Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

Phantom Walkabouts: A Customisable Source Location Privacy Aware Routing Protocol for Wireless Sensor Networks

Chen Gu*[1] | Matthew Bradbury[1] | Arshad Jhumka[2]

[1]WMG, University of Warwick, Coventry, UK
[2]Department of Computer Science, University of Warwick, Coventry, UK

Correspondence
* Chen Gu. International Digital Laboratory, WMG, University of Warwick, Coventry, CV4 7AL, UK. Email: Chen.Gu.1@warwick.ac.uk

Summary

Source location privacy (SLP) is an important property for a large class of security-critical wireless sensor network (WSN) applications such as monitoring and tracking. In the seminal work on SLP, phantom routing was proposed as a viable approach to address SLP. However, recent work has shown some limitations of phantom routing such as poor data yield and low SLP. In this paper, we propose *phantom walkabouts*, a novel and more general version of phantom routing, which performs phantom routes of variable lengths. Through extensive simulations we show that phantom walkabouts provides high SLP level than phantom routing under specific network configuration.

KEYWORDS:
Wireless Sensor Networks, Source Location Privacy, Phantom Walkabouts, Phantom Routing, Biased Random Walk

1 | INTRODUCTION

A wireless sensor network (WSN) consists of a number of tiny devices, known as sensor nodes, that can sense different attributes of the environment and use radio signals to communicate among themselves. WSNs have enabled the development of many novel applications, including asset monitoring[1], target tracking[2] and environment control[3] among others, with low levels of intrusiveness. They are also expected to be deployed in safety and security-critical systems, including military[4] and medical services[5]. The communication protocols used in the WSNs must therefore meet a set of stringent security and privacy requirements, dependent on the application.

Threats to privacy in monitoring applications can be considered along two dimensions: (i) content-based threats, and (ii) context-based threats. Content-based privacy threats relate to use of the content of the messages broadcast by sensor nodes, such as an attacker gaining the ability to read an eavesdropped encrypted message. There has been much research addressing the issue of providing content privacy, e.g., SPINS[6], with most efforts in this area focusing on the use of cryptographic techniques. On the other hand, context-based privacy threats focus on the context in which messages are broadcast and how information can be observed or inferred by attackers. Context is a multi-attribute concept that encompasses situational aspects of broadcast messages, including environmental and temporal information.

It is often desirable for the source of sensed information to be kept private in a WSN. For example, in a military application, a soldier transmitting messages may unintentionally disclose their location, even when encryption is used. Another example is during the monitoring of endangered species where poachers may be tempted to infer the location of the animal to capture it. Real world examples include monitoring badgers[2] and the WWF's Wildlife Crime Technology Report[7], both of which would likely benefit from a context-based security measure. In this paper, we focus on protecting the *source location*.

Techniques that protect this source location are said to provide source location privacy (SLP). SLP is important in many application domains, though it is of utmost concern in security-critical situations. In each of these scenarios, it is important to ensure that an attacker is difficult to find or deduce the location of the asset being monitored, whether it is a soldier or an endangered animal. A WSN designed to forward the information collected about an asset would typically consist of the following: a dedicated node for data collection called a *sink node*, the node(s) involved in sending information about these assets called *source nodes*, and many other nodes in the network used to route/relay messages over multiple hops from the sources to the sink. It has been shown that in a non-SLP protected network, even a weak attacker such as a distributed eavesdropping attacker can backtrack along message paths through the network to find the source node and capture the asset[8]. Thus, there is a need to develop SLP-aware routing algorithms.

In the seminal work on SLP, the phantom routing technique was proposed[8]. Phantom routing is a technique where a source initially sends a message along a random walk (a.k.a. phantom route) of a certain length (typically a few hops). When the message reaches a phantom node at the end of the walk, the phantom node routes the message towards the sink by flooding. Though phantom routing should work well in theory, in practice it does not as a link failure may cause the directed random walk to fail, resulting in a low data yield. Besides, it cannot provide state-of art SLP due to the short random walk. In this paper, we propose a novel, more generalised technique called *phantom walkabouts*, of which phantom routing is a specific instance. Through extensive simulations, we show that phantom walkabouts provides state-of-the-art levels of SLP. The main contributions of this paper are:

- We establish new random walk algorithms which address defects of random walk in phantom routing.
- We propose phantom walkabouts, a novel and more general technique than phantom routing, that help achieve high SLP level.
- We show, via extensive simulations, the viability of phantom walkabouts. For example, under certain parameterisation, phantom walkabouts achieves extremely high SLP.

The remainder of this paper is organised as follows: Section 2 surveys related work in SLP and Section 3 presents the models assumed. In Section 4 we present phantom walkabouts. The adopted system and simulation approach are outlined in Section 5. Section 6 presents the results of the experiments conducted. We provide some discussions about our approach in Section 7. Section 8 concludes this paper with a summary of contributions.

2 | RELATED WORK

The concept of the SLP problem was first posed around 2004 with the proposal of the panda-hunter game, where the poachers only used network traffic flow to track the panda[9]. [8] formalised the SLP issue based on the panda-hunter game[8]. Since then, several techniques have been proposed to address SLP. The solution spectrum spans from simple solutions such as simple random walk[9] to more sophisticated techniques such as fake sources and diversionary routing[10,11,12].

2.1 | Random-Walk Based Techniques

In the seminal work[8], the authors proposed a solution called *phantom routing*, where messages were sent on a directed random walk in which the message was either sent towards or away from a certain node in the network, followed by using the flooding routing protocol. Then a similar approach to phantom routing called *single-path phantom routing* was also proposed in[8]. Instead of using flooding, the authors used single path routing protocols, such as shortest path routing. The combination of the random walk together with such single path routing is often referred to as the phantom single-path routing scheme (PSRS). Phantom routing and PSRS have received a lot of attention in the literature. On the other hand, this class of solution is known to have weaknesses as demonstrated by[13,14,15], ascribing poor SLP performance to the directed random walk reusing the routing path leading to exposure of direction information.

For other random walk algorithms, a new algorithm using location angles was proposed to construct the random walk based on the inclination angle between a node and its neighbour towards the sink[16]. [17] introduced the greedy random walk (GROW). In GROW, one random walk starts from the sink and goes to a randomly chosen receptor-node. The other random walk starts from the source and meets the first random walk at the receptor-node. Then, the receptor-node uses the path established by the random walk from the sink to the receptor-node to route the packet from the source to the sink. However, there is still scope to improve the nodes that are allocated to take part in the directed random walk. Phantom walkabouts was an algorithm also using a random walk technique to provide SLP[18]. The technique used a mix of short and long random walks to achieve a higher level of SLP than phantom routing with a bounded message overhead. Other algorithms use the random walk technique to address the SLP issue such as forward random walk (FRW)[19], trace cost based SLP protection scheme (TCSLP) for smart cities[20] and random routing scheme (RRS)[21].

2.2 | Fake-Source Based Techniques

Algorithms have also been developed that utilise dummy messages sent by a *fake source* to provide SLP. Some nodes are chosen as fake sources and periodically send dummy messages to obfuscate the real traffic. The authors introduced a concept of fake sources and proposed a theoretical algorithm called short-lived fake source routing (SLFSR)[22]. Numerous algorithms have been also proposed with state-of-the-art fake message techniques[23,11,24]. These algorithms based on the fake source technique mentioned so far can only provide SLP against the local attacker who has a local view of the network.

For the scope of the global attacker who has a full view of the network, a global protection scheme called Periodic was developed in which every node sends a message after a fixed period[25]. This provided perfect protection against an attacker with a global view of the network. The authors created a model involving traces of source detection, which was used to measure the privacy of those traces, as well as the energy cost of providing SLP. In addition, a different approach has been developed where statistical techniques were used to show that their global protection scheme provided high levels of SLP[10]. This approach does not provide perfect global SLP as[25] does, but instead provided statistically strong SLP. Their model and solution aims to make the distribution of message broadcasts from nodes indistinguishable from a certain statistical distribution.

Perhaps the most significant disadvantage of the described fake source techniques is the volume of messages broadcast to provide SLP. This leads to increased energy consumption and an increased number of collisions, both of which result in a decreased packet delivery ratio. This means that a tradeoff between energy expenditure and privacy must be made[26], making dummy message schemes challenging for many large-scale networks.

2.3 | Other Techniques

In addition to the techniques described above, authors in[27] proposed ILP routing in which messages were delayed by different amounts, such that they reached a similar point at a certain distance. By doing this the attacker makes less progress, due to messages being grouped at a similar location. Another algorithm was proposed where nodes changed the chronological order of received messages and sent messages which also change the traffic pattern, making it hard for a local adversary to track the traffic to the source node[28]. Mules-saving-source protocol (MussP) use α-angle anonymity to provide SLP by adopting data mules which collect the packets from sources and drop them elsewhere[29].[30] proposed a source location protection protocol based on dynamic routing aims at maximising paths for data transmission to address the SLP problem. Others include using geographic routing[14] and network coding[31] to address the SLP problem. In short, these techniques cause either message complexity or high delivery latency in the network.

3 | MODELS

In this section, we present the various models that underpin this work.

3.1 | SLP Problem Model

The SLP problem model was based on the Panda-Hunter Game proposed by[9] and first formalised by[8]. The aim is for the attacker to find the location of the sources by tracing back the messages sent by the routing protocol $\mathcal{R}$. The aim of network maintainers is to modify or replace $\mathcal{R}$ such that the attacker fails to capture the source. The model is represented by the six-tuple $(G, Sink, Src, \mathcal{R}, \mathcal{A}, \mathcal{M}_{\mathcal{A}})$, where:

- $G = (V, E)$ defines the network graph where V represents the set of sensor nodes, and E is a set of communication links connecting two distinct nodes.
- $Sink \in V$ is the network sink node, to which all communication in the sensor network must ultimately be routed to.

- $Src \in V$ is the source that the sensor network monitors.
- $\mathcal{R}$ is the routing protocol employed by the sensors to protect the source Src from being acquired or tracked by the attacker $\mathcal{A}$.
- $\mathcal{A}$ is the attacker, or hunter, who seeks to acquire or capture the source Src through a set of movement rules $\mathcal{M}_{\mathcal{A}}$.

The following sections will expand on this representation and explain aspects of this model further. Subsection 3.2 will detail the network model including G, $Sink$ and Src. The attacker model including $\mathcal{A}$ and $\mathcal{M}_{\mathcal{A}}$ will be described in Subsection 3.3. The new routing protocol $\mathcal{R}$ to protect SLP will be introduced in Section 4.

3.2 | Network Model

A wireless sensor node is a device with a unique identifier that has limited computational capabilities and is equipped with a radio transmitter for communication. A WSN is a set of wireless sensor nodes with communication links between pairs of nodes. The sensor network is modelled as a graph $G = (V, E)$ where V represents the set of nodes and E is a set of unordered pairs that represent bidirectional links between the nodes. The nodes that are in direct communication range with a node *n* are called the neighbours of *n*. We assume all the nodes to be stationary, i.e., the topology of the network remains constant as well as the neighbourhoods of all the nodes over the lifetime of the network.

There exists a distinguished node in the network called a *sink*, which is responsible for collecting data and which acts as a link between the WSN and the external world. Other nodes sense data and then route the data via messages along a computed route to the sink for collection. It is expected that there are multiple hops between the source and sink. Any node, except for the sink, can be a data source. It is assumed that the network is event-triggered, i.e., when a node senses an object, it starts sending messages periodically to the sink for a certain amount of time.

The messages sent are encrypted and the source node includes its identifier in the encrypted messages. The type of encryption, be it end-to-end, pairwise or some other scheme is left undefined. Using the identifier, the sink can infer an asset's location as it is assumed that the network administrators will record where they put nodes. Nodes are not assumed to know their geographical location by being equipped with Global Positioning System (GPS) chips, which is due to the increase in energy cost that would be incurred.

3.3 | Attacker Model

We assume a patient adversary model, known as a *distributed eavesdropper*, introduced in[8]. The attacker initially starts at the sink and we assume the attacker is equipped with the necessary devices (such as directional antennas) to determine the direction a message originated. When the attacker overhears a new message, it will move to the location of the immediate sender, i.e., the neighbour that last forwarded the message. Once the source has been found, the attacker will no longer move. However, the attacker does not read the messages it overhears, so cannot obtain the contents of a message.

There are two more things that need to explain: First, in this model, the attacker starts at the sink since the sink is the one location in the network where the attacker is guaranteed to eavesdrop any message from the source, irrespective of the routing protocol used. The attacker could potentially start at any location in the network, but may not receive messages due to the location not being on the route from the source to the sink. Second, as we focus on a distributed attacker with a small visibility of the network, the techniques are not designed to protect against a global attacker who has a global view of the network. The reason for this is that for attackers to gain global visibility they will need to expend significant resources. For example, they need build towers with sensitive long range directional antennas or deploy many attackers presenting in the network each with a small visible range. Now we sum up the attacker model below for clarity:

- The attacker is a person physically present in the network.
- The attacker starts at the sink as that is the one location guaranteed to receive messages from the source.
- The attacker has sufficient directional antennas to detect the direction from which a message originates.
- The attacker moves to the location of the proximate node when it eavesdrops a new message.
- The attacker have a local view of the network.

- The attacker does not jam the network (as that would reveal its position).
- The attacker does not do complicated traffic analysis or physically attack the network.
- The attacker does not have knowledge of the routing protocol in the network.

3.4 | Privacy Model

The overall objective of any WSN-based SLP solution is to ensure that the source (at a given location) is never captured. As such, a notion of time boundedness termed as *safety period* in the literature[8] which is the number of messages sent that an attacker needs to capture the source has been developed. The higher the safety period is, the higher the source location privacy level. However, using the safety period metric means that simulation runtime is unbounded and potentially very large.

We use an alternative, but analogous, definition for safety period for each network size and network configuration, and obtain the safety period when protectionless flooding is used as the routing protocol[8]. Flooding is used as it has been argued to provide the *lowest* SLP level, hence any SLP improvement is due to the SLP-aware technique[8]. The safety period is then obtained by increasing this value to account for the attacker potentially making bad moves.

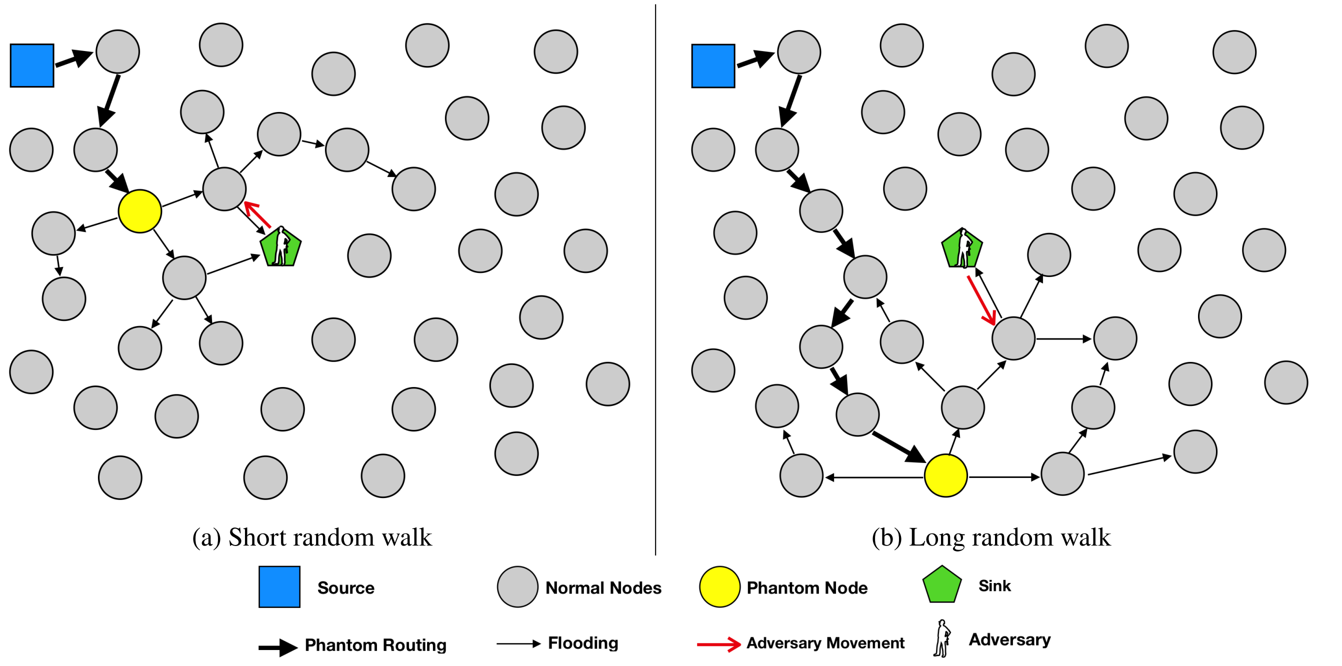
4 | PHANTOM WALKABOUTS

In the section, we propose a novel SLP routing protocol, termed as *phantom walkabouts*, which is a more generic version of *phantom routing* strategy that addresses the impact caused by small random walks in phantom routing [†]. Phantom walkabouts generalises phantom routing with variable random walk lengths, which we will show to provide better performance than phantom routing. We explain the rationale behind the protocol and algorithms for forming the new random walk, the biased random walk and the overall phantom walkabouts algorithm. Table 1 summarises the notation used in this paper.

TABLE 1 Commonly used notations

Notation	Description
msg	The normal message
$\mathcal{S}_{dir}$	The random walk set of a message
$\mathcal{M}_{dir}$	The random walk direction of a message
$\mathcal{B}_{dir}$	The biased random walk direction of a message
$\mathcal{P}_{biased}$	The probability of biased random walk
$\mathcal{T}_{flooding}$	The time taken (seconds) of protectionless flooding
P_{safety}	The safety period (seconds)
M_s	The message with the short random walk
M_l	The message with the long random walk
Δ_{ss}	The distance in hops between the sink and the source
h_{walk}	The remaining hops of the random walk

[†] https://bitbucket.org/Chen_Gu/slp-algorithms-tinyos


(a) Short random walk

(b) Long random walk

FIGURE 1 Short and long random walk routing examples

4.1 | Motivation of Phantom Walkabouts

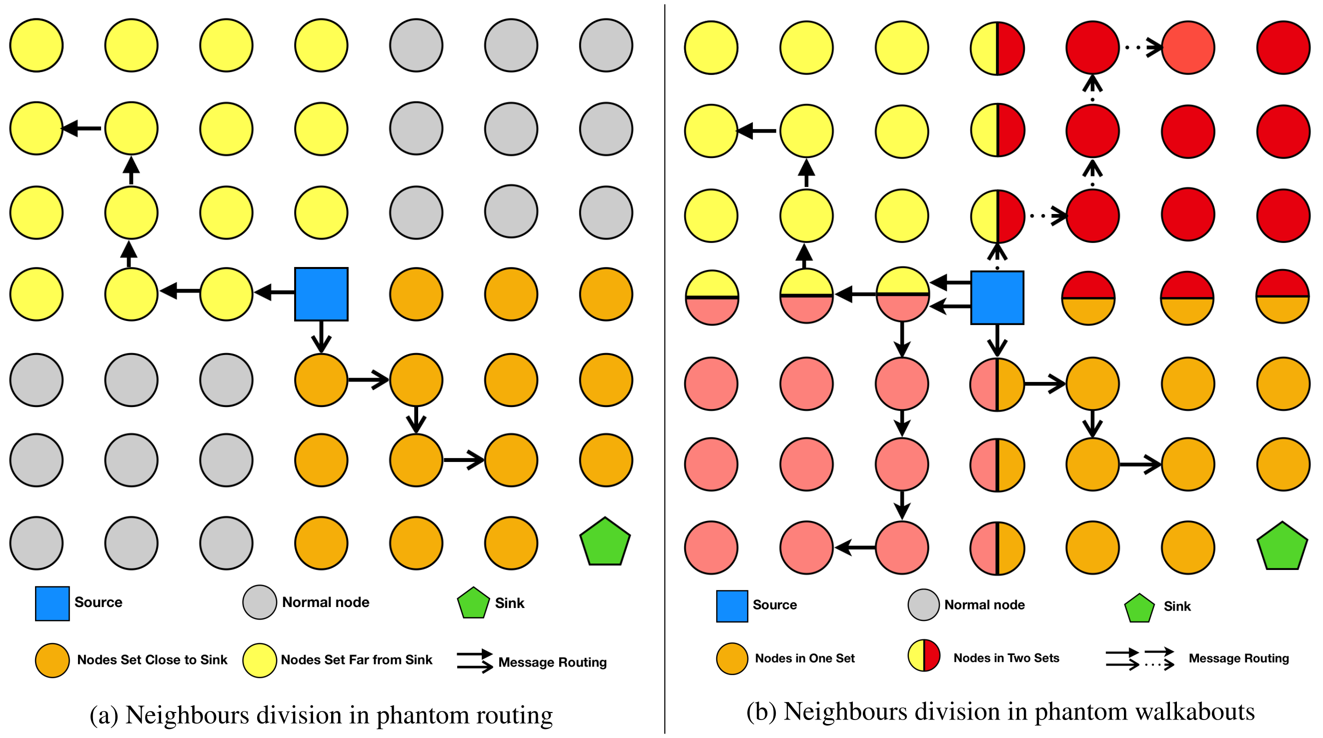
Phantom routing is a routing protocol to provide SLP, and it works by altering a flooding routing protocol to consist of an initial directed random walk followed by flooding. In the random walk phase data are transmitted between two neighbouring nodes through single-hop communication. Therefore, the random walk length is usually very short in the phantom routing since long random walk could cause the directed random walk to fail, resulting in a low data yield.

Figure 1 shows the typical scenario during an execution of phantom routing where the source sends a message to a phantom node which lies somewhere between itself and the sink. When the phantom node floods the message to the sink, the first movement of the attacker is towards the phantom node (see Figure 1a). However, it would be beneficial to have the first movements of the attacker to be away from the source, as shown in Figure 1b. To achieve this, a longer random walk can be used, where the length of the walk exceeds the sink-source distance.

As such, we conjecture that phantom walkabouts with a mix of short and long random walk will achieve a higher level of SLP than phantom routing. We denote a phantom walkabouts parametrisation by $PW(m, n)$, where m, n denote the number of short and long random walk respectively to be performed in a cycle. $PW(1, 1)$ denotes a repeating sequence of 1 short random walk followed by 1 long random walk. Later, in this paper, we investigate the SLP levels and associated receive ratio of $PW(1, 0)$, $PW(1, 1)$, $PW(1, 2)$, and $PW(0, 1)$.

4.2 | New Random Walk Algorithm in Phantom Walkabouts

In phantom routing, each node maintains two sets for all its neighbours: (i) *CloserSinkSet* which contains all the neighbours whose hop counts to the sink are smaller than or equal to the node's hop count to the sink, and (ii) *FurtherSinkSet* which includes neighbours with a larger hop count to the sink. After neighbour nodes are partitioned, the source randomly picks one of these two sets and sends normal messages to one neighbour in the chosen set. During the random walk phase, messages are always sent to the neighbour in the chosen set. If a message is blocked (e.g., there is no neighbour in the chosen set so messages cannot be forwarded) the random walk phase stops. In other cases, when a message travelled s hops (assuming random walk length is s), it has finished the random walk phase. When the random walk phase ends, if a message does not reach the sink node, the message then floods the network so it reaches the sink node.



(a) Neighbours division in phantom routing

(b) Neighbours division in phantom walkabouts

FIGURE 2 Comparisons of neighbour divisions in phantom routing and phantom walkabouts

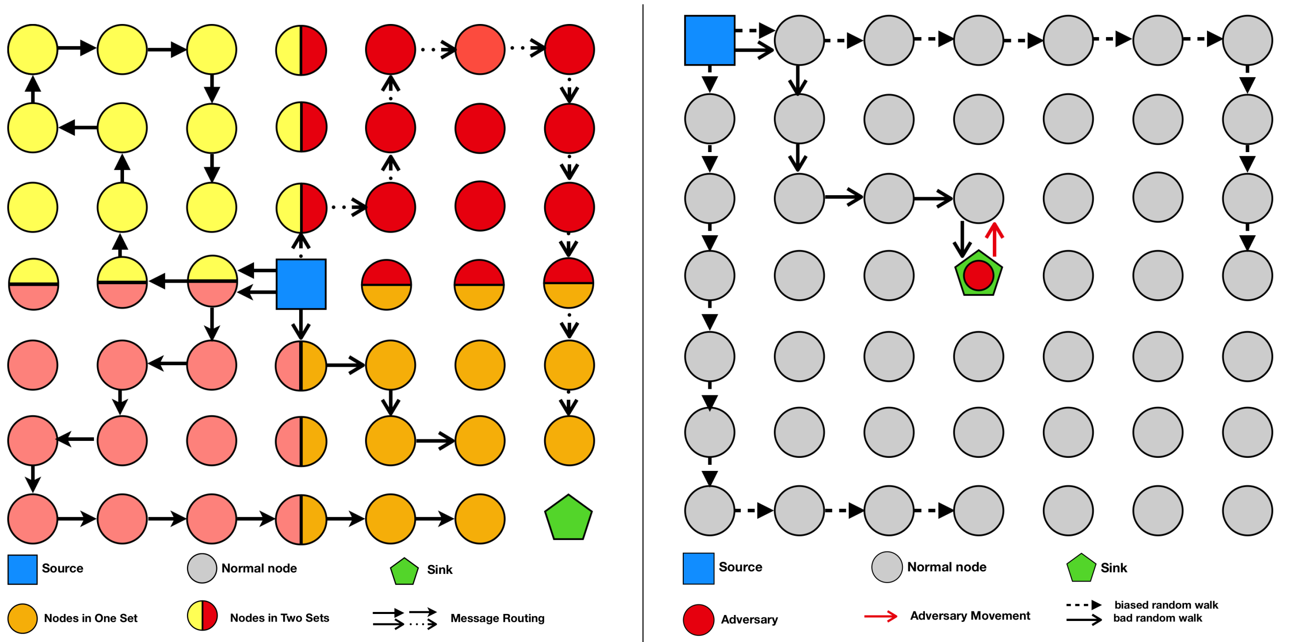
However, in the random walk phase of phantom routing some exceptional situations have never been considered. For example, not every node could become a phantom node; random walk exceptionally terminates at some nodes. Therefore, we describe a new random walk algorithm to deal with neighbour nodes division and exceptional termination of random walks.

- **Neighbour division in random walk phase**
  Different from phantom routing where each node's neighbours are divided into two sets, each node's neighbours are now divided into four sets in different directions[‡]. This division can be done as follows: We choose a node in the network as a landmark node. As shown in phantom routing a landmark node divides a node's neighbours into two sets by flooding *beacon* messages, the chosen landmark node in phantom walkabouts also floods beacon messages to divide a node's neighbours into other two sets, thus achieving the neighbour division into four sets. In other words, in total two waves of beacon messages are flooded by the two selected landmark nodes. Figure 2 demonstrates the difference of neighbours division. Given a grid network configuration, in the random walk phase of phantom routing half of the nodes in the network could be used as phantom nodes (see Figure 2a) whereas almost all the nodes could become phantom nodes in the new random walk routing (see Figure 2b).

- **Random walk termination handling in random walk phase**
  If a message is blocked during the random walk phase (e.g., messages reach the border of the network), a new message direction will be chosen and assigned to the $CloserSinkSet$. This procedure is demonstrated in Figure 3a. In certain extreme situations when $CloserSinkSet$ is void, the random walk terminates and the node becomes a phantom node. Because we believe the phantom node is farthest from the real source node and ensures the safety of the source node that its location will be hard to track. When a message travels l hops (assuming random walk length is l), it has finished the random walk phase.

Similar to phantom routing, if a message does not reach the sink after the random walk phase, the flooding phase will start once the random walk ends. The algorithms of these two phases are shown in Algorithm 1 and Algorithm 2.

[‡] Observe that this does not restrict the network configuration to be a grid, but the nodes can be partitioned into these four sets.



(a) Illustration of random walk routing in phantom walkabouts

(b) Illustration of bad random walks and biased random walks in SourceCorner configuration

FIGURE 3 Random walk in phantom walkabouts

Algorithm 1 Random Walk Phase in Phantom Walkabouts

```
procedure RANDOM WALK PHASE($msg$, $l$)
    $msg.\mathcal{S}_{dir} \leftarrow \perp$
    $msg.\mathcal{M}_{dir} \leftarrow \perp$
    $msg.h_{walk} \leftarrow l$
    $msg.\mathcal{S}_{dir} \leftarrow$ CHOOSEONESET($msg$)
    while $msg.h_{walk} \neq 0$ do
        $msg.\mathcal{M}_{dir} \leftarrow$ CHOOSEONENEIGHBOUR($msg.\mathcal{S}_{dir}$)
        if ISREACHSINK($msg$) $= True$ then
            $msg.h_{walk} \leftarrow 0$
            break
        end if
        if $msg.\mathcal{M}_{dir} = \perp$ then
            $msg.\mathcal{M}_{dir} \leftarrow$ CHOOSEONENEIGHBOUR($CloserSinkSet$)
        end if
        $msg.h_{walk} \leftarrow msg.h_{walk} - 1$
        FORWARDMESSAGE($msg.\mathcal{M}_{dir}$)
    end while
end procedure
```

4.3 | Biased Random Walk Algorithm in Phantom Walkabouts

The new random walk routing algorithm proposed in Subsection 4.2, especially a long random walk ensures that phantom nodes are far away from the real source node. However, there is a weakness that needs to be addressed for certain topologies. Specifically, consider the configuration where a source node lies in the corner of a grid and the sink node in the middle area of the network. As the source is located in the corner, messages will always be transmitted towards the sink node. Owing to the random nature

Algorithm 2 Flooding Phase in Phantom Walkabouts

```
procedure FLOODING PHASE($msg$)
    if $msg.h_{walk} = 0$ then
        if ISREACHSINK($msg$) $= False$ then
            FLOODING($msg$)
        end if
    end if
end procedure
```

of the walk, the random walk may take path through the sink node. In this case, the attacker will notice the message and will move towards the source node, increasing the chance of a source capture. As shown in Figure 3b, the bad random walk is always forwarding messages closer to the sink, so the attacker can always move backwards to the source location.

To address this issue, we developed a *biased random walk* based on our new random walk algorithm proposed in Subsection 4.2, for the specific configuration so as to avoid the risk of a random walk close to the attacker. In the biased random walk, messages are not forwarded in the direction of the sink. Instead, messages are transmitted by following border nodes to avoid being captured in the random walk phase. Biased random routing is described as follows and shown in Algorithm 3 and Algorithm 2.

- The source first chooses a set out of four neighbour sets, and then assigns a direction from the chosen set for a message. The chosen direction is called the biased direction ($\mathcal{B}_{dir}$). The message direction $\mathcal{M}_{dir}$ is always following the $\mathcal{B}_{dir}$.
- When a node receives a message, the random value $r \in [0, 1]$ is generated. The fixed parameter $\mathcal{P}_{biased}$ is set in the experiments to make sure a message has a high probability of walking along the previous biased direction. Normally the value of $\mathcal{P}_{biased}$ is set larger than 0.5 but less than 1. For instance, if $\mathcal{P}_{biased}$ is set to 0.7, it indicates the message has a 70% probability of being transmitted along the previous biased direction. The node decides the message direction $\mathcal{M}_{dir}$ by the following equation:

$$\mathcal{M}_{dir}(r, \mathcal{P}_{biased}, \mathcal{S}_{dir}, \mathcal{B}_{dir}) = \begin{cases} \mathcal{B}_{dir} & \text{if } r \in [0, \mathcal{P}_{biased}] \\ \mathcal{S}_{dir} \setminus \{\mathcal{B}_{dir}\} & \text{otherwise.} \end{cases} \tag{1}$$

- When the message direction is blocked, it indicates that the message reaches the end of this direction. The message will choose a new biased direction to continue the random walk until random walk finishes. If the new biased random walk direction is empty again, the random walk phase stops. Then the flooding phase starts.

4.4 | Phantom Walkabouts

In this section, we formalise the phantom walkabouts technique, which extends the phantom routing protocol by adopting variable lengths of phantom routing. When a source node routes a message M using phantom walkabouts, a decision is needed regarding whether M goes on a short (M_s) or long (M_l) random walk route. The sequencing of messages looks like as follows:

$$\underbrace{M_s, \cdots, M_s}_{m}, \underbrace{M_l, \cdots, M_l}_{n}, \underbrace{M_s, \cdots, M_s}_{m}, \underbrace{M_l, \cdots, M_l}_{n}, \cdots$$

Therefore, we observe that the phantom walkabouts $PW(m, n)$ consists of m messages on short random walk and n messages on long random walk, before the cycle is repeated. The phantom walkabouts adopts all the techniques described in Subsection 4.2, Subsection 4.3 and Subsection 4.4. As shown in Equation 2, when a message is M_s, the new random walk algorithm is adopted.

Algorithm 3 Biased Random Walk in Phantom Walkabouts

```
procedure BIASED RANDOM WALK($msg$, $l$, $\mathcal{P}_{biased}$)
    $msg.\mathcal{S}_{dir} \leftarrow \perp$
    $msg.\mathcal{M}_{dir} \leftarrow \perp$
    $msg.\mathcal{B}_{dir} \leftarrow \perp$
    $msg.h_{walk} \leftarrow l$
    $msg.\mathcal{S}_{dir} \leftarrow$ CHOOSEONESET($msg$)
    while $msg.h_{walk} \neq 0$ do
        $msg.\mathcal{M}_{dir} \leftarrow$ CHOOSEONENEIGHBOUR($msg.\mathcal{S}_{dir}$)
        if $msg.\mathcal{B}_{dir} = \perp$ then
            $msg.\mathcal{B}_{dir} \leftarrow msg.\mathcal{M}_{dir}$
        end if
        $r \leftarrow$ GENERATERANDOMNUMBER(0, 1)
        if ISREACHSINK($msg$) $= True$ then
            $msg.h_{walk} \leftarrow 0$
            break
        end if
        if $r \geq \mathcal{P}_{biased}$ then
            $msg.\mathcal{M}_{dir} \leftarrow$ CHOOSEONENEIGHBOUR($msg.\mathcal{S}_{dir} \setminus msg.\mathcal{B}_{dir}$)
        end if
        if $msg.\mathcal{M}_{dir} = \perp$ then
            $msg.\mathcal{M}_{dir} \leftarrow$ CHOOSEONENEIGHBOUR($msg.\mathcal{S}_{dir}$)
            $msg.\mathcal{B}_{dir} \leftarrow msg.\mathcal{M}_{dir}$
        end if
        $msg.h_{walk} \leftarrow msg.h_{walk} - 1$
        FORWARDMESSAGE($msg.\mathcal{M}_{dir}$)
    end while
end procedure
```

On the other hand, if a message to be M_l, the biased random walk algorithm is only used when the sink in the centre area of the network. Otherwise, the new random walk algorithm is used. Finally, the phantom walkabouts algorithm is shown in Algorithm 4.

$$\text{Phantom Walkabouts}\begin{cases}\text{Short random walk message } (M_s) \longrightarrow \text{New random walk algorithm} \\ \text{Long random walk message } (M_l)\begin{cases}\text{Sink in the centre area} \longrightarrow \text{Biased random walk algorithm} \\ \text{Otherwise} \longrightarrow \text{New random walk algorithm}\end{cases}\end{cases} \tag{2}$$

With long random walks in phantom walkabouts, the receive ratio is low due to the unreliability of network links[32], causing a proportion of messages to never reach the sink. Therefore, there is a need to add a mechanism to provide high message delivery. Retransmission will be used to ensure reliability along the route. It works as follows: When a message is transmitted from node j to a neighbour node k, k may send an acknowledgement (ACK) message when it has received the message from j. If j does not receive any ACK message from k, it means that the message may not have been successfully delivered to k. In this case, j will resend a same message to k. Retransmission will stop when an ACK message is received or the maximum number of retransmissions have been sent.

Algorithm 4 Phantom Walkabouts

```
procedure PHANTOM WALKABOUTS($m$, $n$)
    $m', n' \leftarrow m, n$
    while $True$ do
        if $m' > 0$ then
            $msg \leftarrow$ GENERATESHORTMESSAGE()    ▷ Source generates a message containing short random walk length
            ROUTING($msg$)    ▷ The routing algorithm is based on the Equation 2
            $m' \leftarrow m' - 1$
        else if $m' = 0 \wedge n' > 0$ then
            $msg \leftarrow$ GENERATELONGMESSAGE()    ▷ Source generates a message containing long random walk length
            ROUTING($msg$)
            $n' \leftarrow n' - 1$
        else
            $m', n' \leftarrow m, n$
        end if
    end while
end procedure
```

4.5 | Summary: Difference between Phantom Routing and Phantom Walkabouts

As pointing out some shortcomings of phantom routing, we introduce phantom walkabouts which addresses a number of weaknesses in phantom routing. This section briefly summarises the difference between the phantom routing and phantom walkabouts shown in Table 2.

4.6 | Problem Statement

In a WSN, phantom walkabouts is used as a routing protocol to deliver messages from the source(s) to the sink. When an attacker is initially located at the sink and starts receiving messages sent by the source(s) to the sink, an important problem is to analyse the impact on SLP of phantom walkabouts under various parameterisations. Formally, the problem specification is shown in Figure 4.

Given:

- A WSN topology $G = (V, E)$ where V is a set of wireless sensor nodes and E is a set of edges or links,
- A phantom walkabouts protocol $PW(m, n)$ where a pair (m, n) for short and long random walk lengths,
- A distributed eavesdropper attacker $\mathcal{A}$ that is located at the sink initially,
- A network configuration $\mathbb{C}$ where a source locates in the corner and a sink is in the centre, and
- A safety period δ,

Objective:

- Evaluate the performance of $PW(m, n)$ with various parameterisations of (m, n) over δ in G, $\mathbb{C}$ and $\mathcal{A}$.

FIGURE 4 Problem statement: Evaluation of phantom walkabouts with various parameterisations

Difference	Phantom Routing	Phantom Walkabouts
The length of random walk	The length of random walk is fixed to a few hops	1. The messages contain long random walk which exceeds sink-source distance 2. The short and long random walks repeat in the phantom walkabouts
The neighbour sets of a node	The neighbours of a node are classified into two sets: the *CloserSinkSet* and the *FurtherSinkSet*	More than two neighbour sets of a node are classified depending on the choice of the landmark nodes in the network (e.g., four sets in this paper)
The number of phantom nodes	Half the nodes in the network can be chosen as the phantom nodes (except for the source and the sink)	All the nodes in the network can be chosen as the phantom nodes (except for the source and the sink)
Random walk termination	Random walk stops in some exceptional cases (e.g., a message reaches the border node and cannot be forwarded)	Random walk continues when facing exceptional cases
Random walk techniques in different network configurations	The fixed random walk technique is used for all network configurations	Biased random walk technique is used for special network configuration (e.g., *SourceCorner* configuration)
Acknowledgement messages	No	Yes

TABLE 2 The Differences between phantom routing and phantom walkabouts

5 | EXPERIMENTAL SETUP

In this section we describe the simulation environment, source selection, network configuration and safety period calculation that were used to generate the results presented in Section 6.

5.1 | Simulation Setup

The TOSSIM (V2.1.2) simulation environment was used in all experiments[33]. TOSSIM is a discrete event simulator capable of accurately modelling sensor nodes and the modes of communications between them. An experiment is made of a single execution of the simulation environment using a specified protocol configuration, network nodes and safety period. An experiment terminated when any source node had been captured by an attacker during the safety period or the safety period had expired.

5.2 | Parameter Setup

A square grid network layout of size $n \times n$ was used in all experiments, with $n \in \{11, 15, 21, 25\}$, i.e., networks with 121, 225, 441 and 625 nodes respectively. The source and sink nodes were distinct and assigned positions in the *SourceCorner* configuration from[27], where the sink is in the centre and the source in the corner. The source period at which messages are sent from the real source is set to 1 second per message. The node neighbourhoods were generating using ideal model. Nodes were located 4.5 meters apart. Noise models were created using the first 2500 lines of `casino-lab.txt`[§]. At least 2000 repeats were performed for each combination of parameters.

[§] casino-lab.txt is a noise sample file provided with TOSSIM.

In Subsection 4.3, we introduced parameter $\mathcal{P}_{biased}$ used to implement biased random walk. The larger value of $\mathcal{P}_{biased}$ is, the bigger is the chance that the random walk will avoid walking close to the sink. In the simulation, we set this value to 0.9. When choosing the length of the short and long random walks for phantom walkabouts, a variety of parameter combinations were considered. Our experiments set the short random walk series $S = \{2, 3, \dots, 0.5 \times \Delta_{ss}\}$, and long random walk series $L = \{2 + \Delta_{ss}, \dots, 1.5 \times \Delta_{ss}\}$, where Δ_{ss} is the sink-source distance. In the phantom walkabouts, the short and long random walks are randomly generated from S and L during simulation runtime. The maximum times of message retransmission was set to 5 times.

Intuitively, the safety period captures the time period during which the asset will be at the same location. We calculate different safety periods P_{safety} as the following, where $\mathcal{T}_{flooding}$ is the time taken of an assert being captured for protectionless flooding.

$$P_{safety} = 1.3 \times \mathcal{T}_{flooding} \tag{3}$$

The reason why we choose a factor value 1.3 is because the safety period is longer than the the time taken of protectionless flooding so attackers have time potentially making bad moves. In fact other factor values are also applied. The $\mathcal{T}_{flooding}$ for each network size and source period, for protectionless flooding is shown in Table 3[¶]. Thus the P_{safety} is calculated based on the $\mathcal{T}_{flooding}$ when the source period is 1 second per message for each network size respectively.

TABLE 3 Time taken (seconds) of protectionless flooding

Network	**Source Period (seconds/message)**			
Size	**2.0**	**1.0**	**0.5**	**0.25**
11×11	19.38	9.93	5.29	3.04
15×15	27.47	13.98	7.39	4.07
21×21	40.50	20.40	10.56	5.64
25×25	48.48	24.59	12.72	6.71

6 | RESULTS

In this section we will use a metric called **capture ratio** to evaluate the SLP level. The capture ratio is the percentage of runs in which the source was captured. For example, if the attacker captures the source 20 times within the given safety period out of 100 simulation repeats, the capture ratio is 20%. The lower the capture ratio is, the higher the source location privacy level. Besides, we will also analyse other three key metrics: (i) **receive ratio**: the percentage of messages sent by the source and received at the sink, (ii) **message latency**: the time it takes a message sent by the source to be received at the sink, and (iii) **messages sent per second**: the number of messages sent by all nodes in the network per second.

As explained earlier in Subsection 4.1, we hypothesised that a short random walk will initially direct the attacker towards the source while a longer random walk will direct the attacker away from the source, thereby possibly increasing the SLP level. In this section, we seek to determine whether the hypotheses hold. We will first evaluate the performance of $PW(m, n)$ by varying m and n, and add other state-of-art SLP-aware routing protocols and phantom routing as a baseline for comparison.

6.1 | Results of Phantom Walkabouts

6.1.1 | Receive Ratio

A high receive ratio above 90% is observed. Fewer messages were delivered with larger networks. This suggests that the attacker was hearing most of the source messages, meaning that the privacy level imparted by the phantom walkabouts is due to the efficiency of the protocol and not due to the unreliability of the network. Another observation is that the low source period causes the low receive ratio but not significant. This is due to the traffic congestion with the low source period.

[¶] The results are generated from 10000 repeats of protectionless flooding.

6.1.2 | Capture Ratio

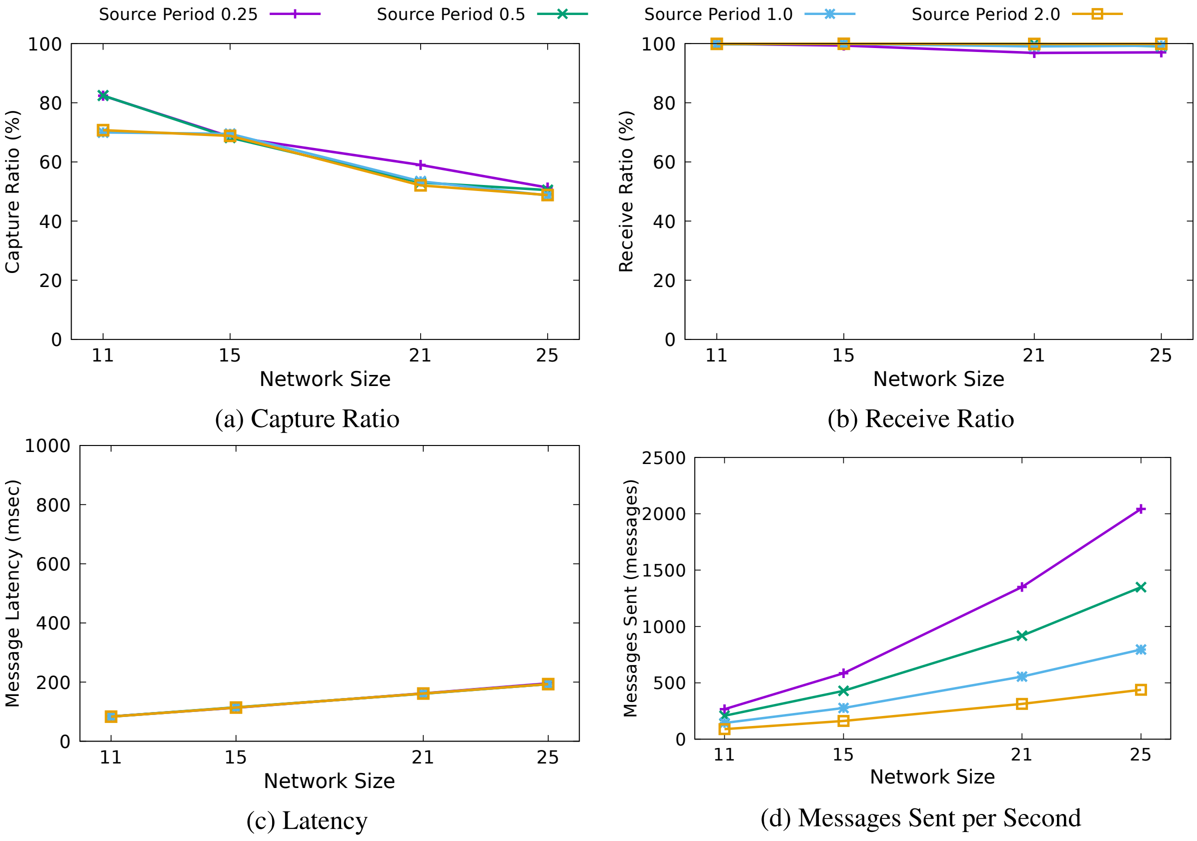
In Figure 5a, over 40% of capture ratio was observed, thereby confirming our conjecture that only short random walk cannot provide high level of SLP. Sequentially we added one long random walk in the phantom walkabouts (i.e., $PW(1,1)$). The results show that the capture ratio decreases to less than 10% (see Figure 6a). With the increase of long random walk, the capture ratio will further decrease. In the extreme case that no short random walk in the phantom walkabouts, the capture ratio is near 0%, providing near-optimal SLP. However, in this case the message latency increases since messages need to travel though long routes to the sink (see Figure 8c).

6.1.3 | Message Latency

The message latency is affected by the network sizes and number of long messages in one phantom walkabouts repeat. A large network size causes latency increase due to the long distance between the source and the sink. In addition, more long random walks ensure that messages costs more time to the phantom nodes before reaching the sink. In Figure 5c, the latency is between 100 milliseconds and 200 milliseconds because of no long random walk in $PW(1,0)$. However, the latency escalates to a new level from 300 milliseconds to 1000 milliseconds in $PW(0,1)$ (see Figure 8c) due to no short random walk message in phantom walkabouts repeat.

6.1.4 | 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 (i) the number of messages sent varies for different network sizes due to the long safety period; (ii) Lower source periods (i.e., faster message rates) require more messages sent per second, and (iii) the different combinations of short and long random walk in phantom walkabouts do not heavily influence the messages sent.



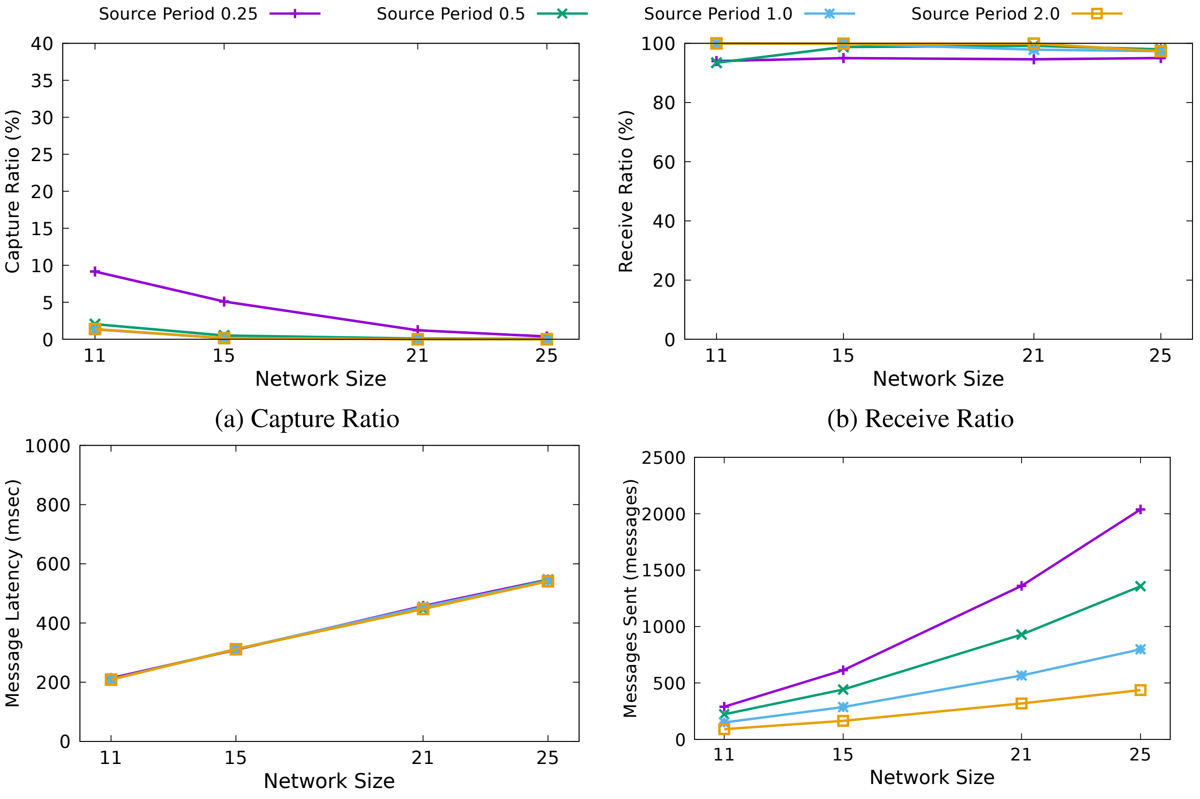
(a) Capture Ratio

(b) Receive Ratio

(c) Latency

(d) Messages Sent per Second

FIGURE 5 Results of PW(1,0): Only short random walk in the phantom walkabouts



(a) Capture Ratio

(b) Receive Ratio

(c) Latency

(d) Messages Sent per Second

FIGURE 6 Results of PW(1,1): Short random walk followed by **one** long random walk in the phantom walkabouts

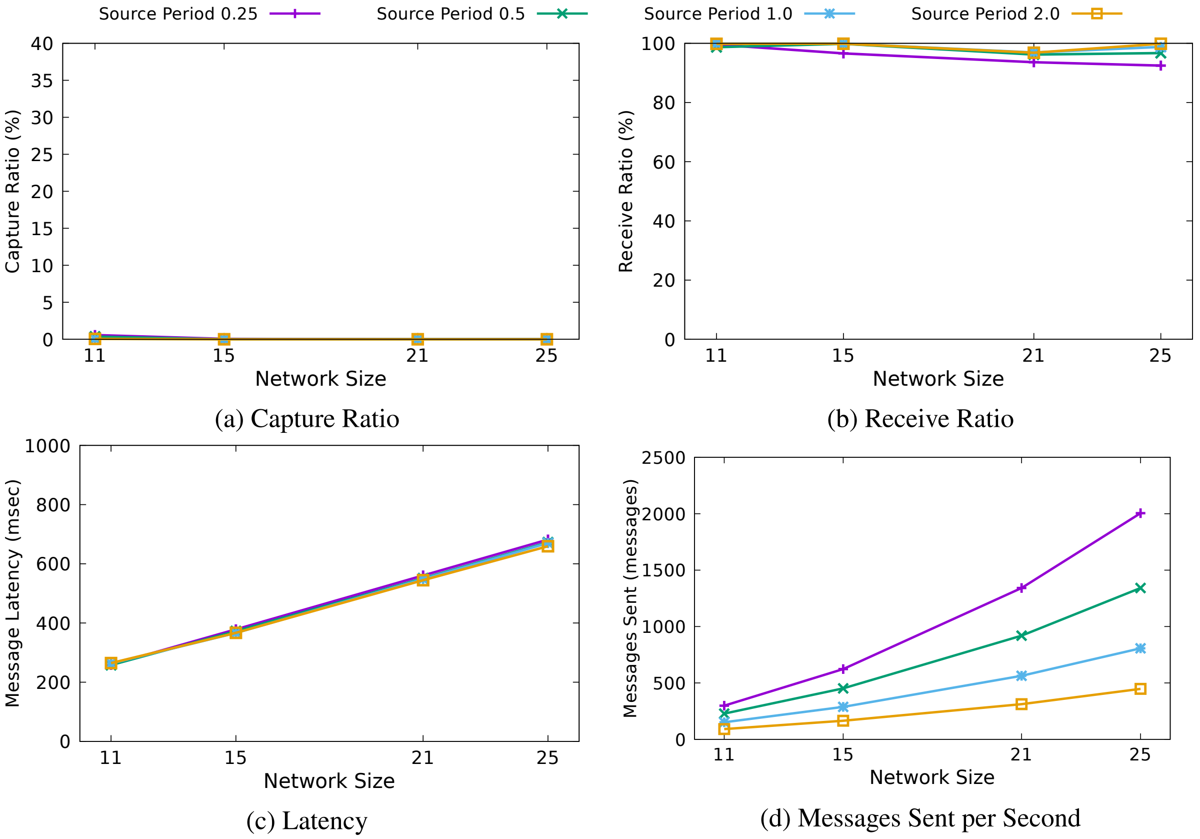
6.2 | Comparison with Other SLP-Aware Routing Protocols

Previous results have shown that our proposed routing protocol can achieve near 0% capture ratio at the best (see Figure 7a and Figure 8a) and very high receive ratio. In this section, to further investigate the performance of our solution, phantom routing is compared as the baseline routing protocol. Besides, we added other two state-of-art protocols that can also achieve SLP for comparison: DynamicSPR[24] and ILP Routing[27]. We choose these two because they adopt different techniques to provide SLP: fake sources and message delay respectively. Those two routing protocols are also instances of two classes of SLP protocols: spatially-aware protocol and temporal-aware protocol respectively[34].

The results were generated under the simulation environment as the same as our solution. All the source period is set to 1 message per second. The results of our proposed solution is from $PW(1,2)$ (see Figure 7). Specific parameters for other protocols are listed as follows:

- Phantom Routing: The random walk is selected as fixed 8 hops due to the fact that long random walks may provide good SLP.
- DynamicSPR: The parameter `Rnd` determines how many fake messages are sent over the lifetime of a temporary fake source. It sends either 1 or 2 messages randomly chosen over the duration.
- ILP Routing: This algorithm 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 1 message. The buffer size was set to 10 messages. Finally, the probability of sending a message directly to the sink was set to 20% as it was identified as a good setting in[27].

Now we make such observations from Figure 9:



(a) Capture Ratio

(b) Receive Ratio

(c) Latency

(d) Messages Sent per Second

FIGURE 7 Results of PW(1,2): Short random walk followed by **two** long random walk in the phantom walkabouts

- Phantom routing produces 20% to 25% of capture ratio. However, the receive ratio is 80%, meaning that an amount of messages cannot be received by the sink (see Figure 9b). Therefore, the low messages sent and capture ratio are due to the low receive ratio (see Figure 9a and Figure 9d). The low receive ratio proves the fact that messages are lost in transmission between two neighbouring nodes without ACK messages.

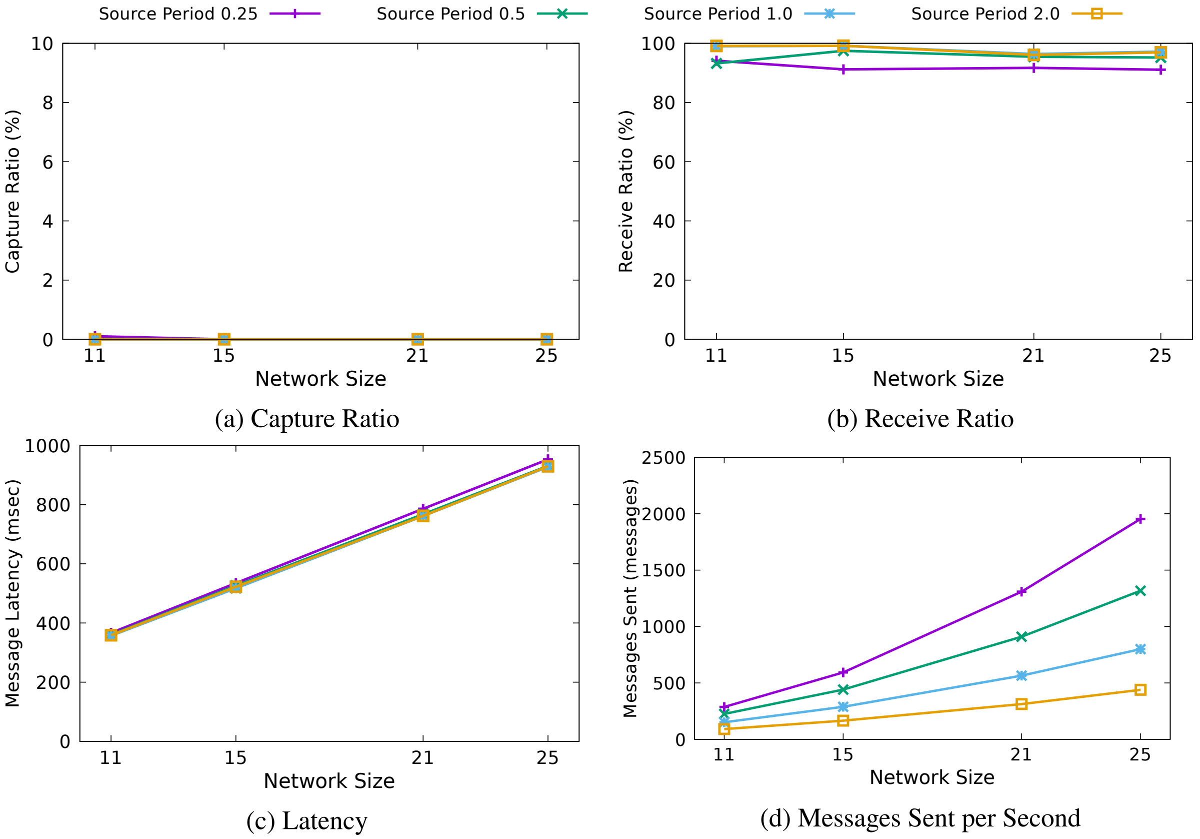
- The results show DynamicSPR, ILP Routing and our proposed solution can achieve near-optimal SLP (see Figure 9a). Meanwhile, the receive ratio of DynamicSPR is near 100% and 80% is observed for ILP Routing (see Figure 9b). However, the weaknesses of both algorithms are to introduce much overheads to achieve such high level of SLP. Specifically, for DynamicSPR the messages sent are higher than our solution and increase greatly with larger network size (see Figure 9d); latency in ILP Routing doubled higher than our solution (see Figure 9c).

- The phantom walkabouts provides much better SLP and much reliable receive ratio than phantom routing but higher message latency and messages sent. This is due to the use of ACK messages, which increases the time costs and number of messages sent between two neighbouring nodes. Therefore, the trade-offs need to be made between SLP and other attributes.

Overall, DynamicSPR and ILP Routing can achieve high level of SLP but not bounding the overheads, which restrict their practical applications. For example, DynamicSPR can only be deployed in the network that nodes are with power scavengers (e.g., solar panels) to support high energy consumption; ILP Routing can be deployed in the delay-tolerant networks, and phantom routing can be deployed in the networks without very high level of SLP requirement.

7 | DISCUSSION

In this section, we discuss some issues and observations that arose as a result of this work.



(a) Capture Ratio

(b) Receive Ratio

(c) Latency

(d) Messages Sent per Second

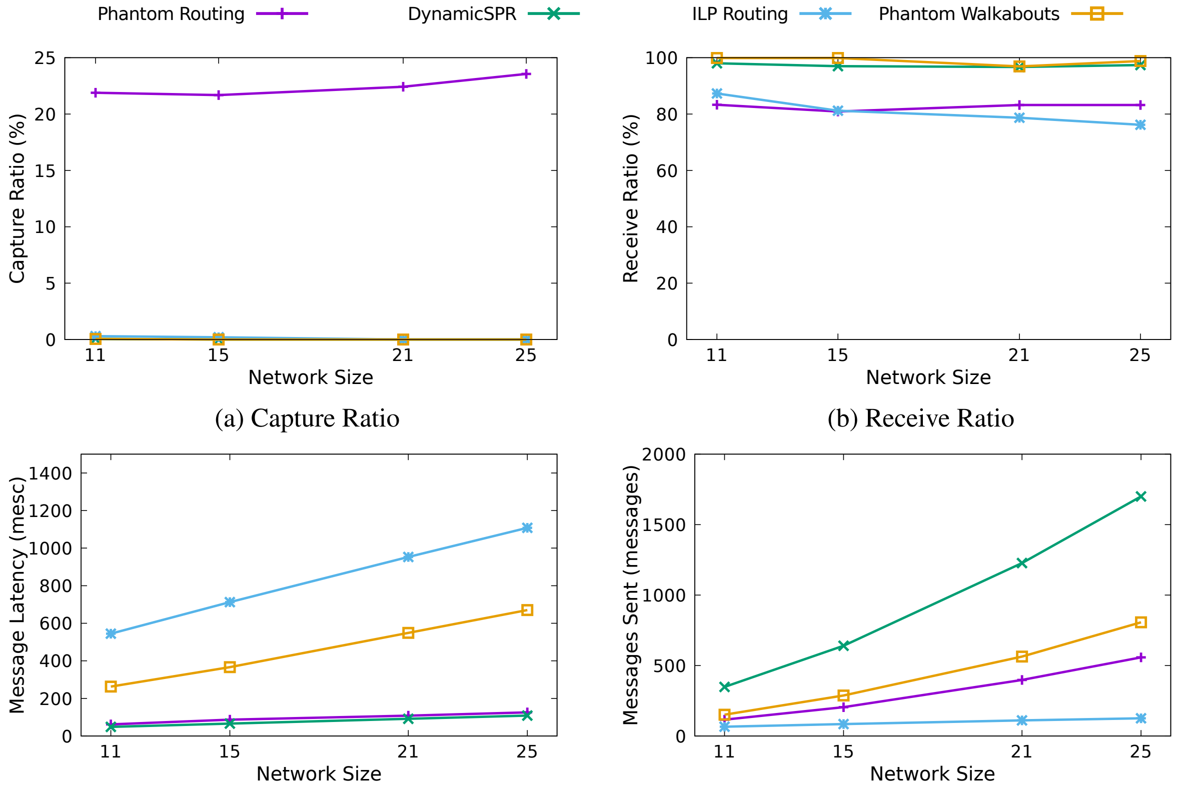
FIGURE 8 Results of PW(0,1): Only long random walk in the phantom walkabouts

7.1 | Flooding phase in phantom walkabouts

In this paper, we use flooding to deliver messages from the source to the sink. The reasons we adopt flooding are: (i) the phantom walkabouts and phantom routing are comparable because they both use flooding; (ii) flooding is simple, high reliable, and it requires no costly topology maintenance nor complex route discovery; (iii) flooding is used as it is the routing algorithm that offers no SLP[8]. Any improvement in SLP levels is then due to the SLP algorithm used. However, flooding has several shortcomings such as overlap and energy consumption[35]. Therefore, the flooding protocol can be replaced by other enhanced protocols such as gossiping[36]. In this case, the impact of using gossiping instead of flooding on SLP needs to be investigated.

7.2 | Different attacker models

In this paper, we have assumed a distributed eavesdropper that backtracks on the network traffic to capture the source within the safety period. This type of attacker monitors a range of messages transmitted and quickly move to the position where data were sent from. We use this attacker model due to its common adoption. However, an alternative to this would be a *patient* attacker model. For a patient attacker, he could wait at one location to gather information. After receiving a number of messages, it could choose to move based on the information gathered. A problem with this attacker model is that waiting reduces the number of moves that the attacker could potentially take. If the attacker waits too long, then the safety period will expire and it will fail to capture the source. While the patient attacker is an alternative attacker model, it may not always perform well. Another possible type of attacker can deduce the routing algorithm in the network by analysing the network traffic. However, this attacker needs to contain more resources including powerful computation and enough storage. Dealing with other attacker models will be part of our future work.



(a) Capture Ratio

(b) Receive Ratio

(c) Latency

(d) Messages Sent per Second

FIGURE 9 Comparison of Other SLP-aware Routing Protocols

7.3 | Choice of Parameters

The proposed routing protocol relies on the choice of parameters short and long random walks in the phantom walkabouts. The parameterisations can be finely tuned and generated by the system administrator or application developer for practical scenarios. For example, in the latency-aware network scenario that messages need to be delivered to the sink within the latency deadline (e.g., battlefield applications), too many long random walk messages should not be used in the phantom walkabouts. In this case, a trade-off need to be made between the capture ratio and message latency. Overall the parameters should vary to fit the specific network environment.

7.4 | Lack of Testbed Experiments

Our results were generated from TOSSIM rather than real testbeds. The reason is that current testbeds are not very fit for testing our solution. For example, the topology of the testbeds such as FlockLab[37] has insufficient space around the sink for messages to be routed so that the efficiency of our protocol cannot be fully tested; FIT/IoT-LAB[38] is a large scale open experimental IoT testbed. However, the network in FIT/IoT-LAB has a very dense topology. What is ideally required is a testbed where node distributed is uniform.

8 | CONCLUSION

In this paper, we firstly show the limitations of phantom routing algorithm. Then we have proposed a novel technique called phantom walkabouts, which extends the phantom routing, to provide a better level of SLP than phantom routing. Phantom walkabouts proposes to interleave sequences of short random walks and long random walks to attempt to make the attacker move in the wrong direction, as opposed to phantom routing (with small random walks) where an attacker moves towards the source.

We also implement it with messages retransmission technique to achieve very high data yield. We have shown that phantom walkabouts provides much better SLP and receive ratio than phantom routing and equal SLP with other state-of-art SLP solutions.

For future work, we plan to investigate phantom walkabouts with dynamic short and long random walks, i.e., in our current experiments, we use one given value for short random walks and a different value for long random walks in one phantom walkabouts repeat. However, this needs not be the case. In fact, we conjecture that better performance can be achieved by dynamically varying the number of the short and long random walks. As described in Section 7, it would also be informative to consider alternative intelligent attacker models and testbed experiments.

References

1. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. A survey on sensor networks. *IEEE Communications Magazine* 2002; 40(8): 102–105. doi: 10.1109/MCOM.2002.1024422

2. Dyo V, Yousef K, Ellwood SA, et al. WILDSENSING: Design and deployment of a sustainable sensor network for wildlife monitoring. *ACM Transactions on Sensor Networks* 2012; 8(4): 1–33. doi: 10.1145/2240116.2240118

3. Mainwaring A, Culler D, Polastre J, Szewczyk R, Anderson J. Wireless Sensor Networks for Habitat Monitoring. In: ACM; 2002; Atlanta, Georgia, USA: 88–97

4. Arampatzis T, Lygeros J, Manesis S. A Survey of Applications of Wireless Sensors and Wireless Sensor Networks. In: Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control. ; 2005; Limassol, Cyprus: 719–724

5. Chipara O, Lu C, Bailey TC, Roman GC. Reliable clinical monitoring using wireless sensor networks. In: Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems. ; 2010; New York, NY, USA: 155–168

6. Perrig A, Szewczyk R, Wen V, et al. SPINS: Security Protocols for Sensor Networks. *Wireless Networks* 2002; 8(5): 521–534. doi: 10.1023/A:1016598314198

7. Wildlife Crime Technology Project. 2012. https://www.worldwildlife.org/projects/wildlife-crime-technology-project (Accessed On: 2018-02-24).

8. Kamat P, Zhang Y, Trappe W, Ozturk C. Enhancing Source-Location Privacy in Sensor Network Routing. In: 25th IEEE International Conference on Distributed Computing Systems (ICDCS). ; 2005; Columbus, OH, USA: 599–608

9. Ozturk C, Zhang Y, Trappe W, Ott M. Source-location privacy for networks of energy-constrained sensors. In: Second IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems. ; 2004; Vienna, Austria: 68–72

10. Yang Y, Shao M, Zhu S, Cao G. Towards Statistically Strong Source Anonymity for Sensor Networks. *ACM Trans. Sen. Netw.* 2013; 9(3): 34:1–34:23. doi: 10.1145/2480730.2480737

11. Jhumka A, Bradbury M, Leeke M. Fake source-based source location privacy in wireless sensor networks. *Concurrency Computation Practice and Experience* 2015; 27(12): 2999–3020. doi: 10.1002/cpe.3242

12. Long J, Dong M, Ota K, Liu A. Achieving Source Location Privacy and Network Lifetime Maximization Through Tree-Based Diversionary Routing in Wireless Sensor Networks. *IEEE Access* 2014; 2: 633–651. doi: 10.1109/ACCESS.2014.2332817

13. Gu C, Bradbury M, Jhumka A, Leeke M. Assessing the Performance of Phantom Routing on Source Location Privacy in Wireless Sensor Networks. In: ; 2015; Zhangjiajie, China: 99–108

14. Shaikh RA, Jameel H, D'Auriol BJ, Lee H, Lee S, Song YJ. Achieving network level privacy in wireless sensor networks. *Sensors* 2010; 10(3): 1447–1472. doi: 10.3390/s100301447

15. Lightfoot L, Li Y, Ren J. Preserving source-location privacy in wireless sensor network using STaR routing. In: IEEE Global Telecommunications Conference. ; 2010; Miami, FL, USA: 1–5

16. Wang WP, Chen L, Wang JX. A source-location privacy protocol in WSN based on locational angle. In: IEEE International Conference on Communications. ; 2008; Beijing, China: 1630–1634

17. Xi Y, Schwiebert L, Shi W. Preserving source location privacy in monitoring-based wireless sensor networks. In: Proceedings 20th IEEE International Parallel and Distributed Processing Symposium. ; 2006; Rhodes Island, Greece: 8 pp.

18. Gu C, Bradbury M, Jhumka A. Phantom walkabouts in wireless sensor networks. In: Proceedings of the ACM Symposium on Applied Computing. ; 2017; Marrakech, Morocco: 609–616

19. Chen H, Lou W. On protecting end-to-end location privacy against local eavesdropper in wireless sensor networks. *Pervasive and Mobile Computing* 2015; 16: 36–50. doi: 10.1016/j.pmcj.2014.01.006

20. Wang H, Han G, Zhu C, Chan S, Zhang W. TCSLP: A trace cost based source location privacy protection scheme in WSNs for smart cities. *Future Generation Computer Systems* 2017. doi: 10.1016/j.future.2017.07.051

21. Luo X, Ji X, Park MS. Location Privacy against Traffic Analysis Attacks in Wireless Sensor Networks. In: International Conference on Information Science and Applications. ; 2010; Seoul, South Korea: 1–6

22. Ozturk C, Zhang Y, Trappe W. Source-location privacy in energy-constrained sensor network routing. In: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks - SASN '04. ; 2004; Washington DC, USA: 88–93

23. Chen H, Lou W. From nowhere to somewhere: Protecting end-to-end location privacy in wireless sensor networks. In: Conference Proceedings of the IEEE International Performance, Computing, and Communications Conference. ; 2010; Albuquerque, NM, USA: 1–8

24. Bradbury M, Jhumka A, Leeke M. Hybrid online protocols for source location privacy in wireless sensor networks. *Journal of Parallel and Distributed Computing* 2018; 115: 67–81. doi: 10.1016/j.jpdc.2018.01.006

25. Mehta K, Liu D, Wright M. Protecting Location Privacy in Sensor Networks against a Global Eavesdropper. *IEEE Transactions on Mobile Computing* 2012; 11(2): 320–336. doi: 10.1109/TMC.2011.32

26. Jhumka A, Leeke M, Shrestha S. On the use of fake sources for source location privacy: Trade-Offs between energy and privacy. *Computer Journal* 2011; 54(6): 860–874. doi: 10.1093/comjnl/bxr010

27. Bradbury M, Jhumka A. A Near-Optimal Source Location Privacy Scheme for Wireless Sensor Networks. In: 2017 IEEE Trustcom/BigDataSE/ICESS. ; 2017; Sydney, NSW, Australia: 409–416

28. Hong X, Wang P, Kong J, Zheng Q, Liu J. Effective probabilistic approach protecting sensor traffics. In: IEEE Military Communications Conference (MILCOM). ; 2005; Atlantic City, NJ, USA: 169–175

29. Li N, Raj M, Liu D, Wright M, Das SK. Using data mules to preserve source location privacy in Wireless Sensor Networks. *Pervasive and Mobile Computing* 2014; 11: 244–260. doi: 10.1016/j.pmcj.2012.10.002

30. Han G, Zhou L, Wang H, Zhang W, Chan S. A source location protection protocol based on dynamic routing in WSNs for the Social Internet of Things. *Future Generation Computer Systems* 2018; 82: 689–697. doi: 10.1016/j.future.2017.08.044

31. Fan Y, Chen J, Lin X, Shen X. Preventing traffic explosion and achieving source unobservability in multi-hop wireless networks using network coding. In: GLOBECOM - IEEE Global Telecommunications Conference. ; 2010; Miami, FL, USA: 0–4

32. Baccour N, Koubaa A, Noda C, et al. Radio Link Quality Estimation in Wireless Sensor Networks: A Survey. *ACM Transactions on Sensor Networks (TOSN)* 2012; 8(4): 34:1–34:33. doi: 10.1145/2240116.2240123

33. Levis P, Lee N, Welsh M, Culler D. TOSSIM: accurate and scalable simulation of entire TinyOS applications. In: Proceedings of the 1st international conference on Embedded networked sensor systems. ; 2003; Los Angeles, California, USA: 126–137

34. Jhumka A, Bradbury M. Deconstructing source location privacy-aware routing protocols. In: Proceedings of the Symposium on Applied Computing. ; 2017; Marrakech, Morocco: 431–436

35. Al-Karaki J, Kamal A. Routing Techniques in Wireless Sensor Networks: A Survey. *IEEE Wireless Communications* 2004; 11(6): 6–28. doi: 10.1109/MWC.2004.1368893

36. Boyd S, Ghosh A, Prabbakar B, Shah D. Gossip algorithms: design, analysis and applications. *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.* 2005; 3: 1653–1664. doi: 10.1109/INFCOM.2005.1498447

37. Lim R, Ferrari F, Zimmerling M. FlockLab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems. In: ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN). ; 2013; Philadelphia, Pennsylvania, USA: 153–165

38. Adjih C, Baccelli E, Fleury E, et al. FIT IoT-LAB: A large scale open experimental IoT testbed. In: IEEE 2nd World Forum on Internet of Things (WF-IoT). ; 2015; Milan, Italy: 459–464

□