A Centralized Approach for Secure Location Verification in Wireless Sensor Networks

Abbas Ghebleh, Maghsoud Abbaspour, and Saman Homayounnejad

Electrical and Computer Department, Shahid Beheshti University, Tehran, Iran
ab.ghebleh@mail.sbu.ac.ir, maghsoud@sbu.ac.ir,
samaaan@aut.ac.ir

Abstract. Location information of sensors is crucial in many applications of Wireless Sensor Networks (WSNs). A lot of work has been done for estimating sensors' location in the network but these approaches are mostly designed without considering security issues which is critical in WSNs. So we should verify the location of sensors to ensure that estimated location is correct. In this paper we propose a novel approach for verifying sensors' location having minor overhead on the network while not using any additional hardware. This approach is centralized and the verification process is performed in the base station that is much more powerful than sensors and is able to perform more sophisticated tests. Simulation studies verify that this approach is able to detect different attacks and anomalies in sensors' location.

Keywords: Wireless Sensor Network, Location Verification, Security.

1 Introduction

Wireless sensor networks (WSNs) have been used in many applications and still have many potential applications. WSNs are composed of a large number of small, low cost, and low power nodes that are equipped with one or more sensors and communicate wirelessly with each other [1].

Many applications proposed for WSNs are based on the knowledge of sensors' locations, such as environment monitoring, target tracking, etc. Therefore sensors should somehow discover their location in the network. This process is called *localization*. Although the simplest way of providing accurate location information is to equip each sensor with a GPS, this is too expensive for sensors; the other way is to use some special nodes that know their location in network and help sensors to estimate their location. These nodes are called *locators* [2].

Recently, a lot of works have been done for localization in WSNs [3-10]. However, these approaches are designed without considering security; while in many applications, WSNs are deployed in unattended and even hostile environments that make WSNs vulnerable to malicious attacks like Wormhole [11]. Security issues should be considered to ensure the operation of the WSNs otherwise, a compromised or malicious node can claim virtually any location and ruins the network.

Some works have been done that locate sensors securely. These works can be categorized in two fields: i) *secure localization* in which the localization process is

G. Parr and P. Morrow (Eds.): S-Cube 2010, LNICST 57, pp. 151–163, 2011.

performed in a secure way like [12-15] and ii) *location verification* in which sensors somehow estimate their location and then some other nodes that are called *verifiers* verify their claimed location like [16-20].

To our knowledge, previous works on localization and location verification mostly are decentralized approaches and sensor centric, i.e. localization and location verification process is performed by sensor nodes not the base station. Almost all of these works either use special hardware like directional antennas, or are too complicated for sensor nodes which have very limited resources [12, 21].

In this paper we propose a novel approach for location verification that is less complicated and doesn't use any special hardware. This approach is centralized that makes it more precise than decentralized approaches [22]; more specifically, the verification process is performed in the base station which is more powerful than regular sensors that makes ability to perform more complicated tests to verify claimed locations. Also it is resilient against common attacks like Wormhole attack. We claim that this approach, as simulation results shows, is more practical, effective and has less overhead in many applications.

The rest of this paper is organized as follows. In the next section we describe the network model and the assumptions that we made. Section 3 explains some preliminary concepts. Our proposed approach is presented in section 4 and its resilience against usual attacks is discussed in section 5. Section 6 presents simulation results and finally we conclude in section 7.

2 Assumptions and Network Model

In this section we describe our network model and assumptions that we made. The WSN architecture is shown in Fig. 1. WSN consists of one or more base stations, some locators and many wireless sensor nodes. Also there might be some malicious nodes that collude together and make wormholes in network; so each of them is called a *wormhole end*. The base station is a powerful node that almost has no limitations in power or computing capabilities. It is the core of the network and manages all the activities in the network. We assume that the base station is trusted and attackers cannot compromise it. Locator is a node that somehow (by using a GPS for example) is aware of its exact location and broadcasts its location information so that sensor nodes use this information and estimate their locations. Wireless sensor is a cheap resource constrained node that monitors the environment, captures events and sends data to the base station; for the sake of simplicity, in the rest of this paper we call them sensors. Finally wormhole end is an adversary node that receives transmitted packets in the network and sends them to the other wormhole end which broadcasts these packets in the network. In fact these nodes collude together to move traffic from one place to another.

In our model none of the nodes uses any special hardware like directional antennas, ultrasound transceivers or nanosecond precision timers. Also we assume that each sensor has a private symmetric key with the base station. This key can be assigned before or after network deployment [23-26], however it seems that assigning keys before deployment is more convenient. Note that this assumption causes serious problems for in network aggregation. But in many applications in which the location of the sensors is important such as environment monitoring, data messages that would send by sensors is pretty small, so aggregation is not critical in these applications. Moreover this property allows the base station to verify the correctness of the sensors' alarms that leads to fewer false alarms.

Finally we do not consider physical layer attacks like jamming. Some approaches like [27, 28] can be used to defend against these kinds of attacks.

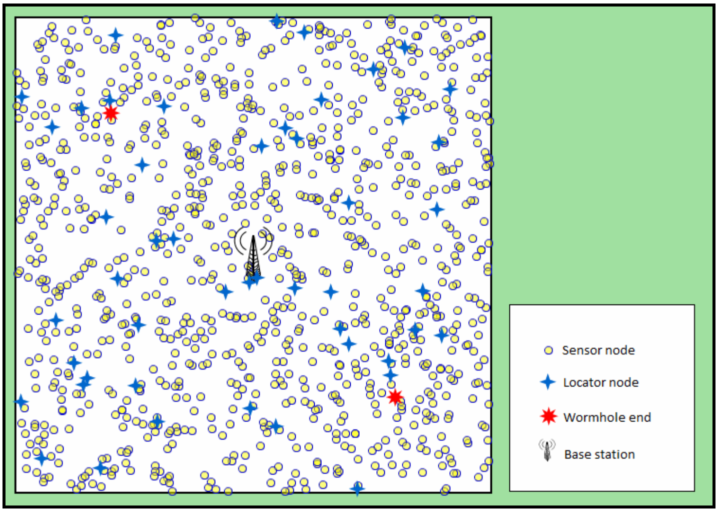

Fig. 1. Considered WSN architecture

3 Preliminaries

As we mentioned before we propose a centralized approach for location verification in WSNs. In this section we discuss why centralized approaches can be effective in this application. After that we describe a tree structure that we use in our approach.

3.1 Centralized vs. Decentralized Approaches

As mentioned before, previous works on secure localization and location verification were mostly designed in decentralized approach. In other words they insist that each sensor should estimate its location on its own securely. But these works usually lead to approaches that are too complicated and expensive for sensor nodes which have very limited resources. In some other works, assumptions are made that are not feasible for sensor nodes. For example, using unusual and expensive hardware like directional antennas or using asymmetric cryptography is not feasible for sensor nodes, since one of the most important properties of sensor nodes is that they are made with low cost.

In general, centralized approaches are simpler and more precise than decentralized approaches; but they are single point of failure, less scalable, and usually have more communication overhead [22]. We claim that disadvantages of centralized approaches are not significant in WSNs and more specifically in localization and location verification, because:

- We assume that the base station is trusted and secure. So if the location verification process is performed in the base station it would be secure too. On the other hand if the base station fails, the whole network would be useless and the location verification would be meaningless.
- WSNs have limitations in expansion since as the WSN expands, the average distance between sensors and the base station increases. This results in more communication cost in the network that leads to more power consumption and less network lifetime.
- WSNs are mostly static, i.e. sensor nodes are fixed and their location would not change after the deployment of the network. So localization and location verification process shall be performed only once and communication overhead would be negligible.

3.2 Tree-Based Message Structure

In our approach sensors send their location information in the tree structure, so we use the following structure that imposes small computation overhead on sensors. In this structure each sensor just has to concatenate received messages and add some data to the beginning and the end of it, so there is no need to decrypt and modify received messages. The structure is shown in Fig. 2-a. This structure is used recursively, i.e. response message in each sensor is a tree which is a subtree of its parent's tree. Fig. 2-b shows an example.

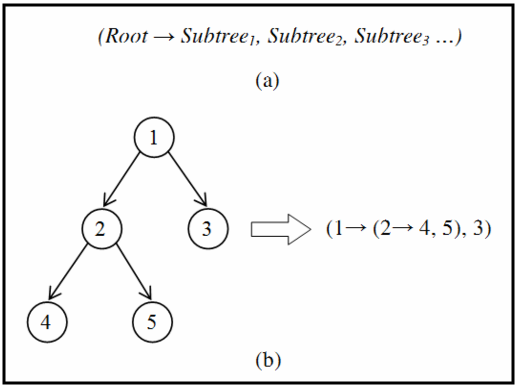

Fig. 2. a) Tree-based message structure b) An example tree

4 Proposed Approach

In this section we describe a centralized approach for location verification in WSNs which is not using any additional hardware and has small memory, computation and communication overhead. This approach is robust enough against different attacks (internal or external) and is able to detect anomalies in the localization (intended or not) quite well.

This approach consists of two phases: *initial phase* and *operation phase*. In the first phase, which is performed just after the deployment of the network, the base station collects location information of sensors in a tree-like structure and then verifies them;

of course some of the sensors may not be verified. The verified ones are called *verified sensors* while those not verified are named *unverified sensors*. The latter ones have a second chance to be verified in the operation phase. In the following sections we describe these phases in more detail.

4.1 Initial Phase

This phase has three steps: *location collection*, *verification*, and *announcement*. At first in the location collection step, the base station collects the location information of sensors, then in the verification step it verifies sensors' claimed locations according to the information that is gathered in the first step. Finally in the announcement step, the base station announces the unverified sensors. Now we describe these steps in detail.

Location Collection. Location information of sensors is collected as follows:

1. At first the base station broadcasts a location request message (LOC_REQ).
2. Each sensor after receiving the first LOC_REQ message sends an acknowledgement message (REQ_ACK) to its sender and sets it as its parent. Then it broadcasts LOC_REQ message for its neighbors. Every LOQ_REQ message that a sensor receives after the first one would be discarded.
3. Each sensor collects its children's location information, concatenates them and appends its own location information that is encrypted with its private key. This procedure makes the location response message (LOC_RSP) that the sensor sends to its parent. The LOC_RSP consists of sensor's ID, its location, response time and checksum. Response time is the period after receiving the first LOC_REQ till sending the LOC_RSP message. Checksum is used for integrity of the message; each sensor calculates checksum over the whole LOC_RSP message.
4. The process in step 3 continues until the location information of sensors arrives in the base station.

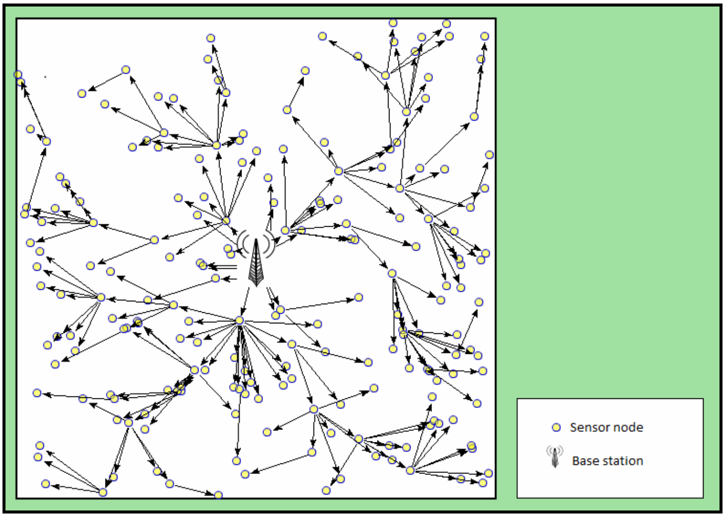

Fig. 3. The network tree after location collection step

When this step is completed, the base station would have an approximate topology of the network in the form of a tree as shown in Fig. 3.

The C++ like pseudocode of what runs in each sensor is shown in Fig. 4. It shows more details about how exactly our protocol works. In the first line sensor performs the localization process using any localization algorithm (e.g. centroid algorithm [4]). In the next three lines it waits until a LOC_REQ message is received, then it saves its parent's address and its depth in the network tree. For enabling sensors to determine their depth, we include the depth parameter in LOC_REQ message. This parameter at first is zero at the base station and each sensor increments it before broadcasting LOC_REQ. This task is performed in the fifth line. In the sixth and seventh lines sensor waits a moment and after that broadcasts the LOC_REQ message for its neighbors. The reason of this is that after broadcasting the LOC_REQ message, all the recipients receive it almost at the same time. So if they broadcast the LOC_REQ as soon as they receive it, the probability of collision would be high. Although it is MAC layer's responsibility to resolve this problem, but it can cause more power consumption. To avoid this, after receiving the LOC_REQ message, each sensor waits a random period and then broadcasts the request. We refer to maximum waiting time before sending the request as REQ_MAX_WT.

Now it's time for collecting response messages from sensor's children. But how a sensor can determine that it is a leaf in the tree? When should it send its response message to its parent? One way is according to the acknowledgements that each sensor receives. If a sensor receives no acknowledgement after broadcasting the request message, it can determine that has no children. Also if it receives acknowledgements, it can determine how many children it has by counting them and after receiving responses from all of them sends its response message to its parent. This method works properly if all the messages are received correctly and there is no malicious entity in the network. Otherwise some sensors may not be able to contribute (e.g. in the case that its acknowledgement message is not received by its parent) or maybe waits forever (e.g. in the case that the response message of one of its children doesn't arrive).

The main vulnerability point of the previous method is that each sensor relies on the other ones that may not be trusted, so sensors should decide independently. For this reason we define a time period called *response time* that each sensor waits after receiving the request to collect responses of its children and when it expires sends its response to its parent. It is obvious that response time cannot be equal for all the sensors, otherwise the network tree would have only one level. Because the sensors that are in the second level receive the request later than the first level sensors and so their timer would expire after first level sensors when they already sent their responses.

Therefore the response time for each sensor should depend on its depth in the network tree. The response time is calculated by the following formula:

$$RSP_{WT} = A \times (B - C \times depth) . \qquad (1)$$

Where A, B, and C are parameters that should be adjusted according to REQ_MAX_WT and network conditions like the height of the network tree.

```
Sensor_Activity() {
    localize ();
    receive (LOC_REQ);
    parent = LOC_REQ.sender;
    depth = LOC_REQ.depth;
    LOC_REQ.depth++;
    wait (random (REQ_MAX_WT));
    broadcast (LOC_REQ);
    collectionWT = calculateCWT (depth);
    startTimer (collectionWT);
    while (timer is not off){
       receive (LOC_RSP);
       myLOC_RSP.add (LOC_RSP);
    }
    myLOC_RSP.finalize ();
    send (parent, myLOC_RSP);
}
```

Fig. 4. Pseudocode of sensors' activity

Verification of Location Information. Existence of malicious nodes in the network can completely ruin the network and falsify the location information of the sensors. For example, Fig. 5 shows the same network of Fig. 3 and its corresponding tree in the case that it is attacked by a Wormhole.

Anomalies in Fig. 5 are obvious, but how can the base station detect them? We describe three properties that should be satisfied if the claimed location is correct and so the base station can use them to verify sensors' locations and detect anomalies in the network. These properties are as follows:

- *Communication range*: neighboring sensors should be in each other's communication range; more specifically the distance between each sensor and its parent should not be more than maximum communication range of sensors.
- *Response time*: the response time of each sensor should be greater than the response time of its children.
- *Uniqueness*: since each sensor sends its location information once and has one parent, so it should appear in the network tree once too.

If any of these properties is not satisfied for a sensor, the base station detects some anomalies about that sensor and marks it as an unverified node.

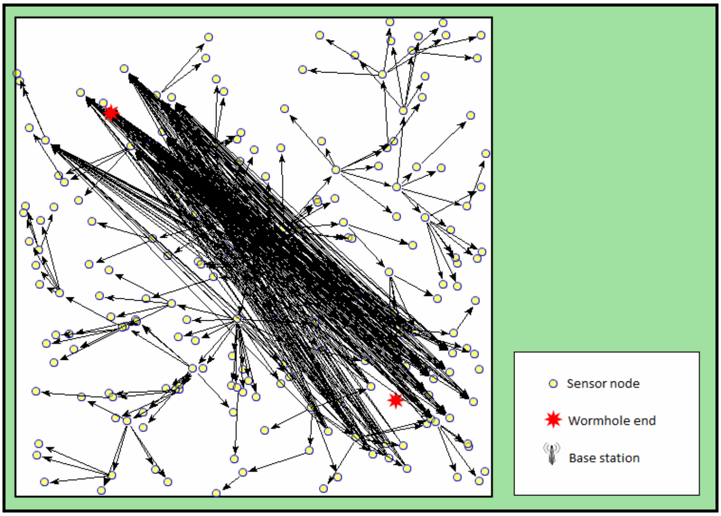

Fig. 5. The network tree in the case of Wormhole attack

After verification, the base station records trusted sensors and their locations, so these sensors do not have to send their location information with their sent data anymore and the base station can retrieve their location by their ID.

Announcing Unverified Sensors. After detecting unverified sensors, the base station announces them to the network. This announcement has two benefits, first, unverified sensors find out that their location was not verified; so they can refine their location and try again in the operation phase. The base station can provide some hints to help these sensors refining their location or they can request their verified neighbors for some hints. Second, other sensors in the network would know unverified sensors and discard messages which they send. This can save the power of sensors since unverified messages would not be forwarded by the other sensors. Note that if an unverified message is forwarded towards the base station, since its sender is not in the verified sensors list, it would be discarded there. Also note that this announcement should be performed in a secure way, otherwise attackers can use this feature and announce some verified sensors as unverified that can cause denial of service attack. For this reason the base station should send an encrypted message consisting of a list of unverified neighbors to each sensor that has at least one unverified neighbor.

4.2 Operation Phase

Sensors that have not been verified or added to the network after the initial phase have the opportunity to join the network in this phase. For this reason these sensors broadcast their location information, any receiving sensor that we call it *introducer* sends this information to the base station. Then the base station would be able to perform verification process according to location information of the requesting sensor and verified introducers. For example, suppose that the request of the requesting sensor is received by the base station through two different verified introducers. If the distance between these introducers exceeds twice of the maximum communication range of sensors, the base station detects some anomalies and the requester sensor will not verified.

5 Resilience against Security Attacks

In this section we describe why our proposed approach is resilient against different attacks.

5.1 Key Capture Attack

Since in this approach each sensor has a private key with the base station and there is not any global key, if an attacker can compromise one or more sensor nodes and captures their credentials, she cannot dig into the whole network and have a significant effect on it.

5.2 Spoofing or Altering Messages

In our approach, since there is no global key in the network and each sensor encrypts its data with its private key, spoofing and altering messages is not possible.

5.3 Sybil Attack

In this attack one node presents multiple identities to other sensors [29]. Since there is only one shared key between each sensor and the base station, a malicious node cannot present different IDs and if it uses the same ID for all the identities it claims, it would violate the uniqueness property and would be detected by the base station in the verification phase.

5.4 Sinkhole Attack

In this attack the adversary tries to somehow attract traffic from particular area or the entire network to a compromised node so that it can perform other attacks like spoofing and altering messages, selective forwarding or Blackhole attack [11].

To countervail this attack, each sensor can just send its data to its parent, or the base station can define routes and introduce some sensors to each sensor that it sends its data via them. Since the base station has a better view of the entire network, it can define routes more securely and more efficiently than decentralized routing algorithms.

5.5 Wormhole Attack

For this attack at least two malicious nodes collude, one of them receives messages and somehow (by powerful transmitters for example) sends it to another to broadcast them [11]. Note that the distance between two ends of wormhole (wormhole length) should be longer than sensors' communication range; otherwise this attack would not be effective anymore. Wormhole could be simplex (one-way) or duplex (two-way). In our approach since encryption is performed in application layer not in MAC or network layer, we assume that wormhole ends have the ability to change source and destination address of messages.

As it will be discussed here the effect of wormhole could be different according to the type of the messages they tunnel.

- LOC_INFO: Tunneling locators' message (LOC_INFO) by the wormhole cause to receiving them by sensors that are not in their communication range. So their estimation would be erroneous. If this error is significant then the base station would detect it in the verification process, because the distance between this sensor and its parent would be further than communication range of sensors and violates communication range property.
- LOC_REQ: In the case that a location request is tunneled by the wormhole, if sensor nodes that receive the tunneled message have received any location request before, then they would ignore it and so it doesn't affect them. Otherwise they will consider sender node as their parent and send their location information to it. Note that this would be possible if the wormhole works 2-way. In this case the base station would detect this anomaly according to distance consistency in the verification process.
- LOC_RSP: In this case, any node that receives a tunneled message would consider the sender as its child while in fact this node is a child of another node in another part of the network. So the uniqueness property would be violated and the base station would detect this anomaly in the verification process. If the attacker could somehow (by jamming for example) prevent the real parent to receive the response message, then the uniqueness property would hold but the distance consistency would be violated and the base station would detect this anomaly.
- DATA: Since data is sent in encrypted form and location information of sensors have been stored in the base station, tunneling data messages does not affect the network and the base station will be able to identify the sender of the message and its location.

6 Simulation and Evaluation

We have simulated the proposed approach using OMNeT++ [30] and MiXiM framework [31], the simulation parameters are shown in Table 1. As it is mentioned in this table centroid algorithm is used for localization but other methods can also be used for this purpose. Due to the localization error, maximum valid distance is considered to be greater than the communication range of sensors. The results set here are the average of 1000 runs with different configurations.

Table 1. Simulation parameters

Parameter	Value
Area	500×500 m^2
No. Sensors	1000
No. Locators	100
Communication range	50 m
MAX Valid Distance	70 m
REQ_MAX_WT	100 sec
A	50
B	200
C	20
Localization method	Centroid algorithm

As it was justified earlier, our approach can detect usual attacks properly. This section shows the approach's performance according to simulation results. We use two metrics: *location accuracy* and *connectivity factor*. Location accuracy which is the Euclidian distance between the estimated location and the exact location of the sensor is used to present localization error. Also connectivity factor indicates the percentage of sensors that contribute and those sensors which are verified in initialization phase. Table 2 shows the values of these parameters in two cases: in normal situation and under Wormhole attack. Note that the connectivity factor is reduced after the consistency check even in normal situation. This is due to the localization error of centroid algorithm not as a result of any kind of attack; in [32] these errors have been discussed in detail. As this table shows the location accuracy in the case of Wormhole attack is almost equal to normal situation.

Table 2. Resulting performance factors

Wormhole Attack	Connectivity Factor		Location Accuracy	
	Before CC	After CC	Mean	Max
No	0.916	0.892	11.552	29.602
Yes	0.912	0.732	11.450	29.861

Table 3. Resulting overhead factors

Level		**Branching Factor**		**Payload Size (bytes)**	
Mean	Max	Mean	Max	Mean	Max
6.138	10	0.966	23.3	37.587	2473.8

In order to show the overheads of our approach, we use the payload size that each sensor has to transmit, since the computation overhead is pretty small. The payload size of each sensor is dependent on its level in the network tree and the number of its children. As is shown in Table 3, the average payload size is pretty small and the maximum payload size is feasible since this transmission is performed only once in the network's lifetime.

7 Conclusion

In this paper we proposed a novel approach for secure location verification in WSNs. This approach is light weight and does not use any additional hardware like directional antennas that makes it suitable for WSNs. It is resilient against different attacks like Wormhole and Sybil attacks and detects anomalies in the localization properly. The computational and memory overheads on the sensors are pretty small. Also simulation results show that the average communication overhead on each sensor is minor and even the worst case is feasible in WSNs.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A Survey on Sensor Networks. IEEE Communications Magazine 40, 102–105 (2002)
2. Niculescu, D., Nath, B.: Ad Hoc Positioning System (Aps). In: IEEE GLOBECOM, pp. 2926–2931 (2001)
3. Want, R., Hopper, A., Falcao, V., Gibbons, J.: Active Badge Location System. ACM Transactions on Information Systems 10, 91–102 (1992)
4. Bulusu, N., Heidemann, J., Estrin, D.: Gps-Less Low-Cost Outdoor Localization for Very Small Devices. IEEE Personal Communications 7, 28–34 (2000)
5. Niculescu, D., Nath, B.: Dv Based Positioning in Ad Hoc Networks. Telecommunication Systems 22, 267–280 (2003)
6. Harter, A., Hopper, A., Steggles, P., Ward, A., Webster, P.: The Anatomy of a Context-Aware Application. Wireless Networks 8, 187–197 (2002)
7. He, T., Huang, C., Blum, B.M., Stankovic, J.A., Abdelzaher, T.: Range-Free Localization Schemes for Large Scale Sensor Networks. In: ACM MOBICOM, pp. 81–95 (2003)
8. Bahl, P., Padmanabhan, V.N.: RADAR: An in-Building RF-Based User Location and Tracking System. In: IEEE INFOCOM, pp. 775–784 (2000)
9. Fang, L., Du, W., Ning, P.: A Beacon-Less Location Discovery Scheme for Wireless Sensor Networks. In: IEEE INFOCOM, pp. 161–171 (2005)
10. Niculescu, D., Nath, B.: Ad Hoc Positioning System (Aps) Using Aoa. In: IEEE INFOCOM, pp. 1734–1743 (2003)
11. Karlof, C., Wagner, D.: Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. In: Ad Hoc Networks, vol. 1, pp. 293–315 (2003)
12. Lazos, L., Poovendran, R.: SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks. In: ACM Workshop on Wireless Security (WiSe), pp. 21–30 (2004)
13. Li, Z., Trappe, W., Zhang, Y., Nath, B.: Robust Statistical Methods for Securing Wireless Localization in Sensor Networks. In: ACM/IEEE IPSN, pp. 91–98 (2005)
14. Liu, D., Ning, P., Du, W.K.: Attack-Resistant Location Estimation in Sensor Networks. In: ACM/IEEE IPSN, pp. 99–106 (2005)
15. Čapkun, S., Hubaux, J.P.: Secure Positioning of Wireless Devices with Application to Sensor Networks. In: IEEE INFOCOM, pp. 1917-1928 (2005)
16. Vora, A., Nesterenko, M.: Secure Location Verification Using Radio Broadcast. IEEE Transactions on Dependable and Secure Computing 3, 377–385 (2006)
17. Sastry, N., Shankar, U., Wagner, D.: Secure Verification of Location Claims. In: ACM WiSe, pp. 1–10 (2003)
18. Ekici, E., McNair, J., Al-Abri, D.: A Probabilistic Approach to Location Verification in Wireless Sensor Networks. In: IEEE ICC, pp. 3485–3490 (2006)
19. Lazos, L., Poovendran, R., Čapkun, S.: ROPE: Robust Position Estimation in Wireless Sensor Networks. In: IEEE IPSN, pp. 324–331 (2005)
20. Al-Abri, D., McNair, J., Ekici, E.: Location Verification Using Communication Range Variation for Wireless Sensor Networks. In: IEEE MILCOM (2007)
21. Hu, L., Evans, D.: Using Directional Antennas to Prevent Wormhole Attacks. In: 11th Network and Distributed System Security Symposium (NDSS), pp. 131–141 (2004)
22. King, J.L.: Centralized Versus Decentralized Computing: Organizational Considerations and Management Options. ACM Computing Surveys 15, 319–349 (1983)
23. Lee, J., Stinson, D.: Deterministic Key Predistribution Schemes for Distributed Sensor Networks. In: ACM Symposium on Applied Computing, pp. 294–307 (2005)

24. Liu, D., Ning, P.: Location-Based Pairwise Key Establishments for Static Sensor Networks. In: ACM Workshop on Security of Ad Hoc and Sensor Networks, pp. 72–82 (2003)
25. Blom, R.: An Optimal Class of Symmetric Key Generation Systems. In: Pichler, F. (ed.) EUROCRYPT 1985. LNCS, vol. 219, pp. 335–338. Springer, Heidelberg (1986)
26. Chan, H., Perrig, A., Song, D.: Random Key Predistribution Schemes for Sensor Networks. In: IEEE Symposium on Security and Privacy, pp. 197–213 (2003)
27. Pickholtz, R.L., Schilling, D.L., Milstein, L.B.: Theory of Spread-Spectrum Communications - a Tutorial. IEEE Transactions on Communications 30, 855–884 (1982)
28. Wicker, S.B., Bartz, M.J.: Type-Ii Hybrid-Arq Protocols Using Punctured Mds Codes. IEEE Transactions on Communications 42, 1431–1440 (1994)
29. Newsome, J., Shi, E., Song, D., Perrig, A.: The Sybil Attack in Sensor Networks: Analysis & Defenses. In: IEEE IPSN, pp. 259–268 (2004)
30. OMNeT++ Community Site, http://www.omnetpp.org
31. MiXiM Project, http://mixim.sourceforge.net
32. Al-Abri, D., McNair, J.: On the Interaction between Localization and Location Verification for Wireless Sensor Networks. Computer Networks 52, 2713–2727 (2008)