ARTICLE IN PRESS

Available online at www.sciencedirect.com
ScienceDirect

Ad Hoc Networks xxx (2007) xxx–xxx

Ad Hoc Networks

www.elsevier.com/locate/adhoc

Secure probabilistic location verification in randomly deployed wireless sensor networks ☆

E. Ekici [a,*], S. Vural [a], J. McNair [b], D. Al-Abri [b]

[a] Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH, United States
[b] Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, United States

Received 24 August 2006; received in revised form 23 November 2006; accepted 27 November 2006

Abstract

Security plays an important role in the ability to deploy and retrieve trustworthy data from a wireless sensor network. Location verification is an effective defense against attacks which take advantage of a lack, or compromise, of location information. In this work, a secure probabilistic location verification method for randomly deployed dense sensor networks is proposed. The proposed *Probabilistic Location Verification* (PLV) algorithm leverages the probabilistic dependence of the number of hops a broadcast packet traverses to reach a destination and the Euclidean distance between the source and the destination. A small number of verifier nodes are used to determine the *plausibility* of the claimed location, which is represented by a real number between zero and one. Using the calculated plausibility metric, it is possible to create arbitrary number of trust levels in the location claimed. Simulation studies verify that the proposed solution provides high performance in face of various types of attacks.

Keywords: Wireless sensor networks; Localization; Location verification; Modeling; Trust

1. Introduction

Wireless sensor networks (WSNs) typically consist of a large number of inexpensive sensor devices that are both transmission and battery power-constrained, and that have limited computation and communication capabilities. However, the flexibility, fault tolerance, high sensing fidelity, low cost, and rapid deployment characteristics of sensor networks create many new and exciting applications for in situ sensing via WSNs. An important concern for various applications of WSNs is the ability to validate the integrity of the sensor network as well as the retrieved data. Various types of security attacks include: (1) the injection of false information into the regular data stream, (2) the alteration of routing paths due to malicious nodes advertising false positions (sink holes and worm holes), and (3) the forging of multiple identities by the same malicious node (Sybil nodes). Thus, location-based security plays an important role in the

☆ A preliminary version of this paper has appeared in the Proceedings of IEEE ICC 2006, Istanbul, Turkey.

* Corresponding author. Tel.: +1 614 292 0495; fax: +1 614 292 7596.

E-mail addresses: ekici@ece.osu.edu (E. Ekici), vurals@ece.osu.edu (S. Vural), mcnair@ece.ufl.edu (J. McNair), alabri@ufl.edu (D. Al-Abri).

1570-8705/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2006.11.006

trustworthiness of WSNs and the results that are obtained from them.

Although secure, point-to-point communication mechanisms can potentially prevent introduction of new adversary nodes into communication stream, it is likely that a compromised node can easily infiltrate such mechanisms. *Location Verification* emerges as a lightweight first line of defense, which makes sure that the information and its claimed source location are associated with a high level of trust. Information for which the source location cannot be verified is deemed not trustworthy and rejected to ensure the integrity of accepted data.

Over the past 5 years, researchers have developed many protocols for localization [1–3]. However, security in localization has been considered only recently [4–6]. In [4], a method is proposed that combines conventional multialteration with distance bounding for computation and verification of positions of wireless devices. However, devices must have a bounded processing time which may not be met by most existing hardware. The technique proposed in [5] uses directional antennae for secure positioning. Other techniques have been proposed using statistical methods, including [6], consistency among beacon signals, and voting schemes [7] to achieve robustness. Recent research also demonstrates that location verification can be combined with a non-secure localization scheme to produce a system that is more robust and resilient to attack than localization alone. The protocol presented in [8] verifies the presence of a node using radio frequency and sound. The hybrid system of [9] combines secure location computation with a location verification step that ensures a node cannot claim to be closer to a locator (reference node) than its actual distance. However, this approach relies on the existence of a secure localization scheme.

In this work, a probabilistic approach to location verification in dense sensor networks, *Probabilistic Location Verification* (PLV) algorithm, is proposed where nodes are deployed randomly. The proposed approach leverages the probabilistic dependence of the number of hops a broadcast packet traverses to reach a destination and the Euclidean distance between the source and the destination. A small number of *verifier nodes* calculate the likelihood that a broadcast packet that contains the geographic location of a node is received over a number of hops recorded in the packet. Observations of individual verifiers are combined to determine the *plausibility* of the location claim, which refers to the level of confidence that the claimed location results in the observed number of hops from the claimant source to all verifiers. The plausibility is represented by a real number between zero and one, which can simply be compared against a threshold to validate or invalidate the claimed location. The non-binary property of plausibility also enables the use of multiple levels of trust in the claimed location. The salient properties of our proposed PLV algorithm can be summarized as follows:

(1) Sensor nodes do not need to be equipped with specialized hardware.
(2) Only a small number of specialized verifiers are needed.
(3) The plausibility of a location claim is expressed as a real-number, not a hard binary decision.
(4) The PLV algorithm is resilient against a number of attacks and provides graceful degradation in performance.

The remainder of the paper is structured as follows: In Section 2, the WSN architecture and assumptions are outlined. In Section 3, a new set of probabilistic tools to verify a node's claimed location in presented, which is based on the comparison of the node's Euclidean distance with the hop count of the verification packet. The probabilistic location verification algorithm is outlined in Section 4 and the performance evaluation results are presented in the same section. In Section 5, the analysis of different types of attacks, solution methods, and associated performance evaluations are introduced. Finally, Section 6 concludes the paper with future research directions.

2. WSN architecture and assumptions

2.1. Network architecture

The sensor network architecture is shown in Fig. 1. The wireless sensors are deployed randomly in the network with a known density, which covers a number of scenarios ranging from battle field surveillance to observation of hazardous environments. We assume that the deployed sensor is of large scale and sensor locations follow a random Poisson point process, resulting in uniform node deployment. Each sensor node i determines its position (x_i, y_i) in a two-dimensional Euclidean coordinate system through a (non-secure) localization method such

Please cite this article in press as: E. Ekici et al., Secure probabilistic location verification in ..., Ad Hoc Netw. (2007), doi:10.1016/j.adhoc.2006.11.006

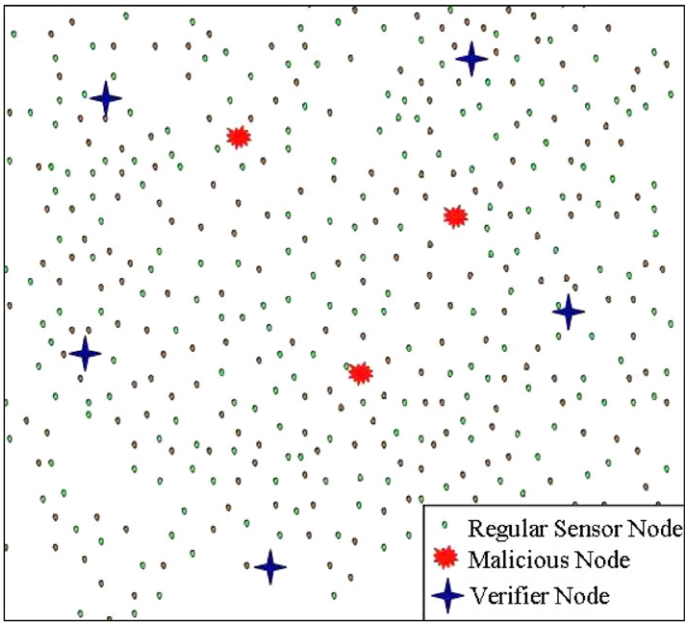

Fig. 1. Proposed WSN architecture.

as presented in [1–3]. It is assumed that all sensors have the same communication range and transmit at the same signal strength. It is worth noting that the communication range may change in an actual deployment even if the transmit power of all sensors is the same. However, the probabilistic estimation of distances used in this paper allows for graceful degradation in case of non-uniform communication range estimations. Furthermore, we also assume that there are no big gaps in the sensor deployment and no big obstacles are present that disturb the uniform distribution assumption. In case such obstacles exist, the performance of our proposed methods decrease.[1]

As shown in Fig. 1, we assume the presence of a small number of verifier nodes, which have the responsibility of verifying the location of the sensor nodes. Although the positions of the verifiers can be random, they must not be closely located to ensure accurate and independent observations. The verifiers know their locations. Location verification is performed either periodically to establish the trustworthiness of a specific node and its reported position, or aperiodically to verify the positional origin of a critical message. Each verifier is assumed to have sufficient computational ability to calculate its own likelihood function based on the packets received from a particular node as well as the overall plausibility value for a claim. The communication between verifiers is assumed to be reliable, and protected by encryption. For the purposes of this analysis, it is assumed that the verifiers are secure and cannot be compromised.

Also, as shown in Fig. 1, we assume the presence of a small number of malicious nodes. It is assumed that the malicious nodes possess the same properties as regular sensor nodes, i.e., the same processing power and the same communication hardware. In other words, a malicious node is assumed to be an equivalent version of a compromised sensor node.

2.2. Authentication of verification messages

We assume that the sensors are able to use two levels of authentication: one using symmetric keys, and another using asymmetric keys. The symmetric key is used to associate the given verification request with a sensor node identity. Every sensor node keeps a key that is used to encrypt the verification request. The key is then used at the verifier to decrypt the request and to map the request to a sensor node ID. Although a unique key for each node is desirable, the large number of sensor nodes makes the unique assignment unfeasible. Thus, a limited number of keys are randomly assigned to the sensor

[1] A similar case is discussed in Section 5.2 and Fig. 6 as related to denial of service attack mitigation.

Please cite this article in press as: E. Ekici et al., Secure probabilistic location verification in ..., Ad Hoc Netw. (2007), doi:10.1016/j.adhoc.2006.11.006

nodes before deployment, and the ID-key associations are stored in the verifiers. The use of the mapping between the keys and the sensor node IDs is described in detail in Section 5.

An asymmetric key is used by each node to help infer the hop count traversed from the length of the received verification packet. A low complexity asymmetric key system is assumed, such as TinyPK [10], where all sensors share the private key to encrypt data, but which they cannot use to decrypt. The public key is maintained only in the verifiers, which is used to decrypt the request packets. The inference of the hop count from the packet size is also described in detail in Section 5.

3. Probabilistic tools to verify location

The main idea behind the proposed mechanism is to leverage the statistical relationships between the number of hops in a sensor network and the Euclidean distance that is covered. The so-called hop-distance relationship has first been investigated in [11] for linear sensor networks and possible extensions to two-dimensional networks have been proposed. In the following sections, relevant outcomes are outlined.

3.1. The CDF of the k-hop distance

The analysis in [11] shows that the distance d_k covered in k hops in a linear network of node density λ and communication range R has a pdf that can successfully be approximated with a Gaussian distribution with the same average and standard deviation. As this analysis is derived for a one-dimensional network, we use a transformation of the two-dimensional network density to make it compatible with the derived results. For this purpose, we assume that there exists a "band" of width $\frac{R}{2}$ along the line connecting two points in the WSN, where the nodes can be assumed to be in a linear formation. Hence, the projected line density λ is calculated as $\lambda = \lambda' \frac{R}{2}$, where λ' is the two-dimensional density of the network. Based on this approximation, the average hop length $\bar{r}$ can be computed by solving the implicit equation

$$R - \bar{r} = \frac{1 - e^{-\lambda \bar{r}}(1 + \lambda \bar{r})}{\lambda(1 - e^{-\lambda \bar{r}})}. \quad (1)$$

Then, the expected value $\bar{r}_k \equiv E[d_k]$ of the k-hop distance d_k is computed simply by multiplying $\bar{r}$ by k:

$$\bar{r}_k \equiv E[d_k] = k \cdot \bar{r}. \quad (2)$$

The computation of the variance of the k-hop distance σ_k^2 follows an iterative formula:

$$\sigma_k^2 = f_2(k) - k^2 \bar{r}^2, \quad (3)$$

where

$$f_2(k) = f_2(k-1) + 2(k-1)\bar{r}^2 + E[r^2], \quad (4)$$

$$f_2(2) = 2E[r^2] + 2\bar{r}^2, \quad (5)$$

$$E[r^2] = -R^2 + 2R\bar{r} + E[r_e^2], \text{ and} \quad (6)$$

$$E[r_e^2] = \frac{-\bar{r}^2 e^{-\lambda \bar{r}} - \frac{2}{\lambda}\bar{r}e^{-\lambda \bar{r}} + \frac{2}{\lambda^2}(1 - e^{-\lambda \bar{r}})}{1 - e^{-\lambda \bar{r}}}, \quad (7)$$

where r_e is defined as $r_e \equiv R - r$ for the first hop. With these statistical measures, the cdf of the k-hop distance d_k can be approximated as follows:

$$Pr\{d_k < d | K = k\} = \int_{-\infty}^{d} \frac{1}{\sigma_k \sqrt{2\pi}} e^{-\frac{(\delta - \bar{r}_k)^2}{2\sigma_k^2}} d\delta = \frac{1}{2}\left[1 + \text{erf}\left(\frac{d - \bar{r}_k}{\sigma_k \sqrt{2}}\right)\right], \quad (8)$$

where K is the random variable representing the number of hops taken, and $\bar{r}_k$ and σ_k are as defined in Eqs. (2) and (3). Obviously, a packet cannot traverse more than $k \cdot R$ in any direction in k hops, and the approximation needs to be upper-bounded in range. Note that the width of the band to calculate λ can be varied according to the density. Our additional analysis has shown that a band width of $\frac{\lambda}{2}$ leads to a successful linear network approximation for two dimensional densities as low as 3×10^{-2} nodes/m^2.

3.2. The PMF of the number of hops k

Consider the case where a node i broadcasts its location (x_i, y_i) in a packet that is flooded in the network. Let us assume that the packet is received in k_v^* hops by a verifier node v located at (x_v, y_v). We would like to know the probability that a packet originating at (x_i, y_i) traverses k hops to be received by v at (x_v, y_v). The conditional CDF given in Eq. (8) can be used to calculate the probability that a message is relayed in k hops to traverse a distance of $d = \sqrt{(x_v - x_i)^2 + (y_v - y_i)^2}$. For this purpose, we define an error margin ϵ used to compute finite probabilities for the hop distances. We use the Baye's Theorem applied on Eq. (8) to compute the PMF of the hop number K conditioned on the distance d:

$$Pr\{K=k|d-\epsilon<d_k\leqslant d+\epsilon\} = \frac{Pr\{d-\epsilon<d_k\leqslant d+\epsilon|K=k\}\cdot Pr\{K=k\}}{Pr\{d-\epsilon<d_k\leqslant d+\epsilon\}}. = \frac{\frac{1}{2}\left[\mathrm{erf}\left(\frac{d+\epsilon-\bar{r}_k}{\sigma_k\sqrt{2}}\right)-\mathrm{erf}\left(\frac{d-\epsilon-\bar{r}_k}{\sigma_k\sqrt{2}}\right)\right]Pr\{K=k\}}{Pr\{d-\epsilon<d_k\leqslant d+\epsilon\}}. \quad (9)$$

In Eq. (9), the two unconditional probabilities must be computed based on the location of the verifier (x_v,y_v), the shape of the sensor field $\mathscr{A}$, the node density λ (and hence, λ'), and the average hop distance $\bar{r}$. The unconditional probability $Pr\{d-\epsilon<d_k\leqslant d+\epsilon\}$ is simply the ratio of the number of nodes in a ring of radius d and thickness 2ϵ around the verifier node v to the total number of nodes. If v is at least $d+\epsilon$ away from all edges of the sensor field, this probability can be computed as follows:

$$Pr\{d-\epsilon<d_k\leqslant d+\epsilon\} = \frac{\lambda'\pi((d+\epsilon)^2-(d-\epsilon)^2)}{\mathscr{N}}, \quad (10)$$

where $\mathscr{N}$ is the total number of nodes in the sensor field. Obviously, if the ring around the verifier node v is not completely contained in the sensor field, the numerator of the fraction should be computed such that only the segments of the ring contained in the sensor field are accounted for. Similarly, if the verifier node v is at the origin of a sensor field $\mathscr{A}$, then the probability that a node is k hops away from v is computed as follows:

$$Pr\{K=k\} = \frac{((k+1)^2-k^2)\cdot\pi\bar{r}^2}{\int\int_{(\theta,r)\in\mathscr{A}}\left\lceil\frac{r}{\bar{r}}\right\rceil r\,\mathrm{d}r\,\mathrm{d}\theta}, \quad (11)$$

where $\bar{r}$ is given in Eq. (1), and (θ,r) corresponds to the polar coordinates of a location inside the sensor field $\mathscr{A}$. Note that the unconditional probability of Eq. (11) is independent of the density of the network. For finite size sensor networks, these quantities can be calculated before deployment numerically considering the intersection of the rings around the verifier nodes and the sensor field. Moreover, $Pr\{K=k|d-\epsilon<d_k\leqslant d+\epsilon\}$ values can also be computed for all values of k and small increments of d offline and stored as tables in verifier nodes. The online computation burden of the verifiers can be minimized by using these tabulated values.

3.3. Relating probabilities with plausibility

After the verifier node v receives the location information (x_i,y_i) of node i, v can compute the conditional probability mass function $Pr\{K=k|d-\epsilon<d_k\leqslant d+\epsilon\}$ for the number of hops needed to cover the distance d. This, along with the actual hop distance covered k_v^*, is used to determine how much a verifier can contribute to the overall decision process. Let us assume that a verifier v computes the distance d from a source claiming to be at (x_i,y_i) based on the information contained in a broadcast packet. Let us also assume that the non-zero probabilities of the PMF of Eq. (9) are $\{0.2,0.3,0.4,0.1\}$ for hop counts $\{4,5,6,7\}$, respectively. The most likely number of hops the packet must have taken is 6 according to the PMF calculation. However, if a packet reaches the verifier in $k^*=5$ hops, the verifier should not declare the claimed location implausible. Furthermore, the relative position of the probability associated with k^* in the entire PMF should also be taken into account.

To this end, we consider the difference between the maximum value in the PMF and the probability associated with k^*: The larger this difference becomes, the less one should trust the claimed location. On the other hand, if this difference is small, the verifier should not be alarmed regardless of the k^* value. Theoretically there are an infinite number of non-zero probabilities for this PMF. However, for the sake of simplicity, we also ignore the cases that have a very small probability associated with them.

Let $P_v^{\max}(d)$ be the maximum probability computed for any number of hops based on (x_i,y_i) and v's location:

$$P_v^{\max}(d) = \max_{n\in N} Pr\{K=n|d-\epsilon<d_k\leqslant d+\epsilon\}, \quad (12)$$

where N is the set of natural numbers. We also define a *probability slack* function $S_v(d,k_v^*)$ which is the difference between the maximum probability the verifier v can provide and the probability of the source being k_v^* hops away:

$$S_v(d,k_v^*) = P_v^{\max} - Pr\{K=k_v^*|d-\epsilon<d_k\leqslant d+\epsilon\}. \quad (13)$$

Scaling $S_v(d,k_v^*)$ by $P_v^{\max}(d)$, i.e., $\frac{S_v(d,k_v^*)}{P_v^{\max}(d)}$, one obtains the *distrust* in the claimed location based on the observed number of hops.

An important observation at this point should be made regarding the distrust levels of individual verifiers. Let us consider two verifiers that calculate PMFs, one resulting in a very "peaked" distribution (say, $\{0.3,0.6,0.1\}$), and the other in a more uniform distribution (say, $\{0.1,0.2,0.2,0.2,0.2,0.1\}$).

Please cite this article in press as: E. Ekici et al., Secure probabilistic location verification in ..., Ad Hoc Netw. (2007), doi:10.1016/j.adhoc.2006.11.006

Let the first verifier compute a distrust value of $\frac{0.6-0.3}{0.6} = 0.5$, and the other verifier compute $\frac{0.2-0.1}{0.2} = 0.5$. Intuitively, one can claim that the second verifier can only make a very uncertain decision because of the shape of the distribution. On the other hand, the first verifier has a "stronger" opinion, be it supporting or against the acceptance of the claimed location. Hence, the second verifier's input should be weighed less than the input of the first verifier. We propose to use $P_v^{\max}(d)$ as a measure of the confidence in a verifier's opinion. Although there exist many other ways to express the level of confidence, such as using a function of the PMF variance, weighing the distrust levels with $P_v^{\max}(d)$ both provides a good measure (as observed through simulations) and simplifies the plausibility calculations. If there are $\mathscr{V}$ verifiers participating in the verification process, then the overall plausibility $\mathscr{P}_i$ of node i's location claim can be computed as

$$\mathscr{P}_i = 1 - \frac{\sum_{j=1}^{\mathscr{V}} \frac{P_j^{\max} - Pr\{K=k_j^* | d-\epsilon < d_k \leqslant d+\epsilon\}}{P_j^{\max}} \cdot P_j^{\max}}{\sum_{j=1}^{\mathscr{V}} P_j^{\max}} = 1 - \frac{\sum_{j=1}^{\mathscr{V}} S_j(d, k_j^*)}{\sum_{j=1}^{\mathscr{V}} P_j^{\max}}. \quad (14)$$

Hence, verifiers only need to exchange $S_v(d, k_v^*)$ and $P_v^{\max}$ values to compute the plausibility $\mathscr{P}_i$.

4. Probabilistic location verification

4.1. The basic probabilistic location verification (PLV) algorithm

The location verification is initiated by a claimant node and only involves broadcasting the location information of the claimant node throughout the network. The broadcast packets should contain the hop count in addition to the claimed location information. The hop count is used for both verification procedures and also during broadcasting: A node receiving a broadcast packet re-broadcasts the packet only if it has the lowest hop count of the same information received so far.

It is assumed that there are $\mathscr{V}$ verifier nodes located at sufficiently distant locations in the network such that the individual observations are assumed independent of each other. The main steps of the proposed algorithm are outlined below:

(1) A node i broadcasts its location (x_i, y_i) in the network using flooding. Every packet must contain the hop count as well as the claimed location information.
(2) Each of the $\mathscr{V}$ verifiers receive the message over k_v^* hops and compute their relative distance d_v.
(3) Using k_v^* and d_v, each verifier node v computes its probability slack $S_v(d, k_v^*)$ and maximum probability $P_v^{\max}(d)$ values.
(4) $S_v(d, k_v^*)$ and $P_v^{\max}(d)$ values of all $\mathscr{V}$ verifiers are collected at a central node, e.g., a designated verifier node, and a common plausibility $\mathscr{P}_i$ for the location advertisement is computed.
(5) P_i value is compared with a set of classification thresholds to assess the trustworthiness of the claimed location. In its simplest form, a single threshold level leads to a decision about acceptance or rejection of this location association.

As outlined above, the PLV algorithm can be used to verify (or reject) a location claim. When a node claims to be at an arbitrary point in the network (x, y), the claim is evaluated by the verifiers, which forward their information to the central node, which then calculates the plausibility of the node's reported location. To illustrate this, we formed a network of 1000 nodes of communication radius $R = 10$ m randomly placed on a $100\,\text{m} \times 100\,\text{m}$ field. The verifiers are placed at four corners, each 5 m away from the closest edges. Fig. 2a–d shows the overall plausibility $\mathscr{P}_i$ (Eq. (14)) of a location claim (x, y) originating from a node i at $(76.1, 33.8)$ (marked with a green line protruding the surface) in the presence of one, two, three, and four verifiers, respectively. As shown in Fig. 2a, a single verifier can accept a significant portion of the location space as plausible. When two verifiers are used, the possibility of finding a unique neighborhood is not guaranteed as shown in Fig. 2b since two "rings" can intersect in two regions. To guarantee a single neighborhood of highest plausibility, at least three verifiers are required as shown in Fig. 2c. Fig. 2d shows the plausibility when four verifiers are used, which results in a small plateau of plausibility value 1 around the source. Hence, a higher number of verifiers can detect a false location claim with a higher probability.

4.2. Performance of the basic PLV algorithm

The performance of the basic PLV algorithm is assessed through simulations. The results presented

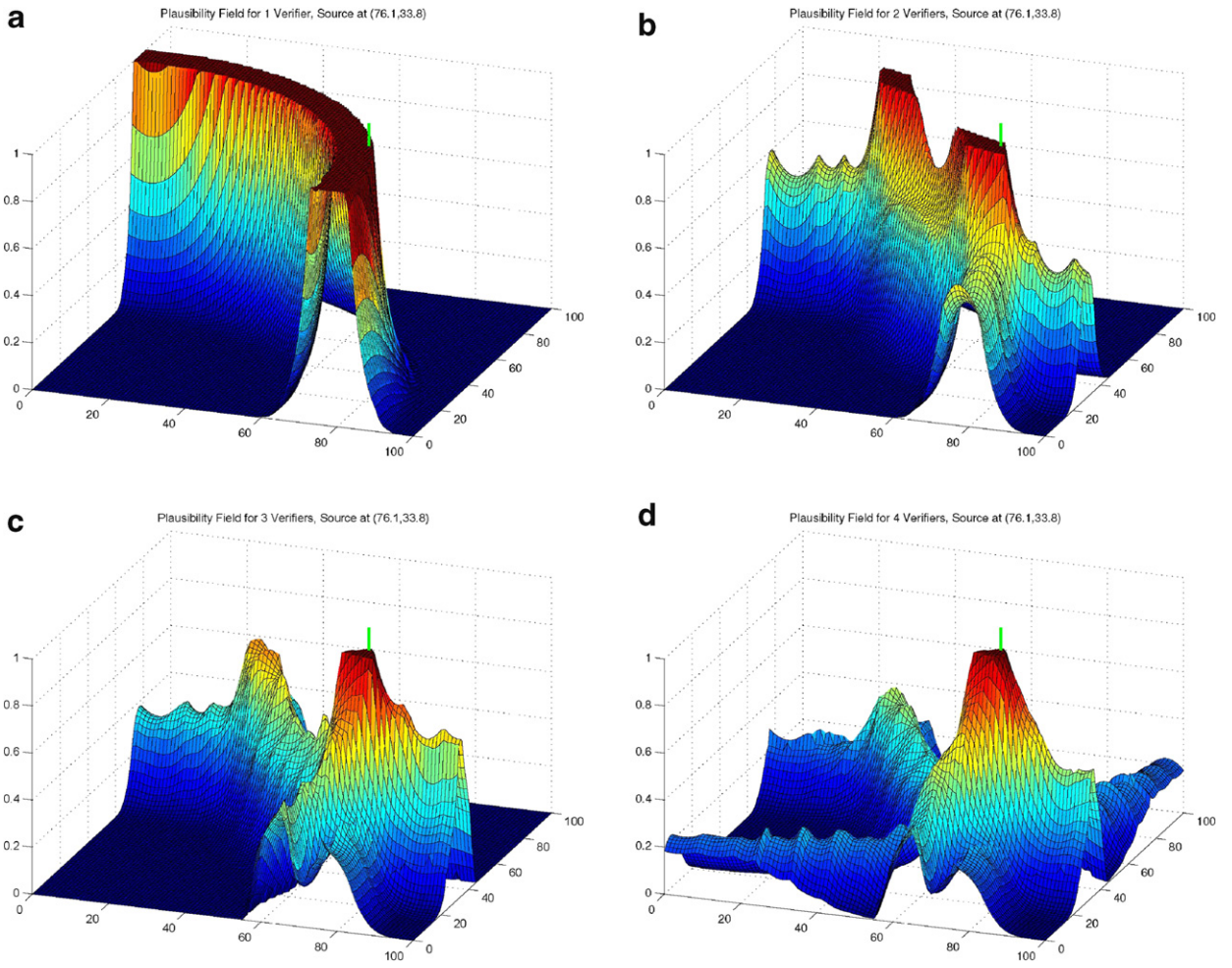

Fig. 2. Plausibility of a node's claimed location for different number of verifiers (a) one verifier, (b) two verifiers, (c) three verifiers, (d) four verifiers.

in this section are the averages of 1000 samples taken from 50 independent random sensor networks. Unless otherwise stated, each of the random networks is composed of 1000 nodes of $R = 10$ m randomly distributed over a $100\,\text{m} \times 100\,\text{m}$ area. A claim location is considered true if it lies within $R/4$ distance of the actual location. We assume a binary decision system where PLV either rejects or accepts a claim by comparing plausibility with a threshold value.

4.2.1. Effect of number of verifiers

Fig. 3 shows the probability of detection as a function of the probability of false alarm for 1–4 verifiers. This plot is also referred to as the *Receiver Operating Curve* (ROC) and indicates the success of a classification method independent of the threshold values. ROCs are generated by sweeping the range of the classification threshold, in our case 0–1 in increments of 0.01. A good classifier provides high detection probability for very small values of false alarm probability, i.e., it is pushed more towards the coordinates $(0, 1)$. Another indication of the quality of the classifier is the area under the ROC. The larger the area under the curve, the better is the discrimination of invalid location claims. As expected, the performance for four verifiers is the strongest, while one verifier is the weakest with respect to both detection and false alarms. As discussed in Section 4.1, at least three verifiers are required to obtain a unique and small plateau of plausibility. Additional verifiers further improve the classification performance.

Please cite this article in press as: E. Ekici et al., Secure probabilistic location verification in ..., Ad Hoc Netw. (2007), doi:10.1016/j.adhoc.2006.11.006

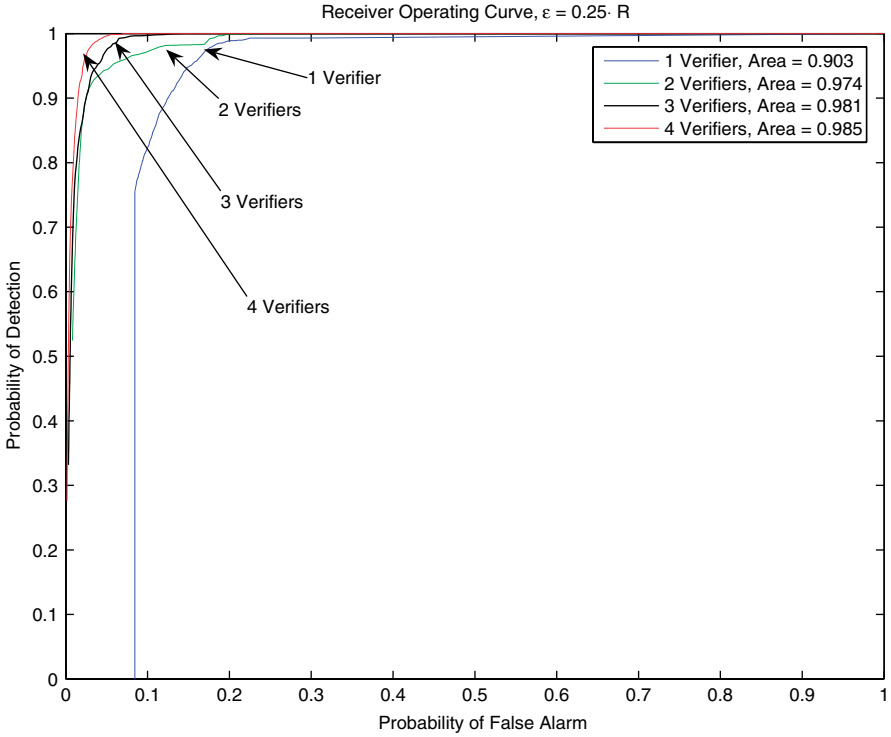

Fig. 3. Receiver operating curve.

4.2.2. Effect of node density

In Fig. 4, the effect of the node density on the classification performance is depicted. The performance curves, which reflect the area under the ROC, show that a high number of verifiers consistently result in higher classification accuracy. Furthermore, as the number of verifiers increases, the effect of the density on the performance decreases. The curve for four verifiers is almost flat while the curve for one verifier shows a clear maximum in the given range. This observation supports the theory that a higher number of verifiers lends robustness to changes in the network characteristics. The performance degradation of one verifier system for high node densities is aligned with the observations reported in [11]: As the node density increases, the accuracy of the Gaussian approximation of the distance covered in k hops decreases. Since there are no other verifiers, errors made during classification become more pronounced.

4.2.3. Effect of verifier separation

Since the PLV mechanism is based on the estimation of hop distances, the estimations of individual verifiers should be as independent as possible. For this purpose, we have proposed in Section 4.1 to place them far apart from each other. We have also run simulations to investigate the effects of the verifier separation on the classification performance. The verifiers are located at the corners of a square with edge lengths ranging between 10 and 90 m and centered at the center of the sensor network. For cases with less than four verifiers, we according number of verifiers were removed from the system. The results shown in Fig. 5 indicate that three and four verifiers reach a steady state performance at 50 m separation. The two verifier performance improves as the separation increases since the distance between the two verifiers increase and the potential for two intersection areas within the sensor network. The same observation can also be made for the single verifier case: Although there is no other verifier in the network, the increasing edge length of the constellation pushes the verifier towards a corner, which reduces the area of the ring inside the sensor network for which the plausibility level is high. Therefore, the performance of the algorithm increases.

5. Security analysis and PLV improvements

In addition to claiming a false location, several other types of attacks can be launched against a

Please cite this article in press as: E. Ekici et al., Secure probabilistic location verification in ..., Ad Hoc Netw. (2007), doi:10.1016/j.adhoc.2006.11.006

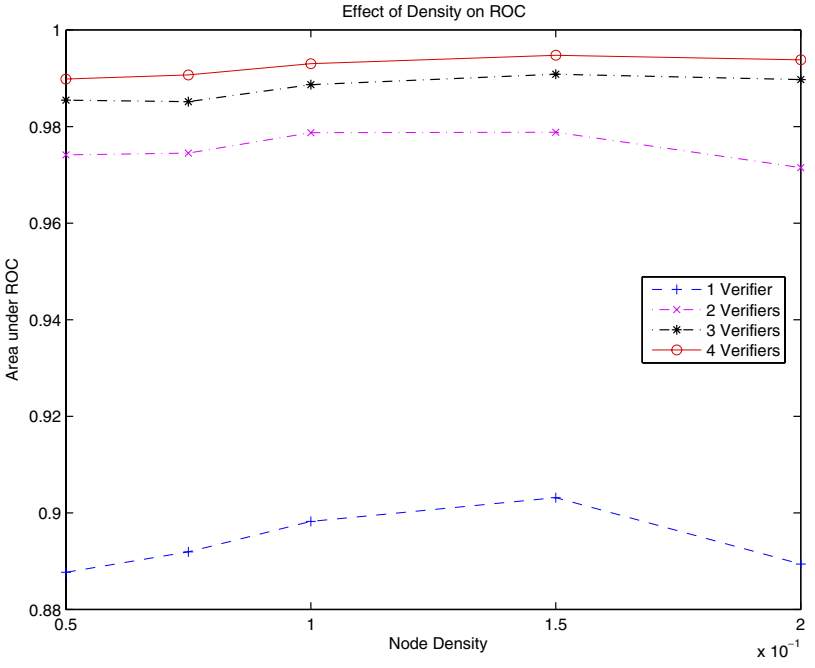

Fig. 4. Effect of node density.

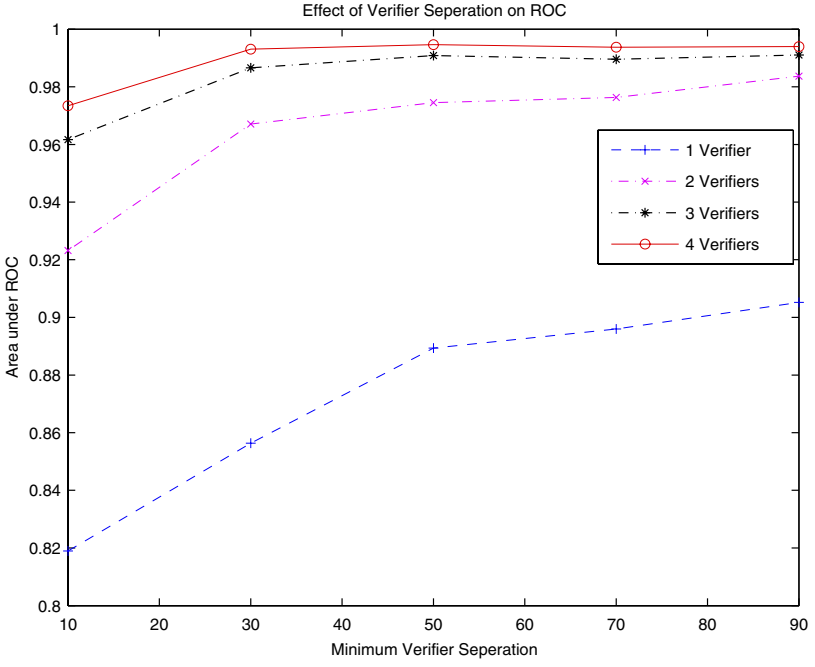

Fig. 5. Effect of minimum verifier separation.

sensor network employing the basic PLV algorithm. Most of these attacks are based on exploiting vulnerabilities of intermediate communication steps. In the following sections, some of these attacks and possible solutions are discussed. The accompanying simulation results are generated

Please cite this article in press as: E. Ekici et al., Secure probabilistic location verification in ..., Ad Hoc Netw. (2007), doi:10.1016/j.adhoc.2006.11.006

based on the same parameters as described in Section 4.2.

5.1. Disreputation through impersonation

5.1.1. Problem description

A malicious node m located at (x_m, y_m) can try to invalidate the trustworthiness of another node i located at (x_i, y_i) by sending a location verification message containing i's identity and some random location information. If the attack succeeds, the verifiers can reject the location claim based on the low plausibility score and blacklist the victim node i. To prevent this attack, node identities must be verified through security mechanisms. As an unencrypted portion of a message is always prone to modification attacks along the way, we need to encrypt the identity along with the location claim.

5.1.2. Proposed solution

A straight-forward way would be to use a symmetric key system, where every sensor node is associated with a unique key. The same key would also be kept at verifiers along with their associated node IDs. To establish the correctness of the identity, verifiers would have to try to decrypt the message with all possible keys (since node IDs are encrypted, as well) and check the matching of the key with the ID contained in a message. Since the number of nodes in a WSN is very high, the use of unique keys is not feasible. Instead, a limited number of private keys N_K, where $N_K \ll \mathcal{N}$, can be randomly assigned to sensors before deployment, and ID-key associations are stored in the verifiers. A verifier receiving an encrypted message would only need to try the valid N_K keys to match the encrypted ID with the key used. With this strategy, a malicious node m would succeed to disrepute a node i with a probability of $1/N_K$, in which case the i and m would have the same key. A high N_K value would yield better performance, but also increase the computational load on verifiers.

5.2. Denial of service through payload alterations

5.2.1. Problem description

Another major type of attack can be performed by altering the payload of the broadcast claim packets to make the verifiers ignore the received messages on the grounds of authenticity. Let a source node i send the claim packet and a malicious node m alter its content during broadcasting. The altered packet would reach a verifier node v only if m lies on a shortest path between i and v. In other cases, the packet m forwards would be suppressed in the network because an intermediate forwarder would prefer a packet with smaller hop count. If the verifiers are distributed far from each other and if there are sufficient number of redundant verifiers, then the effect of such an attack would be minimal. A malicious node m would have the most adverse effect on the claimant nodes in its close neighborhood since m may lie in that case on a number of shortest paths to the verifiers.

5.2.2. Proposed solution 1

If an intermediate node j receives inconsistent copies of a packet, it may infer that there is a malicious node m in its near vicinity. In such a case, a straight-forward precaution is to refrain from forwarding any information. To accomplish this, nodes should wait for a predetermined time period and collect copies of the same claim packet before attempting to forward them. This approach would create "holes" in the network around the malicious nodes where no information is forwarded.

Fig. 6 demonstrates the resilience of our PLV algorithm in the presence of denial of service (DoS) attacks. The PLV algorithm is modified such that nodes refrain from forwarding information as they discover inconsistencies, creating "holes" in the forwarding grid where verification packets are absorbed. Fig. 6 clearly shows the performance degradation as the number of malicious nodes increases. The vulnerability of one- and two-verifier cases is clearly visible, as their performance decreases sharply when the number of malicious nodes increases beyond 3. On the other hand, the three- and four-verifier cases show higher resilience and their performance starts decreasing sharply only at 12 malicious nodes.

5.2.3. Proposed solution 2

It is also possible to identify and suppress nodes that consistently alter packets before forwarding. A similar method has also been proposed in [12] to identify and penalize selfish nodes in an ad hoc network. Let a malicious node m receive a packet from an intermediate node j, and alter its content before forwarding. A third node l, which lies in the transmission range of both j and m, can identify the alteration and record this event. If the occurrence of such alterations exceed a particular threshold, m can be marked by l (and possibly by its other neigh-

Please cite this article in press as: E. Ekici et al., Secure probabilistic location verification in ..., Ad Hoc Netw. (2007), doi:10.1016/j.adhoc.2006.11.006

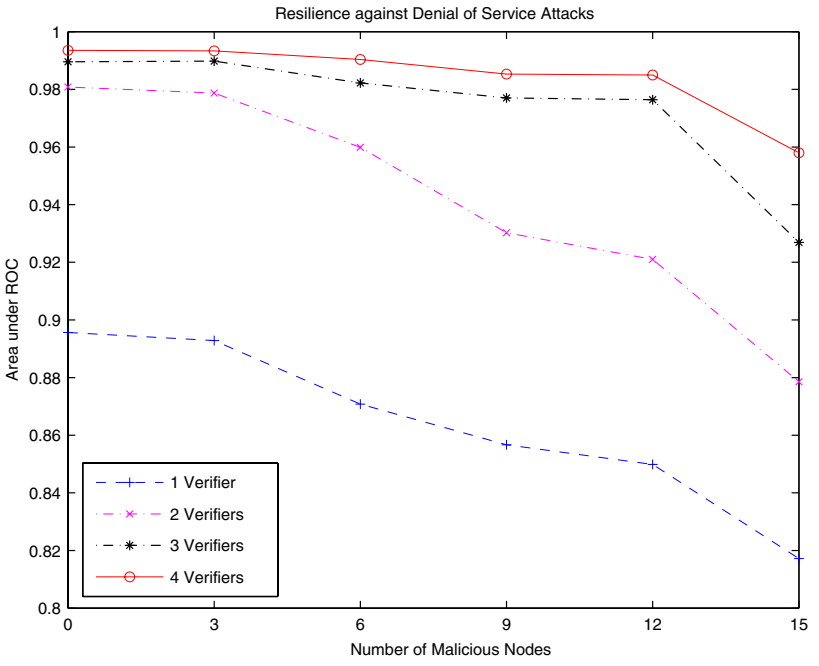

Fig. 6. DoS attacks with payload alterations.

bors) as malicious. Consequently, all neighbors that locate malicious nodes would ignore packets originating from m. The effect of such an exclusion would be the reduction of the node density. As shown in Fig. 4, fluctuations in density is well tolerated by three or more verifier PLV systems. Hence, if possible, malicious node identification should be preferred over refraining from forwarding inconsistent packets.

5.3. Denial of service through hop count alterations

5.3.1. Problem description

During broadcasting, the hop count of a packet must be increased every time it is forwarded to compute the hop distance between a claimant and the verifiers. Since every node should be able to change the hop count, it constitutes a weak spot in the verification system. A malicious node m can easily replace the hop count with a very small number, e.g., 1, without changing the payload. In that case, m would act as a new source and cause a false estimation of the hop distance, which leads to low plausibility values and blacklisting of the claimant node. The effect of such an attack is shown in Fig. 7 assuming the use of the basic PLV algorithm with no countermeasures. The area under ROC for this type of attack drops very rapidly with the increasing number of malicious nodes. Considering a value of 0.5 refers to a completely random decision about plausibility (either accept or reject without regard to the received information), the severity of hop count reset attacks becomes more pronounced.

5.3.2. Proposed solution

To prevent hop count reset attacks, we propose to infer the hop count from the length of the packet rather than a field contained in the header. We assume that there is a low complexity asymmetric key system such as TinyPK [10] where all sensors share the private key K_1 to encrypt data, but which they cannot use to decrypt. The public key K_2 is maintained only in the verifiers, which is used to decrypt data. Let an intermediate node j receive a packet P. Node j forwards the packet after appending a fixed string X to P and encrypting $X + P$ using K_1. Given an encrypted packet, the hop count can be inferred from its packet length if the length of the original message is known. To determine if two packets P_1 and P_2 are copies of each other, the following procedure is used: The hop counts k_1 and k_2 associated with P_1 and P_2 are inferred from the packet sizes. Let $k_1 < k_2$. Then, P_1 is encrypted $k_2 - k_1$ more times, each time after appending X. If the resulting packet is the same as

Please cite this article in press as: E. Ekici et al., Secure probabilistic location verification in ..., Ad Hoc Netw. (2007), doi:10.1016/j.adhoc.2006.11.006

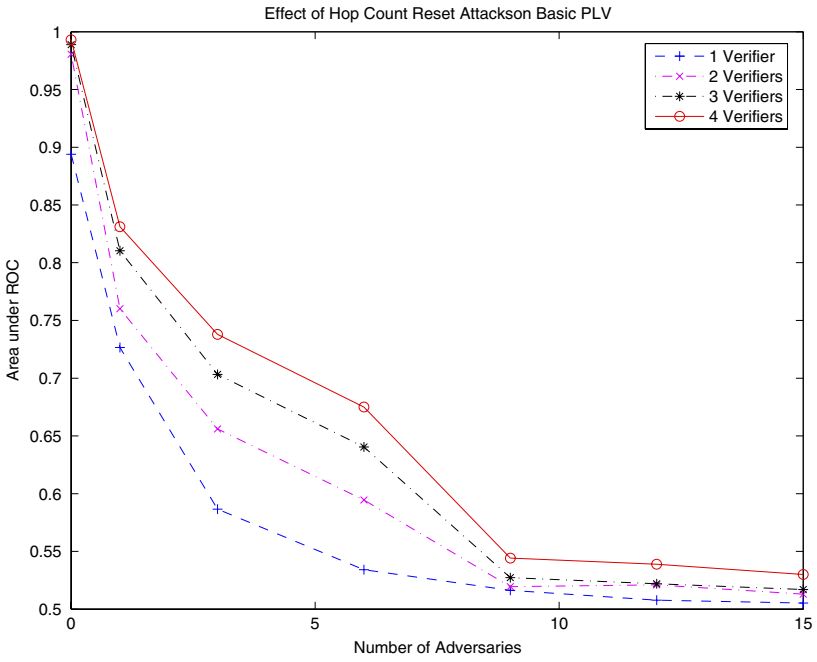

Fig. 7. Effect of hop count reset attacks.

P_2, then P_1 is forwarded instead of P_2. Otherwise, j concludes that P_1 and P_2 are different broadcast packets. When verifiers receive a packet, they use K_2 to decrypt the information.

Using this method, every node can only increase the hop count by appending X and encrypting it, but *cannot decrease the hop count* while preserving the integrity of payload. Increasing the hop count of a packet more than once is still possible. However, such packets are generally absorbed in the network since smaller hop count packets would exist in the WSN with high probability. The effect of hop count increment attacks are shown in Fig. 8, where the malicious nodes increase the hop count by one. The simulation results show that the PLV algorithm can absorb the hop count increment attacks quite successfully. We also observe that the overall performance is better for higher density networks ($\lambda = 0.1$) and the variations are higher for the lower density networks ($\lambda = 0.05$). In additional simulations, we also observed that the PLV performance increases if malicious nodes choose to increase the hop count by more than one. The increase in performance is due to elimination of tempered verification packets during broadcasting. As a final note, malicious nodes that increase the hop counts can also be identified and isolated using third party observations as described in Section 5.2.

5.4. Denial of service through packet replay

5.4.1. Problem description

Yet another denial of service attack can be launched by malicious node by re-inserting cached verification packets into the network. As a result, the innocent claimant node whose identity was included in the verification packet would appear as launching a denial of service (resource exhaustion) attack, and would be blacklisted by the verifiers. Furthermore, such attacks would indeed consume valuable energy resources in the network, whether the original source is blacklisted or not.

5.4.2. Proposed solution

To mitigate the problem of blacklisting, we propose using the well-known sequence number method. The source can include a sequence number in the verification request packet, which is encrypted along with the node ID and the location information. Consequently, verifiers can easily identify whether or not a location verification is a genuine one or a repeated copy of an old claim. The verifiers discard the packets they discover to be repeated. This can also be used as an indication of the presence of malicious nodes in the network. To counter the problem of inherent resource exhaustion for this attack, we can resort to the method of local identi-

Please cite this article in press as: E. Ekici et al., Secure probabilistic location verification in ..., Ad Hoc Netw. (2007), doi:10.1016/j.adhoc.2006.11.006

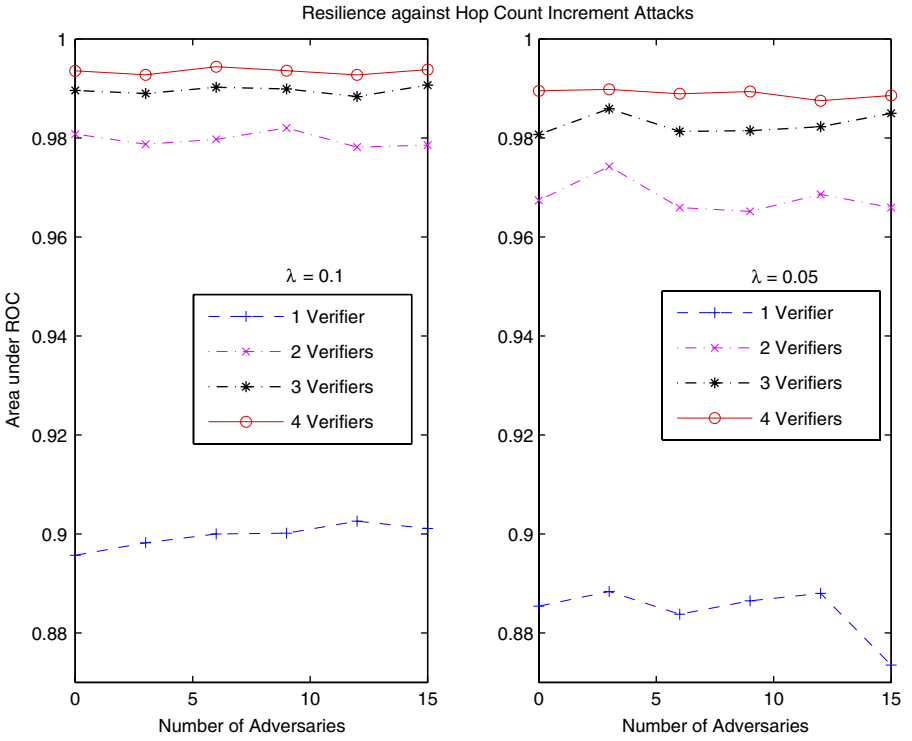

Fig. 8. Resilience against hop count increment attacks.

fication: When the neighbors of the malicious node detects a copy of the same packet multiple times, they can locally blacklist the malicious node and ignore packet sent by it. The sequence number also solves the problem of colluding malicious nodes in the network, where the original packet is relayed to a distant node and replayed at its new location after the original packet has been already disseminated.

5.5. Wormhole attacks

5.5.1. Problem description

In this work, we assume that malicious nodes do not possess more resources than the regular sensor nodes. Therefore, the wormhole attacks can be launched only through tunneling packets between colluding nodes. Once a malicious node m_i receives a verification packet, it simply tunnels the packet through a wormhole to other malicious nodes m_j. The verification packet is inserted into the network by m_j without alteration. This packet would suppress all other copies of the same claim at all verifiers to which m_j is closer than m_i. Consequently, the plausibility score of the claimant node would be calculated incorrectly.

5.5.2. Proposed solution

The wormhole attacks are generally very hard to counter, and this one is not an exception. In the literature, there are several proposals to identify wormholes in a wireless sensor network [13–15]. These methods can be directly used to identify and avoid wormholes (given the sensor network satisfies associated requirements in hardware). In addition to these methods, we propose a two-step approach to address the wormhole attacks using the PLV tools. In the first step, the verifiers need to identify the existence (not the locations) of wormholes in the network. The presence of wormholes can be identified if there is a consistent discrepancy between probability slack values for messages originating from a region. Alternatively, the verifiers may also check for wormholes periodically. The second step deals with the identification of wormholes. To this end, a verifier v_i acts as the source of the verification packets. Note that the verifiers are trustworthy and know their location and their

Please cite this article in press as: E. Ekici et al., Secure probabilistic location verification in ..., Ad Hoc Netw. (2007), doi:10.1016/j.adhoc.2006.11.006

hop mutual hop distances accurately. Whenever a verifier v_j receives a verification packet from the source verifier v_i, v_j can detect any inconsistencies in the hop count. These inconsistencies are interpreted as results of wormhole attacks. Furthermore, the location of one end of the tunnel can be identified by tracing the hop count gradient in reverse. We note that these methods should be further analyzed in detail and defer this task to our future work.

6. Conclusions

In this work, the Probabilistic Location Verification (PLV) algorithm for dense sensor networks is presented. Assuming that the compromised nodes make up a small percentage of the total sensor node population, a small number of *verifier nodes* was used to conclude whether or not the claimed location is a plausible one. It is assumed that the average density of the sensor nodes in the sensing field and the communication range of sensor nodes are known. Using a new set of probabilistic tools, PLV compares the node's Euclidean distance with the hop count of the verification packet. Simulation results confirm the accuracy and effectiveness of this light-weight location verification system. In our future work, the PLV algorithm will be improved to with built-in measures against wormhole attacks and cooperative attacks of multiple malicious nodes. New methods to ensure the hop count and content integrity will also be investigated to reduce the computational burden on sensor and verifier nodes.

References

[1] D. Niculescu, B. Nath, Ad hoc positioning system (aps) using aoa, in: Proc. IEEE INFOCOM, 2003.
[2] T. He, C. Huang, B. Blum, J. Stankovic, T. Abdelzaher, Range-free localization schemes for large scale sensor networks, in: Proc. Mobicom, 2003.
[3] L. Fang, W. Du, P. Ning, A beacon-less location discovery scheme for wireless sensor networks, in: Proc. IEEE INFOCOM, 2005.
[4] S. Capkun, J.-P. Hubaux, Secure positioning of wireless devices with application to sensor networks, in: Proc. IEEE INFOCOM, 2005.
[5] L. Lazos, R. Poovendran, Serloc: secure range-independent localization for wireless sensor networks, in: ACM Workshop on Wireless Security (ACM WiSe), 2004.
[6] Z. Li, W. Trappe, Y. Zhang, B. Nath, Robust statistical methods for securing wireless localization in sensor networks, in: Proc. IPSN, 2005.
[7] D. Liu, P. Ning, W. Du, Attack-resistant location estimation in sensor networks, in: Proc. IPSN, 2005.
[8] N. Sastry, U. Shankar, D. Wagner, Secure verification of location claims, in: ACM Workshop on Wireless Security (ACM WiSe), 2003.
[9] L. Lazos, R. Poovendran, S. Capkun, Rope: Robust position estimation in wireless sensor networks, in: Proc. IPSN, 2005.
[10] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, P. Kruus, TinyPK: Securing sensor networks with public key technology, in: Proc. of ACM Workshop on Security of Ad Hoc and Sensor Networks, 2004, pp. 59–64.
[11] S. Vural, E. Ekici, Analysis of hop-distance relationship in spatially random sensor networks, in: Proc. ACM MobiHoc, 2005, pp. 320–331.
[12] Q. He, D. Wu, P. Khosla, Sori: A secure and objective reputation-based incentive scheme for ad-hoc networks, in: Proc. IEEE WCNC, 2004.
[13] L. Lazos, R. Poovendran, C. Meadows, P. Syverson, L. Chang, Preventing wormhole attacks on wireless ad hoc networks: a graph theoretic approach, in: IEEE Wireless and Communications and Networking Conference (WCNC), 2005.
[14] A. Pirzada, C. McDonald, Circumventing sinkholes and wormholes in wireless sensor networks, in: International Conference on Wireless Ad Hoc Networks (IWWAN), 2005.
[15] E. Ngai, J. Liu, M. Lyu, On the intruder detection for sinkhole attack in wireless sensor networks, in: IEEE International Conference on Communication (ICC), 2006.

Eylem Ekici is an Assistant Professor of Electrical and Computer Engineering at the Ohio State University. He received his Ph.D. degree in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta, GA, in 2002. His current research interests include wireless sensor networks, vehicular communication systems, and next generation wireless networks, with a focus on routing and medium access control protocols, resource management, and analysis of network architectures and protocols. He also conducts research on interfacing of dissimilar networks.

Serdar Vural is pursuing his Ph.D. degree in Electrical and Computer Engineering at the Ohio State University. He received his B.S. degree in Electrical Engineering from Bogazici University, Istanbul, Turkey, in 2003 and his M.S. degree in Electrical and Computer Engineering from the Ohio State University in 2005. He his currently pursuing his Ph.D. degree in Electrical and Computer Engineering at the Ohio State University. His research interests include modeling and development of routing protocols for wireless sensor networks.

Please cite this article in press as: E. Ekici et al., Secure probabilistic location verification in ..., Ad Hoc Netw. (2007), doi:10.1016/j.adhoc.2006.11.006

Janise McNair earned her Ph.D. in Electrical and Computer Engineering from the Georgia Institute of Technology in 2000. Currently, she is an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Florida, where she leads the Wireless and Mobile Systems Laboratory. Her current research interests are integrated wireless infrastructured and ad hoc networks, wireless sensor networks, and next generation wireless networks.

Dawood Al-Abri received his B.S. degree in Electrical and Electronics Engineering from Sultan Qaboos University, Oman in 1999 and a Master of Science degree in Electrical and Computer Engineering from the University of Florida in 2002. He is currently pursuing a Ph.D. in Electrical and Computer Engineering at the University of Florida. His research focus is security and location verification for wireless sensor networks.

Please cite this article in press as: E. Ekici et al., Secure probabilistic location verification in ..., Ad Hoc Netw. (2007), doi:10.1016/j.adhoc.2006.11.006