}$
    $Number_V \leftarrow 0$
    $T_r \leftarrow 0$
    $T_f \leftarrow 0$
    $T_{wait} \leftarrow 0$
    if $X \succ \tau$ AND $d_x = 0$ then
        $d_x \leftarrow 1$
        $T_r \leftarrow Time$
    else
        if $X \prec \tau$ AND $d_x = 1$ then
            $T_f \leftarrow Time$
        end if
    end if
    if $T_f \succ T_r + 10$ then
        $T_{wait} \leftarrow T_f - T_r$
        $Status_v \leftarrow Stop$
    else
        if $T_f = 0$ then
            $Status_v \leftarrow NoVehicle$
        else
            $Status_v \leftarrow Passing$
        end if
    end if
    if $Status_v = Stop$ AND $Sensor_{Level} = 0$ then
        $Number_V \leftarrow 1$
    else
        if $Status_v = Stop$ then
            $Number_V \leftarrow \frac{Sensor_{Level} * D}{L}$
        end if
    end if
end procedure
```

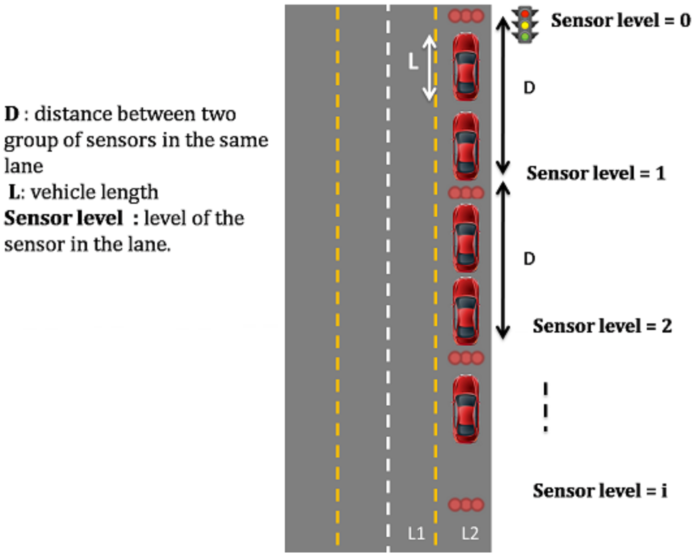

D : distance between two group of sensors in the same lane
L: vehicle length
Sensor level : level of the sensor in the lane.
L
D
D
Sensor level = 0
Sensor level = 1
Sensor level = 2
Sensor level = i
L1
L2

Figure 3. The used parameters to count vehicle number.

3.3.2. The Fog Layer

The fog layer is composed of several fog domains. Each domain encompasses the controllers that manage the traffic lights of one city region. Each controller is an Ethereum client that has an Ethereum address and a private key. Furthermore, each controller can run a common smart contract on the blockchain. This contract was created at the beginning by the trust authority of the system. In addition, all the functions of registration, authentication, and authorization are managed in a decentralized way by the smart contract. The employment of the Ethereum blockchain in this layer ensures the controller's authentication. Moreover, the decentralized nature of the access control technique deals with the Single Point of Failure Problem (SPFP).

3.3.3. The Cloud Layer

The cloud layer manages data processing at the city level. Analysis of data collected from IoT sensors is considered to be a suitable approach to offer valuable services, including comfort during travel, reduced travel time, and short travel routes.

4. Blockchain-Based Authentication Scheme for Collaborative Traffic Light Systems

We designed a novel blockchain-based authentication scheme for a collaborative traffic light management system. In short, this protocol is called a Vehicle Detector Authentication Scheme (VDAS), it permits the authentication of the sensor nodes that detect vehicles and count their number. The proposed VDAS consists of the following phases: the initialization and registration phase and the authentication phase. All parameters used in the protocol are listed in Table 2.

Table 2. Symbol description.

Symbol	Description
TA	Trust authority
ID_s	Identity of sensor S
ID_c	Identity of controller C
T_s	Timestamp
BS_1, b	Random numbers generated by sensor
FCR_1, FCR_2	Random numbers generated by controller
H()	One-way hash function
| |	Concatenation operation
P	A point of the elliptic curve
K_{SC}	Key sensor controller
$\{\}_{K_{SC}}$	AES encryption using the key K_{SC}

4.1. Initialization and Registration Phase of VDAS

In this section, we present a detailed description of our system model that substitutes four entities as follows: a controller, a sensor, a blockchain, and a trusted authority (TA). During this phase, the trust authority, also referred to as an Ethereum client, creates the authentication smart contract. The latter encompasses two main functions and other secondary functions that help to achieve authentication in a more efficient manner. The first function attributes each controller ID_c to its corresponding sensors. Each controller represents an Ethereum client with an Ethereum address and its corresponding private key, allowing the signature of the transactions generated by each controller. The main role of this key is to authenticate the controller, and simultaneously sending a transaction to invoke a function in the smart contract. The smart contract function calls can be of two types: call and transaction. The first type represents a local invocation of a contract function that does not broadcast or publish anything on the blockchain. However, the second type broadcasts a signed transaction to the network. This transaction is processed by miners and, if valid, is published on the blockchain. The second main function of the smart contract manages the sensor authentication request. Its essential goal is to calculate certain parameters that allow us to authenticate the sensor.

During the sensor registration phase, the TA provides a smart card to the sensor node containing the identity of the controller to which it belongs. Further, each controller has enough computing power to authenticate the sensor nodes within its coverage. After the sensor registration phase, the controller authenticates the sensor node to send real-time traffic information.

4.2. Authentication Phase of VDAS

During the authentication phase, the sensor node generates two random numbers: $BS_1, b \in [1, n-1]$ and a timestamp T_s. Then, it calculates $SC_1 = H(ID_s \parallel BS_1)$.P. The sensor sends its ID_s, the calculated SC_1, and T_s to the controller to which it belongs (the ID_c of the controller provided by the trust authority during the registration phase on the smart card).

Upon receiving the sensor message, the controller sends a transaction to the smart contract authentication. This transaction is signed with the controller's private key. First, the smart contract will check if the sensor ID_s belongs to the controller ID_c. If the sensor belongs, the controller will call another function to generate two random numbers $F_{CR1}, F_{CR2} \in [1, n-1]$. Then, it calculates the following parameters:

$$SC_{c1} = H(F_{CR1} \parallel ID_c).P$$
$$\beta = SC_1.H(F_{CR1} \parallel ID_c)$$
$$SC_{c2} = F_{CR2}.\mathrm{P}$$
$$C_{cr} = H(ID_c \parallel ID_s \parallel X_{SC1} \parallel SC_{c2})$$
$$A_{cr} = F_{CR2} + C_{cr}(H(F_{CR1} \parallel ID_c))$$
$$K_{SC} = H(X_{ID_s} \parallel X_\beta \parallel X_{SC_{c2}})$$

The controller sends to the sensor node SC_{c1} and the encryption of ID_c, C_{cr}, and A_{cr} using the session key β. Upon receiving the controller message, the sensor starts by calculating the key β as :

$$\beta = SC_{c1}.H(ID_s \parallel B_{S1}), \text{ then it calculates}$$
$$C_{cr}' = H(ID_c \parallel ID_s \parallel X_{SC_1} \parallel A_{cr}.\mathrm{P} \text{ - } C_{cr}.H(F_{CR1} \parallel ID_c).\mathrm{P})$$
$$\text{if } C_{cr}' = C_{cr}$$
$$\text{the sensor node calculates } B_{S2} = \mathrm{b.P}$$
$$C_S = H(ID_c \parallel ID_s \parallel X_{SC_1} \parallel B_{S2})$$
$$A_S = \mathrm{b} + C_S(H(ID_s \parallel B_{S1})$$

Then it sends to the controller the encryption of C_S and A_S using K_{SC}. The controller calculates C_S' as: $C_S' = H(ID_c \parallel ID_s \parallel X_{SC_1} \parallel A_S.\mathrm{P} \text{ - } C_S.H(ID_s \parallel B_{S1}).\mathrm{P})$.
if the $C_S' = C_S$ then the controller ID_c authenticated the sensor ID_s.

After the authentication phase, the sensor will use K_{SC2} to encrypt the number of vehicles that it detected. K_{SC2} is calculated as: $K_{SC2} = H(X_{ID_c} \parallel X_{ID_s} \parallel X_\beta \parallel X_{SCc2} \parallel X_{BS2})$

Upon receiving the number of vehicles, the controller decrypts this message using the same key. The obtained value will be stored on the blockchain using a transaction signed by the controller. This value can be used by the controllers of adjacent intersections to optimize road traffic and reduce congestion.

Figure 4 gives a summary of the authentication phase of VDAS.

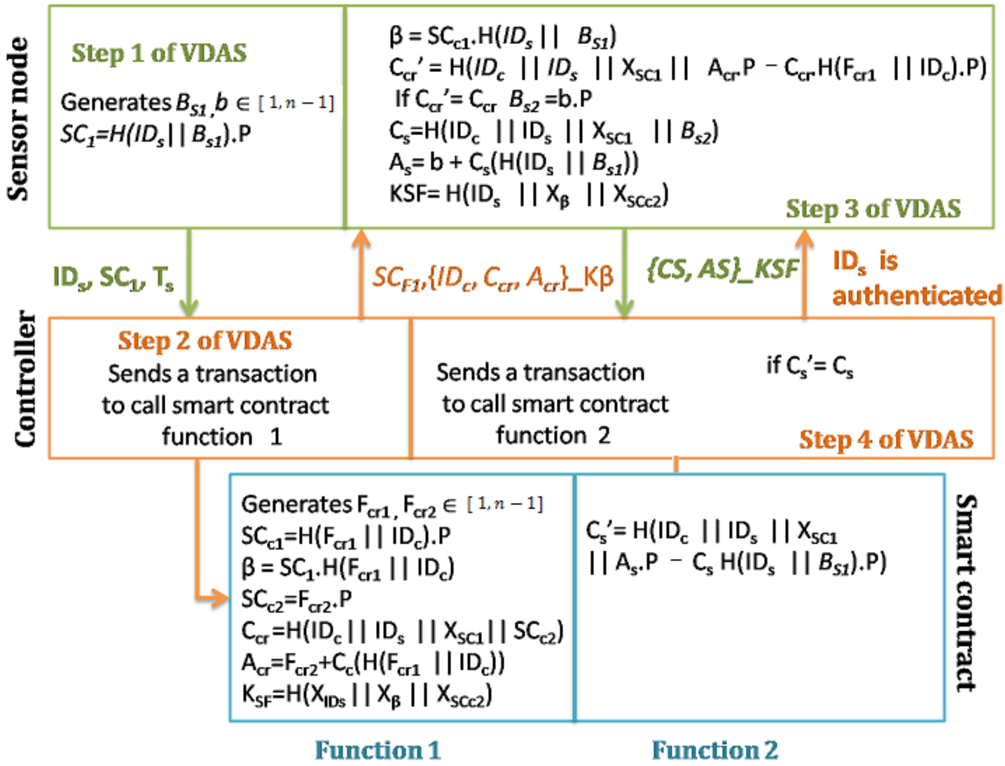

Sensor node
Step 1 of VDAS
Generates B_{S1}, $b \in [1, n-1]$
$SC_1=H(ID_s || B_{s1}).P$
$\beta = SC_{c1}.H(ID_s || B_{S1})$
$C_{cr}' = H(ID_c || ID_s || X_{SC1} || A_{cr}.P - C_{cr}.H(F_{cr1} || ID_c).P)$
If $C_{cr}' = C_{cr}$ $B_{s2} = b.P$
$C_s=H(ID_c || ID_s || X_{SC1} || B_{s2})$
$A_s = b + C_s(H(ID_s || B_{s1}))$
$KSF = H(ID_s || X_\beta || X_{SCc2})$
Step 3 of VDAS
ID_s, SC_1, T_s
$SC_{F1},\{ID_c, C_{cr}, A_{cr}\}\_K\beta$
$\{CS, AS\}\_KSF$
ID_s is authenticated
Controller
Step 2 of VDAS
Sends a transaction to call smart contract function 1
Sends a transaction to call smart contract function 2
if $C_s' = C_s$
Step 4 of VDAS
Generates F_{cr1}, $F_{cr2} \in [1, n-1]$
$SC_{c1}=H(F_{cr1} || ID_c).P$
$\beta = SC_1.H(F_{cr1} || ID_c)$
$SC_{c2}=F_{cr2}.P$
$C_{cr}=H(ID_c || ID_s || X_{SC1} || SC_{c2})$
$A_{cr}=F_{cr2}+C_c(H(F_{cr1} || ID_c))$
$K_{SF}=H(X_{IDs} || X_\beta || X_{SCc2})$
$C_s'= H(ID_c || ID_s || X_{SC1} || A_s.P - C_s H(ID_s || B_{S1}).P)$
Smart contract
Function 1
Function 2

Figure 4. Steps of the VDAS authentication phase.

Figure 5 presents a sequence diagram of the proposed Vehicle Detector Authentication Scheme (VDAS). This diagram summarizes the entire protocol. It begins with the registration phase carried out by the trusted authority. Then follows the authentication step, where each of the actors (sensor node, controller) performs the calculation of its own parameters. The controller uses the smart contract to perform these calculations. Finally, the sensor node is authenticated if the calculated parameters on each side are equal.

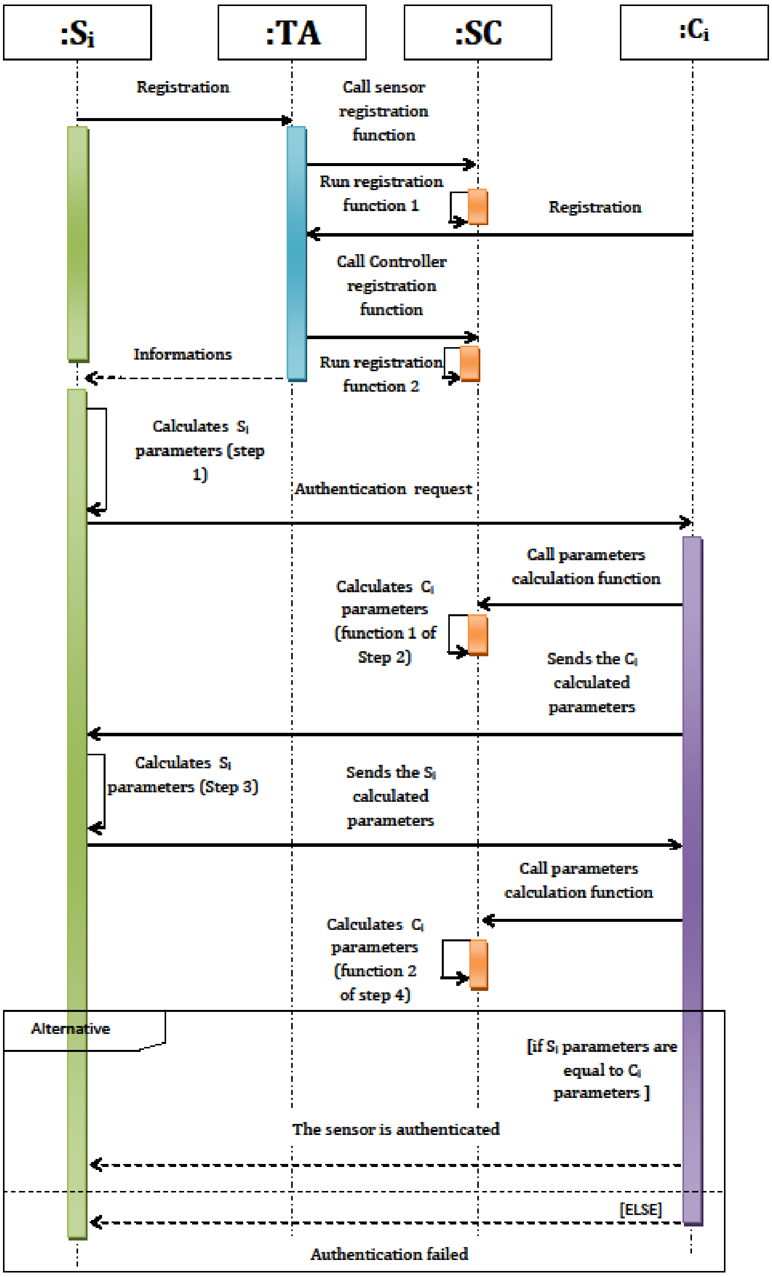

$:S_i$
:TA
:SC
$:C_i$
Registration
Call sensor registration function
Run registration function 1
Registration
Call Controller registration function
Informations
Run registration function 2
Calculates S_i parameters (step 1)
Authentication request
Call parameters calculation function
Calculates C_i parameters (function 1 of Step 2)
Sends the C_i calculated parameters
Calculates S_i parameters (Step 3)
Sends the S_i calculated parameters
Call parameters calculation function
Calculates C_i parameters (function 2 of step 4)
Alternative
[if S_i parameters are equal to C_i parameters]
The sensor is authenticated
[ELSE]
Authentication failed

Figure 5. Sequence diagram of the proposed Vehicle Detector Authentication Scheme.

5. Formal and Informal Verification of the Proposed Vehicle Detector Authentication Scheme

In this section, the formal and informal verifications of the Vehicle Detector Authentication Scheme (VDAS) are presented.

5.1. Informal Verification of the Proposed VDAS

Through the following informal analysis, we also demonstrate that the VDAS can resist various attacks.

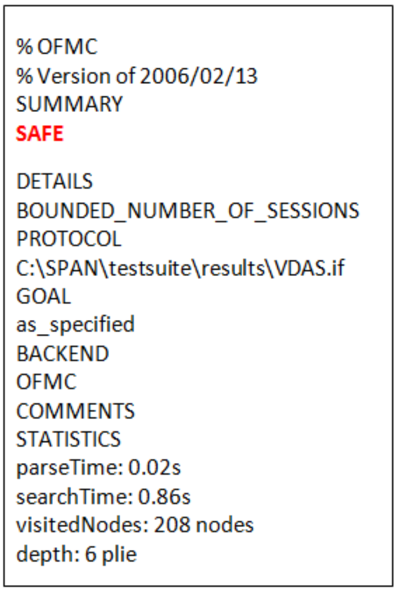
- Perfect Forward Secrecy: Confidentiality in earlier communications is not affected by an opponent learning the key to a recent session. In VDAS, the session key is derived from random numbers (BS_1, b, F_{cr1}, F_{cr2}). Therefore, the VDAS satisfies the PFS property;

- Replay attack: The adversary cannot assume the identity of the sensor or controller because a new random number is generated for each session to provide mutual authentication. Furthermore, our authentication scheme directly recognizes the replay attack because it uses a timestamp;
- Man in the middle attack: the adversary is watching on the communication line. He can change the authentication request on his own. However, the man in the middle attack cannot succeed due to the check-in the second message (the calculation of C_{cr}) and in the third message (the calculation of C_s);
- Side channel attack: VDAS is based on ECC and the elliptic curve discrete logarithm problem (ECDLP). Because of this, the side-channel attack can be recognized by our authentication scheme;
- Modification attack. The use of hash functions in our authentication scheme ensures integrity property;
- Control-key: it is not possible to present the session key shared between the sensor and the controller with a predefined value in VDAS;
- Spoofing attack: due to the verification in the second (the calculation of C_{cr}) and third messages(the calculation of C_s), this attack cannot succeed in VDAS.

5.2. Formal Verification of the Proposed VDAS

This subsection presents a formal verification of the Vehicle Detector Authentication Scheme (VDAS) using the most widely used Automated Validation of Internet Security Protocols and Applications(AVISPA) tool [26]. AVISPA represents an expressive and modular formal language. It permits specifying and analyzing protocols with their security properties. Besides, it supports cryptographical operations from which hash function, and encryption/decryption.

Figure 6 shows that the obtained outcomes of the VDAS scheme are "SAFE" simulated with OFMC back-ends. Besides, the back-end OFMC generates "SAFE" outputs following visiting 208 nodes with a total depth of 6 plies in 0.02 s parse-time and 0.86 s search-time, respectively.



```
% OFMC
% Version of 2006/02/13
SUMMARY
SAFE

DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
C:\SPAN\testsuite\results\VDAS.if
GOAL
as_specified
BACKEND
OFMC
COMMENTS
STATISTICS
parseTime: 0.02s
searchTime: 0.86s
visitedNodes: 208 nodes
depth: 6 plie
```

Figure 6. The simulation results of VDAS.

5.3. Comparative Study

In this subsection, we will use the experimental results of the cryptographic primitives presented in Ref. [15] to calculate the communication and computation costs of the proposed Vehicle Detector Authentication Scheme (VDAS). Bera et al. used MIRACL [27] to perform the cryptographic operations on a Raspberry PI 3 B+ Rev 1.3, 1.4-GHz Quad-core processor, core 4, Ubuntu 20.04 LTS, 64 bits operating system, 1-GB RAM [28]. They executed each primitive for 100 runs to calculate the average execution time for each primitive in milliseconds. Table 3 describes the obtained results.

Table 3. Execution time (in ms) under Raspberry PI 3 [15].

Primitive	Max. Time (ms)	Min. Time (ms)	Average Time (ms)
T_h	0.643	0.274	0.309
T_{mtp}	0.406	0.381	0.385
T_{exp}	0.071	0.037	0.039
T_{ecsigg}	5.175	2.480	2.597
T_{ecsigv}	9.728	4.701	4.901
T_{senc}	0.038	0.017	0.018
T_{sdec}	0.054	0.009	0.014
T_{ecm}	4.532	2.206	2.288
T_{eca}	0.021	0.015	0.016
T_{bp}	32.79	27.606	32.084

The symbols T_h, T_{mtp}, T_{ecsigg}, T_{ecsigv}, T_{senc}, T_{sdec}, T_{ecm}, T_{eca} and T_{bp} are used to denote the time required for "one-way hash function using SHA-256 hashing algorithm", "map to elliptic curve point", "elliptic curve encryption/decryption", "symmetric key encryption/decryption (AES-128)", "elliptic curve point multiplication", "elliptic curve point addition", and "bi-linear pairing", respectively.

In this section, we provide a detailed comparative analysis of the computation and communication costs of a sensor node compared to the costs of tiny devices of other relevant existing competing schemes, such as the schemes of Zhou et al. [9], Rodrigues et al. [11], Malani et al. [12], Ali et al. [13], and Bera et al. [15,16]. The communication computation costs represent the main comparison criteria.

5.3.1. Computation Cost Evaluation

According to the experimental results reported in Table 2, a sensor node S_i requires a computation cost of $3T_h + 2T_{ecm} + T_{enc} + T_{dec}$ = 5.535 ms. Table 4 shows a detailed comparative study on computation costs among the proposed VDAS and other schemes. It is observed that the necessary computational cost for the proposed VDAS is less than that for the schemes of Zhou et al. [9], Rodrigues et al. [11], Malani et al. [12], Ali et al. [13], and Bera et al. [15,16].

Table 4. Computation cost comparison.

Scheme	Year	Tiny Device/Sensor
Zhou et al. [9]	2019	$2T_h + 3T_{ecm} + T_{eca} + T_{mtp} + 3T_{bp}$ = 104.135 ms
Rodrigues et al. [11]	2019	$9T_h + 6T_{ecm}$ = 16.509 ms
Malani et al. [12]	2019	$6T_{ecm} + 7_{Th}/8_{Th} + 2T_{eca}$= 16.232 ms
Ali et al. [13]	2020	$18T_h + T_{fe} + T_{senc}$ = 7.868 ms
Bera et al. [15]	2021	$9T_h + 2T_{senc/sdec} + 2T_{ecm} + T_{eca}$ = 7.405 ms
Bera et al. [16]	2021	$11T_h + 4T_{ecm} + T_{eca}$ = 12.567 ms
VDAS	2022	$3T_h + 2T_{ecm} + T_{enc} + T_{dec}$ = 5.535 ms

5.3.2. Communication Cost Evaluation

In this subsection, the communication cost of the VDAS is evaluated. The bit size considered for identity is 160 bits, whereas the timestamp is fixed as 32 bits long. Besides, we assume that the size of elliptic curve cryptography coordinates is 160. Furthermore, the hash output is fixed to 256 bits (using the SHA-256 algorithm). Moreover, the encryption using the AES algorithm has a bit size of 128 bits. Table 5 gives a comparison of the communication costs among the schemes with the number of messages and the number of bits required during the authentication phase. In the proposed VDAS, we have three exchanged messages between the sensor and the controller, which are: Msg1 = ID_s, SC_1, T_s, Msg2 = SC_{c1}, $\{ID_c, C_{cr}, A_{cr}\}_\beta$, and Msg3 = $\{C_s,\ A_s\}_{k_{SC}}$, of size 160 + 320 + 32 = 512, 320 + 128 = 448, 128 respectively, and these all together need 1088 bits.

Table 5. Communication cost comparison.

Scheme	Number of Messages	Total Cost (in Bits)
Zhou et al. [9]	3	2464
Rodrigues et al. [11]	4	4288
Malani et al. [12]	2	2144
Ali et al. [13]	3	3424
Bera et al. [15]	3	2368
Bera et al. [16]	4	3040
VDAS	3	1088

The analysis of the obtained results showed that the proposed VDAS requires fewer communication costs as compared to other known authentication schemes such as Zhou et al. [9], Rodrigues et al. [11], Malani et al. [12], Ali et al. [13], and Bera et al. [15,16].

6. Implementation of the Proposed Solution

In this section, we highlight the key implementation aspects related to the vehicle counting algorithm presented in Section 3, the smart contract, and the communication protocol, concluding with the performance evaluation.

6.1. Sensor Node Design

The IoT sensor node for vehicle detection has been prototyped and field-tested. The results are reported in Ref. [1]. Figure 7 depicts the identification of two cars crossing a traffic signal system.

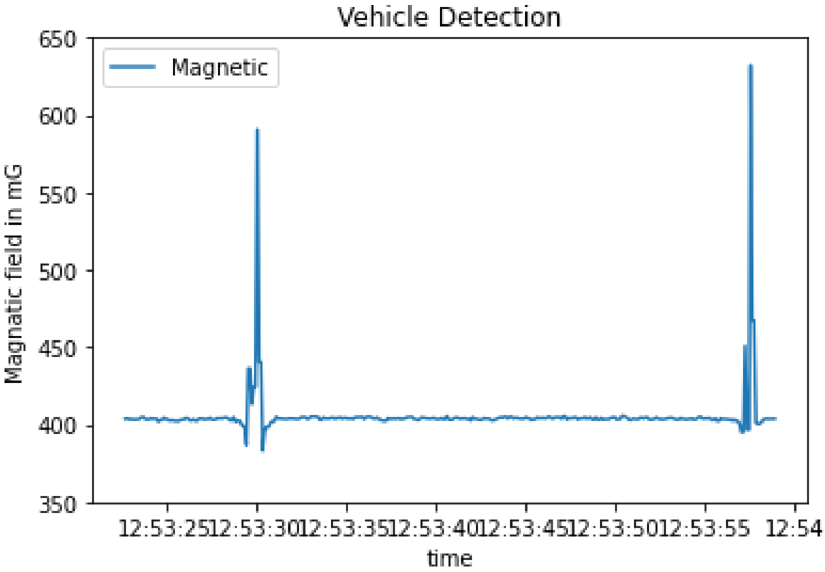
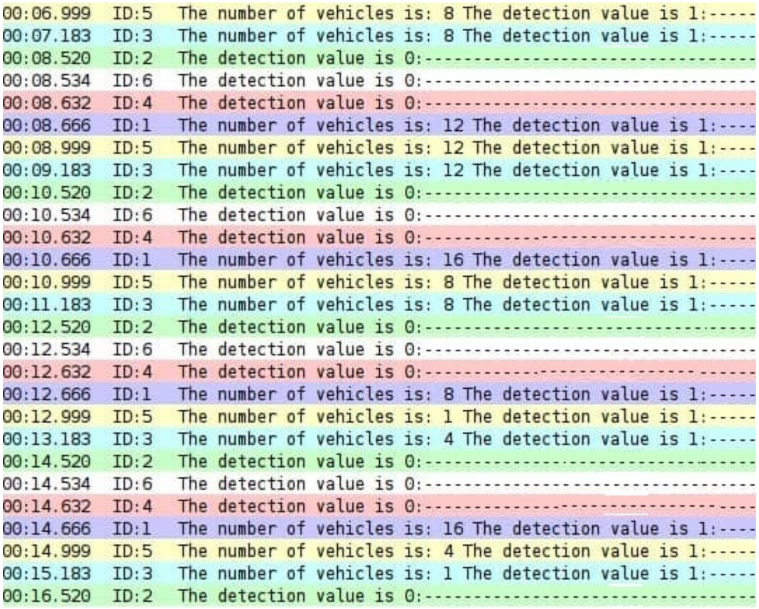


Figure 7. Detection of passing vehicles using our developed platform [1].

6.2. Counting Algorithm Implementation

In this section, we highlight the key implementation aspects related to the counting algorithm implementation. The vehicle counting algorithm is implemented using the Contiki-NG operating system [29]. Figure 8 shows the simulation realized on Cooja simulator [30] with six sky motes to test the proposed algorithm. When a vehicle is detected, the number of vehicles is calculated using the level of the sensor and the vehicle length.



```
00:06.999  ID:5  The number of vehicles is: 8 The detection value is 1:-----
00:07.183  ID:3  The number of vehicles is: 8 The detection value is 1:-----
00:08.520  ID:2  The detection value is 0:----------------------------------
00:08.534  ID:6  The detection value is 0:----------------------------------
00:08.632  ID:4  The detection value is 0:----------------------------------
00:08.666  ID:1  The number of vehicles is: 12 The detection value is 1:----
00:08.999  ID:5  The number of vehicles is: 12 The detection value is 1:----
00:09.183  ID:3  The number of vehicles is: 12 The detection value is 1:----
00:10.520  ID:2  The detection value is 0:----------------------------------
00:10.534  ID:6  The detection value is 0:----------------------------------
00:10.632  ID:4  The detection value is 0:----------------------------------
00:10.666  ID:1  The number of vehicles is: 16 The detection value is 1:----
00:10.999  ID:5  The number of vehicles is: 8 The detection value is 1:-----
00:11.183  ID:3  The number of vehicles is: 8 The detection value is 1:-----
00:12.520  ID:2  The detection value is 0:----------------------------------
00:12.534  ID:6  The detection value is 0:----------------------------------
00:12.632  ID:4  The detection value is 0:----------------------------------
00:12.666  ID:1  The number of vehicles is: 8 The detection value is 1:-----
00:12.999  ID:5  The number of vehicles is: 1 The detection value is 1:-----
00:13.183  ID:3  The number of vehicles is: 4 The detection value is 1:-----
00:14.520  ID:2  The detection value is 0:----------------------------------
00:14.534  ID:6  The detection value is 0:----------------------------------
00:14.632  ID:4  The detection value is 0:----------------------------------
00:14.666  ID:1  The number of vehicles is: 16 The detection value is 1:----
00:14.999  ID:5  The number of vehicles is: 4 The detection value is 1:-----
00:15.183  ID:3  The number of vehicles is: 1 The detection value is 1:-----
00:16.520  ID:2  The detection value is 0:----------------------------------
```

Figure 8. Simulation of the vehicle detection algorithm using Cooja.

6.3. Smart Contract Implementation

The smart authentication contract is implemented using Solidity language [31] in Remix IDE [32] and tested in Ganache [33], which is a personal Ethereum blockchain. The smart contract includes the registration and initialization functions plus two other main functions that manage the authentication request. Our smart contract uses two other contracts named the Elliptic curve and openzeppelin. The first is used to manage elliptic curve operations, while the second provides access control to manage access rights and secure the contract.

On the Ethereum network, gas is a unit of measurement for the amount of resources consumed by transactions [34]. A gas unit is debited from the controller's account when it generates a transaction. Figure 9 shows the gas consumption of the deployment of the authentication contract (CD), the registration and initialization function (RI), the first main function of the contract that manages the authentication request (F1), and the second main function of the contract that gives the final authentication decision (F2). Gas consumption depends on the complexity of the functions. We can inform the public that the smart contract deployment represents an expensive operation in Ethereum.

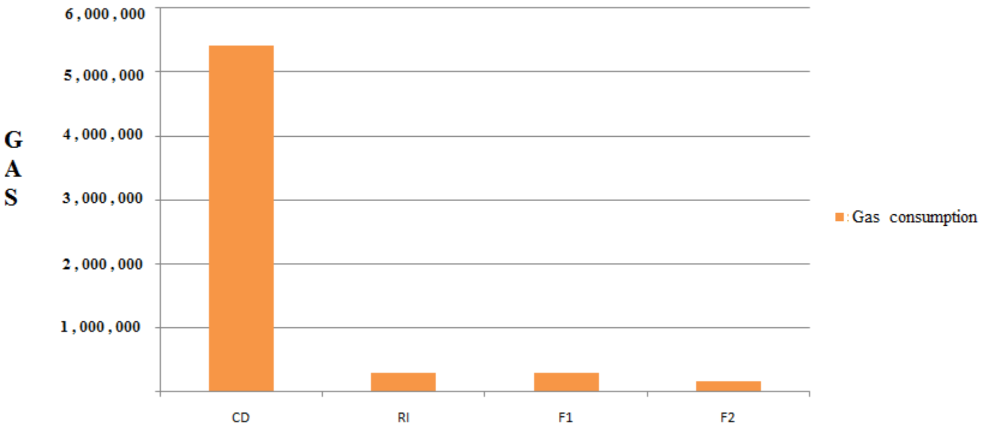


Figure 9. Gas consumption of the smart contract functions.

6.4. Communication Protocol

We used a secure implementation of CoAP/DTLS to set up a secure communication channel between the sensors and controllers. The sensors were simulated using Sky motes in Cooja while the controller was implemented using a Python script running outside the simulation environment. We used the existing TinyDTLS implementation library (a lightweight DTLS libarary) to handle the setup of the secure channel including data session and security handshake. The sensor acted as the CoAP client while the controller acted as the CoAP server.

To facilitate the communication between these two components, we implemented an additional node in Cooja running the RPL border router implementation, which acted as the gateway. The gateway node exposed a configured port that allowed the controller running as the Python script to send and receive messages to the sky mote via a tunnel interface.

6.5. Performance Evaluation

To evaluate the performance of the implementation, we considered the energy consumption and network latency of the setup. In particular, we used the ENERGEST module on Contiki to measure the power consumption of the sensor node running in Cooja. The module can estimate power consumption by tracking the power state of components. It allowed us to determine the CPU usage time, LPM (reduced power CPU), and listen/transmit power consumption. To measure the network latency, we used the timer API (ctimer) provided by Contiki-NG. We considered the total latency for the exchange of the three messages between the sensor and the Python script. We configured the timer before the first message and also after the last message was sent from the sensor to the controller. Table 6 shows the configuration of the simulation environment that was used to evaluate the performance of the implementation.

Table 6. The simulation environment for performance evaluation.

Item	Description
Simulator	Cooja
Sensor device	skymote
Sensor OS	Contiki-NG
DTLS library	TinyDTLS 0.8.1
DTLS cipher suite	TLS PSK WITH AES 128 CCM 8
Network	RPL/IPv6/UDP
Power consumption measuring function	energest() of Contiki-NG
Network latency measuring function	ctimer_set() of Contiki-NG

The sensor node, which is buried in the road, runs the Contiki operating system. Because it is battery-powered, power dissipation is a critical design concern. Furthermore, the detected traffic data should be provided to the traffic controller as quickly as possible. Latency must be assessed and optimized for this purpose. Figures 10 and 11 show the results of the power consumption and network latency evaluation of the implementation. The results show that the proposed authentication scheme has minimal overhead impact on the performance of the implementation.

A large portion of the power is consumed when the sensor received data from the controller during the authentication phase. To further lower the average power, the authentication needs to be carried out less frequently.

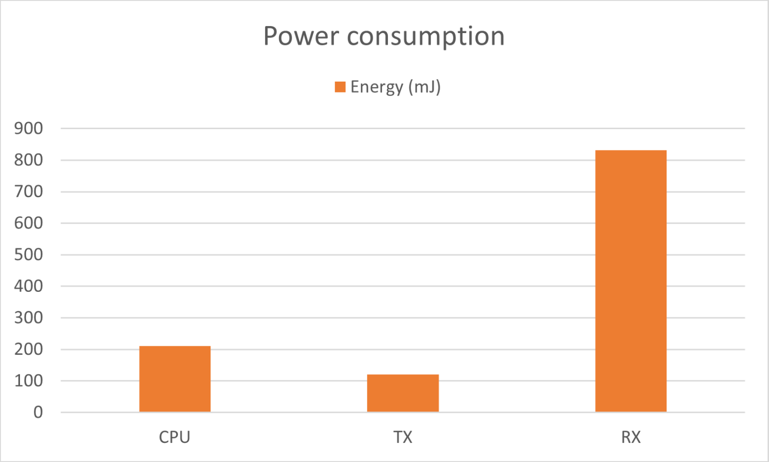


Figure 10. Power consumption performance.

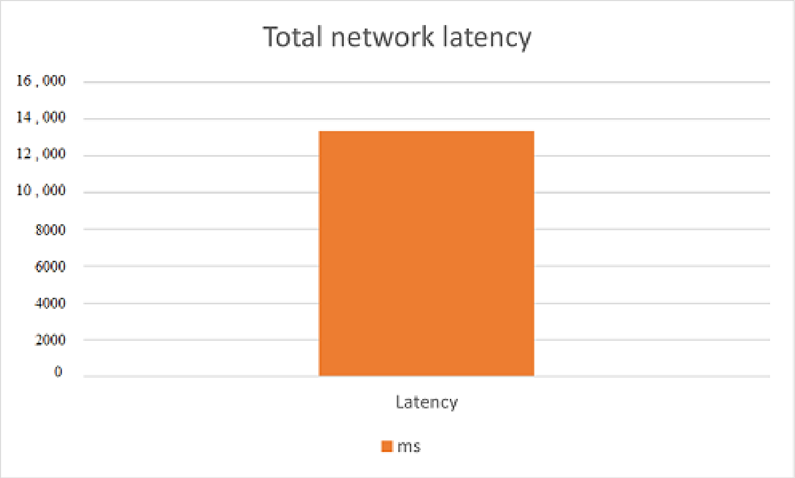


Figure 11. Network latency performance.

7. Discussion

The integration of blockchain technology with the fog computing paradigm enables us to take advantage of blockchain decentralization as well as lower network latency. The computational cost of an authentication procedure is determined by the number and complexity of cryptographic primitives linked to Elliptic Curve Cryptography (ECC), particularly hash functions, scalar multiplication, and random number generation. Furthermore, the communication cost of an authentication protocol is influenced by the number of messages sent and received by the transacting parties: the sensor node and the traffic light controller. Reducing these two costs is a difficult issue since this decrease must be done without jeopardizing the protocol's resilience against known threats. The authentication protocol's dependability and resilience should be verified using both formal and informal methods. The suggested authentication protocol has lower computation and communication costs than state-of-the-art authentication systems, as per VDAS assessment. Furthermore, formal and informal VDAS verifications have demonstrated that it is secure against a wide range of known threats. We assessed the gas usage of the smart contract implementation. This value is determined by the amount and complexity of functions in the smart contract. During the authentication step, the sensor node running the Contiki operating system consumes a significant amount of energy.

8. Conclusions

In this paper, we devised the Vehicle Detector Authentication Scheme (VDAS), a blockchain-based authentication system, and a fog-based architecture for a networked traffic light system. Our primary goal was to address sensor node authentication and securely transfer the number of detected vehicles to the fog node. The use of a smart contract with intersection controllers as Ethereum clients ensures decentralized access control, preventing the involvement of a third party. The protocol's formal and informal verification revealed that it is secure against a number of known attacks. VDAS needs fewer communication and calculation costs than other current authentication systems. Through the implementation of the smart contract, it is safe to estimate the gas consumption of the contract deployment and the functions provided. Hence, the proposed solution satisfied all the comparison criteria studied in the second section. Future work will focus on the implementation of the VDAS authentication scheme (sensor operations), using the Contiki operating system and reducing smart contract gas consumption by improving the functions of this contract.

Author Contributions: Conceptualization, S.N. and I.B.D.; methodology, S.N. , M.A. and I.B.D.; software, S.N. and A.K.; validation, S.N., M.A. and A.K.; formal analysis, S.N., M.A., A.K. and I.B.D.; investigation, S.N. and I.B.D.; writing—original draft preparation, S.N.; writing—review and editing, S.N. and I.B.D.; visualization, S.N. and I.B.D.; supervision, I.B.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. Ben Dhaou, I. A Secure IoT-enabled Sensor Node for Traffic Light Management and Level of Service Computation. In Proceedings of the 2021 18th International Multi-Conference on Systems, Signals & Devices (SSD), Monastir, Tunisia, 22–25 March 2021; pp. 644–648. [CrossRef]
2. Yang, B.; Lei, Y. Vehicle Detection and Classification for Low-Speed Congested Traffic with Anisotropic Magnetoresistive Sensor. *IEEE Sens. J.* **2015**, *15*, 1132–1138. [CrossRef]
3. Navarro-Espinoza, A.; López-Bonilla, O.R.; García-Guerrero, E.E.; Tlelo-Cuautle, E.; López-Mancilla, D.; Hernández-Mejía, C.; Inzunza-González, E. Traffic Flow Prediction for Smart Traffic Lights Using Machine Learning Algorithms. *Technologies* **2022**, *10*, 5. [CrossRef]
4. Shamsi, M.; Rasouli Kenari, A.; Aghamohammadi, R. Reinforcement learning for traffic light control with emphasis on emergency vehicles. *J. Supercomput.* **2022**, *78*, 4911–4937. [CrossRef]
5. Hossan, S.; Nower, N. Fog-based dynamic traffic light control system for improving public transport. *Public Transp.* **2020**, *12*, 431–454. [CrossRef]
6. Stallings, W.; Brown, L. *Computer Security: Principles and Practice*, Global ed.; Pearson: London, UK, 2019.
7. Namane, S.; Ben Dhaou, I. Blockchain-Based Access Control Techniques for IoT Applications. *Electronics* **2022**, *11*, 2225. [CrossRef]
8. Jabbar, R.; Dhib, E.; Said, A.B.; Krichen, M.; Fetais, N.; Zaidan, E.; Barkaoui, K. Blockchain Technology for Intelligent Transportation Systems: A Systematic Literature Review. *IEEE Access* **2022**, *10*, 20995–21031. [CrossRef]
9. Zhou, Y.; Guan, Y.; Zhang, Z.; Li, F. A Blockchain-Based Access Control Scheme for Smart Grids. In Proceedings of the 2019 International Conference on Networking and Network Applications (NaNA), Daegu, Korea, 10–13 October 2019; pp. 368–373. [CrossRef]
10. Kumari, A.; Tanwar, S.; Tyagi, S.; Kumar, N.; Obaidat, M.S.; Rodrigues, J.J.P.C. Fog Computing for Smart Grid Systems in the 5G Environment: Challenges and Solutions. *IEEE Wirel. Commun.* **2019**, *26*, 47–53. [CrossRef]
11. Rodrigues, M.; Amaro, J.; Osório, F.S.; Branco Kalinka, R.L.J.C. Authentication Methods for UAV Communication. In Proceedings of the 2019 IEEE Symposium on Computers and Communications (ISCC), Barcelona, Spain, 29 June–3 July 2019; pp. 1210–1215. [CrossRef]
12. Malani, S.; Srinivas, J.; Das, A.K.; Srinathan, K.; Jo, M. Certificate-Based Anonymous Device Access Control Scheme for IoT Environment. *IEEE Internet Things J.* **2019**, *6*, 9762–9773. [CrossRef]
13. Ali, Z.; Chaudhry, S.A.; Ramzan, M.S.; Al-Turjman, F. Securing Smart City Surveillance: A Lightweight Authentication Mechanism for Unmanned Vehicles. *IEEE Access* **2020**, *8*, 43711–43724. [CrossRef]
14. Srinivas, J.; Das, A.K.; Kumar, N.; Rodrigues, J.J.P.C. TCALAS: Temporal Credential-Based Anonymous Lightweight Authentication Scheme for Internet of Drones Environment. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6903–6916. . [CrossRef]

15. Bera, B.; Das, A.K.; Sutrala, A.K. Private blockchain-based access control mechanism for unauthorized UAV detection and mitigation in Internet of Drones environment. *Comput. Commun.* **2021**, *166*, 91–109. [CrossRef]
16. Bera, B.; Saha, S.; Das, A.K.; Vasilakos, A.V. Designing Blockchain-Based Access Control Protocol in IoT-Enabled Smart-Grid System. *IEEE Internet Things J.* **2021**, *8*, 5744–5761. [CrossRef]
17. Kumari, A.; Gupta, R.; Tanwar, S.; Tyagi, S.; Kumar, N. When Blockchain Meets Smart Grid: Secure Energy Trading in Demand Response Management. *IEEE Netw.* **2020**, *34*, 299–305. [CrossRef]
18. Khalid, U.; Asim, M.; Baker, T.; Hung, P.C.K.; Tariq, M.A.; Rafferty, L. A decentralized lightweight blockchain-based authentication mechanism for IoT systems. *Clust. Comput.* **2020**, *23*, 2067–2087. [CrossRef]
19. Tahir, M.; Sardaraz, M.; Muhammad, S.; Saud Khan, M. A Lightweight Authentication and Authorization Framework for Blockchain-Enabled IoT Network in Health-Informatics. *Sustainability* **2020**, *12*, 6960. [CrossRef]
20. Kumari, A.; Chintukumar Sukharamwala, U.; Tanwar, S.; Raboaca, M.S.; Alqahtani, F.; Tolba, A.; Sharma, R.; Aschilean, I.; Mihaltan, T.C. Blockchain-Based Peer-to-Peer Transactive Energy Management Scheme for Smart Grid System. *Sensors* **2022**, *22*, 4826. [CrossRef] [PubMed]
21. Haqani, E.A.; Baig, Z.; Jiang, F. A Decentralised Blockchain-Based Secure Authentication Scheme for IoT Devices. In *Inventive Systems and Control*; Lecture Notes in Networks and Systems; Suma, V., Baig, Z., Kolandapalayam Shanmugam, S., Lorenz, P., Eds.; Springer Nature: Singapore, 2022; pp. 123–144.
22. Hussain Al-Naji, F.; Zagrouba, R. CAB-IoT: Continuous authentication architecture based on Blockchain for internet of things. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 2497–2514. [CrossRef]
23. Ashi, Z.; Al-Fawa'reh, M.; Al-Fayoumi, M. Fog Computing: Security Challenges and Countermeasures. *Int. J. Comput. Appl.* **2020**, *175*, 30–36. [CrossRef]
24. Choudhary, V.; Singh, Y.; Anand, P. Smart Traffic Monitoring with Fog and Cloud Computing. In *Emerging Technologies for Computing, Communication and Smart Cities*; Singh, P.K., Kolekar, M.H., Tanwar, S., Wierzchoń, S.T., Bhatnagar, R.K., Eds.; Springer Nature: Singapore, 2022; pp. 317–327.
25. Dhingra, S.; Madda, R.B.; Patan, R.; Jiao, P.; Barri, K.; Alavi, A.H. Internet of things-based fog and cloud computing technology for smart traffic monitoring. *Internet Things* **2021**, *14*, 100175. [CrossRef]
26. AVISPA. Automated Validation of Internet Security Protocols and Applications. 2019. Available online: https://www.avispa-project.org/ (accessed on 22 November 2022).
27. MIRACL. Cryptographic SDK: Multiprecision Integer and Rational Arithmetic Cryptographic Library. Available online: https://github.com/miracl/MIRACL (accessed on 1 September 2022).
28. Raspberry Pi 3 Model B+. Available online: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/ (accessed on 1 September 2022).
29. Contiking. Contiking Operating System. Available online: https://github.com/contiki-ng/contiki-ng (accessed on 7 August 2022).
30. Cooja. Cooja Simulator. Available online: https://anrg.usc.edu/contiki/index.php/Cooja_Simulator/ (accessed on 7 August 2022).
31. Solidity. Solidity Language. Available online: https://soliditylang.org/ (accessed on 20 September 2022).
32. Remix. Remix IDE. Available online: http://remix.ethereum.org/ (accessed on 20 September 2022).
33. Ethereum. GANACHE. Available online: https://trufflesuite.com/ganache/ (accessed on 1 August 2022).
34. Laurent, A.; Brotcorne, L.; Fortz, B. Transaction fees optimization in the Ethereum blockchain. *Blockchain Res. Appl.* **2022**, *3*, 100074. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.