An Active Global Attack Model for Sensor Source Location Privacy: Analysis and Countermeasures

Yi Yang Sencun Zhu Guohong Cao Thomas LaPorta
Department of Computer Science and Engineering,
Pennsylvania State University, University Park, PA 16802, USA,
Email:{yy5, szhu, gcao, tlp}@cse.psu.edu

***Abstract*—Source locations of events are sensitive contextual information that needs to be protected in sensor networks. Previous work focuses on either an active local attacker that traces back to a real source in a hop-by-hop fashion, or a passive global attacker that eavesdrops/analyzes all network traffic to discover real sources. An *active global* attack model, which is more realistic and powerful than current ones, has not been studied yet. In this paper, we not only formalize this strong attack model, but also propose countermeasures against it.**

As case studies, we first apply such an attack model to previous schemes, with results indicating that even these theoretically sound constructions are vulnerable consequently. We then propose a lightweight dynamic source anonymity scheme that seamlessly switches from a statistically strong source anonymity scheme to a k-anonymity scheme on demand. Moreover, we enhance the traditional k-anonymity scheme with a spatial l-diversity capability by cautiously placing fake sources, to thwart attacker's on-site examinations. Simulation results demonstrate that the attacker's gain in our scheme is greatly reduced when compared to the k-anonymity scheme.

I. Introduction

Source location privacy is an important privacy issue in both civilian and military applications of sensor networks, because the exposure of source location information may result in catastrophic damages. In an asset monitoring network [1], [2], when an endangered animal (e.g., panda) appears in the network, an event notification message will be delivered to the base station (BS). A nonconforming hunter may identify the source location and capture the animal by monitoring network traffic. In a battlefield scenario, the communication between soldiers and their surrounding sensors could reveal the positions of the soldiers, putting them in great danger as the opposing force may locate and accurately attack them.

Prior work on sensor source location privacy has explored two different adversarial models. In an *active local* attack model[1] [1], [2], [3], an attacker's hearing range is assumed to be comparable to that of regular sensors. The attacker tries to trace back to the real source in a hop-by-hop fashion, given that the real event source emits packets continuously for a period of time. Countermeasures in this category [1], [2] focus on confusing or misleading the attacker by introducing random or additional paths. Although such solutions have been shown to be effective, the local adversarial model is relatively weak. An attacker, with a hearing range more than three times of individual sensors, may locate the real source with a chance as high as 97% [1].

Recently, a *passive global* attack model has been studied [4], [5], [6], [7], where the attacker is assumed to be capable of monitoring all the network traffic by either deploying simple sensors covering the network or employing powerful site surveillance devices with hearing range no less than the network radius. With the collected network-wide traffic, the attacker can conduct traffic analysis to identify the potentially real sources. Under such a strong attack model, the corresponding countermeasures focus on making all sensors [4], [5], [6], [7] or k sensors [4] transmit (dummy) messages at the same or similar pattern to disguise the real source location. In general, such approaches are more robust to traffic analysis, at the cost of higher message overhead. This passive global attack model, however, is not realistic because it assumes that *an attacker merely monitors the traffic without taking any action.* Thus, although it is theoretically interesting, its real-world application is unclear. We believe in a real attack, an attacker will try to locate the real source by all means, as in the local attack model.

In this work, we focus on an *active global* attack model, in which the attacker is not only a global eavesdropper but also a realistic tracker that devises an optimal route to traverse suspicious spots one by one to find real events, under certain constraints, such as time, resource, and event duration. Compared with previous attack models, this is a more practical and powerful attack model. We formalize such a strong attack model, analyze it, and propose countermeasures against it.

In particular, we devise a dynamic programming algorithm and a greedy algorithm, based on which the attacker can derive the optimal traversal route to identify real events. To demonstrate both the procedure and the effectiveness of this attack model, we apply it to two existing schemes: a statistically strong source anonymity scheme [5] (referred to as SSSA scheme hereinafter) and a k-anonymity scheme [4]. We show that although the SSSA scheme provides strong source location privacy with statistical testing, under our attack model an attacker can gain some information about the locations of real sources when the message rate of a real event becomes high. The second scheme cannot provide actual k-anonymity because on average the attacker needs to check $k/2$ sources to find out the real one. Indeed, no schemes are perfectly secure under our attack model, because there is always some

[1]Note that our differentiation of "active" and "passive" attackers is based on whether the attacker actively takes actions to visit suspicious spots or not. This is different from the traditional one based on whether the attacker actively manipulates packets or not.

chance for the attacker to find out the real sources through his investigation, even if the attacker just randomly picks up places to check. For example, the constant-rate based schemes [4], [7] are just the special case of the k-anonymity scheme where k equals to n, the total number of cells in the network.

Our research is not only to demonstrate the power of this attack, but also to propose viable solutions to defend against such an attack. Specifically, as no schemes can completely prevent the real sources from being identified, our goal is to devise efficient mechanisms which will *minimize the location information disclosure*, under certain resource constraints of the attacker. We notice that while the SSSA scheme has the advantage of greatly reducing the transmission latency for real event messages compared to the constant-rate schemes, it also introduces *continuous, network-wide* dummy messages. For high message rate applications, the transmission overhead could be prohibitively high. As a tradeoff between privacy and performance overhead, the k-anonymity scheme could largely reduce the message overhead, not only because only k data sources are involved, but also because dummy messages are triggered by real events and stop once the events complete.

To leverage the advantages from both the worlds, we propose a lightweight dynamic source anonymity scheme that seamlessly switches from a low-rate SSSA scheme to a k-anonymity scheme on demand. Moreover, we enhance the k-anonymity scheme with the property of *spatial l-diversity* to maximize the attacker's cost. Our simulation results show that with our defense the attacker's gain can be much reduced while his cost is increased compared to the k-anonymity scheme.

The main contributions of this work are summarized below.

- First, we formalize a new attack model, where an active global attacker designs an optimal route to check suspicious spots in the whole network;
- Second, we apply this attack model to the existing source anonymity schemes and demonstrate their limitations;
- Third, to thwart the attack, we propose a new dynamic scheme that seamlessly transits from a low-rate SSSA scheme to a spatial l-diversity enhanced k-anonymity scheme.

The rest of this paper is organized as follows. The active global adversary model is built up in Section II. Case studies on existing source anonymity schemes are addressed in Section III. Then, the dynamic source anonymity scheme is discussed in Section IV. Finally, after describing the related work in Section V, we conclude this paper in Section VI.

II. An Active Global Adversary Model

In this section, we formalize the active global attack model and discuss details of the attacker's investigation. The attacker may employ a dynamic programming algorithm or a greedy algorithm to devise an optimal route for the investigation. We compare the results of these two algorithms through simulation and make clear the application scenario for each algorithm.

A. Modeling of Network

We consider a cell-based (or grid-based) network model. Deployment area of the network is partitioned into cells, which is the smallest unit of event detection: $\mathcal{N} = \{c_1, c_2, \cdots, c_n\}$, where n is the total number of cells. Every cell has a unique id $i(1 \leq i \leq n)$ and multiple sensors may reside in one cell. Each pair of sensors in neighboring cells could directly communicate with each other. A cell head, which is elected and rotated among all sensors in the cell, coordinates all the operations inside the cell. A base station (BS), connecting to the outside infrastructure such as the Internet, collects data from the network and reports them to a remote commander.

B. Modeling of Events

We assume that the total *information quantity* of a real event is $y_0(>0)$ and a real event will last for time $t_0(>0)$ once it happens. We model the information quantity of a real event at any time t as a function $f(t)$. In general, the choice of $f(t)$ is subject to the characteristics of the application. To be concrete, here we select a linear decrease function, as shown in Figure 1. If the attacker checks a real event after time t, the remaining information quantity could be modeled by:

$$f(t) = \begin{cases} y_0 - \frac{y_0}{t_0}t, & 0 \leq t \leq t_0 \\ 0, & t > t_0. \end{cases} \quad (1)$$

This means that if the attacker reaches the spot at the very first beginning of the real event then the attacker can get the maximum information y_0. The quantity of information that the attacker may obtain decreases after that. If the attacker reaches the real event spot after t_0, then he cannot obtain any information.

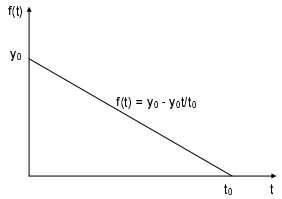


Fig. 1. Information quantity function of events.

C. Investigation of Attacker

Although an attacker may have resources to check all the cells one by one, this is not an intelligent choice because real events often last only for a short time period. If the attacker spends too much time on fake sources and thus reaches the real source too late, real events may have already disappeared. Hence, the attacker faces the following two challenges:

- First, how many suspicious cells to check?
- Second, what is their visiting sequence to maximize the attacker's gain in information quantity?

Next, we discuss how these challenges can be addressed. After observing and collecting network traffic for some time, the attacker first determines a suspicious level for each cell through traffic analysis. Then, the attacker decides a threshold. If a cell's suspicious level is higher than this threshold, this cell will be marked as a suspicious cell. The determination of this threshold value depends on many factors, e.g., the balance between the attacker's gain and the cost involved to achieve that gain. Note that even in the worst case for the attacker: every source have the same communication pattern, the attacker can still randomly select places for investigation and there is a certain chance for the attacker to find out the real sources.

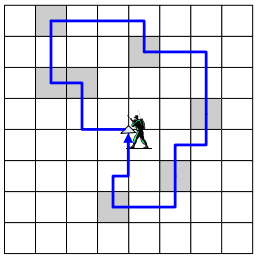


Fig. 2. The attacker traverses suspicious cells (highlighted as gray squares). We consider a network with $n(=\sqrt{n}\times\sqrt{n})$ cells that cover a rectangle deployment area. Each cell has a unique id, ranging from 1 to n.

Given the positions of all suspicious cells, the attacker optimizes the checking route. Clearly, the attacker's ultimate goal is to maximize the overall gain. Therefore, suspicious cells with higher suspicious levels should be checked at higher priorities. As shown in Figure 2, we assume that the attacker always starts from the center of the deployment area, traversing along a predetermined checking path under a specific velocity v. The main constraints of the attacker are time and resources. For each round of the real event investigation, there is a time limit τ, by which the attacker shall return to the starting point to start the next-round investigation based on newly collected data. τ could be the same as t_0 or other values determined by the attacker's resources.

Here we define *weighted gain*, which equals to the information gained from the suspicious cell if this cell is a real source *times* the probability of the cell being a real source. Assume there are totally s suspicious cells. The weighted quantity of information that the attacker could obtain from the jth $(1 \leq j \leq s)$ suspicious cell is:

$$\psi(j) \quad = \quad f(t_j) \cdot \xi_j, \tag{2}$$

where t_j is the time to reach the jth suspicious cell and ξ_j is the suspicious level of the jth cell. Note that $t_j = \sum_{k=1}^{j} \tau_k$, in which $\tau_k = \frac{distance(c_{k-1},c_k)}{v}$ is the time to travel from the (k-1)th cell to the kth cell. Therefore, the total information quantity that the attacker could obtain is

$$info_{total} \quad = \quad \sum_{j=1}^{s} \psi(j). \tag{3}$$

Given all the suspicious cells, intuitively, we can have a brute force method to design an optimal route for the attacker's investigation. In this brute force method, the attacker permutes all the possible traverse sequences and finds one with the maximum gain. For s suspicious cells, there are $s!$ permutations. For each permutation, the total number of summations is s. Hence, the time complexity of a brute force solution is $O(s * s!)$. Since the factorial time complexity in brute force is too high, in the following we discuss other two more efficient algorithms.

1) A Greedy Algorithm: The attacker prefers to checking the most suspicious cells while trying to maximize the total number of cells that he can check in a limited time τ. In a greedy algorithm (Algorithm 1), at his current location, each time when the attacker selects the next suspicious cell, he chooses the one with the maximum *ratio* of suspicious level to its distance from the current location. The greedy algorithm is efficient since it finishes in polynomial time. However, because every time a local optimum is chosen, there is no guarantee that a global optimum will be output from the greedy algorithm finally.

Algorithm 1 Attacker's Greedy Traversal Algorithm

Input: a set of suspicious cells: $S = \{1, 2, \cdots, s\}$; each cell i's corresponding suspicious level $\xi_i (1 \leq i \leq s)$; starting point $\mathcal{SP}$;
Output: whether there are real events in suspect cells;
Procedure:
1: current point is $\mathcal{SP}$;
2: time = 0;
3: **repeat**
4: select a cell c from S with maximum ratio of ξ_c/distance(current point, c);
5: go to cell c to check;
6: time += distance(current point, c)/v;
7: output whether there is real event in cell c;
8: current point is cell c;
9: $S = S - \{c\}$;
10: **until** $S = \emptyset$ or time$\approx \tau$

2) A Dynamic Programming Algorithm: We also propose a dynamic programming [8] based solution (Algorithm 2) for path selection. This algorithm could output a global optimal result in a relatively efficient way.

The basic idea of Algorithm 2 is as follows. Let S be a subset of $\{1, 2, \cdots, s\}$ and t_S is $\sum_{j \in S} \tau_j$. We denote $C(S)$ as the maximum gain that could be obtained by traveling all cells in S until time t_S. For $l \in S$, let $S - l$ denote a set obtained by removing element l from set S. Then, the following recurrence could be derived. When the size of S is 1, for any $l \in S$, we have

$$C(\{l\}) = \psi(l); \tag{4}$$

when the size of S is larger than 1,

$$C(S) = max_{l \in S}[C(S - l) + \psi_S(l)], \tag{5}$$

where the subscript S in $\psi_S(l)$ denotes the weighted information quantity obtained from cell l influenced by traversed cells in subset S prior to cell l in the sequence. That is, in an optimal traversal sequence for the cells with indices in S, a certain cell with index l must be the last one to visit and the remaining cells must be traversed in an optimal order in the time interval $[0, t_{S-l}]$. Then the overall maximum gain by such an ordering will be $C(S - l) + \psi_S(l)$. Taking the maximum over all the choices of l, we can derive the above equation (5).

Here is an example to illustrate the whole process (Figure 3). To simplify the presentation, we assume that the information quantity which the attacker could obtain is inversely proportional to the distance between two cells under consideration. Then, for each subset with size 2, there are two ways to traverse these two cells since each cell could be the last one in the sequence once. The traverse sequence with the maximum gain for every subset is recorded in a table[2]. Note that if there are equal values the attacker just randomly keeps one of the solutions (e.g., the first solution with the equal value). The procedure is the same for all the subsets with size 3. At

[2]In practice, if $\psi_S(l) = 0$, the attacker does not need to reach the cell since the weighted gain is zero. Here, we may still keep those values for comparison reason.

Algorithm 2 Attacker's Traversal Algorithm by Dynamic Programming

Input: a set of suspicious cells: $\{1, 2, \cdots, s\}$; each cell i's corresponding suspicious level $\xi_i (1 \leq i \leq s)$; starting point $\mathcal{SP}$;
Output: whether there are real events in suspicious cells;
Procedure:

$x = 2^s - 1$;
$\{P_1, P_2, \cdots, P_x\} = permutation(s)$;{function $permutation$ returns a set, including all the subsets of $\{1, 2, \cdots, s\}$ except Φ, sorted by increasing sizes}
for $i = 1$ to x **do**
  $S = P_i$;
  if $size(S) = 1$ **then**
    $C(S) = \psi(element(S))$;{function $element$ returns the element in the set}
    continue;
  else
    $C(S) = 0$;
  end if
  for all l such that $l \in S$ **do**
    if $C(S - l) + \psi_S(l) > C(S)$ **then**
      $C(S) = C(S-l)+\psi_S(l)$;{$\psi_S(l) = f_S(t_l) * \xi_l$ is the weighted information quantity}
    end if
  end for
end for

for $j = s$ to 1 **do**
  $S' = \{1, 2, \cdots, j\}$;{trace back to obtain the optimal traverse path}
  for $k = 1$ to j **do**
    if $C(S') = C(S' - k) + \psi_{S'}(k)$ **then**
      $next_j = k$;
    end if
  end for
end for

for $i = 1$ to s **do**
  if $v(\tau - \sum_{k=1}^{i} \tau_k) \leq distance(\mathcal{SP}, i)$ **then**
    break;
  else
    move to cell $next_i$;{traverse following the optimal path}
    output whether there is a real event happening in cell $next_i$;
  end if
end for
move back to the starting point $\mathcal{SP}$;

last, for the subset $\{1, 2, 3, 4\}$, there are four ways for traverse sequence. Similarly, the sequence with the maximum gain is put into the table. The current l, which is 1, will be the last cell for the attacker to visit. $S - l$ is $\{2, 3, 4\}$. By looking up the table, the attacker may find that when $l = 2$ the subset of $\{2, 3, 4\}$ has the maximum gain of 34. Hence, 2 is the second to the last cell to visit. Through using the same method to trace back, the attacker obtains the optimum traversal sequence that is $(3, 4, 2, 1)$ at last.

We analyze the time complexity of Algorithm 2. The function $permutation$ returns $2^s - 1$ subsets, so the time complexity is $O(2^s)$. The time complexity for finding out the traversal sequence with the maximum gain is approximately $\sum_{k=1}^{s} k\binom{s}{k} = s2^{s-1} = O(s2^s)$. The time complexity for tracing back the optimum path is $\sum_{k=1}^{s} k = \frac{s(s+1)}{2} = O(s^2)$. Traversing along the optimal path takes $O(s)$. Therefore, the time complexity of Algorithm 2 is $O(s2^s)$. Although it is still exponential, it is better than the factorial time complexity of the brute force algorithm.

3) Simulation Results: We use simulations to compare the results of greedy algorithm (Algorithm 1) and dynamic programming algorithm (Algorithm 2). In the simulation setting [3], the total number of cells $num_{cell} = 100(10 \times 10)$. The number of suspicious cells that are checked is $s = 9$ and $t_0 = 50$, $y_0 = 50$.

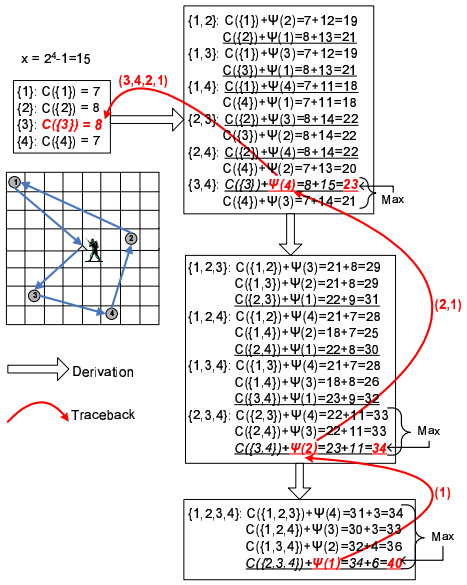


Fig. 3. An example of the traversal sequence for the attacker via Algorithm 2.

The results from our dynamic programming algorithm match those from the brute force method very well, which means Algorithm 2 can output the optimal traversal sequence accurately. When s is relatively small, e.g., less than 10, this algorithm could output results within ten seconds in a 2.0GHz processor PC. On the other hand, we find the greedy algorithm can generate sequences close to those from the brute force method. This algorithm can generate data in one second even when s is relatively large (e.g., in tens).

In conclusion, the dynamic programming algorithm is more accurate but slower, whereas the greedy algorithm quickly generates solutions close to the optimal ones. Which algorithm the attacker should choose depends on the relative criticality of accuracy and time consumption to the attacker. In the following sections, to achieve accuracy, we use the dynamic programming algorithm to find an optimal route for attacker's investigation.

III. Case Studies

To examine the impact of the proposed attack model, we apply it to two existing schemes, an SSSA scheme [5] and a k-anonymity scheme [4]), as case studies. Both schemes exhibit limitations under our attack model.

A. The SSSA Scheme

We first apply the active global attack model to the SSSA scheme [5]. We briefly introduce this scheme, followed by simulations to illustrate the attacker's investigation process as well as results.

[3]This is a default setting in the following simulations.

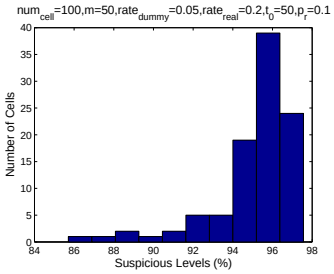


Fig. 4. Distribution of cells' suspicious levels when real message rate is high.

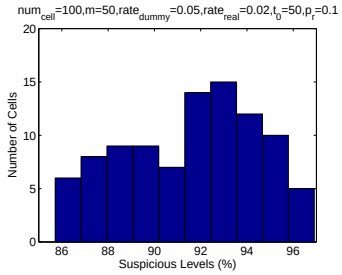


Fig. 5. Distribution of cells' suspicious levels when real message rate is low.

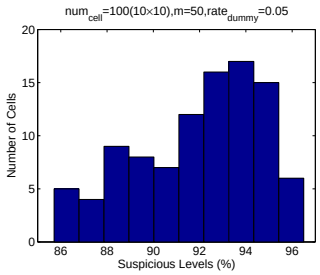


Fig. 6. Distribution of cells' suspicious levels when there are no real messages.

1) Scheme Overview: To hide real messages that report the occurrence of real events, a straightforward solution is to employ network-wide constant-rate traffic. Since every cell has the same transmission pattern, an attacker would not be able to distinguish the real sources from the fake ones. To reduce the network-wide message overhead, message transmission rate should be as low as possible. In this case, however, the transmission of real messages will need to be delayed more until the next transmission point. Therefore, there is an intrinsic difficulty to determine the message transmission rate because of the necessary tradeoff among privacy level, message overhead and real message latency.

In [5], a statistically strong source anonymity scheme (SSSA) is proposed to trade privacy level for reduced real message latency. In this scheme, real messages at the sources are transmitted as early as possible while the overall transmission pattern of a cell remains the same, in the sense that some existing well-known statistical testing methods [9], [10], [11] would not be able to detect the changes. As an instance, if the normal inter-message time intervals follow a predetermined exponential distribution, then after the changes (perturbation) made for real messages, the overall distribution looks the same under statistical tests. As such, the real message transmission latency is reduced and meanwhile a statistically strong source anonymity property for sensor networks is achieved.

2) Attacker's Detection: We discuss the attacker's operation in two steps: Step I identifying the suspicious cells and Step II investigating suspicious cells.

Step I: Identifying Suspicious Cells Intuitively, when real messages are relatively rare, it is unlikely for an attacker to notice the perturbation of the distribution due to the random nature of the variables forming the distribution. However, when a real event lasts for some time $t_0(t_0 > 0)$ or when an event report has to be divided into multiple packets (in Mica motes, the typical packet size is only 36 bytes), a real source will need to transmit multiple real messages continuously. Since the SSSA scheme tries to reduce the waiting times of multiple real messages, our intuition is that the the actual distribution will become farther and farther away from the ideal one. In [5], Anderson-Darling test [9] and Kolmogorov-Smirnov Test [10] are employed for goodness of fit test and Sequential Probability Ratio Test (SPRT) [11] is proposed for mean test. Although under these test models the SSSA scheme has been shown to be statistically strong, an attacker may apply some other testing methods. To identify suspicious cells, the attacker may check continuous small message time intervals from each cell and quantify its suspicious level.

Small message time intervals are defined as those smaller than the mean. If the number of continuous small message time intervals is larger than a threshold, say three, then it is called a *cluster* and the number of continuous small time intervals is called *cluster size* n_s. Denote the number of clusters as n_c. An attacker may infer the suspicious level of a cell based on $x = \sum n_s * n_c$. More formally, to normalize the suspicious level of a cell into the $[0, 1)$ range, the attacker may construct a suspicion function $g(x)$ (Figure 7) with x as input parameter:

$$g(x) = 1 - \frac{1}{x+1}, 0 \leq g(x) < 1, x \geq 0. \quad (6)$$

For example, if message time intervals from a cell has two clusters of size five, then the suspicious level of this cell is quantified as 90.9%; suppose message time intervals from another cell has three clusters of size six, then the suspicious level of this cell will be quantified to 94.7%.

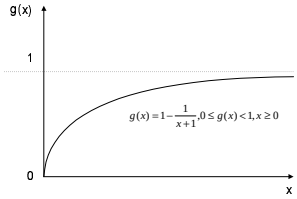


Fig. 7. Suspicion function to evaluate a cell's suspicious level.

We use simulations to check the distribution of cells' suspicious levels under high and low real message rates, with results shown in Figure 4 and Figure 5, respectively. In the simulation, the dummy message rate is 0.05. The probability for a cell to become the real source is 0.1. The attacker keeps the most recent 50 message time intervals from each cell. When real message rate is high (i.e., $rate_{real} = 0.2$), in most cases the suspicious levels of cells range from 94% to 97%, because the suspicious levels of real sources are normally high. On the other hand, when real message rate is low (i.e., $rate_{real} = 0.02$), The suspicious levels of cells are more uniformly distributed over the whole range. The attacker is able to determine a threshold accordingly. Considering the cost in checking suspicious cells, the attacker may want to control the number of suspicious cells to be checked as a relatively low value, e.g., about 10 out of 100 cells. In this case, 96% may be a good choice for the threshold.

In addition, we compare the distribution of suspicious levels under low real message rate with that when there are no real messages (i.e., when no events happen and all messages are

dummy). Our purpose is to show the false positive of the attacker's detection. From Figure 5 and Figure 6, we can observe that there are no big difference between these two figures. Thus, the attacker will not gain more from the low-rate SSSA scheme than a perfect secure scheme. Therefore, in our following dynamic source anonymity scheme, we switch from a low-rate SSSA scheme to a k-anonymity scheme.

Step II: Investigating Suspicious Cells After identifying the suspicious levels of cells in Step I, the attacker will arrange an optimal route to check these suspicious cells in Step II. In this section, we use simulations to check the attacker's gain in the SSSA scheme.

We first evaluate the attacker's gain and cost as a function of s, the number of suspicious cells to be checked. As shown in Figure 8, when the number of suspicious cells checked is increased from 3 to 9, the attacker's maximum gain is increased from 124.9 to 342.2. This is because if the attacker checks more suspicious cells a real source is more likely to be discovered. On the other hand, we observe that the attacker's traveling distance also increases from 15.1 to 29.5. This indicates that if the attacker wants to increase the maximum gain by checking more suspicious cells his cost will also increase. In practice, the attacker can decide a maximum traveling distance according to the maximum cost the attacker is willing to pay.

Next, we show how the attacker's gain varies with the real message rate in Figure 9. In the simulation, the probability of a cell being a real source is 0.1, and the average message rate of each cell is 0.05. We observe that when under a fixed t_0, the attacker's gain increases with the real message rate. This is because of two factors. First, with more real messages, more clusters will be observed in a real source and hence the suspicious levels of the real sources will increase. Second, s, the number of suspicious cells that are actually checked by the attacker, could also increase as more cells have a suspicious level over the threshold value. We also observe that the attacker's information gain increases with real event's duration t_0. This is because the remaining information $f(t)$ increases with t_0 at the same t.

From the above simulation results, we can see that the SSSA scheme, a theoretically sound privacy scheme, exhibit some limitations under the active global attacker model: the information quantity that the attacker could obtain increases with the real message rate. To address this problem, the SSSA scheme has to increase the overall message rate (including dummy and real ones). In other words, dummy message rate has to be adjusted to a larger value to cover the real messages, resulting in potentially prohibitively high message overhead for resource constrained sensor networks. Hence, the SSSA scheme is best applicable when real message rate is low.

B. The k-Anonymity Scheme

Next we apply the proposed active global attack model to the k-anonymity scheme [4]. We first briefly introduce this scheme, then check the active global attacker's gain in this scheme and compare it with the gain under a passive global attack.

1) Scheme Overview: k-anonymity used to be employed to improve the privacy of database without influencing data usability. The basic idea is that each individual data record can be released only when there are at least $k-1$ other distinct individuals whose associated records are indistinguishable from this record with respect to the quasi-identifiers [12], [13].

In [4], the idea of k-anonymity is adopted to provide source location privacy under global passive attacks. Basically, to disguise a real source, $k-1$ fake sources are randomly selected. All k sources start to transmit messages at the similar patterns to confuse the attacker. On one hand, to effectively hide a real source, k should be large enough in the k-anonymity scheme. On the other hand, a large k will lead to higher traffic overhead as more dummy messages will be introduced. One extreme case is when $k = n$, the total number of cells in the network. The highest level of privacy is achieved at the cost of the highest message overhead. Even so, we notice that the k-anonymity scheme has the advantage of on-demand traffic, compared to a constant-rate scheme [4]. That is, fake sources are introduced when real events occur and they are dismissed when real events complete. By adjusting the parameter k, we can have the flexibility of on-demand message overhead rather than a constant quantity of high-volume traffic.

2) Attacker's Detection: To attack the k-anonymity scheme, Step I for the attacker is to identify suspicious cells. This is quite simple, because it is obvious that the real source is one of the k cells that transmit messages to the BS. Also, according to the property of k-anonymity, ideally all these k cells have the same suspicious level. Then, the attacker may design an optimum route to check the k cells with these k cells as input. The only difference from the attack in the SSSA scheme is that the algorithm runs until the real source has been discovered (instead of time limit τ being reached). Since the real source could be any cell in the k sources, on average the algorithm will stop until $k/2$ sources have been traversed.

We use simulation to check the attacker's gain under the active global attack model. We also compare the result to that from the passive global attack model. Based on the property of k-anonymity, a *passive* attacker cannot differentiate the k sources, so the only thing he can do is to randomly select a source and claim it to be the real source. It is equivalent to pick up and check one out of k cells for the attacker. Actually, the probability for this cell to be the real source is only $\xi = 1/k$. Therefore, according to Equation (2), the weighted information quantity that a *passive* attacker could obtain is $\psi = y_0 \cdot \xi = y_0/k$. As shown in Figure 10, the gain of a passive attacker is much less than that of the active attacker. Clearly, we must take some steps to reduce the active attacker's gain.

IV. A Dynamic Source Anonymity Scheme

Our previous discussion showed that a low-rate SSSA scheme is robust to the active global attack. It also has low message overhead. However, it does not adapt well to the case of high-rate real messages. With more real messages, the buffer of the real source might be overflowed if the messages are not delivered promptly. Also, the delivery latency of all the real messages will become very high. In a k-anonymity scheme, as long as the transmission pattern of a high-rate real

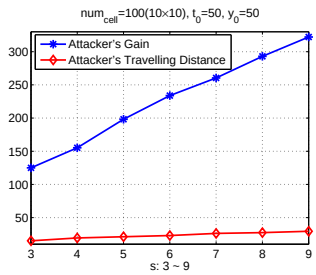


Fig. 8. The attacker's gain and cost as a function of s.

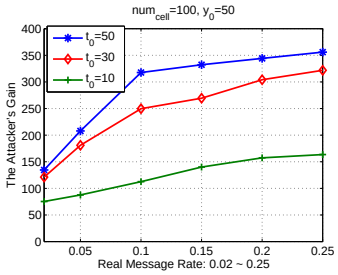


Fig. 9. The attacker's gain increases with the real message rate.

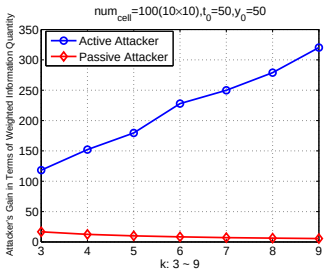


Fig. 10. Comparing the gains of passive and active attackers in k-anonymity scheme.

source can be estimated, $k-1$ fake sources can be dynamically selected. Based on these observations, we devise a dynamic source anonymity scheme, which seamlessly integrates the merits of both the SSSA scheme and the k-anonymity scheme. The basic idea is when the real event rate exceeds a threshold, the network switches to a k-anonymity scheme. The process starts with an event notification message from the real source to the BS. This message contains information such as how many packets are to be sent and the transmission pattern (e.g., constant rate). It is encrypted and looks the same as all the other messages in the network. The BS then selects $k-1$ fake sources and notify them to start transmissions at the similar patterns.

Although conceptually straightforward, our scheme has to answer the following questions.

- First, what is an appropriate switching point?
- Second, how to securely bootstrap the k-anonymity scheme?
- Third, how to enhance the security of our dynamic scheme against the active global attack?
- Fourth, how to evaluate the privacy level of our scheme?

The answer to the first question has to take into account many factors, including message overhead, latency, and privacy level. As message overhead is normally the biggest energy expenditure for sensor networks, here we will consider message overhead as the premiere criterion in determining the switching point. Note that our proposed techniques are independent of the way a switching point is selected. The second question exists because upon the occurrence of a real event, the event notification message is also under the monitoring of the attacker. If the attacker can figure out which message is an event notification message, he will be able to easily identify the real source. Note that in [4] it does not mention how to bootstrap the k-anonymity scheme. The third question arises because the k-anonymity scheme is not very robust to active attacks, as shown previously. Correspondingly, we will reduce the information gain of the attacker as much as possible. The fourth question has to be answered when evaluating our scheme.

To clearly explain our solutions to the questions, we will first need to make some formal definitions (Section IV-A), then describe the solutions in details in the remaining subsections.

A. Problem Definitions

Let $\mathcal{N}$ denote the set of all the n cells in the network. In our case, first, we have a definition of temporal k-anonymity as follows.

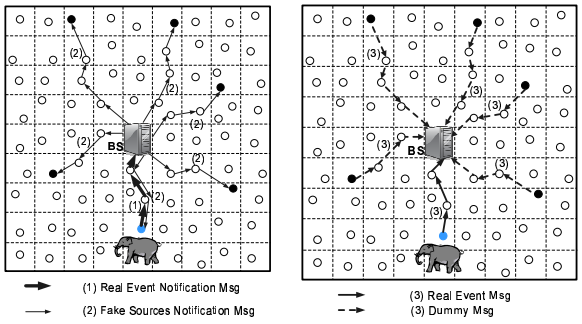
Definition 1: (Temporal k-anonymity). *A real source $r \in \mathcal{N}$ is temporal k-anonymous if there exist at least $k-1$ other cells $c_1, c_2, \cdots, c_{k-1} \in \mathcal{N}$ such that transmission patterns of all the sources $c_r, c_1, c_2, \cdots, c_{k-1}$ are indistinguishable from each other.*

Two transmission patterns are indistinguishable from each other, if their message transmission time intervals follow the same distribution with the same parameters. For example, if message time intervals from two cells follow an exponential distribution with the same mean, we can say that transmission patterns of these two cells are indistinguishable.

According to [14], k-anonymity alone is not sufficient to guarantee database privacy. For example, a person with background knowledge is able to figure out sensitive information from a table with k-anonymity property. l-diversity, which means that for each sensitive attribute there are at least l well-represented different values, is thereby presented to improve the diversity and also the robustness of data items against the above attack. Different from l-diversity in database privacy, we propose a definition of spatial l-diversity, which is adapted properly to our case, in order to improve location diversity of fake sources (and to decrease the attacker's gain). To be more specific, suppose the deployment area of the network is divided into $L(L > 0)$ partitions with almost the same size. Let $\mathcal{C}(\mathcal{C} \subseteq \mathcal{N})$ denote a set of cells and $\mathcal{P}(\mathcal{C})$ denote the total number of different partitions that cells in $\mathcal{C}$ are from. We then have the following definition of spatial l-diversity:

Definition 2: (Spatial l-diversity). *A set of cells $\mathcal{C}(\mathcal{C} \subseteq \mathcal{N})$ has the property of spatial l-diversity if $\mathcal{P}(\mathcal{C}) \geq l (0 < l \leq L)$.*

To quantify the level of source location privacy, next, we exploit the metric of normalized entropy proposed in [15]. This metric is defined based on probability: after the observation, the attacker assigns to the ith subject a probability p_i to be the source, with the sum of probabilities for all the subjects in the set of size n to be 1. For a given distribution of probabilities, the concept of entropy in information theory [16] provides a measure of the information contained in that distribution. Let X be the discrete random variable with probability mass function $p_i = Pr(X = i)$, where $i (1 \leq i \leq n)$ represents each possible value that X may take. The entropy of X is denoted as $H(X) = -\sum_{i=1}^{n} p_i \log_2(p_i)$. The maximum entropy of the scheme $H_M = \log_2(n)$, which could be achieved when all subjects have the same probability $\frac{1}{n}$ to become the source so that the attacker obtains no information about the source after observation. Therefore, the information that could be learned by the attacker is expressed as $H_M - H(X)$. Apparently,



(a) BS picks up fake sources. (b) Fake sources send bogus msgs.

Fig. 11. The notification of fake sources in the dynamic source anonymity scheme.

we want the entropy of the scheme $H(X)$ to be as large as possible so that the possible information that could be obtained by the attacker is minimized, since the maximum entropy H_M is fixed under a specific n. Accordingly, we have the following definition on privacy level.

Definition 3: (Privacy level). *The level of source location privacy is defined as* $l_p = \frac{H(X)}{H_M}$*, where* $H(X)$ *is the entropy of the scheme and* H_M *is the maximum entropy of the scheme.*

B. Scheme Description

1) Determining the Switching Point: To cover high-rate real messages, suppose the overall message rate (including dummy and real ones) in the SSSA scheme needs to be increased from λ_1 to λ_2. If $\lambda_2 - \lambda_1 > \delta_1$ where δ_1 is a predetermined threshold, the increase in message overhead is considered intolerable. Then the SSSA scheme needs to be switched to the k-anonymity scheme. Given the total number of cells n, we may determine a proper value for k. Assuming there is a system parameter $\delta_2 > 0$. Then, k needs to satisfy the following two constraints: $1 < k < n$ and $n\lambda_1 - k\lambda_2 > \delta_2$. Such a switch can reduce the overall message overhead significantly.

2) Secure Bootstrapping: The on-demand switching process is bootstrapped securely as follows. When there are no or low-rate real events, cells send dummy messages to the BS at a low rate following the SSSA scheme. After a real event is detected or a switching is needed, the cell detecting this event will send an event notification message to the BS. After the BS receives this message, it selects $k-1$ fake sources to generate bogus messages (note that fake sources are carefully chosen). Because the event notification message is just one message and it is easily hidden among the dummy traffic in the SSSA scheme. All the following data messages from the real source are covered by the bogus messages from the fake sources, so these k cells are indistinguishable from each other in the attacker's view. Hence, the k-anonymity scheme can be bootstrapped securely.

In more detail, as shown in Figure 11, after cell u detects a real event:

- Cell u notifies the BS that a real event happens (message 1 in Figure 11(a));
- After an appropriate delay ζ (which will be discussed later), the BS sends out notifications to $k-1$ fake sources as well as the real source, asking them to start transmitting messages (message 2 in Figure 11(a)) at the same rate or same pattern;
- All the cells receiving the notifications start to send messages to the BS (message 3 in Figure 11(b)).

We notice that when the BS receives an event notification message it should not send out k source nomination messages right away. Otherwise, a global observer will easily realize that the message coming to the BS just before BS emits those k messages corresponds to a real source. Therefore, in order to cover the real event notification message, the BS will need to wait for an appropriate time ζ before transmitting k source nomination messages.

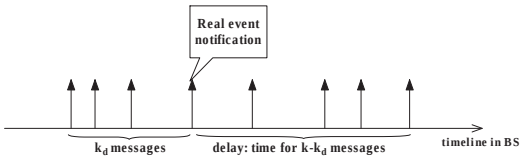


Fig. 12. The introduction of delay in the BS after a real event notification is received.

To determine ζ, the BS will need to first collect k messages from k different cells including the real source. As can be seen in Figure 12, to keep the k-anonymity property and make the notification message indistinguishable from other $k-1$ normal messages received by the BS, the BS picks k_d, a random number between 0 and k. It will wait until another $k - k_d$ messages are received. These k messages form an anonymity set and their sources are selected as the k sources. After that, it distributes source nomination messages to these cells. Hence, from the attacker's point of view all these k messages are equally likely to be a real notification message and their sources are equally likely to be a real source.

Given a proper k, the average delay ζ before the BS sends out k source nomination messages could be derived as follows. In the SSSA scheme, all the nodes send bogus messages with intervals following a mean $1/\lambda_1$. Then, in the BS, the event of incoming messages could be modeled as the sum of n distributions with an overall mean $\frac{1}{n\lambda_1}$. The delay is related to the time for $k - k_d$ messages received by the BS, which is $\frac{k}{2}$ on average since k_d is a random value between 0 and k. Therefore, the average delay in the BS is

$$\zeta = \frac{k}{2n\lambda_1}. \tag{7}$$

3) Introduction of Spatial l-Diversity: We notice that if it happens that all the k sources are close to each other then the attacker gain a lot within a short time. Therefore, in practice, the BS may selectively choose fake sources that are separated far away from each other. After the BS divides the deployment area into L logical partitions evenly with approximately the same size, we have an algorithm by which the BS achieves spatial l-diversity, as shown in Algorithm 3.

Note that in practice an application may have an upper bound on event notification delay, which is denoted as ω. Hence, the ideal l-diversity may not be attainable because it

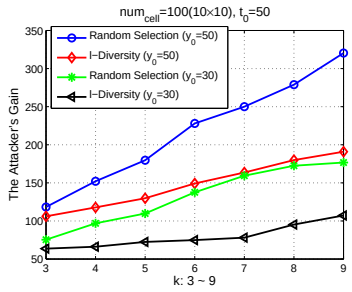


Fig. 13. The attacker's gain in the dynamic source anonymity scheme.

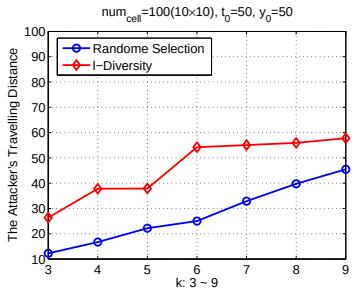


Fig. 14. The attacker's traveling distance in the dynamic source anonymity scheme.

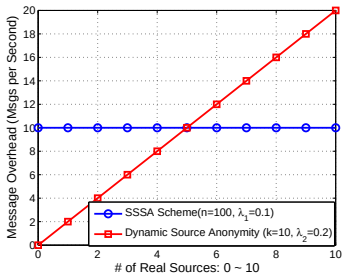


Fig. 15. Comparison of schemes in overhead.

Algorithm 3 The Spatial l-Diversity Algorithm by the BS

Input: a sequence of messages from different cells $\{msg_1, msg_2, \cdots\}$, each message carries such information as which cell and partition it is from;
Output: a set $\mathcal{C}$ of size k indicating where sources are, including the real source as a default item and $k-1$ fake sources;
Procedure:

```
C is initialized to be a set including the real source c_r and cells c_1, ..., c_{k-1} where the first k-1 messages are from;
calculate d = sum_distance(C);{function sum_distance() returns the sum of distances of cells in set C, starting from c_r;}
P is initialized to be a set including the partition p_r of real source c_r;
for j = 1 to k-1 do
    p_j is the partition of c_j;
    if p_j ≠ any partition from set P then
        put p_j into set P;
    end if
end for
repeat
    take an incoming message msg as input;
    obtain the cell c and partition p of msg;
    for i = 1 to k-1 do
        C' = swap(c_i, c);{replace c_i with c in set C}
        d' = sum_distance(C');
        if d' > d then
            d = d';
            C = C';{record set C with maximum distance}
            if p ≠ any partition from set P then
                put p into set P;
            end if
        end if
    end for
until (latency ω is reached)&&(size(P) ≥ l)
return current set C;
```

requires to receive k messages from l partitions (which could take a longer time). So the basic idea of the algorithm is as following. The BS keeps a set $\mathcal{C}$, which is initialized to include the real source and $k-1$ cells that the first $k-1$ messages originate from. Each time when a message is received by the BS, the BS tries to swap its source cell with every other cell except the real source in the current set, as long as such a swap could increase the overall distance of the cells in set $\mathcal{C}$. This procedure is repeated until the limit of latency ω is reached and the total number of partitions is larger than l. At this time, the current set $\mathcal{C}$ will be output.

The result of the algorithm is related to the values of ω and l. If they are larger, then k sources may be farther away from each other. Therefore, to reduce the attacker's gain and increase his traversal cost, the BS may wait for a longer time to choose the $k-1$ fake sources, so that all the k sources are from at least l different partitions.

We use simulation to verify the above statement. First, we check the attacker's gain as a function of the total number of sources k. We compare two options for fake source selection: random selection [4] and spatial l-diversity. In the simulation, the deployment area (10×10) is divided into nine partitions with approximately the same size. $l = k$ under different ks ($3 \leq k \leq 9$). As shown in Figure 13, the attacker's gain increases with k and y_0 (the total information quantity of a real event). Also, the technique of spatial l-diversity could largely reduce the quantity of information that the attacker obtains, compared with random selection.

Second, we check the attacker's traveling distance as a function of k. As shown in Figure 14, the attacker's traveling distance is increased by about 1.5 times because of the spatial l-diversity technique, compared with random selection of fake sources. When $k = 8$, the attacker's traveling distances in random selection of fake sources and spatial l-diversity are 39.7 and 55.9, respectively.

4) Analysis of Privacy Level: First, we have the following theorem about the initial privacy level of our dynamic scheme (before the attacker's check).

Theorem 1: The k-anonymity scheme with n cells and one real source has an initial privacy level of $l_p = \frac{\log_2 k}{\log_2 n}$, where $k-1$ is the number of fake sources.

Proof: Although there are n cells in the network, the number of active cells transmitting messages at a specific time is only k. Therefore, at any time, the attacker knows the probability for each of the rest $n-k$ cells to be the source is 0. Since the first message sent by the real source is buried in the dummy traffic and the paces of sending messages for all the fake sources as well as the real source are synchronized, the attacker cannot differentiate these k cells. Hence, from the attacker's view the probability for each of the k cells to be the real source is the same. The sum of these probabilities is 1, so every probability equals to $1/k$. Then, the entropy of this scheme

$$H(X) = -\sum_{i=1}^{n} p_i \log_2(p_i) = -\sum_{i=1}^{k} \frac{1}{k} \log_2(\frac{1}{k}) = \log_2(k),$$

whereas the maximum entropy of this scheme is $H_M = \log_2(n)$. Therefore, the initial privacy level for this scheme before the attacker's check is $\frac{H(X)}{H_M} = \frac{\log_2(k)}{\log_2(n)}$.

We notice that during the attacker's check the k-anonymity scheme has a dynamic privacy level as follows.

Corollary 1: During the attacker's check, the k-anonymity scheme with n cells and one real source has a dynamic privacy

level:

$$l_p = \begin{cases} 0, & \textit{if real source;} \\ \frac{\log_2(k')}{\log_2(n)}, & \textit{otherwise,} \end{cases} \quad (8)$$

where $k'(k' \leq k)$ is the number of sources that have not been checked by the attacker.

Proof: As presented in Theorem 1, the initial privacy level of the k-anonymity scheme is $\frac{\log_2(k)}{\log_2(n)}$. After the attacker checks one out of k sources, the privacy level of this scheme becomes:

$$l_p = \begin{cases} 0, & \text{if real source;} \\ \frac{\log_2(k-1)}{\log_2(n)}, & \text{otherwise.} \end{cases}$$

In general, when there are $k'(k' \leq k)$ sources that have not been checked by the attacker, all these k' sources have the equal probability $\frac{1}{k'}$ to be the real source. Hence, the entropy of the scheme at this time is

$$H(X) = -\sum_{i=1}^{k'} \frac{1}{k'} \log_2 \frac{1}{k'} = \log_2(k').$$

The privacy level is $\frac{\log_2(k')}{\log_2(n)}$. However, at any time when the attacker discovers the real source, the privacy level of this scheme becomes 0.

Clearly, the selection of k reflects a tradeoff between performance and privacy. A larger k means higher latency and message overhead. Simultaneously, a larger k also leads to higher privacy level based on Theorem 1. In practice, we can decide k according to the application's requirement in latency and overhead. After k is decided, actually the privacy level of the scheme has already been determined. The privacy level of the k-anonymity scheme depends on the ratio of $k(0 < k < n)$ and n. Since the privacy level of the SSSA scheme is close to 100%, normally, the privacy level of the k-anonymity scheme is lower than that of the SSSA scheme.

C. Discussions

1) Mobility of Object: In many cases, an object may go through several cells, which is referred to as a *handoff* problem. After an object moves to another cell, if the BS randomly chooses another $k-1$ fake sources, the attacker may be able to detect the real source. This is because the locations of the real sources that report the movement of this object actually form a trajectory, whereas the locations of the randomly chosen fake sources do not form a real trajectory. To address this problem, the next fake source should be picked up based on the position of the old fake source, to ensure that positions of these fake sources also form a seemingly real trajectory. This is a hard problem while implementation because building and simulating the object's mobility profile are still open research topics [4]. We may investigate more on this issue, e.g., how to solve the handoff problem in a secure and distributed manner, in our future work.

2) Multiple Real Sources: Considering the different mobility pattern of different objects, we cannot use the same set of fake sources for different real sources. The starting and ending time for different objects may be different, so using a fake source to serve multiple real sources is not feasible. Therefore, for each real source, the BS needs to assign a group of $k-1$ fake sources to simulate the real source. The maximum number of real sources that could be serviced at the same time will be $\lfloor n/k \rfloor$. The message overhead of our dynamic scheme increases with the number of real sources. At some point, it may be increased to a value that is more than that of the SSSA scheme (Figure 15). Therefore, the dynamic scheme is best applicable when there are few real sources continuously sending messages at a relatively high rate.

V. Related Work

In [17], techniques for hiding the base station (message destination) from an external global adversary are studied. In their schemes, secure multi-path routing to multiple destination base stations is designed to provide intrusion tolerance against isolation of base station and anti-traffic analysis is proposed to disguise the location of base station. [18] proposes a location-privacy routing protocol that provides path diversity combined with fake packet injection to protect receiver-location privacy. Complementary to their work, we are interested in source location privacy.

In [1], [2], a random walk based phantom routing scheme is proposed to defend against an external adversary who attempts to trace back to the data source in a sensor network, where sensor nodes report sensing data to a fixed base station for a certain period. A more recent work [3] proposes a two-way random walk algorithm, in which the routing path is obfuscated from both the source and sink. In [19], a path confusion algorithm is presented to increase source location anonymity. Note that these schemes work for a local adversary model. In our scheme, we consider a global attacker who has the view of all the network traffic.

[20] presents pDCS, a privacy-enhanced Data-Centric Sensor networks that offers different levels of data privacy based on different types of cryptographic keys. Under a global attacker model, in [21], two schemes are proposed. The first one is a ConstRate scheme; the second one is a k-anonymity based source-simulation scheme. Analytical results show how much communication overhead is needed to achieve a certain level of privacy. [7] addresses source location privacy against laptop-class attackers by proposing four schemes: naive, global, greedy, and probabilistic. In [6], to provide source event unobservability, schemes like ConstRate or ProbRate are used by the sensors. The focus of this work is to reduce the overall network traffic by proactively dropping dummy messages on their way to the BS.

[5] concentrates on reducing the latency of real messages under a global attacker model, by sending real messages as early as possible, in a way that the disturbance cannot be detected by available statistical tests. [22] also considers anonymous networking with minimum latency. Mixes are used for individual relays. The introduction of a limited number of dummy messages leads to a significant reduction in network latency. Information theoretical measurement is employed to analyze the relationship between anonymity level and latency. [23] provides temporal privacy protection for wireless sensor networks. In our work, we further improve the power of the attacker and consider a more realistic global attacker model in

which the attacker can go to suspicious spots and check real events by himself.

VI. Conclusion and Future Work

Previous work in sensor source location privacy mainly considers either a local tracker or a global eavesdropper in the attack model. We study a even more powerful and realistic attack model, in which a global attacker goes to suspicious spots and check real events by himself after monitoring all the network traffic. We formalize such a strong attack model and discuss countermeasures against it. An important future direction will be the development of a distributed way to solve the handoff problem under a mobile object in the dynamic source anonymity scheme. Other adversary models such as insider attackers are also of interest to us.

References

[1] C. Ozturk, Y. Zhang, and W. Trappe, "Source-location privacy in energy-constrained sensor networks routing," *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04)*, October 2004.

[2] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," in *Proceedings of 25th International Conference on Distributed Computing Systems (ICDCS 2005)*, June 2005.

[3] Y. Xi, L. Schwiebert, and W. Shi, "Preserving source location privacy in monitoring-based wireless sensor networks," in *in Proceedings of the 2nd International Workshop on Security in Systems and Networks(SSN '06)*.

[4] K. Mehta, D. Liu, and M. Wright, "Location privacy in sensor networks against a global eavesdropper," in *Proceedings of the IEEE International Conference on Network Protocols (ICNP 2007)*, October 2007.

[5] M. Shao, Y. Yang, S. Zhu, and G. Cao, "Towards statistically strong source anonymity for sensor networks," in *Infocom'08: the 27th Conference on Computer Communications*, April 2008.

[6] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao, "Towards event source unobservability with minimum network traffic in sensor networks," in *Proceedings of The ACM Conference on Wireless Network Security (WiSec)*, 2008.

[7] Y. Ouyang, Z. Le, D. Liu, J. Ford, and F. Makedon, "Source location privacy against laptop-class attacks in sensor networks," in *SecureComm*, 2008.

[8] M. Held and R. M. Karp, "A dynamic programming approach to sequencing problems," *J. Soc. Indust. Appl. Math.*, March 1962.

[9] T. W. Anderson and D. A. Darling, "A test of goodness of fit," *Journal of the American Statistical Association*, vol. 49, no. 268, pp. 765–769, December 1954.

[10] J. L. Romeu, "Kolmogorov-simirnov: A goodness of fit test for small samples," *START: Selected Topics in Assurance Related Technologies*, vol. 10, no. 6, 2003.

[11] A. Wald, *Sequential Analysis*. New York: J. Wiley & Sons, 1947.

[12] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[13] B. Gedik and L. Liu, "A customizable k-anonymity model for protecting location privacy," in *ICDCS*, 2005.

[14] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," in *ICDE 2006*.

[15] C. Díaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*, R. Dingledine and P. Syverson, Eds. Springer-Verlag, LNCS 2482, April 2002.

[16] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[17] J. Deng, R. Han, and S. Mishra, "Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks," *International Conference on Dependable Systems and Networks (DSN'04)*, June 2004.

[18] Y. Jian, S. Chen, Z. Zhang, and L. Zhang, "Protecting receiver-location privacy in wireless sensor networks," in *IEEE INFOCOM*, 2007.

[19] B. Hoh and M. Gruteser, "Protecting location privacy through path confusion," *securecomm*, vol. 0, pp. 194–205, 2005.

[20] M. Shao, S. Zhu, W. Zhang, and G. Cao, "pdcs: Security and privacy support for data-centric sensor networks," in *IEEE INFOCOM*, 2007.

[21] K. Mehta, D. Liu, and M. Wright, "Location privacy in sensor networks against a global eavesdropper," in *Proceedings of ICNP*, 2007.

[22] P. Venkitasubramaniam and L. Tong, "Anonymous networking with minimum latency in multihop networks," in *IEEE Symposium on Security and Privacy*, 2008.

[23] P. Kamat, W. Xu, W. Trappe, and Y. Zhang, "Temporal privacy in wireless sensor networks," in *ICDCS '07: Proceedings of the 27th International Conference on Distributed Computing Systems*, 2007.