On the Trade-Off Between User-Location Privacy and Queried-Location Privacy in Wireless Sensor Networks[⋆]

Ryan Vogt, Mario A. Nascimento, and Janelle Harms

Department of Computing Science, University of Alberta, Canada
{vogt, mn, harms}@cs.ualberta.ca

Abstract. By eavesdropping on a user's query in a sensor network, an adversary can deduce both the user's location and his/her area of interest. In many domains it is desirable to guarantee privacy of both places. Relying on the principle of k-anonymity, we propose an effective way to measure how well a disperse set of k queries protects the user's area of interest. However, issuing k queries instead of one facilitates the adversary determining the user's location. To address that issue, we define a quantitative measure of how much information the k queries leak about the user's location. Experiments reveal that how dispersed the k queries are has no effect on the privacy of the user's location. However, smaller k, randomized routing, and non-broadcast transmission improve the user's location privacy. We also show that compromising nodes in the user's network yields no significant advantage to the adversary over an eavesdropping strategy.

1 Introduction

Privacy is an important challenge in many wireless sensor network applications. Consider a sensor network where a user with a portable device interacts with the nodes, for instance, querying a sensor's data at a remote location. As a practical example, the sensor network could be a military one that troops need to query in preparation for an offensive. That query would be received by a nearby node and routed through the sensor network to the location of interest, then processed and returned using one of a large number of previously proposed algorithms. Given an assumed sensitive nature of the information returned by the query, encryption should be used to protect against eavesdropping. Either symmetric encryption or public-key encryption that is sufficiently inexpensive to be performed by a low-power sensor node [1] could be used. However, it is not just the information returned by the queried sensor node that has value to an adversary. By listening to network traffic, using either compromised nodes in the user's sensor network or eavesdropping nodes, an adversary could learn: (a) *where the queried sensor is*, thus providing significant insight into the user's intentions; and (b) *where the query originated*, thus revealing the user's current location. Investigating the *trade-off* between protecting these two locations is the main goal of this paper.

In the context of our aforementioned military scenario, we consider an adversary that wants to learn the user's current location or location of interest. In previous works,

[⋆] **Technical Report TR 09-07. April 2009. Dept. of Computing Science. University of Alberta. Canada. All rights reserved. Submitted for publication.**

a common assumption was the adversary was a mobile single physical agent able to move towards the user. We, on the other hand, assume that the adversary is not a single physical agent; rather, it is virtually present at different points of the network by virtue of being able to simultaneously eavesdrop on the on-going communication at those points. This "omnipresence" makes our model of an adversary stronger than a single mobile adversary. We also assume that the adversary has complete knowledge of all possible routes that a message could take between two sensors in the user's network (i.e., complete knowledge of the routing tables), as this information could be obtained through eavesdropping on legitimate traffic.

Considering this problem under this threat model, we offer the following contributions. We introduce novel metrics that enable the user to quantitatively evaluate how well he/she can protect the privacy of his/her location of interest by issuing k queries, one directed to the real location of interest and the others directed to $k-1$ distinct fake locations of interest. This technique for protecting location-of-interest privacy requires no changes, hardware nor software, to an existing sensor network; it is implemented entirely in software on the mobile device interacting with the sensor network, and introduces a multiplicative overhead of factor k in terms of communication in the sensor network. We also discuss in detail the trade-off yielded by the fact that issuing more queries generates more traffic from the user's current location, thus helping an adversary discover that current location. Our experimental results show that how well dispersed the k queries are has no effect on the privacy of the user's current location. Nonetheless, smaller values of k, randomized routing, and non-broadcast transmission between nodes can significantly improve the user's current-location privacy. Finally, we show the surprising result that an adversary who gathers information from compromised nodes in the user's network has no significant advantage over an eavesdropping adversary when broadcast transmission is used.

The remainder of this paper is structured as follows. Section 2 discusses related work on anonymous communication. We then formalize the metrics that we use in Section 3; these metrics define how well the user's location of interest and current location are protected by the k queries issued. Section 4 describes how the fake locations of interest should be chosen. We then present experimental results in Section 5 to illustrate how various parameters affect the user's privacy, before concluding and discussing possible future work in Section 6.

2 Related Work

There are known ways to achieve anonymity of two communication endpoints in some networks. The Tor network implementation of onion routing [2] has the source choose a random route to the destination. The source uses layers of public-key encryption, allowing each host in the route to see only the next host in the route, and none of the data being transmitted. However, this approach demands many public-key cryptography operations that may be too costly or even impossible for some sensor nodes. Finally, Tor does not protect against eavesdropping at communication endpoints, which is a significant threat in a broadcast medium. If an adversary were to overhear a query exiting the Tor network, all privacy about the location being queried would be lost.

Misra and Xue [3] look at anonymity specifically for sensor networks. They show how clusters of nodes can generate and share pseudonyms used as node identities when communicating with a sink. Their work is extended by Ouyang et al. [4] to account for shared keys being compromised. However, these works only look at nodes communicating with a sink, using pseudonyms known only to the endpoints to ensure that eavesdroppers will not know which node is sending information. These schemes are not applicable when nodes need to communicate with each other, as in our scenario.

Ozturk et al. [5] and Kamat et al. [6] examine a problem similar to our user's queries being tracked to their source. However, these papers have a different focus, namely an adversary that moves over time towards a source node that produces a continuous stream of data. They propose a solution called phantom routing, in which each epoch's data is routed in a random directed walk away from the source, before being flooded to the sink. This solution would not be appropriate for our scenario; e.g., a message intercepted during the directed walk phase carries enough information to immediately yield the number of hops and direction to the source of the user's query. A related problem, in which the adversary moves towards the receiver of sensor network traffic over time, is investigated by Jian et al. [7]. While these approaches are useful for protecting privacy in certain situations, they are not applicable to the situation described in this study, in which the adversary does not have to move towards the user's current location or location of interest. In a military scenario, for example, just learning either of these locations could be sufficient for the adversary.

In our work, we rely on obfuscating the user's location of interest by querying k locations, an idea inspired by Sweeney's concept of k-anonymity [8]. The goal then was to allow a data holder (e.g., a hospital) to release personal data to researchers such that no set of data could possibly belong to fewer than k individuals. To our knowledge, no one has investigated the trade-off imposed by preserving the privacy of both the location of interest as well as the user's location, using the notion of k-anonymity.

3 Privacy Metrics

We begin by defining our notation. The sensor network consists of a set $\mathcal{N}$ of nodes, where $|\mathcal{N}| = n$. The user will issue k queries, Q_i, each directed to a location L_i, where $\mathcal{L} = \{L_1, L_2, \ldots, L_k\} \subseteq \mathcal{N}$. One node, $L \in \mathcal{L}$, is the user's real location of interest, and the remaining $k - 1$ nodes are fake locations of interest.

To study how well this k-anonymity scheme preserves the privacy of the user's location of interest (LOI) and current location (CL), we require formal methods to measure the privacy levels that result from any given set of k queries. In the following section, we define the metric used to determine how well the LOI-privacy — and, more specifically, the area in which the LOI lies — is protected. We then define how well the privacy of the current location is protected in Section 3.2.

3.1 Privacy of the Location and Area of Interest

If it is equally probable that any of the k queried nodes is the real LOI, then the adversary cannot learn which of the k nodes queried is the real LOI. Given this assumption (which we revisit in Section 4), what is actually meant when we discuss protecting the privacy of the user's LOI? Consider the two scenarios in Figure 1. In both cases, the user's real LOI is the starred node in the northeast, and $k-1=3$ fake LOIs are chosen to disguise the real LOI. However, in Figure 1(a), all four LOIs are clustered in the east. While an adversary who overhears these queries would not know which node is of interest to the user, it would be obvious that the user is interested in the eastern region of the sensor network. In Figure 1(b), the four LOIs are dispersed throughout the sensor network, obfuscating the area of the network in which the user is interested. We need to define a measure of how dispersed the k LOI choices are. That is, we want to measure how well-protected the privacy of the user's area of interest (AOI) is.

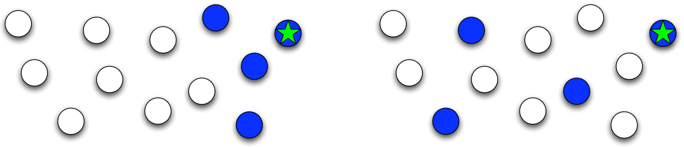

(a) A poor choice. (b) A good choice.

Fig. 1. A comparison of two choices for $k-1=3$ fake LOIs (dark nodes) given one fixed real LOI (starred node).

To measure how well the set $\mathcal{L}$ of k LOIs preserves the user's AOI-privacy, we define a function $\sigma(\mathcal{L},\mathcal{N})$ to measure how dispersed the LOIs are over the network $\mathcal{N}$. To allow for comparisons of different methods of choosing the fake LOIs over networks with different topologies, we normalize the score returned by σ. Let $\sigma_{min}(k,\mathcal{N})$ and $\sigma_{max}(k,\mathcal{N})$ be the minimal and maximal values returned by σ, over all $\binom{n}{k}$ possible sets of k LOIs. The normalized measure of AOI-privacy is defined as

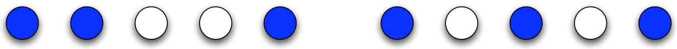
$$\mathrm{M}_{AOI}(\mathcal{L},\mathcal{N}) = \begin{cases} 1 & \text{if } \sigma_{min}(k,\mathcal{N}) = \sigma_{max}(k,\mathcal{N}) \\ \dfrac{\sigma(\mathcal{L},\mathcal{N}) - \sigma_{min}(k,\mathcal{N})}{\sigma_{max}(k,\mathcal{N}) - \sigma_{min}(k,\mathcal{N})} & \text{otherwise.} \end{cases}$$

The function σ must have the property that it returns large values for sets $\mathcal{L}$ with minimal clustering of the LOIs, and small values otherwise. It must also be easy to compute on the user's low-powered, mobile device prior to issuing a query. Next, we discuss some alternatives for σ.

Variance-Based σ. One straightforward approach is to compute a variance-like quantity for the positions of the nodes in $\mathcal{L}$, measuring the squared distance between LOIs:

$$\sigma(\mathcal{L},\mathcal{N}) = \sum_{i=1}^{k-1}\sum_{j=i+1}^{k} \mathrm{D}\left(L_i, L_j\right)^2 \quad ,$$

where D is the Euclidean distance between two nodes. However, this definition of σ does not penalize clustering properly. Consider the examples in Figure 2. The choice of fake LOIs in Figure 2(a) has two of the LOIs clustered together. In Figure 2(b) the three LOIs are dispersed evenly. However, the σ value for the first choice is higher.

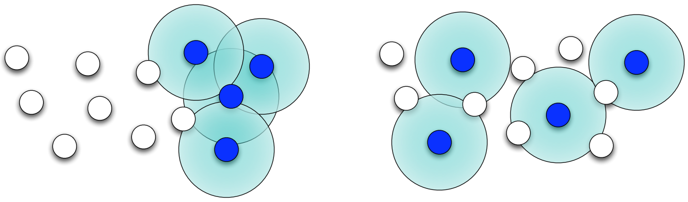


(a) A choice of LOIs that results in two clusters.

(b) A choice of LOIs that disperses the LOIs across the entire network.

Fig. 2. A comparison of two choices for $k = 3$ LOIs in a five-node sensor network.

Union of Circles σ. Another approach is to determine how much overlap exists among the regions around each LOI. Specifically, one can draw a circle of radius r around each LOI and let σ be the area of the union of the k circles. The more clustered the LOIs, the more overlap there would be among the circles, resulting in a smaller area. An example is presented in Figure 3. With k clustered LOIs, shown in Figure 3(a), there is significant overlap among the circles, unlike with the non-clustered choice shown in Figure 3(b).



(a) A choice of LOIs with significant overlap among the circles.

(b) A choice of LOIs with minimal overlap among the circles.

Fig. 3. A comparison of two choices for $k = 4$ LOIs in a sensor network, each surrounded by a circle with a fixed, arbitrary radius.

One way to choose r is to define d as the maximum distance between any two nodes in $\mathcal{N}$, then set $r = \frac{d}{2}$. This definition ensures that there is no overlap between the regions of the two sensor nodes that are farthest apart, but the regions surrounding any two sensor nodes that are closer together will overlap. While this σ properly penalizes clustering, computing σ is computationally expensive. A Monte Carlo algorithm for computing the area of the union of circles requires a large number of sample points. It may not be practical for the user to run an expensive algorithm to compute the AOI-privacy level of every set of k queries he/she poses. Even more expensive to compute are $\sigma_{min}(k, \mathcal{N})$ and $\sigma_{max}(k, \mathcal{N})$, though these values could be precomputed, provided the sensor network topology does not change.

Sum of Minima σ. Measuring the sum of minimum distances from each LOI to any other LOI penalizes clustering properly and is easy to compute. Formally, let

$$\sigma(\mathcal{L}, \mathcal{N}) = \sum_{i=1}^{k} \min_{j \neq i} \left\{ \mathrm{D}\left(L_i, L_j\right) \right\} \quad ,$$

where D is the Euclidean distance between two nodes. That is, the farther away each of the L_i are from each other, the higher the value of σ. Returning to the five-node examples in Figure 2, this σ returns a higher value in the scenario where the LOIs are non-clustered; and, it can be computed quickly, given the location of every node in $\mathcal{L}$.

Considering these three possible σ functions, we use the sum of minima function for the remainder of the paper. We chose this σ since we assume that the user, prior to issuing any set of k queries, will want to precompute the AOI-privacy that will result from the queries using a mobile device with limited processing power. If processing power were not a concern, the union of circles metric could be used instead.

3.2 Privacy of the Current Location

The previous section described a static measurement for how well k queries protected the user's AOI-privacy. How much information these queries leak about the user's current location (CL), on the other hand, cannot be analyzed statically, as that depends on how much information the adversary is able to overhear during the routing of the queries. To determine the CL-privacy that results from a set of queries, we will simulate the user issuing queries while malicious nodes (either compromised nodes in the user's network, or eavesdropping nodes outside the network) attempt to ascertain the origin of those queries.

This section demonstrates how an adversary could use the information captured by malicious nodes to narrow down the possible locations where the user could be. Central to this technique is the concept of a *possible route*. In the sensor network, which we assume to be connected, there must be a routing algorithm capable of routing messages from any source node to any destination node. Consider a route $\mathcal{R} = (N_1, N_2, \ldots, N_l)$, which is a sequence of l nodes. $\mathcal{R}$ is a possible route from N_1 to N_l if it possible that the routing algorithm used in the sensor network could have routed a message from N_1 to N_l along the path $N_1, N_2, \ldots, N_l$.

In this paper, we consider two routing algorithms: fixed shortest-path routing and random shortest-path routing. Both guarantee that any message from N_1 to N_l will arrive in the fewest possible hops. In random shortest-path routing, each node maintains a table indexed by the destination of a message, containing all possible next hops that the message could take to arrive in the fewest hops. When a message arrives at a node N_i destined for node N_l, N_i will look into its table at index N_l, and randomly choose one of the entries as the next hop. In fixed shortest-path routing, each node stores only a single next-hop choice for each possible destination. There is exactly one possible route from N_1 to N_l when fixed shortest-path routing is used, but there can be many possible routes between N_1 and N_l with random shortest-path routing.

Recall that the user uses a mobile device to communicate with a nearby sensor node, in order to route queries through the sensor network to the LOIs. The user sends the k queries, denoted $Q_1, Q_2, \ldots, Q_k$, to the closest sensor node, C. We assume that the mobile device and C communicate using low-power communication. That is, the adversary will only overhear communication between the mobile device and C if C is compromised. We also assume that the adversary knows k — the implication being that if the adversary overhears fewer than k of the user's queries, the adversary knows how many queries were not overheard. Each Q_i takes a route $\mathcal{R}_i$ through the network, starting at C and ending at L_i. Denote $\mathcal{R}_i$ as a sequence of nodes with length l_i, $\mathcal{R}_i = (N_{i,1}, N_{i,2}, \ldots, N_{i,l_i})$, where $N_{i,1} = C$ and $N_{i,l_i} = L_i$. The goal of the adversary is to determine C.

Clearly, a query message that is being routed from C to L_i cannot contain references to C, nor can a reply; otherwise, the adversary could easily determine C. Query messages contain four pieces of information in addition to the query itself: a unique query identifier for Q_i; the destination L_i (which may be the real location of interest or a fake one); the identifier for the node currently transmitting the query, $N_{i,j}$; and, the next hop in the route, $N_{i,j+1}$. When node $N_{i,j+1}$ receives the query, it remembers the previous node in the route for query Q_i, $N_{i,j}$. Replies to the query message contain only the query identifier for Q_i. When node $N_{i,j+1}$ receives a reply to the query, to be routed back to C, $N_{i,j+1}$ uses its memory to identify $N_{i,j}$ as the next hop in the reply path, and sends the reply to $N_{i,j}$ (without unnecessary information such as the identity of $N_{i,j}$ or $N_{i,j+1}$). However, we assume the worst case: the adversary is able to determine which sensor node is transmitting a reply message if that reply is overheard.

If a query message for query Q_i is overheard, the adversary learns one of the LOIs, L_i. Additionally, the adversary learns one hop that the query took along the route from C to L_i: $N_{i,j}$ and $N_{i,j+1}$, for $1 \leq j < l_i$, where j and l_i are unknown to the adversary. That is, the adversary learns two consecutive elements in the route, but neither their position in the route nor the length of the route. If a reply message for query Q_i is overheard, the adversary learns only a single node that was involved in the route: $N_{i,j}$, for $1 < j \leq l_i$, again with unknown j and l_i.

The adversary can also construct a list of sensor nodes that were certainly not involved in routing Q_i. Because the adversary has complete knowledge of every message that was routed through compromised nodes in the user's sensor network, the adversary knows which compromised nodes were not involved in routing Q_i. An additional consideration for the adversary is that the malicious nodes (either compromised nodes in the user's network, or the adversary's own nodes eavesdropping on the network) could monitor all communication by honest sensor nodes in their communication range. An adversary may conclude that if a malicious node that is monitoring an honest node within its range, H, did not hear H produce any traffic regarding query Q_i, then H must not have been involved in routing Q_i. However, the malicious node may not have heard a message transmitted by H due to interference or a collision. As such, we assume that the adversary will restrict the list of nodes that certainly were not involved with query Q_i to the set of compromised nodes in the sensor network that did not route Q_i.

The key insight for the adversary, having collected information on the routes of the queries, is that if no possible route from a sensor node N to the known destination of

Q_i, L_i, is consistent with the known information about Q_i, then N could not have been the origin of the query. Formally, a route $\mathcal{R} = \{N_1, N_2, \ldots N_l\}$ from N to L_i is *consistent* with that information, assuming the destination L_i of Q_i is known, if:

1. $\mathcal{R}$ is a possible route from N to L_i (i.e., $N_1 = N$, $N_l = L_i$, and the routing algorithm could have used this route);
2. For every query message about Q_i overheard by the adversary, sent by $N_{i,j}$ to $N_{i,j+1}$, there is some $k < l$ such that $N_k = N_{i,j}$ and $N_{k+1} = N_{i,j+1}$;
3. For every reply message about Q_i overheard by the adversary, sent by $N_{i,j}$, there is some $k > 1$ such that $N_k = N_{i,j}$; and,
4. No node that is known not to have routed Q_i appears in $\mathcal{R}$.

There is a second insight, about queries for which the adversary does not know the destination. The adversary may still know some information about such a Q_i (e.g., overheard reply messages, or knowledge about compromised nodes that did not route Q_i). Let $\mathcal{D}$ be the set of all possible destinations for the queries with unknown destinations. Specifically, $\mathcal{D}$ is the set of all nodes in $\mathcal{N}$ that are not compromised and are not the destination of a query with a known destination. Let $\mathcal{U}$ be the set of queries for which the destination is unknown to the adversary, where $0 \leq |\mathcal{U}| \leq k$. If it is not possible to assign a unique destination from $\mathcal{D}$ to each query in $\mathcal{U}$, in such a way as to ensure that there is a possible route from sensor node N to the destination of each query in $\mathcal{U}$ that is consistent with all of the known information about that query, then N cannot be the origin of the user's queries.

In summary, an adversary who eavesdrops on the user's network can overhear nodes that are involved in routing queries; an adversary who is able to compromise nodes in the user's network can also know for certain if those compromised nodes were not involved in routing a particular query. Pseudo-code for the algorithm the adversary can use to narrow down the user's possible current locations is presented in Figure 4. This algorithm will narrow down the possible current locations of the user to $\mathcal{P} \subseteq \mathcal{N}$. Denoting $\mathcal{N}_{honest} = \{N \in \mathcal{N} \mid N \text{ is not compromised}\}$, it is guaranteed that $1 \leq |\mathcal{P}| \leq |\mathcal{N}_{honest}|$. Normalizing between these two bounds defines the metric for evaluating CL-privacy,

$$\mathrm{M}_{CL}(\mathcal{P}) = \begin{cases} 0 & \text{if } |\mathcal{P}| = 1 \\ \dfrac{|\mathcal{P}| - 1}{|\mathcal{N}_{honest}| - 1} & \text{otherwise.} \end{cases}$$

4 Choosing the Fake LOIs

Given the metrics necessary to measure how well a set of k queries preserves the privacy of the user's area of interest (AOI) and current location (CL), we now investigate how the user should choose the $k - 1$ fake locations of interest (LOIs) to query, given one real LOI. Regardless of how the fake LOIs are chosen, recall our assumption that, from the point of view of the adversary, it is equally probable that any of the k queried nodes is the real LOI. Any implementation of k-anonymity for sensor network queries must take into account real-life limitations. For example, if the troops using a military sensor network

```
function NARROW-POSSIBLE-CLS
    if the node $C$ with which the user is communicating is compromised then
        return $\mathcal{P} = \{C\}$
    $\mathcal{N}_{honest} \leftarrow \{N \in \mathcal{N} \mid N$ is not compromised$\}$
    $\mathcal{P} \leftarrow \mathcal{N}_{honest}$
    for all queries $Q_i$ for which the destination is known do
        for all $p \in \mathcal{P}$ do
            if POSSIBLE-CONSISTENT($p, Q_i$) = FALSE then
                Remove $p$ from $\mathcal{P}$
    $\mathcal{U} \leftarrow \{Q_i \mid L_i$ is unknown$\}$
    $\mathcal{D} \leftarrow \{N \in \mathcal{N}_{honest} \mid N$ is not a destination for any $Q_i$ with known destination$\}$
    for all $p \in \mathcal{P}$ do
        $foundAssignment \leftarrow$ FALSE
        for $\mathcal{D}' \leftarrow$ all $\binom{|\mathcal{D}|}{|\mathcal{U}|}$ choices of $|\mathcal{U}|$ destinations from $\mathcal{D}$ do
            for all $|\mathcal{D}'|!$ assignments of nodes in $\mathcal{D}'$ as destinations for the queries in $\mathcal{U}$ do
                if POSSIBLE-CONSISTENT($p, Q$) = TRUE $\forall\ Q \in \mathcal{U}$ then
                    $foundAssignment \leftarrow$ TRUE
        if $foundAssignment$ = FALSE then              ▷ Note: vacuously never happens if $|\mathcal{U}| = 0$
            Remove $p$ from $\mathcal{P}$
    return $\mathcal{P}$

function POSSIBLE-CONSISTENT($src, Q$)
    $L \leftarrow$ the destination of query $Q$
    if $\exists$ a possible route from $src$ to $L$, consistent with all known information about $Q$ then
        return TRUE
    else
        return FALSE
```

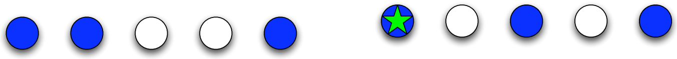
Fig. 4. Pseudocode for finding all possible current locations for the user.

queried one node at a location of strategic importance, but $k-1$ nodes at strategically irrelevant positions, the adversary could guess the real LOI with high probability. In this paper we assume a homogeneous environment where each node is equally important.

Another situation that could leak the real LOI is if the user issues multiple queries to the real LOI. Any algorithm used to choose the $k-1$ fake LOIs based on the real LOI is required to choose the same $k-1$ fake LOIs each time the user issues a real query. Otherwise, if the user first queries the k nodes in $\mathcal{L}_1$ then later issues queries to the k nodes in $\mathcal{L}_2 \neq \mathcal{L}_1$, and if the adversary can correctly guess that the user was issuing repeat queries to the same node (it is overly optimistic to assume otherwise), the adversary would learn that the real LOI is in $\mathcal{L}_1 \cap \mathcal{L}_2$.

Let F be the choice function that takes the real LOI L and returns a set $\mathcal{L}$ of k LOIs to query, with $L \in \mathcal{L}$. If F is deterministic, repeat queries are of no concern, since $\mathrm{F}(L) = \mathcal{L}$ for each call to $\mathrm{F}(L)$. If F is a random function, taking L and a random seed s_L as parameters (and using a cryptographically secure pseudo-random number generator [9]), the user should encapsulate F in a deterministic function. For example, the user could call $\mathrm{F}'(L) = \mathrm{F}(L, s_L)$, where $s_L = \mathrm{H}(s \circ L)$ is computed using a stored secure seed, s, and cryptographic hash function H, with $\circ$ representing concatenation.

It is not only multiple queries to the real LOI that pose a problem; even a single set of k queries could leak the real LOI. Knowledge of how F works could be sufficient for the adversary to determine the real LOI, given the set $\mathcal{L}$ of all k nodes queried. Consider a deterministic F that takes the real LOI L and returns a list $\mathcal{L}$ that contains L and the $k-1$ fake LOIs, such that $\sigma(\mathcal{L}, \mathcal{N})$ is maximized over all possible choices of $k-1$ fake LOIs. This function, which maximizes the user's AOI-privacy, may leak the user's real LOI. Consider the example in Figure 5. If the $k = 3$ nodes illustrated in Figure 5(a) are queried, the adversary would know that the interior LOI is the real LOI. Were the westernmost node the real LOI, the central node and the easternmost node would have been chosen by F as the fake LOIs, as illustrated in Figure 5(b), to maximize σ. Similarly, the easternmost node could not be the real LOI.



(a) The $k = 3$ sensor nodes that were actually queried by the user.

(b) The nodes that would have been chosen as $\mathcal{L}$ were the westernmost node the real LOI.

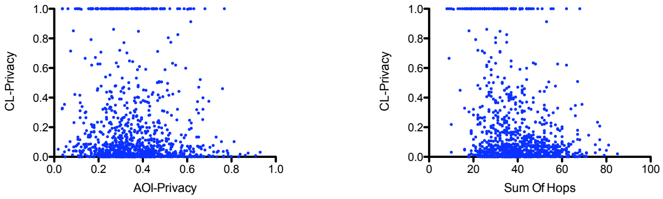
Fig. 5. An example of the σ-maximizing choice function leaking the real LOI.

For F not to leak information in this fashion, F must generate a set $\mathcal{L} \subseteq \mathcal{N}$ that is closed under F. For a deterministic F, this closure requirement means: if $\mathrm{F}(L) = \mathcal{L}$ for some $L \in \mathcal{N}$, then $\mathrm{F}(L_i) = \mathcal{L}$ for all $L_i \in \mathcal{L}$. For a non-deterministic F, this closure requirement means that there must be no candidate $L \in \mathcal{L}$ that was more likely than the other elements in $\mathcal{L}$ to have generated $\mathcal{L}$. Define $\mathrm{P}(\mathrm{F}, \mathcal{L}, L_i)$ as the proportion of all seeds in the seed space for which F will generate $\mathcal{L}$, given L_i and a seed as arguments. The closure property necessary for a non-deterministic F is: if $\mathrm{F}(L, s) = \mathcal{L}$ for some $L \in \mathcal{N}$ and seed s, then $\mathrm{P}(\mathrm{F}, \mathcal{L}, L_i) = \mathrm{P}(\mathrm{F}, \mathcal{L}, L)$ for all $L_i \in \mathcal{L}$. An unbiased random choice of the $k-1$ fake LOIs — the method of choosing fake LOIs we use for the remainder of this paper — meets this closure requirement. Future work should examine how to generate fake LOIs in such a way that guarantees a minimum AOI-privacy level, without leaking the real LOI.

5 Experimental Results

The area-of-interest (AOI) privacy that results from a set of queries to k locations of interest (LOIs), $k-1$ of which are fake, can be computed by the user before he/she issues the queries. The resulting current-location (CL) privacy, on the other hand, is dependent on how much information the adversary overhears, and cannot be computed by the user. Ideally, the user would like to predict the CL-privacy that will result from his/her queries, based on information that can be known ahead of time.

We first investigated how CL-privacy is correlated with the AOI-privacy of the queries, and also how CL-privacy is correlated with the amount of communication generated by the k queries. As a measure of the communication generated by a set of k queries, we define its *sum of hops* as $\sum_{i=1}^{k} (|\mathcal{R}_i| - 1)$. We simulated a user issuing a set of $k = 3$ queries in a sensor network with 400 nodes, arranged in a 20×20 grid. The nodes communicate over a broadcast medium, using fixed shortest-path routing. The adversary has compromised 10 random nodes in the network, but is guaranteed not to have compromised the node with which the user is directly communicating. We ran 1000 trials in which the user was placed at a random location in the sensor network and queried a random LOI, along with $k - 1 = 2$ fake LOIs chosen at random. In each trial, we recorded the resulting CL-privacy, AOI-privacy, and sum of hops. The results of this experiment are illustrated as scatterplots in Figure 6.



(a) The resulting AOI-privacy and CL-privacy.

(b) The sum of hops metric and the resulting CL-privacy.

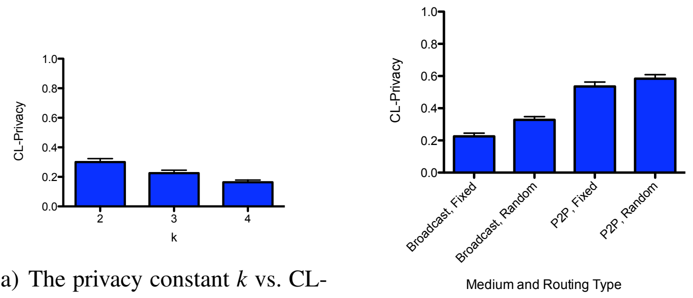
Fig. 6. Scatterplots comparing precomputed metrics to the resulting CL-privacy over 1000 trials.

These results indicate that the AOI-privacy level from a random selection of $k - 1$ fake LOIs is highly variable, as seen in Figure 6(a). That scatterplot also shows that the user cannot predict the resulting CL-privacy of his/her queries based on the precomputed AOI-privacy. While there is an inverse correlation between these two metrics, it is not a strong correlation ($R^2 = 0.0109$, meaning that only 1.09% of the variance in the CL-privacy can be explained by variations in the AOI-privacy value). From both the value of R^2 and visual inspection of the scatterplot, it is apparent that we cannot fit a function to the graph that could predict CL-privacy based on AOI-privacy. While the correlation between CL-privacy and the sum of hops metric is slightly stronger, as seen in Figure 6(b), it is still not a strong one ($R^2 = 0.0861$).

There is no clear method for predicting the resulting CL-privacy given a set of k queries. However, there are numerous factors that do affect the resulting CL-privacy. In order to investigate the effect of k, we repeated our earlier experiment, but using $k = 2$ and $k = 4$. The results, illustrated in Figure 7(a), show that incrementing k from 2 to 3 and from 3 to 4 did significantly decrease the CL-privacy.[1] A conclusion that can be drawn from this set of experiments is that if the privacy of his/her current location

[1] Significance tests, except where otherwise noted, use Kruskal-Wallis ANOVA with Dunn's multiple comparison test [10], with significant results having $P < 0.001$ and non-significant

is important, the user should sacrifice some of the privacy of the location of interest (measured as the number of fake LOIs queried).



(a) The privacy constant k vs. CL-privacy.

(b) Routing method and transmission type vs. CL-privacy.

Fig. 7. The effect of various parameters on the resulting CL-privacy, with error bars representing the 95% confidence interval around the mean over 1000 trials.

Until now, we have assumed that fixed shortest-path routing is used; however, the nodes may use random shortest-path routing to prevent messages from being traced back to their source as easily, which we hypothesize will increase CL-privacy. We have also assumed that the nodes communicate over a broadcast medium (e.g., radio). We hypothesize that migrating to a point-to-point (P2P) medium (e.g., optical transmissions) would reduce opportunities for an adversary to intercept traffic, increasing CL-privacy. We assume that point-to-point transmissions are only heard by the two communicating nodes; so, an adversary would only overhear traffic that involves a compromised node.

Next, we re-ran our previous experiment with $k = 3$, but used either random shortest-path routing, a point-to-point medium, or both. The results of this experiment, illustrated in Figure 7(b) suggest that moving from fixed shortest-path routing to random shortest-path routing, regardless of the transmission medium, significantly increased the resulting CL-privacy. Similarly, moving from a broadcast medium to a point-to-point medium, regardless of the routing algorithm, significantly increased CL-privacy.

To determine which of the two possible changes is more valuable, we performed a two-way ANOVA test. The routing method and transmission medium are not independent changes — the transmission medium used will have an effect on how much the routing method affects the CL-privacy, and vice versa. The choice of transmission medium was more important to the resulting CL-privacy, though. This result means that migrating to a point-to-point transmission system is more important for protecting the user's CL-privacy. However, using random shortest-path routing (which may be an easier change to make, as new hardware might not be required) still helps.

results having $P \geq 0.05$. This test compares three or more means without assuming Gaussian distribution, then analyzes significance between any desired pairs of means.

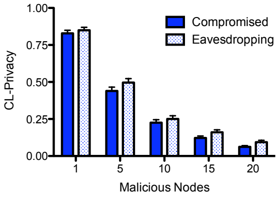


Fig. 8. The effect of compromised nodes and eavesdropping nodes on the resulting CL-privacy, with error bars representing the 95% confidence interval around the mean over 1000 trials.

To investigate whether the number and type of malicious nodes would affect the resulting CL-privacy, we re-ran our previous experiment with $k = 3$ using a broadcast medium and fixed shortest-path routing. This time, we varied the number of malicious nodes among 1, 5, 10, 15, and 20; we also changed whether the malicious nodes were compromised sensor nodes in the user's network or eavesdropping nodes deployed randomly within the range of at least one of the user's nodes. The results of this experiment are illustrated in Figure 8. Each increase in the number of compromised nodes significantly decreased CL-privacy, as did each increase in the number of eavesdroppers, i.e., more malicious nodes will result in decreased CL-privacy. Surprisingly, there is no statistically significant difference between any pair of results with a given number of malicious nodes for all five pairs. That is, there is no significant difference between the resulting CL-privacy when there are ten compromised nodes and when there are ten eavesdropping nodes, and so forth. While there does appear to be a small numerical difference between the resulting CL-privacy when there are compromised nodes and when there are eavesdropping nodes, based on visual inspection of Figure 8, these statistical results tell us that the extra information gained from compromised nodes is not that meaningful overall. This result does not bode well for a user interested in protecting the privacy of his/her current location; an adversary could deploy eavesdropping nodes, and gain essentially the same amount of information about the user's current location as if the adversary had performed the more complex task of compromising sensor nodes.

6 Conclusions and Future Work

The k-anonymity method for protecting the privacy of a location of interest allows the user to control the trade-off between privacy and communication cost. Both the privacy of the location of interest and the communication cost increase with the number of queries sent, k. We demonstrated a simple metric, based on the sum of minimum distances between queried nodes, that allows a user to know how well his/her selection of k nodes protects the privacy of the area of interest — that is, the area of the sensor network that contains the location of interest. Best of all, this k-anonymity scheme can be implemented over existing query mechanisms in a sensor network, with no need for any changes (hardware or software) to the sensors. This scheme is controlled entirely from the user's mobile device, which interacts with the sensors, and the scheme imposes a multiplicative overhead of factor k to sensor network communications.

However, the adversary can use the traffic generated by the *k* queries to find the user's current location. There was no clear way for the user to predict, prior to issuing a set of *k* queries, how much information will be leaked about his/her current location. However, we found factors that, in general, result in higher privacy for the user's current location: lower values for *k*, random routing approaches, avoiding the use of a broadcast medium, and having fewer malicious nodes available to the adversary. Surprisingly, an adversary that is able to compromise nodes in the user's sensor network does not have a significantly improved ability to locate the user, compared to an adversary who simply scatters eavesdropping nodes that do not participate in the sensor network. This observation provides important insight into the task of securing sensor network communications in sensitive areas.

This paper sheds some light on the trade-off between preserving the privacy of the queried and current locations simultaneously. Future work should examine how the *k*-anonymity method can be extended to further protect the user's privacy. For example, a user may add random delays in between sending each of his/her *k* queries into the sensor network. These random delays could make it more difficult for an adversary to know if any queries were not overheard by the malicious nodes.

Another topic for future work is to compare CL-privacy results from networks that use geographic routing to the results in this paper from shortest-path-routing networks. Changing the nature of the adversary could also yield interesting results. Our adversary, for example, only used compromised nodes in the user's sensor network to gain information about what routes queries did or did not take. An active adversary may additionally drop query packets, either as a denial of service attack, or as an effort to make the user re-route queries, generating more traffic. Understanding how different types of adversary could interact with this *k*-anonymity approach could provide additional insight into privacy in sensor networks.

Acknowledgments

The authors' research is funded in part by the Natural Sciences and Engineering Research Council of Canada, iCORE and Alberta Advanced Education & Technology, and Alberta Scholarship Programs.

References

1. Lopez, J.: Unleashing public-key cryptography in wireless sensor networks. Journal of Computer Security **14**(5) (2006) 469–482
2. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceedings of the 13th USENIX Security Symposium. (2004) 303–320
3. Misra, S., Xue, G.: Efficient anonymity schemes for clustered wireless sensor networks. International Journal of Sensor Networks **1**(1/2) (2006) 50–63
4. Ouyang, Y., Le, Z., Xu, Y., Triandopoulos, N., Zhang, S., Ford, J., Makedon, F.: Providing anonymity in wireless sensor networks. In: Proceedings of the IEEE International Conference on Pervasive Services. (2007) 145–148

5. Ozturk, C., Zhang, Y., Trappe, W.: Source-location privacy in energy-constrained sensor network routing. In: Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks. (2004) 88–93
6. Kamat, P., Zhang, Y., Trappe, W., Ozturk, C.: Enhancing source-location privacy in sensor network routing. In: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems. (2005) 599–608
7. Jian, Y., Chen, S., Zhang, Z., Zhang, L.: Protecting receiver-location privacy in wireless sensor networks. In: Proceedings of the 26th IEEE International Conference on Computer Communications. (May 2007) 1955–1963
8. Sweeney, L.: *k*-anonymity: A model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems **10**(5) (2002) 557–570
9. Blum, L., Blum, M., Shub, M.: A simple unpredictable pseudo-random number generator. SIAM Journal on Computing **15**(2) (May 1986) 364–383
10. Siegel, S., Castellan Jr., N.J.: Nonparametric Statistics for the Behavioral Sciences. 2nd edn. McGraw-Hill, New York (1988)