electronics

MDPI

Article

A Distance Vector Hop-Based Secure and Robust Localization Algorithm for Wireless Sensor Networks

Rameez Asif [1,*], Muhammad Farooq-i-Azam [2], Muhammad Hasanain Chaudary [3], Arif Husen [3,4] and Syed Raheel Hassan [1]

[1] School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, UK
[2] Department of Electrical and Computer Engineering, COMSATS University Islamabad, Lahore Campus, Lahore 54000, Pakistan
[3] Department of Computer Science, COMSATS University Islamabad, Lahore Campus, Lahore 54000, Pakistan
[4] Department of Computer Science and Information Technology, Virtual University of Pakistan, Lahore 54000, Pakistan
* Correspondence: rameez.asif@uea.ac.uk

Abstract: Location information of sensor nodes in a wireless sensor network is important. The sensor nodes are usually required to ascertain their positions so that the data collected by these nodes can be labeled with this information. On the other hand, certain attacks on wireless sensor networks lead to the incorrect estimation of sensor node positions. In such situations, when the location information is not correct, the data may be labeled with wrong location information that may subvert the desired operation of the wireless sensor network. In this work, we formulate and propose a distance vector hop-based algorithm to provide secure and robust localization in the presence of malicious sensor nodes that result in incorrect position estimation and jeopardize the wireless sensor network operation. The algorithm uses cryptography to ensure secure and robust operation in the presence of adversaries in the sensor network. As a result of the countermeasures, the attacks are neutralized and the sensor nodes are able to estimate their positions as desired. Our secure localization algorithm provides a defense against various types of security attacks, such as selective forwarding, wormhole, Sybil, tampering, and traffic replay, compared with other algorithms which provide security against only one or two types. Simulation experiments are performed to evaluate the performance of the proposed method, and the results indicate that our secure localization algorithm achieves the design objectives successfully. Performance of the proposed method is also compared with the performance of basic distance vector hop algorithm and two secure algorithms based on distance vector hop localization. The results reveal that our proposed secure localization algorithm outperforms the compared algorithms in the presence of multiple attacks by malicious nodes.

Keywords: secure localization; positioning; distance vector hop; DV-Hop; security attacks; wireless sensor network

Citation: Asif, R.; Farooq-i-Azam, M.; Chaudary, M.H.; Husen, A.; Hassan, S.R. A Distance Vector Hop-Based Secure and Robust Localization Algorithm for Wireless Sensor Networks. *Electronics* **2023**, *12*, 2237. https://doi.org/10.3390/electronics12102237

Academic Editors: Hirokazu Kobayashi and Pingyi Fan

Received: 11 February 2023
Revised: 28 March 2023
Accepted: 10 May 2023
Published: 15 May 2023

1. Introduction

Location information of sensor nodes in a wireless sensor network (WSN) is considered important due to several factors. For example, the data gathered by the sensor nodes must be labeled with the coordinates of the geographic location from where these are collected. Without location information, the data may not make much sense [1]. Examples of such applications where position information is significant include area surveillance [2], habitat monitoring [3], agricultural monitoring [4], and rescue operations [5]. Position information also enables the WSN to make route decisions in the case of certain routing protocols. Using such routing decisions, the data may be routed, for example, to the closest sink [6]. Transmission and communication costs are reduced in this way and the network is energy

efficient. Location information also enables the sensor nodes to self organize and form an optimized WSN [7].

Due to aforementioned significance of location information, unknown sensor nodes, i.e., the sensor nodes which do not know their positions, employ a localization algorithm to estimate their position coordinates in the sensor network [8]. By using a localization algorithm, the unknown sensor nodes usually estimate their positions with the help of a few beacon nodes [9]. The beacon nodes, also called anchor nodes, reference nodes, or landmark nodes, know their position coordinates a priori either because these are deployed at known positions or are equipped with a location finding device, such as a global navigation satellite system (GNSS) receiver. A number of localization algorithms for WSNs have been proposed in the literature. A localization scheme for WSN proposed in [10] relies on Voronoi diagram-based grouping tests. This approach involves dividing the sensor nodes in a WSN into several groups and utilizing the closest corresponding Voronoi cells to determine location information. A localization method for WSN which does not need anchor nodes and instead uses cross technology for communication has been proposed in [11]. Instead of using anchor nodes, the method exploits the position information of wireless fidelity (Wi-Fi) access points (APs) for range estimation. Once an unknown node has ascertained its position, it helps other unknown nodes to estimate their positions. A localization algorithm based upon a selection strategy of appropriate beacon nodes has been proposed in [12]. The algorithm uses the signal strength information between the nodes for the selection strategy. With the help of signal strength information topology diagram of a set of nodes is formed. This diagram is then further exploited for position estimation. Localization in WSN is an active area of research and many other location estimation algorithms have also been proposed, such as [13–19].

The majority of these localization algorithms do not take security into consideration. Therefore, these algorithms are prone to various types of security attacks. As a result of these attacks, different types of problems may arise in the localization process. The positions estimated by some of the sensor nodes may have large errors. It is also possible that some nodes are not able to estimate their positions at all due to a security attack. To counter these problems, security measures and secure localization algorithms are being proposed. Two secure localization algorithm against different types of security attacks have been presented in [20]. The first algorithm, named improved randomized consistency position algorithm, exploits position information of beacon nodes and particle swarm optimization (PSO) for localization of unknown nodes. The second algorithm, referred to as the enhanced attack-resistant secure localization algorithm, utilizes a combination of methods, including a voting system, location optimization, and PSO, to estimate the positions of sensor nodes whose locations are unknown. The method proposed in [21] utilizes a blockchain based trust management model to combat malicious nodes in a sensor network. The trust evaluation is composite and involves behavior and data for this purpose. Different parameters, such as honesty, closeness, frequency of interaction, and intimacy, are used for the evaluation of behavior-based trust of the beacon nodes. Honesty is measured using the number of successful and unsuccessful interactions among sensor nodes. The number of sensor nodes covered by a beacon node in one hop neighborhood determines the closeness factor. The frequency of interaction is dependent upon total number of interactions between beacon nodes. Intimacy is quantified by the time of interaction. The beacon nodes with the least trust values are discarded to ensure localization reliability. A received signal strength-based localization algorithm for a WSN with malicious nodes has been proposed in [22]. The algorithm uses different localization techniques, i.e., weighted least square, secure weighted least square, and two norm-based techniques. The different techniques are meant to counter different types of security attacks.

Traditionally, cryptography is used to counter different types of security attacks in various categories of networks. However, conventional cryptography may not be used in resource constrained networks, such as WSN. Therefore, lightweight cryptography techniques have been proposed for such networks. A lightweight public key infrastructure

(PKI) has been proposed in [23] for networks with limited resources, such as the Internet of things (IoT) and WSN. PKI is a security system that uses encryption to authenticate the identity of devices and secure the communication between them. However, the PKI was not designed for devices with constrained resources. Therefore, the conventional PKI system may also not be deployed in networks, such as WSN and IoT, where the devices have small energy resource in the form of a battery, limited memory and storage, and small processing power. The work in [23] has developed a lightweight public key infrastructure (PKI) for registration and distribution of digital certificates in networks with highly constrained devices. The proposed lightweight PKI can be used to secure IoT and WSN devices in a variety of industries, such as healthcare, industrial, transportation, and smart cities. An aggregate signature technique based on a linearly homomorphic signature for resource constrained electronic healthcare system has been proposed in [24]. By combining the advantages of aggregate signature and linearly homomorphic signature, this method offers benefits from both. Under the security model, an aggregate signature is considered valid only if each individual signature utilized to construct the aggregate signature is also valid. Lightweight security algorithms have been used in [25] for reliable data collection from healthcare WSN and to improve security efficiency. The scheme uses elliptic curve digital signature algorithm with BLAKE2bp for the security. Privacy of the patients is ensured by masking the sensor identifications with pseudonyms. Similar works, such as [26–30] have proposed lightweight cryptography techniques for WSN and IoT.

In our work, we propose a secure and robust localization algorithm for WSN. The proposed algorithm is based on distance vector hop (DV-Hop) localization [31,32], which is a popular technique for position estimation in WSN. The traditional DV-Hop method is prone to different types of security attacks. We employ cryptography techniques to provide a secure localization algorithm, which we call the Secure DV-Hop. Compared to other secure algorithms based on DV-Hop which provide protection only against a single type of attack, our proposed secure localization algorithm provides security against multiple types of attacks. The performance of the proposed algorithm is evaluated and compared with the benchmark traditional DV-Hop algorithm and two other secure algorithms based on DV-Hop using simulation experiments. Results show that our proposed algorithm provides a secure, robust, and consistent performance in the presence of malicious nodes.

The rest of this paper is organized as follows. We discuss previously published research related to our work in Section 2. In the next Section 3, we delineate the network model. The DV-Hop localization algorithm is described in Section 4. We present our secure localization algorithm in Section 5. Performance evaluation of the algorithm is reported in Section 6. We finally conclude with Section 7.

2. Related Work

Previously, work has been performed to investigate different types of attacks in wireless sensor networks and their impact on localization and positioning accuracy. In this section, we describe and discuss the related work that has been completed to develop secure localization algorithms for wireless sensor networks.

The work in [33] proposed to secure the DV-Hop localization algorithm against wormhole attacks. The wormhole attack is usually carried out by more than one node in the network. One of the malicious nodes collects and forwards data from the compromised nodes through a tunnel to another malicious node located somewhere else in the network. The secondary malicious node then may transmit the data to the destination while masquerading the identity of the compromised nodes. In this way, the receiving node may be lead to believe that the sender is located at a different hop count other than the actual value. As a result, the localization process may be severely disrupted and the reported positions may have large errors.

Chen et al. analyzed the impact of the wormhole attack and thereby proposed a label-based secure DV-Hop scheme to mitigate this attack in [33]. The proposed method consists of three phases. In the first phase, the beacon nodes are labeled according to their

geographic locations. Next, in the second phase, the sensor nodes are differentiated and labeled according to beacon node labeling results. By exploiting these labels, malicious wormhole communication links between the nodes can be prevented. In the final and third phase, the localization process may be completed by using the DV-Hop. This scheme, however, does not take packet loss into consideration. Moreover, it assumes that all nodes have the same transmission radii and does not consider the scenario where different nodes may have different transmission coverage.

Another secure localization algorithm, which is based on DV-Hop and provides defense against the wormhole attack was presented in [34]. This work considers the default wormhole attack with an out of band hidden channel and without data modification. All the nodes in the network are aware of their identification numbers except for the attack nodes. The proposed scheme comprises three stages, i.e., detection of the wormhole attack, resistance against the wormhole attack, and error sources analysis. At first, the proposed scheme establishes a neighbor node relationship list through a broadcast mechanism. The suspect nodes are then identified by comparing the actual number with the theoretical number of nodes. Further, to isolate the actually attacked beacon nodes, the suspect nodes estimate distances from other nodes in their neighbor node relationship list. After the victim nodes have been identified, the attacked nodes mark themselves as either type 1 or type 2 depending upon the attacker node and assuming that there are only two types of attacker nodes in the network. Next, the unknown nodes also mark themselves as either type 1 or type 2 according to their neighbor nodes relationship list. Finally, the nodes marked as type 1 disconnect from nodes marked as type 2 and vice versa to mitigate the wormhole attack. After the attack has been mitigated the localization can be performed. The main limitation of this proposed scheme is that the attack model considers only two attacker nodes. Information modification is also not considered in the attack model.

Prashar et al. proposed a secure localization algorithm for WSN using digital signatures in [35]. At first, the private and public key pair for each node are created. Next, digital signatures for the nodes are generated so that the nodes can authenticate each other. After this, secure localization is performed based upon a procedure derived from DV-Hop. In the DV-Hop localization algorithm, the essential steps for node localization are, hop count determination, average hop size calculation, distance estimation and position determination using trilateration. However, the method proposed in [35], uses a scheme called hyperbolic and mid-perpendicular with centroid to estimate the node positions. If the unknown node is an immediate neighbor of an anchor node, then the mid perpendicular with centroid method is used. Otherwise, hyperbolic scheme is leveraged for position determination.

Another secure localization algorithm for WSN was presented in [36]. The work proposes a malicious node detection algorithm and also presents its extended version. The proposed algorithm, which is range-based, has four stages. In the first stage, the location data of an unknown are obtained using trilateration. In the second stage, the location data are divided into normal and abnormal clusters using self-adaptive density-based spatial clustering of applications with noise. Next, in the third stage, the reference error interval is calculated for the difference between two separate distance measurements based on time of arrival and received signal strength of the reference node. In the final fourth stage, a sequential probability ratio test is performed to test the difference between two measured distances of the suspected malicious node. After all these four stages have been completed, the malicious nodes are detected and the information provided by these malicious nodes can be discarded and the locations of the unknown nodes can be estimated through multilateration.

A secure localization algorithm against the Sybil attack was proposed in [37]. In the Sybil attack, a malicious node may monitor, listen, capture, and modify the data in a network. As a result, the malicious node is able to forge and present multiple identities to the other nodes in the network. This is accomplished by either generating false identities or by simply stealing and spoofing identities of other legitimate nodes on the network. The nodes with forged identities are usually referred to as the Sybil nodes [38]. The

Sybil nodes communicate with other nodes in the network using the forged identities and propagate false information. As a result, the integrity of the data in the network is compromised and network functions based upon this false information are severely damaged. The work in [37] proposed a defense against the Sybil attack which is based upon number allocation and neighbor nodes guarantee. Each node in the network is allotted a number by guaranteed nodes. The number acts as the identity of the node and is verified by its guaranteed node. As a result, any malicious nodes which are not able to present a valid number can be identified and isolated thereby securing the network and the localization process.

Another work in [39] has proposed secure localization using DV-Hop against the Sybil attack. In this proposed method, the beacon nodes broadcast test information. The replies from the neighbor nodes are monitored and a neighbor list is established. If a node has a different neighbor list, then it is concluded that the node is under Sybil attack. If the node has the same neighbor list, then the hop difference between the nodes in the neighbor list is determined. If the hop difference is zero, then it is concluded that the node is under Sybil attack. All the nodes which are found to be under the attack are added to a black list. All the remaining nodes then estimate their positions using the DV-Hop localization algorithm. This proposed method provides protection against only Sybil attack and does not provide defense against other types of attacks on the confidentiality, integrity, and availability of information.

3. Network Model

We consider a WSN deployed in a two-dimensional unconstrained sensor field. The sensor field has finite geographic boundaries. Two types of nodes are deployed in the WSN. The beacon nodes, also known as anchor, landmark or reference nodes, are fixed nodes which know their exact position coordinates. This is possible because these beacon nodes are equipped with navigation devices, such as a global positioning system (GPS), which is a type of global navigation satellite system (GNSS) or because the beacon nodes are deployed at known position coordinates. The other type of nodes in the sensor field are the sensor nodes which perform the sensing and collect the required data. These nodes are not aware of their location. Therefore, these nodes are usually termed as unknown nodes. Alternatively, some literature may refer to these nodes with less plausible names, such as dumb nodes or blind nodes. The unknown nodes estimate their positions with the help of the beacon nodes using a localization algorithm.

In our present work, the localization algorithm to be used by the unknown nodes is DV-Hop ad hoc positioning system. An assumption is made that all nodes in the network have the same radio range. However, the radio range of the unknown nodes is greater than their sensing range. This results in a higher sensing granularity of the WSN, allowing the transmission of sensed data over longer distances. Additionally, all nodes are outfitted with omnidirectional antennas, enabling them to communicate equally well in all directions. We represent a beacon node as B_i where $B_i \in \mathcal{B} = \{B_1, B_2, B_3, \cdots, B_L\}$. So, B_i is a member of $\mathcal{B}$, where the number of beacon nodes in the set is L. The position of a beacon node B_i is given by (x_{Bi}, y_{Bi}). Similarly, we represent an arbitrary unknown sensor node as U_i, where $U_i \in \mathcal{U} = \{U_1, U_2, U_3, \cdots, U_N\}$. Therefore, there are N unknown sensor nodes in the set $\mathcal{U}$ which are deployed in the sensor field. The actual position of an unknown node U_i is represented using (x_{Ui}, y_{Ui}), whereas the estimated position is denoted by $(\hat{x}_{Ui}, \hat{y}_{Ui})$. Each node in the network is pre-installed with a secret key K for encryption and decryption using secret key cryptography. Each node also generates a public and private key pair using an asymmetric encryption algorithm. The network also operates a lightweight public key infrastructure (PKI) for secure management and distribution of the public keys. Secret key encryption is used to ensure confidentiality whereas public key encryption is employed for authentication of hash values only as the latter encryption technique is computationally expensive [40]. The cryptographic keys are stored using a secure storage mechanism [41–46], such as a hardware security module.

We consider that the sensor network is deployed in a hostile environment where malicious nodes are present. The malicious nodes can launch one or a combination of security attacks to disrupt the network operations and localization system. It is considered that the malicious nodes are able to use different types of attacks, including wormhole, tampering, Sybil, traffic replay, and selective forwarding attacks. In the wormhole attack, the malicious nodes create a tunnel between two points in the network. Packets are captured at one point and tunneled to the other point. In the tampering attack, a malicious node modifies the contents of the intercepted packets, such as changing of beacon node position coordinates in the beacon message. Consequently, the position estimated by the unknown nodes is not correct. In the Sybil attack, a malicious nodes uses forged identities to spread false information and disrupt localization system and network operations. A malicious node can intercept and capture packets in a network communication and then later replay the packets to impersonate the identity of one of the nodes involved in the original communication. This type of attack falls in the category of traffic replay attack. In the selective forwarding attack, a malicious node selectively forwards some of the packets while dropping the other packets.

4. Distance Vector Hop Localization

In this section, we briefly describe and discuss the DV-Hop ad hoc positioning system [31,32] for wireless sensor networks.

The DV-Hop algorithm uses distributed processing. To estimate its location, each unknown node calculates its distance from three or more beacon nodes and then uses multilateration to calculate position coordinates. In a multi-hop sensor network, an unknown node may not have direct communication link with three beacon nodes. In other words, the unknown node may be more than one hop away from the beacon nodes. To address this problem, the DV-Hop localization algorithm leverages the connectivity information and the hop count to estimate the distance of an unknown node which may be at a multi-hop distance from the beacon node. Similar to the nature of operation of distance vector (DV) routing protocols, the DV-Hop localization algorithm uses flooding to propagate information in the multi-hop sensor network [47]. Beginning with the beacon nodes, each of the nodes propagates information only to its immediate first hop neighbors. Leaving out the next hop nodes saves bandwidth and power making the approach suitable for WSNs with limited resources. The signaling complexity of this scheme depends upon the number of beacon nodes in the sensor field and average degree of each node, i.e., the number of single hop neighbors of a node.

All the unknown and the beacon nodes in the WSN maintain a table with an entry corresponding to each of the beacon nodes from which it receives messages. The entry is of the form $\{x_{Bi}, y_{Bi}, h_i\}$, where (x_{Bi}, y_{Bi}) are the position coordinates of the beacon node B_i and h_i is the hop count of the node maintaining the table from the beacon node B_i. To obtain the hop count, the hop count field in the beacon message is incremented as the message is transmitted from the beacon node to its nearest neighbor nodes and so on. The beacon nodes in the WSN also maintain this table. After a beacon node B_i has obtained position information and hop count of all other beacon nodes B_j from which it receives messages, it proceeds to ascertain the average size of a hop [31] as follows,

$$c_i = \frac{\sum \sqrt{(x_{Bi} - x_{Bj})^2 + (y_{Bi} - y_{Bj})^2}}{\sum h_j}, \tag{1}$$

for all beacon nodes B_j and $B_j \neq B_i$. The numerator of Equation (1) is the sum of the distances between a beacon node B_i and other beacon nodes B_j. The denominator is the sum of hop counts between the beacon node B_i and other beacon nodes B_j. Therefore, Equation (1) gives average size of a hop as the sum of distances divided by the sum of hop counts. The DV-Hop algorithm terms this average size of the hop c_i, calculated by the beacon node B_i, as the correction factor. Using controlled flooding, this correction factor is

propagated through the network as described earlier. After receiving the correction factor and with the knowledge of position coordinates of at least three beacon nodes, an unknown node performs multilateration to estimate its own position information. The steps involved in the position estimation using the DV-Hop ad hoc positioning system are summarized as follows:

- At the beginning of the algorithm, the beacon nodes transmit their location data to their nearest neighbor nodes in the first hop.
- All the other nodes in the network receive and propagate the beacon node position coordinates using the same method as the distance vector routing protocol. The intermediate nodes increment the hop count field as they propagate the information to the next hop neighbor. Eventually, all the nodes in the network obtain the position coordinates of all the beacon nodes along with the hop count to these beacon nodes.
- After a beacon node has obtained position coordinates of other beacon nodes and the hop count to them, it computes the average hop size using Equation (1).
- A beacon node propagates its computed average hop size as correction factor throughout the network using the controlled flooding approach of the DV protocol.
- Once an unknown node receives the correction factor, it calculates the distance to the beacon node from which it received the correction factor. The distance is calculated by multiplying the hop count to the correction factor, i.e., $h_i \times c_i$.
- After knowing the position coordinates of at least three beacon nodes and distance estimates to them, the unknown node performs multilateration to estimate its position.

It should be noted that the correction factor calculated by one beacon node may differ from the correction factor computed by another beacon node. Moreover, each unknown node will receive different correction factors from different beacon nodes. The DV-Hop ad hoc positioning system [31,32] suggests that, for position estimation, an unknown node should store and utilize the initial correction factor it receives and disregard any other correction factors received subsequently.

5. Secure Localization

In this section, we describe our secure localization algorithm based on DV-Hop using cryptography. In the ensuing description of our proposed secure localization algorithm, concatenation of two items, a and b, is denoted by $a||b$. We denote the encryption operation of a message M using key K to obtain ciphertext C by $C = E(K, M)$. The decryption operation of the ciphertext C using the key K to obtain the message M is denoted by $M = D(K, C)$. When A sends a message M to B, we represent this as follows.

$$A \xrightarrow{M} B \tag{2}$$

At the time of first deployment, an unknown node U_i sends a registration request to the nearest beacon node B_i from which it receives messages. The registration message M_{Ui} is prepared as follows,

$$M_{Ui} \leftarrow ID_{Ui}||REG||N_{Ui}||S_{Ui}||T_{Ui}, \tag{3}$$

where ID_{Ui} is the unique public identification of the unknown node U_i, REG represents registration request, N_{Ui} is a cryptographic nonce, S_{Ui} is the sequence number, and T_{Ui} is the time stamp by the unknown node U_i. The unknown node U_i computes one way cryptographic hash of the message M_{Ui} using an agreed upon hash function h. The computed hash of the message M_{Ui} is encrypted using the private key PR_{Ui} of the unknown node U_i to obtain $C_{Uih} = E(PR_{Ui}, h(M_{Ui}))$. The message M_{Ui} and the hash value are then encrypted using the secret key K as $C_{Ui} = E(K, M_{Ui}||E(PR_{Ui}, h(M_{Ui})))$ and transmitted to the beacon node B_i as follows,

$$U_i \xrightarrow{E(K, M_{Ui}||E(PR_{Ui}, h(M_{Ui})))} B_i. \tag{4}$$

The localization algorithm identifies a sensor node with the help of the application layer identifier ID_{Ui}. Legitimate nodes are able to decrypt the encrypted application layer messages and, hence, are able to retrieve the application layer identifier ID_{Ui}. The cryptographic nonce N_{Ui} serves the purpose of authentication as a legitimate receiver should be able to retrieve it from the encrypted message and send it back. The time stamp T_{Ui} serves as a defense against the traffic replay and other man in the middle attacks. Sequence numbers prevent disruption of traffic by an attacker by reordering the packets. The sequence numbers are unpredictable and are generated according to Algorithm 1. The length of the message is added to the sequence number to ensure that each message has a unique sequence number and that any messages received out of order can be identified. When a receiver receives a message with a sequence number that is not an expected number, it knows that some data have been lost or delivered out of order. This also helps thwart selective forwarding attack.

Algorithm 1 Sequence number generator.

```
1:  procedure SEQUENCENUMBER(SeqNumber)
2:      if (SessionStart==true) then
3:          SeqNumber ← Secure.Random(value)
4:      else
5:          SeqNumber ← SeqNumber + Length(Message)
6:          if (SeqNumber > MAXSEQNUM) then
7:              SeqNumber ← 0
8:          end if
9:      end if
10:     return SeqNumber
11: end procedure
```

Upon receiving the encrypted message, the beacon node B_i decrypts it as $D(K, C_{Ui})$. The beacon node B_i is able to determine the length of the message after this decryption process. The encrypted hash value is further retrieved using the public key PU_{Ui} of the unknown node U_i as $D(PU_{Ui}, C_{Uih})$. If the beacon node B_i is able to successfully decrypt the encrypted hash using the public key PU_{Ui}, it is confirmed that the message was indeed sent by the sensor node U_i as no other node could have encrypted the hash using the private key PR_{Ui} of the sensor node U_i. The beacon node B_i also computes the hash value of the message using the hash function h. If the computed and the retrieved hash values do not match, the message is discarded. However, if the two values match, then the beacon node B_i prepares the following message M_{Bi} for the unknown node U_i.

$$M_{Bi} \leftarrow ID_{Bi}||N_{Ui}||N_{Bi}||S_{Bi}||T_{Bi}, \tag{5}$$

where ID_{Bi} is the unique identification of the beacon node B_i, N_{Ui} is the cryptographic nonce which was sent by the unknown node U_i, N_{Bi} is the cryptographic nonce prepared by the beacon node B_i, S_{Bi} is the sequence number generated according to Algorithm 1, and T_{Bi} is the time stamp by the beacon node B_i. The hash of this message is encrypted using the private key PR_{Bi} of the beacon node B_i and concatenated with the message M_{Bi}. This is then encrypted using the secret key K to produce ciphertext $C_{Bi} = E(K, M_{Bi}||E(PR_{Bi}, h(M_{Bi})))$, which is transmitted to the unknown node U_i.

$$B_i \xrightarrow{E(K, M_{Bi}||E(PR_{Bi}, h(M_{Bi})))} U_i \tag{6}$$

The unknown node U_i decrypts this message as $D(K, C_{Bi})$. If it is unable to retrieve the cryptographic nonce N_{Ui}, the message is discarded and is not processed further. However,

if the nonce is retrieved successfully, it generates a code word W_{Ui} and sends it to the beacon node B_i using the following message.

$$M_{Ui} \leftarrow ID_{Ui}||N_{Bi}||W_{Ui}||S_{Ui}||T_{Ui}. \quad (7)$$

Similar to the previous messages, ciphertext $C_{Ui} = E(K, M_{Ui}||E(PR_{Ui}, h(M_U i)))$ is prepared and sent to the beacon node B_i as follows,

$$U_i \xrightarrow{E(K,M_{Ui}||E(PR_{Ui},h(M_U i)))} B_i. \quad (8)$$

The beacon node B_i decrypts the received message as $D(K, C_{Ui})$. If it cannot find the nonce N_{Bi} in the message, it discards the message and does not process it further. However, if the nonce N_{Bi} is retrieved successfully, it proceeds to process the received code word W_{Ui}. The beacon node B_i adds a salt to the code word W_{Ui}, computes the hash of the salted code word. The hash is stored along with the salt and the ID_{Ui} of the unknown node U_i. This process is performed as depicted in Algorithm 2. The registration procedure of an unknown node with a beacon node is illustrated in Figure 1.

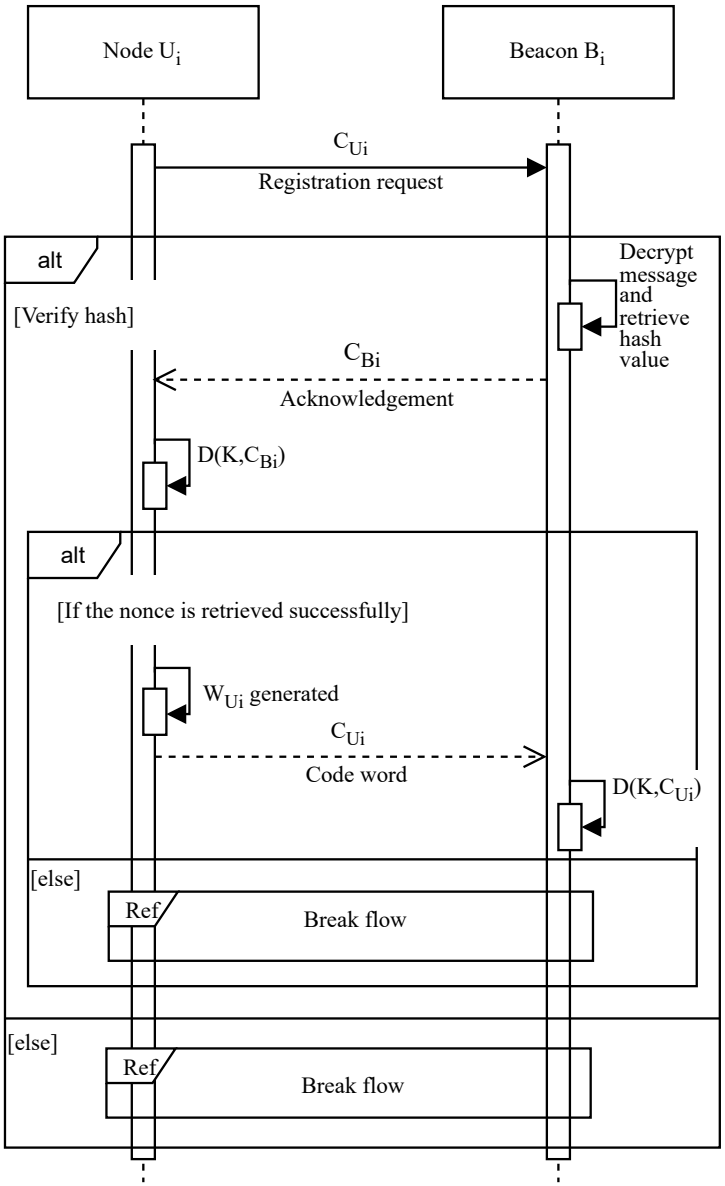

Figure 1. Registration procedure of an unknown node with a beacon node.

Algorithm 2 Code Word Storage.

```
procedure UIDSTORAGE(ID_Ui, CodeWord, SaltLength)
    AllowedChars ← "abcdefghijklmnopqrstuvwxyz"
    AllowedChars ← AllowedChars + "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    AllowedChars ← AllowedChars + "0123456789"
    AllowedChars ← AllowedChars + "!@#$%^&*"
    MAX ← Length(AllowedChars)
    Salt ← ""
    Count ← 0
    while (Count < SaltLength) do
        RandNum ← Secure.Random(0, MAX)
        Salt ← Salt + AllowedChars[RandNum]
        Count ← Count + 1
    end while
    WordHash ← HashAlgo(CodeWord + Salt)
    Handle ← Open(SecureFile)
    Write(Handle, ID_Ui, WordHash, SaltLength)
    Close(Handle)
end procedure
```

Subsequently, if the unknown node U_i wants to communicate with another unknown node U_j, the latter asks U_i to provide its surety. The unknown node U_j prepares the following message for this purpose.

$$M_{Uj} \leftarrow ID_{Uj}||SURETY||N_{Uj}||S_{Uj}||T_{Uj} \tag{9}$$

Next, this message and its hash are encrypted as $C_{Uj} = E(K, M_{Uj}||E(PR_{Uj}, h(M_{Uj})))$ and sent to the node U_i, as follows,

$$U_j \xrightarrow{E(K,M_{Uj}||E(PR_{Uj},h(M_{Uj})))} U_i. \tag{10}$$

After decrypting this message and confirming its validity with the help of the hash, the unknown node responds with the following message.

$$C_{WUi} \leftarrow E(PU_{Bi}, W_{Ui}||E(PR_{Ui}, h(W_{Ui}))) \tag{11}$$

$$M_{Ui} \leftarrow ID_{Ui}||N_{Uj}||N_{Ui}||ID_{Bi}||C_{WUi}||S_{Ui}||T_{Ui} \tag{12}$$

$$U_i \xrightarrow{E(K,M_{Ui}||E(PR_{Ui},h(M_{Ui})))} U_j. \tag{13}$$

The unknown node U_j decrypts and verifies this message using the hash function. It also retrieves the encrypted code word C_{WUi} and sends it to the beacon node B_i for verification.

$$M_{Uj} \leftarrow ID_{Uj}||N_{Uj}||C_{WUi}||S_{Uj}||T_{Uj} \tag{14}$$

The encrypted text $C_{Uj} = E(K, M_{Uj}||E(PR_{Uj}, h(M_{Uj})))$ is prepared and sent to the beacon node B_i as follows.

$$U_j \xrightarrow{E(K,M_{Uj}||E(PR_{Uj},h(M_{Uj})))} B_i. \tag{15}$$

The beacon node B_i decrypts and checks the validity of the message as described previously. It then decrypts C_{WUi} and retrieves the code word of the unknown node U_i. It confirms its validity by computing its hash using the stored salt and then comparing with the stored value of the hash. It then communicates the result back to the unknown node U_j.

$$M_{Bi} \leftarrow ID_{Bi}||N_{Uj}||N_{Bi}||RESULT||S_{Bi}||T_{Bi} \tag{16}$$

$$B_i \xrightarrow{E(K,M_{Bi}||E(PR_{Bi},h(M_{Bi})))} U_j, \tag{17}$$

where the variable *RESULT* contains *OK* if the code word is verified or *NOK* otherwise. The node U_j proceeds with its data exchange with the node U_i in the former case and drops the communication in the latter instance. The procedure to store the code word and to establish trust between two nodes takes place only once. The authentication process of two unknown nodes with the help of a beacon node is illustrated in Figure 2.

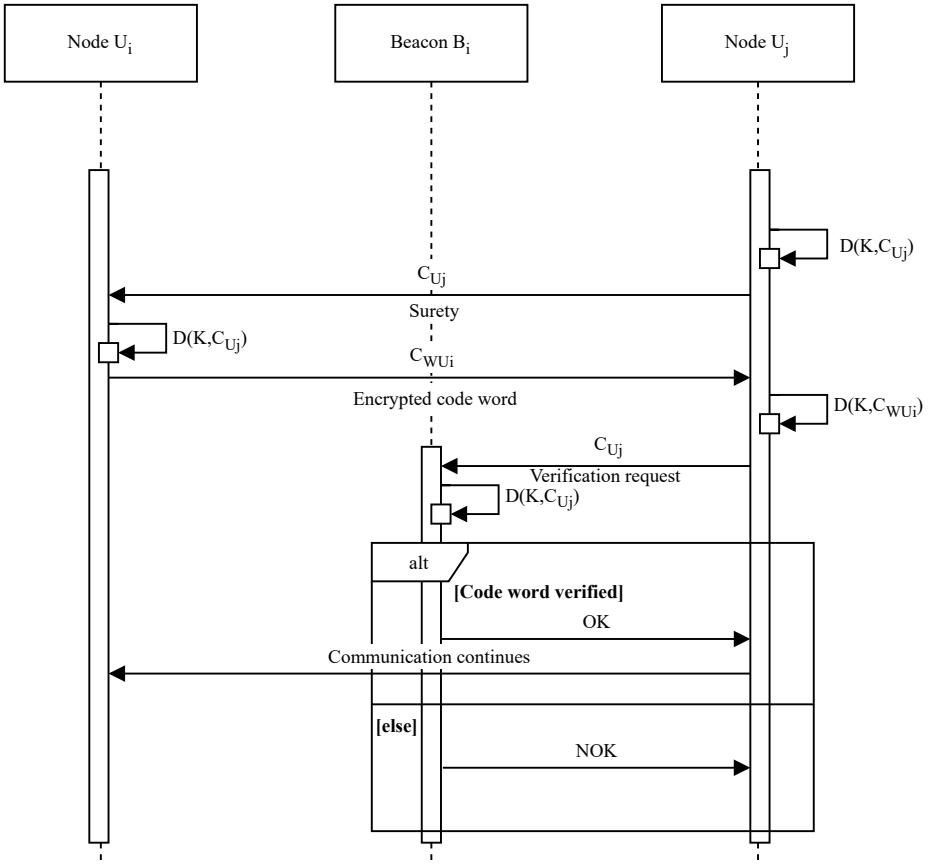

Figure 2. Authentication process between two unknown nodes.

The beacon nodes broadcast their positions using beacon messages at regular intervals. The message may contain the identification of the beacon node and a time stamp. For example, a typical beacon message M_{Bi} of a beacon node B_i is as follows.

$$M_{Bi} \leftarrow ID_{Bi}||(x_{Bi}, y_{Bi})||hc_{Bi}||T_{Bi}, \tag{18}$$

where ID_{Bi} is the unique identification of the beacon node B_i, (x_{Bi}, y_{Bi}) is its position information, hc_{Bi} is a variable to store the hop count and is initialized to zero, and T_{Bi} is the time stamp by the beacon node B_i. The beacon node B_i computes one way cryptographic hash of the message using a hash function h. The computed cryptographic hash of the message M_{Bi} is encrypted using the private key PR_{Bi} of the beacon node B_i to obtain $C_{Bih} = E(PR_{Bi}, h(M_{Bi}))$. The message M_{Bi} and its encrypted hash are then broadcast. The broadcast message is $M_{Bi}||E(PR_{Bi}, h(M_{Bi}))$, and is depicted as below.

$$B_i \xrightarrow{M_{Bi}||E(PR_{Bi},h(M_{Bi}))} \text{All} \tag{19}$$

When an unknown node U_i receives this message, it decrypts the encrypted hash using the public key PU_{Bi} of the beacon node B_i using $D(PU_{Bi}, C_{Bih})$. It also computes the one

way cryptographic hash of the received message M_{Bi} using the same hash function which was used by the beacon node B_i. If $h(M_{Bi}) \neq D(PU_{Bi}, C_{Bih})$, that is, the hash computed by the unknown node U_i does not match the received hash value, then the unknown node U_i discards the message. However, if the computed hash and the received hash values match each other, i.e., $h(M_{Bi}) = D(PU_{Bi}, C_{Bih})$, then the message is considered legitimate. The unknown node U_i stores the position (x_{Bi}, y_{Bi}) of the beacon node B_i. Moreover, the unknown node U_i increments the hop count variable hc_{Bi}, and constructs a message M_{Ui} for the next hop neighbor as follows.

$$M_{Ui} \leftarrow ID_{Ui}||M_{Bi}||hc_{Bi}||T_{Ui}, \tag{20}$$

where ID_{Ui} is the unique identification of the unknown node U_i, hc_{Bi} is the hop count variable with incremented value, and T_{Ui} is the time stamp by the unknown node U_i. The unknown node U_i also computes one way cryptographic hash of the message using the hash function h. The cryptographic hash value of the message M_{Ui} is then encrypted using the private key PR_{Ui} of the unknown node U_i to obtain $C_{Uih} = E(PR_{Ui}, h(M_{Ui}))$. The message M_{Ui} containing the new hop count and the encrypted hash value are then sent to the next hop neighbor U_j as follows.

$$U_i \xrightarrow{M_{Ui}||E(PR_{Ui}, h(M_{Ui}))} U_j \tag{21}$$

Upon receiving this message, the node U_j performs a procedure similar to the procedure performed by the node U_i when it received the beacon message. It decrypts the encrypted hash as $D(PU_{Ui}, C_{Uih})$ and also computes the hash value $h(M_{Ui})$. The message is processed if the two hash values match and is discarded otherwise. The message is propagated further until it reaches another beacon node.

After a beacon node B_i has obtained position coordinates of other beacon nodes and the hop count to them, it computes the average hop size or the correction factor c_{Bi} using Equation (1), as stated earlier. The beacon node B_i, then prepares a message M_{Bi} to propagate this correction factor as follows.

$$M_{Bi} \leftarrow ID_{Bi}||(x_{Bi}, y_{Bi})||c_{Bi}||T_{Bi} \tag{22}$$

This message and its cryptographic hash encrypted using the private key of the beacon node B_i are concatenated as $M_{Bi}||E(PR_{Bi}, h(M_{Bi}))$. This is then propagated through the next hop neighbors U_i as follows.

$$B_i \xrightarrow{M_{Bi}||E(PR_{Bi}, h(M_{Bi}))} U_i \tag{23}$$

When an unknown node U_i receives the correction factor c_{Bi}, it then computes the distance to the beacon node B_i from which it received the correction factor. The distance is calculated by multiplying the hop count hc_{Bi} to the correction factor c_{Bi}, i.e., $hc_{Bi} \times c_{Bi}$. After an unknown node U_i has received the position coordinates of at least three beacon nodes and estimated distance to them, the unknown node performs multilateration to estimate its position as already described in Section 4. Propagation of beacon messages and correction factor is illustrated in Figure 3.

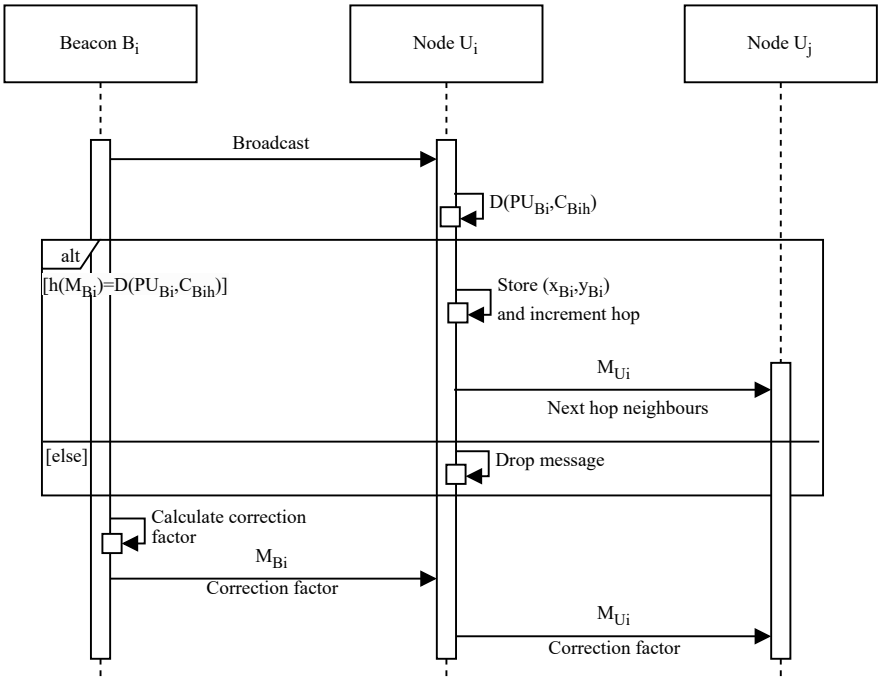

Figure 3. Propagation of beacon messages and correction factor.

6. Simulation Results

We evaluate the performance of our proposed Secure DV-Hop localization algorithm using simulation experiments. A sensor field with dimensions of 100 m × 100 m is considered for the experiments. The number of sensor nodes is 100. The number of beacon nodes and the number of malicious nodes are varied for the performance evaluation. Moreover, the performance is also evaluated both in the absence and presence of the malicious nodes. The malicious nodes use different types of security attacks which include wormhole, Sybil, tampering, traffic replay, and selective forwarding attacks. In addition, performance of the proposed Secure DV-Hop algorithm is compared with those of basic DV-Hop [31], label-based DV-Hop (LBDV-Hop) [33] and Security Positioning DV-Hop (SPDV-Hop) [39] localization algorithms.

Localization error of a single node is the distance between the actual position and the estimated position. Therefore, if the actual position of a sensor node U_i is (x_{Ui}, y_{Ui}) and the estimated position is $(\hat{x}_{Ui}, \hat{y}_{Ui})$, then the localization error, e_L, of an unknown node U_i is given by,

$$e_L = \sqrt{(\hat{x}_{Ui} - x_{Ui})^2 - (\hat{y}_{Ui} - y_{Ui})^2}. \tag{24}$$

The average normalized localization error, e_{LN}, of the sensor network is given by,

$$e_{LN} = \frac{\sum_{i=1}^{N} \sqrt{(\hat{x}_{Ui} - x_{Ui})^2 - (\hat{y}_{Ui} - y_{Ui})^2}}{NR}, \tag{25}$$

where N is the total number of unknown nodes and R is the radio range of a sensor node.

The localization efficiency η_L is the ratio of the number of unknown sensor nodes which are able to estimate their positions to the total number of unknown sensor nodes [48]. The unknown sensor nodes which are able to ascertain their positions may be termed as settled nodes. If the total number of settled nodes is represented by N_s, then the localization efficiency, η_L, is given by

$$\eta_L = \frac{N_s}{N} \times 100. \tag{26}$$

In Figure 4, we plot the average normalized localization error for the basic DV-Hop, SPDV-Hop, LBDV-Hop, and the Secure DV-Hop localization algorithms as the number of

beacon nodes is varied in the sensor field in the absence of any attack. The localization efficiency of these algorithms against the varying number of beacon nodes in the absence of any attack is plotted in Figure 5. From both Figures 4 and 5, it can be observed that all the compared localization algorithms perform as good as the basic DV-Hop methods in the absence of any attack. Therefore, these algorithms work in the same fashion under normal circumstances. This also validates the localization performance of the proposed algorithm.

However, all these algorithms perform differently when malicious nodes are introduced in the sensor network. This can be observed from Figure 6 where average normalized localization error of each of the proposed Secure DV-Hop and three compared algorithms is plotted for varying number of malicious nodes and 20% beacon nodes. It is evident that, when the malicious nodes are present, the basic DV-Hop, SPDV-Hop, and LBDV-Hop localization algorithms do not perform the way as they do in the absence of any attack. The average normalized localization error for each of these algorithms increases significantly as the count of the malicious nodes is increased while keeping the number of beacon nodes fixed at 20%. However, our proposed Secure DV-Hop localization algorithm remains unaffected and shows consistent results in the presence of any number of malicious nodes.

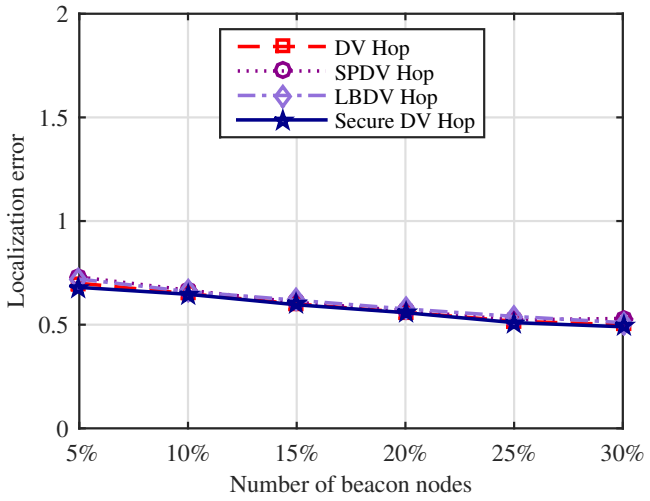

Figure 4. Average normalized localization error as the number of beacon nodes is varied in the absence of an attack.

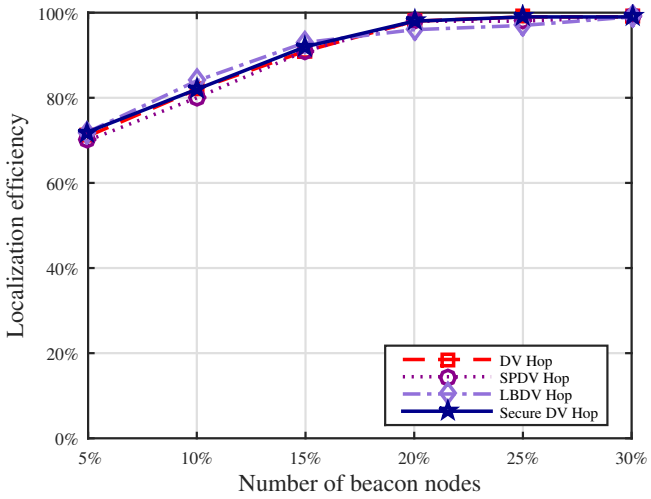

Figure 5. Localization efficiency as the number of beacon nodes is varied in the absence of an attack.

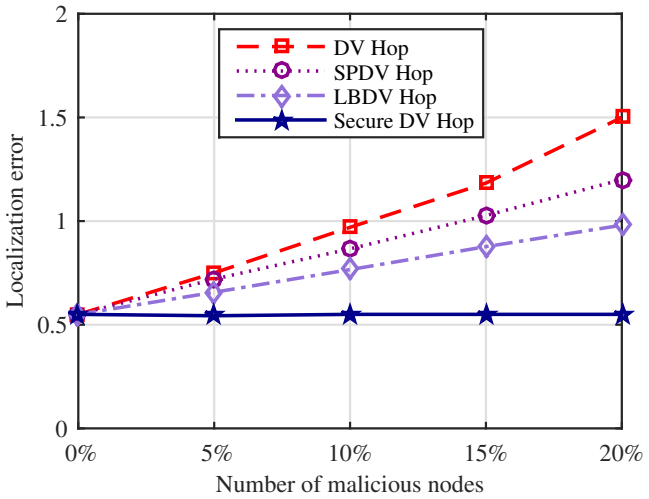

Figure 6. Average normalized localization error as the number of malicious nodes is varied with fixed 20% beacon nodes in the sensor network.

We plot average normalized localization error with a varying number of beacon nodes and a fixed 10% number of malicious nodes in Figure 7. It can be observed that the localization error decreases for all the compared localization algorithms as the number of beacon nodes is increased in the sensor field. However, in the case of DV-Hop, SPDV-Hop, and LBDV-Hop, when we compare their performance in Figure 4 in the absence of attack to their performance in Figure 7 when attacked by 10% hostile nodes, a degradation in the performance is clearly observed. For example, from Figure 4, in the absence of attack, the localization error resulting from DV-Hop with 20% beacon nodes is almost 0.5. However, in the case of Figure 7, the localization error with 20% beacon nodes in the presence of 10% malicious nodes is almost 1. This is twice as high as the error in Figure 4 for the same number of beacon nodes. Similar observation can be made for SPDV-Hop and LBDV-Hop as well. On the other hand, the Secure DV-Hop algorithm remains robust and its localization results remain unaffected by the malicious nodes. If we compare the performance of our proposed Secure DV-Hop localization algorithm in Figures 4 and 7, we see that it provides similar performance in the presence or absence of the malicious nodes.

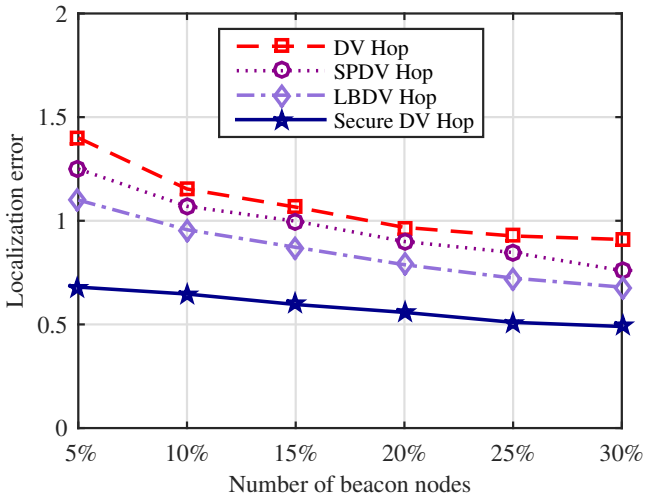

Figure 7. Average normalized localization error with varying number of beacon nodes when the number of malicious nodes is fixed at 10%.

Localization efficiency of each of the compared algorithms is plotted in Figure 8 against a varying number of malicious nodes when the number of beacon nodes is 20%. The localization efficiencies of the DV-Hop, SPDV-Hop, and LBDV-Hop algorithms decrease

as the number of malicious nodes in the sensor field increases. This implies that lesser and lesser number of unknown sensor nodes are able to estimate their positions as the number of malicious nodes increases. On the other hand, the Secure DV-Hop localization algorithm is not affected and its localization efficiency does not change with the varying number of malicious nodes in the sensor field.

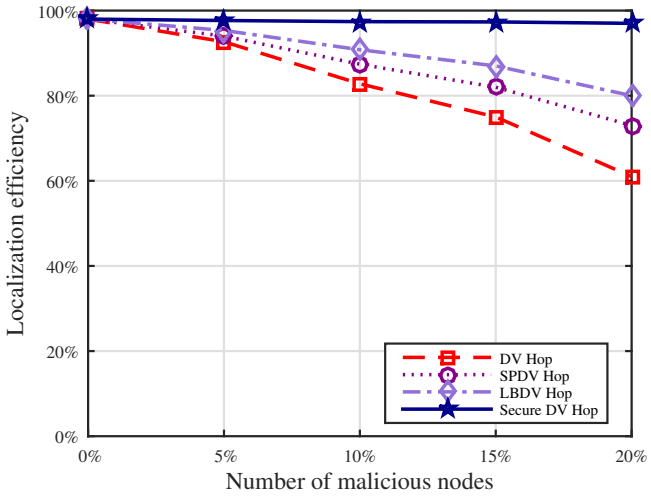

Figure 8. Localization efficiency when the number of malicious nodes is varied and the number of beacon nodes is fixed at 20%.

In Figure 9, we plot localization efficiencies of DV-Hop, SPDV-Hop, LBDV-Hop, and Secure DV-Hop algorithms against a varying number of beacon nodes in the presence of 10% malicious nodes. Results in Figure 9 corroborate previous findings. Although localization efficiencies of DV-Hop, SPDV-Hop, and LBDV-Hop increase with an increase in the number of beacon nodes in the sensor field, these do not attain the same values as they do in the absence of malicious nodes. However, Secure DV-Hop localization algorithm once again shows consistent and robust performance even in the presence of adverse conditions. From Figures 5 and 9, it can be readily observed that the localization efficiency of the Secure DV-Hop algorithm remains unaffected in the presence or absence of the malicious nodes.

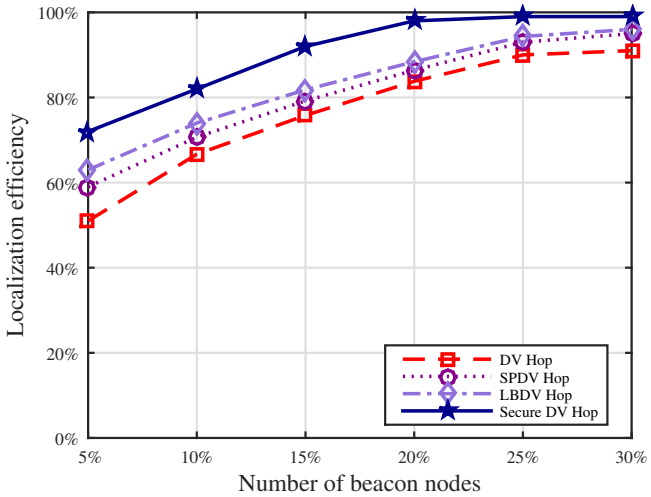

Figure 9. Localization efficiency with varying number of beacon nodes when the number of malicious nodes is fixed at 10%.

This robust performance of Secure DV-Hop localization algorithm can be attributed to effective authentication and communication implemented through encryption. SPDV-Hop provides protection against Sybil attack only and LBDV-Hop is designed for security

against wormhole attack alone. However, both these algorithms do not provide effective protection against other types of attacks, such as tampering, selective forwarding, and traffic replay. On the other hand, robust authentication and communication implemented by the Secure DV-Hop localization algorithm protect against all these types of attacks. Hence, it remains unaffected by these security attacks.

7. Conclusions

In this work, we have proposed distance vector hop-based secure and robust localization algorithm for wireless sensor networks. The algorithm uses secret and public key cryptography to secure the localization process against different types of security attacks. These attacks include wormhole, Sybil, selective forwarding, traffic replay and tampering attacks. A number of simulation experiments were performed to evaluate the performance of the proposed algorithm both in the presence and the absence of malicious nodes using these attacks. The results were compared with the basic distance vector hop method and two secure algorithms based on distance vector hop localization. The average normalized localization error and the localization efficiency were measured in the presence, as well as in the absence of malicious nodes. The results revealed that the performance of the compared algorithms was severely affected in the presence of malicious nodes. However, the proposed secure localization algorithm provided secure and robust performance in either scenario. As a result of the countermeasures, the algorithm provided similar performance in the presence of adversaries as it did in the absence of any attacks. The secure localization algorithm can be implemented in a wireless sensor network which is deployed in a hostile environment and where unknown sensor nodes have to estimate their position coordinates. Future work includes improvement of localization performance of the algorithm and its implementation and practical evaluation in a real wireless sensor network.

Author Contributions: Conceptualization, M.F.-i.-A., M.H.C. and A.H.; methodology, M.F.-i.-A.; software, R.A.; validation, M.H.C. and A.H.; formal analysis, M.F.-i.-A.; investigation, M.F.-i.-A. and S.R.H.; resources, R.A.; data curation, M.H.C. and A.H.; writing—original draft preparation, M.F.-i.-A., M.H.C. and A.H.; writing—review and editing, M.F.-i.-A., R.A. and S.R.H.; visualization, M.H.C. and A.H.; supervision, M.F.-i.-A.; project administration, M.F.-i.-A., R.A. and S.R.H. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the internal research start-up fund, reference: 1012606FA1, from University of East Anglia (UEA), Norwich, UK.

Data Availability Statement: Data sharing not applicable.

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

References

1. Najarro, L.A.C.; Song, I.; Kim, K. Fundamental limitations and State-of-the-art Solutions for Target Node Localization in WSNs: A Review. *IEEE Sens. J.* **2022**, 22, 23661–23682. [CrossRef]
2. Benahmed, T.; Benahmed, K. Optimal barrier coverage for critical area surveillance using wireless sensor networks. *Int. J. Commun. Syst.* **2019**, *32*, e3955. [CrossRef]
3. Lahoz-Monfort, J.J.; Magrath, M.J. A comprehensive overview of technologies for species and habitat monitoring and conservation. *BioScience* **2021**, *71*, 1038–1062. [CrossRef] [PubMed]
4. Abdollahi, A.; Rejeb, K.; Rejeb, A.; Mostafa, M.M.; Zailani, S. Wireless sensor networks in agriculture: Insights from bibliometric analysis. *Sustainability* **2021**, *13*, 12011. [CrossRef]
5. Bravo-Arrabal, J.; Zambrana, P.; Fernandez-Lozano, J.; Gomez-Ruiz, J.A.; Barba, J.S.; García-Cerezo, A. Realistic deployment of hybrid wireless sensor networks based on ZigBee and LoRa for search and Rescue applications. *IEEE Access* **2022**, *10*, 64618–64637. [CrossRef]
6. Nakas, C.; Kandris, D.; Visvardis, G. Energy efficient routing in wireless sensor networks: A comprehensive survey. *Algorithms* **2020**, *13*, 72. [CrossRef]
7. Ullah, I.; Youn, H.Y. A novel data aggregation scheme based on self-organized map for WSN. *J. Supercomput.* **2019**, *75*, 3975–3996. [CrossRef]
8. Huang, X.; Han, D.; Cui, M.; Lin, G.; Yin, X. Three-Dimensional Localization Algorithm Based on Improved A* and DV-Hop Algorithms in Wireless Sensor Network. *Sensors* **2021**, *21*, 448. [CrossRef]

9. Messous, S.; Liouane, H.; Cheikhrouhou, O.; Hamam, H. Improved Recursive DV-Hop Localization Algorithm with RSSI Measurement for Wireless Sensor Networks. *Sensors* **2021**, *21*, 4152. [CrossRef]
10. Li, G.; Xu, M.; Teng, G.; Yang, W.; Mak, S.L.; Li, C.Y.; Lee, C.C. A Voronoi Diagram-Based Grouping Test Localization Scheme in Wireless Sensor Networks. *Electronics* **2022**, *11*, 2961. [CrossRef]
11. Jing, N.; Zhang, B.; Wang, L. A Novel Anchor-Free Localization Method Using Cross-Technology Communication for Wireless Sensor Network. *Electronics* **2022**, *11*, 4025. [CrossRef]
12. Liu, W.; Luo, X.; Wei, G.; Liu, H. Node localization algorithm for wireless sensor networks based on static anchor node location selection strategy. *Comput. Commun.* **2022**, *192*, 289–298. [CrossRef]
13. Kaur, A.; Gupta, G.P.; Mittal, S. Comparative study of the different variants of the dv-hop based node localization algorithms for wireless sensor networks. *Wirel. Pers. Commun.* **2022**, *123*, 1625–1667. [CrossRef]
14. Yuvarasu, M.; Balaram, A.; Chandramohan, S.; Sharma, D.K. A Performance Analysis of an Enhanced Graded Precision Localization Algorithm for Wireless Sensor Networks. *Cybern. Syst.* **2023**, 1–16. [CrossRef]
15. Liouane, H.; Messous, S.; Cheikhrouhou, O. Regularized least square multi-hops localization algorithm based on DV-Hop for wireless sensor networks. *Telecommun. Syst.* **2022**, *80*, 349–358. [CrossRef]
16. Luomala, J.; Hakala, I. Adaptive range-based localization algorithm based on trilateration and reference node selection for outdoor wireless sensor networks. *Comput. Netw.* **2022**, *210*, 108865. [CrossRef]
17. Liu, J.; Liu, M.; Du, X.; Stanimirovi, P.S.; Jin, L. An improved DV-Hop algorithm for wireless sensor networks based on neural dynamics. *Neurocomputing* **2022**, *491*, 172–185. [CrossRef]
18. Du, J.; Yuan, C.; Yue, M.; Ma, T. A novel localization algorithm based on RSSI and multilateration for indoor environments. *Electronics* **2022**, *11*, 289. [CrossRef]
19. Zhang, H.; Yang, J.; Qin, T.; Fan, Y.; Li, Z.; Wei, W. A Multi-Strategy Improved Sparrow Search Algorithm for Solving the Node Localization Problem in Heterogeneous Wireless Sensor Networks. *Appl. Sci.* **2022**, *12*, 5080. [CrossRef]
20. Nguyen, T.N.; Le, V.V.; Chu, S.I.; Liu, B.H.; Hsu, Y.C. Secure Localization Algorithms Against Localization Attacks in Wireless Sensor Networks. *Wirel. Pers. Commun.* **2022**, *127*, 767–792. [CrossRef]
21. Kim, T.H.; Goyat, R.; Rai, M.K.; Kumar, G.; Buchanan, W.J.; Saha, R.; Thomas, R. A Novel Trust Evaluation Process for Secure Localization Using a Decentralized Blockchain in Wireless Sensor Networks. *IEEE Access* **2019**, *7*, 184133–184144. [CrossRef]
22. Mukhopadhyay, B.; Srirangarajan, S.; Kar, S. RSS-Based Localization in the Presence of Malicious Nodes in Sensor Networks. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–16. [CrossRef]
23. Höglund, J.; Lindemer, S.; Furuhed, M.; Raza, S. PKI4IoT: Towards public key infrastructure for the Internet of Things. *Comput. Secur.* **2020**, *89*, 101658. [CrossRef]
24. Gu, Y.; Shen, L.; Zhang, F.; Xiong, J. Provably Secure Linearly Homomorphic Aggregate Signature Scheme for Electronic Healthcare System. *Mathematics* **2022**, *10*, 2588. [CrossRef]
25. Al-Zubaidie, M.; Zhang, Z.; Zhang, J. REISCH: Incorporating Lightweight and Reliable Algorithms into Healthcare Applications of WSNs. *Appl. Sci.* **2020**, *10*, 2007. [CrossRef]
26. Revanesh, M.; Acken, J.M.; Sridhar, V. DAG block: Trust aware load balanced routing and lightweight authentication encryption in WSN. *Future Gener. Comput. Syst.* **2022**, *140*, 402–421.
27. Nagarajan, M.; Rajappa, M.; Teekaraman, Y.; Kuppusamy, R.; Thelkar, A.R. Renovated XTEA encoder architecture-based lightweight mutual authentication protocol for RFID and green wireless sensor network applications. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 8876096. [CrossRef]
28. Mezrag, F.; Bitam, S.; Mellouk, A. An efficient and lightweight identity-based scheme for secure communication in clustered wireless sensor networks. *J. Netw. Comput. Appl.* **2022**, *200*, 103282. [CrossRef]
29. 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. [CrossRef]
30. Chen, C.M.; Chen, Z.; Kumari, S.; Lin, M.C. LAP-IoHT: A lightweight authentication protocol for the internet of health things. *Sensors* **2022**, *22*, 5401. [CrossRef] [PubMed]
31. Niculescu, D.; Nath, B. DV Based Positioning in Ad Hoc Networks. *Telecommun. Syst.* **2003**, *22*, 267–280. [CrossRef]
32. Niculescu, D.; Nath, B. Ad hoc positioning system (APS) using AOA. In Proceedings of the IEEE INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428), San Francisco, CA, USA, 30 March–3 April 2003; Volume 3, pp. 1734–1743. [CrossRef]
33. Chen, H.; Lou, W.; Wang, Z.; Wu, J.; Wang, Z.; Xia, A. Securing DV-Hop localization against wormhole attacks in wireless sensor networks. *Pervasive Mob. Comput.* **2015**, *16*, 22–35. [CrossRef]
34. Li, J.; Wang, D.; Wang, Y. Security DV-hop localisation algorithm against wormhole attack in wireless sensor network. *IET Wirel. Sens. Syst.* **2018**, *8*, 68–75. [CrossRef]
35. Prashar, D.; Rashid, M.; Siddiqui, S.T.; Kumar, D.; Nagpal, A.; AlGhamdi, A.S.; Alshamrani, S.S. SDSWSN—A Secure Approach for a Hop-Based Localization Algorithm Using a Digital Signature in the Wireless Sensor Network. *Electronics* **2021**, *10*, 3074. [CrossRef]
36. Liu, X.; Su, S.; Han, F.; Liu, Y.; Pan, Z. A Range-Based Secure Localization Algorithm for Wireless Sensor Networks. *IEEE Sens. J.* **2019**, *19*, 785–796. [CrossRef]

37. Tang, Q.; Wang, J. A secure positioning algorithm against Sybil attack in wireless sensor networks based on number allocating. In Proceedings of the 2017 IEEE 17th International Conference on Communication Technology (ICCT), Chengdu, China, 27–30 October 2017; pp. 932–936. [CrossRef]
38. Douceur, J.R. The Sybil Attack. In *Peer-to-Peer Systems*; Druschel, P., Kaashoek, F., Rowstron, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 251–260.
39. Dong, S.; Zhang, X.G.; Zhou, W.G. A Security Localization Algorithm Based on DV-Hop Against Sybil Attack in Wireless Sensor Networks. *J. Electr. Eng. Technol.* **2020**, *15*, 919–926. [CrossRef]
40. Council, N.R. *Trust in Cyberspace*; The National Academies Press: Washington, DC, USA, 1999; p. 126. [CrossRef]
41. Sidhu, S.; Mohd, B.J.; Hayajneh, T. Hardware security in IoT devices with emphasis on hardware Trojans. *J. Sens. Actuator Netw.* **2019**, *8*, 42. [CrossRef]
42. Mehrabi, M.A.; Doche, C.; Jolfaei, A. Elliptic curve cryptography point multiplication core for hardware security module. *IEEE Trans. Comput.* **2020**, *69*, 1707–1718. [CrossRef]
43. Khan, A.A.; Laghari, A.A.; Shaikh, A.A.; Dootio, M.A.; Estrela, V.V.; Lopes, R.T. A blockchain security module for brain–computer interface (BCI) with multimedia life cycle framework (MLCF). *Neurosci. Inform.* **2022**, *2*, 100030. [CrossRef]
44. Butun, I.; Sari, A.; Österberg, P. Hardware security of fog end-devices for the internet of things. *Sensors* **2020**, *20*, 5729. [CrossRef]
45. Pearson, B.; Luo, L.; Zhang, Y.; Dey, R.; Ling, Z.; Bassiouni, M.; Fu, X. On misconception of hardware and cost in IoT security and privacy. In Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–7.
46. Kim, T.; Park, J.; Woo, J.; Jeon, S.; Huh, J. Shieldstore: Shielded in-memory key-value storage with sgx. In Proceedings of the Fourteenth EuroSys Conference 2019, Dresden, Germany, 25–28 March 2019; pp. 1–15.
47. Farooq-i-Azam, M.; Ayyaz, M.N. Location and position estimation in wireless sensor networks. In *Wireless Sensor Networks: Current Status and Future Trends*; CRC: Boca Raton, FL, USA, 2016; pp. 179–214.
48. Farooq-I-Azam, M.; Ni, Q.; Ansari, E.A. Intelligent Energy Efficient Localization Using Variable Range Beacons in Industrial Wireless Sensor Networks. *IEEE Trans. Ind. Inform.* **2016**, *12*, 2206–2216. [CrossRef]