computers

MDPI

Article

A Novel Security Protocol for Wireless Sensor Networks with Cooperative Communication

Abdullah Al Hayajneh [1,*], Md Zakirul Alam Bhuiyan [2] and Ian McAndrew [1]

[1] Department of Doctoral Programs, Capitol Technology University, 1301 Springfield Rd, Laurel, MD 20708, USA; irmcandrew@captechu.edu
[2] Department of Computer and Information Sciences, Fordham University, New York, NY 10458, USA; mbhuiyan3@fordham.edu
* Correspondence: asal-hayajneh@captechu.edu

Received: 6 December 2019; Accepted: 16 January 2020; Published: 20 January 2020

Abstract: This paper builds upon the foundation and clarifies specifications for a necessary security protocol in Wireless Sensor Networks (WSNs) with cooperative communications. It is designed to enhance performance and resiliency against cyber-attacks. Recent literature has shown that developing a WSN with Cooperative Communication greatly increases the performance of the network, but also exposes new vulnerabilities. The technique operates by transmitting packets of data to neighboring relay nodes in a cooperative fashion to reach the destination. In this paper, we consider security issues in WSNs with cooperative communication on each layer of the OSI model: physical layer, data link layer, network layer, service (topology) layer, and application layer. For each layer, we clarify the main task, enumerate the main attacks and threats, specify the primary security approaches and techniques (if any), and discuss possible new attacks and problems that may arise with the use of cooperative communications. Furthermore, we show for some attacks (e.g., jamming, packet dropping, and wormhole) that using cooperative communication improves the network resiliency and reliability. Finally, we propose a security protocol that addresses many of these shortcomings, while outlining the remaining issues that need further work and research.

Keywords: cooperative communication; WSN; security attacks; resiliency

1. Introduction

A wireless sensor network (WSN) is a network of nodes that are dispersed throughout a geographical area and have the ability to read sensor data, such as temperature, sound, vibration, pressure, motion, or pollutants [1]. To produce a large volume of these devices, they need to be small and inexpensive, containing only integral components which include sensing units, processing units, transceiver units, and power units [2]. Due to their size and lack of equipment, they are low resource devices that can not handle extensive computational processing [1]. WSNs have been applied to a variety of fields such as healthcare, wildlife, manufacturing machinery, environmental studies, and military; additionally, they have the potential to enhance the performance and study of all disciplines [3]. At a cursory glance, hiding data may seem irrelevant for some tasks. Experienced attackers may use the sensed data creatively out of malicious intent. For example, a sensor that senses when parts are being moved in a factory may give a competitor information on production volumes. It is also important to protect these devices from receiving falsified information, like wrongly injecting insulin into a patient with a blood-sugar sensor.

When transmitting messages in a wireless environment, signals can be vulnerable to high levels of noise, which can increase packet error rate [4]. To reduce the error ratio, we use a technique known as Cooperative Communication (CC). CC is a network design that allows single antenna nodes to

generate a virtual multi-antenna transmitter by working together and sharing their antennas with their neighboring nodes. This yields a higher transmit diversity and more redundancy in signal transmission. Studies have shown that this strategy is more efficient than the traditional method of single antenna transmission [5] in terms of bit error rate (BER), packet error ratio, and capacity [3,6].

As mentioned earlier, one of the most crucial concerns with applications implementing WSNs is the security, which is actually stymieing widespread adoption. Since nodes leverage the public wireless medium, they become easy targets for adversaries to corrupt legitimate messages or insert new ones. Therefore, a security protocol that proves the integrity of the data (and potentially hides it) is needed to secure the network. The addition of more nodes in a system to relay sensitive information creates more targets to attack and new creative attacking techniques. A malicious node has complete control over relayed data and can even choose not to relay at all [7]. The addition of the CC architecture could make designing a secure protocol more difficult as more nodes will participate in the sending of a message.

To address these issues in WSNs, this paper explores the vulnerabilities at the main protocol layers of the OSI model: physical, data link, network, and application. Regarding the OSI model layers, we will summarize their primary functions and define possible threats. We will then discuss new potential attacks and issues that may occur due to the use of Cooperative Communication. Moreover, we will demonstrate that, for several types of attacks such as jamming, packet dropping, and wormhole, utilizing cooperative communication enhances the network reliability and stability.

A significant amount of research has been developed in the field of wireless communication, more specifically WSNs. The concept of CC has been around for over almost two decades with research primarily focusing on efficiency in terms of routing protocols and energy efficiency. In terms of security, researchers have evaluated the benefits of CC in [8]. Researchers in [9] have developed a protocol that uses cryptographic techniques to provide secrecy to the system. However, this implementation uses asymmetric cryptography and can be too heavy for low resource device implementation. Message authentication schemes have been developed for traditional WSNs [10–12], as well as vehicle area networks (VANETS) [13–15]. However, there no schemes at the time of writing this paper that apply message authentication to WSNs with cooperative communication. Likewise, there are many reputation schemes that have been created for generic ad-hoc and wireless sensor networks [16–20]. One study [21] examines reputation among cooperative sensing in cognitive radio networks, which is different than CC discussed in this paper in that sensors in those networks cooperate to make decisions dynamically based on data, instead of sending data back to a base station, to simulate cognition. The closest scheme on reputation in CC is [22], but it is unclear as to how they define cooperation; they appear to refer to the relaying aspect, rather than the redundancy of messages. Either way, the scheme proposed is only used to detect selfish nodes, while our scheme serves to detect inefficient nodes in general, whether they are malicious, corrupted, or simply defective. This allows the network to re-route around these nodes that have become obstacles and increase overall efficiency.

To the best of our knowledge, none of the previous work in the literature has proposed a protocol to secure CC in wireless networks. The main contribution of this papers is the development of a novel security protocol using a Message Authentication Code (MAC) to verify authenticity of messages, and a reputation table to identify malicious or defective nodes. Finally, an analysis of the protocol is given that justifies its security and value against other methods.

The rest of this paper is structured as follows. The next section provides the background of cooperative communication and its evolution in the field. Section 3 identifies the main OSI model layers and possible new attacks and problems that arise with the use of CC. Section 4 describes the proposed security protocol for WSNs with CC and an analysis will be given in Section 5. Finally, Section 6 presents the results and Section 7 gives the conclusions.

2. Background of Cooperative Communication

The concept of cooperative communication traces back to research enhancing efficiency in mobile networks. These are similar to WSNs in the sense that they both use a wireless medium based on radio frequencies and that they both consist of independent users in the system reporting to an ultimate destination (cellular tower or aggregator, respectively). The driving force that led researchers to develop this technique was the desire to optimize signal-to-noise ratio (SNR) by increasing path diversity in transmission. The concept of user cooperation began with the works of Sendonaris et al. [23,24], where they presented an implementation called code-division multiple-access (CDMA): two users cooperate with each other by sending their information to one another to construct a combined noisy transmission signal, bit by bit, before sending it to the ultimate destination, as well as sending a message directly (as shown in Figure 1) with X being a message. This work was then expanded upon [25,26] to account for power outages and addressed with research concluding that an amplify-and-forward (AF) technique was the optimal option.

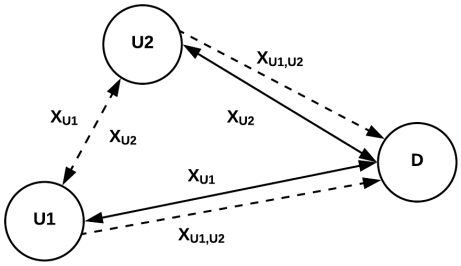

Figure 1. Diagram of user cooperation.

This concept of cooperation among users was taken farther to develop other frameworks, such as coded cooperation [27–30]. With the previous user cooperation model, two users would combine their signals bit by bit to construct the new transmission signal. In this framework, the users instead divide their information into blocks and send a part of the full message to the other user. The user then decodes-and-forwards (DF) this information, along with part of its own information, to the destination. The destination will then have a redundancy of the messages sent to significantly decrease the bit error rate (BER). This idea was then expanded upon [31] in a framework known as space-time cooperation to even further optimize SNR and BER utilizing the same channel (space), but sending the messages at different frames (time).

Cooperation of nodes in a system combat the problem of fading channels [32] as transmission distance increases. The implementation of relays allow transmissions to travel farther distances while maintaining an acceptable signal strength. However, the literature on cooperation at this point only took into account the case in which the nodes have a set number of cooperating partners (usually one partner). Research has shown that diversity gain increases as the number of cooperating relays increases [33]. This began a study to optimize relay selection in a network to not only increase the diversity effects of CC, but to also create a reliable path to spread out overall transmission and lower power consumption for an individual node; the more nodes in a path, the less power required for an individual node in the path [33–39]. Beamforming was used to narrow down the transmission signal to specifically target a node and decrease BER [40]. Without beamforming, other transmissions pass through the signal radius and interfere with the transmission. A smaller signal means a smaller probability of interference. Cooperative beamforming techniques have been applied to cooperative communication to increase overall effectiveness in terms of SNR, BER, and power consumption [41,42].

Due to the low resource nature of wireless sensor devices, power consumption is a paramount concern. Significant research has been dedicated to this subject to try to minimize power outages of sensors. Many techniques have been developed to attack this issue from different angles. Some researchers focus

on MAC protocols and the timing of signal reception, such as [43,44], while others looked at data-driven solutions. A full comprehensive taxonomy up until 2009 and more information on energy conservation in WSNs can be found in [45], as well as [46]. Energy harvesting has become very popular in recent years as research developed techniques to prolong battery life of sensors by having them harvest energy in different ways, either from the environment or through transmissions reception [47–49].

3. Security in Cooperative Communication

The research paper done by Mau and Wu [50] studied the security associated with CC considering a situation where the relay nodes are compromised. The concept of CC was introduced using a model with two relay nodes implementing the decode-and-forward protocol. The researchers demonstrated the possible presence of a passive adversary that would view the transmitted messages, and an active adversary that would modify messages or send garbled signals. They claimed that an application-layer protocol would not be able to track compromised nodes, but we will demonstrate in the paper that our protocol accomplishes this.

In [9], the researchers proposed a security protocol that gave the MIMO system protection from active compromised nodes using a secure key management system as a cryptographic technique. This protocol was tested on a MIMO, instead of a MISO, and uses a process that is high in consumption. Researchers [51] analyzed the issues with CC transmission and sought to improve the current state at the time. They tested the use of relay selection to enhance secrecy and show the efficiency of cooperative transmission.

Likewise, researchers in [52] proposed a framework that analyzes the efficiency of cooperative transmission. The results showed that, when compared to SISO transmission, cooperative MISO has a greater performance in the presence of malicious and compromised nodes in WSNs. In [53], they discussed the potential security issues that cooperation may raise in wireless networks when limited ranged jammers are placed both randomly and strategically. Furthermore, they proposed two security schemes to address those concerns.

As stated, Cooperative Communications helps improve the efficiency of a WSN immensely, but the addition of this design proposes new challenges as well. This paper will examine the security threats that could arise at the main layers of the OSI model for WSNs: physical, data link, network, services (topology), transport, and application. In this section, we will identify the main layers that affect the use of CC and discuss possible new attacks and problems that could arise with the use of Cooperative Communications. We will also show that CC helps with resiliency and reliability for some attacks (e.g., jamming, packet dropping, and wormhole).

The main layers that seem to affect the use of Cooperative Communication are the Physical Layer, Data Link Layer, and routing Routing Layer. These are all lower level layers that deal with the transportation and handling of information. Adversaries in these layers try to harm the nodes at their most basic operations, as those are usually the most vulnerable. For example, the reason why Routing attacks are so easy is due to the lack of authentication in the network while sending broadcast messages for rerouting. Cooperative Communication is both affected by and mitigates several of the attacks in these layers. Table 1 shows a visualized summary of Cooperative Communications involvement with the attacks of these major layers.

The previous work in the literature did not propose a security protocol to address the aforementioned challenges. Our security protocol in this paper is designed to address the main vulnerabilities created by the use of cooperative communication while maintaining the benefits it yields. This protocol also improves the system's security and reliability to allow seamless data transmission from the source node, through the relay nodes, to the destination node.

Table 1. Cooperative Communication (CC) on known WSN attacks.

Attacks	Outcomes with CC
Jamming	Increases diversity to combat jammed nodes. Increased packet delivery ratio.
Tampering	Increases the chance of malicious node manipulating cooperative messages.
Eavesdropping	Even though cooperating packets mix, single channel packets remain full in the open.
Collision	With a CMAC protocol, travel time shortens.
Selfish Nodes	Lower number of backoffs. In [54], researchers implemented a cooperative jamming mechanism that used friendly interference to confuse the eavesdropper, with the hopes of increasing uncertainty.
Packet Dropping	Increased diversity to decrease chance of packet dropping attacks.
Wormholes	Makes detection of Wormhole attacks easier.
Sybil Attack	Attack is easier to perform due to more nodes to cooperate with.

4. Proposed Security Protocol

Creating a new security protocol is essential to WSNs that use CC because they require secure communication. One approach is testing to see how using cooperative communication increases protocol performance in terms of finding solutions and preventing common attacks on WSNs. The use of Cooperative Communication results in an improved BER and packet loss ratio, making the network more flexible and secure. Lastly, the security protocols should adopt lightweight cryptographic algorithms for wireless sensors as they have low resources compared to normal-sized, large sensors [55–58].

For this protocol, we aim to create a scheme that adds network authentication, message integrity, message freshness, intrusion detection, and anomaly prevention.

The security protocol this paper outlines is designed to address the vulnerabilities created by the use of cooperative communication while maintaining the benefits it yields. This protocol also adds a secure and reliable transmission of data from the source node, through the relay nodes, to the destination node. The protocol contains two main sections to help accomplish its goal: (1) Reputation Table and (2) Message Authentication Code. We will discuss the functions of each, how they work, and what vulnerabilities they look to solve in the following subsections.

4.1. Reputation Table

We have shown in the previous sections that the implementation of cooperative communication helps to decrease the BER and packet loss ratio, but perhaps we can attempt to decrease this even further. We propose the implementation of a reputation table embedded in each node that will record the statistics of each other node in the network. The nodes will then share their table with the other nodes in the network to have a consensus on the status of each node. This will allow the nodes to know what normal behavior should look like and figure out which nodes could be compromised (jammed or tampered). Once a node receives a certain degree of "bad reputation", the other nodes will no longer trust that node and prefer messages coming from more honest transmissions. The nodes will also have a trust in other nodes when receiving the table. They will compare reports with the reports of the other nodes, and, if the numbers are significantly different from the average, then the trust level will go down until it is no longer trusted.

Reputation tables will allow nodes to detect malicious, selfish, or defective nodes and reject transmissions from them, as well as take them out of the relay route. In Table 2, we give an example of the table and demonstrate how it would operate in Algorithm 1. Each node is listed given its identification number and contains information of its transmission history. There is a column that tallies the number of suspicious messages and total messages since the last update, collects the total for the suspicious messages and total messages, and calculates the trust and the suspicious packet ratio.

Once the update occurs, the nodes will all broadcast their tables to every other node in the network, sharing the information it gathered since the last update. The node receiving the table will then add up each measurement that each node has for the other nodes in the system to the respective suspicious and total messages since the last update and store the claimed ratio for that node in its respective Gossip array. The node will then add its own recordings for each node to the totals, set the values to 0, and calculate an average ratio. Then, the node will check to see if the reporting nodes were lying, or if there was some issue. It will add up the reported ratios all nodes will have for a particular node, get the average, then compare each reporting to the average. If the reported ratio is significantly different than the average ratio, then the trust value of that node will go down by a decided reduction amount. If the numbers matched up with the average, then the reduction amount will be added back to the node for each correct answer. The highest the value can go is 1. The trust value also has a threshold; if the trust value goes below a certain point, then that indicates that the node is either trying to poison the table, or is having trouble receiving. Either way, the node will also be rejected from transmission routes to optimize efficiency. The update can occur at a fixed time, depending on the implementation of the nodes. For more security, a more frequent update can occur, but this will require more consumption from the nodes to broadcast and do calculations.

Table 2. Example of Reputation Table.

Node	Suspicious Messages (Since Last Update)	Messages (Since Last Update)	Total Suspicious Messages	Total Messages	Trust	Ratio
R_1						
R_2						
.						
.						
.						
R_n						

4.2. Message Authentication Code (MAC)

We have established a method to be able to detect malicious nodes using the reputation table based on the number of suspicious messages sent. Now, we must implement a protocol that will detect weather or not a message is valid, or suspicious in any way. The purpose of designing this protocol is to assure secure transmission between nodes and to make sure the messages are fresh and came from the correct source. As you will see later on, confidentiality is not a main concern in this protocol, but an adjustment can be made to create confidentiality if the need is there.

4.2.1. Key Distribution

Each node when manufactured and initially booted will contain a symmetric master key built in. This master key is used to welcome new nodes into the system and create pairwise keys between them. The protocol is simple: The new node coming into the system is required to know the master key. If known, the new node will generate a new key and encrypt it with the master key. When the other nodes receive this message, they will decrypt it to reveal the new key and send an acknowledgement of reception encrypted using the new key, to prove knowledge. This prevents rouge nodes from entering the system to cause havoc, like rerouting other nodes and causing sinkholes. If the master key is compromised, however, then like any password-based authentication, the rogue node will have access to join the network by creating its own pairwise keys with the other nodes.

Algorithm 1 Algorithm for Reputation Table validation and update.

```
initializeTable()
while true do
  if messageReceived() == true then
    if messageValid() == false then
      suspiciousMessagesSinceUpdate++
      totalMessagesSinceUpdate++
    else
      totalMessagesSinceUpdate++
    end if
  end if
  if updateTable() == true then
    Gossip[n][n-1]
    for i in Node do
      for j in Node[i] do
        Gossip[j][i] = Node[i][j].suspiciousMessagesSince Update/Node[i][j].totalMessagesSince
        Update
        Node[j].suspiciousMessages += Node[i][j].suspiciousMessagesSinceUpdate
        Node[j].totalMessages += Node[i][j].totalMessagesSinceUpdate
      end for
    end for
    for i in Node do
      Node[i].suspiciousMessages += Node[i].suspiciousMessagesSinceUpdate
      Node[i].totalMessages += Node[i].totalMessagesSinceUpdate
      Node[i].suspiciousMessagesSinceUpdate = 0
      Node[i].totalMessagesSinceUpdate = 0
      Node[i].ratio = Node[i].suspiciousMessages / Node[i].totalMessages
    end for
    for i in Gossip do
      tempSum = 0
      for j in Gossip[i] do
        tempSum += Gossip[i][j]
      end for
      for j in Gossip[i] do
        if abs(Gossip[i] - (tempSum / length(Gossip[i])) >suspicionThreshold then
          Node[i].trust -= reduction
        else
          if Node[i].trust <1.0 then
            trust += reduction
            if (1-Node[i].trust) <0 then
              trust -= (trust - 1)
            end if
          end if
        end if
      end for
    end for
  end if
end while
```

4.2.2. MAC Design

Let us work with the scenario of four nodes in a network, for simplicity, demonstrated in Figure 2. The source node (S) will begin by transmitting a message to both relays 1 and 2, R_1 and R_2 , which may have been encrypted by S with a key K_{S-D} shared by S and the destination D, but not necessary. The message will be concatenated with a MAC to prove its authenticity to the next node. The MAC will be the hash of a concatenation of the message, a synchronized counter, and a shared secret key between S and D, K_{S-D}. All of this will then be concatenated with a synchronized counter to give us the final message sent by the source node:

$$m \| C \| h(m \| C \| K_{S-D})$$

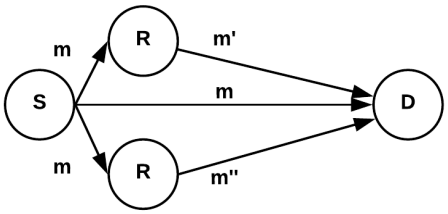

Figure 2. Example of MISO Cooperative Communication (CC).

With this message sent from both relays, the destination node will know the message and a synchronized counter. The counter will show D if the message took a suspicious amount of time to send by a given distance away from the expected counter. Out-of-order messages are possible in transmission, and are error corrected by a small variation from what the destination node expects. Shared keys are already established between neighboring nodes in a path. Now, D will take message, concatenate it with the counter from the transmission and the shared private key, and hash them all together. D will then compare the hash it calculated with the hash sent in the message to see if something was changed. If they do not match up, then the message was altered or the key is wrong and did not come from the expected source. After validation of the message, the destination node will compare its counter to the counter sent to see the message was sent too late. If any of these conditions are met, it is then safe to conclude that the message sent by that relay is indeed suspicious and therefore cannot be trusted. Figure 3 shows a data packet sent by the source, breaking each section in two and sending half of each section to the relay and holding the other halves to send with the relays data.

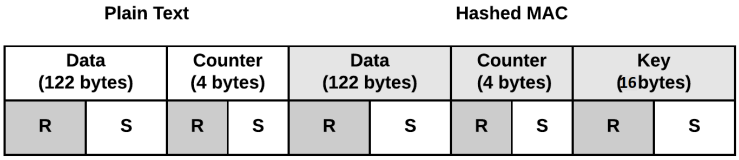

Figure 3. Data packet sent by the source.

As in the traditional CC schemes used in [28], part of the message is sent to a relay for redundancy. The relay, however, does not get the full message, but only part of it to combine with its own. The nodes leave it up to the the receiving node to put the messages together. Whether this is a single-hop or multi-hop, the message will maintain its MAC code through the relays, unless it get lost or dropped. Only the destination node will be able to verify the MAC, and hence the message cannot be changed.

5. Analysis of Security Protocol and Cooperative Communication

5.1. Attack Analysis

5.1.1. Jamming

For jamming attacks, we have shown that the cooperative communication model yields a lower packet loss ratio as multiple packets will be sent out from the source to reach the node in different paths, should one node be jammed. The more jammers there are in the network, the more packets will be dropped. However, the destination and relay nodes are expecting packets from the previous nodes. If they are not delivered, this will reflect on the table as well.

5.1.2. Tampering

Cooperative communication alone failed to stand against tampering attacks, as a malicious node corrupts a trusted node in the system and is free to do as it pleases. However, with the reputation table, any malicious activity will be reflected on the table and sure enough that node will be shunned from the routing protocol if it continues.

5.1.3. Eavesdropping

As we mentioned before, this protocol is designed to provide integrity and authentication to the messages, not privacy. Therefore, the proposed system does not combat eavesdropping alone, but an extra layer of security can be added on top of this protocol by adopting a lightweight encryption scheme to protect the secrecy of messages.

5.1.4. Collision and Selfish Nodes

Collision can still occur in this system as there are no preventative measures implemented. However, the reputation table will recognize suspicious activity if too many collisions are happening from one node and deem it either malicious or defective. Either way, the harmful node will be taken out of the system to increase efficiency and decrease packet loss and re-transmissions from collisions. The same applies with selfish nodes if the node tries to rush the signal to jam the reception.

5.1.5. Packet Dropping

This protocol is specifically meant to protect against attacks on routing protocols that try to drop packets. A malicious node could attempt to redirect traffic through it to become a black hole, but it will quickly get noticed by the reputation table. Likewise, a grayhole might have a better chance at avoiding the table, but the attacks would have to be carefully timed and very spread out, which reduces the severity of the damage.

5.1.6. Wormholes

The addition of the counter in the protocol not only helps to authenticate the messages, but keeps the messages fresh with an indicator of time sent. In addition, the hashing properties will not allow a node to verify that the message is designated for it because it will not have the correct key to compare hashes. Therefore, the check will fail and the sending node will be seen as a fraud. Both properties of the MAC lead to a detection by the reputation table.

5.1.7. Sybil Attack

The Sybil attack attempts to create rogue nodes in the network to interact with the other nodes by impersonating them. With this protocol, the malicious Sybil node could impersonate a node, but will not have the key to verify the hashes. Therefore, that node will be excluded from the network. The one issue this protocol overlooks is the case in which the Sybil node purposefully impersonates a genuine

node and starts performing malicious activity. That way, the other nodes will reject the genuine node and experience of DoS. Further computations and algorithms that will increase storage consumption will be required in future research.

5.2. Discussion

This protocol can be used as a solution to the presence of malicious, selfish, and unreliable nodes. Nodes detected to be untrustworthy, by tampering, will be rejected by other nodes and the network will look for new paths around the malicious node. This new path may be slower, but it will be more efficient. There could also be the case where a non corrupted massage that was eavesdropped by an outside party is sent again as a replay attack. The protocol we set up in the next subsection will demonstrate the detection of this delayed message as well and be counted as a suspicious message.

The use of the MAC is a reliable way to verify that a message came from the expected source and no changes have been made to the message. The addition of counters ensures that the message is fresh and is not the victim of a replay attack. Some MAC protocol prefer a timestamp over a counter; a usual drawback to using time stamps in systems is the fact that internal clocks are not always synchronized. The counter will ensure that the nodes are in sync and are counting at the same pace. If a node is compromised, it will affect its count and an adversary could not guess where the destination node is up to on the counter, unless that is compromised as well. To prevent the integer overflow issue, the receiver will request the sender to reset the counter, if necessary. This will be projected on the reputation table and the system will work around it. The tables can be monitored from an outside device and sensor check ups could be preformed. Sensors can also be tampered to send the wrong messages as well or jammed to send messages later or not at all. The table will show this as well as the receiving node will expect a message coming from a neighbor node and expect it to be correct. The addition of encrypting the message, counter, and hash with a private key held by the source, or signing it in asymmetric cryptography, ensures that the message indeed came from the source as only the source should know its own private key and the message can be verified by anyone. This will, however, add complexity and processing consumption. We see in [59] that the process of asymmetric encryption, such as signing, may be so complex that it will not be appropriate for a WSN environment. There is, of course, the possibility that the source node be compromised and the symmetric and asymmetric keys are taken for an active man-in-the-middle attack, but this is far easier to detect than a passive attack, which we have made corrected for in this code.

The purpose of implementing cooperative communication is to increase the efficiency of packet transmission in a WSN. We have seen that MIMO and MISO cooperative sensor networks have more resilience and efficiency because of the alternative means for the destination node to receive the data packets from the source nodes. Relay nodes allow transmissions over longer distances by passing along messages through the relays. The protocol we propose is simple and lightweight to authorize the messages and keep them safe. Let us look at a comparison between this method and the standard single path communication.

In the single communication scheme, we can apply our protocol of message verification with MAC to detect corrupt or delayed messages. Once D finds that the message is not valid, it will disregard it and request a re-transmission. This is time-consuming and unnecessary. Another situation would be the case of a dropped message, where D will be waiting for the message for a fixed amount of time until requesting a re-transmission. Again, time-consuming, but more so.

As shown in Figure 4, the cooperative communication model allows for a node to be corrupt, but still manages to get message to its destination in a timely manner. The reason this works is by sending more than one message you lower the risk of interruptions or obstacles by going through more than one path. The more messages you send out at once, the higher probability of a successful transmission. In this paper, we have assumed a master key is pre-installed at each node prior deployment. Such an approach is usually hard to implement and has a common vulnerability that, if the master key is compromised, then the node cannot be recovered. A better approach would be to use

a key distribution scheme that will address this issue; this is something that we would consider in our future work. Moreover, overloading relay nodes and the energy consumption is a common issue with CC in general and has been studied by other researchers that analyzed CC operation. Since this paper focuses on the security issues of CC, we consider this issue beyond the scope of our work.

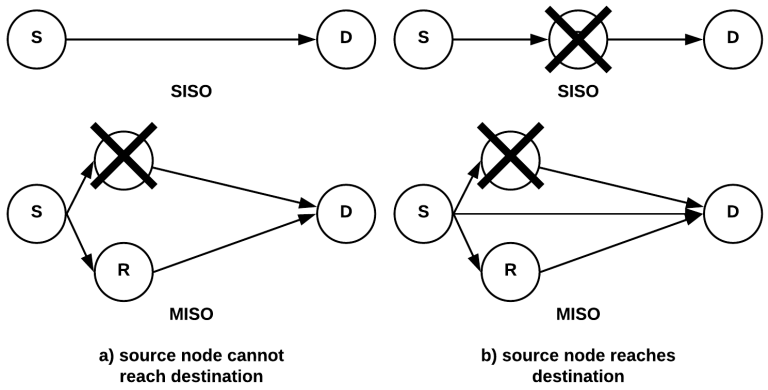

Figure 4. Single-in-single-out vs. Multi-in-single-out (CC) with a corrupt relay node.

We have achieved the goals we sought to accomplish with this protocol in terms of the following:

- **Network Authentication:** Key distribution system required new nodes to have knowledge of the master key to be given a pairwise key and join the network.
- **Message integrity:** The MAC ensures that the message attached to the hash is the same as what was used in the hash value with trust from knowledge of the pairwise key.
- **Message freshness:** The counters provide freshness to the message which is provided in plaintext, as well as used to create the hash for authentication. The counter will show the recipient if the message is unusually late.
- **Intrusion detection:** By means of the reputation table, malicious and unusual activity by intruders will be picked up and reflected on the tables for the network to see.
- **Anomaly prevention:** Once a node is not trusted, the rest of the nodes will no longer include them in routes and remove them from the network.

5.3. Storage and Space Analysis

Being that sensor nodes are low in resources, namely storage and power, it is important to analyze this protocol against those factors. To begin, we will calculate the number of bytes our protocol requires while continuously running. Since the algorithm is static by nature, and no variables are being created dynamically, we can quantify the number of bytes that will be allocated for all temporary variables at boot. This memory allocation will take place in the main memory of the sensor node.

To start, we will account for all static variables that are known from the beginning and used for reference. These variables will be the suspicion Threshold, the reduction coefficient, the symmetric master key, and the symmetric pairwise keys. To keep things simple, we will make the symmetric keys integer values to know exactly how many bytes we are working with in each case. An integer variable reserves 4 bytes of memory. Four integer variables will give us 16 bytes. The pairwise keys will be a function of the number of nodes in the system. If we have n amount of other nodes, then the number of bytes allocated to hold these keys will be $16n$. Then, we have our temporary variable within loops and function executions that will also be allocated at runtime. This variables is the tempSum and is also an integer. This leaves us at 16 bytes plus $16n$ allocated so far, simplified to $16n + 16$.

Now, we will incorporate another first order polynomial portion of the algorithm that will allocate memory scaling to the number of nodes there are in the system. This will be demonstrated in the table

itself. Since there are seven variables in the table, which are all integers, they will take up four bytes each, making one row, or node entry, reserve 28 bytes. Since this is per node in the table, the number of bytes taken up will be $28n$. At this stage in the algorithm, we are allocating $44n + 16$ bytes. The heaviest part of the algorithm is performed by the Gossip array, which is a two-dimensional integer array. This array is a second order polynomial in that its scales with $n * n - 1$. With four bytes per integer, we simplify the equation to $16n^2 - 16n$. Adding this to what we derived in the previous steps, we get the memory allocation equation:

$$16n^2 - 16n + 44n + 16.$$

Wireless sensors have limited capabilities in terms of memory. The sensors are not able to perform very large operations because of their relatively low computational and storage. For instance, mote-type sensor has 8-bit 4 MHz microcontroller, 40 KB of memory, and a radio with a bit rate of about 10 kbps. However, midrange nodes, such as "UCLA/ROCKWELL'S WINS", have a strong ARM 1100 processor, 128 KB of RAM, and a 100 Kbps radio [60]. For our analysis in this section, we will assume that the sensors utilize a memory up to 10 kilobytes (1 kilobyte = 1024 bytes).

By applying the quadratic equation and setting our formula equal to 10,240 as the upper bound, we get an n of 24. This means that there can be a maximum of 24 nodes in the system before a memory shortage. This algorithm is exceptionally heavy due to the two-dimensional array that created the second order polynomial. This has been added as an extra security measure which prevented a table poisoning attack. If this preventative measure was removed, however, we could have up to 145 nodes in the system. One solution to these issues is to break the area into smaller clusters of small sizes (less than 24 nodes). We can then assign clusters' heads and have them build a new reputation table. This will address the scalability issue of the proposed protocol and allow the system to efficiently address networks with large number of nodes.

6. Results

Our goal was to mimic a real implementation as close as possible. For this, we needed to represent the low resources devices that run on a Linux environment. To test our implementation, we wrote a C program to run on a Linux operating system that is installed on a device which had a 1.60 GHz Atom processor and 1 GB RAM. In our code, we implemented the algorithms described with the proposed security protocol (Section 4). The library we used for cryptographic hashing was OpenSSL. We operated multiple terminals simultaneously, to represent nodes in the system: one source node, one destination node, and as many relay nodes as needed. The simulation parameters were varied in testes and experiments to examine a large set of practical scenarios. The trials were run 100 times and an average time was taken. This set was then run another five times and a total average time was taken for each condition, for assurance.

In our experiments, we tested execution time of our protocol based on two main factors: Number of Nodes, and Message Size in Bytes. Both were hypothesized to change execution times, but the experiment yielded interesting results, Table 3 and Figure 5. The number of nodes seemed to have a direct correlation to the total average time of the process. The time referred to in the results are simulation run time units. This is due to the fact that the source nodes need time to send out their message to each relay node, and the destination node needs to receive and check each MAC. The destination node could be slightly jammed by the other nodes and delay the reception. There could also be a confusion of the signals with so many nodes receiving messages at once. This is due to the noise of frequency. A re-implementation of this experiment with better software that facilitates concurrent processing may lead to different results.

Table 3. Average run times of 1 byte input with different numbers of nodes.

Trials Per Set	Nodes	Set 1 Avg	Set 2 Avg	Set 3 Avg	Set 4 Avg	Set 5 Avg	Total Avg
100	10	590	622	620	633	623	617.6
99	9	564	585	547	546	579	564.2
104	8	521	507	497	515	508	509.6
105	7	482	475	483	447	449	467.2
102	6	425	420	408	429	412	419.8
100	5	367	363	360	361	365	363.2
100	4	341	306	317	321	335	324
99	3	270	272	277	268	269	271.2
100	2	226	224	231	232	223	227.2

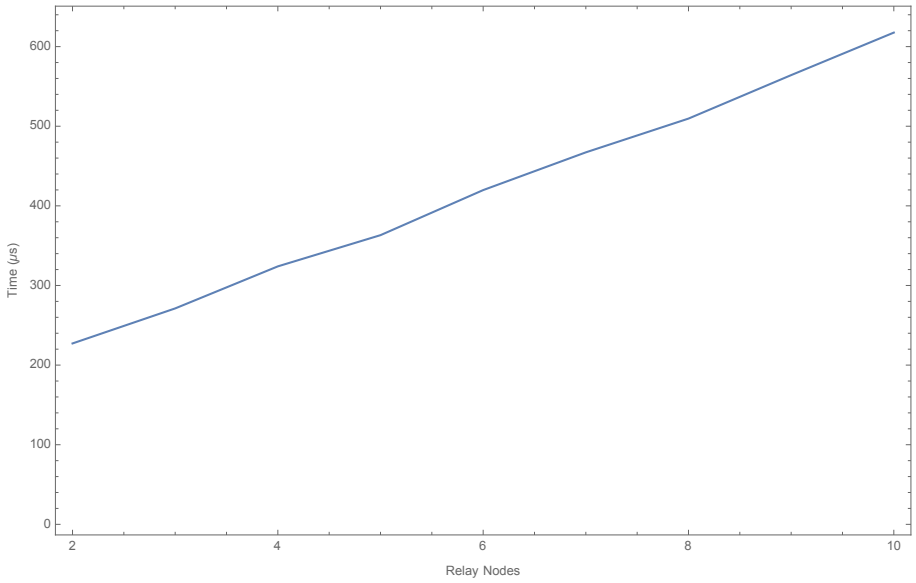

Figure 5. Linear relationship between number of nodes and total average time of execution with 1 input byte.

The difference in number of bytes in the input message yielded more unexpected results, Table 4 and Figure 6. The average time didn't seem to move much going from one to two bytes; in fact, it went down. The massive fluctuations, again, has to do with the noise associated with transmission, appearing that it went down. This test was run with 10 nodes, creating a lot of noise in transmission. We also didn't see much of a difference between one and ten. This is because, for all intensive purposes, they are basically the same speed. The change in input size does not have that much of a significant change in total speed. However, we do see a significant change when we go from one to one hundred and to five hundred. We hypothesize that the speed changes in a exponential fashion with changes in byte size. An attempt was made using 10,000 (multiplying 100 by 100), but the program couldn't handle that many bytes and crashed. Again, this would like to be tested with more sophisticated technology. We would like to note that the more nodes in transmission, the higher the deviation in average test runs, and vice versa. Likewise, the more amount of bytes also has a similar effect to the deviation.

Table 4. Average run times of different numbers of bytes input with 10 nodes.

Trials Per Set	Bytes Input	Set 1 Avg	Set 2 Avg	Set 3 Avg	Set 4 Avg	Set 5 Avg	Total Avg
100	1	590	622	620	633	623	617.6
100	2	604	619	627	614	612	615.2
100	10	600	582	624	602	613	604.2
100	100	651	704	630	658	637	656
100	500	1457	1334	1589	1338	1425	1428.5

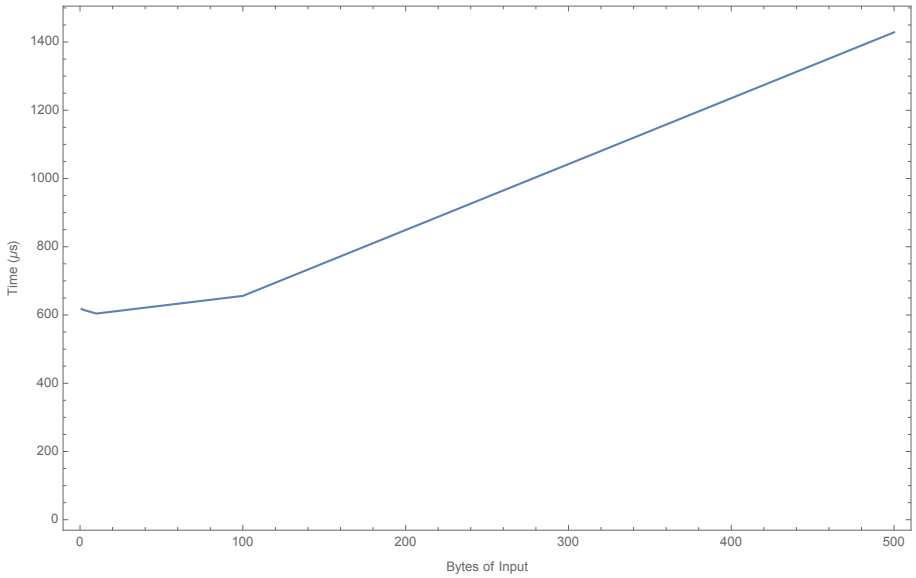

Figure 6. Exponential relationship between number of input bytes and total average time with 10 nodes.

7. Conclusions

Cooperative communication enhances the performance of WSNs; however, it opens new vulnerabilities and security holes. For some attacks (e.g., jamming, packet dropping, and wormhole), using cooperative communication improves the network resiliency and reliability. There is a need to design a new security protocol for WSNS with cooperative communication. This paper proposed a security MAC protocol at the application layer that implements a cryptographic hash to validate the integrity of the message sent with a simple key distribution scheme for authenticated entry to the network. An encryption of the data is possible, but the complexity and consumption would increase drastically, as well as the transmit time, which may not be appropriate for WSNs. We have also proposed a simple Reputation Table which all nodes will share to be able to identify any that are tampered, compromised, or malfunctioning. This algorithm, however, is of second order polynomial in terms of memory consumption versus number of nodes in the system, and is therefore considered heavy for low resource devices. Although it is appropriate for use in most scenarios where not many sensors are required in the network, more research needs to be done to improve this protocol.

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

Funding: This research received no external funding.

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

References

1. Lewis, F.L. Wireless sensor networks. In *Smart Environments: Technologies, Protocols, and Applications*; John Wiley: New York, NY, USA, 2004; pp. 11–46.
2. Hayajneh, T.; Khasawneh, S. Analysis and Evaluation of Random Placement Strategies in Wireless Sensor Networks. *J. Circuits Syst. Comput.* **2014**, *23*, 1450138. [CrossRef]
3. Perrig, A.; Stankovic, J.; Wagner, D. Security in wireless sensor networks. *Commun. ACM* **2004**, *47*, 53–57. [CrossRef]
4. Hayajneh, T.; Almashaqbeh, G.; Ullah, S.; Vasilakos, A.V. A survey of wireless technologies coexistence in WBAN: Analysis and open research issues. *Wirel. Netw.* **2014**, *20*, 2165–2199. [CrossRef]

5. Lee, K.; Lee, H. An energy-efficient cooperative communication method for wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2014**, *10*, 689710. [CrossRef]
6. Zhang, D.; Chen, Z. Energy-efficiency of cooperative communication with guaranteed E2E reliability in WSNs. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 532826. [CrossRef]
7. Nosratinia, A.; Hunter, T.E.; Hedayat, A. Cooperative communication in wireless networks. *IEEE Commun. Mag.* **2004**, *42*, 74–80. [CrossRef]
8. Fitzek, F.H.; Katz, M.D. *Cooperation in Wireless Networks: Principles and Applications*; Springer: Berlin, Germany, 2006.
9. Hong, L.; Chen, W. Information theory and cryptography based secured communication scheme for cooperative MIMO communication in wireless sensor networks. *Ad Hoc Netw.* **2014**, *14*, 95–105. [CrossRef]
10. Karlof, C.; Sastry, N.; Wagner, D. TinySec: A link layer security architecture for wireless sensor networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*; ACM: New York, NY, USA, 2004; pp. 162–175.
11. ZIGBEE Specification. Available online: http://www.olmicrowaves.com/menucontents/designsupport/zigbee/1171625602_ZigBee-Specification-2006-r13.pdf (accessed on 20 January 2020).
12. Luk, M.; Mezzour, G.; Perrig, A.; Gligor, V. MiniSec: A secure sensor network communication architecture. In Proceedings of the 2007 6th International Symposium on Information Processing in Sensor Networks, Cambridge, MA, USA, 25–27 April 2007; pp. 479–488.
13. Zhang, C.; Lin, X.; Lu, R.; Ho, P.H.; Shen, X. An efficient message authentication scheme for vehicular communications. *IEEE Trans. Veh. Technol.* **2008**, *57*, 3357–3368. [CrossRef]
14. Hao, Y.; Cheng, Y.; Zhou, C.; Song, W. A distributed key management framework with cooperative message authentication in VANETs. *IEEE J. Sel. Areas Commun.* **2011**, *29*, 616–629. [CrossRef]
15. Lin, X.; Li, X. Achieving efficient cooperative message authentication in vehicular ad hoc networks. *IEEE Trans. Veh. Technol.* **2013**, *62*, 3339–3348.
16. He, Q.; Wu, D.; Khosla, P. SORI: A secure and objective reputation-based incentive scheme for ad-hoc networks. In Proceedings of the 2004 IEEE Wireless Communications and Networking Conference, Atlanta, GA, USA, 21–25 March 2004; Volume 2, pp. 825–830.
17. Ganeriwal, S.; Balzano, L.K.; Srivastava, M.B. Reputation-based framework for high integrity sensor networks. *ACM Trans. Sens. Netw. TOSN* **2008**, *4*, 15. [CrossRef]
18. Khalid, O.; Khan, S.U.; Madani, S.A.; Hayat, K.; Khan, M.I.; Min-Allah, N.; Kolodziej, J.; Wang, L.; Zeadally, S.; Chen, D. Comparative study of trust and reputation systems for wireless sensor networks. *Secur. Commun. Netw.* **2013**, *6*, 669–688. [CrossRef]
19. Alzaid, H.; Alfaraj, M.; Ries, S.; Jøsang, A.; Albabtain, M.; Abuhaimed, A. Reputation-based trust systems for wireless sensor networks: A comprehensive review. In *IFIP International Conference on Trust Management*; Springer: Berlin, Germany, 2013; pp. 66–82.
20. Han, G.; Jiang, J.; Shu, L.; Niu, J.; Chao, H.C. Management and applications of trust in Wireless Sensor Networks: A survey. *J. Comput. Syst. Sci.* **2014**, *80*, 602–617. [CrossRef]
21. Zhang, T.; Safavi-Naini, R.; Li, Z. ReDiSen: Reputation-based secure cooperative sensing in distributed cognitive radio networks. In Proceedings of the 2013 IEEE International Conference on Communications (ICC), Budapest, Hungary, 9–13 June 2013; pp. 2601–2605.
22. Rodriguez-Mayol, A.; Gozalvez, J. Improving selfishness detection in reputation protocols for cooperative mobile ad-hoc networks. In Proceedings of the 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Instanbul, Turkey, 26–30 September 2010; pp. 2533–2538.
23. Sendonaris, A.; Erkip, E.; Aazhang, B. User cooperation diversity. Part I. System description. *IEEE Trans. Commun.* **2003**, *51*, 1927–1938. [CrossRef]
24. Sendonaris, A.; Erkip, E.; Aazhang, B. User cooperation diversity-Part II: Implementation aspects and performance analysis. *IEEE Trans. Commun.* **2003**, *51*, 1939–1948. [CrossRef]
25. Laneman, J.N.; Wornell, G.W.; Tse, D.N. An efficient protocol for realizing cooperative diversity in wireless networks. In Proceedings of the 2001 IEEE International Symposium on Information Theory, Washington, DC, USA, 29 June 2001; p. 294.
26. Laneman, J.N.; Tse, D.N.; Wornell, G.W. Cooperative diversity in wireless networks: Efficient protocols and outage behavior. *IEEE Trans. Inf. Theory* **2004**, *50*, 3062–3080. [CrossRef]

27. Hunter, T.E.; Nosratinia, A. Cooperation diversity through coding. In Proceedings of the IEEE International Symposium on Information Theory, Lausanne, Switzerland, 30 June–5 July 2002; p. 220.
28. Hunter, T.E.; Nosratinia, A. Coded cooperation under slow fading, fast fading, and power control. In Proceedings of the Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 3–6 November 2002; Volume 1, pp. 118–122.
29. Hunter, T.E.; Nosratinia, A. Diversity through coded cooperation. *IEEE Trans. Wirel. Commun.* **2006**, *5*, 283–289. [CrossRef]
30. Stefanov, A.; Erkip, E. Cooperative coding for wireless networks. *IEEE Trans. Commun.* **2004**, *52*, 1470–1476. [CrossRef]
31. Janani, M.; Hedayat, A.; Hunter, T.E.; Nosratinia, A. Coded cooperation in wireless communications: Space-time transmission and iterative decoding. *IEEE Trans. Signal Process.* **2004**, *52*, 362–371. [CrossRef]
32. Laneman, J.N.; Wornell, G.W. Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks. *IEEE Trans. Inf. Theory* **2003**, *49*, 2415–2425. [CrossRef]
33. Madan, R.; Mehta, N.B.; Molisch, A.F.; Zhang, J. CTH17-2: Energy-Efficient Cooperative Relaying over Fading Channels with Simple Relay Selection. In Proceedings of the IEEE Globecom 2006, San Francisco, CA, USA, 27 November–1 December 2006, pp. 1–6.
34. Bletsas, A.; Lippnian, A.; Reed, D.P. A simple distributed method for relay selection in cooperative diversity wireless networks, based on reciprocity and channel measurements. In Proceedings of the 2005 IEEE 61st Vehicular Technology Conference, Stockholm, Sweden, 30 May–1 June 2005; Volume 3, pp. 1484–1488.
35. Cardei, M.; Wu, J.; Yang, S. Topology control in ad hoc wireless networks using cooperative communication. *IEEE Trans. Mob. Comput.* **2006**, *5*, 711–724. [CrossRef]
36. Bletsas, A.; Khisti, A.; Reed, D.P.; Lippman, A. A simple cooperative diversity method based on network path selection. *IEEE J. Sel. Areas Commun.* **2006**, *24*, 659–672. [CrossRef]
37. Ibrahim, A.S.; Sadek, A.K.; Su, W.; Liu, K.R. Cooperative communications with relay-selection: When to cooperate and whom to cooperate with? *IEEE Trans. Wirel. Commun.* **2008**, *7*, 2814–2827. [CrossRef]
38. Wang, B.; Han, Z.; Liu, K.R. Distributed relay selection and power control for multiuser cooperative communication networks using buyer/seller game. In Proceedings of the 26th IEEE International Conference on Computer Communications, Barcelona, Spain, 6–12 May 2007; pp. 544–552.
39. Wang, B.; Han, Z.; Liu, K.R. Distributed relay selection and power control for multiuser cooperative communication networks using stackelberg game. *IEEE Trans. Mob. Comput.* **2009**, *8*, 975–990. [CrossRef]
40. Van Veen, B.D.; Buckley, K.M. Beamforming: A versatile approach to spatial filtering. *IEEE ASSP Mag.* **1988**, *5*, 4–24. [CrossRef]
41. Talebi, A.; Krzymien, W.A. Multiple-antenna multiple-relay cooperative communication system with beamforming. In Proceedings of the IEEE Vehicular Technology Conference, Singapore, 11–14 May 2008; pp. 2350–2354.
42. Yang, Y.; Li, Q.; Ma, W.K.; Ge, J.; Ching, P. Cooperative secure beamforming for AF relay networks with multiple eavesdroppers. *IEEE Signal Process. Lett.* **2013**, *20*, 35–38. [CrossRef]
43. Ye, W.; Heidemann, J.; Estrin, D. An energy-efficient MAC protocol for wireless sensor networks. In Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, New York, NY, USA, 23–27 June 2002; Volume 3, pp. 1567–1576.
44. Van Dam, T.; Langendoen, K. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*; ACM: New York, NY, USA, 2003; pp. 171–180.
45. Anastasi, G.; Conti, M.; Di Francesco, M.; Passarella, A. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Netw.* **2009**, *7*, 537–568. [CrossRef]
46. Rault, T.; Bouabdallah, A.; Challal, Y. Energy efficiency in wireless sensor networks: A top-down survey. *Comput. Netw.* **2014**, *67*, 104–122. [CrossRef]
47. Chalasani, S.; Conrad, J.M. A survey of energy harvesting sources for embedded systems. In Proceedings of the IEEE SoutheastCon 2008, Huntsville, AL, USA, 3–6 April 2008; pp. 442–447.
48. Sudevalayam, S.; Kulkarni, P. Energy harvesting sensor nodes: Survey and implications. *IEEE Commun. Surv. Tutor.* **2011**, *13*, 443–461. [CrossRef]
49. Shaikh, F.K.; Zeadally, S. Energy harvesting in wireless sensor networks: A comprehensive review. *Renew. Sustain. Energy Rev.* **2016**, *55*, 1041–1054. [CrossRef]

50. Mao, Y.; Wu, M. Security issues in cooperative communications: Tracing adversarial relays. In Proceedings of the 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, Toulouse, France, 14–19 May 2006; Volume 4, p. IV.
51. Yang, W.; Wang, S.; Xu, X. Cooperative transmission for security enhancement in clustered wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 751456. [CrossRef]
52. Aksu, A.; Krishnamurthy, P.; Tipper, D.; Ercetin, O. On security and reliability using cooperative transmissions in sensor networks. *Mob. Netw. Appl.* **2012**, *17*, 526–542. [CrossRef]
53. Makda, S.; Choudhary, A.; Raman, N.; Korakis, T.; Tao, Z.; Panwar, S. Security implications of cooperative communications in wireless networks. In Proceedings of the 2008 IEEE Sarnoff Symposium, Princeton, NJ, USA, 28–30 April 2008; pp. 1–6.
54. Rohokale, V.M.; Prasad, N.R.; Prasad, R. Reliable and secure cooperative communication for wireless sensor networks making use of cooperative jamming with physical layer security. *Wirel. Pers. Commun.* **2013**, *73*, 595–610. [CrossRef]
55. Hayajneh, T.; Doomun, R.; Al-Mashaqbeh, G.; Mohd, B.J. An energy-efficient and security aware route selection protocol for wireless sensor networks. *Secur. Commun. Netw.* **2014**, *7*, 2015–2038. [CrossRef]
56. Mohd, B.J.; Hayajneh, T.; Vasilakos, A.V. A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues. *J. Netw. Comput. Appl.* **2015**, *58*, 73–93. [CrossRef]
57. Mohd, B.J.; Hayajneh, T.; Shakir, M.Z.; Qaraqe, K.A.; Vasilakos, A.V. Energy model for light-weight block ciphers for WBAN applications. In Proceedings of the 4th International Conference on Wireless Mobile Communication and Healthcare (Mobihealth), 2014 EAI, Athens, Greece, 3–5 November 2014; pp. 1–4.
58. Hayajneh, T.; Ullah, S.; Mohd, B.J.; Balagani, K.S. An Enhanced WLAN Security System with FPGA Implementation for Multimedia Applications. *IEEE Syst. J.* **2017**, *11*, 2536–2545. [CrossRef]
59. Hayajneh, T.; Mohd, B.J.; Imran, M.; Almashaqbeh, G.; Vasilakos, A.V. Secure authentication for remote patient monitoring with wireless medical sensor networks. *Sensors* **2016**, *16*, 424. [CrossRef]
60. Djedouboum, A.; Abba Ari, A.; Gueroui, A.; Mohamadou, A.; Aliouat, Z. Big Data Collection in Large-Scale Wireless Sensor Networks. *Sensors* **2018**, *18*, 4474. [CrossRef]