Secure Distance-based Localization in the Presence of Cheating Beacon Nodes

Murtuza Jadliwala, *Member, IEEE,* Sheng Zhong, *Member, IEEE,* Shambhu Upadhyaya, *Senior Member, IEEE,* Chunming Qiao, *Senior Member, IEEE* and Jean-Pierre Hubaux, *Fellow, IEEE*

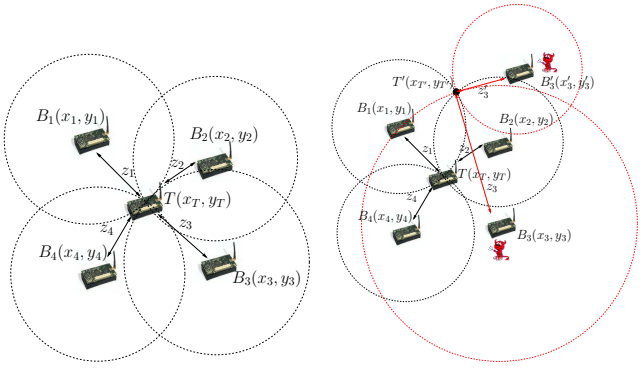
Abstract—Secure distance-based localization in the presence of cheating beacon (or anchor) nodes is an important problem in mobile wireless ad hoc and sensor networks. Despite significant research efforts in this direction, some fundamental questions still remain unaddressed: In the presence of cheating beacon nodes, what are the necessary and sufficient conditions to guarantee a bounded error during a two-dimensional distance-based location estimation? Under these necessary and sufficient conditions, what class of localization algorithms can provide this error bound? In this paper, we attempt to answer these and other related questions by following a careful analytical approach. Specifically, we first show that when the number of cheating beacon nodes is greater than or equal to a given threshold, there do not exist any two-dimensional distance-based localization algorithms that can guarantee a bounded error. Furthermore, when the number of cheating beacons is below this threshold, we identify a class of distance-based localization algorithms that can always guarantee a bounded localization error. Finally, we outline three novel distance-based localization algorithms that belong to this class of bounded error localization algorithms. We verify their accuracy and efficiency by means of extensive simulation experiments using both simple and practical distance estimation error models.

Index Terms—Wireless networks, distance-based localization, security.

1 Introduction

Localization or location discovery in distributed wireless networks is the problem of determining the location, with respect to some local or global coordinate system, of a (mobile) device in the network in an efficient and accurate fashion. Distributed localization protocols in such networks can be broadly classified into *range-based* and *range-free* techniques [1]. Range-based techniques can be further classified into two broad categories, viz., (a) *Beacon-based* techniques and (b) *Beacon-free* techniques. In this work, we focus primarily on beacon-based localization algorithms. Beacon-based algorithms such as [2], [3], [4], [5], [6], [7], [8], [9] require the presence of special nodes, called *beacon* or *anchor* nodes, which know their own location and are strategically placed in the network. Other nodes first compute the distance (or angle) estimates to a set of neighboring beacons and then estimate their own location using basic trilateration (or triangulation). The working of a two-dimensional beacon-based localization scheme using distance estimates to neighboring beacons is shown in Figure 1(a).

In Figure 1(a), nodes B_1, B_2, B_3 and B_4 located at positions (x_1, y_1), (x_2, y_2), (x_3, y_3) and (x_4, y_4), respectively, act as beacon nodes. The target node T estimates distances z_1, z_2, z_3 and z_4, respectively, to these beacon nodes and computes its own location (x_T, y_T) by trilateration. Efficient techniques for estimating distances such as *Received Signal Strength Indicator (RSSI)* [10], *Time of Arrival (ToA)* [11], and *Time Difference of Arrival (TDoA)* [12] exist and have been successfully used in the various beacon-based localization protocols listed above. Although beacon-based techniques are very popular in most wireless systems, they have one shortcoming. Most beacon-based techniques in the literature assume that the nodes acting as beacons always behave honestly. It is not surprising that beacon-based methods perform well when all the beacon nodes are honest. But their accuracy suffers considerably in the presence of malicious or cheating beacon nodes. Beacons can cheat by broadcast-



(a) (b)

Fig. 1. Distance-based (range-based) localization (a) Trilateration (b) Cheating beacons

- *Murtuza Jadliwala and Jean-Pierre Hubaux are with the Laboratory for computer Communications and Applications (LCA1) at the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, CH-1015, Switzerland. E-mail: {murtuza.jadliwala,jean-pierre.hubaux}@epfl.ch.*
- *Sheng Zhong, Shambhu Upadhyaya and Chunming Qiao are with the Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260, USA. E-mail: {szhong,shambhu,qiao}@cse.buffalo.edu.* *Sheng Zhong was supported in part by NSF CNS-0845149.*

ing their own locations inaccurately or by manipulating the distance estimation process, thus adversely affecting the location computation by the other nodes. This is depicted in Figure 1(b). In this figure, we can see that beacon nodes B_1, B_2 and B_4 behave honestly, whereas beacons B_3 and B_3' cheat. This causes the target node T to compute its location incorrectly as $(x_{T'}, y_{T'})$ instead of (x_T, y_T).

Earlier research efforts in securing distance-based localization techniques have focused on either removing this (over)dependence on beacon nodes ([13], [14], [15]) or on minimizing the effects of malicious beacons ([16], [17]) during localization. But before delving into the possible solutions for secure localization, we feel that there is a need to address the following questions that have been ignored by earlier research efforts: Under what condition(s) do there exist algorithms that can overcome the cheating effect of malicious beacons? How do we determine these algorithms when these condition(s) are satisfied, if at all? What kind of guarantee on the solution quality (in terms of bounds on the error in localization) can such algorithms provide? None of the research efforts undertaken previously provide an answer to all these questions. Eren et al. [18] study the problem of distance-based localization from a theoretical standpoint and provide conditions for unique network localization using graph rigidity theory, but their results assume non-cheating beacon nodes. What has been missing in the literature is a comprehensive theoretical framework for studying the hardness and feasibility of the distance-based localization problem in the presence of cheating beacons. A systematic analytical study would not only help in designing efficient algorithms to solve this problem, but would also help in deriving performance bounds guaranteed by these algorithms, thus facilitating an effective comparative analysis. In this paper, we attempt to fill this gap between theory and practice by first establishing the necessary and sufficient conditions for the problem of secure distance-based localization in the presence of cheating beacon nodes and then outlining a class of algorithms that can always guarantee a bounded localization error.

Specifically, we make the following contributions. First, we prove that if the number of malicious beacons is greater than or equal to $\frac{n-2}{2}$, where n is the total number of beacons providing distance information, then no algorithm can guarantee a bounded localization error for all cases. In other words, as long as the above inequality holds, any distance-based algorithm will fail to estimate the target location within a small error bound for at least one scenario or set-up of beacons. Next, we show that there exist algorithms that provide a guaranteed degree of localization accuracy (for all the cases), if the number of malicious beacons is less than or equal to $\frac{n-3}{2}$. These two inequalities are also referred to as the *necessary and sufficient conditions* for robust localization. Given the above conditions, we define a class of distance-based localization algorithms that can always localize with a bounded error. We transition from theory to practice by proposing three illustrative algorithms that belong to this class of robust distance-based algorithms. The first algorithm, called the *Polynomial Time* algorithm, uses an exhaustive search strategy to provide good localization accuracy with a polynomial (cubic) run-time complexity (in terms of the number of available beacons) in the worst case. But in practice, the Polynomial Time algorithm runs very inefficiently. To overcome this problem, we propose two other algorithms. These algorithms use simple heuristics to securely compute locations and have a much better execution efficiency. Finally, we verify the performance of these algorithms through extensive simulation experiments and present a detailed comparative analysis based on the simulation results. We also extend the existing localization framework to include more practical distance estimation error models and also study their effect on the accuracy of the proposed localization algorithms.

The rest of the paper is organized as follows. In Section 2, we provide some background on secure localization and discuss the related work, and in Section 3 we present the network and adversary model. In Section 4, we derive the conditions for secure distance-based localization and define the class of bounded error distance-based localization algorithms. In Section 5, we propose three algorithms that belong to this class and in Section 6 we discuss their simulation results. In Section 7, we extend the existing localization framework to include more practical distance estimation error models. We conclude the paper with a summary of contributions and some directions for future research in Section 8.

2 Background and Related Work

In this section, we survey some earlier research efforts towards securing distance-based localization schemes. Most of the prior works in this area have followed one of the following two themes – (1) detection and elimination of cheating nodes, or (2) localization in the presence of cheating nodes and large errors.

2.1 Malicious Node Detection and Elimination

One approach followed by researchers to secure distance-based localization approaches is to detect the cheating nodes and eliminate them from consideration during the localization process. Liu et al. [17] propose a method for securing beacon-based localization by eliminating malicious data. This technique, called *attack-resistant Minimum Mean Square Estimation (MMSE)*, takes advantage of the fact that malicious location references introduced by cheating beacons are usually inconsistent with the benign ones. Similarly, the *Echo* location verification protocol proposed by Sastry et al. [19] can securely verify location claims by computing the relative distance between a prover and a verifier node using the time of propagation of ultrasound signals. Čapkun et al. [20] shortlist various attacks related to node localization

in wireless sensor networks and propose mechanisms such as authenticated distance estimation, authenticated distance bounding, verifiable trilateration and verifiable time difference of arrival, to secure localization. Pires et al. [21] propose protocols to detect malicious nodes in distance-based localization approaches by detecting message transmissions whose signal strength is incompatible with its originator's geographical position. In another similar work by Liu et al. [22], the authors propose techniques to detect malicious beacon nodes by employing special *detector nodes*.

2.2 Robust Localization Schemes

The second approach towards securing localization is to design techniques that are robust enough to tolerate the cheating effect of malicious nodes (or beacons), rather than explicitly detecting and eliminating them. Priyantha et al. [4] propose the *CRICKET* system that eliminates the dependence on beacon nodes by using communication hops to estimate the network's global layout, and then apply force-based relaxation to optimize this layout. Some other research attempts also try to solve the secure localization problem by formulating it as a global optimization problem. For example, Li et al. [16] develop robust statistical methods such as *adaptive least squares* and *least median squares* to make beacon-based localization attack-tolerant. Alternatively, Doherty et al. [23] address the problem of beacon-based localization in the presence of large range measurement errors, and describe a localization method using connectivity constraints and convex optimization. Moore et al. [24] formulate the localization problem in wireless sensor networks as a *two-dimensional graph realization problem* and describe a beaconless (anchor-free), distributed, linear-time algorithm for localizing nodes in the presence of large range measurement noise. Liu et al. [17] design an intelligent strategy, called *voting-based scheme*, where the deployment area is divided into a grid of cells such that the target node resides in one of the cells. Every beacon node votes on each cell depending on the distance between the target node and itself and the location of the target node is estimated as being within the cell that had the maximum number of beacon votes. In another approach, Yi et al. [25] and Ji et al. [14] apply efficient data analysis techniques such as *Multi-Dimensional Scaling (MDS)* using connectivity information and distances between neighboring nodes to infer target locations. Fang et al. [15] model the localization problem as a statistical estimation problem. The authors use *Maximum Likelihood Estimation (MLE)* in order to estimate the most probable node location, given a set of neighborhood observations. Recently, ideas from coding theory have also been applied to achieve robust localization, for example [26], [27]. In another work, Lazos et al. [28] propose a range independent distributed localization algorithm using sectored antennas, called *SeRLoc*, that does not require any communication among nodes. However, SeRLoc is based on the assumption that jamming of the wireless medium is not feasible. To overcome this problem, Lazos et al. [29] also present a hybrid approach, called RObust Position Estimation (ROPE), which unlike *SeRLoc*, provides robust location computation and verification without centralized management and vulnerability to jamming. In another recent research effort by Misra et al. [30], the authors propose a convex optimization based scheme to secure the distance-based localization process, which uses Barrier's method to solve the optimization problem.

2.3 Discussion

Malicious node detection and elimination strategies, as discussed in Section 2.1, take into account the inconsistency (caused by cheating behavior) in measurement of a particular network parameter in order to detect cheating nodes. One shortcoming of such an approach is the requirement that verifier nodes have to be completely honest. Moreover, these solutions do not provide any fixed guarantees of the number of detected cheating beacon nodes or the accuracy of the ensuing localization algorithms. Any undetected cheating beacon node will only add to the error of the localization algorithm.

On the contrary, a majority of the localization schemes discussed in Section 2.2 attempt to improve the robustness of the localization procedure by employing optimization techniques. The main focus of these schemes is to minimize the effect of inconsistent or erroneous data on the overall localization accuracy. Some shortcomings of such a strategy includes the complexity of the proposed solutions, e.g., [14], [25]; or sometimes the requirement of special hardware and equipment, e.g., [28]. Moreover, most of the research efforts in this direction have failed to study the feasibility of the distance-based localization problem under adverse conditions.

In view of the above, our primary goal here is to conduct a thorough analytical study of the distance-based localization problem in the presence of cheating beacons. The secure distance-based localization framework and the associated results that we present in this paper are very general. The algorithms for secure localization that we propose achieve provable security and are computationally feasible and efficient. As a matter of fact, it will be clear later that the class of bounded error distance-based localization algorithms proposed in this paper also includes other algorithms such as the optimization-based scheme by Misra et al. [30] and the voting-based technique by Liu et al. [17]. Next, we first outline the network and adversary model for the secure distance-based localization framework.

3 Network and Adversary Model

In our network model, a device M in a non-trustworthy environment, wants to compute its own location by using distance estimates to a set of beacon nodes. These beacon nodes know their own locations and may or may

not cheat about their locations to the other nodes. The target node M and the beacon nodes are currently assumed to be located on a two-dimensional area (plane), i.e., the location of each of these entities can be represented as two-dimensional coordinates (x, y) where, $x, y \in \mathbb{R}$.

Suppose that the target node M has a total of n beacon nodes available for localization. Let these beacon nodes be denoted as B_1, ..., B_n. Among these n beacons, some beacons are malicious (or cheating beacons). Let k denote the number of malicious or cheating beacons. It is important to note that k is not necessarily known to the target node or to any of the honest beacons. However, the value of k clearly has a great influence on whether a bounded localization error can be achieved or not. Let k_{max} ($\leq n$) be an upper bound on the number of malicious nodes, i.e., k_{max} is the maximum number of malicious nodes that can exist in the network at any time. The parameter k_{max} is a system or environment-dependent constant and is assumed to be known to the localization algorithm.

Beacons that are not malicious are honest, i.e., they fully cooperate with the localization protocol by disclosing the information as truthfully as possible. More details on the cheating behavior by the beacon nodes will follow shortly. Regardless of being honest or dishonest, each beacon B_i provides[1] M with a measurement $\tilde{d}_i$ of the distance between B_i and M. The precise distance between B_i and M is the Euclidean distance between the position coordinates of B_i and M, and is denoted by $dst(B_i, M)$. Let the set of honest beacons be denoted by H. Then, for each beacon $B_i \in H$, $\tilde{d}_i$ is a random variable that follows some probability distribution, denoted as $msr(dst(B_i, M))$, such that $E[\tilde{d}_i] = dst(B_i, M)$, i.e., the expected (mean) value of the estimated distance $\tilde{d}_i$ for each beacon B_i in H, is the precise distance between the beacon B_i and the node M. In the case when B_i is honest, the difference between the estimated and the true distance is very small, i.e.,

$$|\tilde{d}_i - dst(B_i, M)| < \epsilon \quad (1)$$

where ϵ is the maximum distance estimation error. Ideally, this difference should be zero, but such discrepancies in distance estimates can occur due to *measurement errors*, either at the source or target. Currently, ϵ can be assumed to be a small constant. Later in Section 7, we extend the current network model to include a more practical representation for the distance estimation error.

For each beacon $B_i \notin H$, i.e., a cheating beacon, the corresponding $\tilde{d}_i$ is a value selected (possibly arbitrarily) by the adversary such that it may or may not follow Equation 1. Note that we allow colluding attacks in this model, i.e., we assume that a single adversary controls all the malicious beacon nodes (all $B_i \notin H$) and decides $\tilde{d}_i$ for them. This is a very strong adversary model that in addition to independent adversaries also covers all possibilities of collusion.

As a distance-based localization strategy is assumed here, the output O of the corresponding localization algorithm can be defined by a function F of the measured distances ($\tilde{d}_i$) from the device M to every available beacon node, i.e., $O = F(\tilde{d}_1, \ldots, \tilde{d}_n)$.

The error e of the localization algorithm is the expected value of the Euclidean distance between the actual position of M and the one output by the algorithm, i.e., $e = E[dst(M, O)]$.

In the next section, we outline the framework for bounded error distance-based localization in the presence of malicious beacon nodes.

4 Bounded Error Distance-based Localization

Before describing our secure localization framework, we derive the necessary condition for bounded error localization in the presence of cheating beacons. This condition fixes the minimum number of beacons required to correctly compute the target node location by using just the distance information, assuming that some of the beacon nodes will cheat during localization.

4.1 Necessary Condition

In order to achieve a bounded localization error, the first step is to derive a threshold for the number of malicious beacons k (in terms of the total number of available beacons n) such that if k is greater than or equal to this threshold then no algorithm would be able to guarantee a bounded localization error just based on the distances to the beacon nodes. Consequently, having the number of malicious beacons below this threshold is a necessary condition for getting a bounded localization error out of any distance-based localization algorithm. This condition is given by Theorem 4.1.

Theorem 4.1. *Suppose that $k \geq \frac{n-2}{2}$. Then, for any distance-based localization algorithm, for any locations of the beacons, there exists a scenario in which e is unbounded.*

For the sake of brevity, we skip the proof of this theorem. Interested readers can find the proof in [31].

Theorem 4.1 proves that having $\frac{n-2}{2}$ or more cheating beacons makes it impossible to compute the location of the target node M with a bounded error. In the next set of results, we establish that having $\frac{n-3}{2}$ or fewer cheating beacons makes it possible to compute the location of M with a bounded error. This condition can also be regarded as a *sufficient* condition for secure and robust distance-based localization.

1. In practice, each beacon B_i actually provides M with some information from which the distance $\tilde{d}_i$ can be computed efficiently by M. In order to simplify the current exposition, we assume that B_i provides M the distance measurement $\tilde{d}_i$ directly. This should not affect the presented results.

4.2 Class of Robust Localization Algorithms

Before defining the class of algorithms that can achieve bounded error localization in the presence of cheating beacons, let us introduce some terminology used for its definition (See Figure 2). For each beacon B_i, define a

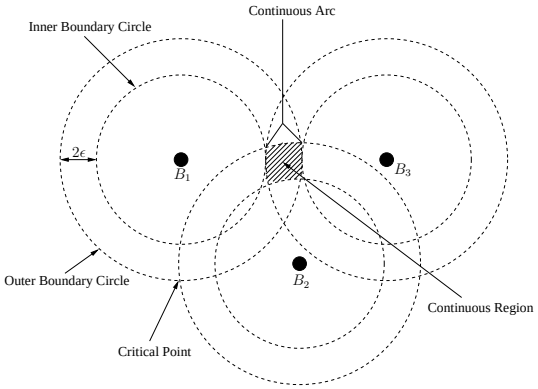


Fig. 2. Terminology for the class of robust localization algorithms

ring[2] R_i using the following inequality:

$$\tilde{d}_i - \epsilon < dst(B_i, X) < \tilde{d}_i + \epsilon.$$

As mentioned in Section 3, ϵ is assumed to be a constant denoting some (small) maximum distance estimation error. Clearly, there are altogether n rings. The boundaries of these n rings consists of $2n$ circles — called the *boundary circles*. In particular, the inner circle of the ring is called an *inner boundary circle* and the outer circle is called an *outer boundary circle*.

Definition 4.1. *A point is a* critical point *if it is the intersection of at least two boundary circles.*

Definition 4.2. *An arc is a* continuous arc *if it satisfies the following three conditions:*

- *The arc is part of a boundary circle.*
- *It is either a complete circle or an arc with two distinct end points, both of which are critical points.*
- *There is no other critical point inside the arc.*

Definition 4.3. *An area is a* continuous region *if it satisfies the following two conditions:*

- *The boundary of this area is one or more continuous arcs.*
- *There is no other continuous arc inside the area.*

The class of robust localization algorithms can then be defined as follows.

Definition 4.4. *A localization algorithm is in the* class of robust localization algorithms *if its output is a point in a continuous region r such that r is contained in the intersection of at least $k+3$ rings.*

The class of robust localization algorithms defined above is a *non-empty* class of algorithms. This statement follows from the following theorem that proves that as long as $k \leq \frac{n-3}{2}$, it is always possible to find a non-empty continuous region r satisfying the requirements of Definition 4.4.

Theorem 4.2. *For $k \leq \frac{n-3}{2}$, there exists a non-empty continuous region r in the intersection of at least $k+3$ rings.*

For the sake of brevity, we skip the proof for this theorem. Interested readers can find the proof in [31].

In fact, an example algorithm that belongs to this class is the voting-based localization scheme proposed by Liu et al. [17]. In this scheme, the authors compute the intersection region by dividing the entire localization area into a square grid, and then take a vote for each candidate location on the grid. The candidate locations with the maximum votes belong to the intersection area. In another similar research effort, Misra et al. [30] estimate the target location by approximating the centroid of the intersection region using convex optimization. Although reasonably accurate, both the voting-based scheme by Liu et al. and the optimization technique by Misra et al. are computationally intensive. In Section 5, we will propose three novel algorithms in this class of robust localization algorithms. But, first we derive the worst-case error bound for this class of algorithms.

4.3 Error Bound Analysis

To analyze the error bound of algorithms in this class, two new definitions are needed.

Definition 4.5. *The* beacon distance ratio (γ) *is defined as the minimum distance between a pair of beacons divided by the maximum distance between a beacon and the target device.*

$$\gamma = \frac{\min_{B_i,B_j} dst(B_i, B_j)}{\max_{B_i} dst(B_i, M)}$$

Definition 4.6. *Consider the lines going through pairs of beacons. Denote by $ang(B_iB_j, B_{i'}B_{j'})$ the angle between lines B_iB_j and $B_{i'}B_{j'}$ — to avoid ambiguity, we require that $0^\circ \leq ang(B_iB_j, B_{i'}B_{j'}) \leq 90^\circ$. The* minimum beacon angle (α) *is defined as the minimum of such angles.*

$$\alpha = \min_{B_i,B_j,B_{i'},B_{j'}} ang(B_iB_j, B_{i'}B_{j'})$$

The following theorem bounds the maximum localization error possible in the presented robust localization framework.

Theorem 4.3. *For $k \leq \frac{n-3}{2}$, if $\epsilon \ll \min_{B_i} dst(B_i, M)$ and there are no three beacons in the same line, then the output error of any algorithm in the class of algorithms for robust localization, as defined in Definition 4.4, is*

$$e < \frac{2\epsilon}{\min\left\{\sin\frac{\arcsin(\gamma\sin(\alpha/2))}{2}, \cos\frac{\arcsin(\gamma\sin(\alpha/2))}{2}\right\}}$$

2. Note that although we use a ring to model the error between the actual distance and the measured distance, it does not imply that we assume a circular or disc like coverage for each beacon. Given that each beacon is equipped with an omni-directional antenna (with an irregular transmission/coverage pattern) with some maximum transmission range and maximum distance estimation error ϵ, the distance sent from each beacon to M, regardless of where M is relative to the beacon, can be assumed to lie within a ring.

Proof: Consider the continuous region r in the intersection of at least $k+3$ rings (by Theorem 4.2). As there are at most k dishonest beacons, at least 3 of these rings belong to the set of honest beacons. Suppose that R_1, R_2, and R_3 are these three rings, and let r' be the continuous region in the intersection of R_1, R_2, and R_3. It is clear that r' contains r. As O is in r, clearly O is also in r'. Next, let's show that M is also in r'. As M is also in the intersection of R_1, R_2, and R_3, to show that M is in r' only the following lemma is needed, a proof of which can be found in the Appendix:

Lemma 4.4. *If $\epsilon \ll \min_{B_i} dst(B_i, M)$ and there are no three beacons in the same line then the intersection of R_1, R_2, and R_3 has only one continuous region.*

From Lemma 4.4 we have established that both M and O are in r'. We will use this fact to show that

$$e < \frac{2\epsilon}{\min\left\{\sin\frac{\arcsin(\gamma\sin(\alpha/2))}{2}, \cos\frac{\arcsin(\gamma\sin(\alpha/2))}{2}\right\}}$$

But before this result can be proved, we need another lemma that characterizes the angle formed by M with the honest beacon pairs. The proof of the lemma can be found in the Appendix.

Lemma 4.5. *If there are no three beacons in the same line, then either*

$$ang(B_1M, B_2M) \geq \arcsin(\gamma\sin(\alpha/2)),$$

or

$$ang(B_1M, B_3M) \geq \arcsin(\gamma\sin(\alpha/2)).$$

Using Lemma 4.5, without loss of generality let us assume that

$$ang(B_1M, B_2M) \geq \arcsin(\gamma\sin(\alpha/2)).$$

Denote by r'', the continuous region in the intersection of R_1 and R_2 that contains r'. As both M and O are in r', they should also be in r''.

Each of the two rings involved has a pair of circles. Consider the four intersection points of these two pairs of circles. Without loss of generality, suppose that the four intersection points are V_1, V_2, V_3, and V_4, ordered in the clockwise direction, and that $\angle V_2V_1V_4$ is acute. As $\epsilon \ll \min_{B_i} dst(B_i, M)$, r'' can be approximated using the quadrangle $V_1V_2V_3V_4$. It is easy to show that

$$ang(V_1V_2, B_1M) \approx 90^\circ \approx ang(V_3V_4, B_1M)$$

Thus, it is clear that the line V_1V_2 is parallel to the line V_3V_4. Similarly, we can get that the line V_1V_4 is parallel to the line V_2V_3. Therefore, $V_1V_2V_3V_4$ is a parallelogram. Furthermore, it can be seen that

$$\begin{aligned}\angle V_2V_1V_3 &= \arcsin\left(\frac{2\epsilon}{dst(V_1, V_3)}\right)\\ &= \angle V_3V_1V_4.\end{aligned}$$

Therefore, $V_1V_2V_3V_4$ is actually a rhombus. In a rhombus, the farthest distance between two points is the length of its longer diagonal line. Therefore,

$$\begin{aligned}e = dst(M, O) &\leq \frac{2\epsilon}{\sin(\angle V_2V_1V_3)}\\ &= \frac{2\epsilon}{\sin\left(\frac{\angle V_2V_1V_4}{2}\right)}\\ &\approx \frac{2\epsilon}{\min\left\{\sin\left(\frac{ang(B_1M,B_2M)}{2}\right), \sin\left(90^\circ - \frac{ang(B_1M,B_2M)}{2}\right)\right\}}\\ &\leq \frac{2\epsilon}{\min\left\{\sin\left(\frac{\arcsin(\gamma\sin(\alpha/2))}{2}\right), \cos\left(\frac{\arcsin(\gamma\sin(\alpha/2))}{2}\right)\right\}}\end{aligned}$$

□

4.4 Discussion

We now discuss the security implications of the analytical results that have been presented so far. Theorems 4.1 and 4.2 prove that if a total of n beacons are available for localization, then secure distance-based localization is possible if and only if there are no more than $\frac{n-3}{2}$ cheating beacons among them. In other words, if these conditions are satisfied, then no matter how all the malicious beacons cheat, i.e., individually or in collusion with each other, a bounded error (given by Theorem 4.3) can always be guaranteed. It is not possible for cheating beacons, even by colluding with every other cheating beacon, to localize the target node such that the localization error is greater than this upper bound, i.e., they cannot succeed in localizing the target node outside of the continuous region formed by the intersection of at least $k+3$ rings. In the worst case, the cheating beacons (maximum k) can influence the size of this continuous region in the intersection of at least $k+3$ rings (still bounded by the 3 honest beacon rings) or can collude to maximize the localization error (Theorem 4.3) of the target node within the continuous region. This can easily be thwarted by considering the continuous region in the intersection of a maximum number, but at least $k+3$, rings. Finally, if Theorem 4.1 is not satisfied, then a continuous region in the intersection of at least $k+3$ rings cannot be guaranteed and cheating beacons can make the localization error arbitrarily large. If the adversary model, in this case, is relaxed to remove the possibility of collusion, then simple majority-based schemes such as the voting [17] can be used for securing localization.

In the next section, we propose three novel algorithms that belong to the class of robust localization algorithms and can guarantee a bounded localization error.

5 Bounded Error Algorithms

The class of robust localization algorithms, as defined in Definition 4.4, contains algorithms that output the location of a target in the continuous region of at least $k+3$ rings. In this section, we propose three algorithms

that belong to this class. The first algorithm, called the Polynomial Time algorithm, has a polynomial time (in terms of number n of available beacons) worst-case computational complexity, which is much faster than an exhaustive search of all the grid points [17]. However, in practice it is still very slow. We also propose two heuristic-based algorithms. It is not known if their worst-case complexity is any better than that of the Polynomial Time algorithm. Yet, the probability of reaching the worst-case is less and the heuristic-based algorithms run efficiently in most cases and for most network topologies. Recall that all the three algorithms work under the condition $k \leq \frac{n-3}{2}$. Thus, an upper bound for k (number of malicious beacons) can be defined as $k_{max} = \frac{n-3}{2}$. All the algorithms presented here output a point within the continuous region r in the intersection of $k_{max}+3$ rings as the location of the target node, but they differ in the way they determine this point.

5.1 Polynomial Time Algorithm

Before outlining details of the Polynomial Time algorithm, we give a lemma that defines the relationship between a continuous region and a continuous arc.

Definition 5.1. *A ring is* related *to a continuous arc if the continuous arc is inside, but not on the boundary of this ring.*

Lemma 5.1. *Suppose that r is a continuous region and c is a continuous arc on the boundary of r. Then, r is in the intersection of at least $k+3$ rings if and only if at least $k+2$ rings are related to c.*

(We skip the proof of Lemma 5.1 as it is very straightforward.)

The main idea behind the Polynomial Time algorithm is that in order to determine a continuous region in the intersection of at least $k_{max}+3$ rings, it is sufficient to count the number of rings related to each continuous arc, and then find a continuous arc such that at least $k_{max}+2$ rings are related to it (It is easy to check whether a ring is related to a *continuous* arc by comparing the distance between the arc's end points and the center of the ring to the inner and outer radii of the ring). Once such an arc is found, depending on whether the arc is on an outer boundary circle or an inner boundary circle, a point can be picked from either the inner region or the outer region of the arc respectively. The details of the Polynomial Time algorithm are shown in Algorithm 1.

Lemma 5.2. *The worst-case time complexity of the Polynomial Time algorithm (Algorithm 1) is $O(n^3 \log n)$.*

Although the worst-case time complexity of Algorithm 1 is polynomial (cubic) in terms of the total number of available beacons, it does not perform very efficiently in practice. Simulation experiments (discussed later in Section 6.2) show that it runs rather slowly for most cases. This is because, it always computes all the possible continuous arcs and searches among them for a related arc that satisfies Lemma 5.1. In other words, it first uses an exhaustive search to determine the boundary of the continuous region in the intersection of $k_{max}+3$ rings and then outputs a point within it as the target location. But there are other efficient ways to estimate such a point with a high probability, as discussed next.

```
1: Let S be a set initially containing the two boundary
   circles of ring R_1
2: for i = 2, ..., n do
3:    Let S_i be a set initially containing the two bound-
      ary circles of ring R_i
4:    for each arc in S and each arc in S_i do
5:       if the above two arcs intersect then
6:          Split each of these two arcs using the intersec-
            tion(s), and replace them in the corresponding
            arc sets (S or S_i) with the new splitted arcs
            (result of the splitting operation)
7:       end if
8:    end for
9:    Let S = S ∪ S_i
10: end for
11: for each arc c_j in S do
12:    Set the corresponding counter λ_j to 0
13:    for i = 1, ..., n do
14:       if R_i is related to c_j then
15:          λ_j = λ_j + 1
16:       end if
17:    end for
18:    if λ_j ≥ k_max + 2 then
19:       if c_j is on an inner boundary circle then
20:          Output is defined on the side out of this circle
21:       else if c_j is on an outer boundary circle then
22:          Output is defined on the side inside this circle
23:       end if
24:       Stop the algorithm
25:    end if
26: end for
```

Algorithm 1: Polynomial Time Algorithm

5.2 Heuristic 1

The first heuristic attempts to estimate the target location around a critical point that lies on the intersection of a large number of rings. It can be observed that $k_{max}+3$ is already a large number of rings (more than half of the total number of rings in the network). We need to determine the region r contained in at least $k_{max}+3$ rings. It is highly probable that the rings containing such a region r are intersecting with large numbers of other rings. In other words, if a ring, say R_i, is intersecting with a large number of rings then it is very likely that R_i contains r. Therefore, the heuristic first considers the rings intersecting with a large number of other rings in order to determine the critical point around which the target location is guessed. This continues until a target location within the continuous region in the intersection of at least $k_{max}+3$ rings is estimated. The details of

Heuristic 1 are outlined in Algorithm 2, as shown below.

```
1: Count the number of rings intersecting with each ring
2: for each ring R_i, in the order of decreasing number of rings intersecting with it do
3:   for each ring R_j, R_j ≠ R_i, in the order of decreasing number of rings intersecting with it do
4:     Compute the intersection points of the boundary circles of R_i and R_j
5:     for m = 1, ..., γ do
6:       Choose a random intersection point computed above
7:       Choose a random point O near this intersection point (such that the distance between them is less than ε)
8:       Count the number of rings containing O
9:       if there are at least k_max + 3 rings containing O then
10:        Output O
11:        Stop the Algorithm
12:      end if
13:    end for
14:  end for
15: end for
```

Algorithm 2: Heuristic 1

The next heuristic attempts to further improve the quality of localization, by trying to estimate a point closer to the center of the continuous region formed by $k_{max}+3$ intersecting rings.

5.3 Heuristic 2

The second heuristic tries to guess the location of the target closer to the center (or centroid) of the continuous region of at least $k_{max}+3$ intersecting rings. This is because the actual location of the target is more likely to be near the center of the continuous region than near the boundary. Thus, assuming that the continuous region is convex, we first compute three distinct critical points, instead of just one, that lie on the intersection of a large number of rings. If (x_1, y_1), (x_2, y_2) and (x_3, y_3) are the coordinates of these critical points, the coordinates (x_M, y_M) of the target location are guessed by computing the centroid of the triangle formed by (x_1, y_1), (x_2, y_2) and (x_3, y_3), as shown below:

$$x_M = \frac{x_1 + x_2 + x_3}{3}$$
$$y_M = \frac{y_1 + y_2 + y_3}{3}$$

If this guessed point (x_M, y_M) lies in the intersection of $k_{max}+3$ rings, then it is output as the location of the target, otherwise the procedure is repeated for a new set of critical points. Details of this heuristic are outlined in Algorithm 3 (or Heuristic 2) shown below.

```
1: Count the number of rings intersecting with each ring
2: for each ring R_i, in the order of decreasing number of rings intersecting with it do
3:   for each ring R_j, R_{j+1}, R_{j+2} | R_j, R_{j+1}, R_{j+2} ≠ R_i, in the order of decreasing number of rings intersecting with it do
4:     Compute the intersection points of the boundary circles of R_i and R_j, R_i and R_{j+1} and R_i and R_{j+2}
5:     Choose a point (x_1, y_1) from the intersection of the ring pair R_i, R_j at random. Similarly, choose intersection points (x_2, y_2) and (x_3, y_3) from the other two pairs
6:     Compute O = ((x_1+x_2+x_3)/3, (y_1+y_2+y_3)/3)
7:     Count the number of rings containing O
8:     if there are at least k_max + 3 rings containing O then
9:       Output O
10:      Stop the Algorithm
11:    end if
12:  end for
13: end for
```

Algorithm 3: Heuristic 2

6 EVALUATION

The evaluation of the proposed robust localization algorithms includes the verification of accuracy and efficiency of each of these algorithms and comparison with other known techniques such as the voting-based scheme by Liu et al. [17]. The simulations for these algorithms are carried out for varying values of parameters such as beacon node distribution, number of malicious beacon nodes and distance estimation error of the target node. Currently, we do not evaluate any network-specific property of these algorithms such as the communication overhead. This is because these algorithms, as proposed currently, are very general and properties such as communication overhead would depend on network specific factors such as hardware, signal type, ranging technique and the network topology. In the first part of this simulation-based analysis, we aim to compare the performance of the proposed algorithms under ideal network conditions with a small independent distance estimation error. Later in Section 7, we extend the initial simple simulation setup to include more realistic distance estimation error models. Results from these initial simulation experiments will serve as a stepping stone for improving these algorithms further and porting them to more complex network platforms and environments.

6.1 Simulation Setup

The simulation area consists of a $500m \times 500m$ two dimensional terrain. The optimal number and placement of beacon nodes is important. But as optimal beacon placement is not the main focus of this paper, we assume

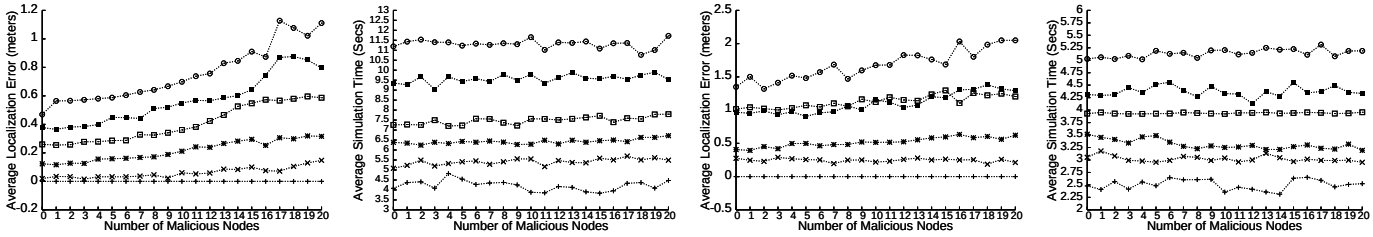


Fig. 3. Simulation of the Polynomial Time algorithm (a) Localization error vs No. of malicious nodes and (b) Simulation time vs No. of malicious nodes with measurement error uniformly distributed between $[-\epsilon, +\epsilon]$. (c) Localization error vs No. of malicious nodes and (d) Simulation time vs No. of malicious nodes with measurement error normally distributed between $[-\epsilon, +\epsilon]$ with mean 0 and standard deviation $\frac{\epsilon}{2}$.

a small but reasonable beacon node population of 43 beacon nodes (approximately 1 beacon for every $10m \times 10m$), which is scattered uniformly over the $500m \times 500m$ area. The position of the target node is also uniformly selected and there is no node mobility (beacon or target). Currently, the maximum radio ranges of the nodes are selected such that every beacon node is available for localization ($\approx 250m$). In this set of simulations, we assume an independent distance estimation error selected from some fixed distribution. In order to verify the accuracy and efficiency of the proposed algorithms for different distributions of the distance estimation error, we simulate the algorithms for both uniformly and normally distributed distance estimation errors. For each of these distributions, we intend to study the influence of the number of malicious beacons (k) and the maximum distance measurement error (ϵ) on the localization error and the execution time of the algorithms.

6.2 Polynomial Time Algorithm

In this section, we discuss the simulation results for the Polynomial Time algorithm.

6.2.1 Experiments with Uniform Measurement Error

In the first set of simulations, we evaluate the Polynomial Time algorithm for the case when the distance estimation error is uniformly distributed between $[-\epsilon, \epsilon]$. We observe the performance of the algorithm for increasing values of ϵ, as the number k of malicious nodes increases from 0 up to some maximum tolerable value. As the total number of available beacons is fixed ($n = 43$), the maximum number of malicious beacons that the algorithm can tolerate is $\frac{43-3}{2} = 20$ (from Theorem 4.2). The algorithm is executed for each value of ϵ from $0m$ to $5m$ in steps of $1m$ and for each value of k from 0 to 20 ($k_{max} = 20$). We then plot the average localization error e as an average of the error in localization of the target over 100 runs of the algorithm (See Figure 3). In each new run, the beacon and target nodes are assigned new positions, the coordinates of which are uniformly selected over the $500m \times 500m$ area. From Figure 3(a), it can be seen that the average localization error shows an increasing trend when ϵ increases, which is very natural. When $\epsilon = 0$, e is also 0. The reason is that in this case the continuous region is just a single point in the intersection of at least $k_{max} + 3$ rings. Also it can be seen that e increases as k increases. This is consistent with the intuition that more number of malicious beacon nodes should decrease localization precision. For lower values of k, i.e., $k < k_{max}$, more honest rings are available for localization, resulting in a smaller sized continuous region and thus a more accurate localization. As the number of malicious nodes increases, the number of honest rings diminishes and thus the quality of localization decreases.

Figure 3(b) depicts the average execution time of the Polynomial Time algorithm under varying values of k and ϵ. From the figure, we can see that the average simulation time does not increase very sharply with k. This observation is also not surprising because in all the cases the Polynomial Time algorithm always computes all the possible continuous arcs. Increasing the value of k does not guarantee a lower number of continuous arcs because the locations of the malicious beacons are selected uniformly over the $500m \times 500m$ area. But the simulation time increases with an increase in the value of ϵ. This is because, for lower values of ϵ, the inner and outer boundary circles are much closer to each other (width of the ring is smaller) as compared to higher values of ϵ, thus resulting in lesser number of possible continuous arcs. In summary, the maximum localization error of the Polynomial Time algorithm is less than $1m$ for a maximum distance error of $5m$ (for the Uniform distribution case), which is an error ratio ($\frac{e}{\epsilon}$) of approximately 0.2. The maximum simulation time for this case is just under 12 secs, which is a bit high.

6.2.2 Experiments with Normal Measurement Error

To verify that the evaluation results are consistent and not restricted to a particular distribution, we repeat the simulations for the Polynomial Time algorithm using a normally distributed distance measurement error. All other simulation parameters are kept unchanged except that the distance measurement error takes values from a

truncated normal distribution with mean 0 and standard deviation $\frac{\epsilon}{2}$. To make sure that the distance estimation error always takes values between $-\epsilon$ and ϵ, the normal distribution is modified such that the probability density outside $[-\epsilon, +\epsilon]$ becomes 0; the probability density inside the interval $[-\epsilon, +\epsilon]$ is scaled up a little, accordingly.

Figures 3(c) & 3(d) show the plots for the simulation results. Figure 3(c) plots the average localization error for each pair of (k, ϵ) when the distance estimation error follows a normal distribution. Figure 3(d) shows the corresponding simulation time plot. We can observe that these plots are analogous to Figures 3(a) and 3(b), respectively, except that the localization error increases more slowly with k in the current case. These plots verify that the behavior of the Polynomial Time algorithm is consistent for other distributions of the distance measurement error as well. In summary, we observe that although the accuracy of the Polynomial Time algorithm is good, it is very inefficient and slow, with execution time in the order of seconds.

6.3 Heuristic 1

In this section, we discuss the evaluation of Heuristic 1.

6.3.1 Experiments with Uniform Measurement Error

Similar to the Polynomial Time algorithm, we first evaluate Heuristic 1 for uniformly distributed values of the distance measurement error. The simulation of the algorithm is run for each value of ϵ from $0m$ to $50m$ in steps of $10m$ and for each value of k from 0 to 20 ($k_{max} = 20$). Note that here we have drastically increased the value of ϵ, compared to the previous experiments. It would be worthwhile to observe the effects of larger measurement errors on the localization accuracy and execution time of Heuristic 1. Average localization error e is plotted as an average of the error in localization of the target node over 1000 runs (See Figure 4 (a) & (b)). In each run, the beacons and target node are assigned new positions, coordinates of which are uniformly selected over the $500m \times 500m$ area.

From Figure 4(a), we can see that the average localization error e increases as ϵ increases, which is an intuitive observation. Also, e increases as k increases. This is also consistent with the intuition that more number of malicious beacon nodes decreases localization precision. For lower values of k, i.e., $k < k_{max}$, more honest rings are available for localization, resulting in a smaller region of intersection and eventually a more precise localization. As the number of malicious nodes increases, the number of honest rings reduces (but still satisfying the necessary and sufficient conditions), and thus the quality of localization decreases.

Figure 4(b) shows that the average simulation time of Heuristic 1 increases in k, but increases *only very slightly*. This observation is also not surprising because the algorithm is computing the intersection of the same number of rings for each value k. The main reason for the slight increase in the simulation time is that more number of malicious beacons make it harder to find the right continuous region (in the intersection of $k_{max} + 3$ rings). For all values of k and ϵ, the average localization error of Heuristic 1 is just under $25m$, which is an error ratio ($\frac{e}{\epsilon}$) of around 0.5, whereas the execution time in the worst case is less than 0.035 secs.

6.3.2 Experiments with Normal Measurement Error

Once again, to ensure that the evaluation results are not restricted to only uniformly distributed errors, the simulations for Heuristic 1 are repeated with a normally distributed distance estimation error. All other experiment parameters are unchanged. The distance measurement error follows a normal distribution with mean 0 and standard deviation $\frac{\epsilon}{2}$. As before, the distribution is modified such that the probability density outside $[-\epsilon, +\epsilon]$ becomes 0.

Figure 4(c) plots the average localization error e for each pair of (k, ϵ) when the measurement error follows a normal distribution. Figure 4(d) plots the corresponding simulation time. We can observe that the curves are analogous to those in Figures 4(a) and 4(b) respectively, except that the localization error e increases much more slowly with k.

6.4 Heuristic 2

The values of the simulation parameters for Heuristic 2 are similar to those used for Heuristic 1. As before, we evaluate Heuristic 2 for both uniformly and normally distributed distance measurement errors. Plots of the simulation results are shown in Figures 4 (e), (f), (g) & (h). One very obvious trend in the plot for average localization error e, as can be seen from Figures 4 (e) & (g), is that the error does not increase with k, but increases with ϵ. In other words, k does not influence the localization accuracy of the algorithm in a major way, which is a good thing. This trend in the localization accuracy is also not surprising. Because here we are computing the centroid of the three boundary points, the localization accuracy depends on the width of rings, which in turn depends on the value of ϵ. The execution time, however, decreases with the increase in ϵ. This is because, for larger values of ϵ, the continuous region is larger thus making it more probable that the computed centroid lies within the continuous region. For the uniform distribution case, the error ratio ($\frac{e}{\epsilon}$) is just under $\frac{10}{50} = 0.2$, which is similar to the one provided by the Polynomial Time algorithm. Also, the execution time in the worst case is around 0.01 seconds (see Figure 4 (f)), which is much faster (roughly, 1000 times) as compared to the Polynomial Time algorithm.

From the above experimental results, we can conclude that both the Polynomial Time algorithm and Heuristic 2 have very good localization accuracy, but Heuristic 2 runs very efficiently compared to the other two algorithms and outperforms them in execution speed.

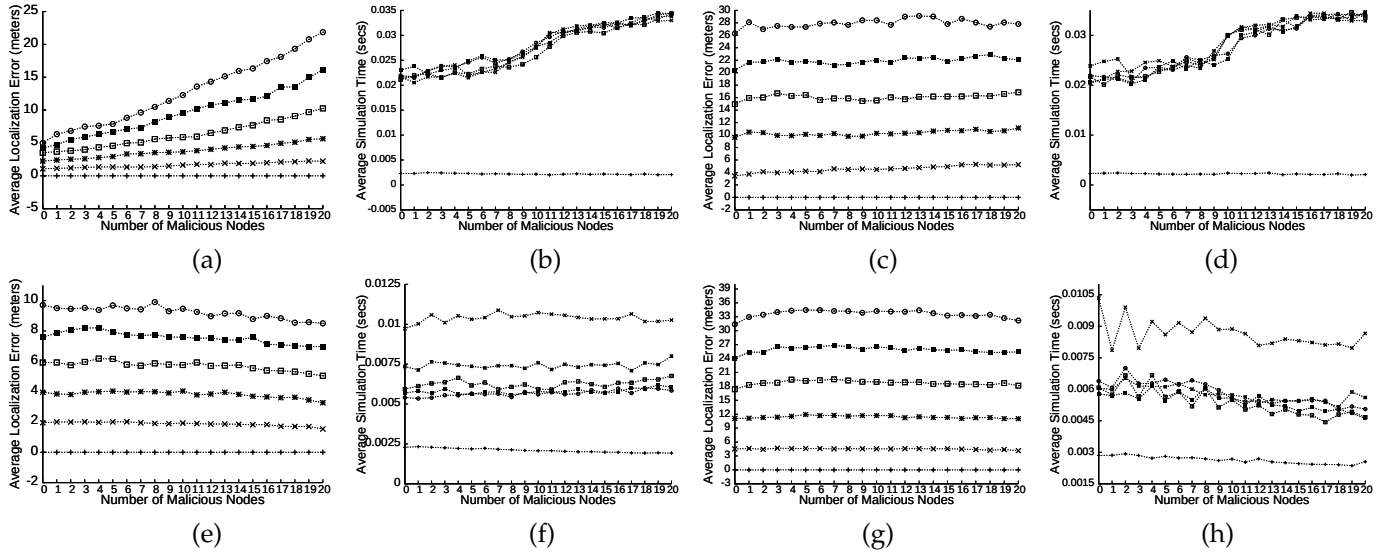


Fig. 4. (a) & (b) Heuristic 1 with measurement error uniformly distributed between $[-\epsilon, +\epsilon]$; (c) & (d) Heuristic 1 with measurement error normally distributed between $[-\epsilon, +\epsilon]$ with mean 0 and standard deviation $\frac{\epsilon}{2}$; (e) & (f) Heuristic 2 with measurement error uniformly distributed between $[-\epsilon, +\epsilon]$; (g) & (h) Heuristic 2 with measurement error normally distributed between $[-\epsilon, +\epsilon]$ with mean 0 and standard deviation $\frac{\epsilon}{2}$

6.5 Comparison with Voting-based Scheme

In this section, we compare the proposed heuristic-based algorithms with the voting-based scheme by Liu et al. [17], which was discussed earlier in Section 2. The setup and simulation parameters remain unchanged. Plots of the simulation results for uniformly distributed distance estimation errors are shown in Figures 5 (a) & (b).

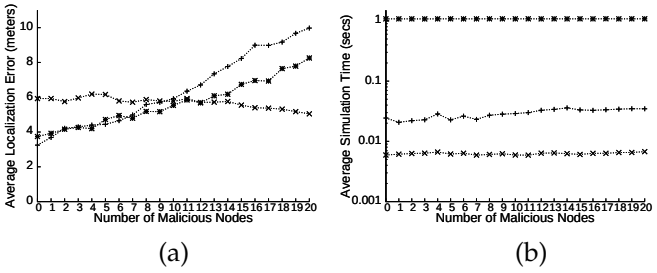


Fig. 5. Comparison with voting-based scheme of [17] (a) Localization error and (b) Simulation time with measurement error uniformly distributed between $[-30, +30]$

From Figure 5 (a), we can see that for lower values of the number of malicious nodes k, the voting-based scheme has similar accuracy as Heuristic 1, but at higher values of k it performs slightly better. Moreover, Heuristic 2 outperforms both Heuristic 1 and the voting-based scheme at higher values of k. Also, the localization accuracy of the voting-based scheme decreases with increase in k. In terms of execution efficiency, the voting-based scheme performs the worst, almost 100 times slower than the other algorithms.

7 Practical Error Models

From Theorem 4.3, we can see that the maximum localization error for the class of robust localization algorithms is proportional to the maximum distance estimation error ϵ. Up to this point, we have assumed that for a given target-beacon pair, the distance measurement error is selected from a fixed uniform or non-uniform distribution and is bound by some constant ϵ. In practice, however, modeling distance estimation error is not so straightforward. Errors can be introduced during distance estimation due to a variety of factors including the actual distance between the target and beacon nodes, antenna gains of the transceivers, technique employed for distance estimation and signal type, and environmental factors such as obstacles and noise. Thus in this section, we consider more practical distance estimation error models and investigate their effect on the performance of the class of robust distance-based localization algorithms.

7.1 Modeling Distance Measurement Error

In wireless networks, such as ad hoc and sensor networks, various schemes have been proposed for distance estimation [1]. Consider as an example the Received Signal Strength Indicator or RSSI technique. In this technique, the target node observes the power loss of the received beacon radio signal and uses known (through theoretical and empirical results) power loss models to estimate the distance between itself and the corresponding beacon node. Errors in RSSI-based distance estimation techniques can be attributed to various factors such as reflection, scattering and diffraction of radio signals, as well as interference due to noise, signal

fading due to multipath propagation and node mobility, and other Non-Line-Of-Sight (NLOS) errors due to obstructions. Although the precise radio signal loss model would depend on network specific factors (including environment, mobility, transmission bandwidth and radio hardware), the most commonly used theoretical and empirical propagation models indicate that the average received radio signal power decreases logarithmically with the distance between the nodes. Such models have been used extensively in the literature to model distance estimation errors. For example, Slijepcevic et al. [32] model the distribution $\tilde{d}$ of the measured distance between the target and the beacon node using the *Log-normal shadowing model* described in [33] as follows.

$$10\eta log(\frac{\tilde{d}}{d_0}) - 10\eta log(\frac{d}{d_0}) = X_\sigma \quad (2)$$

where d is the correct distance between the nodes, η is the path loss exponent that indicates the rate at which the path loss increases with distance, d_0 is the close-in reference distance that is determined from measurements close to the transmitter and X_σ is a zero-mean Gaussian distributed random variable (in dB) with standard deviation σ (also in dB). From Equation (2):

$$\tilde{d} = d + d(1 - 10^{\frac{X_\sigma}{10\eta}}) \quad (3)$$

Here $\tilde{\epsilon} \equiv d(1 - 10^{\frac{X_\sigma}{10\eta}})$ is the distribution of the distance measurement error. From Equation (3), we can see that the distance estimation error in RSSI-based techniques depends on the distance between the target and the beacon node. This dependency is also verified through empirical measurements by Savvides et al. [34].

Similarly, acoustic signals can also be used for ranging, and Savvides et al. [34] show that it achieves better accuracy than RSSI. Slijepcevic et al. [32] model the distance measurement error in acoustic ranging techniques based on the results reported in [35], and show that there are three important sources of error in acoustic ranging techniques, namely, NLOS error, speed of sound error and orientation error. These components of the distance estimation error also depend on the distance between the source and destination transceivers. For the sake of brevity, we do not provide the details of the acoustic error model here. However, we can conclude from the above discussion that the distance estimation error between the target and any beacon node depends (in addition to other factors) on the actual distance between the two nodes. In the following section, we discuss how such a practical distance estimation error model affects the maximum error bound for the class of robust localization algorithms.

7.2 Maximum Error Bound

Let us assume that the maximum distance estimation error is some function f of the distance between the target and the beacon node. Then, if $k \leq \frac{n-3}{2}$ and $\max_{i=1}^{n}\{\epsilon_i = f(dst(B_i, M))\} \ll min_{B_i} dst(B_i, M)$, where k is the number of malicious beacons, there will always exists a non-empty continuous region r in the intersection of at least $k+3$ rings. In other words, the condition for bounded-error distance-based localization (Theorem 4.2) also holds in this case. Moreover, it is easy to see that the size of this continuous region r is bound by $\max_{i=1}^{n}\{\epsilon_i = f(dst(B_i, M))\}$, i.e., the largest value of the distance estimation error in the network of n beacon nodes. Thus in this case, the maximum localization error of any algorithm that always outputs the location of the target within the continuous region of at least $k+3$ beacon rings can be given by a more generic restatement of Theorem 4.3 as follows. If $k \leq \frac{n-3}{2}$ and $\epsilon_i = f(dst(B_i, M))$ for some function f, and if $\max_{i=1}^{n}\{\epsilon_i = f(dst(B_i, M))\} \ll \min_{B_i} dst(B_i, M)$, and there are no three beacons in the same line, then the output error of any algorithm in the class (defined in Definition 4.4) of algorithms for robust localization is

$$e < \frac{2 \times \max_{i=1}^{n}\{\epsilon_i = f(dst(B_i, M))\}}{\min\left\{\sin\frac{\arcsin(\gamma\sin(\alpha/2))}{2}, \cos\frac{\arcsin(\gamma\sin(\alpha/2))}{2}\right\}} \quad (4)$$

Next, we re-evaluate using simulation experiments the heuristic-based algorithms, as well as the voting-based technique of [17], by assuming the distance estimation error model of Equation 3. The simulation results are outlined in the following section.

7.3 Evaluation

In this set of experiments, we repeat the simulations of Heuristic 1, Heuristic 2 and the voting-based technique [17] using the distance estimation error model of Equation (3). In order to simulate a realistic propagation environment such as a building with obstructions, we choose a high value for the path loss exponent η (From Table 4.2 in [33]). Specifically, the value of η is chosen as 4.0 and standard deviation σ in X_σ is chosen as 1.0. All other simulation parameters are kept unchanged. The average localization error e over 1000 runs of the simulations is plotted against the number of malicious nodes k, as shown in Figure 6.

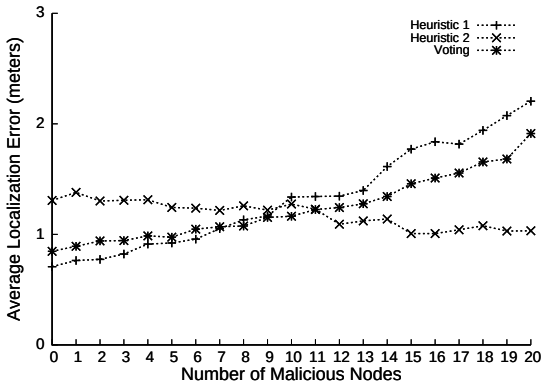


Fig. 6. Evaluation of robust localization algorithms using a practical error model

We can see from Figure 6 that the plot trends are very similar to the case with the distance-independent error models (as shown in Figures 4(a), 4(e) and 5 (a) respectively). Even in the current practical setting, the average localization error for Heuristic 1 and the voting-based scheme increases only slowly with the number of malicious beacons whereas the average localization error for Heuristic 2 is almost constant. We can also see that the average localization error of all the algorithms is lower, compared to the case with the distance-independent errors. This is because, all the beacon nodes that are closer to the target node have a lower distance estimation error, thus reducing the overall localization error. The localization accuracy of Heuristic 1 is very similar to the one provided by the voting-based approach, but the voting based approach is very slow, almost 100 times slower than Heuristic 1. For conciseness, we do not plot the simulation time comparisons here. We also observe that Heuristic 2 outperforms both Heuristic 1 and the voting-based approach at higher values of number of cheating beacons. In summary, we observe that both the proposed heuristic-based algorithms perform consistently and provide good localization accuracy even under a practical setting. They also outperform the voting-based approach in execution efficiency.

8 Conclusion and Future Work

In this paper, we have addressed the problem of secure distance-based localization in the presence of cheating beacon nodes. By means of a sound mathematical analysis, we have derived the conditions for secure and robust distance-based localization in the presence of cheating beacons. Specifically, we have outlined the necessary and sufficient conditions for achieving a bounded localization error, and defined a non-empty class of algorithms that can achieve such a bounded error.

We have also proposed three novel distance-based localization algorithms, specifically a polynomial time algorithm and two heuristic-based algorithms that belong to this class of bounded error distance-based localization algorithms. We have verified the localization accuracy and execution efficiency of these algorithms using measurements from simulation experiments. Experimental results show that all the algorithms performed consistently for different distributions of the distance measurement error. We have also extended the existing localization framework to include more practical models for the distance measurement error and have verified the performance of the algorithms under such scenarios.

The error model for radio signals currently used in the analysis can be further improved to characterize errors in specific hardware technologies and environments. The path loss parameters in the current distance estimation error model can be adjusted depending on network specific factors including obstructions, interference due to noise and multipath fading. Well-known statistical models such as Rayleigh or Rician distributions [33] or published signal measurement data sets for specific wireless systems can be used for this purpose. Distance estimation error models for other technologies such as acoustic and UWB can also be used to further analyze the proposed secure localization framework. Although the analytical results and bounds presented here are very general and have been verified for simple error models, it would be worthwhile to observe how these results (both theoretical and empirical) would extend to specific wireless environments and systems. This will be undertaken as future research on this topic.

Appendix

Proof of Lemma 4.4. A contradiction argument is used to prove this lemma. Refer to the Figure 7. Suppose that

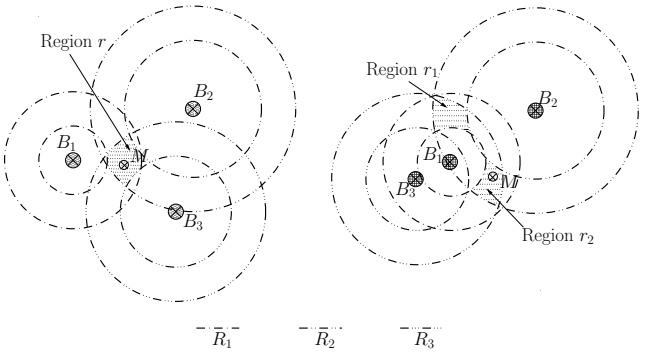


Fig. 7. Intersection of rings (Lemma 4.4)

the intersection of R_1, R_2, and R_3 has two continuous regions r_1 and r_2. Choose arbitrary points X_1 from r_1 and X_2 from r_2. Denote by X_1' (resp., X_2') the intersection of the line segment B_1X_1 (resp., B_1X_2) and the circle

$$dst(X, B_1) = \tilde{d}_1 - \epsilon.$$

Similarly, denote by X_1'' (resp., X_2'') the intersection of the line segment B_3X_1 (resp., B_3X_2) and the circle

$$dst(X, B_3) = \tilde{d}_3 - \epsilon.$$

Then clearly,

$$0 \leq dst(X_1, X_1'), dst(X_1, X_1''), dst(X_2, X_2'), dst(X_2, X_2'') \leq 2\epsilon. \tag{5}$$

We can see that,

$$\begin{aligned} ang(B_1B_3, B_1X_1) &= \arccos(dst(B_1, X_1)^2 \\ &\quad + dst(B_1, B_3)^2 - dst(X_1, B_3)^2) \\ &= \arccos((dst(B_1, X_1') \\ &\quad + dst(X_1, X_1'))^2 + dst(B_1, B_3)^2 \\ &\quad - (dst(X_1'', B_3) + dst(X_1, X_1''))^2) \\ &= \arccos((\tilde{d}_1 - \epsilon + dst(X_1, X_1'))^2 \\ &\quad + dst(B_1, B_3)^2 \\ &\quad - (\tilde{d}_3 - \epsilon + dst(X_1, X_1''))^2). \end{aligned}$$

Note that $\tilde{d}_1 > dst(B_1, M) - \epsilon \gg \epsilon$. Similarly, $\tilde{d}_3 \gg \epsilon$. Combining these facts with Equation (5) we have

$$\begin{aligned} ang(B_1B_3, B_1X_1) &= \arccos((\tilde{d}_1 - \epsilon + dst(X_1, X_1'))^2 \\ &\quad +dst(B_1, B_3)^2 \\ &\quad -(\tilde{d}_3 - \epsilon + dst(X_1, X_1''))^2) \\ &\approx \arccos((\tilde{d}_1)^2 + dst(B_1, B_3)^2 \\ &\quad -(\tilde{d}_3)^2) \\ &\approx \arccos((\tilde{d}_1 - \epsilon + dst(X_2, X_2'))^2 \\ &\quad +dst(B_1, B_3)^2 \\ &\quad -(\tilde{d}_3 - \epsilon + dst(X_2, X_2''))^2) \\ &= \arccos((dst(B_1, X_2') + \\ &\quad dst(X_2, X_2'))^2 + dst(B_1, B_3)^2 \\ &\quad -(dst(X_2'', B_3) + dst(X_2, X_2''))^2) \\ &= \arccos(dst(B_1, X_2)^2 \\ &\quad +dst(B_1, B_3)^2 - dst(X_2, B_3)^2) \\ &= ang(B_1B_3, B_1X_2). \end{aligned} \quad (6)$$

Similarly, we can show that

$$ang(B_1B_2, B_1X_1) \approx ang(B_1B_2, B_1X_2). \quad (7)$$

However, when the two equations above (equations (6) and (7)) are put together, a contradiction is reached. Without loss of generality, we assume that

$$ang(B_1B_2, B_1X_1) < ang(B_1B_3, B_1X_1),$$

since otherwise the indices 2 and 3 can be switched. It is easy to see that

$$\begin{aligned} ang(B_1B_2, B_1X_1) &= ang(B_1B_3, B_1X_1) \\ &\quad -ang(B_1B_2, B_1B_3) \\ &\leq ang(B_1B_3, B_1X_1) - \alpha \\ &\approx ang(B_1B_3, B_1X_2) - \alpha \\ &= ang(B_1B_2, B_1X_2) \\ &\quad -ang(B_1B_2, B_1B_3) - \alpha \\ &\leq ang(B_1B_2, B_1X_2) - 2\alpha \\ &\approx ang(B_1B_2, B_1X_1) - 2\alpha, \end{aligned}$$

which is a contradiction. □

Proof of Lemma 4.5. Since $ang(B_1B_2, B_1B_3) \geq \alpha$, either $ang(B_1B_2, B_1M) \geq \alpha/2$ or $ang(B_1B_3, B_1M) \geq \alpha/2$. Below it is shown that, if $ang(B_1B_2, B_1M) \geq \alpha/2$ then

$$ang(B_1M, B_2M) \leq \frac{\arcsin(\gamma \sin(\alpha/2))}{2}.$$

Similarly, if $ang(B_1B_3, B_1M) \geq \alpha/2$ then

$$ang(B_1M, B_3M) \leq \frac{\arcsin(\gamma \sin(\alpha/2))}{2}.$$

Denote by D, the distance from B_2 to the line B_1M. Then,

$$\begin{aligned} ang(B_1M, B_2M) &= \arcsin\left(\frac{D}{dst(B_2, M)}\right) \\ &= \arcsin\left(\frac{dst(B_1, B_2)\sin(ang(B_1B_2, B_1M))}{dst(B_2, M)}\right) \\ &\geq \arcsin\left(\frac{dst(B_1, B_2)\sin(\alpha/2)}{dst(B_2, M)}\right) \\ &\geq \arcsin(\gamma \sin(\alpha/2)). \end{aligned}$$

□

ACKNOWLEDGMENTS

The authors would like to thank the editors and all the anonymous reviewers for their helpful suggestions and feedback. A preliminary version of this material appeared at the 27^{th} IEEE Computer Communications Conference (INFOCOM '08) [31].

REFERENCES

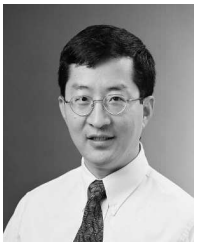
[1] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *Computer*, 2001.
[2] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The Active Badge Location System," *ACM Transaction on Information Systems*, 1992.
[3] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based User Location and Tracking System," in *Proceedings of the 19^{th} IEEE Computer Communications Conference (INFOCOM '00)*, Tel-Aviv, Israel, 2000.
[4] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in *Proceedings of the 6^{th} Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, Boston, USA, 2000.
[5] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low Cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications Magazine*, 2000.
[6] D. Niculescu and B. Nath, "DV based Positioning in Ad hoc Networks," *Journal of Telecommunication Systems*, 2003.
[7] R. Stoleru and J. A. Stankovic, "Probability Grid: A Location Estimation Scheme for Wireless Sensor Networks," in *Proceedings of the 1^{st} IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON '04)*, Santa Clara, USA, 2004.
[8] M. W. Carter, H. H. Jin, M. A. Saunders, and Y. Ye, "Spaseloc: An Adaptive Subproblem Algorithm for Scalable Wireless Sensor Network Localization," *SIAM J. on Optimization*, 2006.
[9] J. Liu, Y. Zhang, and F. Zhao, "Robust Distributed Node Localization with Error Management," in *Proceedings of the 7^{th} ACM International Symposium on Mobile Ad hoc Networking and Computing (MobiHoc '06)*, Florence, Italy, 2006.
[10] G. Mao, B. D. O. Anderson, and B. Fidan, "Path Loss Exponent Estimation for Wireless Sensor Network Localization," *Computer Networks*, 2007.
[11] R. Moses, D. Krishnamurthy, and R. Patterson, "A self-localization method for wireless sensor networks," *Eurasip Journal on Applied Signal Processing, Special Issue on Sensor Networks*, 2003.
[12] J. Xiao, L. Ren, and J. Tan, "Research of TDOA Based Self-localization Approach in Wireless Sensor Network," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006.
[13] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Poster abstract: Anchor-free Distributed Localization in Sensor Networks," in *Proceedings of the 1^{st} International Conference on Embedded Networked Sensor Systems (SenSys '03)*, 2003.
[14] X. Ji and H. Zha, "Sensor Positioning in Wireless Ad-hoc Sensor. Networks using Multidimensional Scaling," in *Proceedings of 23^{rd} IEEE Computer Communications Conference (INFOCOM '04)*, Honk Kong, China, 2004.
[15] L. Fang, W. Du, and P. Ning, "A Beacon-Less Location Discovery Scheme for Wireless Sensor Networks," in *Proceedings of the 24^{th} IEEE Computer Communications Conference (INFOCOM '05)*, Miami, USA, 2005.
[16] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust Statistical Methods for Securing Wireless Localization in Sensor Networks," in *Proceedings of the 4^{th} International Symposium on Information Processing in Sensor Networks (IPSN '05)*, Los Angeles, USA, 2005.

[17] D. Liu, P. Ning, and W. Du, "Attack-Resistant Location Estimation in Sensor Networks," in *Proceedings of the 4^{th} International Symposium on Information Processing in Sensor Networks (IPSN '05)*, Los Angeles, USA, 2005.
[18] T. Eren, D. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. Anderson, and P. Belhumeur, "Rigidity, Computation and Randomization of Network Localization," in *Proceedings of the 23^{rd} IEEE Computer Communications Conference (INFOCOM '04)*, Hong Kong, China, 2004.
[19] N. Sastry, U. Shankar, and D. Wagner, "Secure Verification of Location Claims," in *Proceedings of the 2^{nd} ACM Workshop on Wireless Security (WiSe '03)*, San Diego, USA, 2003.
[20] S. Čapkun and J.-P. Hubaux, "Secure Positioning in Wireless Networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, 2006.
[21] W. Pires, T. H. de Paula Figueiredo, H. C. Wong, and A. A. Loureiro, "Malicious Node Detection in Wireless Sensor Networks," in *Proceedings of the 18^{th} International Parallel and Distributed Processing Symposium (IPDPS '04)*, Santa Fe, USA, 2004.
[22] D. Liu, P. Ning, and W. Du, "Detecting Malicious Beacon Nodes for Secure Location Discovery in Wireless Sensor Networks," in *Proceedings of the 25^{th} International Conference on Distributed Computing Systems (ICDCS '05)*, Columbus, USA, 2005.
[23] L. Doherty, L. E. Ghaoui, and K. S. J. Pister, "Convex Position Estimation in Wireless Sensor Networks," in *Proceedings of the 20^{th} IEEE Computer Communications Conference (INFOCOM '01)*, Anchorage, USA, 2001.
[24] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust Distributed Network Localization with Noisy Range Measurements," in *Proceedings of the 2^{nd} International Conference on Embedded Networked Sensor Systems (SenSys '04)*, Baltimore, USA, 2004.
[25] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from Connectivity in Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, 2004.
[26] S. Ray, R. Ungrangsi, F. de Pellegrini, A. Trachtenberg, and D. Starobinski, "Robust Location Detection in Emergency Sensor Networks," in *Proceedings of the 22^{nd} IEEE Computer Communications Conference (INFOCOM '03)*, San Francisco, USA, 2003.
[27] K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan, "Ecolocation: A Sequence Based Technique for RF-only Localization in Wireless Sensor Networks," in *Proceedings of the 4^{th} International Conference on Information Processing in Sensor Networks (IPSN '05)*, Los Angeles, USA, 2005.
[28] L. Lazos and R. Poovendran, "SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks," in *Proceedings of the ACM Workshop on Wireless Security (WiSe '04)*, Philadelphia, USA, 2004.
[29] L. Lazos, R. Poovendran, and S. Čapkun, "Rope: RObust Position Estimation in Wireless Sensor Networks," in *Proceedings of the 4^{th} International Symposium on Information Processing in Sensor Networks (IPSN '05)*, Los Angeles, USA, 2005.
[30] S. Misra, G. Xue, and S. Bhardwaj, "Secure and Robust Localization in a Wireless Ad Hoc Environment," *IEEE Transactions on Vehicular Technology*, 2008.
[31] S. Zhong, M. Jadliwala, S. Upadhyaya, and C. Qiao, "Towards a Theory of Robust Localization Against Malicious Beacon Nodes," in *Proceedings of the 27^{th} IEEE Computer Communications Conference (INFOCOM '08)*, Phoenix, USA, 2008.
[32] S. Slijepcevic, S. Megerian, and M. Potkonjak, "Location Errors in Wireless Embedded Sensor Networks: Sources, Models, and Effects on Applications," *SIGMOBILE Mobile Computing and Communications Review*, 2002.
[33] Theodore S. Rappaport, *Wireless Communications: Principles and Practice, 2^{nd} Edition*. Pearson Education, Inc., 2003, ch. Mobile Radio Propagation: Large-Scale Path Loss.
[34] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic Fine-grained Localization in Ad-Hoc Networks of Sensors," in *Proceedings of the 7^{th} Annual International Conference on Mobile Computing and Networking (MobiCom '01)*, Rome, Italy, 2001.
[35] L. Girod and D. Estrin, "Robust Range Estimation using Acoustic and Multimodal Sensing," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, USA, 2001.

Murtuza Jadliwala is a senior researcher at the Laboratory for computer Communications and Applications (LCA), Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. He received his BE (2000) from Mumbai University, India, and his MS (2004) and PhD (2008) from State University of New York at Buffalo, all in Computer Science. His research interests are in secure and robust communication-based services in wireless networks, privacy, combinatorial optimization and approximation algorithms.

Sheng Zhong is an assistant professor at State University of New York at Buffalo. He received his BS (1996) and ME (1999) from Nanjing University, and his PhD (2004) from Yale University, all in Computer Science. His research interests are in non-cooperative behavior, especially economic incentives and data privacy in mobile computing.

Shambhu Upadhyaya is a Professor of Computer Science and Engineering at the State University of New York at Buffalo where he also directs the Center of Excellence in Information Systems Assurance Research and Education (CEISARE), designated by the National Security Agency and the Department of Homeland Security. Prior to July 1998, he was a faculty member at the Electrical and Computer Engineering department. His research interests are information assurance, computer security, fault diagnosis, fault tolerant computing, and VLSI testing.



Chunming Qiao is a Professor at University at Buffalo (SUNY), where he directs the Lab for Advanced Network Design, Evaluation and Research (LANDER). He is an editor of several journals and magazines including IEEE/ACM Transactions on Networking (ToN),IEEE Transactions on Parallel and Distributed Systems (TPDS), and a guest editor for several issues of IEEE JSAC and ACM MONET. He chaired the IEEE Technical Committee on High-Speed Networks, and is the founding chair of the new IEEE sub-TC on integrated optical and wireless networks.

Jean-Pierre Hubaux joined the faculty of EPFL in 1990. His research activity is focused on wireless networks, with a special interest in security and cooperation issues. In 1991, he designed the first curriculum in Communication Systems at EPFL. He was promoted to full professor in 1996. He is co-founder and chairman of the steering committee of the ACM Conference for Wireless Network Security (WiSeC). He is also the chairman of the steering committee of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc). He is a member of the Federal Communications Commission (ComCom), the "Swiss FCC". He held visiting positions at the IBM T.J. Watson Research Center and at UC Berkeley.