electronics

Article

A Secure and Efficient Method to Protect Communications and Energy Consumption in IoT Wireless Sensor Networks

Safwan Mawlood Hussein [1,*], Juan Antonio López Ramos [2] and Abubakar Muhammad Ashir [1]

[1] Department of Computer Engineering, Tishk International University, Erbil 44001, Kurdistan Region, Iraq
[2] Department of Mathematics, University of Almería, 04120 Almería, Spain
[*] Correspondence: safwan.mawlud@tiu.edu.iq

Abstract: The rapid growth of technology has resulted in the deployment of a large number of interconnected devices, resulting in a wide range of new societal services. Wireless sensor networks (WSNs) are a promising technology which is faced with the challenges of operating a large number of sensor nodes, information gathering, data transmission, and providing a means to act in different scenarios such as monitoring, surveillance, forest fire detection, and many others from the civil to military spectrum. The deployment scenario, the nature of the sensor-equipped nodes, and their communication methods make this architecture extremely vulnerable to attacks, tampering, and manipulation than conventional networks. Therefore, an optimal solution to ensure security in such networks which captures the major constraints of the network in terms of energy utilization, secured data transmission, bandwidth, and memory fingerprint to process data is required. This work proposes a fast, reliable, and secure method of key distribution and management that can be used to ensure the integrity of wireless sensor networks' communications. Moreover, with regards to efficient energy utilization, an improvement of the Low Energy Adaptive Clustering Hierarchy (LEACH) algorithm (a cluster routing protocol that is mainly used in WSN) has been proposed to enhance the networks' energy efficiency, simplicity, and load-balancing features. Therefore, in this paper, we propose a combination of a distributed key exchange and management methods based on elliptic curve cryptography to ensure security of node communication and an improved routing protocol based on the LEACH protocol to demonstrate better performance in parameters such as network lifespan, dead nodes, and energy consumption.

Keywords: WSN; group key management; ECC; smart agriculture; routing protocols; LEACH

check for updates

Citation: Hussein, S.M.; López Ramos, J.A.; Ashir, A.M. A Secure and Efficient Method to Protect Communications and Energy Consumption in IoT Wireless Sensor Networks. *Electronics* **2022**, *11*, 2721. https://doi.org/10.3390/electronics11172721

Academic Editor: Cheng-Chi Lee

Received: 24 July 2022
Accepted: 25 August 2022
Published: 30 August 2022

1. Introduction

Technology has dramatically changed the way we live, learn, work, and communicate, and it is changing individual behaviour. New technologies and trends can lead to incredible success for organizations, individuals, and governments. These new technologies can provide better services and enhance citizens' lifestyles, and one example of this is the Internet of Things (IoT) [1,2].

The IoT paradigm can be extremely massive and complex. It may contain tens of thousands of sensors, actuators, and gateways. Devices can communicate with gateways via a different number of protocols, whereas gateways may connect with the internet and cloud-based apps via a similarly diverse range of protocols.

In this complex architecture, data can be processed by various heterogeneous entities. Data transmission and data security and integrity are key aspects to be considered. As a result, protocols and technologies are required to provide data security, access management, and flow data transmission. In recent years, many studies have been conducted to cope with security issues in the IoT paradigm. Some of these studies concentrate on security issues at a particular layer, whereas other approaches aim at providing end-to-end security [3].

Meanwhile several methods, and protocols have been suggested, primarily concerned with reducing energy consumption and increasing the network lifetime [4,5].

Two of the major challenges in WSN are security and energy management. WSNs consist of a large number of sensor nodes which are highly decentralized and deployed in a difficult environment with unreliable channels of communication. As a result of the complexity and interconnection of a wide range of devices and protocols, as well as the variety of routing protocols and services available, it is extremely difficult to apply IT network solutions in WSNs. Hence, most of the current security mechanisms in use are either insufficient or incompatible. Many techniques and trends have been used to achieve a certain range of security levels, such as trust management and encryption techniques using lightweight methods to have low-cost encryption for low-power and constrained devices. Routing protocols may also be vulnerable to serious attacks, such as the introduction of false or malicious routing information into a network, causing delays or packet losses due to routing conflicts. Many solutions have been proposed to avoid routing attacks such as encryption and information correlation between multiple nodes [6]. Cryptographic methods are then proposed to ensure security, but due to the constraints of the devices, such as energy and memory capacity, only a lightweight encryption technique might be applied. These algorithms can be implemented in software or in hardware using an integrated circuit (IC). Each of these solutions incurs an additional expense for the sensor node manufacturer since they both require the use of additional resources.

Moreover, wireless sensor nodes are firmly restricted in relation to transmission energy, bandwidth, capacity, storage, and on-board energy. Due to such dissimilarities, a number of new routing protocols have been proposed in order to cope with these routing challenges in wireless sensor networks. Wireless sensor networks are mostly battery-powered. An energy shortage is a major issue in these sensor networks especially in harsh and unfriendly terrains. The performance of sensor nodes is adversely affected when the battery level falls below a pre-defined battery threshold level. Energy presents a main challenge for designers while designing sensor networks. Each node in this network has restricted energy resources due to a partial amount of power. So, the routing protocol should be energy efficient. Different routing protocols for wireless sensor networks (WSNs) have been proposed in recent years. Routing protocols for WSN may be classified into three categories: flat, location-aware, and hierarchical. The first category has a high energy consumption due to the methodology of work in which the whole traffic is transferred to the base station. In the case of hierarchical routing protocols, energy consumption decreases and thus the lifetime of the network increases. This category relies on establishing a cluster, and a head node is allocated to each cluster. Data from each cluster is gathered and consolidated at the head node before being transferred to the base station. Finally, the location-aware category forwards the data to the base station using geographic forwarding to trace the node's position.

This paper proposes a model to address two of the major challenges mentioned, namely, security and energy in WSN. Security is considered while data is being transferred between authorised nodes by means of a distributed key management scheme using elliptic curve cryptographic operations. In addition, we proposed an enhanced version of the LEACH routing protocol that has been redesigned to provide better and more efficient power consumption, thus maximizing the life of the network in addition to less message overhead. The proposed model is scalable and depicts the primary response patterns of a network node. The proposed model has been evaluated, showing better performance figures on WSN's. The major contributions of this research can be identified as follows:

- Propose a method for key generation, distribution, and management using an elliptic curve approach with fewer overheads and time for secure, fast and reliable data exchange between nodes;
- Propose a dynamic approach to shared key regeneration based on sensing network topology changes or attacks to prevent intrusion and ensure that data integrity is not compromised;

- Propose an enhanced low energy consumption adaptive routing protocol to reduce overall network energy consumption and thereby extending the lifespan of the network.

2. Theoretical Background

2.1. Wireless Sensor Network Routing Protocols

A wireless sensor network consists of a finite set of devices named sensors that can sense or regulate physical characteristics such as sound, light, temperature, humidity, and others in a geographical area. Wireless sensor nodes communicate with the base station or sink and other nodes through wireless channels as shown in Figure 1. Sensor nodes have limited energy, memory, and CPU capacity. A sensor node has a power unit, a processing unit, one or more sensing units, a transceiver, an antenna, and optional components like a position finding system, a power generator, and an actuator. The volume of sensor nodes varies from cubic nanometres to cubic decimetres. [7]. The node location of sensor nodes in a network may be known or unknown, actual, or logical communication between network nodes and all other devices determines the topology. A WSN can have different topologies depending on the network and node tasks. WSN relies heavily on the speed and reliability of data delivery. Routing protocols are in charge of establishing how data moves through the network. Routing methods for WSNs should take energy usage, coverage area, and other considerations into account. Based on the network topology, WSN's routing protocols may be classified as either flat, location-aware, or hierarchical [8].

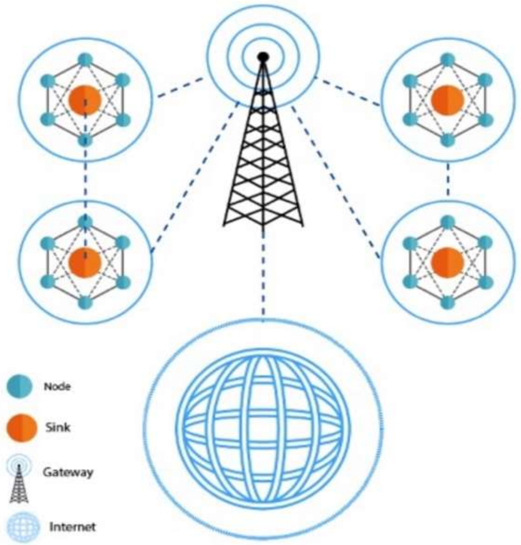

Figure 1. Conventional architecture of a wireless sensor network.

2.2. IoT Architecture in Wireless Sensor Network

The Internet of Things (IoT) architecture as depicted in Figure 2, consists of three domains: device or hardware, network or communication, and a layer of interfaces and services. The first layer includes sensors and actuators. For instance, an irrigation system might include individual sprinkler heads, moisture sensors, temperature sensors, and actuators, which are connected to the network using different protocols. In this layer, the end devices use a microprocessor based on the Interplanetary File System (IPFS), which is used for distributed data storage and data integrity while taking into account the problems with centralized systems [9], ARM, or X86 architectures, and a Real Time Operating System (RTOS) hardware operating system. For the application software layer, there are custom applications, cryptographic protocols, and third-party libraries and drivers. The network domain, or communication layer, presents the process of collecting and transferring data to the application domain. Different protocols are used to connect objects with the gateways and these to the internet and cloud applications.

The limitations of IoT sensor nodes require having a communication channel between sensors and a gateway device. The gateway translates wireless sensor network (WSN) traffic to IP protocol traffic that can travel on conventional data networks. Some wireless sensor networks can have a large number of sensor nodes. These nodes may only be able to operate on battery power and have very little computational capability. Due to power limitations, these nodes are only able to communicate across extremely limited distances. In this instance, protocols are employed to allow sensor data to pass from node to node until the data reaches the gateway. IoT wireless protocols can be formed into different wireless topologies using the IEEE 802.15.4 standard, such as hub-and-spoke, star, mesh, and cluster tree topologies. The data is forwarded from one node to another in order to reach a gateway, while if the nodes have their own IPv6 protocol stacks and messaging protocols, they can communicate directly with the cloud or data centre without requiring translation into IP by an IoT gateway. The Application Domain (interfaces and services) layer provides management services such as data processing and analytics, storage, smart energy and connectivity management, or any application that consumes the data from IoT devices [10].

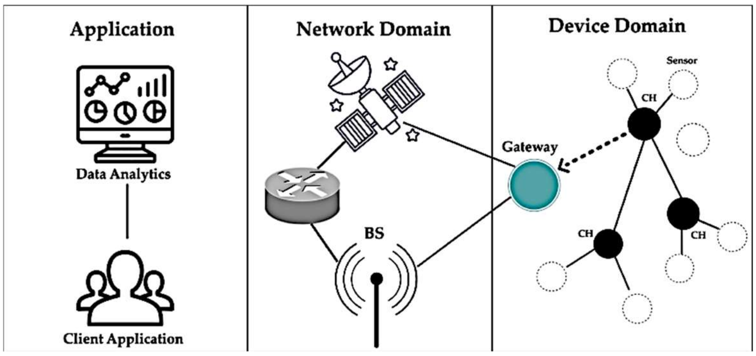

Figure 2. Architecture layer of IoT scheme.

Types of Possible Attacks in IoT

The Internet of Things (IoT) paradigm involves such a wide array of devices and equipment, ranging from tiny, embedded processing chips to enormous high-end servers, that it is necessary for it to address concerns regarding security on multiple levels.

1. The first level is concerned with the security issues at the network interface layer, where adversaries try to disrupt the normal operation of the network, such as:
   - A jamming attack can be introduced as a high radio frequency signal that is used against wireless devices in WSN and IoT devices to disrupt the normal operation of the network. It might cause the system to fail or behave in unexpected ways [11];
   - Node capturing: this type of attack is regarded as one of the most serious types of wireless network attack. This attack is launched by an adversary in order to gain partial or complete control over the sensor nodes. The adversary captures the legal sensor nodes to extract important security information regarding shared secrets and cryptographic keys and uses them to conduct additional assaults [12,13];
   - Sybil attacks are a type of attack that can be deployed by attackers on both low- and intermediate-level layers to reduce the performance of sensor and IoT networks and even violate data privacy. An adversary may eavesdrop on the information of a legitimate node and use fake identities to conduct a sybil attack [3,14].
2. Intermediate-level IoT security concerns the communication, routing, and session management occurring at the network and transport layers in the TCP/IP model. Both layers are vulnerable to a variety of attacks. This type of attack involves re-routing all

traffic to a compromised node that appears to be a real node. The consequence is that an adversary can disrupt the flow of traffic by modifying routing information [15].

- Sinkhole Attack: This is considered a common type of active network layer attack in which an adversary node modifies routing packets and informs neighbor nodes of a false shortest path to a cluster or base node, making other nodes route their traffic through it. The consequence of this type of attack over the network is increased latency. A sinkhole attack is a severe threat to WSN, and most WSN routing protocols are incapable of detecting it [16,17];
- Wormhole Attack: This is a form of layer three attack that employs more than one malicious node. The nodes utilized in this assault are more powerful than average ones, making it possible for them to set up more reliable long-distance communication channels. The goal of this assault is to forward the data from a hacked node through a tunnel to a second malicious node farther down the network. This type of attack causes issues with the WSN's routing algorithm since it tricks other nodes into thinking they're closer to other nodes than they actually are. Data packets could also be tampered with by infected nodes. This form of assault is similar to the sinkhole attack in that one of the conniving nodes could falsely be represented as the sink node, attracting more traffic than usual. Wormhole attacks can be used with sinkhole assaults to increase their effectiveness [18];
- Clone ID and Sybil Attacks: Attackers use clone IDs to compromise another node by transferring the identities of legitimate nodes. This can be used for a variety of purposes, including bypassing voting restrictions, and gaining access to otherwise inaccessible areas of the network. Sybil attacks, which are similar to clone ID attacks, involve the deployment of many logical entities on a single physical node. Sybil attacks allow for the takeover of huge portions of a network without the need to physically deploy any nodes [19].

3. High-level security issues such as overwhelm, repudiation, data corruption, and malicious code injection are all examples of attacks that can be carried out at the application layer. The assault causes excessive energy consumption across nodes and uses up all available network bandwidth [16].
    - CoAP security: In order to enable effective application-level communication for Internet of Things devices, the Constrained Application Protocol (CoAP) has been created as a protocol that operates over UDP. CoAP messages typically contain instructions for the nodes to carry out a particular activity. Replaying these messages has the potential to alter the behavior of the nodes in the 6LoWPAN. In the absence of filtering, the replayed packet has the potential to have a significant impact on the performance of the network, with effects comparable to those that result from flooding attacks [20];
    - Middleware security: The IoT middleware developed to enable communication across heterogeneous IoT entities must be safe enough to provide services. To keep communications private and secure, there must be a number of different interfaces and environments that use middleware [3].

2.3. Security in Wireless Sensor Networks

In most cases, WSNs are not managed centrally, are deployed in unsuitable environments, and use unreliable channels of communication. Therefore, WSN's security mitigation cannot be relied on standard IT network solutions due to the complexity and interconnection of a wide range of devices and protocols, as well as the variety of routing protocols and services available. Hence, current security mechanisms in use are insufficient. Many techniques and trends have been used to achieve a certain range of security levels, such as trust management and encryption techniques using lightweight methods to have a low-cost encryption for low-power and constrained devices. Routing protocols might also be susceptible to serious attacks such as injecting false or malicious routing information into a network, resulting in delays or packet losses due to routing conflicts. Many solutions have been proposed to avoid routing attacks such as encryption and information correlation between multiple nodes [6].

Any proposed protocol should compromise between two competing priorities: security and performance. Low energy consumption, processing, and storage usage are all sacrificed for better performance against security level and vice versa. Key management is one method that may be used to ensure the security of WSNs [21]. On the basis of security policy, it is described as a set of activities and mechanisms that facilitate key distribution and adhere to the criteria of the keying process among nodes. It is necessary to generate, maintain, distribute, protect, and control the use of cryptographic keys. Since they have the capacity to update the keys in the sensor nodes, the techniques of key management are classified as either static or dynamic [22]. In static key management, the encryption key will be stored in a secret manager, the principles of the encrypted keys are identified before they are distributed, and the keys themselves remain unaltered for the duration that the network is operational, which makes it more vulnerable to attacks. Conversely, in dynamic key management, keys are updated frequently depending on the network policies.

Group key management is broadly used in various modern collaborative and distributed applications to provide security [23]. Based on [22], group key management provides good performance in terms of energy and memory consumption and security level, in addition to low communication overhead with high scalability while it is used in WSNs.

2.3.1. Elliptic Curve Cryptography

The concept of public-key cryptography was introduced in 1976 by Whitfield Diffie and Martin Hellman (DH) [24] to solve the issue of key management. In their concept, each party gets a pair of keys: one called a “public key” and the other called a “private key”. These two keys are mathematically related to each other. DH is intended for two-party communication. However, it does not work well in situations where group members change frequently. The extension of Diffie–Hellman is Group Diffie–Hellman key exchange (GDH) [25], which supports group communications in which more than two members can participate.

Many methods have been proposed to improve the Diffie–Hellman key exchange protocol. Cliques [25] proposed two distinct extensions of the DH. The results demonstrate efficient performance in the rekeying process, and they do so by just using a single message. The drawback of [26] has been mentioned in paper [27]. The authors approved the control of communication of the last two particular parties for only the duration of the key exchange using active attack. Steiner [28] made some modifications to the previous protocol and used double DH key exchange. In all the above-mentioned protocols, the issue is sending many messages and processing many rounds during the communication while establishing an initial key agreement. The use of public key cryptography based on elliptic curves could solve problems related to key management in WSNs.

Using elliptic curve cryptography (ECC), which is most representative of lightweight asymmetric-key algorithms, can provide 128-bit cryptographic security using a 256-bit key, which is significantly smaller than the 3072-bit key of the most widely used public-key encryption algorithm RSA [29]. ECC has been applied to various cryptographic algorithms, including elliptic curve Diffie–Hellman (ECDH) and the elliptic curve digital signature algorithm (ECDSA).

Elliptic Curve Groups

Let an integer p be a prime number, and $\mathbb{F}_p$ represent a field of integers modulo p. An elliptic curve $\boldsymbol{E}$ over $\mathbb{F}_p$ can be denoted by an equation of the form defined below.

$$y^2 = x^3 + ax + b \tag{1}$$

Where terms a and b are integer constants such that $a, b \in \mathbb{F}_p$ and satisfy the condition in Equation (2).

$$4a^3 + 27b^2 \not\equiv 0\ (mod\ p) \tag{2}$$

A pair (x, y) on the curve $\boldsymbol{E}$ such $x, y \in \mathbb{F}_p$, is a point on the curve if (x, y) satisfies the equation. The set of all the points on E is expressed as denoted by $\boldsymbol{E}(\mathbb{F}_p)$.

To implement discrete logarithm systems, cyclic subgroups of a point $P(x, y)$ on an elliptic curve $\boldsymbol{E}(\mathbb{F}_p)$ can be used.

2.3.2. Elliptic Curve Key Generation

Consider an elliptic curve $\boldsymbol{E}$ defined over a finite field of integers $\mathbb{F}_p$ with G *as* a point in $\boldsymbol{E}(\mathbb{F}_p)$ and suppose point G has a prime order n. The cyclic subgroup of $\boldsymbol{E}(\mathbb{F}_p)$ generated by point G *is* then $G = \{\ \infty,\ G,\ 2G,\ 3G, \ldots\ldots (n-1)G\}$. The prime p, the equation of the elliptic curve E, and the point G and its order n are the public domain parameters.

An integer private key d, which is selected uniformly at random from the interval [1, n−1], can be used to generate the corresponding public key Q (treated as a point on the curve) using the relation below as a scalar multiplication of private key d with generator point G as represented by Equation (3).

$$Q = dG \tag{3}$$

The problem of determining d given the domain parameters and Q is the elliptic curve discrete logarithm problem.

2.4. LEACH Protocol

The LEACH proposed by [30] employs a hierarchical structure routing technique. In addition to providing data fusion, the LEACH protocol's primary purpose is to improve the efficiency with which sensor nodes utilize the power they consume and to maximize the amount of time a network may remain operational by combining many different types of approaches, including cluster-based routing and MAC-layer techniques [31]. The sensor nodes that make up the network are responsible for delegating tasks to one another. For instance, one of the nodes is designated as the cluster head, while the rest are classified as leaf nodes or non-cluster heads. The sensed data by sensor nodes (leaf) are transmitted to the cluster head within a particular cluster group. The cluster head is responsible for transmitting the collected data to the sink node or the base station. Nodes in a cluster consume significantly less power than the cluster head. When the cluster head dies, the leaf nodes lose their ability to communicate.

Low-energy adaptive clustering hierarchy LEACH [4,30] is one of the most efficient and simple clustering and routing protocols in wireless sensor networks. The core objective of the protocol is to minimize the overall network energy consumption requirements for creating and maintaining clusters in order to improve the lifetime of the network. The general working principle of the LEACH protocol is hierarchical where nodes transmit to cluster heads, and the cluster heads aggregate the data and forward it to the base station (sink). Each node uses a stochastic algorithm at each round to determine whether it will become a cluster head in this round. Nodes that have been cluster heads cannot become cluster heads again for a period of P rounds, where P is the desired percentage of cluster heads. Thereafter, each node has a $1/P$ probability of becoming a cluster head again. At the end of each round, each node that is not a cluster head selects the closest cluster head and joins that cluster.

The algorithm of the LEACH protocol operates in two phases: setup and steady state. In the first phase, the cluster and cluster head will be determined. The amount of energy that is still available is used by each node to determine whether or not it will act as a cluster head (CH) throughout the session. The cluster head broadcasts a message to all other nodes that become a CH, then the nodes request a CH to join the cluster. Based on the number of nodes joining the cluster, the CH creates a TDMA schedule which contains slots of time equal to the number of member nodes in the cluster. The second phase is responsible for identifying the process of data transmission [4,32].

3. Methodology

This section discusses the testbed for the experiments of the proposed key distribution and management using ECC and an enhanced version of the LEACH algorithm. We are also going to address the impact of the proposed methods; lightweight cryptography to avoid external entities participating in the group of sensors, and LEACH enhancement to improve power utilization and extend network lifespan.

3.1. Proposed ECC Key Distribution and Management

In this proposed scheme, we have used a private key generated using a random integer in a certain range of the curve's field size. The scheme is summarized into four phases starting with: identifying the number of clusters, authenticating all the nodes, identifying the cluster head, key distribution by all the members in the group, generating the routing table using routing protocol, and the last phase is the rekeying process in case of any change in the group. A list of notations is explained in Table 1.

Table 1. List of notations.

Notation	Description
$\mathrm{ID_{GHi}}$	Identity for group head $\mathrm{GH_i}$
$\mathrm{GH_i}$	Group heads (gateway, sink node)
$\mathrm{r_g}$	Random number generated by each node
$\mathrm{A_{UG}}$	Authentication message by group head
x_i	Private key for node i
$\mathrm{U_i}$	Node in the group
S_{id}	Source ID
D_{id}	Destination ID
M_{id}	Message ID
NC	Node counts, indicates the number of nodes in the network that computed their points on the elliptic curve.
P_x, $\mathrm{P_y}$	P_x and P_y are the x and y coordinates of a point P on the EC.
P*mid*	Previous message id, which is used to save the id of the last seen message
NN	Total number of nodes in the network
Skey	Shared key
P	Odd prime modulus
G	Generator base point on the EC with G_x *and* G_y as x and y coordinates
N	EC group order
a,b	Constant integers for the curve
H	Co-factor

The scheme operates on the assumption that the network contains several nodes, and all the nodes have the same characteristics with the same level of authority. At each instance, any node that has data to send uses a procedure defined by the protocol to decide on where to send it. Each node in the network can compute and generate its private key in the ECC, which are integers in the range of the curve's field size, which is 256-bit integers.

This study uses the elliptic curve digital signature algorithm (ECDSA) on the secp256k1 curve, which is mostly used in cryptocurrency systems. secp256k1 is constructed in a special non-random way which allows for especially efficient computation. In many cases, a sufficiently optimized implementation of secp256k1 is 30% or more faster than other known EC curves. secp256k1, often referred to as the Koblitz curve, has six domain parameters specified as a sextuple T = (P, a, b, G, n, and h) over a curve $\boldsymbol{E}(\mathbb{F}_p)$ defined by Equation (1) [32].

Each node in the group computes a new point (public key) on the curve by taking the scalar product between its private key and the reference point on the curve. The protocol that we have used to agree on a common key in a distributed way was introduced in [33] as an extension of the foundational paper by W. Diffie and M. Hellman. The protocol runs as it is shown in Algorithm 1.

Algorithm 1: Cyclic ECC Key generation and distribution

```
1.  Node $N_g$ broadcast beginning of keying process
2.  Initialize $P_{i-1} \leftarrow G$
3.  Choose node j to generate its shared key $P_j$
4.  while i <= $N-1$ do:
5.      If $i \neq j$
6.          Send public key $P_{i-1}$ to node $i$
7.              Node $j$ computes new: $P_{i-1} \leftarrow r_i P_{i-1}$
8.  End
9.  Send public key $P_{i-1}$ to node $j$
10. node $j$ computes it shared key as: $P_j = r_j.P_{i-1}$
11. repeat step 2 to 10 until all nodes generate their shared keys
```

Steps (1–2). Initially, GHi (N_g) prepares a broadcast message M1 and transmits it to all the members in the network, requesting to abort transmission for a period of time which is specified by the protocol for key initialization.

Steps (3–8). Group head initializes key exchange by sending point P_{i-1} to node I, which is the generator point G on the EC curve. The group head implements a cyclic public key P exchange with all other member nodes in the network except j. Hence, all the nodes are involved in the key exchange process by receiving a point P_{i-1} from the N_g, and by computing a new point using its private key based on EC scalar multiplication as shown in the formula (2), where r_i is the private key of the node i and P_{i-1} is the public key computed by the previous node $i-1$.

$$P = \prod_{i=1,\ i\neq j}^{N-1} r_i P_{i-1} \tag{4}$$

Steps (9–10). After the cyclic exchange is complete; node j receives the last computed public key P from N_g. It then uses it to compute its own shared key, as shown using Equation (5).

$$P_j = r_j.P \tag{5}$$

Step (11). In this step, the process will continue until all the nodes in the network compute their shared key (see Figure 3).

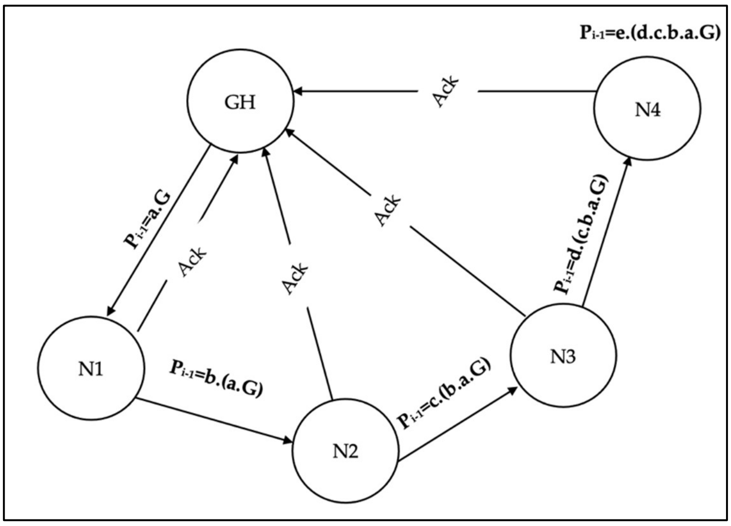

Figure 3. Key exchange process depicting how node N4 computes its shared key.

For example, with five nodes (GH, N1, N2, N3, and N4), having corresponding private keys $a,\ b,\ c,\ d,\ and\ e$, respectively, if GH is the group head node, according to Figure 3, the shared keys (P_{n1} and P_{n2}) for node N1 and N2 is generated and distributed in the following order.

$$P_{n1} = \left(r_{gh}.r_{n2}.r_{n3}.r_{n4}.G\right).r_{n1}$$
$$P_{n2} = \left(r_{gh}.r_{n1}.r_{n3}.r_{n4}.G\right).r_{n2}$$

where G is the generator point on the curve E, r_{n1}, s is the corresponding integer private key for node N1, and "." is scalar multiplication.

We propose a family of distributed and localized group rekeying methods to solve the group rekeying issue for wireless sensor networks as shown in Algorithm 2 Table. When there is a change in group membership, the group key must be updated and redistributed to the rest of the nodes in the network in a way that is safe, dependable, and done in a timely manner. The group rekeying strategy in this paper relies on independence while generating a new set of keys to avoid any kind of prediction by intruders. In addition, the routing table is recalculated using the modified LEACH routing protocol.

Algorithm 2: EC Key Generation

```
    Start
1   Initialize all the buffers in the nodes, i.e., $P_{mid} - = -1$, NN, P, $G$ and $x_i$
2   Choose a node to start keying
3   Generate new message with fields TYP=0, NC=0, $P_x$=$G_x$, $P_y$=$G_y$,$S_{id}$
4   Set the new message Id $M_{id} = P_{mid} + 1$
5   Compute the new points using EC scalar multiplication
6   Increment NC by 1 and choose the next destination node $D_{id}$ sequentially.
7     While $D_{id}$ are not completely used up, do the following:
8           Route the message over the network and performs the following at each new hop
9             If $P_{mid}$ is equals to $M_{id}$ (i.e., old message hoping over)
10                Route the message over the network to another node
11          If $P_{mid}$ is not equals to $M_{id}$ (i.e., new message arrival)
12            Update the points on the curve $P_x$ and $P_y$ such that $P_x = x_i P_{x\_old}$ and $P_y = x_i P_{y\_old}$
13              Increment NC by 1
14            Check if NC is equals to NN-1 (i.e., destination node after visiting all other nodes)
15                Return the shared key for the node i as $S_{key} = P_x\ or\ P_y$
16                Repeat step 3 to 18
17            If NC is not equals to NN-1 (i.e., new message arrives at non-destination node)
18                Repeat step 7 to 18
19    stop
```

3.2. Enhanced LEACH Routing Protocol

One of the major shortcomings of such a protocol is that no consideration is made of the residual energy of the node when selecting a new cluster head, and a new cluster head is only selected after a complete round. If a chosen cluster header dies midway through the round, all the aggregate data for that cluster header is lost, and no data can be received or transmitted with success until the next round when a new cluster head is chosen. Another drawback of LEACH is that a cluster head is chosen randomly or based on the cluster with the highest energy, but this selection process does not take into consideration the distance between the cluster and the sink or the package size, which determine the amount of energy required to transmit the message. A node with the highest energy may not be the most qualified node if it is the farthest away from the base station. In addition to this, as a summary, usually the parameters that define if the CH needs to be replaced are related to not reaching the minimum level of energy needed to gather all the packets sent from the leave nodes or light nodes or sensor nodes and prepare them to submit this data via regular links with the power consumption required. Another parameter that might trigger a replacement request is the detection of an Advanced Persistent Threat (APT) affecting the CH. This can be easily detected if we use DLTs, which refers to a distributed ledger technology that is a database administered by a group of users spread out across multiple nodes [34]. With smart contracts, which are simply programs stored on a blockchain that run when predetermined conditions are met [34]. Under these circumstances, the whole group of nodes is set into an alert mode in which an urgent rekeying operation will be triggered.

To address these problems, we proposed two modifications to the original LEACH protocol as follows:

Before transmitting a packet, the cluster header computes the energy required to transmit the packet, E_{tx}. If the required energy to transmit a packet, E_{tx}, is less than or equal to the residual energy ($E_{residual}$) of the node (i.e., $E_{tx} \leq E_{residual}$), then the cluster header will relinquish its status and a new cluster header will be chosen. The aggregated data stored in the old cluster header should be transferred to the new cluster header. For a packet of size $PacketLength$, the required transmit energy can be computed using Equations (6) and (7).

$$E_{tx} = S_{Etx} \times PacketLength + S_{Efs} \times d^2 \tag{6}$$

$$E_{tx} = S_{Etx} \times PacketLength + S_{Emp} \times d^4 \tag{7}$$

where S_{Etx} is base station transmitter energy parameter and S_{Emp} and S_{Efs} are the energy parameters for the radio transmitter type in the multipath and free space models, respectively. The parameter d is the distance between the node and the base station S. Using Euclidean distance, the distance between node i and base station S can be calculated using Equation (8).

$$d_i = \sqrt[2]{(x_i - x_s)^2 + (y_i - y_s)^2} \tag{8}$$

where x any x are the two-dimensional coordinates of the node or the base station. The cluster header is chosen based on the distance energy-factor F_{ed}, defined using Equation (2). The node with the highest F_{ed} is always chosen as the cluster header. This ensures that the distance between the node and the base station is also considered in the selection process as against the random selection procedures or using nodes with the highest energy only in the LEACH protocol.

$$E_{ed} = E_{residual} - E_{tx} \times \frac{1}{d^2} \tag{9}$$

Moreover, during the cluster head selection process, only nodes that satisfy the requirement that their residual energy $E_{residual}$ is greater than transit energy E_{tx} are considered as a potential cluster header candidate.

3.3. An Extension to the Proposal: DLT as a Means to Create Trust-Based Environments

In the scenario we are moving into, we consider that the network nodes are not evenly distributed throughout the study area, but rather are homogeneously distributed in small areas, and these areas are heterogeneously distributed in a geographical area where communication and reliability of the data collected is already a challenge. In this extension, we propose an extension to the proposal we have made which consists of using a lightweight DLT to create clusters in which each node assumes that its group of communicating nodes is trusted. As shown in Figure 4:

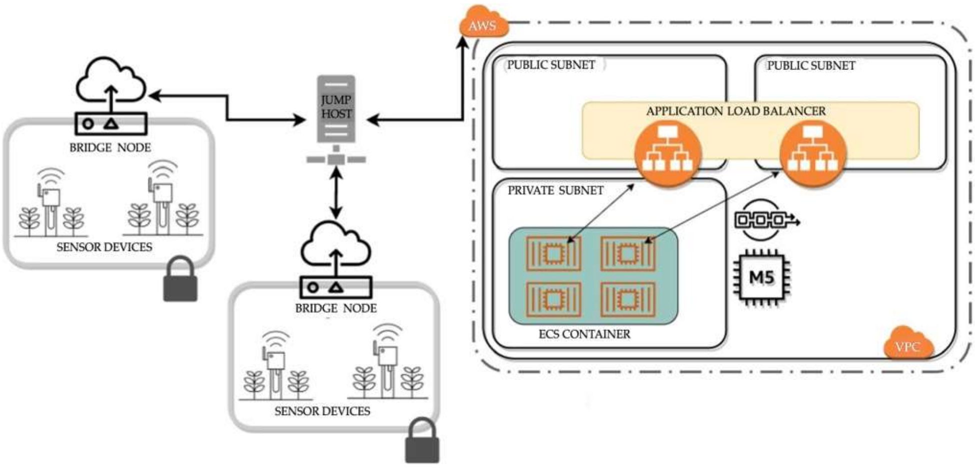

Figure 4. DLT and AWS EC2 architectures.

As can be seen in Figure 4, we create two levels of communication. In the first level (device farm), the nodes use the proposed cryptographic protocol to create a secure group. These nodes communicate with the designated CH (which may vary according to the modification we have proposed to the LEACH protocol). The CH nodes are part of a second level where the trustworthiness is imposed by the DLT itself. Communications between CHs is guided through the DLT update protocols, and the messages exchanged are supported by Message Queueing Telemetry Protocol (MQTT), a machine-to-machine protocol commonly used in the IoT paradigm to message end exchange data between IoT devices [35].

This second layer, which connects through IPFS to the CH nodes, allows us to make use of the smart contracts stored in the EWS instances that receive the information from all the CHs. A smart contract has been developed to analyze the coherence of the data received by each CH according to the temporal sequence.

A smart contract is triggered by the occurrence of a predefined event, such as a CH recording inconsistent behaviour. The smart contract relies on the ability to store information in the DLT itself, specifically in a storage system called the state-database, which stores what is necessary for the execution of the contract. The result of the execution of a contract modifies the content of the database and is stored in the DLT blockchain. It is most common to find contracts designed to be stored sequentially in Amazon Elastic Cloud 2 (EC2) allowing developers to customize the resources of nodes based on requirements and provide an effective way to monitor the project [36]. Recently, it has become possible to launch more than one thread successfully. In our case, we show how this extension means the ability to reduce the power consumption in the CHs since the sending of information is optimized thanks to IPFS (a file system distributed among all the CHs) and how the existence of a contract that analyses the behavior of the CHs, in addition to gathering all the data sent from each CH to later be inserted into Big Data pipelines, means that the performance is not affected at all. Figure 5 shows how we were able to process up to eight CH concurrently in the EC2 instance.

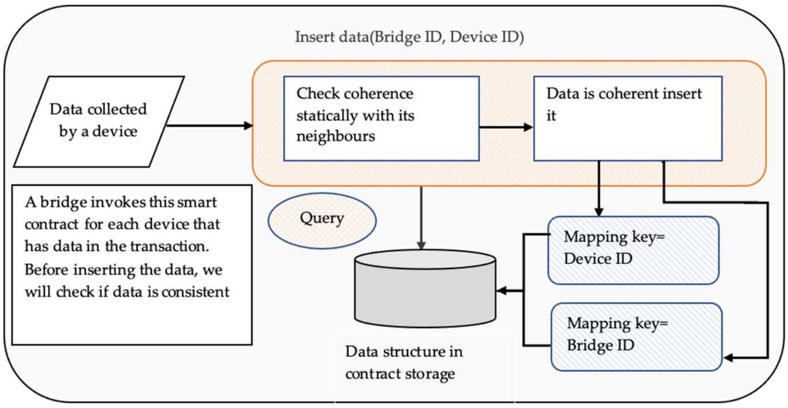
Insert data(Bridge ID, Device ID)
Data collected by a device
Check coherence statically with its neighbours
Data is coherent insert it
A bridge invokes this smart contract for each device that has data in the transaction. Before inserting the data, we will check if data is consistent
Query
Mapping key= Device ID
Mapping key= Bridge ID
Data structure in contract storage

Figure 5. Device smart contract.

4. Simulation Setup and Experimental Results

In this section, it is important to note that, traditionally, the relationship between the security mechanisms applied to the cluster of nodes is inversely proportional to the energy savings in each node of the network. The use of AES has been almost constant in these types of configurations, so using the proposed cryptographic methods not only represents an advantage in processing time, as shown in the experimental results, but also represents a clear energy saving in each node and the assurance that an additional layer of security is applied to protect the CH.

To test the rekeying environment, we have configured physical nodes following the specifications described in Table 2.

Table 2. Simulation parameters.

Component	Value
Processor	Quad Core 1.2 GHz Broadcom BCM2837 64-bit CPU
Internal RAM	1 GB
Antenna	BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board.

The scope of the paper is not the implementation of an efficient node, but instead a node to test the rekeying and strength of the protocol. To this end, we have used the physical nodes described in Table 2, on which we have installed Ubuntu 18.04 LTS and all the custom drivers. We are aware that by installing a kernel designed using YOCTO [23,24,37,38], the savings in power consumption are close to the real environment of an embedded sensor device. To test the strength of the key management protocol, it needed the deployment of a higher number of nodes, such as the one described in Table 2, so we have emulated them in the physical devices. To test the rekeying speed operation, we have used up to eight physical nodes. In addition, we have scaled up to 1024, but from eight we have emulated the processors using threads. The results of the rekeying process are described in the results section. To test the proposed protocol and the energy consumption fingerprint, we have emulated a complete node using MATLAB. One of the challenges in evaluating the performance of modified routing protocols and the security properties of the methods used in this paper in contrast to other relevant algorithms is simulator selection. To emulate wireless networks, many simulation systems do exist and all of them are equally valid, such as Objective Modular Network Testbed in C++ (OMNET++), Network Simulator2 (NS2), and MATLAB [25,39]. As previously mentioned, the cryptographic foundations were tested on real nodes. Deploying an extensive infrastructure of communicating nodes needs the support of emulation tools which provide information of interest to evaluate the approach being presented. In our work, we used MATLAB R2022a to test the performance of the WSN under various scenarios and then compare the results with other relevant works that use similar tools. The configuration used for the extension of the proposal (using DLT to make communications restricted to a closed group) is described below in Table 3:

Table 3. Physical components of DLT.

Component	Value
Instance	AWS EC2 M5
Processors	Intel Xeon Platinum 8175M 3.1 GHz (AVX-512)
Internal RAM	1 GB
Instance store	NVMe SSD
Distributed FS	Interplanetary File System

4.1. ECC Key Distribution and Management Experimental Results

We have tested the encryption approach proposed using different parameters, including power consumption, hop counts, transferred number of messages, and packet header size. In this part, we present the experiment results. The following table depicts how the rekeying operation performs efficiently on a conventional embedded processor that is running a full operating system instead of a customized and optimized image. In Table 4, the header specifies the number of computing nodes participating in the cluster. As we have used eight physical devices, those scenarios (marked with an asterisk) in which we needed a number higher than eight computational threads inside each device were used. The threads were evenly distributed among the physical devices. For example, in the case where we used 16 the number of threads obviously exceeded the number of physical devices. In this case, we created two threads per device. Each thread emulated a node. In the last case in which we tested 1024 threads, a total of 128 computational threads were run on each processor.

Table 4. Rekeying operation performance time is calculated in milliseconds.

No. of Threads (Nodes)	8	16	32	64	128	256	512	1024
Time consumption (ms)	0.000548	0.0011	0.001478	0.004942	0.013898	0.19402	0.051176	0.14292

For n number of nodes in the sensor network, computational complexity of the cyclic ECC key generation and distribution in the algorithm one table can be computed as follows using generator point G, the elliptic curve, and r_i as the private key for node i. To generate a shared key for a node, scalar multiplication has to be performed as indicated in Equation (10).

$$P_i = (r_1.r_2.r_3.r_4 \ldots\ldots\ldots.r_n).G \tag{10}$$

where "." operator is an EC scalar multiplication. Hence for n nodes the total complexity can be calculated using Equation (11).

$$\mathcal{O}(n(r^n.G)) \tag{11}$$

4.2. Improved LEACH Protocol Experimental Results

Regarding the energy consumption of each node, when we are applying the key sharing in groups and the head node changes when the key conditions for it are given, we can see in the following graphs how they behave. In this simulation, we have used an example cluster containing 50 nodes. In this cluster, 10 rounds of the protocol have been performed. A total of 10% of the nodes in the network have been configured as CH. After the simulation, 37 nodes out of 50 were excluded as they did not reach the minimum energy threshold. Table 5 presents the experimental parameters associated with the sensor nodes used to obtain the simulation results.

Table 5. Parameters used for proposed improved LEACH protocol.

Definition	Average Value
Size of testbed (number of nodes)	50,100 homogenous
Number of base stations (Bs)	1
Initial energy for each sensor	0.005 j
Radio circuitry energy dissipation, Eelec	50 nj/bit
Energy dissipation of amplifier in free-space, Efs	10 pj/bit
Energy dissipation of amplifier in multipath, Emp	0.0013 pj/bit
Energy consumption for data aggregation, Eda	5 nj/bit
Global testbed area	$n \times n$
Local area (cluster size)	n/nc
Time	10 rounds
Packet size	400-bits
Message size	328-bits
Encryption key length	256-bits

The result shows the speed-up from the scenario in which we process the status of each CH sequentially to the situation in which we execute up to eight threads. In each scenario, we were able to process up to 10,000 transactions, where a transaction means requests from the CH to the DLT.

In order to confirm these analytical findings, we performed MATLAB simulations on a network of 50 and 100 nodes respectively, we changed the number of clusters from 1 to 4, and then ran both protocols for 1500 simulated rounds. The nodes were placed randomly and equally in each cluster, and we made no restrictions on the distance between the nodes and their cluster heads as shown in Figure 6.

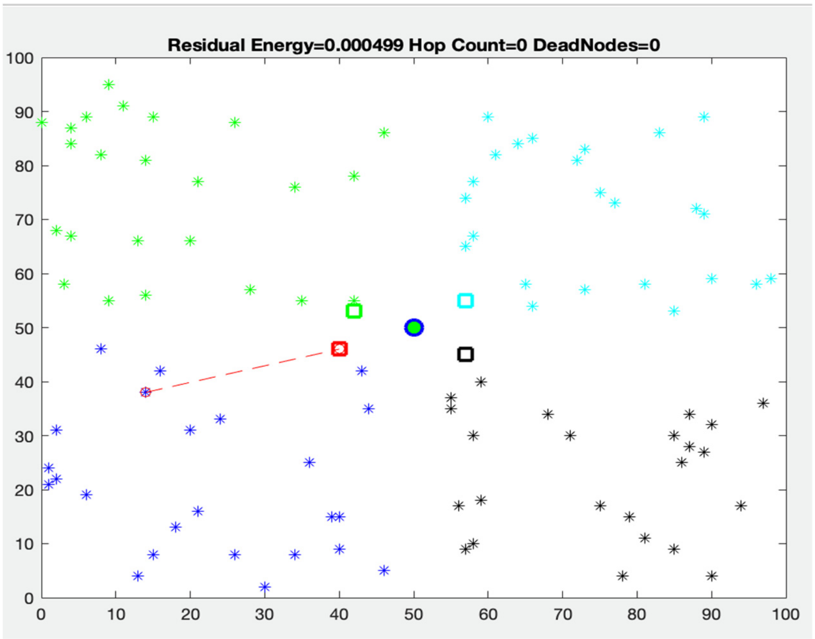

Figure 6. Topology of wireless sensor networks for a 100-node network.

Figure 7 illustrates the difference between the LEACH and modified LEACH routing protocols. The number of dead nodes in the conventional LEACH from round 5 starting up is clearly visible when compared to the proposed protocol. In addition, the last dead node of the proposed protocol extends up to ten rounds, while the last dead node in the original LEACH is in round nine. The enhanced threshold condition in the proposed protocol is mainly responsible for the network's extension stability. Comparisons are made between proposed protocols for various network sizes to determine the networks' lifespan and the impact of increasing node separation.

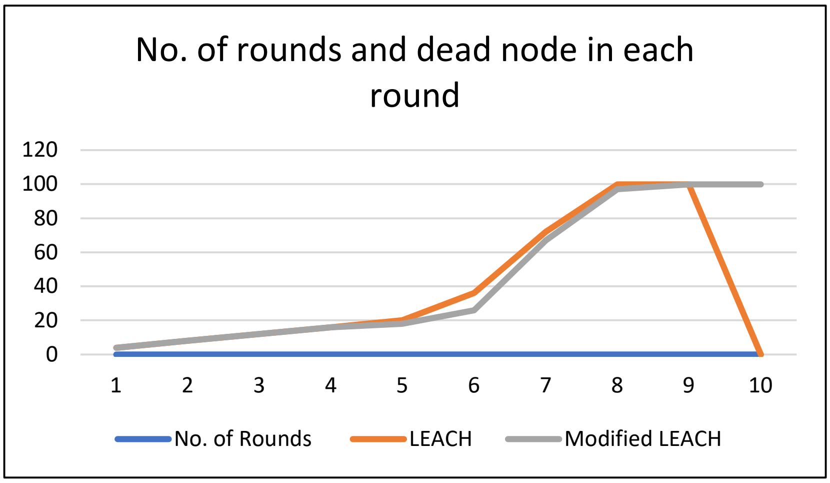

Figure 7. Dead nodes vs. rounds for a 100-node network.

Figure 8 indicates that the time consumption to compute a key in each round takes less compared to the original LEACH in the network. This is mostly due to the fact that the distance factor and remaining energy are taken into account during the selection of cluster heads. In the proposed protocol, closer nodes are chosen as head nodes more often than nodes that are farther apart. This means that the total amount of time needed for each round is less.

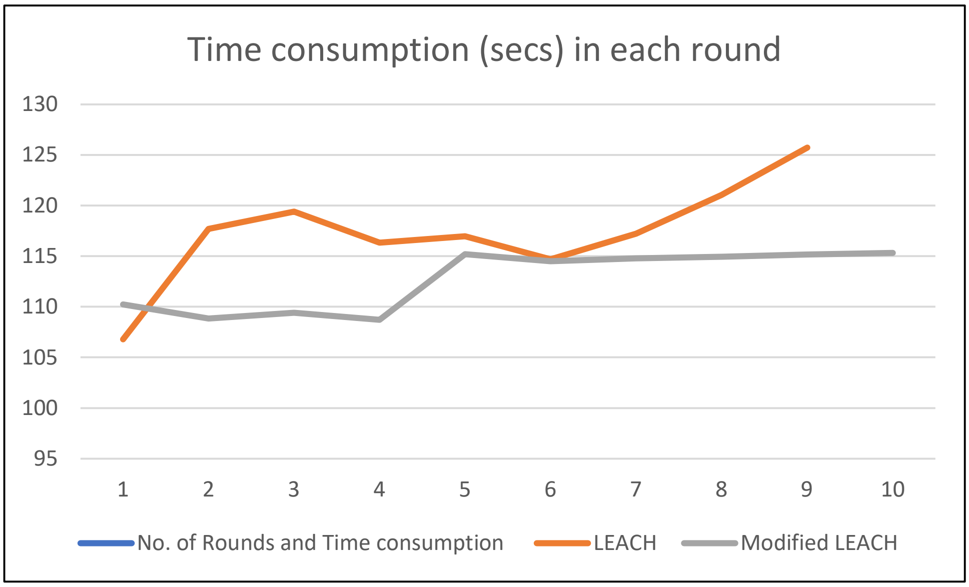

Figure 8. Overall time consumption per round for a 100-node network.

In Figure 9, the links between the number of hops that are required to find CH are depicted. The most used hop in each round is done by the LEACH protocol, even though the percentage difference is too small. As the network in terms of size increases, the number of hops will also increase.

The results of the simulation and analysis reveal that the proposed protocol is capable of achieving a high level of security, outperforming other schemes that had been offered earlier, and improving the effectiveness and efficiency of transmitted data among sensors in a network.

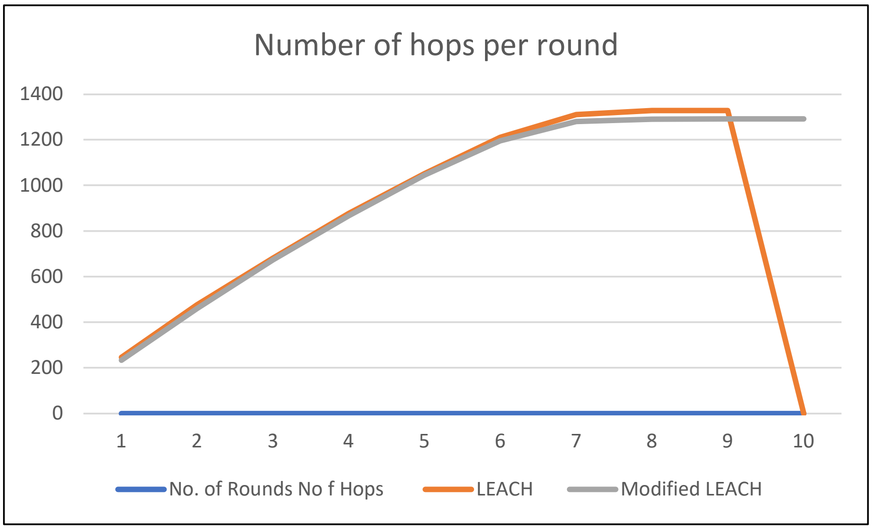

Figure 9. Overall hop count per round for a network of 100 nodes.

Figure 10 presents a performance comparison between our proposed enhanced LEACH protocol (E-LEACH), the original LEACH (LEACH), and another version of the improved LEACH protocol (E-DEEC) proposed in reference [26,40]. The simulation was conducted using the same node parameters presented in Table 5 In each case, 100 sensors were placed randomly in the field and the positions of the nodes were maintained throughout the experiment. The simulation was set up to run for 1500 rounds and the corresponding number of dead nodes were recorded after each round as depicted in Figure 10. It can be seen from the figure that in the original LEACH protocol, instances of dead nodes started to appear at around 600 rounds and all the nodes died at about 1100 rounds. Meanwhile, the E-DEEC of reference [26,40] has improved on the original LEACH by prolonging the instance of the appearance of the first dead node to around 900 rounds, but eventually all nodes die at about the same time with the original LEACH. However, the proposed E-LEACH has improved on both the initial incidence of dead nodes (around 904 rounds) and after 1496 rounds there are still nodes that are alive which is an indication of better energy management and extends the lifespan of the network.

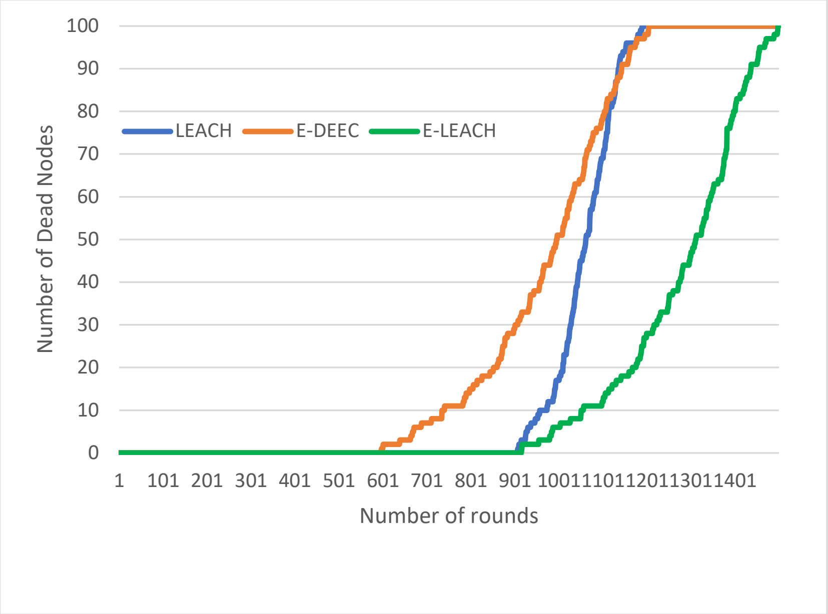

Figure 10. Performance comparison of Enhanced LEACH with other versions.

4.3. Discussion

The important parameters that determine the impact of our proposal are node efficient energy utilization, network lifetime and packet transmission reliability, and secured communication among nodes. With respect to power consumption, it is worth considering that this paper was developed in an agribusiness context where sensors might be operated by a single battery or aided by solar power charging cycles. This context is so particular that the nodes are not distributed with homogeneity in a perfectly isolated area. For defined areas, or plots, where sensors are deployed to collect information about the state of the environment and the crop, for each plot, or for each specified area, we can have a set of nodes. The distribution in farms can be such that we can find different factors such as nodes of other farmers, shadow zones in communications due to adverse weather phenomena, recalibration of any of the sensor nodes due to hostile conditions in which the nodes are deployed, etc. These are, in short, circumstances that invite reconsideration of the dynamic re-election of a head-end (CH) node not only when it drains all its energy. In addition, the possibility of collision with nodes from other plots suggests the use of the proposed cryptographic method to avoid data contamination, together with the reduced power consumption. Finally, an extension to the method based on DLT and IPFS will be proposed to federate sensors into two levels. We will use this technique to guarantee the minimum packet loss in transmissions [32,41].

Regarding energy: we evaluated the evolution in the draining of energy as long as the algorithm evolves over time. The transmissions from node to node are fixed and the conditions are defined in Table 1. Sensor nodes operating in the field are optimized to gather data and even to synthetize by comprising it into a series. Furthermore, the communication rate is constant, as well as the size of the packet. The amount of energy required to transmit a packet depends on the routing strategy and the packet size. We propose a first approach where nodes are clustered using modified LEACH with a dynamically elected CH and a second proposal where nodes are grouped into two layers: the first one where sensor nodes are light nodes of a DLT and the CH is the one that keeps the log of the DLT; the second one grouping all the CH into a different DLT (second layer of trust where all the logs are shared and the smart contracts are executed for the sake of the packet transmission reliability). Regarding network lifetime, we can consider that the network has expired its lifetime when the CH node has wasted all of its potential energy to transmit data packets or even to receive small control packets. Even though it still has power, we consider it wasted if the operative threshold (oTh) is not met. It is also worth noting that the simulation presented above is applied to a two-dimensional space where the surfaces where nodes are placed are relatively flat. For a surface where there are obstructions and an uneven surface level, a three-model might have sufficed.

Packet transmission reliability concerns the number of times that a packet is not acknowledged at its destination or fails to hit its target node over the network. Packet loss in an IoT transmission can be due to a myriad of issues, from farm machinery operating near the nodes in the 2.4 GHz band to DoS attacks preventing nodes from sending data to the CHs. In all cases, packet loss impacts node power consumption. Since as with all radio transmissions (and we do not act on these parameters as it is not the scope of the proposal we present), power is increased if the link is lost. If nodes come under attack, it is easy for an attacker to identify the CH (since it is the destination of all transmissions from nodes in the area) and try to assemble it to bring down the network. Herein lies one of the strengths of our protocol: on the one hand, it prevents packet injection and is able to make the CHs identify quickly which packets not to process (affecting the savings in energy consumption); on the other hand, it prevents an attacker from rescuing useful information from a transmission; and finally, before the imminent fall of a node, another one is immediately selected. In the extension of the proposal, where DLT is used, the fall of a node or its compromise can alert (since smart contracts can identify the pattern of an attack) and disconnect all nodes until the source of the attack is solved or identified.

5. Conclusions

In recent years, the Internet of Things (IoT) and wireless sensor networks (WSNs) have stimulated academics and gained great attention due to their ability to serve a wide range of applications. Security and privacy, in addition to routing tasks, are the most challenging areas of the WSNs due to the constraints of the sensor devices. We present in this paper an efficient key management method to generate, distribute, and re-key processes using elliptic curve cryptography for enabling security services in challenging WSN scenarios. The results show secure and reliable connections with fewer overheads and time consumption. Details about the key generation, distribution, and rekeying processes, in addition to the proposed enhanced LEACH routing protocol, have been presented. Finally, a performance evaluation has been conducted based on metrics such as the number of rounds, energy consumption, time consumption, dead nodes, and hop counts to demonstrate the method's effectiveness compared to other routing protocols in WSN networks.

Despite the impressive and promising results obtained from the proposed method, the model considers only a two-dimensional topology for the sensors' placement, where this assumption may not be properly applicable in places where sensors are placed on an uneven surface. Another limitation is that a homogenous setting was considered where all sensors have the same properties, whereas in reality there may be cases of heterogeneity where sensors of different characteristics are used to monitor different physical events. In future work, an experimental test of the proposed solution in more complex WSN systems to analyse the robustness of the method against some types of attacks, as well a 3D implementation, are planned.

Author Contributions: Conceptualization, S.M.H., J.A.L.R. and A.M.A.; methodology, S.M.H. and J.A.L.R.; software, S.M.H. and A.M.A.; writing—original draft preparation, S.M.H., J.A.L.R. and A.M.A.; writing—review and editing, S.M.H. and J.A.L.R.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Kouicem, D.E.; Bouabdallah, A.; Lakhlef, H. Internet of things security: A top-down survey. *Comput. Netw.* **2018**, *141*, 199–221. [CrossRef]
2. Bertino, E. Data Security and Privacy in the IoT. In Proceedings of the 19th International Conference on Extending Database Technology (EDBT), Bordeaux, France, 15–18 March 2016.
3. Khan, M.A.; Salah, K. IoT security: Review, blockchain solutions, and open challenges. *Future Gener. Comput. Syst.* **2018**, *82*, 395–411. [CrossRef]
4. Birajdar, D.M.; Solapure, S.S. LEACH: An energy efficient routing protocol using Omnet++ for Wireless Sensor Network. In Proceedings of the 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 10–11 March 2017; pp. 465–470. [CrossRef]
5. Rahmadhani, M.A.; Yovita, L.V.; Mayasari, R. Energy Consumption and Packet Loss Analysis of LEACH Routing Protocol on WSN Over DTN. In Proceedings of the 2018 4th International Conference on Wireless and Telematics (ICWT), Nusa Dua, Bali, Indonesia, 12–13 July 2018; pp. 1–5. [CrossRef]
6. Oreku, G.S.; Pazynyuk, T. *Security in Wireless Sensor Networks*; Springer: Vienna, Austria, 2016.
7. Sohraby, K.; Minoli, D.; Znati, T. *Wireless Sensor Networks: Technology, Protocols, and Applications*; John Wiley and Sons: Hoboken, NJ, USA, 2007; pp. 15–18, ISBN 978-0-471-74300-2.
8. Devika, B.S.R.; Sivasubramanian, T. Survey on Routing Protocol in Wireless Sensor Network. *Int. J. Eng. Technol.* **2013**, *5*, 6. [CrossRef]
9. Javed, M.U.; Rehman, M.; Javaid, N.; Aldegheishem, A.; Alrajeh, N.; Tahir, M. Blockchain-Based Secure Data Storage for Distributed Vehicular Networks. *Appl. Sci.* **2020**, *10*, 2011. [CrossRef]
10. Mohamad, M.B.; Noor Hassan, W.H. Current research on Internet of Things (IoT) security: A survey. *Comput. Netw.* **2019**, *148*, 283–294. [CrossRef]

11. Xu, W.; Trappe, W.; Zhang, Y.; Wood, T. The feasibility of launching and detecting jamming attacks in wireless networks. In Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Urbana, Champaign, IL, USA, 25–27 May 2005; pp. 46–57.
12. Mishra, A.K.; Turuk, A.K. A Key Renewal Model for Wireless Sensor Network Under Node Capture Attack. In Proceedings of the 2011 Fourth International Conference on Emerging Trends in Engineering & Technology, Port Louis, Mauritius, 18–20 November 2011; pp. 301–305.
13. Jokhio, S.H.; Jokhio, I.A.; Kemp, A.H. Node capture attack detection and defence in wireless sensor networks. *IET Wirel. Sens. Syst.* **2012**, *2*, 161–169. [CrossRef]
14. Bhattasali, T.; Chaki, R. A Survey of Recent Intrusion Detection Systems for Wireless Sensor Network. In *Advances in Network Security and Applications*; Wyld, D.C., Wozniak, M., Chaki, N., Meghanathan, N., Nagamalai, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 268–280.
15. Prodanović, R.; Rančić, D.; Vulić, I.; Zorić, N.; Bogićević, D.; Ostojić, G.; Sarang, S.; Stankovski, S. Wireless Sensor Network in Agriculture: Model of Cyber Security. *Sensors* **2020**, *20*, 6747. [CrossRef] [PubMed]
16. Pirzada, A.A.; McDonald, C. Circumventing sinkholes and wormholes in wireless sensor networks. In *Proceedings of the International Workshop on Wireless Ad-hoc Networks, 2005*. Available online: https://www.researchgate.net/profile/Chris-Mcdonald-2/publication/250774526_Circumventing_Sinkholes_and_Wormholes_in_Wireless_Sensor_Networks/links/54da83df0cf261ce15cd4e69/Circumventing-Sinkholes-and-Wormholes-in-Wireless-Sensor-Networks.pdf (accessed on 21 August 2022).
17. Ali, M.; Nadeem, M.; Siddique, A.; Ahmad, S.; Ijaz, A. Addressing Sinkhole Attacks in Wireless Sensor Networks—A Review. *Int. J. Sci. Technol.* **2020**, *9*, 406–411.
18. Patil, H.K.; Chen, T.M. Chapter 18—Wireless Sensor Network Security: The Internet of Things. In *Computer and Information Security Handbook*, 3rd ed.; Vacca, J.R., Ed.; Morgan Kaufmann: Boston, MA, USA, 2017; pp. 317–337.
19. Wallgren, L.; Raza, S.; Voigt, T. Routing attacks and countermeasures in the RPL-based internet of things. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 794326. [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 21st International conference on computer communications and networks (ICCCN), Munich, Germany, 30 July–2 August 2012; pp. 1–5.
21. Di Pietro, R.; Guarino, S.; Verde, N.V.; Domingo-Ferrer, J. Security in wireless ad-hoc networks—A survey. *Comput. Commun.* **2014**, *51*, 1–20. [CrossRef]
22. Yousefpoor, M.S.; Barati, H. Dynamic key management algorithms in wireless sensor networks: A survey. *Comput. Commun.* **2019**, *134*, 52–69. [CrossRef]
23. Zhang, L.; Wu, Q.; Qin, B.; Domingo-Ferrer, J. Identity-Based Authenticated Asymmetric Group Key Agreement Protocol. Springer: Berlin/Heidelberg, Germany, 2010; pp. 510–519.
24. Whitfield Diffie, M.E.H. New Directions in Cryptography. In *Secure Communications and Asymmetric Cryptosystems*, 1st ed.; Simmons, G.J., Ed.; Routledge: New York, NY, USA, 1982; p. 38.
25. Steiner, M.; Tsudik, G.; Waidner, M. Diffie-Hellman key distribution extended to group communication. In Proceedings of the 3rd ACM Conference on Computer and Communications Security, New Delhi, India, 14–15 March 1996.
26. Steiner, M.; Tsudik, G.; Waidner, M. Key agreement in dynamic peer groups. *IEEE Trans. Parallel Distrib. Syst.* **2000**, *11*, 769–780. [CrossRef]
27. Reto Schnyder, J.A.L.-R. Joachim Rosenthal, Davide Schipani, An Active Attack on a Multiparty Key Exchange. *arXiv* **2018**, arXiv:1509.01081. [CrossRef]
28. Ateniese, G.; Steiner, M.; Tsudik, G. New multiparty authentication services and key agreement protocols. *IEEE J. Sel. Areas Commun.* **2000**, *18*, 628–639. [CrossRef]
29. Seok, B.; Sicato, J.C.S.; Erzhena, T.; Xuan, C.; Pan, Y.; Park, J.H. Secure D2D Communication for 5G IoT Network Based on Lightweight Cryptography. *Appl. Sci.* **2020**, *10*, 217. Available online: https://www.mdpi.com/2076--3417/10/1/217 (accessed on 21 March 2021). [CrossRef]
30. Heinzelman, W.B.; Chandrakasan, A.P.; Balakrishnan, H. An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. Wirel. Commun.* **2002**, *1*, 660–670. [CrossRef]
31. Kamarudin, L.M.; Ahmad, R.B.; Ndzi, D.L.; Zakaria, A.; Kamarudin, K.; Ahmed, M.E.E.S. Simulation and analysis of LEACH for wireless sensor networks in agriculture. *Int. J. Sens. Netw.* **2016**, *21*, 16–26. [CrossRef]
32. Pote, S.; Sule, V.; Lande, B.K. Arithmetic of Koblitz Curve Secp256k1 Used in Bitcoin Cryptocurrency Based on One Variable Polynomial Division. In Proceedings of the 2nd International Conference on Advances in Science & Technology (ICAST), Mumbai, India, 8–9 April 2019. Available online: https://ssrn.com/abstract=3367674 (accessed on 10 January 2021).
33. Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 7 January 2000; Volume 2, p. 10. [CrossRef]
34. Ellul, J.; Galea, J.; Ganado, M.; McCarthy, S.; Pace, G.J. Regulating Blockchain, DLT and Smart Contracts: A technology regulator's perspective. *ERA Forum.* **2020**, *21*, 209–220. [CrossRef]
35. Vatsal Gupta, S.K.; Turk, N. MQTT protocol employing IOT based home safety system with ABE encryption. *Multimed. Tools Appl.* **2021**, *80*, 19.

36. Marozzo, F. *Infrastructures for High-Performance Computing: Cloud Infrastructures*; Elsevier: Amsterdam, Netherlands, 2019; pp. 240–246.
37. Yocto Project | Open Source Embedded Linux Build System, Package Metadata and SDK Generator. Available online: https://www.yoctoproject.org/ (accessed on 3 February 2017).
38. Navik, A.P.; Muthuswamy, D. Dual band WLAN gateway solutions in Yocto Linux for IoT platforms. In Proceedings of the 2017 International Conference on Internet of Things for the Global Community (IoTGC), Funchal, Portugal, 10–13 July 2017; pp. 1–3. [CrossRef]
39. Antonio Virdis, M.K. *Recent Advances in Network Simulation The OMNeT++ Environment and Its Ecosystem (EAI/Springer Innovations in Communication and Computing)*; Springer: Cham, Switzerland, 2019.
40. Saini, P.; Sharma, A.K. E-DEEC-Enhanced Distributed Energy Efficient Clustering scheme for heterogeneous WSN. In Proceedings of the 2010 First International Conference on Parallel, Distributed and Grid Computing (PDGC 2010), Solan, India, 28–30 October 2010; pp. 205–210. [CrossRef]
41. Zheng, X.; Lu, J.; Sun, S.; Kiritsis, D. Decentralized Industrial IoT Data Management Based on Blockchain and IPFS. In *Advances in Production Management Systems*; Towards Smart and Digital Manufacturing; Lalic, B., Majstorovic, V., Marjanovic, U., von Cieminski, G., Romero, D., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 222–229.