Towards Understanding Source Location Privacy in Wireless Sensor Networks Through Fake Sources

Arshad Jhumka, Matthew Bradbury and Matthew Leeke
Department of Computer Science
University of Warwick, Coventry
United Kingdom, CV4 7AL
{*arshad, csujbt, matt*}*@dcs.warwick.ac.uk*

***Abstract*—Source location privacy is becoming an increasingly important property in wireless sensor network applications, such as asset monitoring. The original source location problem is to protect the location of a source in a wireless sensor network from a single distributed eavesdropper attack. Several techniques have been proposed to address the source location problem, where most of these apply some form of traffic analysis and engineering to provide enhanced privacy. One such technique, namely fake sources, has proved to be promising for providing source location privacy. Recent research has concentrated on investigating the efficiency of fake source approaches under various attacker models. In this paper, we (i) provide a novel formalisation of the source location privacy problem, (ii) prove the source location privacy problem to be NP-complete, and (iii) provide a heuristic that yields an optimal level of privacy under appropriate parameterisation. Crucially, the results presented show that fake sources can provide a high, sometimes optimal, level of privacy.**

***Keywords*-Complexity; Distributed Eavesdropper; Fake Source; Security; Source Location Privacy; Wireless Sensor Networks**

I. Introduction

The emergence of ad-hoc networks, such as wireless sensor networks (WSNs), has enabled several novel classes of application, including monitoring and tracking. For monitoring applications, the deployment of WSNs varies from safety-critical monitoring applications such as military, health and radiation monitoring, to non-critical applications such as temperature and humidity control. For safety-critical applications, the dependability if a WSN is an important consideration. In particular, privacy, which can be generally described as the guarantee that information can only be observed or deciphered by those intended to observe or decipher it [1], is an important property. As WSNs operate in a broadcast medium, attackers can easily intercept messages and launch attacks based on the information they derive.

The security threats that exist for WSNs can be classified along two dimensions: (i) *content-based* privacy threats and (ii) *context-based* privacy threats. The privacy threats relating to content are based on the contents of messages, i.e., attacks are launched with regard to the data generated by the higher network layers - data generated either at the application level, e.g., values sensed by sensors, or lower-layer levels, e.g., location and time-stamps. In such attacks, attackers try to capture data to learn about the status of the network so that relevant attacks can be launched. Much research has addressed content-based attacks [2]. On the other hand, context-based privacy threats are those that are based on the context associated with the measurement and transmission of sensed data. Context is a multi-attribute concept that captures several aspects of sensed data, some of which are environmental. These aspects include location and time, both of which allow proper semantics to be given to sensed data. While content-based threats have been widely addressed [2], context-based threats are becoming increasingly popular. For content-based threats, nodes launching attacks are often modelled as Byzantine nodes [3] [4], with cryptographic techniques often being used to address these problems [2] [3] [5]. However, cryptographic techniques cannot help with handling context-based threats. New techniques must be introduced to handle context-based threats.

One attribute of context that is crucial in many application domains is *location*. Location information can be embedded in a message but remain inaccessible to an attacker due to message encryption. Hence, as location information can not be obtained directly, an attacker may opt to infer it. An important problem in monitoring applications is the problem of *source location privacy* (SLP). In this problem, a WSN is monitoring an asset, such as an endangered animal. Then, periodically, the nodes detecting the asset, which we term *source nodes*, will send messages to a dedicated node, called a *sink*, for data collection. If the location of the source nodes is compromised then an attacker can easily capture the asset.

It is possible to infer message location information through various techniques, depending on the power of the attacker. For example, Metha et.al [6] assumes that an attacker has a small wireless network of his own that captures messages, and shows how the attacker can infer the location of nodes after messages have been intercepted. On the other hand, Kamat et.al [1] assume a single attacker, who uses the routing protocol to infer the location of the source. For example, in a military environment, soldiers out on a surveillance mission may relay information to a sink. An attacker can intercept these messages, and trace them back

to the soldiers, thereby compromising their safety. Several possible techniques to handle the SLP problem have been proposed [1] [6] [7] [8] [9]. In this paper, we focus on the *fake source* technique, first proposed by seminal work in this area [1]. When applying this technique, a (set of) node(s) is chosen to act as a decoy for the real source, i.e., to act as a fake source. The fake source generates messages to engineer network traffic in such a way so as to confuse an attacker. Despite the intuitive nature of the approach, little work has been done to understand the detailed working of the fake source technique. Recently, two algorithms were proposed that address the tradeoffs between privacy and energy usage [10], where these algorithms were shown to provide a balance between privacy and energy consumption.

A. Contributions

In this paper, we provide a novel formalisation of the SLP problem, allowing us to identify two important parameters that underpin the efficiency of algorithms that provide SLP: (i) message rates, and (ii) fake message transmission duration. Under this formalisation, we show the problem to be NP-complete. We subsequently propose a heuristic, parameterised by these considerations, and evaluate its efficiency in providing SLP. Our results show that, under certain parameterisation, it is possible to obtain optimal privacy, i.e., the real source is never captured.

B. Paper Structure

This paper is structured as follows. In Section II we provide a survey of related work. In Section III, we define the adopted network and attacker models. In Section IV, we present two of the contributions of the paper, namely proof of NP-hardness and an algorithm that provides SLP. In Section V, we outline the experimental approach employed in this paper. The results generated by this experimental approach are presented and discussed in Section VI. Finally, Section VII concludes with a contribution summary and a discussion of future work.

II. Related Work

The problem of SLP first appeared around 2005 in a seminal paper by Kamat *et al.* [1], which was shortly followed by [11]. The authors of [1] proposed a formalisation of the SLP problem, and subsequently investigated several algorithms to enhance SLP. They proposed the fake source technique, but indicated that it has poor performance despite being an expensive technique. They went on to propose an algorithm called *phantom routing*, where messages are initially sent on a random walk of a given length, followed by a normal flooding. The overall result implies that attackers cannot fully trace back messages to the real source. Since then, research has addressed the problem using a variety of attacker models and assumptions. Different attacker models and assumptions lead to different types of solutions or techniques for enhancing SLP. Subsequently, an attack was shown to subvert the phantom routing technique that was proposed by Kamat et.al [1], with the added assumption that nodes have access to their location using say GPS devices. In this work, the attacker is *local*, meaning it does not have instant access to global network information. Rather, the attacker will have to slowly accumulate knowledge to gain global network information. We focus on the fake source technique with such an attacker in this paper.

There are several other research directions relating to privacy in WSNs. Some have investigated the problem of base station-location privacy [12]. Others have focused on more powerful attackers, such as in [6] and [13], whilst other research has focused on temporal privacy [14]. However, the main limitations of existing work relating to the fake source technique are the limiting assumptions made and the lack of proposals relating to algorithms with high levels of privacy. For example, a global eavesdropper is assumed in [12], whilst [6] and [13] assume attackers have powerful sensors.

III. Models

In this section, we provide definitions of the adopted network and attacker models.

A. System Model

We define a wireless sensor *node* as a computing device equipped with a wireless interface and associated with a unique identifier. Communication from a node is typically modelled with a circular communication range centred on the node. With this model, a node is thought to able to exchange data with all devices within its communication range. A *link* exists between two nodes m and m' if both m and m' can communicate with each other.

A WSN is a set of wireless sensor nodes with links between pairs of nodes. We assume that all nodes in the network have the same communication range. Such a network is modelled as an undirected graph $G = (V, E)$, where the set of vertices V represents the set of N wireless sensor nodes and the set of edges E represents the set of links between the nodes. Two nodes $m \in V$ and $m' \in V$ are said to be 1-hop neighbours (or neighbours) iff $\{m, m'\} \in E$, i.e., m and m' are in each other's communication range. We denote by M the set of m's neighbours. The graph $G = (V, E)$ defines the topology of the network. In this paper, we focus on grid-like network topology, i.e., network of size $n * n = N$. There exists a distinguished node in the network called a sink S, which is responsible for collecting data. Other nodes $v \in V \setminus \{S\}$ sense data and then route the data to the sink for collection. In general, any node can be a source of sensed data. In this paper, we assume that the data source is not close to the attacker. We denote the distance between the sink and a node $n \in V$ by δ_n. There exists a relation on V, denoted $\prec_H$, such that $m \prec_H n$ iff

$H(\delta_m, \delta_n)$. For example, if H is the function "closer", then the relation captures which of nodes is closer to the sink.

Sensor nodes route messages to the sink, generally using data aggregation convergecast protocols [15]. We assume that there can be several nodes acting as message sources at the same time. We assume that the network is event-triggered - when a node senses an object of interest, it starts sending messages to the sink over a certain time period.

B. Attacker Model

In general, an attacker can be considered to be a set of sensor nodes. It has been proposed in [16] that the strength of an attacker can be factored along two main dimensions: (i) presence, and (ii) actions. For example, presence can be local or global, while actions can be eavesdropping, or reprogramming among others. Using these two dimensions, a lattice of attacker strengths was developed. Based on this lattice, we consider one type of attacker, namely a *distributed eavesdropping* attacker. There can be different implementations of this type of attacker. For example, such an attacker can be a single mobile person equipped with a sensor node allowing him to eavesdrop. Another implementation can be multiple persons, each with a sensor node, eavesdropping on the network [10]. In this paper, we consider the single person implementation of the distributed eavesdropper attacker.

When a source sends a message, we assume the message to be encrypted. We assume that the source includes its ID in the encrypted messages, but only the sink can tell a nodes location from its ID. As a result, even if the attacker is able to break the encryption in a reasonably short time frame, it cannot tell the sources location. We assume the distributed eavesdropper attacker to be equipped with devices, such as antenna and spectrum analysers, so it can measure the angle of arrival of a message and the received signal strength to identify the immediate sender and move to that node. We point out that the attacker cannot learn the source of a message by merely observing a relayed version of the message. We also assume that the attacker can move at any speed and place no restrictions on their power consumption. In addition, it also has a large amount of memory to keep track of information such as messages that have been heard and nodes that have been visited.

In assessing the privacy of a system, one should always assume that the worst case scenario, in that the attacker knows the methods being used by the system. Therefore, we assume that the attacker knows (i) the location of the sink node, (ii) the network topology, but cannot however infer the location of a message source based on a relayed message, and (iii) the routing algorithm used. However, the attacker does not know the number of assets being monitored, and the possible location of the asset, i.e., the asset can be randomly located in the network. These assumptions imply that an attacker has no way of determining if a message is a fake one or a real one. Apart from these assumptions, the only knowledge a distributed eavesdropper has is that which is deduced based on eavesdropping across the network. For example, when he hears a (relayed) message coming from a (legitimate) node within its neighbourhood, he can locate the sender of that message (not the source of the message). We also assume that the attacker does not know the number of possible assets being monitored. This can happen when monitoring endangered animal species, which hunters may try and poach.

IV. Understanding the Fake Source Problem - Problem Statement, Complexity and Heuristics

In this section, we briefly explain the fake source technique for SLP in a WSNs. We subsequently present a formalisation of the problem, and prove it to be NP-complete.

A. Fake Sources: An Informal Introduction

The fake source problem, as per the original setup [1], is as follows: There is a message source (real source) in a WSN, with an attacker initially positioned at the sink. The attacker is positioned at the sink to ensure that it captures messages being sent there. As the sink receives messages from the source, usually along the shortest path, the attacker can trace these messages, hop by hop, back to the source. The fake source technique, as its name suggests, involves selecting a subset of nodes to act as fake sources to simulate the real source. In their seminal work on fake sources, Kamat *et al.* [1] proposed a formalisation of SLP, and the fake source technique as a viable approach. They also proposed two types of fake sources: (i) short-lived fake sources, and (ii) permanent fake sources. The work concluded that permanent fake sources outperform temporary fake sources [1]. However, in a recent work, it was shown that implementations of permanent fake sources that attempt to trade-off energy against privacy may not impart the required level of privacy [10]. In this paper, we focus on both of types of fake sources. The intuition is to use temporary fake sources to quickly "lure" the attacker away from the source, and to use permanent fake sources to then keep the attacker away from the real source. Modelling transient and permanent fake sources can be achieved using a duration parameter, where the duration of a permanent fake source is typically ∞ and the duration of a temporary fake source is finite and bounded. This will enable a tradeoff between energy and privacy [17], though the main objective of this paper is to understand the complexity of providing source location privacy.

Solving a specific instance of the source location privacy problem corresponds to the location of the source. Unless the source is static, which is not the focus of this paper, a fake source has a time bound during which to send fake messages. Specifically, this is the deadline by which the last fake message will be sent in that instance of the fake source

problem. Note that attempting to attract an attacker towards a fake source is done in the context of a routing structure.

A real source is characterised by the following parameters: location and message transmission rate. Any implementation of the fake source technique will have to investigate the impact of at least these two parameters. It has been argued that a fake source will have to be a similar distance away from the sink as the real source for better privacy [1].

B. Fake Sources - Problem Statement

In this section, we provide a novel formalisation of the fake source problem. Specifically, we formalise the problem as a scheduling problem, whereby fake sources are scheduled to send fake messages to lure an attacker away from the real source.

A fake source F_i is a determined by the *duration* d_i during which it acts as a fake source. If d_i is (small and) finite, then the fake source is *temporary*, else it is *permanent*. The selection of fake sources is important as it plays an important role in ensuring high levels of source location privacy. A fake source selection algorithm chooses a set of fake sources from a set of candidate nodes, by "tagging" the relevant nodes with given durations, i.e., sets the fake source flag of a chosen node to be 1 and sets its duration to d_i. Since fake sources send at different times, there exists a "precedence" relationship between the nodes. In this work, there exists a precedence relationship between temporary fake sources and permanent fake sources, where temporary fake sources try to first "lure" attacker away, and then permanent fake sources ensure the attacker is kept away from the real source.

The source location privacy (SLP) problem can be stated as follows:

Definition 1 (SLP): Given a graph $G = (V, E)$, a set F of fake source tags $\{F_1 \dots F_f\}$, each F_i associated with duration d_i, a relation $\prec$ on V, a set N of m nodes $\{n_1 \dots n_m\} \subset V$, a deadline τ and a routing strategy R, assign tags in T to nodes in N to obtain an m-node schedule σ for T that meets the deadline τ under R and obeys $\prec$.

C. SLP Complexity

We now present the first contribution of this paper - a proof that the formalised SLP problem is NP-complete.

Theorem 1 (SLP Complexity): The Source Location Privacy problem is NP-complete.

Proof: We reduce the multiprocessor scheduling with precedence constraints (MSPC) to SLP. We first define the MSPC problem, and then identify the mapping between MSPC and SLP.

Instance: The MSPC problem is as follows: Given a set $T = \{T_1 \dots T_n\}$ of tasks, with task T_i having execution time e_i, a set P of $m \in \mathbb{Z}^+$ processors, partial order $\lessdot$ on T, and a deadline $D \in \mathbb{Z}^+$.

Question: Is there an m-processor schedule σ for T that meets the overall deadline D and obeys the precedence constraints, i.e., such that $T_i \lessdot T_j$ implies that $\sigma(T_j) \geq \sigma(T_i) + e_i$.

Mapping:

- $T \mapsto F$
- $e_i \mapsto d_i$
- $P \mapsto N$
- $D \mapsto \tau$
- $\lessdot \mapsto \prec$

Also, given a schedule σ for SLP, and a routing strategy R, it can be verified in polynomial time if σ completes before deadline τ, implying *SLP* $\in NP$.

D. A Heuristic for Source Location Privacy

In this section, we present the second contribution of this paper - a heuristic that can provide near-optimal source location privacy under specific parameterisation. We develop the heuristic based on observations made in previous work [1]. It was previously observed that a flooding algorithm provides no source location privacy [1]. We thus use the flooding algorithm as baseline protocol, and enhance it for source location privacy. Thus, any improvement in the level of source location privacy will thus be due to the enhancement (and not due to the flooding protocol). We thus built our heuristic on top of a simple flooding algorithm which the real source uses to send messages to the sink. This assumption is more general than assuming a single shortest path routing algorithm from source to sink. Also, observe that the flooding algorithm is outside the heuristic we are proposing, and that the heuristic will work with other routing techniques, though with possibly different efficiency.

Flooding Algorithm: The adapted flooding protocol is implemented as follows (and is shown in Figures 1 and 2): The (real) source (Figure 2) generates an application message, as a result of detecting the asset, and broadcasts it to every normal node (Figure 1) in its neighbourhood. The message contains a *sequence number*, denoted by *count*, and a field, called *hop*, that keeps track of the (hop) distance the message has travelled. The *hop* value is initialised to 0 by the real source node.

When a (normal) node receives the message, it checks if the message is new, i.e., whether or not it has previously seen the same sequence number. If it is new, then the node increments the *hop* value by one and broadcasts the message. This process is repeated until the message reaches

the sink. The value of the *hop* count at the sink represents the distance of the real source from the sink.

Protocol Extension for Fake Sources: The protocol extension is shown in Figures 3 and 4. When the sink receives the first such message (Figure 3), it broadcasts a special message, called a *FakeNode* message, instantiated with the value of *hop*, and contains the sequence number *count* of the message for which fake sources have to be selected. When a node receives the *FakeNode* message, it checks if it has seen such a message with the *count* sequence number. If it has not, then it checks if the *hop* value is 1. If the *hop* value is greater than 1, then the node becomes a *temporary fake source* (Figure 1). This means that the node starts sending *a certain number of messages* over a *time duration* d. When d expires, the node broadcasts the *FakeNode* message, with the *hop* value decremented by 1.

If, on the other hand, the *hop* value is 1, then the node generates a random number and becomes a *permanent fake source* if the number is greater than a given threshold σ (Figure 1), i.e., when it becomes a permanent fake source, it transmits fake messages indefinitely. The generation of a random number is done so that the number of permanent fake sources is controlled. Fake sources send fake messages at a given rate over a certain duration (Figure 4).

Implementation Issues: The structure of the messages sent by the temporary and permanent fake sources are identical to those sent by the real source. The only difference is in the payload, where in the case of the fake sources, the payload is random. Based on this, we assume an attacker cannot distinguish between a real message and a fake one. Note that, as our assumption is that real messages are encrypted, so are the fake messages. This means that only legitimate nodes can read the fake messages. However, when there is more than one fake source operating, this may result in a legitimate intermediate node dropping messages from two different fake source nodes on the basis that the messages were identical.

V. Experimental Setup

In this section we describe the simulation environment and protocol configurations that were used to generate the results presented in Section VI.

A. Simulation Environment

The simulation environment was based on the JProwler network simulator [18]. JProwler is a discrete event simulator that can accurately model sensor nodes and the communications between them. JProwler provides two radio models, Gaussian and Rayleigh, which determine the signal level of transmissions and the communication range of nodes. The Rayleigh model was selected for use in all experiments because it models the situation where sensor nodes have

process j - If node is a normal node
variables
% Messages seen
messages: set of int init $\emptyset$

% The distance from the source to this node
realhop: int init 0;

% Number of messages seen from source
messagecounter: int init 0;

% Ignore choice variable
ignorechoose: int init 0;

constants
% Distance to the sink, probability threshold
Δ, σ: int, real;

actions
% Receiving choose message
receiveChoose:: **rcv**$\langle Choose, hash, ssd, hop, count\rangle \rightarrow$
if (*hash* $\notin$ *messages* $\wedge$ *ignorechoose* $= 0$) **then**
messages := *messages* $\cup \{hash\}$;
if ($\Delta = ssd$) **then**
possiblyBecomeFS(infinite duration, σ);
else
possiblyBecomeFS(temp duration);
fi; **fi**;

% Receiving fake messaget
receiveFake:: **rcv**$\langle Fake, hash\rangle \rightarrow$
if (*hash* $\notin$ *messages*) **then**
messages := *messages* $\cup \{hash\}$;
BCAST$\langle Fake, hash\rangle$;
fi;

% Receiving normal message
receiveNormal:: **rcv**$\langle Normal, hash, ssd, hop, count\rangle \rightarrow$
if (*hash* $\notin$ *messages*) **then**
messages := *messages* $\cup \{hash\}$;
messagecounter, realhop := *count*, $hop + 1$;
if (*messagecounter* $= 1 \wedge$ *realhop* $<= \frac{3}{4}$*ssd*) **then**
ignorechoose := 1;
fi;
BCAST$\langle Normal, hash, ssd, hop + 1, count\rangle$;
fi;

% Receiving away messaget
receiveAway:: **rcv**$\langle Away, hash, ssd, hop, count\rangle \rightarrow$
if (*hash* $\notin$ *messages*) **then**
messages := *messages* $\cup \{hash\}$;
if (*messagecounter* $<$ *count* $\vee$ *realhop* $>$ *ssd*) **then**
BCAST$\langle Choose, \textbf{hash}(Away), ssd, hop + 1, count\rangle$;
possiblyBecomeFS(temp duration);
fi;
fi;

Figure 1: Source Location Privacy Algorithm - Normal.

```
process j - If node is Source
variables
    % The number of messages sent
    count: int init 1;

    % rate: how fast messages are sent.
    rate: timer init δ;

constants
    % Distance to the sink
    Δ: int;

actions
    % Sending normal messages
    sendNormal:: timeout(rate) →
        BCAST⟨Normal, hash(Normal), Δ, 0, count⟩;
        count := count + 1;
        set(rate, δ);
```

Figure 2: Source Location Privacy Algorithm - Source.

```
process j - If node is Sink
variables
    % Messages seen
    messages: set of int init ∅

    % Sink sent indicator
    sinksent: int init 0;

actions
    % Receiving fake message
    receiveFake:: rcv⟨Fake, hash⟩ →
        if (hash ∉ messages) then
            messages := messages ∪ {hash};
            BCAST⟨Fake, hash⟩;
        fi;

    % Receiving normal message
    receiveNormal:: rcv⟨Normal, hash, ssd, hop, count⟩ →
        if (hash ∉ messages) then
            messages := messages ∪ {hash};
            if (sinksent = 0) then
                sinksent := 1;
                BCAST⟨Away, hash(Away), ssd, hop + 1, 1⟩;
            fi;
        fi;
```

Figure 3: Source Location Privacy Algorithm - Sink.

```
process j - If node is fake source
variables
    % rate: how fast messages are sent.
    % duration: how long we will stay a fake source.
    rate, duration: timer init α, β;

actions
    % Sending fake messages
    sendFake:: timeout(rate) →
        if (duration >= (currenttime − starttime)) then
            BCAST⟨Choose, hash(Choose), ssd, hop + 1, count⟩;
            BECOME NORMAL;
        else
            BCAST⟨Fake, hash(Fake)⟩;
            set(rate, α);
        fi;
```

Figure 4: Source Location Privacy Algorithm - Fake Source.

mobility, which is consistent with the assumption that an attacker will have mobility within a sensor network. The difference between the two radio models is that, with the Rayleigh model, the neighbourhood is updated whenever there is movement, which is important for the attacker. In case nodes are static, the Rayleigh model is similar to the Gaussian model.

B. Network Configuration

A square grid network layout of size $n \times n$ was used in all experiments, where $n \in \{11, 15, 21, 25\}$, i.e., networks with 121, 225, 441 and 625 nodes respectively. A single source node generated messages and a single sink node collected messages. The source and sink nodes were distinct. The rate at which messages from the real source were generated was varied. The sets of experiments for each network size and parameter configuration were performed for five source node locations; the four corners of the grid and a random location at the perimeter of the grid. To ensure the validity of the results presented, 500 repeats were performed for each source location, and for each combination of parameters. The sink node was located at the centre of the grid. Nodes were located 28 meters apart. The node separation distance was determined analytically, based on the static fading values calculated by the adopted radio model. This separation distance ensured that messages (i) pass through multiple nodes from source to sink, (ii) can move only one hop at a time and (iii) can only be passed to horizontally or vertically adjacent nodes.

C. Protocol Configuration

The flooding protocol was used as a baseline against which the extended flooding protocol was measured. Initially, we set the threshold σ to 0, i.e., when the *hop* value is 1, all nodes receiving the *FakeNode* message becomes a permanent fake source. In this paper, we are interested in the impact three factors about the protocol implementation have on the level of source location privacy. These are:

1) The duration d_i a temporary fake source n_i sends messages.
2) The rate δ_i at which a temporary or permanent fake source n_i sends messages.
3) The impact of threshold σ on capture ratio.

In the simulation, the durations over which a temporary fake source send messages were 1, 2 and 4 seconds respectively. For the message rates, we vary for the fake sources as well as for the real source. For the real source, the message rates used were 1, 2 and 4 messages per second, whereas the message rates for the fake sources, in general, were 2, 4 and 8 messages per second. We also never run simulations where the message rate of the real source is higher than that of the fake source, since the capture ratio will then be 100% (as the attacker is "attracted" faster towards the real source than towards the fake source).

Table I: Safety period for each network size and send rate.

Network Size	Safety Period			
	1/sec	2/sec	4/sec	8/sec
11×11	31.85	16.43	8.16	8.24
15×15	46.35	23.27	12.25	13.17
21×21	67.30	34.23	18.71	19.30
25×25	84.88	42.75	22.79	23.95

D. Safety Period

A concept called *safety period* was introduced in [1] to capture the number of messages that has to be sent by the real source before it gets detected. In general, for maximum privacy, the safety period should ideally be very high. In this paper, we use an alternative, but similar, definition for safety period: for each network size and source rate, using flooding, we calculate the average time it takes to detect the real source (i.e., capture the asset). When we run simulations for SLP, we allow for a higher safety period, since the premise is that our proposed extended routing algorithm will provide a higher source location privacy, and hence may require more time for source capture. The reason for using this definition of safety period is that it bounds the simulation time. The safety period, for each network size and rate, for flooding is shown in Table I. The safety period is double the average time taken for source detection, i.e., for capture, since it allows an attacker to go at the opposite end of the network and back.

E. Simulation experiments

An experiment constituted a single execution of the simulation environment using a specified protocol configuration, network size and safety period. An experiment terminated when the source node had been captured by an attacker or the safety period has expired. JProwler was extended to allow the safety period and the capture ratio to be monitored during simulation.

VI. Results

In this section, we present the results of our experimentation under various protocol parameterisations.

A. Metrics

We use the metric *capture ratio* to indicate the level of SLP. The capture ratio metric is calculated as the ratio of the number of runs in which the source is captured to the total number of runs. If the capture ratio is 0, then it means that the SLP is maximal, whilst the SLP is minimal if the capture ratio is 1. Further, we assume all fake message transmission duration t_i to be equal for all temporary fake sources, and the message rate λ_i to be equal for all temporary fake sources.

B. Impact of Temporary Fake Source Duration on SLP

In this section, we investigate the impact of message rates on SLP. Specifically, we wish to understand how the duration during which a fake source sends fake messages affect SLP. Intuitively, the higher the duration, the higher the chance of the attacker being "pulled back", decreasing the capture ratio.

It can be observed from Figure 5 that, for a given real source broadcast rate and fake source broadcast rate, the capture ratio generally decreases for increasing temporary fake source duration. A similar observation can be made from Figure 6. We also observe in Figures 5 and 6 that under a specific parameterisation of the protocol, it is possible to obtain (almost) maximal SLP, i.e., capture ratio = 0. However, when capture ratio is not 0, then there was a high number of message collisions due to high message rate or low network size (thereby causing a high message density, with a higher likelihood of collisions).

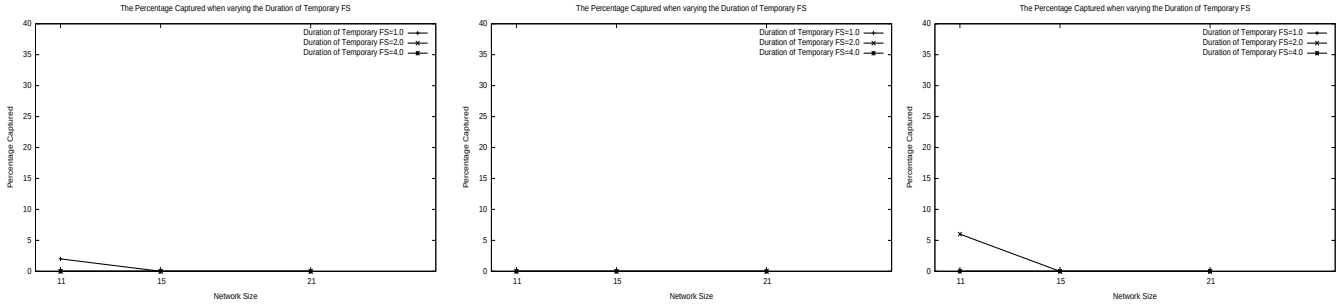
C. Impact of Message Rates on SLP

In this section, we analyse the impact of varying the rates at which the real source and the fake sources send messages on SLP. Specifically, we wish to understand how the rates at which the real and the fake sources send messages affect SLP. Intuitively, the higher the rate for the fake sources, the higher the chance of the attacker being "pulled back", decreasing the capture ratio. Also, the lower the message rate for the real source, the lower the capture ratio, as the real source cannot "attract" the attacker fast enough.

It can be observed from Figures 7 and 8 that increasing the rate at which fake sources send messages leads to a decrease in the capture ratio. For example, in Figure 8c, when the fake sends 8 messages per second (period of 0.125s), the capture ratio is almost 0, whereas the capture ratio increases when the message rate is decreased to 0.5. We also observe that the capture ratio increases whenever the real source sends messages at a higher rate (as can be observed from Figures 7 and 8), thereby confirming our initial hypothesis. Another observation in Figures 7 and 8 is that, under a specific parameterisation of the protocol, it is possible to obtain near-maximal SLP, i.e., capture ratio = 0.

D. Limitations

The algorithm we have proposed has some limitations. First, the algorithm works for the network structure we have defined, i.e., a grid structure with a sink at the centre. However, we emphasise that the network need not be a grid, as long as it can be "construed" as a grid. Further, for the algorithm to work in a different network topology, the protocol for the fake sources needs to be altered. Second, we have assumed that the attacker follows messages whenever he receives one. This precludes an "intelligent" attacker who can decide to go in one direction only, when he observes that messages may be coming from various directions. However,

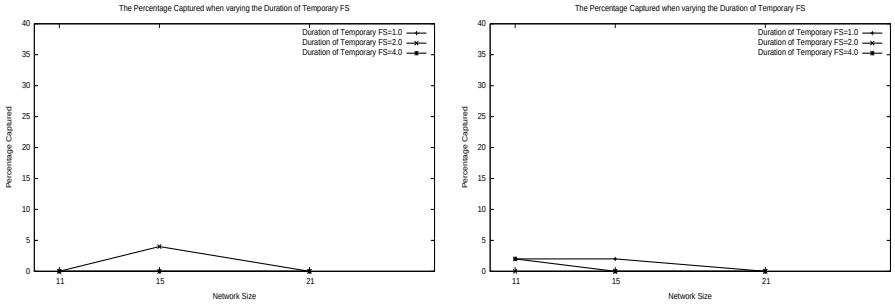


(a) Capture ratio: Real Source Rate: 1/s, Fake Source Rate: 8/s

(b) Capture ratio: Real Source Rate: 1/s, Fake Source Rate: 4/s

(c) Capture ratio: Real Source Rate: 1/s, Fake Source Rate: 2/s

Figure 5: Duration: Capture ratio for varying duration and fake message rates.



(a) Capture ratio: Real Source Rate: 2/s, Fake Source Rate: 8/s

(b) Capture ratio: Real Source Rate: 2/s, Fake Source Rate: 4/s

Figure 6: Duration: Capture ratio for varying duration and fake message rates.

such an assumption provides an upper bound on the level of privacy that can be obtained.

VII. Conclusion

In this paper, we have explored the fake source technique for providing SLP in WSNs. We have made several novel contributions in this paper: (i) we have shown the problem to be NP-complete, (ii) we have identified two important parameters, namely message rates and fake message transmission duration, that impact upon the level of privacy an algorithm can provide, and (iii) we have provided a heuristic which can provide optimal privacy under specific parameter settings. In future work, we will address the limitations we have identified previously. Specifically, we will test our algorithms on various network topologies to ascertain its efficiency. We will consider fake sources in the context of a more perceptive attacker.

References

[1] U. Kamat, Y. Zhang, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, November 2005, pp. 599–608.

[2] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Communications of the ACM - Special Issue on Wireless Sensor Networks*, vol. 47, no. 6, pp. 53–57, June 2004.

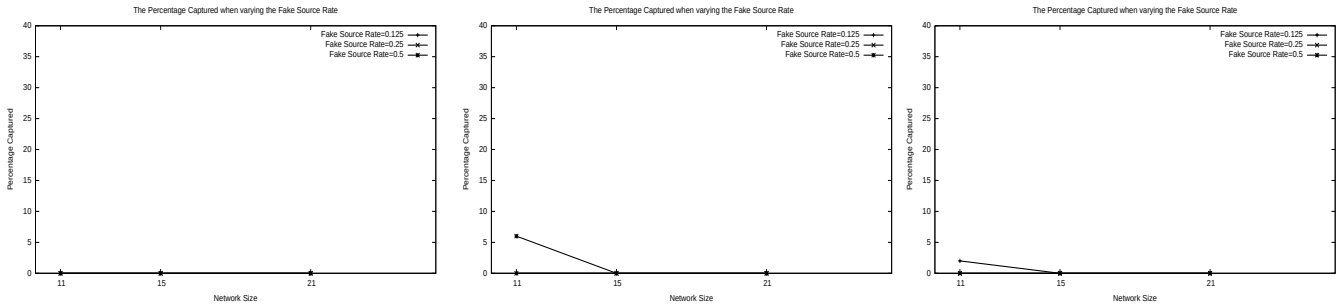
[3] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, July 1982.

[4] M. Nesterenko and S. Tixeuil, "Discovering network topology in the presence of byzantine faults," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 12, pp. 1777–1789, December 2009.

[5] I. C. Avramopoulos, H. Kobayashi, R. Wang, and A. Krishnamurthy, "Highly secure and efficient routing," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, March 2004, pp. 197–208.

[6] K. Mehta, D. Liu, and M. Wright, "Location privacy in sensor networks against a global eavesdropper," in *Proceedings of the 15th IEEE International Conference on Network Protocols*, October 2007, pp. 314–323.

[7] S. Armenia, G. Morabito, and S. Palazzo, "Analysis of location privacy/energy efficiency tradeoffs in wireless sensor networks," in *Proceedings of the 6th International IFIP-TC6 Networking Conference on Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, November 2007, pp. 215–226.

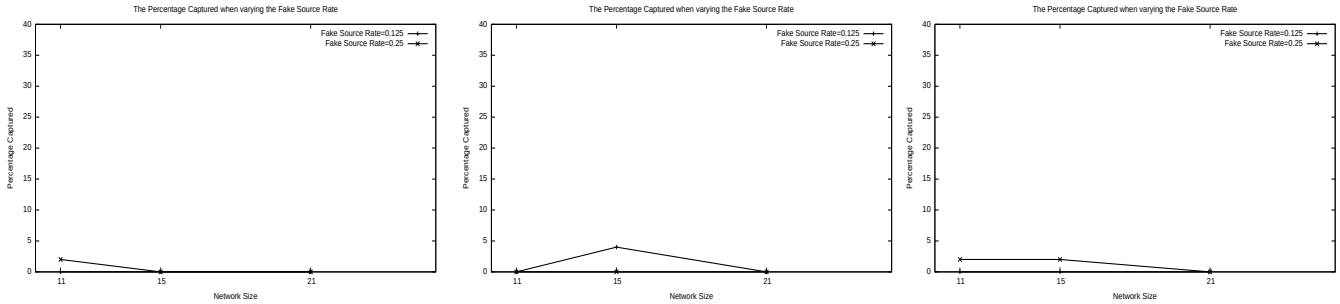


(a) Capture ratio: Real Source Rate: 1/s, Duration of FS: 4 s

(b) Capture ratio: Real Source Rate: 1/s, Duration of FS: 2 s

(c) Capture ratio: Real Source Rate: 1/s, Duration of FS: 1 s

Figure 7: Message rates: Capture ratio for varying duration and fake message rates.



(a) Capture ratio: Real Source Rate: 2/s, Duration of FS: 4 s

(b) Capture ratio: Real Source Rate: 2/s, Duration of FS: 2 s

(c) Capture ratio: Real Source Rate: 2/s, Duration of FS: 1 s

Figure 8: Message rates: Capture ratio for varying duration and fake message rates.

[8] S.-W. Lee, Y.-H. Park, J.-H. Son, S.-W. Seo, U. Kang, H.-K. Moon, and M.-S. Lee, "Source-location privacy in wireless sensor networks," *Korea Institute of Information Security and Cryptology Journal*, vol. 17, no. 2, pp. 125–137, April 2007.

[9] J. Yao and G. Wen, "Preserving source-location privacy in energy-constrained wireless sensor networks," in *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops*, June 2008, pp. 412–416.

[10] A. Jhumka, M. Leeke, and S. Shrestha, "On the use of fake sources for source location privacy: Trade-offs between energy and privacy," *The Computer Journal*, vol. (To Appear), 2011.

[11] C. Ozturk, Y. Zhang, and W. Trappe, "Source-location privacy in energy-constrained sensor network routing," in *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks*, October 2004, pp. 88–93.

[12] J. Deng, R. Han, and S. Mishra, "Countermeasures against traffic analysis attacks in wireless sensor networks," in *Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks*, 2005, pp. 113–126.

[13] Y. Ouyang, Z. Le, D. Liu, J. Ford, and F. Makedon, "Source location privacy against laptop-class attacks in sensor networks," in *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks*, September 2008, pp. 22–25.

[14] P. Kamat, W. Xu, W. Trappe, and Y. Zhang, "Temporal privacy in wireless sensor networks: Theory and practice," *ACM Transactions on Sensor Networks*, vol. 5, no. 4, p. 24 (Article 28), November 2009.

[15] A. Jhumka, "Crash-tolerant collision-free data aggregation scheduling in wireless sensor networks," in *Proceedings 29th IEEE Symposium on Reliable Distributed Systems*, October 2010, pp. 44–53.

[16] Z. Benenson, P. M. Cholewinski, and F. C. Freiling, *Wireless Sensor Network Security*. IOS Press, April 2008, ch. Vulnerabilities and Attacks in Wireless Sensor Networks, pp. 22–43.

[17] A. Jhumka, M. Leeke, and S. Shrestha, "On the use of fake sources for source location privacy: Trade-offs between energy and privacy," *The Computer Journal*, vol. 54, no. 6, pp. 860–874, June 2011.

[18] JProwler, "http://w3.isis.vanderbilt.edu/projects/nest/jprowler/," March 2012.