

A Near-Optimal Source Location Privacy Scheme for Wireless Sensor Networks

Matthew Bradbury and Arshad Jhumka
Department of Computer Science
University of Warwick, Coventry,
United Kingdom, CV4 7AL
{M.Bradbury, H.A.Jhumka}@warwick.ac.uk

Abstract—As interest in using Wireless Sensor Networks (WSNs) for deployments in scenarios such as asset monitoring increases, the need to consider security and privacy issues also becomes greater. One such issue is that of Source Location Privacy (SLP) where the location of a source in the network needs to be kept secret from a malicious attacker. Many techniques have been proposed to provide SLP against an eavesdropping attacker. Most techniques work by first developing an algorithm followed by extensive performance validation. Differently, in this paper, we model the SLP problem as an Integer Linear Programming optimization problem. Using the IBM ILOG CPLEX optimiser, we obtain an optimal solution to provide SLP. However, that solution is centralized (i.e., requires network-wide knowledge) making the solution unsuitable for WSNs. Therefore, we develop a distributed version of the solution and evaluate the level of privacy provided by the distributed solution. The solution is *hybrid* in nature, in that it uses both spatial and temporal redundancy to provide SLP. Results from extensive simulations using the TOSSIM WSN simulator indicate a 1% capture ratio is achievable as a trade-off for an increase in the delivery latency.

Index Terms—Source Location Privacy, Wireless Sensor Networks, Optimal Routing, Integer Linear Programming.

I. INTRODUCTION

As wireless sensor networks (WSNs) are becoming increasingly viable for large scale deployments, applications such as asset monitoring and tracking can be implemented using this technology [1, 2]. In an asset monitoring application, the nodes in the network monitor their surrounding area to detect the presence of a valuable asset. When such an asset is detected a message is transmitted from that *source* node back to a base station called the *sink*. As the range of the sensor nodes is typically less than that between the source and sink, multiple hops need to be used for the message to reach the sink.

To protect the *content* of this data, messages can be encrypted, however, the act of routing the message to the sink reveals *context* information about the event that encryption does not and can not protect. One example of context information that needs to be protected is the location of the source, as revealing this information would allow an attacker to follow messages through the network to the source and capture it. The SLP problem was originally presented as the panda hunter game [3, 4]. In this situation pandas are being monitored by a WSN and information is being reported to conservationists working at a base station. Other situations such routing

messages between soldiers on a battlefield would benefit from applying SLP techniques.

Much work has been done on developing new routing strategies that provide SLP [5] since the seminal work. Most works first develop a technique before using simulations to demonstrate the performance of the technique and some works use techniques such as information theoretic analysis to quantify the privacy loss of a routing strategy. Often, the solution space will capture the trade-offs inherent in the problem such as privacy level, energy usage and message delivery ratio/latency. However, it is often challenging to appropriately parametrise the algorithms to achieve the right trade-offs.

Constraint programming (CP) is a technique which applies a set of constraints to decision variables with the aim to maximise or minimise an objective function. Generic solvers then process all three parts and output an optimal solution (if one is possible). Many problems have been expressed as constraint satisfaction problems (CSPs) such as scheduling, planning and networking. Integer linear programming (ILP) is a subset of CP where the relations in the constraints must be linear. Different to all current approaches, in this paper we model the scheduling of messages being routed from the source to the sink as an ILP problem and then use a solver to obtain an optimal solution, if it exists. However, the model solution assumes network-wide knowledge, making it unsuitable for WSN deployment. Making use of the structure of the optimal solution, we propose a novel distributed hybrid algorithm to provide SLP. Our results show that the protocol is near-optimal under certain parametrisations, independent of network size and application parameters such as the period between the source sending messages.

Our contributions are:

- 1) We model SLP-aware routing protocol from a source to a sink as an ILP optimisation problem.
- 2) We develop a novel distributed routing protocol inspired from the output of the ILP model.
- 3) We perform simulations of the routing protocol using TOSSIM and results show that the protocol is near optimal in terms of privacy level, i.e., a low 1% capture ratio for certain parametrisations.

The rest of the paper is structured as follows. In Section II we discuss the related work. We then describe the ILP model in Section III, the results of the model and how it influenced our algorithm's design in Section IV, and the routing protocol in

Section V. Section VI contains the experimental setup for the results in Section VII. We discuss implications in Section VIII and finally conclude with a summary in Section IX.

II. RELATED WORK

Source location privacy was originally introduced in [3, 4] where the authors introduced Phantom Routing a two-stage routing protocol to provide SLP. The first stage involves a message taking a directed random walk either towards or away from some landmark node. Once the message reaches the landmark node it is forwarded to the sink in the second stage. Two variants were presented, PRS [4] which uses flooding in the second stage and PSRS [3] which uses single path routing.

Many routing-based SLP schemes have since been proposed [5]. Several solutions aimed to improve the directed random walk phase of phantom routing, such as angle-based routing [6] which chooses the next node based on angles between key nodes. Other routing-based techniques involve routing in a ring around the sink before finally reaching it [7].

An alternative technique initially proposed in the seminal work, was to use fake sources that broadcast identical messages to the real sources. The seminal work concluded that on their own they do not assist in providing SLP, but this was rebutted in [8] and many other techniques have since made use of fake messages. Fake sources provide SLP by generating fake messages that lead the attacker away from the real sources [9]. Recent work has focused on dynamically determine good parameters to use online [10]. However, fake source techniques are often criticised for their high energy usage.

Many techniques have since combined routing and fake sources to improve the levels of SLP provided. One example is tree-based diversionary routing [11] which imposes a tree structure on the network and then routes fake messages through the tree. PEM in [12] extends the routing path from source to sink with branches of fake sources. The idea of fogs or clouds [13, 14] is also similar where a message is routed round a group of nodes called a fog and then onwards to other fogs. Fake messages are also used to provide additional privacy.

Other techniques to provide SLP against local attackers include using space in MAC beacon frames to disseminate messages from the source to other areas in the network before being routed to the sink [15]. Coordinated jamming [16] which uses jamming to defeat localisation techniques. And, data mules [17, 18] where a mobile agent gathering messages near the source and moves near the sink before broadcasting them.

The solutions discussed so far have focused on a local distributed eavesdropping attacker and there are other capabilities which can be considered [19]. Attackers with a global view of the network are one such example. Techniques that provide privacy against them tend to take different approaches to local attackers. For example some techniques require that all nodes broadcast periodically according to some ruleset even if there is no message to send [20, 21]. Other traffic decorrelation techniques [22] have also been used.

A popular technique for generating an optimal solution to problems such as (i) optimising sensor node deployment

TABLE I: Model parameters

Symbol	Description
$ V $	The number of nodes
\mathcal{R}	The communication range of the nodes
\mathcal{S}	A set of node ids that send messages
$sink$	The sink id
\mathcal{C}	A mapping between node ids and their 2D coordinates
\mathcal{A}_{start}	The id of the node the attacker starts at
\mathcal{R}_A	The eavesdropping range of the attacker
P_{src}	The time between the source sending messages
P_{safety}	The maximum simulation time
Ω	How many slots a second is divided into

locations [23], (ii) energy efficient routing [24], and (iii) others [25], is constraint programming (CP). CP is a type of programming described using a set of decision variables that store the output of the model, an objective function that ranks the goodness of variables and a set of constraints that define what values for a solution are valid. Using a solver, an optimal solution for the problem can be obtained.

Similar to CP, is integer linear programming (ILP) where the relations specified in the constraints between the variables must be linear. ILP has been used to find solutions that optimise for energy usage or delivery latency in a WSN. Our approach will instead optimise the routing of messages to provide SLP.

III. CONSTRAINT PROGRAMMING MODEL

In this section we describe the ILP model¹, including the rationale for certain constraints. Our ILP model is written in IBM's Optimisation Programming Language (OPL). There are multiple ILP solvers present with IBM ILOG CPLEX version 12.6.3. We used the CP solver as it was able to produce optimal results faster and also scale to larger network sizes than the CPLEX solver. The CP solver proves optimality by showing that no better solution can be found, while the CPLEX solver obtains a lower bound proof using cuts and linear relaxation.²

The network was modelled as a directed graph $G = (V, A)$ where V is the set of nodes and A is the set of arcs. An arc is a 2-tuple (u, v) where u is the origin and v is the target. Each node in V was assigned a 2D coordinate \mathcal{C} and the euclidean distance was calculated between each node. If the distance between two different nodes was less than or equal to the communication range \mathcal{R} of the sensor nodes, then that arc is present in A . The paths that messages can travel along is defined by the arcs in A and the paths the attacker can move along is modelled as a directed graph $G_A = (V, A_A)$ routed at the node with id \mathcal{A}_{start} where the attacker starts. If the distance between two nodes is less than the attacker's eavesdropping range \mathcal{R}_A then A_A contains that arc, this also means the attacker can move along that arc. Once the attacker finds the source it will remain at that location, so the attacker cannot move along arcs which start at the source. As the

¹The source code for this model can be found at bitbucket.org/MBradbury/slp-attacker-ilp/raw/a4e326e/SLP/SLP.mod

²ibm.com/software/integration/optimization/cplex-cp-optimizer/mp-cp

attacker moves through the network, it can only move to be co-located with another node. In this paper, we assume the attacker's eavesdropping range is equal to the sensor node's transmission range ($\mathcal{R} = \mathcal{R}_A$).

The sink is a special node to which messages are routed and the sources (\mathcal{S}) are nodes that generate one message every P_{src} . We assume there is a single sink and source.

There is an upper bound on the duration of the model (P_{safety}), within which routing and attacker movement is considered. Time (\mathcal{T}) is discretised by dividing it into slots, there are Ω slots per second. Each node can send a single message in each slot. A node may choose to send no messages in a given slot. The attacker can therefore either move in response to a single message or not move at all in a given slot.

To ensure the attacker could reach the source, the attacker's starting position was never set so its distance in hops from the source was greater than the $P_{src} \cdot \Omega$. As otherwise the safety period may be reached before the attacker has a chance to capture the source. The attacker can respond to a message a neighbour sends in a time slot if that message had not been previously responded to. If no messages are sent by a neighbour the attacker must remain where it is.

Time 0 is special as it is used to set the attacker's position, no messages are sent at this time. After time 0 when a node broadcasts a message we assume network links are perfect and all neighbours receive the message, even if multiple nodes broadcast in that time slot. In this case we assume a collision detection and retransmission is used to ensure message delivery.

Messages generated by a source (\mathcal{M}) are represented by a 2-tuple where the first element (src) is the id of the source that generated the message and the second element (seq) is the message number. This ensures each message is unique.

$$\widehat{P_{safety}} = [\Omega \cdot P_{safety}] \quad (1)$$

$$\widehat{P_{src}} = [\Omega \cdot P_{src}] \quad (2)$$

$$\mathcal{T} = 0..\widehat{P_{safety}} \quad (3)$$

$$\mathcal{T}_1 = 1..\widehat{P_{safety}} \quad (4)$$

$$\mathcal{M} = \left\{ (src, seq) \mid src \in \mathcal{S}, \right. \\ \left. seq \in 1..\lceil P_{safety} \cdot P_{src} \rceil \right\} \quad (5)$$

$$D(i, j) = \sqrt{(i.x - j.x)^2 + (i.y - j.y)^2} \quad (6)$$

$$A = \{ (u, v) \mid u, v \in V, D(u, v) \leq \mathcal{R} \wedge u \neq v \} \quad (7)$$

$$N(i) = \{ j \mid (i, j) \in A, i \neq j \} \quad (8)$$

$$A_A = \{ (u, v) \mid u, v \in V, D(u, v) \leq \mathcal{R}_A \} \setminus \\ \{ (s, v) \mid s \in \mathcal{S}, v \in V, s \neq v \} \quad (9)$$

$$N_A(i) = \{ j \mid (i, j) \in A_A, i \neq j \} \quad (10)$$

A. Objective Function

Two decision variables are used to capture the output of the model. The broadcasts performed is a three dimensional array of booleans with the node ids, message and time as the dimensions ($\mathcal{B} : V \times \mathcal{M} \times \mathcal{T} \rightarrow \mathbb{B}$). This variable is intended to capture whether a node sends a message at a given time. The other decision variable is the attacker path which is a two dimensional

array of booleans with time and the arcs an attacker can take as the dimensions ($\mathcal{P}_A : \mathcal{T} \times A_A \rightarrow \mathbb{B}$). This variable captures whether at a given time an attacker moves along an arc.

The objective function for this model is to maximise the distance between the attacker's final position and the source(s) in the network. As the aim in providing SLP is to prevent the attacker from reaching the source, we can say the further the attacker is from the source the better SLP has been provided within a safety period.

$$\begin{aligned} & \text{maximise} && \sum_{\substack{s \in \mathcal{S} \\ e \in A_A}} \mathcal{P}_A(\widehat{P_{safety}}, e) \cdot D(s, e.v) \\ & \text{subject to} && \text{Routing Constraints ctR1 to ctR6,} \\ & && \text{Attacker Constraints ctA1 to ctA7.} \end{aligned} \quad (11)$$

B. Routing Constraints

In this section we detail the constraints on how messages are generated by the source and how they are subsequently routed through the network.

ctR1 At $t = 0$, no messages are broadcasted.

ctR2 From $t > 0$, each source node generates a message every P_{src} until the safety period is reached.

ctR3 No node can broadcast more than a single message concurrently. This means that in a given time slot a node must send one message or no messages.

ctR4 Once a message is broadcasted by a node it is not broadcasted by that node again.

ctR5 A node can only forward a message after a neighbour broadcasted that message in a previous time slot.

ctR6 All messages sent by the sources must reach the sink.

$$\forall n \in V \cdot \forall m \in \mathcal{M} \cdot \mathcal{B}(n, m, 0) = 0 \quad (\text{ctR1})$$

$$\forall n \in \mathcal{S} \cdot \forall m \in \mathcal{M} : m.src = n \cdot \\ \mathcal{B}(n, m, (m.seq - 1) \cdot \widehat{P_{src}} + 1) \quad (\text{ctR2})$$

$$\forall \tau \in \mathcal{T}_1 \cdot \forall n \in V \cdot 1 \geq \sum_{m \in \mathcal{M}} \mathcal{B}(n, m, \tau) \quad (\text{ctR3})$$

$$\forall m \in \mathcal{M} \cdot \forall n \in V \cdot \forall \tau_1 \in \mathcal{T}_1 \cdot \\ \mathcal{B}(n, m, \tau_1) \implies 0 = \sum_{\substack{\tau_2 \in \mathcal{T}, \tau_2 > \tau_1}} \mathcal{B}(n, m, \tau_2) \quad (\text{ctR4})$$

$$\forall n \in (V \setminus \mathcal{S}) \cdot \forall m \in \mathcal{M} \cdot \forall \tau_1 \in \mathcal{T}_1 \cdot \\ \mathcal{B}(n, m, \tau_1) \implies 1 \leq \sum_{\substack{neigh \in N(n) \\ \tau_2 \in \mathcal{T}, 0 < \tau_2 < \tau_1}} \mathcal{B}(neigh, m, \tau_2) \quad (\text{ctR5})$$

$$\forall m \in \mathcal{M} \cdot 1 \leq \sum_{\substack{n \in N(sink) \\ \tau \in \mathcal{T}_1}} \mathcal{B}(n, m, \tau) \quad (\text{ctR6})$$

C. Attacker Constraints

This section details the constraints on how the attacker moves through the network. These constraints have been constructed so the attacker will follow the messages it receives. If the attacker was allowed to move independently of messages then it would move away from the source due to the objective function. Therefore, we require that (i) the attacker only moves

in response to messages (ctA4, ctA7) and (ii) the attacker will only consider a message once as otherwise it might follow that message as it moves away from the source (ctA5, ctA6).

To simplify attacker constraints four predicates about the attacker's movement are defined. AM2A checks if the attacker moved to n at time τ . AM2NA checks if the attacker moved to a neighbour of n at τ . ASM checks if the attacker remained where it was at τ . AMBA checks if an attacker moved because of a message m at τ .

$$\text{AM2A}(n, \tau) = 1 = \sum_{e \in A_A, e.v=n} \mathcal{P}_A(\tau, e) \quad (12)$$

$$\text{AM2NA}(n, \tau) = 1 = \sum_{\text{neigh} \in N_A(n)} \text{AM2A}(\text{neigh}, \tau) \quad (13)$$

$$\text{ASM}(\tau) = 1 = \sum_{e \in A_A, e.u=e.v} \mathcal{P}_A(\tau, e) \quad (14)$$

$$\text{AMBA}(m, \tau) = 1 = \sum_{e \in A_A, e.u \neq e.v} \mathcal{P}_A(\tau, e) \wedge \mathcal{B}(e.v, m, \tau) \quad (15)$$

ctA1 At $t = 0$ the attacker moves from the attacker's starting position to that same position.

ctA2 The attacker makes exactly one move each time slot.

ctA3 A move must be from the attacker's current location.

ctA4 If the attacker moves to n from m at time τ , then it must be because at time τ the node n broadcasted a message.

ctA5 If the attacker receives a message that it has not previously moved in response to, then the attacker moves in response to that message.

ctA6 If the attacker moved in response to a message at time τ , then at no time $\tau' > \tau$ will the attacker move in response to that message again.

ctA7 If the attacker is at node n and no neighbours send a message, then the attacker moves along the (n, n) edge.

$$\mathcal{P}_A(0, (A_{\text{start}}, A_{\text{start}})) = 1 \quad (\text{ctA1})$$

$$\forall \tau \in \mathcal{T} \cdot 1 = \sum_{e \in A_A} \mathcal{P}_A(\tau, e) \quad (\text{ctA2})$$

$$\forall \tau \in \mathcal{T}_1 \cdot 1 = \sum_{e_1, e_2 \in A_A, e_1.v=e_2.u} \mathcal{P}_A(\tau-1, e_1) \wedge \mathcal{P}_A(\tau, e_2) \quad (\text{ctA3})$$

$$\forall n \in V \cdot \forall \tau \in \mathcal{T}_1 \cdot$$

$$1 = \sum_{e \in A_A, e.v=n \wedge e.u \neq e.v} \mathcal{P}_A(\tau, e) \implies 1 = \sum_{m \in \mathcal{M}} \mathcal{B}(n, m, \tau) \quad (\text{ctA4})$$

$$\forall n \in V \cdot \forall m \in \mathcal{M} \cdot \forall \tau \in \mathcal{T}_1 \cdot$$

$$\mathcal{B}(n, m, \tau) \wedge \text{AM2NA}(n, \tau-1) \wedge 0 = \sum_{t_2 \in \mathcal{T}, 0 < \tau_2 < \tau} \text{AMBA}(m, \tau_2) \implies \text{AM2A}(n, \tau) \quad (\text{ctA5})$$

$$\forall \tau_1 \in \mathcal{T}_1 \cdot \forall m \in \mathcal{M} \cdot$$

$$\text{AMBA}(m, \tau_1) \implies 0 = \sum_{\tau_2 \in \mathcal{T}, \tau_2 > \tau_1} \text{AMBA}(m, \tau_2) \quad (\text{ctA6})$$

$$\forall \tau \in \mathcal{T}_1 \cdot \forall e \in A_A \cdot$$

$$\mathcal{P}_A(\tau-1, e) \wedge 0 = \sum_{\substack{n \in N_A(e.v) \\ m \in \mathcal{M}}} \mathcal{B}(n, m, \tau) \implies \text{ASM}(\tau) \quad (\text{ctA7})$$

IV. MODEL RESULTS

In this section we describe the output from the ILP model for one combination of parameters³. We ran the model for a variety of different parameters including 3x3, 4x4 and 5x5 grids with the sink and source at different locations. Larger networks became infeasible to run due to the large amount of memory required. The results in Figure 1 show a 5x5 network with the source in the top left corner and the attacker starting at the sink which was positioned in the centre. This configuration was chosen as it has been previously investigated [9, 10] and allowed the state space of the model to be small enough for a solution to be found within a reasonable time.

Figures 1a to 1g show the pattern of broadcasts and Figure 1h shows how the distance of the attacker from the source changes over time. From the results we observed four major trends:

- 1) The routing path should go around the sink and approach from the opposite direction to the source.
- 2) Some routes should take the shortest path from the source to the sink.
- 3) Messages should be delayed so multiple messages are grouped together.
- 4) Messages should be delayed as late as possible with respect to the safety period.

Some of these observations have been used in previous work, while others have not. For example, having the route approach the sink from a direction other than the one the source is in has been used in Phantom Routing [3] and Ring-based Routing [7]. We have also seen algorithms whose routes occasionally take the shortest path [26]. However, we have not seen delaying and grouping messages previously in providing SLP. By delaying and grouping messages the attacker should end up with less time to make moves towards the sink within the safety period as shown by Figure 1h, where the attacker moves mainly near the end of the simulation (towards the top of the graph). By grouping and allowing messages to be delivered out-of-order we also force the attacker to require a larger memory to ensure previously received messages are ignored in the future.

V. ROUTING ALGORITHM

In this section we describe the implementation of the routing algorithm inspired by the model's output. There are four stages:

- 1) **AvoidSink**: In this first stage messages are routed around the sink.
- 2) **Backtrack**: Messages may end up attempting to go towards the sink and not having any valid routes, so they need to backtrack.
- 3) **ToSink**: Once a message has finished routing to avoid the sink it needs to be delivered to the sink.
- 4) **FromSink**: Finally, the message is sent in a starburst from the sink.

A. Stage 1: Avoid Sink

In order to provide SLP the most important thing we need to do is be able to route messages reliably around the sink and have

³Other results are available at warwick.ac.uk/matthewbradbury/slp/ilp

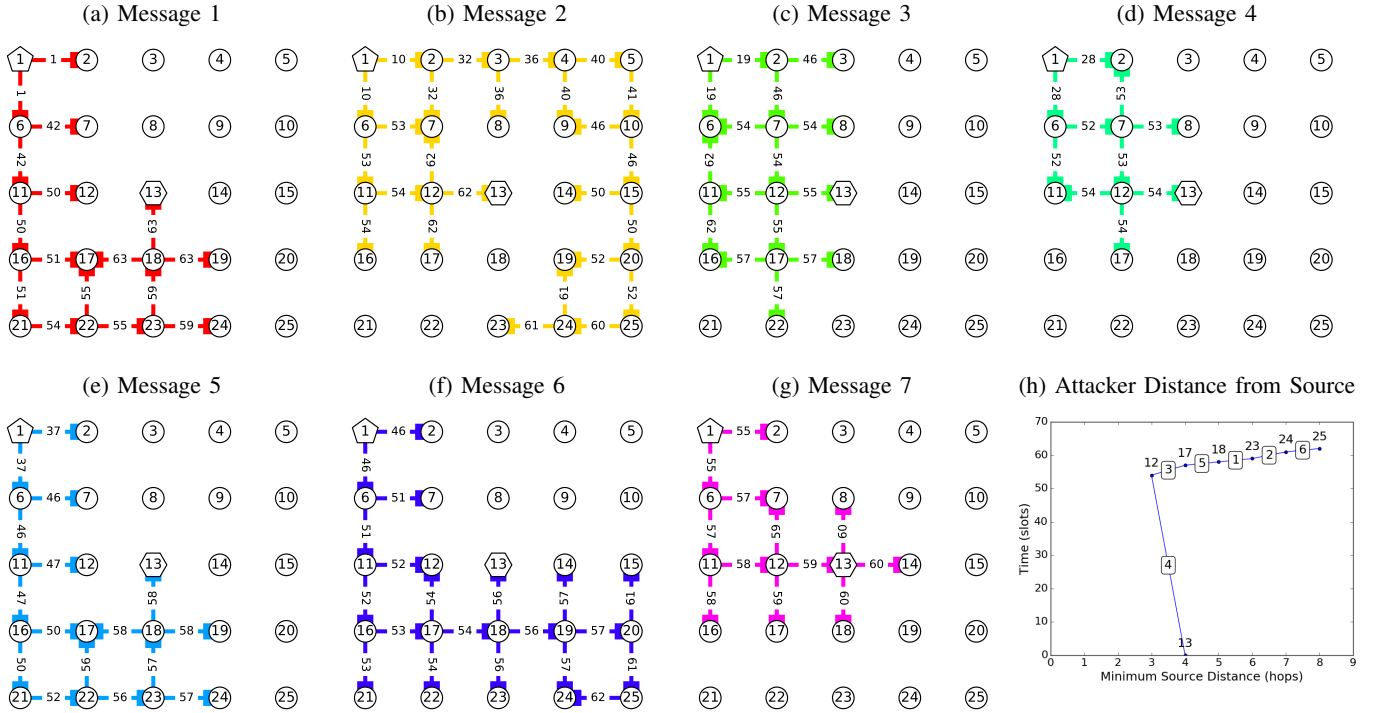


Fig. 1: An output of the ILP model for a 5x5 network with node 1 sending 7 messages to the sink at node 13 with a period of 1 second and 1 second is broken up into 9 slots. The attacker starts at node 13. Message broadcasts are represented by arrows from the sending node to the receiving nodes, the arrows are labelled with the time slot of the broadcast. Lines in (h) are labelled with the message number the attacker responded to and the location of the attacker is shown above the point.

them approach it from the opposite direction that the source is in. This requires every node to be aware of its neighbours, as one will be chosen to be the next in the path. We also require that every node n knows its distance to the sink $\Delta_{sink}(n)$, its distance from the source $\Delta_{src}(n)$ and the distance between the sink and source Δ_{ss} . Each node should know what values its 1-hop neighbours have for this information.

There is a parameter called *message group size* which specifies how many messages to group together. By grouping together the algorithm will delay the messages each hop so that the messages reach Δ_{ss} hops travelled at the same time. Messages sent earlier in the group will be delayed longer. The delay is specified by the equation below, where i is the position in the message group, P_{src} is the source period and α is the time it takes a message to send from one node to another. This is calculated at the source when the message is sent.

$$delay(i) = \frac{i \cdot P_{src} + \alpha \cdot \Delta_{ss}}{\Delta_{ss}} \quad (16)$$

Retransmissions are used to ensure reliability along the route. If retransmissions to a target fail a fixed number of times, then that target is blacklisted and another node will attempt to be used. If further retransmissions fail, then a poll message is sent which asks neighbours to broadcast their information. This is performed to ensure that the most up-to-date information is being used to make the routing decisions. Retransmissions are stopped when an acknowledgement packet is received or the maximum number of retransmissions have been sent.

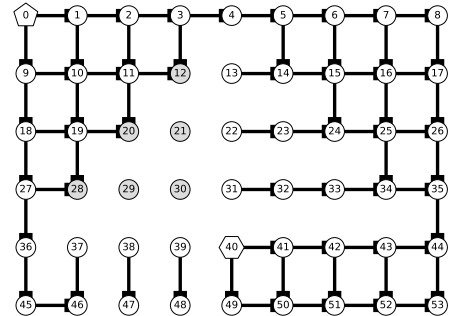


Fig. 2: Showing the links a node n has between each m neighbour in its one hop neighbourhood that satisfies: $\Delta_{src}(m) > \Delta_{src}(n) \wedge (\Delta_{sink}(n) > \frac{\Delta_{ss}}{2} \vee \Delta_{sink}(m) \geq \Delta_{sink}(n))$. Nodes that have no neighbours that satisfy this are marked in gray. The source is a pentagon and the sink is a hexagon.

B. Stage 2: Backtrack

There is the possibility that a message may reach a node which has no further neighbours to choose from, for example node 20 in Figure 2. In this case the message backtracks to a node that is further from the sink that was not the previous hop in the route. The next hop is then chosen using the avoid sink logic. This should allow a message to avoid the area of nodes that should not be used in the routing path (highlighted in grey in Figure 2). Backtracking is only done on node n when $\Delta_{src}(n) < \Delta_{ss}$ as this is a node that would otherwise lead the message closer to the sink.

C. Stage 3: To Sink

When a node's $\Delta_{src}(n) \geq \Delta_{ss}$ then the message has passed the area that would lead it towards the source. When this message has reached a node with no neighbours that are further from the sink, the message is at a local maxima for the sink distance, so the message is routed back towards the sink. When a message reaches a certain number of hops travelled it is routed back to the sink. This is to ensure bounded message latency and to prevent the message going too far on very large networks. As with routing to avoid the sink, retransmissions and blacklisting are used to provide a reliable message transmission to the sink.

D. Stage 4: From Sink

Finally, once the message reaches the sink, it is broadcasted in a starburst pattern away from the sink in all directions for a limited number of hops. Model results where the sink was the furthest point required this behaviour to ensure that the attacker is lured to the furthest location from the source. This behaviour was also observed in Figure 1g.

VI. EXPERIMENTAL SETUP

A. Simulation Environment and Network Configuration

The TOSSIM (V2.1.2) simulation environment was used in all experiments [27]. TOSSIM is a discrete event simulator capable of accurately modelling sensor nodes and the modes of communications between them.

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. The node neighbourhoods were generating using LinkLayerModel⁴ with the parameters in [10, Table I]. Noise models were created using the first 2500 lines of `casino-lab.txt`⁵. A single source node generated messages and a single sink node collected messages. These nodes were assigned positions in the SourceCorner configuration from [9], where the source is in the top left corner and the sink at the centre of the grid. The rate at which messages from the real source were generated was varied, as shown in Section VII. At least 10,000 repeats were performed for each combination of parameters. Nodes were located 4.5 meters apart. The node separation distance was determined experimentally, based on observing the pattern of transmissions in the simulator. 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) will usually only be passed to horizontally or vertically adjacent nodes.

B. Attacker Model

A reactive attacker model based on the patient adversary introduced in [3] is used. The attacker initially starts at the sink. When a message is received the attacker will move to the 1-hop source's location if that message has not been received before. To detect if a message has been received before, we

⁴LinkLayerModel is a tool provided with TOSSIM that extrapolates link strengths from node coordinates based on experimental data.

⁵casino-lab.txt is a noise sample file provided with TOSSIM.

TABLE II: Safety Period (seconds).

Network Size	Source Period (seconds/message)			
	2.0	1.0	0.5	0.25
11 × 11	33.24	16.65	8.56	4.51
15 × 15	49.59	24.82	12.59	6.54
21 × 21	74.52	37.40	18.84	9.66
25 × 25	91.17	45.73	23.03	11.76

assume that an attacker has access to the message type and sequence number. Once the source has been found the attacker will no longer move. This is commensurate with the attacker models used in [8, 9, 10].

C. Safety Period

A metric called *safety period* was introduced in [3] which captured the number of messages needed to capture the source. The higher the safety period is, the higher is the privacy level. This issue with this definition of safety period is that it is unbounded and in problem of asset monitoring it does not make sense to consider an unbounded safety period.

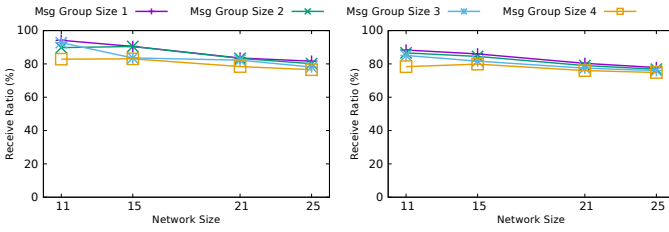
If the asset is immobile, then the attacker can simply perform an exhaustive search of the network and providing SLP is irrelevant. If the asset is mobile, then performing an exhaustive search is less desirable as an asset may move into a previously searched area of the network. So using the context provided by message broadcasts is preferred. This means that the SLP problem can only be considered when it is *time-bounded*, i.e., when the asset has to be captured within a certain time window. Therefore, we use an alternate by analogous definition of *safety period*, where if the attacker fails to capture the source within this period it is considered to have failed to capture the source.

As flooding has been shown to provide no SLP [3] we use the average time it takes the attacker to capture the source as a baseline and double it to calculate the safety period (shown in table II) for the protocol in this paper.

D. Simulation Experiments

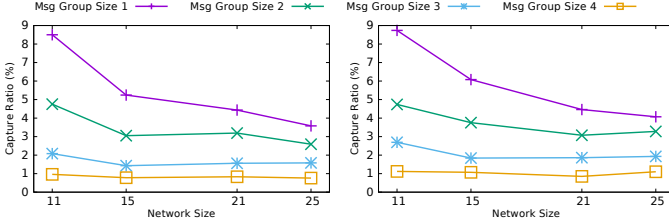
An experiment constituted a single execution of the simulation environment using a specified protocol configuration, network size and safety period. An experiment terminated when the source node had been captured or the safety period had expired. An attacker was implemented based on the log output from TOSSIM. It maintains internal state about its location using node identifiers. When a node receives a message, if the attacker is at that location it will move based on the attacker model.

The algorithm being tested has four parameters: the maximum walk length, the buffer size, the number of messages to group and the probability the message is sent directly to the sink. As the maximum walk length is simply to provide a finite bound in large networks, it was set to 100 hops. The number of messages to group was varied between $\{1, 2, 3, 4\}$. The buffer size was set to 10 messages as we do not expect more than 10 concurrent messages being sent in the network at one time. Finally, the probability of sending a message directly to the sink was varied between $\{0, 0.2, 0.4, 0.6\}$.



(a) Source Period 2.0 seconds (b) Source Period 0.25 seconds

Fig. 3: Results showing the receive ratio.



(a) Source Period 2.0 seconds (b) Source Period 0.25 seconds

Fig. 4: Results showing the capture ratio.

VII. RESULTS

As part of these results we will analyse four key metrics: (i) **receive ratio**: the percentage of messages sent by the source and received at the sink, (ii) **capture ratio**: the percentage of runs in which the source was captured, (iii) **messages sent per second**: the number of messages sent by all nodes in the network per second, and (iv) **latency**: the time it takes a message sent by the source to be received at the sink.

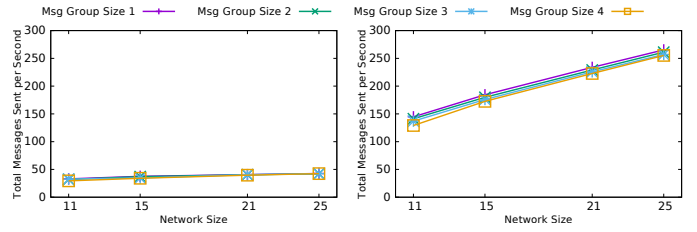
All results shown are for when the probability that a message is sent directly to the sink is 20%. Lower probabilities gave similar, but slightly worse results and higher probabilities produced much worse results. Therefore, for reasons of space, we have omitted results for the probabilities of 0%, 40% and 60% as 20% performed better. Also, only the results for the two source periods 2 and 0.25 seconds per message are included as the patterns between these are similar to the results for 1 and 0.5 seconds per message.

A. Receive Ratio

A high receive ratio between 75% and 95% is observed. Fewer messages were delivered with larger networks and larger messages group sizes. This suggests that the attacker was hearing most of the source messages, meaning that the privacy level imparted by the algorithm is due to the efficiency of the protocol and not due to the unreliability of the network.

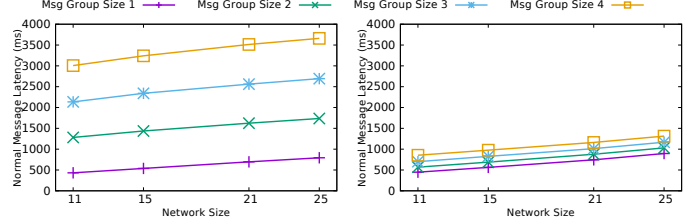
B. Capture Ratio

There is a less than 10% probability of the attacker capturing the source within the safety period for these parameter combinations. As more messages are grouped together, the capture ratio falls lower. This matches with our previous intuition that a greater number of messages grouped together would give the attacker fewer chances to respond to messages within the safety period. On the other hand, with messages



(a) Source Period 2.0 seconds (b) Source Period 0.25 seconds

Fig. 5: Results showing the messages sent per second.



(a) Source Period 2.0 seconds (b) Source Period 0.25 seconds

Fig. 6: Results showing the message latency.

being grouped together, it suggests that the message delivery latency may be increased. When the size of the message group is high, the protocol delivers near optimal privacy level.

As has been previously observed in [9, 10], we also see that larger networks tend to have lower capture ratios.

C. Messages Sent per Second

As the different network sizes being varied each has a different safety period, the number of messages sent has been normalised with respect to the simulation length to allow the results to be compared. The results show that the number of messages sent does not vary greatly for different numbers of messages to group together. For faster message rates, larger message group sizes require slightly fewer messages sent per second.

D. Latency

The longer the time between messages, the larger the latency between a source sending the message and the sink receiving the message. This is because messages are delayed to make sure that the number requested are about the sink-source distance from the source. When there are multiple messages the per hop delay needs to be longer. These results indicate that there is a trade off between latency and capture ratio, to obtain a better capture ratio using this technique a larger latency will be required. For many applications a latency of this magnitude will be acceptable. For example tracking the location of a slow moving panda will not be adversely affected. But in scenarios where very low latency is important, such as on a battlefield, this technique may be less suitable.

VIII. DISCUSSION

The objective function chosen aimed to ensure that the attacker ended up as far from the source nodes as possible. However, this is stricter than the actual requirement of SLP which is that the attacker must not capture (i.e., be co-located) with a source

within the safety period. Attempting to optimise this model for other objective functions such as minimising latency or the total number of messages sent might produce different output indicating other kinds of routing that would provide SLP.

IX. CONCLUSION

In this paper we have presented a novel formalisation of SLP-aware routing as an ILP constraint satisfaction problem. This model produced optimal routes computed based on global network knowledge, making these routes unamenable to deployment. We then presented a distributed routing protocol inspired by the optimal solution and performed large scale simulations to judge the performance. Our results show that we can obtain low capture ratios (high levels of SLP) with the tradeoff being a higher delivery latency. For future work we aim to expand the model to include fake message broadcasts and to also look at optimising for other objective functions such as the number of messages sent and the delivery latency.

REFERENCES

- [1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, ser. WSNA '02. New York, NY, USA: ACM, 2002, pp. 88–97.
- [2] A.-M. Badescu and L. Cotofana, "A wireless sensor network to monitor and protect tigers in the wild," *Ecological Indicators*, vol. 57, pp. 447–451, 2015.
- [3] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," in *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, Jun. 2005, pp. 599–608.
- [4] 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*, ser. SASN '04. New York, NY, USA: ACM, 2004, pp. 88–93.
- [5] M. Conti, J. Willemsen, and B. Crispo, "Providing source location privacy in wireless sensor networks: A survey," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1238–1280, 2013.
- [6] P. Spachos, D. Toumpakaris, and D. Hatzinakos, "Angle-based dynamic routing scheme for source location privacy in wireless sensor networks," in *Vehicular Technology Conference (VTC Spring), 2014 IEEE 79th*, May 2014, pp. 1–5.
- [7] L. Yao, L. Kang, F. Deng, J. Deng, and G. Wu, "Protecting source–location privacy based on multirings in wireless sensor networks," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 15, pp. 3863–3876, 2015.
- [8] 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, 2011.
- [9] A. Jhumka, M. Bradbury, and M. Leeke, "Fake source-based source location privacy in wireless sensor networks," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 12, pp. 2999–3020, 2015.
- [10] M. Bradbury, M. Leeke, and A. Jhumka, "A dynamic fake source algorithm for source location privacy in wireless sensor networks," in *14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Aug. 2015, pp. 531–538.
- [11] J. Long, M. Dong, K. Ota, and A. Liu, "Achieving source location privacy and network lifetime maximization through tree-based diversionary routing in wireless sensor networks," *IEEE Access*, vol. 2, pp. 633–651, 2014.
- [12] W. Tan, K. Xu, and D. Wang, "An anti-tracking source-location privacy protection protocol in WSNs based on path extension," *Internet of Things Journal, IEEE*, vol. 1, no. 5, pp. 461–471, Oct. 2014.
- [13] M. Dong, K. Ota, and A. Liu, "Preserving source-location privacy through redundant fog loop for wireless sensor networks," in *13th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC)*, Liverpool, UK, Oct. 2015, pp. 1835–1842.
- [14] M. Mahmoud and X. Shen, "A cloud-based scheme for protecting source-location privacy against hotspot-locating attack in wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 10, pp. 1805–1818, Oct. 2012.
- [15] M. Shao, W. Hu, S. Zhu, G. Cao, S. Krishnamurth, and T. La Porta, "Cross-layer enhanced source location privacy in sensor networks," in *6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '09)*, 2009, pp. 1–9.
- [16] S. Oh and M. Gruteser, "Multi-node coordinated jamming for location privacy protection," in *Military Communications Conference (MILCOM), 2011 IEEE*, Nov. 2011, pp. 1243–1249.
- [17] M. Raj, N. Li, D. Liu, M. Wright, and S. K. Das, "Using data mules to preserve source location privacy in wireless sensor networks," *Pervasive and Mobile Computing*, vol. 11, pp. 244–260, 2014.
- [18] J. P. Singh, P. K. Roy, S. K. Singh, and P. Kumar, "Source location privacy using data mules in wireless sensor networks," in *2016 IEEE Region 10 Conference (TENCON)*, Nov. 2016, pp. 2743–2747.
- [19] Z. Benenson, P. M. Cholewinski, and F. C. Freiling, *Wireless Sensors Networks Security*. IOS Press, 2008, ch. Vulnerabilities and Attacks in Wireless Sensor Networks, pp. 22–43.
- [20] K. Mehta, D. Liu, and M. Wright, "Protecting location privacy in sensor networks against a global eavesdropper," *IEEE Trans. on Mobile Computing*, vol. 11, no. 2, pp. 320–336, Feb. 2012.
- [21] Y. Yang, M. Shao, S. Zhu, and G. Cao, "Towards statistically strong source anonymity for sensor networks," *ACM Trans. Sen. Netw.*, vol. 9, no. 3, pp. 34:1–34:23, Jun. 2013.
- [22] A. Proaño, L. Lazos, and M. Krunz, "Traffic decorrelation techniques for countering a global eavesdropper in WSNs," *IEEE Trans. on Mobile Computing*, vol. 16, no. 3, pp. 857–871, Mar. 2017.
- [23] A. Boubrima, W. Bechkit, and H. Rivano, "Optimal WSN deployment models for air pollution monitoring," *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, pp. 1–1, 2017.
- [24] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609–619, Aug. 2004.
- [25] R. Asorey-Cacheda, A.-J. Garcia-Sanchez, F. Garcia-Sanchez, and J. Garcia-Haro, "A survey on non-linear optimization problems in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 82, pp. 1–20, 2017.
- [26] C. Gu, M. Bradbury, and A. Jhumka, "Phantom walkabouts in wireless sensor networks," in *Proceedings of the 2017 ACM Symposium on Applied Computing*, ser. SAC'17. New York, NY, USA: ACM, Apr. 2017, pp. 609–617.
- [27] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: accurate and scalable simulation of entire tinyos applications," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, ser. SenSys '03. New York, NY, USA: ACM, 2003, pp. 126–137.