ARTICLE IN PRESS

Ad Hoc Networks xxx (2012) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Practical and secure localization and key distribution for wireless sensor networks ☆

Qi Mi [a], John A. Stankovic [a], Radu Stoleru [b,*]

[a] Department of Computer Science, University of Virginia, United States
[b] Department of Computer Science and Engineering, Texas A&M University, United States

ARTICLE INFO

Article history:
Received 1 August 2011
Received in revised form 14 November 2011
Accepted 17 December 2011
Available online xxxx

Keywords:
Wireless sensor network
Secure localization
Key distribution

ABSTRACT

In many applications of wireless sensor networks, sensor nodes are manually deployed in hostile environments where an attacker can disrupt the localization service and tamper with legitimate in-network communication. In this article, we introduce Secure Walking GPS, a practical and cost effective secure localization and key distribution solution for real, manual deployments of WSNs. Using the location information provided by the GPS and inertial guidance modules on a special master node, Secure Walking GPS achieves accurate node localization and location-based key distribution at the same time. We evaluate our localization solution in real deployments of MicaZ. Our experiments show that 100% of the deployed nodes localize (i.e., have a location position) and that the average localization errors are within 1–2 m, due mainly to the limitations of the existing commercial GPS devices. Our further analysis and simulation results indicate that the Secure Walking GPS scheme makes a deployed WSN resistant to the Dolev-Yao, the wormhole, and the GPS-denial attacks, the scheme is practical for large-scale deployments with resource-constrained sensor nodes and has good localization and key distribution performance.

1. Introduction

Wireless sensor networks (WSNs) are envisioned to be widely used in medical, military, and environmental monitoring applications. A future WSN might consist of hundreds to thousands of deployed sensor nodes which are expected to self-organize into an autonomous network, perform desired sensing tasks, and react properly to the environment or specific events.

Localization is one of the most important services provided by a WSN, because in most applications we are interested not only in the types of events that have taken place, but also in where the events have taken place. For example, sensor nodes can be deployed along the border of a restricted area to detect intruding targets [2] or they can be scattered in a thicket to monitor sunlight and carbon dioxide concentration at different locations [3]. In addition, the normal operation of many other WSN services depends on the correct knowledge of node locations. For example, the geographic forwarding [4,5] protocol makes routing decisions based on the locations of individual sensor nodes. Hence, the locations of the deployed sensor nodes need to be determined accurately.

In many cases, a WSN is manually deployed in a potentially hostile environment and left unattended for a long period of time. As a result, it is vulnerable to various attacks during and after its deployment. An attacker usually launches a malicious attack for three purposes: (1) to steal sensitive data from legitimate messages, (2) to inject false messages into the network, and (3) to disrupt the normal operation of WSN services and applications. Therefore, to ensure that a WSN operates as expected, it is crucial that

☆ A preliminary version of this article was presented at the ACM Conference on Wireless Network Security (WiSec), 2010 [1].

* Corresponding author. Address: Department of Computer Science and Engineering, Texas A&M University, MS 3112, College Station, TX 77843, USA. Tel.: +1 979 862 8349; fax: +1 979 847 8578.

E-mail addresses: qimi@cs.virginia.edu (Q. Mi), stankovic@cs.virginia.edu (J.A. Stankovic), stoleru@cse.tamu.edu (R. Stoleru).

1570-8705/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2011.12.008

WSN designers consider potential attacks and include countermeasures in their designs. In this work, we focus on three typical types of attacks: the Dolev-Yao, the wormhole, and the GPS-denial attacks, and present an integral solution to secure localization and key distribution in manual deployments of large-scale WSNs.

The major contributions of this work are: (1) a practical localization protocol which is secure against the three aforementioned attacks; (2) an integrated localization and key distribution protocol that keeps key sets on deployed nodes very small; thereby meeting memory constraints, and ensures network communication connectivity and protection against wormhole attacks; (3) a security analysis demonstrating the correctness of our solution; and (4) a performance evaluation using parameters from a real WSN deployment, which demonstrates: a high localization accuracy, that almost all nodes are localized, the excellent scaling properties to networks of at least size 1000, the excellent performance even in the presence of realistic irregular communication ranges, and low overhead.

The rest of the article is organized as follows. We present our Secure Walking GPS solution in Section 2 and its security analysis in Section 3. We present the evaluation of our secure localization and key distribution in Section 4. In Section 5 we present the related work and discuss their limitations and conclude our work in Section 6.

2. Secure localization system design

An alternative to the Secure Walking GPS localization scheme is enabling each sensor node with GPS capabilities. This monolithic solution is both expensive and inefficient. In the Secure Walking GPS architecture, however, the system is decoupled into two main components: the master node and the sensor node, as depicted in Fig. 1.

In our solution the master node is present during the deployment of nodes. The master node obtains its current location from an onboard GPS device, and sends it to each newly deployed sensor node wirelessly. An inertial guidance (IG) module complements the function of GPS on the master node. The IG module uses motion and rotation sensors to continuously capture the orientation and velocity of the deployer, and estimates the master node's location (still represented using GPS coordinates) via dead reckoning [6]. Since the IG module does not depend on external resources, it is always available and it serves as a backup source of current location during a GPS-denial attack. Communication keys, for neighborhood communication, are also distributed efficiently to sensor nodes during the node localization process.

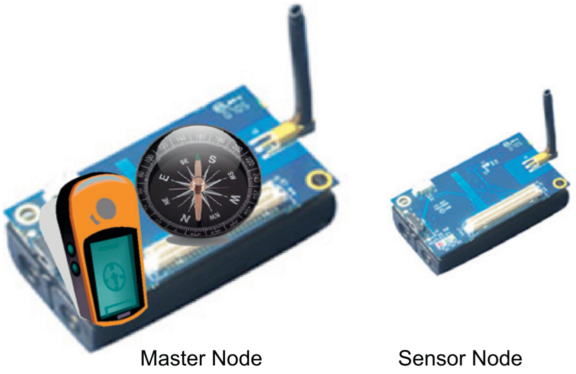

Fig. 1. Decoupling of the Secure Walking GPS localization system into two components: the master node (enabled with a GPS and inertial modules) and the sensor node.

This architecture enabled us to push all complexity derived from the interaction with the GPS device to a single node, the master node, and to significantly reduce the size of the code and data memory used on the sensor node. Through this decoupling, a single master node is sufficient for the localization of an entire sensor network, and the costs are thus reduced.

2.1. Local coordinate system

A GPS location is represented by a latitude λ and a longitude ϕ, which are angular measures from the Equator to North or South, and Prime Meridian to East or West, respectively. A relatively simple design for the master node would have been to use a GPS coordinate system, since actual GPS and IG locations are represented using GPS coordinates. Due to the relatively small size of a sensor network (hundreds to a few thousand meters), the use of global (i.e. GPS) coordinates is very inefficient. The inefficiency stems from the size of the packets used for passing location information – a significant portion of the location is likely to be the same for all sensor nodes – as well as from the computational costs encountered when aggregating data, e.g., triangulation of several GPS coordinates for positioning a target. In order to reduce the aforementioned overhead we use a local, Cartesian, coordinate system. This local coordinate system of reference, which uses linear units, is better suited for WSN, than a global coordinate system.

A local coordinate system is built from a global system, that uses GPS coordinates, in the following way: the local system of reference has an origin (called a Reference Point) specified in terms of global coordinates (GPS coordinates). The distance between this Reference Point (with coordinates λ_1 and ϕ_1) and another point, with a GPS location specified by λ_2 and ϕ_2, can be computed as follows [7]:

$$Distance = \sqrt{(F_{lat}(\phi_1 - \phi_2))^2 + (F_{lon}(\lambda_1 - \lambda_2))^2} \tag{1}$$

where

$$F_{lat} = \frac{\pi}{180}\left(\frac{a^2b^2}{(a^2\cos^2\phi + b^2\sin^2\phi)^{3/2}} + h\right) \tag{2}$$

$$F_{lon} = \frac{\pi}{180}\left(\frac{a^2}{(a^2\cos^2\phi + b^2\sin^2\phi)^{1/2}} + h\right)\cos\phi \tag{3}$$

are conversion factors that represent the distances for 1° change in latitude and longitude, respectively. The unit of measure is meter/degree. The parameters in the above formulas are: a = 6,378,137 m, b = 6,356,752.3142 m and h is the height over the earth ellipsoid. The influence of h on the conversion factors is minimal and a value of 200 m is assumed. The X and Y coordinates of the point with a GPS location specified by λ_2 and ϕ_2 are given by the two additive terms in Eq. (1). The *Y*-axis of the local coordinate system is oriented in the North/South direction and the *X*-axis in the East/West direction. All variables specified in Eqs. (1)–(3) (i.e., λ, ϕ and h) can be directly obtained from a commercial GPS device. The result of our design is that the master node transforms the global coordinates received from the GPS device into local coordinates and broadcasts these local coordinates.

2.2. Attack model and assumptions

2.2.1. Attack model

The goal of an attacker is to mislead sensor nodes into obtaining false locations and also threaten location-dependent services such as tracking.

We explore three types of WSN attacks which are typical and the most threatening to localization, namely the Dolev-Yao attack, the wormhole attack and the GPS-denial attack. The Dolev-Yao and wormhole attacks are the two main security attacks to which wireless sensor networks are very vulnerable [8]. In a Dolev-Yao attack, an attacker can overhear, intercept, and synthesize any message and is only limited by the constraints of the cryptographic methods used [9]. A Dolev-Yao attack compromises the authenticity, legitimacy and confidentiality of messages. In a wormhole attack, an attacker creates a link between two distant locations, tunnels legitimate messages from one end of the link to the other end, and replays them there. A wormhole attacker attempts to make sensor nodes appear closer than they really are, violating the communication range constraint. It is difficult to detect a wormhole attack because the "victim" messages are still legitimate and kept intact. In a GPS-denial attack, an attacker intermittently jams the GPS signals. GPS signals are typically used by WSN anchor nodes (i.e., nodes that know their locations) to obtain their locations.

There are also other WSN attacks such as the physical tampering of sensor nodes and the denial-of-service (DoS) attacks, but they are outside our scope.

2.2.2. Assumptions

We assume that there is an attack-free base station located behind the deployment field, where it is secure to perform any necessary pre-deployment operation, such as downloading program code and distributing an initial key to each sensor node. However, the actual deployment takes place in a two-dimensional infrastructure-less field consisting of open spaces and heavy woods. We assume that the GPS signals are not always available during deployment, either because of temporary lack of Line-of-Sight GPS signals due to the surrounding environment, or because of purposeful GPS-denial attacks. As a result, not all sensor nodes can be localized using the GPS module alone. We also assume that sensor nodes are close to the master node when they are deployed. Therefore, it is reasonable for the master to make all the localization and key distribution decisions and securely inform the sensor node of its decisions.

We assume that the master node is a powerful node and it will not be compromised by any attack. We assume that the inertial guidance (IG) module is always available and provides trustworthy readings. We also assume that when GPS signals are available, they are trustworthy. These assumptions are reasonable, because an IG module relies on its own motion and rotation sensors to infer its location, and a military GPS device usually has anti-spoofing capabilities.

Table 1
Cryptographic notations.

Notation	Meaning
M	The master node
s_i	The *i*th deployed sensor node
$A \rightarrow B$:*msg*	*A* sends the *msg* to *B*
$msg_1 \| msg_2$	The concatenation of msg_1 and msg_2
msg	*msg* in plain text
$\{msg\}_k$	The encryption of *msg* with *k*
k_i^D	The deployment key distributed to s_i
K_i^C	The set of *m* communication keys, ($k_{i,l}^C$ where $l = \overline{1,m}$) distributed to s_i
NID(M)	The node id of *M*
NID(s_i)	The node id of s_i
KID(k)	The key id of *k*

2.3. Pre-deployment

Secure Walking GPS begins with a pre-deployment phase, which takes place in the secure base station. The main goal of pre-deployment is to distribute a unique deployment key to every sensor node in order to bootstrap the secure communication between the master node and each of the sensor nodes during the deployment.

Cryptographic notations describing our security scheme are listed in Table 1.

It is best practice to keep the master node turned on during the entire pre-deployment but allow only one sensor node to be turned on at any time (i.e., so that it can obtain a deployment key). This not only saves the energy of sensor nodes, but also prevents potential interference between sensor nodes. For management purposes, the master node saves all distributed deployment keys, which are indexed by their key ids, in a non-volatile memory so that they are retained even if the master node is turned off. The master node also maintains a list of <*node-id*, *deployment-key-id*> entries, mapping each distributed deployment key to a sensor node to which it has been distributed.

Because the pre-deployment is done in a secure base station, the distribution of deployment keys is done as follows:

$$s_i \rightarrow M: \ \mathrm{NID}(s_i)\|\mathrm{REQ\_PRE\_DEPLOYMENT}$$

$$M \rightarrow s_i: \ \mathrm{NID}(M)\|k_i^D$$

$$s_i \rightarrow M: \ \mathrm{NID}(s_i)\|\mathrm{ACK\_PRE\_DEPLOYMENT}$$

A sensor node s_i sends a message to the master node *M*, containing its node id and a REQ_PRE_DEPLOYMENT request (both of which are in plain text) to request its deployment key, if it has not successfully obtained one from M before. When M receives such a request, it checks whether a deployment key has already been distributed to s_i earlier, by checking the <*node-id*, *deployment-key-id*> entries. If no entry maps to s_i, M generates a new random deployment key k_i^D and sends it to s_i.[1] Meanwhile, M adds a corresponding <*node-id*, *deployment-key-id*> entry for s_i. If, on the other hand, M finds out that a deployment key has been distributed to s_i earlier, M simply resends that key to s_i. This design prevents M from generating and distributing different deployment keys to s_i when s_i is inadvertently turned off and on multiple times during pre-deployment. Once s_i obtains k_i^D, it saves it in a non-volatile memory for later use and replies to M with an acknowledgment message.

Due to the uniqueness of the deployment keys and the fact that each of them is known only by the master node and one sensor node, further messages between the master node and each sensor node can be encrypted, providing cryptographic protection for the vulnerable wireless communication during the deployment.

2.4. Deployment

2.4.1. Secure localization

After the preparation in the pre-deployment phase, the master node and the sensor nodes are taken to the deployment field. During the deployment, the master node remains turned on. Sensor nodes are in the proximity of the master node and are, in arbitrary order, turned on and deployed one after another. A sensor node s_i communicates with the master node M using the following secure protocol to obtain its location and the set of m communication keys:

$$s_i \rightarrow M: \ \mathrm{NID}(s_i) \| \{\mathrm{REQ\_DEPLOYMENT}\}_{k_i^D}$$

$$M \rightarrow s_i: \ \mathrm{NID}(M) \Big\| \{location\}_{k_i^D} \Big\| \left\{k_{i,1}^C, k_{i,2}^C, \ldots, k_{i,m}^C\right\}_{k_i^D}$$

$$s_i \rightarrow M: \mathrm{NID}(s_i) \| \{\mathrm{ACK\_DEPLOYMENT}\}_{k_i^D}$$

After initialization, s_i sends a message to M, containing its node id and a REQ_DEPLOYMENT request. Note that only the REQ_DEPLOYMENT request is encrypted using s_i's deployment key k_i^D. The source id is sent in plain text in order for the master node to index k_i^D from its own memory and decrypt this request message [10] using it. Then M replies with messages to s_i, in which M's source id is sent in plain text, but the location and the m communication keys for s_i are encrypted using k_i^D.[2] If s_i obtains the desired information, it securely acknowledges success to the master node.

Algorithm 1. Location-based key distribution

```
for all $k_j^C$ in $P$ do
  $k_j^C.state \leftarrow never\text{-}distributed$
end for
$S_1 = \phi$
deploy node $s_1$
$K_1^C \leftarrow \{m\ never\text{-}distributed$ keys from $P\}$
$M$ transmits key set $K_1^C$ to node $s_1$
$P' \leftarrow K_1^C$
for all $k_j^C$ in $P'$ do
  $k_j^C.state \leftarrow distributable$
end for
for $i$ from 2 to $n$ do
  deploy node $s_i$
  $S_i = S_{i-1} \cup \{s_{i-1}\} = \{s_1, s_2, \ldots, s_{i-1}\}$
  $K_i^C \leftarrow \mathrm{GET\_KEYS}(S_i, P, P')$
  $M$ transmits key set $K_i^C$ to node $s_i$
  $P' \leftarrow P' \cup K_i^C$
  for all $k_j^C$ in $P'$ do
    $k_j^C.state \leftarrow distributable$
  end for
end for
```

In a WSN deployment using Walking GPS, sensor nodes are physically close to the master node at the time of deployment. Therefore, it is reasonable for a sensor node to take on the master node's current location, when the node is deployed. Given the relatively high accuracy of GPS, locations provided by the GPS module are preferred. Only when the GPS module fails to work due to intermittent or temporary loss of GPS signals will the locations provided by the IG module be used as a backup. Also note that, since the error of the location estimates provided by the IG module alone is likely to accumulate if no remedial measure is taken, IG module needs to be calibrated periodically with the GPS, whenever the GPS signals are available.

Through the use of GPS and IG modules, all the sensor nodes can be localized at deployment time. No further collaboration among neighbors is needed for localization. This eliminates a potential security vulnerability that could occur if collaboration were needed.

2.4.2. Location-based key distribution

In addition to a location, a set of communication keys is distributed to each sensor node when it is deployed. The choice of communication keys that make up this key set is determined by the master node at deployment time, based on the estimated locations of the current sensor node and the sensor nodes which have been deployed earlier. Our key distribution scheme ensures that every deployed node shares at least one communication key with one or more of its neighbors, enabling them to communicate securely using the shared key(s). Note, while

[1] There are a variety of algorithms for key generation, such as a random generation based on a preloaded seed. We do not focus on the specific implementation of the key generation algorithm in this work.

[2] Depending on the maximum message length, the entire encrypted payload may be sent over multiple messages.

our scheme does not guarantee that a sensor node shares a communication key with every neighbor, it attempts to allow a sensor node to share communication keys with as many different neighbors as possible, making it better connected with its neighbors.

The algorithms for our location-based key distribution are presented in Algorithms 1 and 2. In the remaining part of this section, we describe in detail the steps of these algorithms and how we enforce the following two rules:

Algorithm 2. GET_KEYS (S_i, P, P')

```
1:  for $j$ from 1 to $i-1$ do
2:    Calculate $d_{i,j} = |s_i - s_j|$
3:  end for
4:  for $j$ from 1 to $i-1$ do
5:    if $M$ cannot communicate with $s_j$ then
6:      $d_{i,j} \leftarrow +\infty$
7:    end if
8:  end for
9:  $\{\sigma_{(l)} | l = \overline{1,i-1}\} = PERMUTATE\{j | j = \overline{1,i-1}\}$,
    where $d_{i,\sigma_{(l)}} \leqslant d_{i,\sigma_{(l+1)}}$
10: $S_i = A_i \cup B_i$, where
    $A_i = \{s_{\sigma_{(j)}} | d_{i,\sigma_{(j)}} < r \wedge M$ can communicate with $s_j\}$
    and $B_i = S_i - A_i$
11: for $l$ from$(|A_i| + 1)$to$(|A_i| + |B_i|)$ do
12:   for $n$ from 1 to m do
13:     $k^C_{\sigma_{(l)},n}.state \leftarrow non\text{-}distributable$
14:   end for
15: end for
16: $num \leftarrow 0$
17: $K^C_i \leftarrow \phi$
18: $u \leftarrow 1$
19: while $(num < m - 1) \wedge (\exists distributable$ keys in
    $P') \wedge (u < i)$ do
20:   $D_i = \{k^C_{\sigma_{(u)},v} | v = \overline{1,m} \wedge k^C_{\sigma_{(u)},v}.state = distributable\}$
21:   $\{\delta_{(w)} | w = \overline{1,|D_i|}\} = \text{PERMUTATE}\{v | v = \overline{1,|D_i|}\}$,
    where $k^C_{\sigma_{(u)},\delta_{(w)}}.freq \geqslant k^C_{\sigma_{(u)},\delta_{(w+1)}}.freq$
22:   $K^C_i \leftarrow K^C_i \cup \left\{k^C_{\sigma_{(u)},\delta_{(1)}}\right\}$
23:   $num \leftarrow num + 1$
24:   if $d_{i,\sigma_{(u)}} \geqslant r/2$ then
25:     for $w$ from 1 to $|D_i|$ do
26:       $k^C_{\sigma_{(u)},\delta_{(w)}}.state \leftarrow non\text{-}distributable$
27:     end for
28:   else
29:     $k^C_{\sigma_{(u)},\delta_{(1)}}.state \leftarrow non\text{-}distributable$
30:   end if
31:   $u \leftarrow u + 1$
32: end while
33: $K^C_i \leftarrow K^C_i \cup \{(m - num)$ never-distributed
    keys from $P\}$
34: return $K^C_i$
```

Distance Bounding Rule: Two sensor nodes are allowed to share a communication key only if they are physical neighbors.[3]

Connectivity Rule: Each sensor node needs to share a communication key with at least one of its already deployed physical neighbors so as to ensure neighbor connectivity.

In the proposed Secure Walking GPS, the master node maintains a large key pool P, from which m communication keys are carefully chosen and distributed to each sensor node (note: secure communication is possible with a sensor node by using sensor's deployment key). Each communication key in P is randomly generated, unique, and is indexed by a communication key id. Each communication key can be in one of three possible states: *never-distributed*, *distributable* and *non-distributable*. Initially, all have their states set to *never-distributed* (Algorithm 1 Lines: 1–3).

The choice of the set of communication keys for the first sensor node s_1 is trivial. The master node simply chooses m keys with a *never-distributed* state from P and transmits them to s_1 (Algorithm 1 Lines: 4–7). Then the master node sets the states of these m keys to *distributable* so that they may be shared by sensor nodes which are deployed later and become s_1's neighbors (Algorithm 1 Lines: 8–11). For each subsequent sensor node $s_i (i = \overline{2,n})$ deployed, the master node M goes through the following steps to determine which keys should be transmitted to it (Algorithm 1 Lines: 12–21).

Step 1: Find s_i's physical neighbors from the set of sensor nodes that have already been deployed (Algorithm 2 Lines: 1–10).

M first calculates $d_{i,j}$, the distances between s_i and sensor nodes $s_j (j = \overline{1,i-1})$ based on their locations reported by the GPS or IG modules. Then, M attempts to communicate with each of them securely using their respective deployment keys. If a sensor node s_j is unreachable and does not reply, M updates the corresponding distance $d_{i,j}$ to $+\infty$. M sorts these distances in ascending order and partitions the set of already deployed nodes $S_i = \{s_1, s_2, \ldots, s_{i-1}\}$ into A_i and B_i as follows:

$$A_i = \{s_{\sigma_{(j)}} | d_{i,\sigma_{(j)}} < r \wedge M \text{ can communicate with } s_j\}$$
$$B_i = S_i - A_i$$

Note that, due to the actual irregular radio patterns (which are common in WSNs), some sensor nodes in B_i may be able to communicate with M as well. However, we take a conservative approach and only consider the physical neighbors that lie within s_i's theoretical communication range r.

Step 2: Set the states of all the communication keys which have been distributed to the sensor nodes in B_i to non-distributable, in order to satisfy the Distance Bounding Rule (Algorithm 2 Lines: 11–15).

Step 3: Determine which communication keys can be distributed to s_i (Algorithm 2 Lines: 16–33).

If s_i's closest physical neighbor $s_{\sigma_{(1)}}$ has only one distributable communication key, M includes it in s_i's communi-

[3] This means that nodes far apart do not share communication keys. This is important in protecting the WSN against the wormhole attack.

cation key set K_i^C and sets its state to *non-distributable*. Otherwise, if $s_{\sigma_{(1)}}$ has more than one *distributable* communication key, M chooses the one that has been most frequently distributed to s_i's physical neighbors in A_i, includes it in K_i^C, and then sets its state to *non-distributable*. If the distance between $s_{\sigma_{(1)}}$ and s_i is greater than or equal to $r/2$, M also changes the states of $s_{\sigma_{(1)}}$'s remaining communication keys to *non-distributable*. If, however, the distance between $s_{\sigma_{(1)}}$ and s_i is less than $r/2$, M does not make this change. This ensures that s_i shares at most one communication key with each of its physical neighbors which are farther than $r/2$ away, so that s_i has a better chance to share communication keys with more physical neighbors.

After the communication keys of $s_{\sigma_{(1)}}$ have been considered, M considers those of s_i's second, third, ..., closest physical neighbors $(s_{\sigma_{(2)}}, s_{\sigma_{(3)}}, \ldots)$ until $(m-1)$ *distributable* communication keys from s_i's physical neighbors are included in K_i^C or fewer than $(m-1)$ such *distributable* communication keys are available to be included. In either case, remaining communication keys for s_i will be chosen from the *never-distributed* keys in P to make up K_i^C.

Note that M deliberately includes at least one *never-distributed* communication key in K_i^C so that s_i may share it with potential neighbors which have not been deployed.

The above design ensures that every node is able to securely communicate with at least one physical neighbor using a common communication key without violating the Distance Bounding Rule.

Step 4: Send the set of m carefully chosen communication keys to s_i, securely using s_i's deployment key (Algorithm 1 Line: 16).

Step 5: Reset the states of all *non-distributable* communication keys to *distributable* before the next sensor node is deployed (Algorithm 1 Lines: 17–20).

In our key distribution scheme, the total number of communication keys which are distributed to each node is denoted by m, whose value can be specified by the deployer in the program code. Observe that if m is too small, the Distance Bounding Rule and the Connectivity Rule may not be satisfied in arbitrary topology and deployment order of the sensor nodes. However, if m is too large, many of the communication keys may be redundant and take up much memory on resource-constrained sensor nodes. Therefore, a tradeoff exists between the size of a communication key set and the performance of the deployment.

The following theorem gives a theoretical lower bound for m. For simplicity, we assume that each node has the same circular communication range.

Theorem 1. *Let N be the maximum number of neighbors of each sensor node, and m be the required number of communication keys distributed to each sensor node. Assuming that each node has the same circular communication range, in order to satisfy the Distance Bounding Rule and the Connectivity Rule in the arbitrary topology and arbitrary order of deployment, a lower bound of m is given by:*

$$m_{\min}(N) = \begin{cases} N & \text{if } N \leqslant 5 \\ 5 & \text{if } N \geqslant 6 \end{cases}$$

Proof. Before proceeding with the proof, we provide some intuition behind the choice of intervals (i.e., $N \leqslant 5$ and $N \geqslant 6$). Assuming ideal conditions where the communication range is circular and all nodes have equal communication range r, a node s can communicate with any node that is in the circle centered at s with a radius of r. If we divide this circle into six equal sectors, then any two nodes within the same sector can communicate with each other since their distance will be smaller than r. Therefore, the lower bound can be at least as small as 6. As we will show later, the lower bound can be further reduced to 5.

Let N be the maximum number of physical neighbors of each sensor node. Assume that every sensor node has a perfect circular communication range of r.

(a) Case $N \leqslant 5$

Without loss of generality, suppose sensor node S has N physical neighbors. On the one hand, if each of the N physical neighbors uses a unique communication key to communicate with S, the Connectivity Rule is trivially satisfied. So, $m_{min}(N) \leqslant N$. On the other hand, if these N physical neighbors are mutually not physical neighbors to each other, these N nodes are not allowed to share communication keys by the Distance Bounding Rule (Consider the extreme case where the N physical neighbors are uniformly distributed on a circle with a center at S and a radius of $(r-\epsilon)$, and ϵ is infinitely small. Each pair of the physical neighbors are further than r apart.) As a result, each of the N physical neighbors has to share a different communication key with S in order to keep connected to the network. This means that S has at least N communication keys. So, $m_{min}(N) \geqslant N$. Therefore, $m_{min}(N) = N$.

(b) Case $N \geqslant 6$

Since $m_{min}(N)$ is a non-decreasing function of N.

$$m_{min}(N) \geqslant m_{min}(5) = 5$$

when $N \geqslant 6$.

Therefore, it is a necessary condition to distribute five communication keys to every sensor node in order to ensure that the Distance Bounding Rule and Connectivity Rule can be satisfied in arbitrary cases. Next, we show that it is also a sufficient condition.

Assume that the N physical neighbors of S are $A_1, A_2, \ldots, A_N$. We show that we can always group them into six mutually exclusive and exhaustive sets P_1, P_2, P_3, P_4, P_5 and Q, where there always exists a feasible key distribution scheme for these N physical neighbors with the size of their key sets being 5, which satisfies the Distance Bonding Rule and the Connectivity Rule.

Without loss of generality, choose an arbitrary physical neighbor and denote it as A_1. Draw a radial from S to A_1 and sweep this radial clockwise with its end fixed at S. The subscripts of the remaining physical neighbors are assigned in the order that this radial hits them sequentially. Define $\widehat{A_iSA_j}$ as the angle for the radial SA_i to sweep to the radial SA_j in a clockwise fashion.

P_1 is defined as follows:

$$P_1 = \begin{cases} \left\{A_1, A_2, \ldots, A_{i_1} | A_1\widehat{SA}_{i_1} \leqslant \frac{\pi}{3} \wedge A_1\widehat{SA}_{i_1+1} > \frac{\pi}{3}\right\}, \\ \qquad\qquad\qquad\qquad \text{if } A_1\widehat{SA}_N > \frac{\pi}{3} \\ \{A_1, A_2, \ldots, A_N\}, \text{if } A_1\widehat{SA}_N \leqslant \frac{\pi}{3} \end{cases}$$

If $A_1\widehat{SA}_N > \frac{\pi}{3}$, then $\exists i_1$, such that

$$A_1\widehat{SA}_{i_1} \leqslant \frac{\pi}{3} \wedge A_1\widehat{SA}_{i_1+1} > \frac{\pi}{3}$$

Since $d_{A_i,A_j} < r(1 \leqslant i,j \leqslant i_1, i{\neq}j)$, all the nodes in P_1 are allowed to share a communication key with S, say $k^C_{S,1}$.

If $A_1\widehat{SA}_N \leqslant \frac{\pi}{3}$, then P_2, P_3, P_4, P_5, and Q become empty sets. In this case, it is sufficient to distribute five communication keys to each sensor node.

When $A_1\widehat{SA}_N > \frac{\pi}{3}$, we further define P_2 in a similar way:

$$P_2 = \begin{cases} \left\{A_{i_1+1}, A_{i_1+2}, \ldots, A_{i_2} | A_{i_1}\widehat{SA}_{i_2} \leqslant \frac{\pi}{3} \wedge \right. \\ \qquad\qquad \left. A_{i_1}\widehat{SA}_{i_2+1} > \frac{\pi}{3}\right\}, \text{if } A_{i_1}\widehat{SA}_N > \frac{\pi}{3} \\ \{A_{i_1+1}, A_{i_1+2}, \ldots, A_N\}, \text{if } A_{i_1}\widehat{SA}_N \leqslant \frac{\pi}{3} \end{cases}$$

All the nodes in P_2 are allowed to share another communication key with S, say $k^C_{S,2}$.

If $A_{i_1}\widehat{SA}_N \leqslant \frac{\pi}{3}$, then P_3, P_4, P_5, and Q become empty sets. In this case, it is sufficient to distribute five communication keys to each sensor node.

When $A_{i_1}\widehat{SA}_N > \frac{\pi}{3}$, we further define P_3 in a similar way.

If we repeat this process, we can define at most five mutually exclusive (but not necessarily exhaustive) sets P_1, P_2, P_3, P_4, P_5. We are unable to define six such sets, because if we were able to, then:

$$\begin{aligned} & A_1\widehat{SA}_{i_1+1} + A_{i_1+1}\widehat{SA}_{i_2+1} + A_{i_2+1}\widehat{SA}_{i_3+1} + A_{i_3+1}\widehat{SA}_{i_4+1} \\ & \quad + A_{i_4+1}\widehat{SA}_{i_5+1} + A_{i_5+1}\widehat{SA}_{i_6+1} \\ & \quad > 6 \times \frac{\pi}{6} = 2\pi \end{aligned}$$

which is contradictory.

If i_5 is still smaller than N, we can define

$$Q = \{A_{i_5+1}, A_{i_5+2}, \ldots, A_N\}$$

Since:

$$A_1\widehat{SA}_{i_1+1} + A_{i_1+1}\widehat{SA}_{i_2+1} + A_{i_2+1}\widehat{SA}_{i_3+1} + A_{i_3+1}\widehat{SA}_{i_4+1} + A_{i_4+1}\widehat{SA}_{i_5+1} > \frac{5\pi}{3}$$

we have:

$$A_{i_5+1}\widehat{SA}_1 < 2\pi - \frac{5\pi}{3} = \frac{\pi}{3}$$

Therefore:

$$d_{A_i,A_j} < r \text{ and } d_{A_i,A_1} < r, \text{ for } i_5 + 1 \leqslant i,j \leqslant N, i \neq j$$

This means that all the nodes in Q can share any of A_1's communication keys other than $k^C_{S,1}$ in order to keep connected.

In summary, physical neighbors in P_i securely communicate with S using one of $k^C_{s_i}$ (i = 1, 2, 3, 4, 5), while the physical neighbors in Q securely communicate with A_1 using a communication key that is different from $k^C_{s_i}$. Five communication keys are sufficient for all the sensor nodes and the Distance Bounding Rule and the Connectivity Rule are both satisfied. □

Note that the simplifying assumption of circular communication range is used in the theorem only to provide the reader with a general feel for how many communication keys each sensor node should obtain and whether they fit on resource-constrained sensor nodes. According to this theorem, five communication keys suffice in the ideal case. Even in real environments where the radio pattern is irregular, we do not expect m_{min} to increase much beyond five. Our empirical evaluation results in Section 4.2.4 confirm this conclusion.

2.5. Post-deployment

After the deployment, each sensor node has obtained its location and a set of communication keys from the master node. Then each sensor begins to discover its *useful neighbors*, which are within their actual communication ranges and share at least one communication key. To do so, every sensor node broadcasts messages which are encrypted using each of its communication keys. If a sensor node can hear a message from another sensor node and decrypt the message using one of its own communication keys, these two sensor nodes are useful neighbors. So this sensor node replies to the other node with a message which is encrypted with the same communication key. After both sensor nodes discover each other as new useful neighbors, subsequent communication between them is encrypted using any of their shared communication keys.

Some attackers may monitor encrypted messages between two sensor nodes and attempt to recover the key used to encrypt these messages by studying the encryption patterns. Therefore, if two neighboring nodes share two or more communication keys, they can encrypt each message using a key that is randomly chosen from among all shared communication keys instead of encrypting every message with the same shared communication key. Doing so can further confuse the attackers' judgment and defeat their attempt to figure out a correct key. It is important to mention that no matter how sophisticated an encryption technique is, it is subject to be compromised. Randomizing communication keys helps add a second layer of security.

2.6. A key deployment example

In this subsection we briefly give an example of our proposed location-based key distribution scheme. Our example is depicted in Fig. 2 and is further described below.

Let's assume that the communication range of each sensor node is regular and that M distributes a set of five communication keys to each sensor node when it is deployed. Also assume that s_1, s_2, s_3, and s_4 (shown as solid dots with their key sets in curly braces) have already been deployed. When s_5 (shown in the hollow dot) is being deployed, the master node M determines which communication keys can constitute s_5's key set K^C_5. For

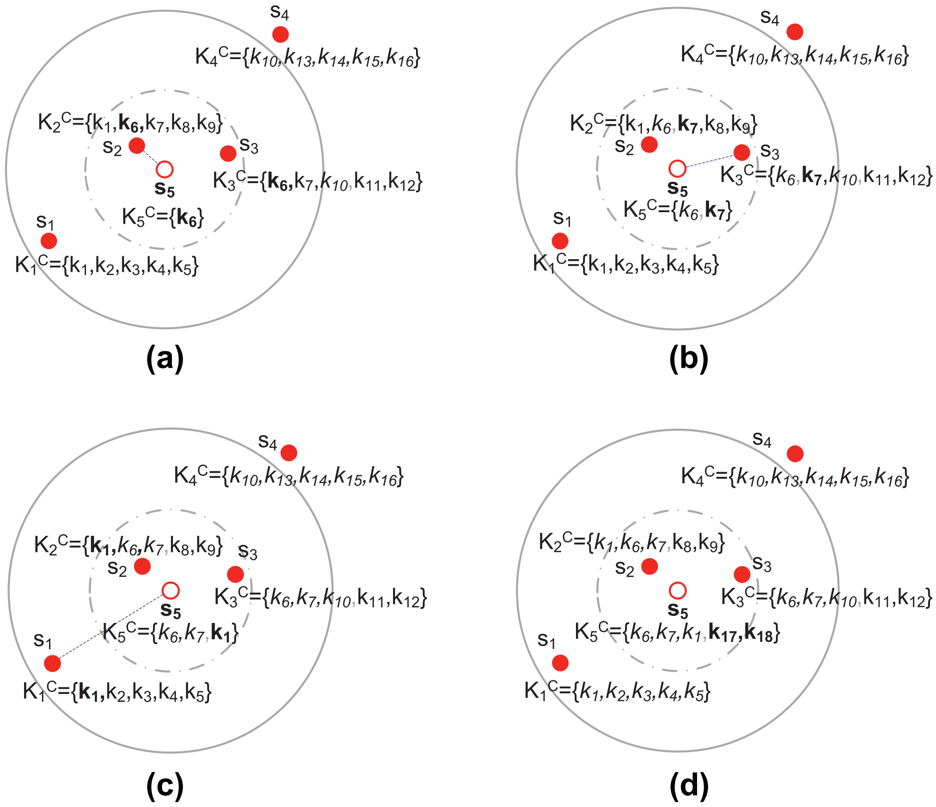

Fig. 2. Example for the location-based key distribution process (keys in "italic": *non-distributable* keys, and keys in "bold": the *distributable* key chosen to be included in K_5^C at this step.)

reference, the dashed circle is centered at s_5 with a radius of $r/2$, while the solid concentric circle has a radius of r.

Since only s_4 is outside s_5's communication range, $A_5 = \{s_1, s_2, s_3\}$ and $B_5 = \{s_4\}$. M sets each key in K_4^C to *non-distributable* to prevent potential wormhole attacks. Since s_2 is s_5's closest neighbor and k_6 in K_2^C has been the most frequently distributed to both s_2 and s_3, k_6 is included in K_5^C (as shown in Fig. 2a). Since $d_{5,2} < r/2$, M only sets k_6 to *non-distributable* and keeps the remaining keys in K_2^C as *distributable*.

As shown in Fig. 2b, M checks K_3^C, the communication key set of s_5's second closest neighbor s_3. Since k_6 and k_{10} have been set to *non-distributable*, only k_7, k_{11}, and k_{12} are available *distributable* keys. Since k_7 has been more frequently distributed than the other two, k_7 is included in K_5^C. Then, M sets k_7 to *non-distributable* before checking K_1^C, the communication key set of the third closest neighbor s_1. Among the *distributable* keys in K_1^C, k_1 has been the most frequently distributed key (to both s_1 and s_2). Therefore, k_1 is also included in K_5^C, as shown in Fig. 2c. Since $d_{5,1} \geqslant r/2$, every key in K_1^C is set to *non-distributable*.

As depicted in Fig. 2d, after each of s_5's neighbors have been checked, M chooses from P two additional *never-distributed* keys to include in K_5^C so that it contains 5 keys. Finally, M transmits K_5^C to s_5 and sets the states of all previously distributed keys, i.e., $k_1, k_2, \ldots, k_{18}$, back to *distributable* before the next sensor node is deployed.

3. Security analysis

In this section we present the security analysis of Secure Walking GPS with respect to Dolev-Yao and Wormhole attacks.

It is worth noting noise/interference effects on Secure Walking GPS. In the pre-deployment phase, we can assume that they are negligible since pre-deployment occurs in a secure base/area. When key distribution takes place during the actual deployment, if the messages between the master node and the sensor nodes are corrupt or lost, in addition to link layer retransmissions, the nodes can always be programmed to indicate the failure to the deployer (e.g., via LED) and auto-retry their communication until it succeeds. If it is impossible to have successful communication, the spot is probably non-deployable. In this case, the deployer can select another nearby spot for deployment. Nevertheless, noise/interference might affect the deployment completion time. Note also that it is unlikely that the sensors mistake a tampered message for a legitimate one, because all messages are encrypted using preset deployment keys.

3.1. Resistance to Dolev-Yao attack

According to our assumption, the secure base station is attack-free. Therefore, a deployer can be assured that legitimate program code is downloaded and that unique deployment key is distributed to each sensor node. Each unique deployment key is known only by the master node and one of the sensor nodes.

During the deployment, all the messages transmitted between the master node and the sensor nodes are encrypted using their respective deployment keys. Transmitted messages include a request message from each sensor node and a message from the master node containing the location and communication key set of the deployed sensor node. Since a Dolev-Yao attacker does not have a legitimate key, it is unable to decrypt legitimate messages and steal sensitive information from them. The attacker is unable to inject false messages either, because these false messages are not encrypted using proper keys and will, therefore, be simply dropped by sensor nodes. Similarly, the post-deployment neighbor discovery process and all subsequent communication between neighbors are encrypted using legitimate communication keys. Therefore, a Dolev-Yao attacker is not a significant threat.

Even if an attacker obtains a legitimate deployment or communication key by chance, its impact is limited because either one is distributed to and shared by only a small number of sensor nodes within a local region according to the Distance Bounding Rule.

3.2. Resistance to wormhole attack

A wormhole attacker deliberately launches this attack to replay legitimate messages at a remote point away from its origin, which violates the communication range constraint. A wormhole attack does not do much harm if the replay point and the origin of the tunneled message are close. In Secure Walking GPS, the master node and each of the sensor nodes are very close during the deployment. Therefore, a wormhole attack that occurs at this time (i.e., a wormhole attacks against the localization) would have limited effect.

For post-deployment inter-node communication, the Distance-Bounding Rule ensures that sensor nodes which are geographically located beyond their communication ranges do not share a communication key. If a node receives a message from a remote node which is tunneled through a wormhole link, it cannot process this message since it does not have a proper shared communication key to decrypt it. As a result, this message will be simply dropped.

Since the locations provided by the master node are not perfectly accurate, a location estimated by the master node may differ from the actual location. Consequently, the master node may consider two sensor nodes whose distance is a little greater than their communication range to be physical neighbors and distribute shared communication keys to them, resulting in a potential wormhole link. However, this vulnerability is insignificant. First, since priorities are given to the communication keys shared by closer neighbors when the master node determines each communication key set, it is less likely for two sensor nodes which are barely neighbors to share a communication key. Therefore, the number of potential wormhole links is relatively low, which means that it is difficult for a wormhole attacker to exploit such vulnerability. Second, even if an attacker launches a wormhole attack through one of the potential wormhole links, it causes limited threat since the replayed message is only tunneled to some point that is a little farther away from its legitimate reach.

In summary, our Secure Walking GPS scheme effectively reduces the impact of the wormhole attack on a WSN.

4. Performance evaluation

For our performance evaluation, we consider the following metrics: (1) the localization error obtained when using Secure Walking GPS; (2) the impact of distributing neighborhood keys on nodes communicating with their neighbors; (3) how successful is Secure Walking GPS in preventing the creation of wormholes (i.e., through its neighborhood key distribution); (4) scalability of Secure Walking GPS; and (5) overhead of our solution. It is worth mentioning that the presence of wormholes (a few might be established, despite our neighborhood keys) will not affect localization accuracy, since nodes obtain their locations directly from the master node, and not through node-to-node communication. The aforementioned metrics of interest, are further described below.

Let p be the probability that GPS signals are available to the master node during the deployment. Let S_{GPS} and S_{IG} be the sets of sensor nodes which are localized by the GPS module and by the IG module, respectively. The total number of sensor nodes n is equal to $|S_{\text{GPS}}| + |S_{\text{IG}}|$. Also let (x_i, y_i) be the reported location of sensor node s_i by the master node and $\left(x_i^{\text{real}}, y_i^{\text{real}}\right)$ be its real location.

The *average localization error* is defined by the cumulative localization error of all the sensor nodes divided by the total number of sensor nodes and can be expressed by:

$$err_{\text{AVG}} = \left(\sum_{s_i \in S_{\text{GPS}} \cup S_{\text{IG}}} \sqrt{\left(x_i - x_i^{\text{real}}\right)^2 + \left(y_i - y_i^{\text{real}}\right)^2} \right) \Big/ n$$

Since part of the average localization error comes from the GPS module and the other part comes from the IG module, we can further express the average localization error in terms of the average GPS localization error $err_{\text{AVG-GPS}}$ and the average IG localization error $err_{\text{AVG-IG}}$ as follows.

$$\begin{aligned} err_{\text{AVG}} = & \left(|S_{\text{GPS}}| \cdot \frac{\sum_{s_i \in S_{\text{GPS}}} \sqrt{\left(x_i - x_i^{\text{real}}\right)^2 + \left(y_i - y_i^{\text{real}}\right)^2}}{|S_{\text{GPS}}|} \right. \\ & \left. + |S_{\text{IG}}| \cdot \frac{\sum_{s_i \in S_{\text{IG}}} \sqrt{\left(x_i - x_i^{\text{real}}\right)^2 + \left(y_i - y_i^{\text{real}}\right)^2}}{|S_{\text{IG}}|} \right) \Big/ n \\ = & \frac{|S_{\text{GPS}}| \cdot err_{\text{AVG-GPS}} + |S_{\text{IG}}| \cdot err_{\text{AVG-IG}}}{n} \approx p \cdot err_{\text{AVG-GPS}} \\ & + (1-p) \cdot err_{\text{AVG-IG}} = f(p, err_{\text{AVG-GPS}}, err_{\text{AVG-IG}}) \end{aligned}$$

For a large-scale wireless sensor network, $err_{\text{AVG-GPS}}$ and $err_{\text{AVG-IG}}$ approximate the nominal localization accuracies of the GPS and the IG modules over which we have no control. Since the GPS module is often more accurate than the IG module, the above expression suggests that the average localization error is approximately a decreasing linear function of the GPS availability probability *p*.

Ideally, if a sensor node can communicate with all of its physical neighbors using some communication key, the ratio of the number of its useful neighbors to the number of its physical neighbors is 1. In reality, since two physical neighbors may not necessarily share a communication key and the fact that physical neighbors may not be able to communicate due to localization errors, this ratio is usually less than 1. The closer this ratio is to 1, the better a sensor node is connected with its neighbors. We define the average of such ratios for all sensor nodes as *average neighbor connectivity* $\overline{N_c}$:

$$\overline{N_c} = \left(\sum_{i=1}^{n} \frac{\#\ of\ s_i\text{'s useful neighbors}}{\#\ of\ s_i\text{'s physical neighbors}} \right) / n$$

This average reflects the degree to which neighboring sensor nodes in the WSN are inter-connected when they are allowed. If two sensor nodes share a communication key and their distance is smaller than their actual communication ranges (which may be different in two directions due to the irregularity and asymmetry of wireless radio patterns), there exists a legitimate link between them. If two sensor nodes share a communication key and their distance is greater than the theoretical communication range *r*, there exists a potential wormhole link between them. On the one hand, the *total number of legitimate links* is another indicator of neighbor connectivity, because the greater it is, the higher the chance neighboring sensor nodes can communicate. On the other hand, the *total number of wormhole links* and the *percentage of the total number of potential wormhole links to the total number of legitimate links* reflect the impact of potential wormhole attacks. A small percentage suggests that the impact of a wormhole attack is not severe to the network.

4.1. System evaluation

The proposed localization scheme requires that the deployer has a master node attached to it. We built a prototype master node that can be worn during deployment. This prototype consists of a GPS device mounted on top of a bicycle helmet. The GPS device is connected through and RS232 cable to the master node that is attached with a velcro to a wristband. Fig. 3 illustrates the prototype.

For the GPS device, we used the eTrex Legend device. The GPS device is WAAS (wide-area augmentation system) enabled, and it provides updated location information with high accuracy (error less than 3 m), at a rate of 1 Hz. Our choice to use a commercial GPS device for experiments was due to its ease of use and seamless integration. More sophisticated and better integrated, but more expensive, solutions are readily available today (e.g., Miniature Inertial Navigation Unit GPS 3DM-GX3-35 from Microstrain). We implemented our localization scheme in nesC (approximately 1500 lines of code) for the TinyOS operating system. For the master node, the total code size was approximately 17 KB and the data size was 595 bytes. The code size for the sensor nodes module was 972 bytes and the data size was 117 bytes. For sensor nodes we used Micaz motes.

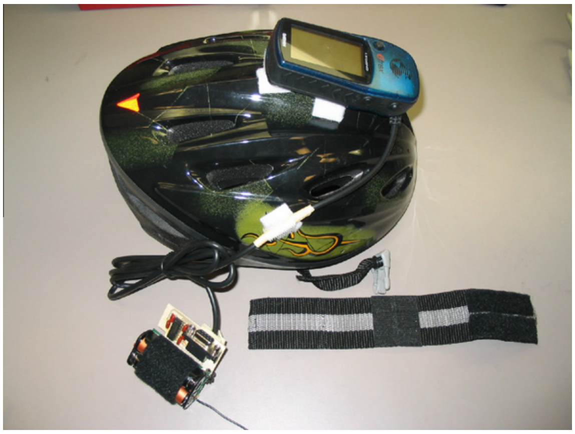

Fig. 3. Master node assembly.

The localization accuracy of the proposed localization solution, when only the GPS device is used, was evaluated in an open field. For an easier estimate of the localization error, we marked a 6 × 5 grid on the ground and we deployed the sensor motes in this grid. We want to emphasize the fact that the deployment being done in a grid was not used in any way during our localization. A deployment in any other regular geometric shape could have been performed. We used a grid because it was easy to create and it was easier to visually assess the performance.

In the experiments that follow, we provide numeric localization errors by performing a manual best fit of a strict grid with unit 10 m, to the experimental data. It is critical in understanding the following experimental results to note that the average location errors are not with respect to the "ground truth" location, but rather are relative to the known geometry of the deployment grid.

4.1.1. Single deployer

In this experiment we evaluated the localization accuracy from a deployment consisting of 30 MicaZ motes, in the aforementioned grid. Each node was turned on at its place of deployment, right before being deployed. The experimental results are shown in Fig. 4. The average

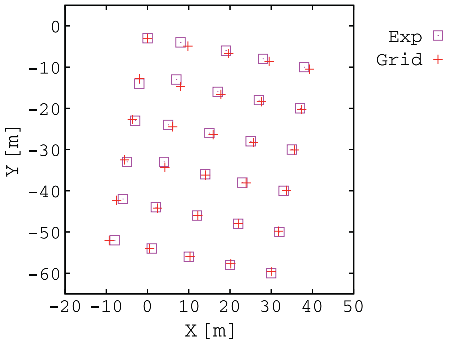

Fig. 4. Performance of the grid deployment with single deployer.

localization error obtained from fitting a grid to the experimental data is 0.8 m with a standard deviation of 0.5 m. From Fig. 4, as well as from the numerical results of the localization error, it can be observed a remarkably good fit. In this deployment type the errors are only due to the estimation of the global coordinate, done by the GPS hardware.

4.1.2. Dual deployer

The purpose of this experiment was to evaluate the performance of the proposed localization scheme when using two commercial GPS devices (the same model). A GPS device, as any other hardware device is dependent on calibration. Even after stringent calibration procedures, some variability in the indicated location is expected. From the direct reading of the global GPS location as shown by two GPS devices positioned next to each other, differences on the order of 1/1000 of a minute and sometimes even 1/100 of a minute, were observed. It was anticipated that these differences will contribute to an even larger localization error.

The deployment in this experiment was done along the length of the grid field (lines containing 6 motes). Three of the vertical lines (the middle and the two extreme ones) were deployed using one of the GPS devices, the other two vertical lines were deployed using the second GPS device. The experimental results are shown in Fig. 5.

The localization error obtained from our fitting of a grid to the experimental data is 1.6 m with a standard deviation of 0.9 m. In this deployment scenario, the average localization error is the largest. In addition to the errors encountered in previous experiments, here, the GPS device calibration has an additional contribution. When comparing the results of this experiment with the previous one, in which only one GPS device was used, it can be observed that the effect the device calibration has on location error was relatively small, of about 0.8 m.

4.2. Simulations

For investigating the accuracy (from the inclusion of the IG system) and robustness of Secure Walking GPS against attacks, we performed simulations. For our simulations we adopt the parameters of a real WSN surveillance system that we had experience with [2]. A large-scale sensor network of n sensor nodes is deployed in an outdoor field where the GPS signals are available to the master node with a probability p. This means that about $p \times 100\%$ of the nodes will be localized by the GPS module and about $(1-p)\times\ 100\%$ will be localized by the IG module. Let the number of communication keys that each node obtains from the master node be 5, and assume that these keys can always be transmitted from the master node to each deployed sensor node during the deployment. Let the localization error of the GPS module be uniformly distributed $\mathcal{U}(-1.5, 1.5)$ m. The localization error of the IG module is a combined result of the error of degree estimation by the rotation sensors and the error of timely movement detection by the motion sensors. Let the rotation sensor error be uniformly distributed $\mathcal{U}(-10, 10)^\circ$, and the motion sensor error result in a reduction of distance estimation of the deployer's path between consecutive sensor nodes which is uniformly distributed $\mathcal{U}(0, 3)$ m. Let the regular communication range of each sensor node r be 30 m. When we consider irregular radio ranges (to evaluate the impact of an asymmetric radio on our proposed secure localization and key distribution scheme), the communication range of a sensor node, in each 1° direction, is uniformly distributed $\mathcal{U}(15, 45)$ m.

4.2.1. Line deployment

First, we consider a line deployment wherein a deployer roughly follows a line and deploys sensor nodes at desired locations. Fig. 6a gives an example of such a deployment, where the dashed line represents the deployment line, solid dots represent deployed sensor nodes, and arrows represent the deployer's path.

We simulate a deployment of 500 sensor nodes with the same regular radio pattern. The horizontal spacing between sensor nodes is normally distributed $\mathcal{N}(10, 2)$ m, and the vertical offset of each sensor node from the deployment line is normally distributed $\mathcal{N}(0, 2)$ m. We evaluate the performance of our scheme at $p = 0.75$, 0.80, 0.85, 0.90, 0.95, 1.00. For each p, we performed 30 simulations

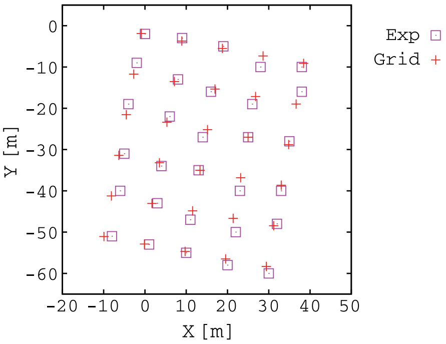

Fig. 5. Performance of the grid deployment with dual deployer.

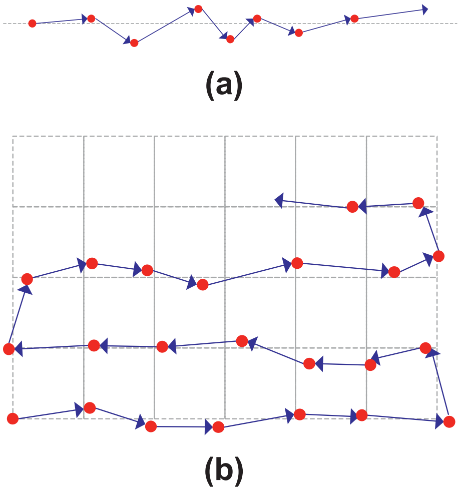

Fig. 6. A line deployment (a), and a grid deployment (b).

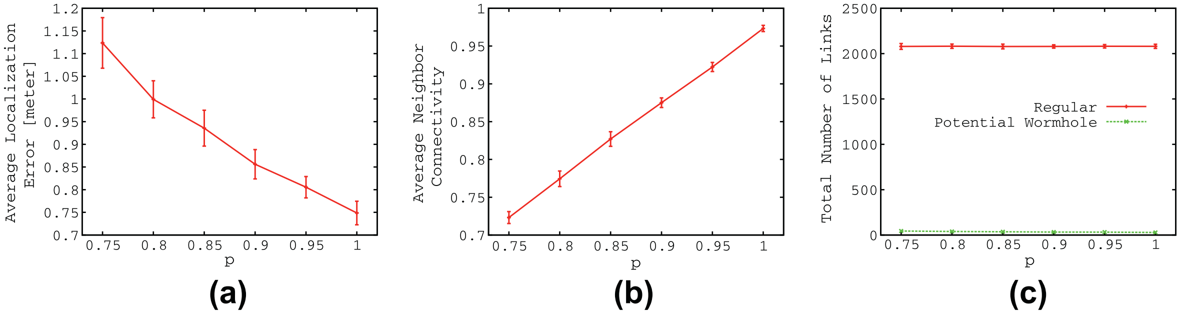

Fig. 7. Performance of the line deployment with regular radio.

and calculated the average localization error, average neighbor connectivity, the total number of legitimate links, and the total number of potential wormhole links. Mean values with one standard deviations for each of these metrics are plotted in Fig. 7.

As shown in Fig. 7a, the average localization errors are between 0.72 m and 1.18 m. We observe a decrease in both the mean and the standard deviation of the average localization error as *p* increases. While the decrease in mean is because more nodes can be localized using the more accurate GPS module, the decrease in the standard deviation is explained by the fact that the smaller the portion of the nodes which are localized using the IG module, the less the impact of its cumulative errors due to more often calibrations with the GPS module during the deployment. The average localization error curve is roughly linear, which confirms that it is a linear function of *p* given an average GPS localization error and an average IG localization error. Fig. 7b shows the average neighbor connectivity with respect to *p*. The average neighbor connectivity ranges between [0.72, 0.97] and is an increasing function of *p*, reflecting the impact of location errors on the key distribution decisions. Fig. 7c depicts the total number of legitimate links in the WSN versus the total number of potential wormhole links. Compared with that of legitimate links (ranging between 2040 and 2100), the number of potential wormhole links is extremely low (below 50). Therefore, a wormhole attacker has only a chance of about 2.5% of successfully exploiting a potential wormhole link and establishing a wormhole attack. Even if a wormhole attack occurs, its impact will be small, due to the Distance Bounding Rule.

4.2.2. Grid deployment

Next, we consider a grid deployment wherein a deployer walks back and forth horizontally through the grid and deploys sensor nodes at desired locations. Fig. 6b gives an example of a small grid deployment to illustrate how the deployer traversed the grid for the deployment. In this figure, dashed lines represent the borders of the grids, solid dots represent deployed sensor nodes, and arrows represent the deployer's path.

Assume that 500 sensor nodes with the same regular radio pattern are going to be deployed in a grid fashion. Let the horizontal spacing between sensor nodes be normally distributed $\mathcal{N}(10, 2)$, and let the vertical offset of each sensor node from each horizontal deployment line be normally distributed $\mathcal{N}(0, 2)$. We performed 30 simulations for each $p = 0.75, 0.80, 0.85, 0.90, 0.95$, and 1.00. We plot our results with mean values and one standard deviation error bars in Fig. 8.

From Fig. 8a, the mean value of the average localization error drops from 1.33 m to 0.73 m, as *p* increases from 0.75 to 1.00. There is also an observable decrease in the standard deviation as well. The average localization error curve is roughly linear with *p*. In Fig. 8b, the average neighbor connectivity is as high as 0.97 when $p = 1.00$. However, it drops to about 0.68 when $p = 0.75$. Since our key distribution scheme attempts to be fair to every neighbor, sensor nodes will have more useful neighbors in a grid deployment. However, the number of shared keys per neighbor will be smaller. Therefore, the combined effect does not cause a significant change in the total number of legitimate links. This is confirmed from the result in Fig. 8c that the total number of legitimate links ranges between 2050

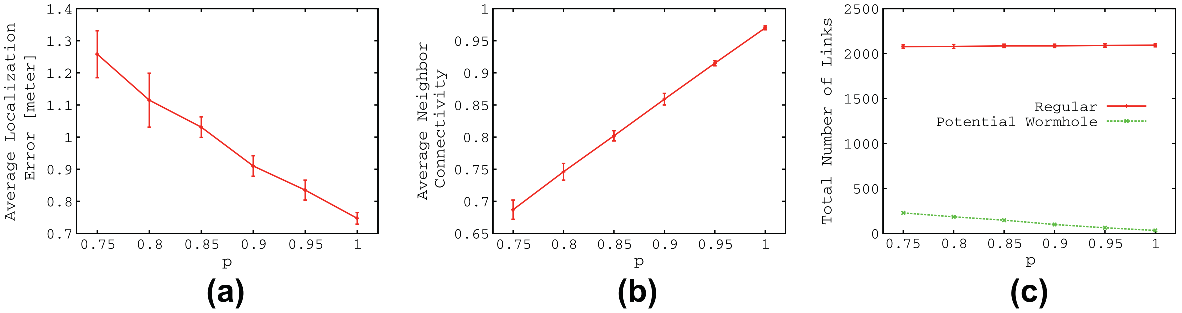

Fig. 8. Performance of the grid deployment with regular radio.

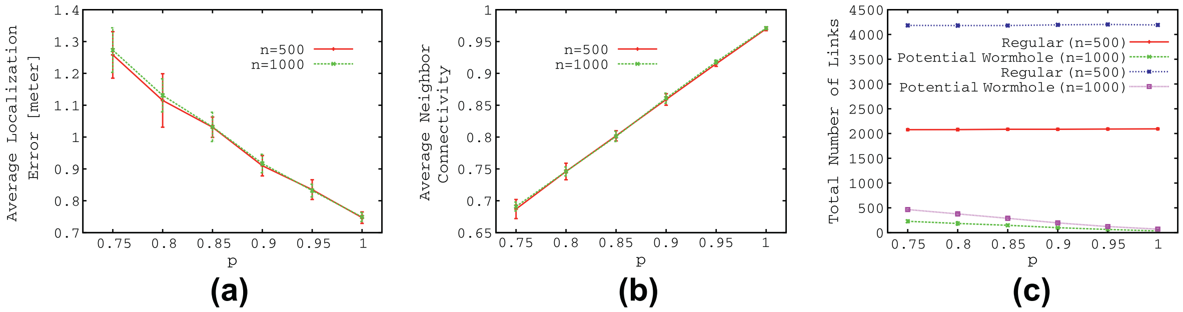

Fig. 9. Performance of the scaled deployment deployment with regular radio ($n = 500$, 1000).

and 2100. However, the total number of potential wormhole links grows to about 250 when p drops to 0.75, due to more localization errors.

4.2.3. Scalability

We evaluated the performance of Secure Walking GPS as the size of a deployed WSN increases. We perform simulations with the total number of sensor nodes being 1000 in a grid deployment with the same regular radio pattern, at $p = 0.75$, 0.80, 0.85, 0.90, 0.95, and 1.00, and compared the results with those in Section 4.2.1. Mean values with one standard deviations for each of the metrics are plotted in Fig. 9. From this figure, we observe that the average localization error and average neighbor connectivity are almost the same for $n = 500$ and $n = 1000$. Therefore, the curves corresponding to different n are quite close to each other both in Fig. 9a and b. In Fig. 9c, the total number of legitimate links and the total number of potential wormhole links increase proportionally with n, the size of the WSN. These results indicate that our scheme is scalable for large-scale WSN deployments.

4.2.4. Radio irregularity

Finally, we performed simulations to explore the impact of irregular radio pattern in a grid deployment. The simulation settings were the same as those in Section 4.2.2, except that the communication range of each sensor node in each direction was uniformly distributed $\mathcal{U}(15, 45)$ m.

The results showed that the irregular radio patterns could reduce the average neighbor connectivity, the total number of legitimate links and the total number of wormhole links: the average localization error range was [0.73, 1.31] m. The average neighbor connectivity ranges between [0.52, 0.85]. The total number of legitimate links is between [1627, 1740], and the total number of potential wormhole links is between [222, 17]. In our 30 runs of the simulation, we have not encountered any (worst) case where more than five communication keys are required for each sensor node to establish neighbor connectivity.

4.3. Overhead

The overhead of our Secure Walking GPS scheme is low in several aspects.

4.3.1. Hardware overhead

The only additional hardware used is the GPS and IG modules, whose costs are fixed and occur only once. Since the size of the sensor network can be arbitrarily large and the hardware can be reused for multiple deployments, the amortized hardware overhead is negligible.

4.3.2. Communication overhead

In pre-deployment and post-deployment, all nodes communicate in a "request-reply" fashion, thus transmitting the minimum necessary number of messages and consuming as little energy as possible. Encrypting every message could lead to an increase in the total number of necessary messages transmitted in the sensor network after the deployment. For example, instead of broadcasting the messages, two physical neighbors may have to use intermediate neighbors to route their messages, when they do not directly share a communication key. However, we are willing to trade this increase for security.

4.3.3. Storage overhead

To enable cryptography, each sensor node needs to store 1 deployment key (for communication with the master node) and m communication keys (for communication with its neighbors). If each key is 16 bytes long, the required amount of memory on each sensor node to store them is only $16 \times (m + 1)$ bytes, which is small and adequately fits well on most of today's sensor nodes. Evaluating the tradeoff between the size of the communication keys and the performance of the deployment would require an implementation of a realistic WSN application. Due to the diversity of WSN applications, is it difficult to precisely measure an "average" effect of communication keys on application performance. Instead, we indicate that the communication keys in Secure Walking GPS require less storage than similar, state of art solutions [11].

Additionally, the number of keys managed by the master node is roughly proportional to the number of sensor nodes. However, this is not a problem for a typical master node, which should be able to support the necessary memory needs.

5. Related work

WSNs are inherently vulnerable to various attacks due to the insecure nature of wireless communication and

Please cite this article in press as: Q. Mi et al., Practical and secure localization and key distribution for wireless sensor networks, Ad Hoc Netw. (2012), doi:10.1016/j.adhoc.2011.12.008

the severe resource constraints on sensor nodes. As a result, determining node locations in a hostile environment is challenging.

Sequence-based localization is an approach to resisting attacks on ranging results in wireless networks. Specifically, a deployment area is divided into non-overlapping subregions by the perpendicular bisectors for the anchor pairs. Each subregion is assigned a unique sequence code word that represents the relative distance ranking of each anchor; and each node is mapped to a subregion once its estimate or measured distances to anchors are available. Observe that if the number of valid sequence code words is considerably smaller than the total number of possible sequence code words, robust detection of attacks and correction of location errors in the sequences can be achieved. The performance of sequenced-based localization is largely dependent on the number of anchors.

Capkun proposed two mechanisms for secure localization in wireless networks [12]. The first one, Verifiable Multilateration, enables secure computation and verification of locations based on distance bounding and authenticated ranging protocols. The second one, Secure Localization with Hidden Base stations, makes use of the unpredictability of base station locations to enable secure localization. Both mechanisms require hardware support such as high clock precision and complex base station infrastructure. Therefore, they may face challenges in resource-constrained sensor networks.

In [13], Park and Shin presented an attack-tolerant localization protocol, Verification for Iterative Localization (VeIL). Localization is achieved using a profile manager that adaptively tracks the profile of normal localization behavior and an attack detector that detects attacks by iteratively verifying location announcements via comparison against the normal profile. However, if the number of anchors is small, or the anchors are non-trustworthy, or the ranging accuracy is low, the performance of VeIL is likely to degrade.

Lazos and Poovendran proposed a range-independent localization algorithm called SeRLoc in [14]. Using message encryption, the properties of sector uniqueness and communication range violation, and the Attach to Closer Locator Algorithm, sensor nodes can determine their locations during wormhole attacks, sybil attacks, and compromised sensors. As a successor to SeRLoc, HiRLoc [15] achieves passive sensor localization based on beacon information transmitted from the locators with improved resolution at the cost of increased computational complexity and communication. In both SeRLoc and HiRLoc, locators are assumed to be trusted and have known locations. However, they are often the actual targets in a real attack.

Liu et al. proposed two methods to achieve attack-resistant beacon-based location estimation in sensor networks in [16]. The first method, attack-resistant Minimum Mean Square Estimation, identifies malicious location references by examining the inconsistency among location references and removes malicious data. The second method quantizes the deployment field into grids and has each location reference vote on the cells where a node may reside. These two methods work under the assumptions that the majority of location references are benign and ranging is accurate, which may not always hold in hostile environments.

Sequence-based localization is an approach to resisting attacks on ranging results in wireless networks [17]. The performance of sequenced-based localization is largely dependent on the number of anchors. In [18], Li et al. developed two robust statistical methods to make localization attack-tolerant. These two methods assume that legitimate distance or signal strength measurements outnumber malicious readings. However, in a sophisticated attack such as the wormhole attack, legitimate measurements may be outnumbered.

Shokri et al. designed a secure neighbor verification protocol with a proof-of-concept implementation on Cricket motes [19]. The protocol involves ranging, neighbor table exchange, and geometric link verification and has been demonstrated to be effective against the wormhole attack. However, it requires that each sensor node has special hardware to perform ranging and be synchronized to microsecond order with each other, which may be difficult to apply to large-scale deployments where cost becomes an issue.

Secure communication between legitimate nodes can be achieved by encrypting and authenticating the messages using keys. As a result, many works have been dedicated to efficient key distribution in a WSN.

In the probabilistic pairwise key predistribution scheme [20] by Eschenauer, each node is preassigned a random set of k keys from a large key pool P. This scheme may require the key manager and sensor nodes to have a large storage capacity in order to hold the keys. In addition, this scheme cannot guarantee that a node will always share a key with a neighbor. In [21], Camtepe and Yener proposed a deterministic implementation of Eschenauer's scheme. Each node still receives a subset of keys from a key pool P. However, rather than choosing each subset randomly, the subsets are constructed to guarantee that each node pair share a key and each key in P appears in the same number of key subsets. The difficulty of this scheme is that the number of nodes must be known in advance when key subsets are generated.

Liu and Ning proposed two location-based pairwise key establishment schemes for static sensor networks [22]. Their schemes have a high probability to establish direct keys between neighbors. However, not only are expected node locations required to be known before key establishment, but specific nodes also need to be correctly placed at their expected locations. These two requirements impose substantial manual work before and during the deployment.

In [23], the authors formalized the modeling of wormhole links using the graph theory and presented two mechanisms to defend against the wormhole attacks. However, their centralized mechanism requires that all node locations be known in advance to a central authority before key distribution and their decentralized mechanism uses multiple special guard nodes where their locations must be determined in some way and they share a global key that is assumed not to be compromisable.

While keys are prepopulated before the deployment in the previous works, Kuo et al. proposed Message-In-A-Bottle (MIB) [24], a scheme to securely deploy keys to sensor nodes inside a shielded Faraday cage during the deployment. Techniques such as key segmentation, activation, and verification are employed to defeat the Dolev-Yao attacks. Nevertheless, this deployment scheme requires much human interaction.

This article extends the results reported in [1,25] with a formal proof for the theorem that gives the lower bound on the number of keys to be distributed on a sensor nodes, a clarifying example, and more extensive security analysis and performance evaluations.

6. Conclusions

In this article, we presented the design and evaluation of Secure Walking GPS, an integral solution for secure localization and location-based key distribution in large-scale and manually deployed WSNs. Secure Walking GPS is practical and low-cost, requires minimal human interaction during the deployment, and makes the deployed WSN resistant to the Dolev-Yao, the wormhole, and the GPS-denial attacks.

In our current version of Secure Walking GPS, the communication among neighbors is mostly unicast or multicast since not all neighbors have the communication key to decrypt any legitimate message that they can hear. We plan to consider the distribution of "neighborhood keys" in our next step so that broadcast communication in the presence of attacks can also be supported in a secure way.

Acknowledgments

This work was supported, in part, by Grants ARO W911NF-06-1-0204, and NSF OCI-1127449 and CNS-0923203.

References

[1] Q. Mi, J.A. Stankovic, R. Stoleru, Secure Walking GPS: a secure localization and key distribution scheme for wireless sensor networks, in: Proceedings of the 3rd ACM Conference on Wireless Network Security (WiSec), ACM, 2010.

[2] T. He, P. Vicaire, T. Yan, L. Luo, L. Gu, G. Zhou, R. Stoleru, Q. Cao, J. Stankovic, T. Abdelzaher, Achieving real-time target tracking using wireless sensor networks, in: Proceedings of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), IEEE Computer Society, 2006, pp. 37–48.

[3] L. Selavo, A. Wood, Q. Cao, T. Sookoor, H. Liu, A. Srinivasan, Y. Wu, W. Kang, J. Stankovic, D. Young, J. Porter, Luster: Wireless sensor network for environmental research, in: Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys), ACM, 2007, pp. 103–116.

[4] K. Seada, M. Zuniga, A. Helmy, B. Krishnamachari, Energy efficient forwarding strategies for geographic routing in lossy wireless sensor networks, in: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys), ACM, 2004, pp. 108–121.

[5] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, in: Proceedings of the 6th ACM International Conference on Mobile Computing and Networking (Mobicom), ACM, 2000, pp. 56–67.

[6] S. Beauregard, Omnidirectional Pedestrian Navigation for First Responders, Tech. Rep., Universitat Bremen, 2007.

[7] V. Adamchuk, Global Positioning System Data Processing, Tech. Rep., University of Nebraska Lincoln, 2010.

[8] A. Srinivasan, J. Wu, A Survey on Secure Localization in Wireless Sensor Networks, CRC Press, Taylor and Francis Group, 2008.

[9] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inform. Theory 29 (2) (1983) 198–208.

[10] A. Wood, J. Stankovic, Poster abstract: AMSecure – secure link-layer communication in tinyos for IEEE 802.15.4-based wireless sensor networks, in: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys), ACM, 2006, pp. 395–396.

[11] P. Traynor, R. Kumar, H. Bin Saad, G. Cao, T. La Porta, LIGER: implementing efficient hybrid security mechanisms for heterogeneous sensor networks, in: Proceedings of the 4th International Conference on Mobile Systems, Applications and Services (MobiSys), ACM, 2006.

[12] S. Capkun, Secure localization in wireless networks (using verifiable multilateration and covert base stations), in: Book Chapter, Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks, Springer, 2007.

[13] T. Park, K.G. Shin, Attack-tolerant localization via iterative verification of locations in sensor networks, ACM Trans. Embed. Comput. Syst. 8 (1) (2008).

[14] L. Lazos, R. Poovendran, Serloc: secure range-independent localization for wireless sensor networks, in: Proceedings of the 3rd ACM Workshop on Wireless Security (WiSe), 2004, pp. 21–30.

[15] L. Lazos, R. Poovendran, Hirloc: high-resolution robust localization for wireless sensor networks, IEEE J. Select. Areas Commun. 24 (2) (2006) 233–246.

[16] D. Liu, P. Ning, W.K. Du, Attack-resistant location estimation in sensor networks, in: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN), IEEE, 2005, pp. 99–106.

[17] B. Krishnamachari, K. Yedavalli, Secure sequence-based localization for wireless networks, in: Book Chapter, Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks, Springer US, 2007.

[18] Z. Li, W. Trappe, Y. Zhang, B. Nath, Robust statistical methods for securing wireless localization in sensor network, in: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN), IEEE, 2005, pp. 91–98.

[19] R. Shokri, M. Poturalski, G. Ravot, P. Papadimitratos, J.-P. Hubaux, A practical secure neighbor verification protocol for wireless sensor networks, in: Proceedings of the 2nd ACM Conference on Wireless Network Security (WiSec), ACM, 2009, pp. 193–200.

[20] L. Eschenauer, V. Gligor, A key-management scheme for distributed sensor networks, in: Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS), ACM, 2002, pp. 41–47.

[21] S. Camtepe, B. Yener, Combinatorial design of key distribution mechanisms for wireless sensor networks, IEEE/ACM Trans. Network. 15 (2) (2007) 346–358.

[22] D. Liu, P. Ning, Location-based pairwise key establishments for static sensor networks, in: Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN), 2003, pp. 72–82.

[23] R. Poovendran, L. Lazos, A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks, Wirel. Netw. 13 (1) (2007).

[24] C. Kuo, M. Luk, R. Negi, A. Perrig, Message-in-a-bottle: user-friendly and secure key deployment for sensor nodes, in: Proceedings of the 5th ACM Conference on Embedded Networked Sensor Systems (SenSys), ACM, 2007, pp. 233–246.

[25] R. Stoleru, T. He, J. Stankovic, Walking GPS: a practical solution for localization in manually deployed wireless sensor networks, in: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN), IEEE Computer Society, 2004, pp. 480–489.

Qi Mi received his BS degree in Electrical Engineering from Shanghai Jiao Tong University, China in 2004 and an ME degree in Computer Engineering from the University of Virginia in 2009. His research interests are wireless sensor networks, node localization, and security. He currently works as a software developer at Microsoft in Redmond, WA.

John A. Stankovic is the BP America Professor in the Computer Science Department at the University of Virginia. In the past he served as Chair of the department for 8 years. He is a Fellow of both the IEEE and the ACM. He also won the IEEE Real-Time Systems Technical Committee's Award for Outstanding Technical Contributions and Leadership. He also won the IEEE Technical Committee on Distributed Processing's Distinguished Achievement Award (inaugural winner). He has won four Best Paper awards in sensor networks including for ACM SenSys 2006. Before joining the University of Virginia, Professor Stankovic taught at the University of Massachusetts where he won an outstanding scholar award. He has also held visiting positions in the Computer Science Department at Carnegie-Mellon University, at INRIA in France, and Scuola Superiore S. Anna in Pisa, Italy. He was the Editor-in-Chief for IEEE Transactions on Distributed and Parallel Systems and was founder and co-editor-in-chief for the Real-Time Systems Journal. His research interests are in cyber physical systems, distributed computing, real-time systems, wireless sensor networks, and security for sensor networks. Prof. Stankovic received his PhD from Brown University.

Radu Stoleru is an Assistant Professor in the Department of Computer Science and Engineering at Texas A&M University, and the head of the Laboratory for Embedded & Networked Sensor Systems (LENSSs). His research interests are in deeply embedded wireless sensor systems, distributed systems, embedded computing, and computer networking. He received his PhD in computer science from the University of Virginia in 2007. While at the University of Virginia, he received from the Department of Computer Science the Outstanding Graduate Student Research Award for 2007. He has authored or co-authored over 50 conference and journal papers with over 1000 citations. He is currently serving as an editorial board member for three international journals and has served as technical program committee member on numerous international conferences.

Please cite this article in press as: Q. Mi et al., Practical and secure localization and key distribution for wireless sensor networks, Ad Hoc Netw. (2012), doi:10.1016/j.adhoc.2011.12.008