Source-Location Privacy in Energy-Constrained Sensor Network Routing

Celal Ozturk, Yanyong Zhang, Wade Trappe
Wireless Information Network Laboratory (WINLAB)
Rutgers University, 73 Brett Rd., Piscataway, NJ 08854
Email: {celal, yyzhang, trappe}@winlab.rutgers.edu

ABSTRACT

As sensor-driven applications become increasingly integrated into our lives, issues related to sensor privacy will become increasingly important. Although many privacy-related issues can be addressed by security mechanisms, one sensor network privacy issue that cannot be adequately addressed by network security is confidentiality of the source sensor's location. In this paper, we focus on protecting the source's location by introducing suitable modifications to sensor routing protocols to make it difficult for an adversary to backtrack to the origin of the sensor communication. In particular, we focus on the class of flooding protocols. While developing and evaluating our privacy-aware routing protocols, we jointly consider issues of location-privacy as well as the amount of energy consumed by the sensor network. Motivated by the observations, we propose a flexible routing strategy, known as *phantom routing*, which protects the source's location. Phantom routing is a two-stage routing scheme that first consists of a directed walk along a random direction, followed by routing from the phantom source to the sink. Our investigations have shown that phantom routing is a powerful technique for protecting the location of the source during sensor transmissions.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection (e.g.,firewall)*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing protocols*

General Terms

Security, Algorithms

Keywords

Source-Location Privacy, Context Privacy, Sensor Networks Privacy, Flooding

1. INTRODUCTION

Over the past few decades, the coupling of advancements in wireless communication technologies and distributed computing paradigms has initiated a change in the purpose for which networks are used. As the utility of affordable and power-efficient sensors is realized, it is expected that a broad spectrum of remote-sensing applications will emerge, ranging from collecting data regarding highway traffic to monitoring cardiologic data for at-risk heart patients. Due to the fact that sensor networks will affect our social fabric, one of the most notable challenges looming on the horizon and threatening successful deployment of sensor networks is privacy. Providing privacy in sensor networks is complicated by the fact that sensor networks consist of low-cost radio devices that will employ readily-available, standardized wireless communication technologies. As a result of the open-architecture of the underlying sensor technology, adversaries will be able to easily gain access to communications between sensor nodes either by purchasing their own low-cost sensor device and running it in a monitor mode, or by employing slightly more sophisticated software radios capable of monitoring a broad array of radio technologies.

The privacy threats that exist for sensor networks may be categorized into two broad classes: content-oriented security/privacy threats, and contextual privacy threats. Content-oriented security and privacy threats are issues that arise due to the ability of the adversary to observe and manipulate the exact content of packets being sent over the sensor network, whether these packets correspond to actual sensed-data or sensitive lower-layer control information. A first line of defense for protecting the content of sensor communications involves the use of appropriately designed network security protocols[1].

Although issues related to sensor security are important, we believe many of the core problems associated with sensor security are on the road to eventual resolution. Contextual privacy issues, associated with sensor communication, however, have not been as thoroughly addressed. In contrast to content-oriented security, the issue of contextual privacy is concerned with protecting the *context* associated with the measurement and transmission of sensed data. In many scenarios, general contextual information surrounding the sensor application, such as the location of the message originator, are sensitive and need to be protected. The underlying challenge of source-location privacy is to make it difficult for an adversary to trace his way, hop-by-hop, back to the origin of a communication.

Contextual privacy issues have been examined in the context of general networks, particularly through the methods of anonymous communications. Chaum proposed a model to provide anonymity against an adversary doing traffic analysis[5]. In the IP routing space, onion routing [13, 17] uses this model to provide anonymous connections. Similarly, the Mixmaster remailer[12] is an email implementation of Chaum mixes. Chaum mixes provide destination privacy when an attacker knows the source. This problem is the reverse of the problem we explore in this paper, and consequently does not apply to sensor source-location privacy. In [8, 9], a distributed anonymity algorithm was introduced that serves to remove fine levels of detail that could compromise the privacy associated with user locations in location-oriented services.

SASN'04, October 25, 2004, Washington, DC, USA.

Many of these methods are not appropriate for sensor networks, particularly sensor networks that are deployed for detecting and monitoring valuable assets. In particular, location-privacy techniques built using network security mechanisms, such as the anonymity provided by mixes, incur additional communication, memory, and computational overhead that are prohibitive for use in resource-constrained environments. Consequently, full-fledged privacy solutions are not appropriate, and light-weight, resource-efficient alternatives should be explored.

To alleviate this problem, this paper conducts a thorough analysis of the source-location privacy problem for sensor networks. Since privacy is often an elusive concept to put one's hands on and to evaluate quantitatively, we have provided a concrete metric for measuring location privacy in sensor networks through a new privacy measure called the privacy period. We have examined the popular routing techniques that are being in use in today's sensor networks, considered important systems-oriented issues like energy consumption, and found that most of their implementations cannot provide source-location privacy. In order to enhance source-location privacy, we developed various techniques to augment these routing protocols. One of our strategies, a technique we have called phantom routing, has proven to be flexible and capable of preventing the adversary from tracking the source location. Further, this technique does not incur any significant increase in energy overhead.

2. PANDA-HUNTER GAME AND THE SIMULATION MODEL

In this paper, we will focus on a generic sensor-network application, which we have called the *Panda-Hunter Game*. Before setting up the Panda-Hunter Game, we begin by defining privacy. *Privacy is the guarantee that information, in its general sense, is observable or decipherable by only those who are intentionally meant to observe or decipher it.* The phrase "in its general sense" is meant to imply that there may be types of information besides the message content that are associated with a message transmission. Source location, e.g. the location of the sensed event, is an important type of information whose privacy needs to be protected.

In the Panda-Hunter Game, a large array of panda-detection sensor nodes have been deployed by the Save-The-Panda Organization to monitor a vast habitat for pandas. As soon as a panda is observed, the corresponding *source* node will make observations, and report data (e.g., what the panda is doing, etc.) periodically to the *sink* via multi-hop routing techniques. The game also features a hunter in the role of the adversary, who tries to capture the panda by back-tracing the routing path until it reaches the source. As a result, a privacy-cautious routing technique should prevent the hunter from locating the source, while delivering the data to the sink.

The primary concern for the operator of the sensor network is the safety of the panda. In this sense, keeping the location of the source of a sensor reading unknown to the hunter is the primary underlying privacy goal. In order to explore this further, let us examine the operation of the Panda-Hunter Game in more detail. In the game, the panda pops up at a random location, and stays there until it is captured by the hunter. Once the hunter gets close to the panda (i.e., within δ hops from the panda), the panda is considered captured and the game is over. At the beginning of the simulation, the panda will appear at a random location, and the corresponding sensor node, which becomes the source, will start sending packets to the sink reporting its behavior. The simulator uses a global clock to synchronize all the activities within the network. The source generates a new packet every T clock ticks until the simulation ends. The packets are of the same length and will be encrypted, and the hunter cannot break the encryption. We employ a simple approach to model the communication links: a message will reach all the neighbors (i.e. the nodes that are within the sender's radio range R) of the sender at the next clock tick with the probability p, where p denotes the reliability of the channel (also modeling MAC-layer collisions), and $1 - p$ denotes the loss factor of the channel. The simulation ends either when the hunter catches the panda or when the hunter cannot catch the panda within a threshold amount of time (e.g. the panda has returned to its cave).

We assumed that the hunter is mobile with unlimited amount of power, yet a limited amount of memory. The hunter starts at the sink's location, where it is guaranteed that sensor packets must ultimately arrive. The hunter is constantly in a listening/receiving mode. Once it hears the first message, it knows which node among its neighborhood sent that message, and it will move to the transmitting sender node. It should be emphasized that, due to the multi-hop nature of routing protocols, a transmitter node may differ from the original message source. We further assume that the hunter has a message cache which stores the most recent M messages that have been heard. Every time the hunter moves to a new location, he continues to listen to the channel until he receives a new message. Multiple copies of the same message may traverse different portions of the network, and hence it is possible for the hunter to receive multiple copies of the same message at different times. The hunter will want to differentiate between new messages and previously observed messages. Therefore, we assume that the hunter can tell whether a message is new or not by comparing it with all the messages in its cache. Further, since the hunter has limited memory, we assume that he employs an LRU (Least Recently Used) cache replacement policy in order to ensure that the most recently heard messages (which are hopefully the most recent messages) are always kept in the cache. Once a new message is heard, the hunter makes another movement towards to its sender. If no new messages are heard within a specified period of time (T_{listen}), the hunter concludes that the current node he is at is not on the routing path, and must return to a former location. In addition, we also assume the hunter has a location cache which records the locations of the last N nodes it has visited to avoid loops. As soon as the hunter gets reasonably close to the panda (within the capture range δ), we assume the panda is caught and the game/simulation will end.

This paper is intended to design a family of routing strategies to conserve source-location privacy in sensor networks. In order to quantify the tradeoff between energy, privacy, and performance of these routing techniques, we have defined three main performance metrics: (1) privacy conservation level, which is measured by the number of new messages the source has sent before the panda is caught (referred to as safety period); (2) energy efficiency which is measured by the number of messages sent by the entire network; and (3) delivery quality which is measured by delivery latency and delivery ratio.

3. ROUTING TECHNIQUES FOR LOCATION PRIVACY

Rather than build a completely new layer for privacy, we take the viewpoint that existing technologies can be suitably modified in order to achieve desirable levels of privacy preservation. We will therefore examine several existing routing schemes to protect the source's location, while simultaneously exploring how much energy they consume. A wide range of routing techniques have been proposed for sensor networks, and it is infeasible to study all of them in this paper. In this study, we focus on flooding-based routing protocols [3, 6, 10, 11], though we have developed comparable methods for single path routing and found the similar results hold.

3.1 Baseline Flooding

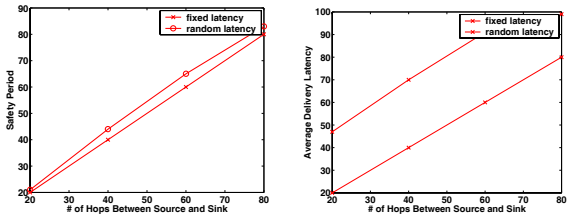
In our baseline implementation of flooding, we have made sure that every node in the network only forwards a message once, and no node retransmits a message that it has previously transmitted. When a message reaches an intermediate node, the node first checks whether it has received and forwarded that message before. If this is its first time, the node will broadcast the message to all its neighbors. Otherwise, it just discards the message. Realistically, this would require a cache at each sensor node. However, since sensor messages are typically small, and the delay between source messages is typically longer than the maximum time needed for a message to traverse the network, the cache size can be kept small. It is thus reasonable to expect that each sensor device will have enough cache to keep track of enough messages to determine whether it has seen a message before.

It is evident that flooding involves significant energy consumption. If we suppose there are n nodes in the sensor network, then the total number of transmissions that will take place (per new message) is upper bounded by n. We note that the actual amount of transmissions that occur is also affected by the packet reception rate p, and it is entirely possible that some sensors will never receive a message and hence never transmit that message themselves. For reasonable values of p, the energy spent by the entire network on a single message transmission will increase linearly with the size of the network n. Further, under realistic conditions, where packet collisions, packet retransmissions, and packet drops frequently occur, flooding is an energy expensive approach to message delivery. In our simulation studies, we found the number of transmissions per message for $p = 1$ and verified that it is equal to the number of nodes in the network.

Before we delve into the location-privacy protection capability of flooding, we first present the best strategy that an adversary can adopt. The adversary should start at the sink, and wait until it hears a message. If it first hears the message, it moves to the immediate sender of the message until it gets to the source. In this algorithm, the adversary must be able to tell if the message it has received is a new one or not, which is accomplished using the LRU-message cache.

At first glance, one may think that flooding can provide strong privacy protection since almost every node in the network will participate in data forwarding, and that the adversary may be led to the wrong source. Further inspection, however, reveals the contrary. We would like to emphasize that *flooding provides the least possible privacy protection since it allows the adversary to track and reach the source location within the minimum safety period.* For instance, if the shortest path length between the source and sink is 80, then the safety period is 80 as well.

We now provide an explanation for the poor privacy performance of flooding. Let us look at the set of all paths produced by the flooding of a single message. This set consists of a mixture of different paths, some longer than others, and it is clear that the shortest path between the source and the sink is contained in the collection of paths produced by flooding. Therefore, the first message that the adversary receives while waiting around the sink will correspond to a message that follows the shortest path, and as a result the adversary will be able to jump to the forwarding node on the last hop in the shortest path. Now, while the adversary is sitting at this new position, the source produces the next message. Due to the fact that the adversary is on the shortest path, the adversary will subsequently receive the next message via the subpath of the source-sink shortest path. Thus, the adversary will be able to jump to the previous forwarding node on the source-sink shortest path. Repeating iteratively, the adversary will capture every message on the shortest path, and ultimately reach the source via the shortest path.



(a) Safety Period (b)Delivery Latency

Figure 1: Flooding Protocol with Random Latency for Each Link.

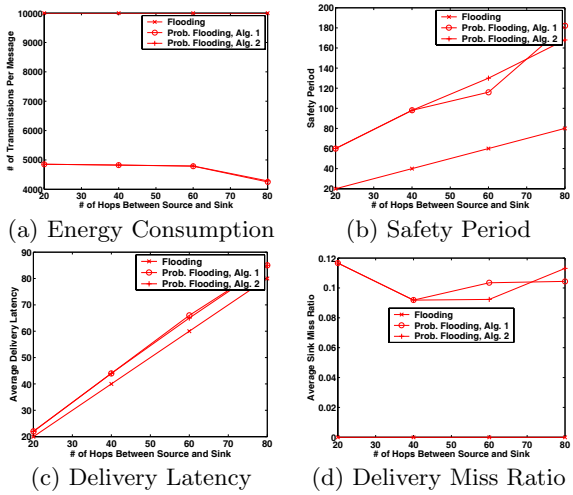
We can also explain the poor privacy performance of flooding by looking at the message latency. Suppose the source sends out m events, $e_1, e_2, \cdots, e_m$. The sink will receive multiple copies of each event, depending on the number of immediate neighbors it has. Among all the receiving times for the same event, the sink records the minimum time stamp, and further computes the minimum latency between the source and sink. We use d_i to denote the minimum latency for event e_i. Then the average of the d_i values is called the *average shortest latency*. We observe that the average shortest latency is always equal to the number of hops as it takes 1 clock tick to travel each hop. This further confirms that the messages always arrive earlier from the shortest path, and thus the adversary can easily locate the source.

One may argue that this observation might be just an artifact that results from the way we implement the flooding protocol. In the simulation, we assumed that every link incurs the same transmission delay of 1 clock tick. It is arguable that this fixed latency can lead to a fixed shortest path, thereby making the adversary's job easier. However, in order to demonstrate that our observation holds under more general and realistic network conditions, we also modeled the link latency as a number that is randomly selected with equal probability from $\{1, 2, 3\}$ clock ticks, thereby allowing the shortest path between different event deliveries to vary.

In this paper, we conducted simulations to study the privacy characteristics of flooding for a uniformly distributed network consisting of $n = 10,000$ nodes. The results with random-latency links are presented in Figures 1(a)-(b), for $p = 1$. Regardless of the link latency, every event is transmitted the same number of times. Since the average link latency increases in the random latency scenario, the average shortest latency also increases. We observe that the latencies with random delay are roughly around 1.2 times longer than the fixed-latency configuration across different network setups (Figure 1(b)). More importantly, in spite of the increase in the average latency, Figure 1(a) shows that the gap between safety periods for these two configurations is negligible (always below 10%). This result implies that the adversary can easily locate the source even when every link has a random latency, which supports the observation that the flooding technique does not provide privacy protection. (This is because different links in the network have the same delay distributions.) Therefore, in the remainder of this paper we will focus on the constant delay case.

3.2 Probabilistic Flooding

Probabilistic flooding [4, 7] was first proposed as an optimization of the baseline flooding technique to cut down energy consumption. In probabilistic flooding, only a subset of nodes within the entire network will participate in data forwarding, while the others simply discard the messages they receive. One possible weakness of this approach is that some messages may get lost in the network and as a result affect the overall network connectivity. However, as we shall explain later in this section, this problem does not appear to be a significant factor. The probability that a node for-



(a) Energy Consumption (b) Safety Period

(c) Delivery Latency (d) Delivery Miss Ratio

Figure 2: Probabilistic flooding with different adversary strategies. $P_{forward} = 0.5$, and $p = 1$.

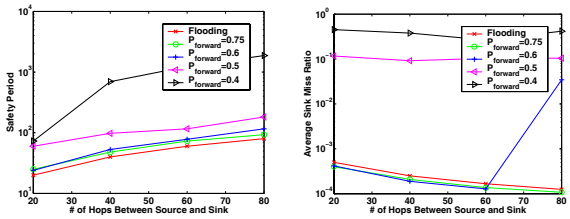
wards a message is referred to as the *forwarding probability* ($P_{forward}$).

In our simulation, we implement probabilistic flooding as follows. Every time a node receives a message, it will generate a random number q that is uniformly distributed between 0 and 1. If $q < P_{forward}$, the node will forward/broadcast this message to its neighbors. Otherwise, it will just discard that message. The parameter, $P_{forward}$, is important to the overall performance of this approach. A small value can help reduce the energy consumption though at the expense of lower network connectivity, while a large value can ensure a higher network connectivity but will have a correspondingly higher energy consumption.

In addition to its energy efficiency, probabilistic flooding can improve the privacy protection as well. Imagine there exists a path $\{1, 2, 3, 4, sink\}$, and the adversary is waiting for a new message at node 4. In flooding, the subsequent message will certainly arrive at node 4, though after some delay. However, in probabilistic flooding, the subsequent message may not arrive at node 4 because neighboring nodes may not forward, or take longer to arrive. As a result, the source will likely have to transmit more messages in order for the adversary to work his way back to the source. The more messages the adversary misses, the larger the safety period for the panda, and hence source location protection is provided.

Figures 2(a)-(d) compare the two adversary strategies with the baseline flooding technique by looking at: (a) number of transmissions per message; (b) safety period; (c) average shortest latency; and (d) average sink miss ratio ($\frac{n_{missed}}{n_{sent}}$). The first observation is that, in general, probabilistic flooding can significantly improve the safety period compared to baseline flooding (the improvement is at least 100%, sometimes better than 200%). Further, probabilistic flooding also significantly improves the energy consumption (50%). At the same time, probabilistic flooding techniques only increase the average shortest latency marginally, with the observed increase always within 10% of pure flooding. Although probabilistic flooding with $P_{forward} = 0.5$ causes the sink to miss a few messages, the miss ratio is also marginal (always less than 12%). The second observation is that there is not a noticeable difference between the two adversary strategies, and hence the adversary gains no advantage by employing one method instead of the other. Finally, we examined other values for $P_{forward}$ and observed similar trends.

In probabilistic flooding, the adversary will often not stay on the shortest path between the source and sink since there is a positive probability that a message will not be delivered



(a) Safety Period (b) Delivery Miss Ratio

Figure 3: Probabilistic flooding with different $P_{forward}$. Y-axis uses log scale.

on the shortest path. Let us pick a random node within the network, and a random path which connects that node and the source. Suppose that this path has length l, and let us use P_{path} to represent the probability of the node getting a message from this particular path. Then we have

$$P_{path} = P_{forward}^{l}. \quad (1)$$

If the shortest path fails, there is a high likelihood that *at least* one longer path will succeed, drawing the hunter away from the shortest path, putting the adversary on a less-efficient path. It should be noted that the improvement provided by probabilistic flooding is not unrestricted. There is a natural probabilistic pull that draws the adversary back towards shorter paths. To see this, suppose the hunter migrated and followed a longer path of length l_2. Further, if there is a shorter path with length l_1 that passes through the node that he is now at, then applying Equation 1, we have $P_{path_2} = P_{forward}^{l_2}$ and $P_{path_1} = P_{forward}^{l_1}$, hence $P_{path_2} < P_{path_1}$. Thus there is a higher likelihood that the hunter will drift back towards a shorter path, and therefore the hunter will ultimately receive the majority of his new messages from a set of reasonably short paths. We would emphasize that the safety period of probabilistic flooding will stay the same even though the network size increases as long as the source and sink are the same.

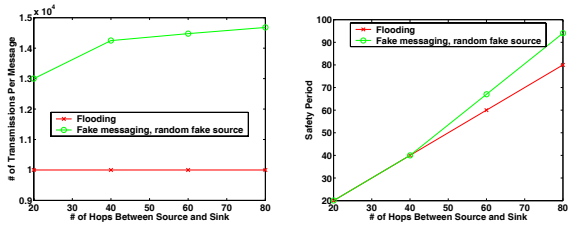
Figures 3(a) and (b) present the safety period and sink miss ratio for different forwarding probabilities. In these figures, we have depicted the y-axis using the log scale. As we decrease the forwarding probability, the safety period improves significantly, but at the same time the sink miss ratio substantially drops. Through experimentation, we have found that $P_{forward} = 0.5$ achieves a good balance between these two factors.

3.3 Flooding with Fake Messages

As mentioned in Section 3.1, flooding cannot provide privacy protection because the adversary can easily identify the shortest path between the source and the sink, allowing him to back trace to the source location. One of the reasons this happens is due to the fact that we only have one source in the network. This observation suggests that one approach we can take to alleviate the risk of source-location privacy breaching is to augment the flooding protocols to introduce more sources that inject fake messages into the network.

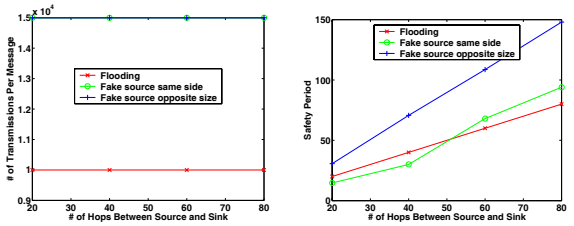
In order to demonstrate the effectiveness of fake messaging, we assume that these messages are of the same length as the real messages, and that they are encrypted as well. Therefore, the adversary cannot tell the difference between a fake message and a real one. As a result, when a fake message reaches the hunter, he will think that it is a legitimate new message, and will be guided towards the fake source.

One challenge with this approach is how to inject fake messages. We need to first decide how to create the fake sources, and when and how often these fake sources should inject false messages. Specifically, we want these fake sources



(a) Energy Consumption (b) Safety Period

Figure 4: Fake messaging with a random fake source.



(a) Energy Consumption (b) Safety Period

Figure 5: Fake messaging with a persistent fake source. The fake message frequency is half of the real message frequency.

to start only after the event is observed, otherwise the use of fake sources would consume precious sensor energy although there is no panda present to protect.

First, let us look at one naive injection strategy that does not require any additional overhead, which we refer to as the *Short-lived Fake Source* strategy. This strategy uses the constant P_{fake} to govern the fake message rate, and choose $P_{fake} \propto \frac{1}{n}$. For any node within the network, after it receives a real message, it generates a random number q that is uniformly distributed between 0 and 1. If $q < P_{fake}$, then this node will produce a fake packet and flood it to the network. We can tune the value of P_{fake} to balance the trade off between energy consumption and privacy protection. In this strategy, the fake source changes from one fake message to another.

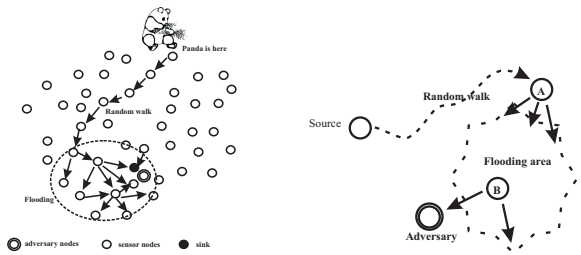
Figure 4 shows the performance of the Short-lived Fake Source strategy. Compared to the baseline flooding technique, the energy consumption increased by a factor of 1.5 (Figure 4(a)) due to fake messages, but its safety period is only marginally improved ((Figure 4(b)). The reason for its poor privacy protection is due to the fact that the fake sources are short-lived. Even if the Hunter is guided by one fake message to a wrong location, there are no subsequent fake messages around that location to draw him even further away, so he can catch the next real messages. As a result, we need a persistent fake source to mislead the Hunter.

Next, we discuss such a fake message injection method, referred to as *Persistent Fake Source*. The basic idea of this method is that once a node decides to become a fake source, it will keep generating fake messages regularly so that the Hunter can be misled. A node decides whether or not to become a fake source based upon the value P_{fake} as described above. Once the fake source is persistent, its location is the key to the success of the scheme. For example, Figure 5 shows the performance of different fake source locations. If the fake source is along the trace from the sink to the source, the fake message may be able to direct the Hunter to the real source, especially if the fake messaging frequency is smaller (Figure 5(b)). However, if the fake source is on the opposite side of the real one, then the safety period is improved by up to 85%.

3.4 Phantom Flooding

Phantom flooding shares the same insights as probabilistic flooding in that they both attempt to direct messages to different locations of the network so that the adversary cannot receive a steady stream of messages to track the source. As we pointed out in Section 3.2, probabilistic flooding is not very effective in achieving this goal because shorter paths are more likely to deliver more messages. Therefore, what we would like to do is somehow entice the hunter away from the source and towards a fake source, called the phantom source.

In phantom flooding, every message experiences two phases: (1) a walking phase, which may be a random walk or a directed walk, and (2) a subsequent flooding meant to deliver



(a)phantom flooding protocol (b)example scenario

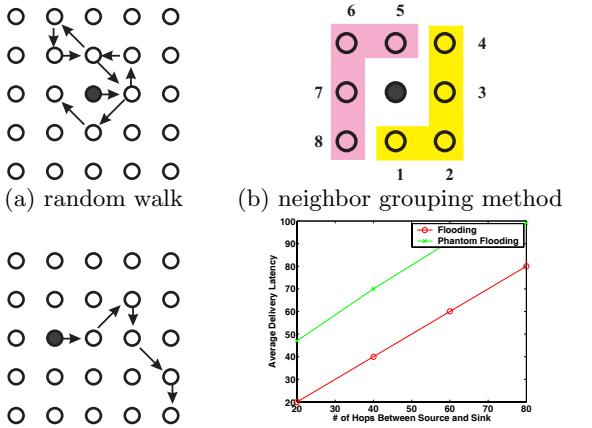
Figure 6: Illustration of Phantom Flooding.

the message to the sink. When the source sends out a message, the message is unicasted in a random fashion within the first h_{walk} hops (referred to as random walk phase). After the h_{walk} hops, the message is flooded using the baseline flooding technique described in Section 3.1 (referred to as flooding phase). The algorithm is illustrated in Figure 6(a). The implementation of the flooding phase has been discussed earlier.

Phantom flooding can significantly improve the network safety period because every message may take a different (shortest) path to reach any node within the network. As a result, after the adversary hears message i, it may take a long time before it receives $i + 1$. When it finally receives message $i+1$, the immediate sender of that message may lead the adversary farther away from the source. In the example shown in Figure 6(b), the adversary is already pretty close to the source before it receives the next new message. This new message goes through the random walk phase and reaches node A, and then goes through the flooding phase. The adversary receives this message from node B, and according to its strategy, it will be duped to move to node B, which is actually farther away from the source compared to the current location of the source.

Another advantage of phantom flooding is that its privacy protection improves as the network size and intensity increase because the path diversity between different messages will become more substantial.

It is not a trivial task to implement random walk. The purpose of the random walk is to send a message to a random location away from the real source. However, if the network is more or less uniformly deployed, and we let those nodes randomly choose one of their neighbors with equal probability, then the resulting random walk path is essentially an unbiased, discrete two-dimensional Brownian motion. Therefore, there is a large chance that the message path will loop around the source spot, and branch to a random location not far from the source (illustrated in Figure 7(a)). Our simulation results further confirm this observation, but due to space limitations, the results are not shown here. In



(a) random walk (b) neighbor grouping method

(c) directed walk (d) Latency for Phantom Flooding

Figure 7: Illustration of Directed Walk Algorithm.

order to avoid random walks cancelling each other, we need to introduce bias into the walking process, and therefore we propose the use of *directed walk* to provide location-privacy. In directed walk, we separate the neighbors into two groups so that those nodes whose directions are opposite to each other do not belong to the same group, as illustrated in Figure 7(b). At the first step of the directed walk, the node randomly picks one group, and later steps will only choose neighbor nodes from that specific group. This method can remove the paths that loop back upon themselves in the random walk. As a result, the routing can leave the source area and reach a random location (illustrated in Figure 7(c)).

Directed walk requires a node knows the relative position of its neighbors. Such knowledge can be obtained by using ranging [2, 14, 16] and angle of arrival (AOA) [15] measurements.

In this simulation, we varied the source location by varying the shortest path between the source and the sink as in earlier sections. We also varied the directed walk length (h_{walk}) to study its bearing on the privacy level. We have found that even with a directed walk length of 10, the Hunter cannot track the source location. Phantom flooding successfully protects the source location privacy.

Compared to baseline flooding, phantom flooding does not increase the energy consumption because each node at most forwards the same message once. However, phantom flooding can potentially increase the average message latency because every message is directed to a random location first. We expect that the latency should be increased at least by the factor h_{walk}. Figure 7(d) shows the average shortest message latency for different source locations. We find that the increase in latency is always between 20 and 30. As the network size increases, this relative increase is negligible.

4. CONCLUDING REMARKS

Networks of energy-constrained sensor nodes are increasingly being deployed for monitoring and data collection applications. The very nature of sensor networks such as their location-dependency, their context sensitivity, and the challenges of the underlying wireless communication protocols has created a new set of problems surrounding the security and privacy of the sensor communications. An important aspect of the communication context is the source location. In many applications, if the adversary observes traffic within the network, he may be able to back track these messages to locate the event source, which can be a serious privacy breach for many monitoring and remote-sensing application scenarios.

In this paper, we have identified this important problem, and indicated that the source location privacy can be strongly influenced by the data dissemination techniques or routing protocols. We have examined one of the most popular families of routing protocols in sensor networks, namely flooding. Based on our analysis and simulations, we have found out that neither of these protocols are capable of providing source location privacy.

We have proposed a family of techniques for the flooding routing classes that enhance their privacy protection. After observing the privacy performance and energy consumption characteristics of these different methods, we have proposed a very powerful strategy, known as phantom routing. Through our simulations, we have shown that phantom routing is capable of keeping the adversary virtually lost within the sensor network, thus significantly enhancing source-location privacy, while not incurring any significant energy overhead.

5. REFERENCES

[1] Wireless security workshop. See http://www.ece.cmu.edu/ adrian/wise2004/.

[2] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of IEEE INFOCOM'00*, 2000.

[3] C. L. Barrett, S. J. Eidenbenz, L. Kroc, M. Marathe, and J. P. Smit. Parametric probabilistic sensor network routing. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, 2003.

[4] D. Braginsky and D. Estrin. Rumor routing algorthim for sensor networks . In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, 2002.

[5] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24:84–88, 1981.

[6] Z. Cheng and W. Heinzelman. Flooding Strategy for Target Discovery in Wireless Networks. In *proceedings of the Sixth ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2003)*, 2003.

[7] P. Th. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, and A.-M. Kermarrec. Lightweight probabilistic broadcast. *ACM Transactions on Computer Systems (TOCS)*, 21(4):341 – 374, November 2003.

[8] M. Gruteser and D. Grunwald. Anonymous Usage of Location-based Services through Spatial and Temporal Cloaking. In *Proceedings of the nternational Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2003.

[9] M. Gruteser, G. Schelle, A. Jain, R. Han, and D. Grunwald. Privacy-aware location sensor networks. In *Workshop on Hot Topics in Operating Systems (HotOS)*, 2003.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networks (MobiCOM)*, August 2000.

[11] H. Lim and C. Kim. Flooding in Wireless Ad-hoc Networks. In *IEEE computer communications*, 2000.

[12] Mixmaster remailer. http://mixmaster.sourceforge.net/.

[13] M.Reed, P. Syverson, and D. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16:482–494, May 1998.

[14] D. Niculescu and B. Nath. Ad Hoc Positioning System (APS). In *Proceedings of the IEEE GLOBECOM 2001*, November 2001.

[15] D. Niculescu and B. Nath. Trajectory Based Forwarding and its Applications. In *Proceedings of the Ninth Annual ACM/IEEE International Conference on Mobile Computing and Networks (MobiCOM)*, pages 260–272, September 2003.

[16] A. Savvides, C. Han, and M. B. Strivastava. Dynamic fine-grained localization in Ad-Hoc networks of sensors. In *International Conference on Mobile Computing and Networks (MobiCOM)*, pages 166–179, 2001.

[17] P. Syverson, M. Reed, and D. Goldschlag. Onion routing access configurations. In *DISCEX 2000: Proceedings of the DARPA Information Survivability Conference and Exposition*, pages 34–40, January 2000.