A Probabilistic Approach to Location Verification in Wireless Sensor Networks

Eylem Ekici
Department of Electrical and Computer Engineering
Ohio State University, Columbus, OH 43210
Email: ekici@ece.osu.edu

Janise McNair and Dawood Al-Abri
Department of Electrical and Computer Engineering
University of Florida, Gainesville, FL 32611
Email: mcnair@ece.ufl.edu, alabri@ufl.edu

***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 first line of defense against attacks which take advantage of a lack, or compromise, of location information. In this work, a probabilistic approach to location verification in 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 source and destination. A small number of verifier nodes 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**

I. Introduction

Wireless sensor networks (WSNs) typically consist of a large number of simple and inexpensive sensor devices equipped with wireless communication interfaces. An important concern for numerous applications of WSNs is the ability to validate the integrity of the network and 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 trustworthiness of WSNs and obtained results.

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 infiltrates such mechanisms. *Location Verification* emerges as a lightweight first line of defense, which ensures 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 five years, researchers have developed many protocols for localization [1], [2], [3]. However, researchers have just recently begun to address the issue of security in localization [4], [5], [6]. In [4], a technique is proposed that combines conventional multilateration with distance bounding for computation and verification of sensor positions. However, sensors must have a bounded processing time which may not be met by most existing hardware. In [5], a secure positioning technique using directional antennae is proposed. Several techniques have been proposed using statistical methods [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. In [8], a protocol is described that verifies the presence of a node using radio frequency and sound. In [9], a hybrid system is proposed that 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 requires a secure localization scheme.

In this work, a probabilistic approach to location verification in dense and random WSNs, *Probabilistic Location Verification* (PLV) algorithm, is proposed. PLV 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, a number between zero and one. It is the level of confidence that the claimed location results in the observed number of hops from the claimant source to all verifiers. The plausibility can be compared against a threshold to validate 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.

1-4244-0355-3/06/$20.00 (c) 2006 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

II. WSN Architecture and Assumptions

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. 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 as presented in [1], [2], [3]. It is assumed that all sensors have the same communication range and transmit at the same signal strength. In addition, we assume that the sensors are able to use an asymmetric key protocol to encrypt information.

We assume the presence of a small number of malicious nodes and a small number of verifiers responsible for verifying the location of sensors. Although the positions of the verifiers can be random, they must not be closely located to ensure accurate and independent observations. The verifiers should know their exact locations, as well. The location verification is performed either periodically to establish the trustworthiness of a specific node and its position, or aperiodically to verify the positional origin of a critical message. Each verifier has enough 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 verifiers are also assumed to communicate among themselves through encrypted messages using symmetric keys. Hence, the communication between verifiers are assumed to be reliable. For the purposes of this analysis, it is assumed that the verifiers are secure and cannot be compromised. It is also assumed that the malicious nodes possess the same properties as regular sensor nodes, i.e., they have the same processing power and same communication hardware. In other words, a malicious node is assumed to be an equivalent version of a compromised sensor node.

The location verification is always 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 only re-broadcasts a packet if it has the lowest hop count of the same information received so far. To protect the integrity of the packets, we only assume simple encryption capabilities of sensor nodes.

III. 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 [10] for linear sensor networks and possible extensions to two-dimensional networks have been proposed. In the following sections, relevant outcomes are outlined.

A. The CDF of the k-Hop Distance

The analysis in [10] 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[1]. Hence, the 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 1:

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

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:

$$\begin{aligned}
\sigma_k^2 &= f_2(k) - k^2 \bar{r}^2, \text{ where} && (3)\\
f_2(k) &= f_2(k-1) + 2(k-1)\bar{r}^2 + E[r^2], && (4)\\
f_2(2) &= 2E[r^2] + 2\bar{r}^2, && (5)\\
E[r^2] &= -R^2 + 2R\bar{r} + E[r_e^2], \text{ and} && (6)\\
E[r_e^2] &= \frac{-\bar{r}^2 e^{-\lambda \bar{r}} - \frac{2}{\lambda}\bar{r}e^{-\lambda \bar{r}} + \frac{2}{\lambda^2}\left(1 - e^{-\lambda \bar{r}}\right)}{1 - e^{-\lambda \bar{r}}}, && (7)
\end{aligned}$$

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:

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

where K is the random variable representing the number of hops taken, and $\bar{r}_k$ and σ_k are as defined in Equations 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.

B. 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^* 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 Equation 8 can be used to calculate the probability that a message is relayed in k hops to traverse a distance of $d = \sqrt{(x_v - x_i)^2 + (x_v - x_i)^2}$. For this purpose, we define an

[1]The width of the band can be varied according to the density. Our additional analysis has shown that this band width leads to a successful linear network approximation for two dimensional densities as low as $3 \cdot 10^{-4}$ nodes/m^2.

error margin ϵ used to compute finite probabilities for the hop distances. We use the Baye's Theorem applied on Equation 8 to compute the PMF of the hop number K conditioned on the distance d:

$$\begin{aligned} &Pr\{K=k \mid d-\epsilon < d_k \leq d+\epsilon\} \\ &= \frac{Pr\{d-\epsilon < d_k \leq d+\epsilon \mid K=k\} \cdot Pr\{K=k\}}{Pr\{d-\epsilon < d_k \leq d+\epsilon\}} \\ &= \frac{\frac{1}{2}\left[erf\left(\frac{d+\epsilon-\bar{r}_k}{\sigma_k\sqrt{2}}\right) - erf\left(\frac{d-\epsilon-\bar{r}_k}{\sigma_k\sqrt{2}}\right)\right] Pr\{K=k\}}{Pr\{d-\epsilon < d_k \leq d+\epsilon\}}. \end{aligned} \quad (9)$$

In Equation 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 $\mathcal{A}$, the node density λ (and hence, λ'), and the average hop distance $\bar{r}$. The unconditional probability $Pr\{d-\epsilon < d_k \leq d+\epsilon\}$ 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:

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

where $\mathcal{N}$ 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 $\mathcal{A}$, then the probability that a node is k hops away from v is computed as follows:

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

where $\bar{r}$ is given in Equation 1, and (θ, r) corresponds to the polar coordinates of a location inside the sensor field $\mathcal{A}$. Note that the unconditional probability of Equation 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, $Pr\{K=k \mid d-\epsilon < d_k \leq d+\epsilon\}$ 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.

C. 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 $Pr\{K=k|d-\epsilon < d_k \leq d+\epsilon\}$ 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[2] of the PMF of Equation 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. 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:

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

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:

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

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, $\{0.1,\ 0.2,\ 0.2,\ 0.2,\ 0.2,\ 0.1\}$). 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 $\mathcal{V}$ verifiers participating in the verification process, then the overall plausibility $\mathcal{P}_i$ of node i's location claim can be

[2]Theoretically, there are infinite number of nonzero probabilities for this PMF. However, for the sake of simplicity, we choose to ignore the cases that have a very small probability associated with them.

computed as:

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

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

IV. The Proposed Probabilistic Location Verification (PLV) Algorithm

A. The Basic PLV Algorithm

It is assumed that there are $\mathcal{V}$ verifier nodes located at uncorrelated locations in the networks 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. Packets contain the hop count, as well.
2) Each of the $\mathcal{V}$ verifiers receive the message over k_v^* hops and compute their relative distance d_v.
3) Using k_v^* and d_v, each 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 $\mathcal{V}$ verifiers are collected at a central node, e.g., a designated verifier node, and a common plausibility $\mathcal{P}_i$ 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. A single threshold level leads to a decision about acceptance or rejection of this location association.

The PLV algorithm can be used to verify (or reject) a location claim. When a node claims to be at an arbitrary point (x, y), the claim is evaluated by verifiers. Evaluations are set to the central node, which then calculates the plausibility of the claim. To illustrate this, we formed a network of 1000 nodes of communication radius $R = 10m$ randomly placed on a $100m \times 100m$ field. The verifiers are placed at four corners, each $5m$ away from the closest edges. Figures 1(a), 1(b), and 1(c) show the overall plausibility $\mathcal{P}_i$ (Equation 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, and four verifiers, respectively. As shown in Figure 1(a), 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 Figure 1(b) since two "rings" can intersect in two regions. To guarantee a single neighborhood of highest plausibility, at least three verifiers are required. Figure 1(c) 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.

In the following a set of attacks that can be launched against the PLV algorithm and possible solutions are discussed.

B. Disreputation through Impersonation

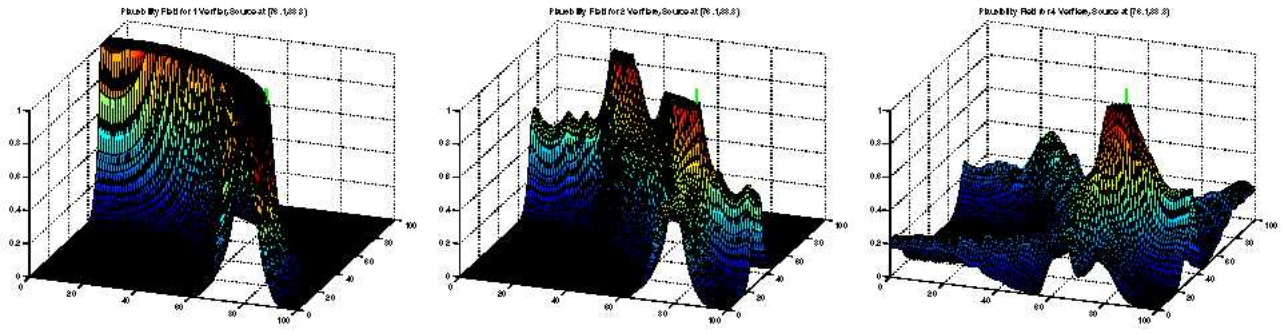
A malicious node m located at (x_m, y_m) can try to invalidate the trustworthiness of another node i located at (x_i, y_i) by sending a location verification message containing i's identity and some random location information. If the attack succeeds, the verifiers can reject the location claim based 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.

A straight-forward way would be to use an symmetric key system, where every sensor node is associated with a unique key, which are also available at verifiers. 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 is high, the use of unique keys is not feasible. Instead, a random key distribution with limited number of keys N_K, where $N_K \ll \mathcal{N}$, can be used, and ID-key associations are stored in 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 private key. A high N_K value would yield better performance, but also increase the computational load on verifiers.

C. Denial of Service through Payload Alterations

Another major type of attack is performed by altering the payload of the broadcast claim packets to make the verifiers ignore the received messages on the grounds of authenticity. Let the source node i send the claim packet and the 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.

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. Such an approach would create "holes" in the network around the malicious nodes where no information is forwarded. However, it is also possible to identify and suppress nodes that consistently alter packets before forwarding. A similar method has also been proposed in [11] to identify and



(a) One Verifier (b) Two Verifiers (c) Four Verifiers

Fig. 1. Plausibility of a Node's Claimed Location for Different Number of Verifiers

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 neighbors) as malicious. Consequently, all neighbors that locate malicious nodes would ignore packets originating from them. The effect of such an exclusion would be reduction of the node density, which may not effect the overall performance based on the assumption that there are only a small number of malicious nodes in the system.

D. Denial of Service through Hop Count Alterations

During broadcasting, the hop count of a packet must be increased every time it is forwarded to compute the hop distance. 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 and blacklisting of the claimant.

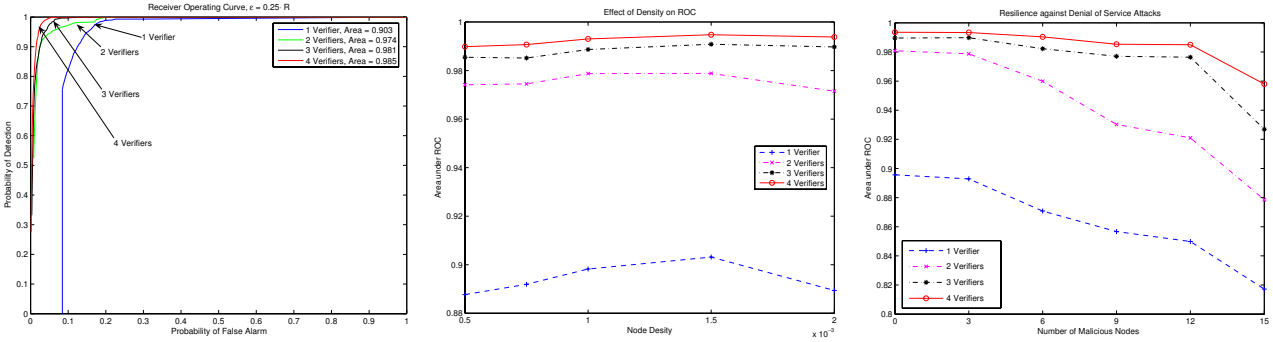
To prevent such an attack, we propose to infer the hop count from the length of the packet rather than a field contained in the header. We assume that there is a low complexity asymmetric key system such as [12] where all sensors share the private key K_1 to encrypt data. The public key K_2 is maintained only in the verifiers and 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 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, hop count can be increased by appending X and encrypting the packet, but cannot be decreased while preserving payload integrity. 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. Furthermore, such nodes can also be identified and isolated using third party observations as described in Section IV-C.

V. Performance Evaluation

The performance of PLV is assessed through simulations. The results presented 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 $R = 10m$ randomly distributed over a $100m \times 100m$ 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. We evaluate the effect of the number of verifiers on the classification performance, the effect of the node density on the classification performance, and the effect of denial of service (DoS) attacks countered with "no forwarding" response.

Figure 2(a) shows the probability of detection as a function of the probability of false alarm for 1 through 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 through 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 $(0, 1)$, or the area under the ROC is close to one. As expected, the performance for four verifiers is the strongest, while one verifier is the weakest with respect to both detection and false alarms. At least three verifiers are required to obtain a unique



(a) Receiver Operating Curve (b) Effect of Node Density (c) Resilience Against DoS Attacks

Fig. 2. Simulation Results for the PLV Algorithm

and small plateau of plausibility. Additional verifiers further improve the classification performance.

In Figure 2(b), 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, the effect of the density on the performance decreases with increasing number of verifiers. 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. The performance degradation of one verifier system for high node densities is aligned with the observations reported in [10]: 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.

Figure 2(c) demonstrates the resilience of PLV in the presence of denial of service (DoS) attacks. We assume that nodes refrain from forwarding information as they discover inconsistencies, creating "holes" in the forwarding grid where verification packets are absorbed. Figure 2(c) 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. If regular nodes can identify and isolate the malicious nodes, then the malicious nodes would effective be removed from consideration, only reducing the local node densities. As shown in Figure 2(b), 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.

VI. Conclusion

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 counter 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 and B. Nath, "Ad hoc positioning system (aps) using aoa," in *Proc. IEEE INFOCOM*, 2003.

[2] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proc. Mobicom*, 2003.

[3] L. Fang, W. Du, and P. Ning, "A beacon-less location discovery scheme for wireless wensor networks," in *Proc. IEEE INFOCOM*, 2005.

[4] S. Capkun and J.-P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *Proc. IEEE INFOCOM*, 2005.

[5] L. Lazos and 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, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *Proc. IPSN*, 2005.

[7] D. Liu, P. Ning, and W. Du, "Attack-resistant location estimation in sensor networks," in *Proc. IPSN*, 2005.

[8] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *ACM Workshop on Wireless Security (ACM WiSe)*, 2003.

[9] L. Lazos, R. Poovendran, and S. Capkun, "Rope: Robust position estimation in wireless sensor networks," in *Proc. IPSN*, 2005.

[10] S. Vural and E. Ekici, "Analysis of Hop-Distance Relationship in Spatially Random Sensor Networks," *Proc. ACM MobiHoc*, pp. 320–331, May 2005.

[11] Q. He, D. Wu, and P. Khosla, "Sori: A secure and objective reputation-based incentive scheme for ad-hoc networks," in *Proc. IEEE WCNC*, 2004.

[12] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: Securing Sensor Networks with Public Key Technology," *Proc. of ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 59–64, Oct. 2004.