GRADIENT DESCENT APPROACH FOR SECURE LOCALIZATION IN RESOURCE CONSTRAINED WIRELESS SENSOR NETWORKS

Ravi Garg, Avinash L. Varna, and Min Wu
{ravig, varna, minwu}@umd.edu
University of Maryland, College Park, MD, USA.

ABSTRACT

Many sensor network related applications require precise knowledge of the location of constituent nodes. In these applications, it is desirable for the wireless nodes to be able to autonomously determine their locations before they start sensing and transmitting data. Most localization algorithms rely on anchor nodes whose locations are known to determine the positions of the remaining nodes. In an adversarial scenario, some of these anchor nodes could be compromised and used to transmit misleading information aimed at preventing the accurate localization of the remaining sensors. In this paper, a computationally efficient algorithm to determine the location of sensors that can resist such attacks is described. The proposed algorithm combines gradient descent with a selective pruning of inconsistent measurements to achieve good localization accuracy. Simulation results show that the proposed algorithm has performance comparable to existing schemes while requiring less computational resources.

Index Terms— Secure localization, Wireless sensor networks, Gradient descent.

1. INTRODUCTION

Wireless sensor networks (WSNs) are expected to form the backbone of future intelligent networks for a broad range of applications such as military surveillance, traffic monitoring, wildlife monitoring, underwater surveillance and habitat monitoring. In many of these applications, sensor nodes are deployed within an area using mechanisms such as helicopters and road vehicles, where the sensor locations are often not completely deterministic or known a priori. An important first step in setting up a sensor network is to accurately determine the position of the individual nodes, which is referred to as localization.

Most localization schemes rely on a set of beacon or anchor nodes with known location information to identify the positions of other nodes. In adversarial scenarios, some of these anchor nodes may be compromised by an adversary and used to transmit false information to prevent accurate localization of the remaining nodes and thus prevent the entire network from properly functioning. Hence, there is a need to design secure localization algorithms that are robust to such intentional attacks and determine the position of sensor nodes in the presence of adversaries. At the same time, as the sensors have limited memory and computational resources, these secure localization algorithms should be computationally efficient.

The problem of secure localization in WSNs has received a good amount of interest from the research community in recent years. Liu et al. proposed a greedy approach to find the most consistent location using data received from the anchor nodes [1]. They also proposed a voting based scheme in which the localization area is divided into grids and each grid point receives votes based on its distance from the anchor node and the distance measurement. Similar kind of voting approach with the help of sectored antennas and beacon nodes was proposed by Lazos et al. [2]. Li et al. proposed a Least Median Square approach to solve the localization problem by finding the median of residues for cases where less than 50% nodes are malicious [3]. Most of these methods have been shown to localize the node with small error in the presence of certain maximum percentage of malicious users. However, the memory requirement and computational cost of running these algorithms is still high and needs improvement.

In this paper, we propose a new method based on gradient descent approach to solve the problem of secure localization. This method works in two stages. In stage 1, gradient is calculated using data from all the anchor nodes. In stage 2, selective pruning of inconsistent measurements is done to mitigate the effect of malicious nodes on gradient calculations. As will be shown later in the paper, the proposed method can achieve localization accuracy comparable to existing algorithms in a computationally efficient manner.

2. PROBLEM DESCRIPTION

In this section, we describe the problem setup for secure localization. Let L be the number of anchor nodes whose location is known. Such anchor nodes may represents nodes that are deployed at known locations and serve to bootstrap the localization of the other sensors in the network. Once a node has determined its own location, it can function as an anchor node for localizing the remaining nodes.

The wireless node to be localized receives the location of each of the anchor nodes (x_k, y_k) and an estimate of the distance (d_k) between the anchor node and itself. The d_k could be obtained through different techniques such as hop count [4], time difference of arrival, time of arrival or received signal strength. These distance measurements may be noisy in practice, and we model the measurement errors as additive Gaussian noise with zero mean and variance σ^2. Given the set

of measurements $\{(x_k, y_k, d_k)\}, k = 1, 2, \ldots, L$, an estimate for the node's location (x_0, y_0) can be obtained by solving the over-determined system of equations in a Least Square (LS) sense:

$$d_k^2 = (x_k - x_0)^2 + (y_k - y_0)^2 \qquad k = 1, 2, \ldots, L$$

In the presence of adversaries, compromised nodes may intentionally report wrong information about their (x_k, y_k, d_k) measurements. In such cases, the LS estimate may be quite far from the true location. Thus, we need secure localization algorithms that are resilient to such attacks. In this paper, we consider two kinds of attack scenarios by the compromised nodes - non-coordinated and coordinated attacks and propose an algorithm that is robust against these attacks.

Non-coordinated attacks: In non-coordinated attacks, the compromised nodes *independently* modify the distance estimates to prevent the localizing node from determining its position accurately. We model this scenario by adding a zero-mean uniform random variable with variance σ^2_{attack} to the actual distance estimate from each malicious node and provide this erroneous information to the localizing node.

Coordinated attacks: A stronger attack against the network can be launched by compromised nodes acting together to make a localizing node estimate its position as (x_{mal}, y_{mal}), where (x_{mal}, y_{mal}) is an arbitrary point determined by the attackers. We model this scenario by reporting the distance between (x_{mal}, y_{mal}) and (x_k, y_k) as the distance estimate d_k and characterize the strength of such a coordinated attack by the distance between the actual position and the position reported by the malicious nodes $d_a = \sqrt{(x_{mal} - x_{true})^2 + (y_{mal} - y_{true})^2}$.

3. PROPOSED METHOD

We propose a computationally efficient secure localization algorithm by combining gradient descent with a selection stage to filter the malicious measurements. We note that when the measurement noise is Gaussian and in the absence of malicious nodes, the probability that any given point (X, Y) is the true sensor location is given as

$$P(x_{true} = X, y_{true} = Y) \propto \exp\left\{-\sigma^{-2}\sum_{k=1}^{L}\left(d_k - \sqrt{(x_k - X)^2 + (y_k - Y)^2}\right)^2\right\}$$

The Maximum Likelihood (ML) estimate for the position can then be found by maximizing this probability, or equivalently, by minimizing the negative of the exponent. We adopt a gradient descent approach to perform the minimization and determine the ML estimate for the position of the sensor.

Conceptually, the algorithm can be visualized by drawing circles around the position of the anchor nodes (x_k, y_k) with radius d_k. As the true position of the sensor should lie close to all these circles, in each iteration, we move the estimated position closer to these circles. To aid in the description, we define a "force vector" to be the vector with direction along the line joining the current estimate and the position of the anchor node (x_k, y_k) and magnitude equal to the distance between the current estimate and the corresponding circle. The sum of these force vectors gives the overall gradient.

Input: L=number of anchor nodes; M= number of iterations; S={set containing all anchor nodes}; $\{(\mathbf{P_k}, d_k)\}$, where $\mathbf{P_k} = [x_k, y_k]^T, k = 1, 2, \ldots, L$
Output: Estimated coordinates $\mathbf{P_0}(M) = [x_0, y_0]^T$
Initialization: a random point $\mathbf{P_0}(0)$; $stage = 1$
for $i = 1 : M$ **do**
  for $j = 1 : L$ **do**
    $\mathbf{g_j} = \frac{d_j - \|\mathbf{P_0}(i-1)) - \mathbf{P_j}\|}{\|\mathbf{P_0}(i-1) - \mathbf{P_j}\|} \times (\mathbf{P_0}(i-1) - \mathbf{P_j})$;
  end
  if $(\|\mathbf{g}(i)\| < threshold) \parallel (stage == 2)$ **then**
    $stage$ = 2; //switch to selection stage
    S={set containing a fraction of minimum length force vectors};
  end
  $\mathbf{g}(i) = \sum_{j \epsilon S} \mathbf{g_j}$; //gradient
  Update: $\mathbf{P_0}(i) = \mathbf{P_0}(i-1) + \delta(i) \times \frac{\mathbf{g}(i)}{\|\mathbf{g}(i)\|}$;
end

Fig. 1. Gradient descent algorithm for secure localization.

We initialize the algorithm at a random point and in the ith iteration, we update the estimate by moving it a step size $\delta(i)$ in the direction of the gradient. This update rule results in a new estimate that has a higher probability of being the true location of the node. This gradient descent algorithm eventually converges to the LS estimate. As described previously, in the presence of adversaries, the LS estimate can have large errors. Hence, once the gradient descent algorithm converges, we switch to a selection stage in which some force vectors are pruned as discussed below.

Selection Stage: In our experiments, we observed that the estimate obtained from the first stage of the algorithm, which minimizes the sum of the distance from all the circles around the anchor nodes, is closer to the correct estimate as opposed to the position (x_{mal}, y_{mal}) chosen by the malicious nodes. Hence, the force vectors pointing towards the correct estimate will be of smaller length compared to those pointing towards the wrong position. We use this fact to prune out the malicious references and use the smaller force vectors to determine the direction of descent for the subsequent iterations. The final algorithm is shown in Fig. 1.

Discussion: The proposed method is faster than the Least Median Square and Voting based schemes. The gradient descent algorithm performs better than the voting based scheme [1] in time and memory complexity. To obtain better localization in the voting based scheme, the grid needs to be quantized more finely, leading to higher memory and computational requirements. In contrast, the proposed method calculates the distance of the current estimate from the anchor nodes and does not require much physical memory to

Table 1. Comparison of run-time complexity of different algorithms. (n_1 is the size of the grid in Voting scheme; n is the number of nodes in each of M_1 sub-sets of Least Median Square algorithm; L and M are the number of anchor nodes and iterations, respectively. The values of the parameters are described in section 4.)

Method	Run-time(in sec)	Complexity
Least Median Square	0.2811	O(nM_1L)
Voting based scheme	0.0343	O(n_1^2L)
Gradient Descent	0.0152	O(ML)

store the data. Similarly, the Least Median Squares [3] approach requires a certain minimum number of subsets of nodes to ensure that one estimate is the correct estimate with very high probability. The LS estimate is found for each of these subsets which is computationally expensive.

Table 1 shows the computational complexity for each algorithm and the average runtime in a set of experiments on the MATLAB platfrom. From this table, we see that the computational complexity of our method increases linearly with the number of iterations. In Voting based scheme, complexity increases with the square of the grid size. For Least Median Square, complexity depends on the number of sub-sets required which increases as the percentage of malicious nodes increase. Thus, computational complexity of our scheme is independent of the number of malicious nodes and the localization accuracy pertaining to grid size.

4. SIMULATION RESULTS

We experimentally compare our method with existing methods in the literature. We use similar simulation parameters as [1] to allow for comparison of the results. 30 anchor nodes are randomly deployed in an area of 60m $\times$ 60m. Without loss of generality, we assume that each adversary node modifies the d_k value of the (x_k, y_k, d_k) triplet, as modifying any other parameter can be transformed into an equivalent modification of the d_k value. The measurement noise standard deviation is set to be $\sigma = 2$ meters. We prune 50% of the force vectors during selection stage of our algorithm.

For the Least Median Square method, the number of subsets is 20 and the number of nodes in each subset is 4. For the LS solution, we have used the computationally efficient approximate Linear Least Square solution presented in [5].

We test our method using variable step size and fixed step size for the gradient descent. In the fixed step size version of the algorithm $\delta(i) = 1, \forall i$, while in the variable step size algorithm, $\delta(i)$ is a monotonic decreasing function starting from a high value and decreasing linearly towards zero as i increases. The number of iterations for the gradient descent algorithm is $M = 200$. The results shown are obtained by averaging over 1500 runs of simulations.

Figure 2 shows the convergence curves for 30% coordinated attack by malicious nodes for variable and fixed step size. Using a variable step size results in low error but slow convergence, while the fixed step size method shows a slightly higher error but converges faster. This is a well known phenomenon encountered in gradient descent algorithms. We also observe from the figure that the localization error suddenly drops at around iteration number 100 for fixed step size and at around iteration number 150 for variable step size. The reason for this behavior is that the algorithm reaches the selection stage at this point and the estimate starts to move away from the LS estimate towards the correct estimate by pruning the malicious force vectors in the selection stage of algorithm.

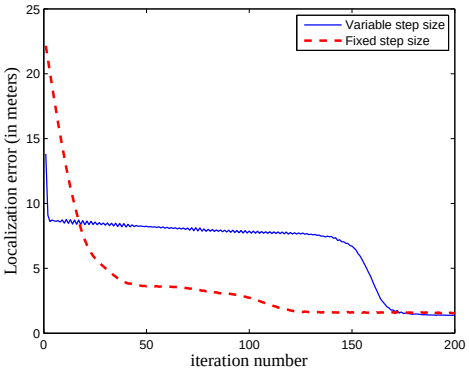


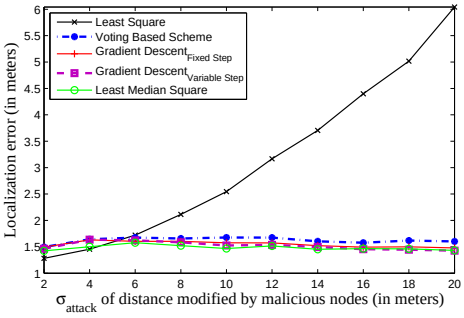
Fig. 2. Convergence curves for coordinated attacks by 30% of the nodes using fixed and variable step size. Malicious nodes attempts to shift estimate at a distance of 28 meters.

4.1. Non-coordinated attacks

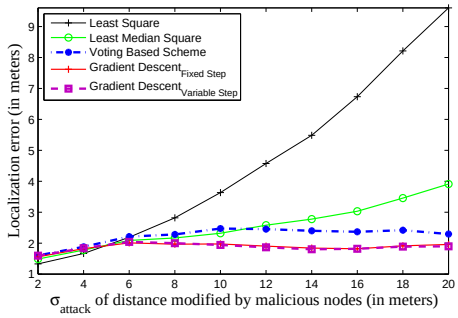
The localization accuracy achieved by the various secure localization algorithms under non-coordinated attacks with different parameters is shown in Fig. 3. Fig. 3(a) shows the localization error as a function of the noise standard deviation σ_{attack} added by the malicious nodes when 30% of the nodes are compromised. Fig. 3(b) shows the corresponding results when 60% of the nodes are compromised.

From the figure, we observe that the proposed scheme can tolerate more than 50% malicious nodes under non-coordinated attacks. The independent random perturbations in distance by the malicious nodes result in randomly oriented force vectors, so that they have a mutually canceling effect when summed up to compute the overall gradient. Hence, the proposed algorithm is robust against non-coordinated attacks. We also observe that the localization error does not increase as the attack noise variance σ^2_{attack} increases.

The localization error using our method is comparable to the other schemes. The slightly higher error in our method with fixed step size compared to the Least Median Square algorithm for 30% malicious nodes is due to the inherent error in gradient descent based algorithms that arises from a finite step size. For 60% malicious nodes, the Least Median Square method results in a localization error that increases as the attack strength increases because Least Median Square cannot tolerate attack by more than 50% of the users, but the pro-



(a) 30% malicious nodes



(b) 60% malicious nodes

Fig. 3. Comparison of different localization methods for non-coordinated attacks.

posed method can still localize the node with reasonable accuracy. The voting based scheme also gives good localization accuracy but the error is slightly higher than the gradient descent method because of the discrete nature of the grid points. The localization accuracy in the voting based scheme can be increased by finely quantizing the grid points at the cost of higher computation.

4.2. Coordinated attacks

Fig. 4 shows the localization error under coordinated attack by 30% of the nodes. The x-axis represents the distance d_a between the true location of the sensor and the point (x_{mal}, y_{mal}) chosen by the malicious nodes. From the figure, we observe that when the fraction of malicious nodes is 30%, the localization accuracy for all the methods except LS is almost the same. We obtained similar results when the fraction of malicious nodes is 35%. The localization error for the proposed gradient descent method is approximately 1 meter higher than the other techniques under this setting. The reason for this behavior is that as the percentage of malicious nodes increases, even the few anchor nodes whose distances from malicious position and true position are approximately the same can cause data to be more consistent with malicious position. Hence, gradient algorithm performs slightly worse than the Least Median Square and Voting based scheme for higher percentage of malicious nodes. In general, the proposed scheme can tolerate coordinated attacks when the fraction of correct measurements reported is larger than 50%.

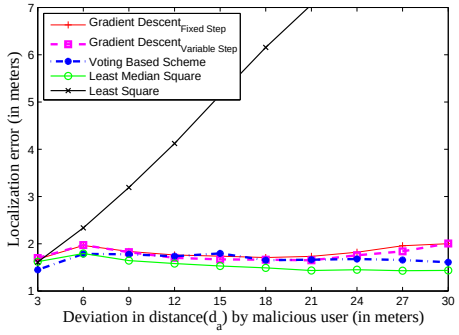


Fig. 4. Comparison of different localization methods for coordinated attacks by 30% of the nodes.

5. CONCLUSIONS

This paper describes a computationally efficient method for localization in wireless sensor networks in adversarial scenarios. The proposed method utilizes a gradient descent approach combined with pruning of inconsistent measurements to determine the location of a node in a given network. Simulation results show that the proposed method has localization accuracy comparable to other methods for coordinated attacks. Under non-coordinated attack by a large fraction of nodes, the proposed method has 10% less error compared to existing techniques, while requiring lesser computational resources.

In the future, we will explore robust fitting methods such as RANSAC [6] for secure localization. These algorithms have shown good results in fields such as computer vision, but need to be adapted for the low computational resources available on wireless sensors.

6. REFERENCES

[1] D. Liu, P. Ning, A. Liu, C. Wang, and W. Du, "Attack-resistant location estimation in wireless sensor networks," *ACM Trans. on Info. and Sys. Security*, vol. 11, no. 4, pp. 1–39, 2008.

[2] L. Lazos and R. Poovendran, "SeRLoc: secure range-independent localization for wireless sensor networks," in *Proc. of the 3rd ACM Wksp. on Wireless Security*, 2004, pp. 21–30.

[3] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *Int. Sym. on Info. Proc. in Sensor Networks*, 2005, pp. 91–98.

[4] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1-4, pp. 267–280, 2003.

[5] A. Savvides, C. Han, and M. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proc. of ACM MobiCom*, 2001, pp. 166–179.

[6] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.