Secure Walking GPS: A Secure Localization and Key Distribution Scheme for Wireless Sensor Networks

Qi Mi [1], John A. Stankovic [1], Radu Stoleru [2]
[1] Department of Computer Science, University of Virginia, USA
[2] Department of Computer Science and Engineering, Texas A&M University, USA
[1] {qimi, stankovic}@cs.virginia.edu, [2] stoleru@cse.tamu.edu

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 paper, we introduce Secure Walking GPS, a secure localization and key distribution solution for 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. Our 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, has good localization and key distribution performance, and is practical for large-scale WSN deployments.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General-Security and protection, (e.g., firewalls)

General Terms

Algorithm, Design, Security

Keywords

wireless sensor network, secure localization, key distribution

1. INTRODUCTION

Wireless sensor networks (WSNs) are 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. When a WSN is manually deployed in a potentially hostile environment and left unattended for a long period of time, it is vulnerable to various attacks during and after its deployment. For example, an attacker may try to steal sensitive data from the legitimate messages, to inject false messages into the network, or to disrupt the normal operation of WSN services and applications. Therefore, to ensure that a WSN operates as expected, it is crucial that 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) an extension to Walking GPS [15], making it 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.

2. SECURE WALKING GPS

Walking GPS [15] is a practical localization scheme for manually deployed WSNs. However, it suffers from Dolev-Yao, wormhole and GPS-denial attacks due to lack of adequate security protection. Our solution to this is Secure Walking GPS, an extension to Walking GPS, that securely localizes sensor nodes and distributes carefully chosen communication keys to nodes being deployed. Secure Walking GPS also uses a master node during node deployment, which obtains its current location and sends it to each newly deployed sensor node wirelessly. However, Secure Walking GPS is different from Walking GPS in two key aspects:

(1) Communication keys, for neighborhood communication, are efficiently distributed to sensor nodes during the node localization process. These communication keys help the WSN effectively resist the Dolev-Yao and the wormhole attacks during and after the deployment.

(2) An inertial guidance (IG) module complements the function of GPS on the master node. The IG module uses

WiSec'10, March 22–24, 2010, Hoboken, New Jersey, USA.
Copyright 2010 ACM 978-1-60558-923-7/10/03 ...$10.00.

motion sensors to continuously capture the orientation and velocity of the deployer, and estimates its location via dead reckoning. 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.

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 and routing. 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. 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 [3]. 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 can compromise the second phase of Walking GPS where node collaboration is involved. In a GPS-denial attack, GPS signals are intermittently lost due to physical obstacles or purposeful jamming. This also poses an issue for Walking GPS, as the master node derives its location solely based on the GPS signals.

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 (as physical obstacles). Sensor nodes do not know which other nodes would become their neighbors until after they are actually deployed. Also, we assume that the GPS signals are not always available during the deployment, either because of temporary lack of Line-of-Sight GPS signals due to the surrounding environment, or because of purposeful GPS-denial attacks.

We assume that the master node is a powerful node and it will not be compromised by any attack. We assume that the IG module is always available and it 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 sensors to infer its location, and a military GPS device usually has anti-spoofing capabilities.

2.3 Design Details

2.3.1 Pre-Deployment

Secure Walking GPS begins with a pre-deployment phase in the secure base station, whose main aim 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.

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 can be 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 one sensor node to which this key has been distributed. In the following, we use the notations described in Table 1.

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, ($k_{i,l}^{\mathrm{C}}$ where $l = \overline{1, m}$) distributed to s_i
NID($node$)	the id of $node$
KID(k)	the key id of k

Since the pre-deployment is done in the secure base station, deployment keys can be distributed in plain text:

$$\begin{aligned}
&s_i \rightarrow M : \mathrm{NID}(s_i) \| \mathrm{REQ\_PRE\_DEPLOYMENT} \\
&M \rightarrow s_i : \mathrm{NID}(M) \| k_i^{\mathrm{D}} \\
&s_i \rightarrow M : \mathrm{NID}(s_i) \| \mathrm{ACK\_PRE\_DEPLOYMENT}
\end{aligned}$$

A sensor node s_i sends a message to the master node M, containing its node id and a REQ_PRE_DEPLOYMENT request to ask for its deployment key, if it has not successfully obtained one from M before. When M receives it, M 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. 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 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 its non-volatile memory for later use and replies to M with an acknowledgement message.

Since each deployment key is unique and 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.

2.3.2 Deployment

A. 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 protocol to obtain its location and the set of m communication keys securely:

$$s_i \to M : \mathrm{NID}(s_i) \| \{\text{REQ\_DEPLOYMENT}\}_{k_i^{\mathrm{D}}}$$
$$M \to s_i : \mathrm{NID}(M) \| \{location\}_{k_i^{\mathrm{D}}} \| \{k_{i,1}^{\mathrm{C}}, k_{i,2}^{\mathrm{C}}, \cdots, k_{i,m}^{\mathrm{C}}\}_{k_i^{\mathrm{D}}}$$
$$s_i \to M : \mathrm{NID}(s_i) \| \{\text{ACK\_DEPLOYMENT}\}_{k_i^{\mathrm{D}}}$$

After initialization, s_i sends a message to M, containing its node id and a REQ_DEPLOYMENT request. Note that the REQ_DEPLOYMENT request is encrypted using s_i's deployment key k_i^{D}, but the source id is sent in plain text so that the master node can use it to look up k_i^{D} from its own memory and decrypt the request message 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}. If s_i receives them, it securely acknowledges success to the master node.

In a WSN deployment using Secure 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, the IG module needs to be calibrated periodically with the GPS module, whenever the GPS signals are available.

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

B. Location-Based Key Distribution

In addition to a location, a set of m communication keys is distributed to each sensor node when it is deployed so that it can have secure communication with neighboring nodes. The choice of communication keys to make up the key set is determined by master node at real time during deployment, based on the estimated locations of the current sensor node and all 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 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.

We enforce two rules for our location-based key distribution and present the algorithms in Algorithms 1 and 2.

Distance Bounding Rule: *Two sensor nodes can share a communication key only if they are physical neighbors.* [1]

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 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 securely using their respective deployment keys. Each communication key in P is randomly generated and unique. It is indexed by a communication key id and 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 Location-based Key Distribution

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

Choosing 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 securely transmits them to s_1. 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. For each subsequent sensor node s_i ($i = \overline{2, n}$) deployed, the master node M goes through the following steps to determine which communication keys should be chosen for it.

Step 1: Find s_i's *physical neighbors* from the set of sensor nodes that have already been deployed.

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 attemps 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 $S_i = \{s_1, s_2, \cdots, s_{i-1}\}$ into A_i and B_i, where $A_i = \{s_{\sigma_{(j)}} \mid d_{i,\sigma_{(j)}} < r \ \wedge \ M$ can communicate with $s_j\}$ and $B_i = S_i - A_i$.

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.

Step 3: Determine which m communication keys can be distributed to s_i.

If s_i's closest physical neighbor $s_{\sigma_{(1)}}$ has only one dis-

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

Algorithm 2 GET_KEYS (S_i, P, P')

```
for $j$ from $1$ to $i-1$ do
    Calculate $d_{i,j} = |s_i - s_j|$
end for
for $j$ from $1$ to $i-1$ do
    if $M$ cannot communicate with $s_j$ then
        $d_{i,j} \leftarrow +\infty$
    end if
end for
$\{\sigma_{(l)} \mid l = \overline{1, i-1}\}$ = PERMUTATE$\{j \mid j = \overline{1, i-1}\}$,
    where $d_{i,\sigma_{(l)}} \leq d_{i,\sigma_{(l+1)}}$
$S_i = A_i \cup B_i$, where $A_i = \{s_{\sigma_{(j)}} \mid d_{i,\sigma_{(j)}} < r \wedge$
    $M$ can communicate with $s_j\}$ and $B_i = S_i - A_i$
for $l$ from $(|A_i| + 1)$ to $(|A_i| + |B_i|)$ do
    for $n$ from $1$ to $m$ do
        $k^{\mathrm{C}}_{\sigma_{(l)},n}.state \leftarrow$ non-distributable
    end for
end for
$num \leftarrow 0$
$K^{\mathrm{C}}_i \leftarrow \phi$
$u \leftarrow 1$
while $(num < m-1) \wedge (\exists$ distributable keys in $P') \wedge$
    $(u < i)$ do
    $D_i = \{k^{\mathrm{C}}_{\sigma_{(u)},v} | v = \overline{1,m} \wedge k^{\mathrm{C}}_{\sigma_{(u)},v}.state = distributable\}$
    $\{\delta_{(w)} \mid w = \overline{1, |D_i|}\}$ = PERMUTATE$\{v \mid v = \overline{1, |D_i|}\}$,
        where $k^{\mathrm{C}}_{\sigma_{(u)},\delta_{(w)}}.freq \geq k^{\mathrm{C}}_{\sigma_{(u)},\delta_{(w+1)}}.freq$
    $K^{\mathrm{C}}_i \leftarrow K^{\mathrm{C}}_i \cup \{k^{\mathrm{C}}_{\sigma_{(u)},\delta_{(1)}}\}$
    $num \leftarrow num + 1$
    if $d_{i,\sigma_{(u)}} \geq r/2$ then
        for $w$ from $1$ to $|D_i|$ do
            $k^{\mathrm{C}}_{\sigma_{(u)},\delta_{(w)}}.state \leftarrow$ non-distributable
        end for
    else
        $k^{\mathrm{C}}_{\sigma_{(u)},\delta_{(1)}}.state \leftarrow$ non-distributable
    end if
    $u \leftarrow u + 1$
end while
$K^{\mathrm{C}}_i \leftarrow K^{\mathrm{C}}_i \cup \{(m - num)$ never-distributed keys from
    $P\}$
return $K^{\mathrm{C}}_i$
```

tributable communication key, M includes it in s_i's communication key set K^{C}_i 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^{C}_i, 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, $\cdots$, closest physical neighbors $(s_{\sigma_{(2)}}, s_{\sigma_{(3)}}, \cdots)$ until $(m-1)$ *distributable* communication keys from s_i's physical neighbors are included in K^{C}_i 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^{C}_i.

Note that M deliberately includes at least one *never − distributed* communication key in K^{C}_i so that s_i may share it with future neighbors which have not been deployed yet. The 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 carefully chosen communication keys to s_i, securely using its deployment key.

Step 5: Reset the states of all *non-distributable* communication keys to *distributable* before the next sensor node is deployed.

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 (proof provided in [11]) 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:*

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

Note that the simplifying assumption of circular communication range is used in the theorem only to provide 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, 5 (five) communication keys suffice in the ideal case. Even in real environments where the radio pattern is irregular, we don't expect m_{min} to increase much beyond 5. Our empirical evaluation results in [11] confirm this conclusion.

2.3.3 Post-Deployment

After the deployment, each sensor node has obtained a location and m communication keys from the master node. Then the sensor nodes try to discover their *useful neighbors*, which are within their actual communication ranges and share at least one communication key. To do so, each sensor node repeatedly broadcasts messages that are encrypted using each of its communication keys. If s_i can hear a message from s_j and decrypt it using one of its own communication keys, s_i replies to s_j with a message encrypted with the same communication key. This process allows s_i and s_j to discover that the other node is a useful neighbor. As a result, subsequent communication between useful neighbors can be encrypted using any of their shared communication keys.

3. SECURITY ANALYSIS

Resistance to Dolev-Yao Attack According to our assumption, the secure base station is attack-free. Therefore, legitimate program code is downloaded and a 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. Since a Dolev-Yao attacker does not have a legitimate key, it is unable to decrypt these messages and steal sensitive information from them. It 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 shared by only a small number of sensor nodes within a local region according to Distance Bounding Rule.

Resistance to Wormhole Attack 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 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, the threat is small since the replayed message is tunneled to some point that is a little farther away from where it can reach.

Resistance to GPS-Denial Attack The IG module comes into play when the GPS module does not work, making our scheme resistant to the GPS-denial attack.

4. PERFORMANCE EVALUATION

In this section, we study the robustness of our scheme to a GPS-denial attack and explore how likely a wormhole attack may succeed, assuming that the radio pattern is regular [2].

The *average localization error* is defined by the cumulative localization error of all the sensor nodes divided by the total number of deployed sensor nodes n.

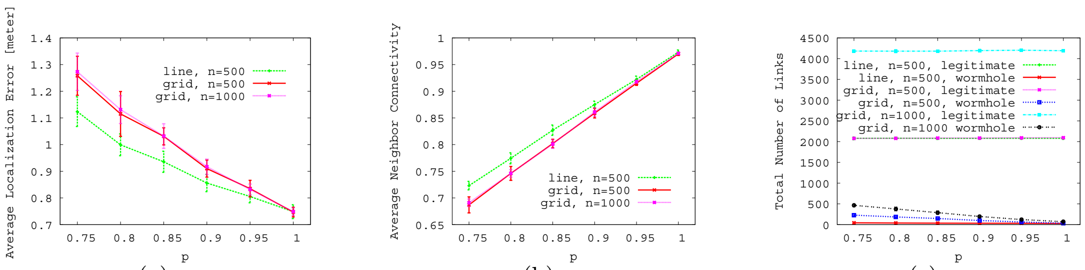
Ideally, if a sensor node can communicate with all of its physical neighbors using some communication key, the ratio of the number of its useful neighbors to the number of its physical neighbors is 1. In reality, since two physical neighbors may not necessarily share a communication key and the fact 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 the *average neighbor connectivity*: $\overline{N_c} = \frac{1}{n} \cdot \left(\sum_{i=1}^{n} \frac{\#\ of\ s_i\text{'s useful neighbors}}{\#\ of\ s_i\text{'s physical neighbors}} \right)$.

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 a potential wormhole attack. A small percentage suggests that the impact of a wormhole attack is not severe to the network.

To simulate real deployments, we adopt the parameters of VigilNet [5], a real WSN surveillance system. A 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 received by each sensor node during deployment. Let the localization error of the GPS module be uniformly distributed $\mathcal{U}$(-1.5,1.5) meters. 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 acceleration sensors. Let the rotation sensor error be uniformly distributed $\mathcal{U}$(-10,10) degrees, and the acceleration 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) meters. Let the regular communication range of each sensor node r be 30 meters.

Consider three typical deployment scenarios with a regular radio pattern: (1) A line deployment of 500 nodes where the horizontal spacing between sensor nodes is normally distributed $\mathcal{N}$(10,2) meters, and the vertical offset of each sensor node from the deployment line is normally distributed $\mathcal{N}$(0,2) meters. (2) A grid deployment of 500 nodes where the horizontal spacing between sensor nodes is normally distributed $\mathcal{N}$(10,2), and the vertical offset of each sensor node from each horizontal deployment line is normally distributed $\mathcal{N}$(0,2). (3) A grid deployment, similar to the second scenario, except that $n = 1000$. For each scenario, we evaluate its performance at p = 0.75, 0.80, 0.85, 0.90, 0.95, 1.00. For

[2] An evaluation of our scheme under the irregular radio pattern is provided in [11].



(a) (b) (c)

Figure 1: Simulation Performance with Regular Radio

each p, we performed 30 runs of simulations 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 Figure 1.

We observed that our scheme rendered consistent performance in all three scenarios. There is 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 neighbor connectivities roughly range between [0.7, 0.96], and they are an increasing function of p, reflecting the impact of localization errors on the key distribution decisions. Also, the number of potential wormhole links is quite low, compared to that of legitimate links in the same scenario (the ratio ranges from 2.5% to 10%), meaning that a wormhole attacker has a low chance of exploiting a potential wormhole link and creating an attack. Even if such an attack occurs, its impact would be small, due to the Distance Bounding Rule. In Figure 1, the error and connectivity curves corresponding to $n = 500$ and 1000 in grid deployment are quite close to each other. The total number of legitimate links and the total number of potential wormhole links increase proportionally with n, the size of the WSN. They indicate that our scheme is scalable for large-scale WSN deployments.

5. RELATED WORK

WSNs are inherently vulnerable to various attacks due to the insecure nature of wireless communication and the severe resource constraints on sensor nodes. As a result, determining node locations in a hostile environment is challenging. A lot of work has been done on secure localization for wireless sensor networks [2], [12], [7], [8], [13], [10], [14]. However, they either make strong assumptions about the deployments or require sophisticated and costly hardware support. Similarly, there is significant work on key distribution which is the basis for secure communication between legitimate nodes [4], [1], [9], [6]. They either are non-deterministic, or require the total number of nodes or nodes' locations be known in advance. Therefore, many of them are not practical for real WSN deployments.

6. CONCLUSION

In this paper, 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.

7. ACKNOWLEDGMENT

This work was supported, in part, by grants ARO W911NF-06-1-0204, and NSF CNS-0834299.

8. REFERENCES

[1] S. Camtepe and B. Yener. Combinatorial design of key distribution mechanisms for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 15(2), 2007.

[2] S. Capkun, M. Cagalj, and M. Srivastava. Securing localization with hidden and mobile base stations. In *INFOCOM*, 2006.

[3] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Trans. Inf. Theory*, 29(2), 1983.

[4] L. Eschenauer and V. Gligor. A key-management scheme for distributed sensor networks. In *CCS*, 2002.

[5] T. He, P. Vicaire, T. Yan, L. Luo, L. Gu, G. Zhou, R. Stoleru, Q. Cao, J. Stankovic, and T. Abdelzaher. Achieving real-time target tracking using wireless sensor networks. In *RTAS*, 2006.

[6] C. Kuo, M. Luk, R. Negi, and A. Perrig. Message-in-a-bottle: User-friendly and secure key deployment for sensor nodes. In *SenSys*, 2007.

[7] L. Lazos and R. Poovendran. Serloc: Secure range-independent localization for wireless sensor networks. *WiSe*, 2004.

[8] L. Lazos and R. Poovendran. Hirloc: High-resolution robust localization for wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 24(2), 2006.

[9] D. Liu and P. Ning. Location-based pairwise key establishments for static sensor networks. In *SASN*, 2003.

[10] D. Liu, P. Ning, and W. K. Du. Attack-resistant location estimation in sensor networks. In *IPSN*, 2005.

[11] Q. Mi, J. Stankovic, and R. Stoleru. Secure walking gps: A secure localization and key distribution scheme for wireless sensor networks. In *Technical Report*, http://www.cs.virginia.edu/∼qm8e/papers/swgps-full.pdf.

[12] T. Park and K. G. Shin. Attack-tolerant localization via iterative verification of locations in sensor networks. *ACM Trans. on Embedded Computing Systems*, 8(1), Dec 2008.

[13] R. Poovendran and L. Lazos. A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks. *Wireless Networks*, 13(1):27–59, January 2007.

[14] R. Shokri, M. Poturalski, G. Ravot, P. Papadimitratos, and J.-P. Hubaux. A practical secure neighbor verification protocol for wireless sensor networks. In *WiSec*, 2009.

[15] R. Stoleru, T. He, and J. Stankovic. Walking GPS: A practical solution for localization in manually deployed wireless sensor networks. In *LCN*, 2004.