Sensors **2010**, *10*, 1447-1472; doi:10.3390/s100301447

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Achieving Network Level Privacy in Wireless Sensor Networks †

Riaz Ahmed Shaikh [1], Hassan Jameel [2,‡], Brian J. d'Auriol [1], Heejo Lee [3], Sungyoung Lee [1,⋆] and Young-Jae Song [1]

[1] Department of Computer Engeering, Kyung Hee University, Global Campus, Korea;
E-Mails: riaz289@acm.org (R.A.S.); dauriol@oslab.khu.ac.kr (B.J.d'A.); yjsong@khu.ac.kr (Y.J.S.)

[2] Computing Department, Macquarie University, NSW, Australia; E-Mail: hasghar@science.mq.edu.au

[3] Department of Computer Science & Engeering, Korea University, Seoul, Korea;
E-Mail: heejo@korea.ac.kr

† This paper is an extended version of our paper entitled "Network level privacy for wireless sensor networks" that has been published in proceedings of the 4^{th} International Conference on Information Assurance and Security (IAS 08), that was held in Naples, Italy in September 2008 (pp. 261-266).

‡ Contributed to this work when he was in Kyung Hee University, 1 Hoegi-dong, Dongdaemun-gu, Seoul 130-701, Korea.

⋆ Author to whom correspondence should be addressed; E-Mail: sylee@oslab.khu.ac.kr.

Received: 17 December 2009; in revised form: 4 February 2010 / Accepted: 9 February 2010 / Published: 26 February 2010

Abstract: Full network level privacy has often been categorized into four sub-categories: *Identity*, *Route*, *Location* and *Data* privacy. Achieving full network level privacy is a critical and challenging problem due to the constraints imposed by the sensor nodes (e.g., energy, memory and computation power), sensor networks (e.g., mobility and topology) and QoS issues (e.g., packet reach-ability and timeliness). In this paper, we proposed two new identity, route and location privacy algorithms and data privacy mechanism that addresses this problem. The proposed solutions provide additional trustworthiness and reliability at modest cost of memory and energy. Also, we proved that our proposed solutions provide protection against various privacy disclosure attacks, such as eavesdropping and hop-by-hop trace back attacks.

Keywords: anonymity; eavesdropping; hop-by-hop trace back; privacy; routing; wireless sensor networks

---

1. Introduction

With the spreading application of Wireless Sensor Networks (WSNs) in various sensitive areas such as health-care, military, habitat monitoring, *etc*, the need to ensure security and privacy is becoming imperatively important. For example, in battlefield application scenario, "the location of a soldier should not be exposed if he initiates broadcast query" [1]. In the meantime, query must be transferred to the destination in an encrypted manner via only trusted en-route nodes. Similarly, in habitat monitoring application scenarios, such as Great Duck Island [2] or Save-the-panda application [3] where large numbers of sensor nodes are deployed to observe the vast habitat of ducks and pandas, an adversary can try to capture the panda or duck by back-tracing the routing path until it reaches the source sensor nodes. Therefore, in order to prevent the adversary from back-tracing, the route, location and data privacy mechanisms must be enforced. With respect to these application scenarios, network level privacy has often been categorized into four categories:

1. Sender node identity privacy: no intermediate node can get any information about who is sending the packets except the source, its immediate neighbors and the destination,
2. Sender node location privacy: no intermediate node can have any information about the location (in terms of physical distance or number of hops) about the sender node except the source, its immediate neighbors and the destination,
3. Route privacy: no node can predict the information about the complete path (from source to destination). Also, a mobile adversary gets no clue to trace back the source node either from the contents and/or directional information of the captured packet(s), and
4. Data packet privacy: no node can see the information inside in a payload of the data packet except the source and the destination.

Existing privacy schemes such as [1, 3–7], that have specifically been proposed for WSNs only provide partial network level privacy. Providing a full network level privacy is a critical and challenging issue due to the constraints imposed by the sensor nodes (e.g., energy, memory and computation power), sensor network (e.g., mobility and topology) and QoS issues (e.g., packet reach-ability and trustworthiness). Thus, an energy-efficient privacy solution is needed to address these issues.

In order to achieve this goal, we incorporate basic design features from related research fields such as geographic routing and cryptographic systems. To our knowledge, we propose the first full network level privacy solution for WSNs. Our contribution lies in following features.

- A new Identity, Route and Location (IRL) privacy algorithm is proposed that ensures the anonymity of source node's identity and location. It also assures that the packets will reach their destination by passing through only trusted intermediate nodes.
- A new reliable Identity, Route and Location (r-IRL) privacy algorithm is proposed, which is the extension of our proposed IRL algorithm. This algorithm has the ability to forward packets from multiple secure paths to increase the packet reach-ability.

- A new data privacy mechanism is proposed, which is unique in the sense that it provides data secrecy and packet authentication *in the presence of identity anonymity*.

Our solutions collectively provide protection against various privacy disclosure attacks such as eavesdropping and hop-by-hop trace-back attacks. Also, our solutions are lightweight, hence consume modest memory and energy.

The rest of this paper is organized as follows: Section 2. contains related work, Section 3. articulates the network model, assumptions and adversary model. Section 4. describes the proposed privacy schemes, Section 5. consists of analysis and evaluation, and Section 6. concludes the paper.

2. Related Work

2.1. Privacy Schemes

A number of a privacy schemes such as [1, 3–7] have been proposed for WSNs that are discussed below.

C. Ozturk *et al*. [3] proposed a phantom routing scheme for WSNs, which helps to prevent the location of a source from the attacker. In this scheme, each message reaches the destination in two phases: 1) a walking phase, in which the message is unicasted in a random fashion within first h_{walk} hops, 2) after that, the message is flooded using the baseline flooding technique. The major advantage of their scheme is the source location privacy protection, which improves as the network size and intensity increase because of high path diversity. But on the other hand, if the network size increases, the flooding phase will consume more energy. This scheme does not provide identity privacy. Also, it is unable to provide data secrecy in the presence of identity privacy.

P. Kamat *et al*. [4] proposed a phantom single-path routing scheme that works in a similar fashion as the original phantom routing scheme [3]. The major difference between these two schemes is that after the walking phase, a packet will be forwarded to the destination via a single path routing strategy such as the shortest path routing mechanism. This scheme consumes less energy and requires slightly higher memory as compared to first one. This scheme also does not provide identity privacy. Also, it is unable to provide data secrecy in the presence of identity privacy.

S. Misra and G. Xue [5] proposed two schemes: Simple Anonymity Scheme (SAS) and Cryptographic Anonymity Scheme (CAS) for establishing anonymity in clustered WSNs. The SAS scheme use dynamic pseudonyms instead of true identity during communications. Each sensor node needs to store a given range of pseudonyms that are non-contiguous. Therefore, the SAS scheme is not memory efficient. On the other hand, the CAS scheme uses keyed hash functions to generate pseudonyms. This scheme is memory efficient as compare to the SAS but it requires more computation power. The authors do not propose any routing scheme. Sender node may always send packets to the destination via shortest path. In that case, for an adversary who is capable of performing hop-by-hop trace back (with the help of direction information) can find out the location of the source node.

Y. Xi *et al*. [1] proposed a Greedy Random Walk (GROW) scheme to protect the location of the source node. This scheme works in two phases. In a first phase, the sink node will set up a path through random walk with a node as a receptor. Then the source node will forward the packets towards the receptor in a random walk manner. Once the packet reaches at the receptor, it will forward the packet to the sink

node through the pre-established path. Here receptor is acting a central point between the sink and the source node for every communication session. A criterion of selecting a trustworthy receptor is essential, however not defined in the author's work.

Y. Ouyang *et al*. [7] proposed a Cyclic Entrapment Method (CEM) to minimize the chance of an adversary in finding out the location of the source node. In the CEM, when the message is sent by the source node to the base station, it will activate the predefined loop(s) along the path. An activation node will generate the fake message and forwarded it towards the loop, and original message is forwarded to the base station via specific routing protocol such as shortest path. Energy consumption in the CEM scheme is mainly dependent on the number of existing loops in the path and their size.

2.2. Geographic Routing Schemes

Our proposed privacy solutions incorporate the basic design features from geographic routing schemes [6, 8–10] that are discussed below.

M. Zorzi and R. R. Rao [8, 9] proposed Geographic Random Forwarding (GeRaF) scheme for ad hoc and sensor networks. This scheme is based on broadcast transmission and the sender only requires the position of its own and the destination. All active neighborhood nodes who receive the packet will go through the contention phase. Once the contention phase is complete, the winner (the node that is closest to the destination) will relay the packet using the same mechanism. This process will repeat until the destination becomes one-hop away. The authors assumed that all nodes in the neighborhood do not remain active all the time. Due to the dynamics of the sleep modes, different sets of potential relays will be available. However, mostly the potential route is close to the same or shortest route, which makes easier for an adversary to trace back to the sender.

A. Capone *et al*. [10] proposed Simple Forwarding over Trajectory (SiFT) scheme. This scheme is based on broadcast transmission and does not maintain neighborhood positions and states. Each node who receives the packet will make the decision of forwarding that packet based only on its own position, the position of a transmitter and the trajectory. The difference between the GeRaF [8, 9] and the SiFT scheme is that, the GeRaF does not use trajectories but the position of the destination. If nodes are static then similar to the GeRaF the potential route is close to the same or shortest route, which makes it easier for an adversary to trace back to the sender.

A. D. Wood *et al*. [6] have proposed a configurable secure routing protocol family called Secure Implicit Geographic Forwarding (SIGF) for WSNs. The SIGF is based on the Implicit Geographic Forwarding (IGF) protocol [11], in which a packet is forwarded to the node that lies within the region of 60° sextant, centered on the direct line from the sender to the destination. The SIGF protocol provides some aspects of networks privacy such as data, route and location privacy, but it does not provide identity privacy. Another limitation of the SIGF protocol is that, when there is no trusted node within a forwarding area (assuming 60° sextant), it will forward the packet to an un-trusted node. So, the reliability of the path is affected.

Table 1 compares the proposed privacy preserving schemes. It clearly shows that none of the schemes currently provide full network level privacy.

Table 1. Comparison of privacy preserving schemes.

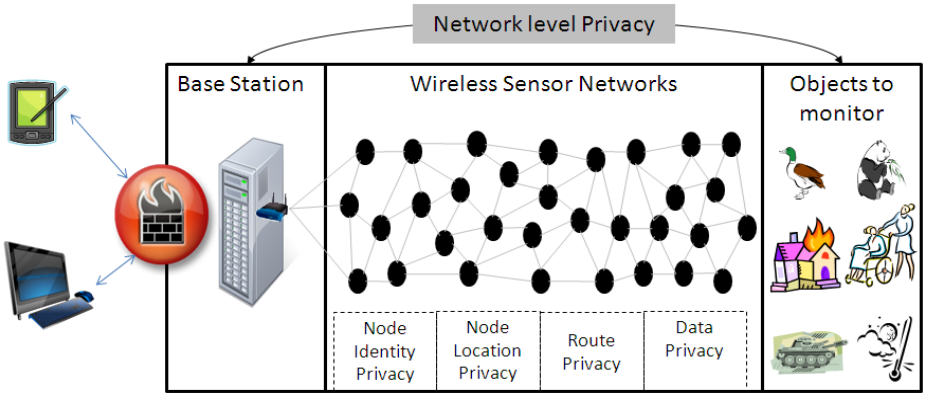
	PFR [3]	PSR [4]	SAS & CAS [5]	CEM [7]	SIGF [6]	GeRaF [8, 9]	SiFT [10]
Required information for routing	ID of destination	Routing table (e.g., destination ID, # of hops *etc*.)	Depending on a routing scheme	Depending on a routing scheme	Own, destination, & neighborhood locations	Own and destination location	Destination trajectory and own location
Transmission mechanism	1st phase: Point-to-point; 2nd phase: Broadcast	Point-to-point	Depending on a routing scheme	Depending on a routing scheme	Point-to-point	Broadcast	Broadcast
Decision place for forwarding	1st phase: Transmitter; 2nd phase: Receiver	Transmitter	Depending on a routing scheme	Depending on a routing scheme	Transmitter	Receiver	Receiver
Criteria for forwarding packet to next hop	1st phase: random; 2nd phase: flooding	1st phase: random; 2nd phase: shortest in terms of hops	Depending on a routing scheme	Depending on a routing scheme	Randomly select any trusted node lies in forwarding region	Node that is closer to the destination in terms of location	Node that is closer to the destination in terms of trajectory
Identity privacy	Not Available	Not Available	Available	Not Available	Not Available	Not Applicable	Not Applicable
Route privacy	Available	Available	Depending on a routing scheme	Depending on a routing scheme	Available	Available	Available
Location privacy	Available	Available	Not Available	Available	Available	Not Applicable	Not Applicable
Data privacy	Not Available	Not Available	Available	Available	Available	Not Applicable	Not Applicable

3. Network, Assumptions and Adversary Model

3.1. Network Model

A wireless sensor network (WSN) is composed of large number of small sensor nodes that are of limited resource and densely deployed in an environment. Whenever end users require information about any event related to some object(s), they send a query to the sensor network via the base station. And the base station propagates that query to the entire network or to a specific region of the network. In response to that query, sensor nodes send back required information to the base station. A typical wireless sensor network scenario is shown in Figure 1. Links are bidirectional. Also, sensor nodes use IEEE 802.11 standard link layer protocol, which keeps packets in its cache until the sender receives an acknowledgment (ACK). Whenever a receiver (next hop) node successfully receives the packet it will send back an ACK packet to the sender. If the sender node does not receive an ACK packet during predefined threshold time, then the sender node will retransmit that packet.

Figure 1. Typical WSN scenario.



3.2. Assumptions

For reason of scalability, it is assumed that no sensor node needs to know the global network topology, except that it must know the geographical location of its own, its neighboring nodes and the base station. In order to find out the location information, any proposed mechanism could be used, such as [12, 13].

It is assumed that each sensor node in the network can share a unique secret key with the base station [14, 15]. These keys are periodically updated. The public key of the base station is also assumed known to all the nodes in the network. Sensor nodes do not require their own public and private keys; because computation cost of public and private keys is generally consider being high. However, many researchers [16, 17] have shown the feasibility of using public key cryptography in wireless sensor networks. It is also assumed that sensor nodes are capable of performing encryption and decryption of the data by using any cipher algorithm such as DES, AES *etc*. This provides an additional layer of security.

This paper only focuses on the development of a prevention strategy against network level privacy disclosure attacks, such as eavesdropping, traffic analysis and hop-by-hop trace back attacks. Other general attacks, such as flooding attacks, could be detected and prevented by using any IDS scheme proposed for WSNS.

3.3. Adversary Model

We have assumed that an adversary can mostly perform passive attacks (like eavesdropping [18], and traffic analysis), since such attacks helps to conceal the adversary's presence in the network. Nevertheless, the adversary is also capable of performing some active attacks like fabrication and packet drop attacks. We also assumed that the adversary is both device-rich and resource-rich [4]. These characteristics are defined below.

- Device-rich: the adversary is equipped with devices like antenna and spectrum analyzers, so that the adversary can measure the angle of arrival of the packet and received signal strength. These devices will help the adversary to find out the immediate sender of the packet and move to that node. This kind of hop-by-hop trace back mechanism will be carried out by the adversary until the actual sender node is reached.

- Resource-rich: the adversary has no resource constraint in computation power, memory or energy.

It is also assumed that the adversary has some basic domain knowledge like the range of identities assigned to the sensor nodes, the public key of the base station and information about the cipher algorithms used in the network. However, adversary has no knowledge which identity is physically associated with which node.

A detection and prevention strategy against non-privacy disclosure attacks at various layers such as jamming attacks is out of the scope of this paper. However, trust management methodology (Section 4.1) that we adopted in this paper is useful to detect and prevent some non-privacy disclosure threats such as, black hole attack, sink hole attack, and selective forwarding or gray hole attack.

4. Proposed Scheme

4.1. Concepts and Definitions

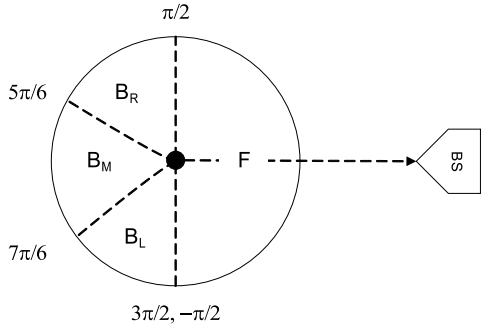
In our proposed algorithms, we have used two notions: direction and trust. Both these notions (direction and trust) are used to provide reliable (non-malicious and non-faulty) secure paths for achieving robust route privacy. Direction information will help to forward packet to the destination in a timely manner and trust will help to forward the packets via reliable nodes. Detail definitions of both notions are given below.

Direction: The first notion used in our algorithms is that of direction. The physical location of the base station is the reference point for each sensor node. Based on this reference point, each node classifies its neighboring nodes into four categories: (1) forward neighboring nodes (F), (2) right side backward neighboring nodes (B_r), (3) left side backward neighboring nodes (B_l), and (4) middle backward neighboring nodes (B_m). The objective of this categorization is to provide more path diversity as discussed in Section 4.2. A node x classifies its neighboring node y in following fashion:

$$C_{x,y} = \begin{cases} F & \frac{-\pi}{2} \leq \theta \leq \frac{\pi}{2} \\ B_r & \frac{\pi}{2} < \theta \leq \frac{5\pi}{6} \\ B_m & \frac{5\pi}{6} < \theta \leq \frac{7\pi}{6} \\ B_l & \frac{7\pi}{6} < \theta < \frac{3\pi}{2} \end{cases} \quad (1)$$

where θ is the angle between the node x and its neighboring node y with respect to the line joining node x and the base station as shown in Figure 2.

Figure 2. Neighbor node classification



Trust: The second notion used in our algorithms is that of trust. The definition of a trust here is based on our other paper [19] and restated here.

A node can be classified into one of the three categories [20]: trustworthy, untrustworthy, and uncertain. A node is considered trustworthy if it interacts successfully most of the time with the other nodes. A node is considered untrustworthy if it tries to do as many unsuccessful interactions as possible with the other nodes. An untrustworthy node could be a faulty [21] or malicious node. A node is considered uncertain if it performs both successful and unsuccessful interactions. Detailed definition of the successful and unsuccessful interactions and trust calculation methodology is available in our paper [22] and provided here in a simplified form.

A sender will consider an interaction successful if the sender receives confirmation that the packet is successfully received by the neighbor node and forwarded towards the destination in an unaltered fashion. The first requirement of successful reception is achieved on the reception of the link layer acknowledgment (ACK). The second requirement of forwarding towards the destination is achieved with the help of enhanced passive acknowledgment (PACK) by overhearing the transmission of a next hop on the route, since they are within the radio range [23]. If the sender node does not overhear the retransmission of the packet within a threshold time from its neighboring node or if the overheard packet is found to be illegally fabricated (by comparing the payload that is attached to the packet), then the sender node will consider that interaction as unsuccessful.

With the help of this simple approach, several attacks can be prevented, *i.e.*, the black hole attack is straightforwardly detected when malicious node drops the incoming packets and keeps sending self generated packets [24]. Similarly, sink hole attack [25], an advanced version of the black hole attack, is also easily detectable by looking at the passive acknowledgment. Likewise, the selective forwarding attack [26] and gray-hole attack [27] can also be eliminated with the aid of above mentioned approach.

Based on these successful and unsuccessful interactions node x can calculate the trust value of node y in following fashion:

$$T_{x,y} = \left[100 \left(\frac{S_{x,y}}{S_{x,y} + U_{x,y}} \right) \left(1 - \frac{1}{S_{x,y} + 1} \right) \right] \tag{2}$$

where [.] is the nearest integer function, $S_{x,y}$ is the total number of successful interactions of node x with y during time δt, and $U_{x,y}$ is the total number of unsuccessful interactions of node x with y during time δt. After calculating trust value, a node will quantize trust into three states as follows:

$$Mp(T_{x,y}) = \left\{ \begin{array}{cc} \text{trustworthy} & 100 - f \leq T_{x,y} \leq 100 \\ \text{uncertain} & 50 - g \leq T_{x,y} < 100 - f \\ \text{untrustworthy} & 0 \leq T_{x,y} < 50 - g \end{array} \right\}. \tag{3}$$

where, f represents half of the average values of all trustworthy nodes and g represents one-third of the average values of all untrustworthy nodes. Both f and g are calculated as follows:

$$f_{j+1} = \left\{ \begin{array}{cc} \left[\frac{1}{2} \left(\frac{\sum_{i \in R_x} T_{x,i}}{|R_x|} \right) \right] & 0 < |R_x| \leq n - 1 \\ f_j & |R_x| = 0 \end{array} \right. \tag{4}$$

$$g_{j+1} = \begin{cases} \left[\frac{1}{3}\left(\frac{\sum_{i \in M_x} T_{x,i}}{|M_x|}\right)\right] & 0 < |M_x| \leq n-1 \\ g_j & |M_x| = 0 \end{cases} \quad (5)$$

where [.] is the nearest integer function, R_x represents the set of trustworthy nodes for node x, M_x the set of untrustworthy nodes for node x, and n is the total number of nodes that contains trustworthy, untrustworthy and uncertain nodes. The initial trust values of all nodes are 50, which represents the uncertain state. Initially f and g are equal to 25 and 17 respectively, although other values could also be used by keeping the following constraint intact: $f_i - g_i \geq 1$, which is necessary for keeping the uncertain zone between a trusted and untrustworthy zone. The values of f and g are adaptive. During the steady-state operation, these values can change with every passing unit of time which creates dynamic trust boundaries. At any stage, when $|R_x|$ or $|M_x|$ becomes zero, the value of f_{j+1} or g_{j+1} remains the same as the previous values (f_j and g_j). The nodes whose values are above $100 - f$ will be declared as trustworthy nodes (Equation 3), and nodes whose values are lower than $50 - g$ will be consider as untrustworthy nodes (Equation 3). After each passage of time, Δt, nodes will recalculate the values of f and g. This trust calculation procedure will continue in this fashion.

The time window length (Δt) could be made shorter or longer based on the network analysis scenarios. If Δt is too short, then the calculated trust value may not reflect the reliable behavior. On the other hand, if it is too long, then it will consume too much memory to store the interaction record at the sensor node. Therefore, various parameters can be used to adjust the length of Δt.

4.2. Identity, Route, and Location Privacy (IRL)

Our proposed identity, route and location privacy scheme works in two phases. The first is neighbor node state initialization phase, and the second is routing phase.

Route Privacy: In initialization phase, let the node i have m neighboring nodes in which t nodes are trusted. So, $0 \leq t \leq m$ and $M(t) = M(t_F) \cup M(t_{B_r}) \cup M(t_{B_l}) \cup M(t_{B_m})$. Here $M(t_F)$, $M(t_{B_r})$, $M(t_{B_l})$, and $M(t_{B_m})$ represent the set of trusted nodes that are in the forward, right backward, left backward, and middle backward directions, respectively. These neighbor sets ($M(t_F)$, $M(t_{B_r})$, $M(t_{B_l})$, and $M(t_{B_m})$) are initialized and updated whenever a change occur in neighborhood. For example, the entrance of a new node, change of a trust value, *etc*.

Whenever a node needs to forward a packet, the routing phase (Algorithm 1 for source node and Algorithm 2 for intermediate node) of IRL algorithm is called.

Whenever a source node (Algorithm 1) wants to forwards the packet, it will first check the availability of the trusted neighboring nodes in its forward direction set $M(t_F)$ (Line 2). If trusted nodes exists then it will randomly select one node as a next hop (Line 3) from the set $M(t_F)$ and forward the packet towards it (Lines 13:21). If there is no trusted node in its forward direction, then the source node will check the availability of a trusted node in the right ($M(t_{B_r})$) and left ($M(t_{B_l})$) backward sets. If the trusted nodes are available then the source node will randomly select one node as a next hop (Line 3) from these sets and forward the packet towards it (Lines 13:21). If the trusted node does not exist in these sets either, then the source node will randomly select (Line 8) one trusted node from the backward middle set ($M(t_{B_m})$) and forward the packet towards it (Lines 13:21). If there are no trusted nodes available in all of the sets then the packet will be dropped (Line 9:10).

Algorithm 1 IRL - Routing at Source Node.

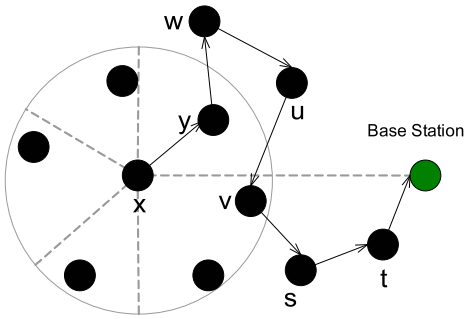
1: $prev_{hop} \leftarrow \emptyset$; $next_{hop} \leftarrow \emptyset$;
2: **if** $M(t_F) \neq \emptyset$ **then**
3: $next_{hop}(k)$ = Rand($M(t_F)$);
4: **else**
5: **if** $M(t_{B_r}) \cup M(t_{B_l}) \neq \emptyset$ **then**
6: $next_{hop}(k)$ = Rand($M(t_{B_r}) \cup M(t_{B_l})$);
7: **else if** $M(t_{B_m}) \neq \emptyset$ **then**
8: $next_{hop}(k)$ = Rand($M(t_{B_m})$);
9: **else**
10: Drop packet and Exit;
11: **end if**
12: **end if**
13: Set $prev_{hop} = my_{id}$;
14: Form pkt $p = \{prev_{hop}, next_{hop}, seqID, payload\}$;
15: Create Signature and save in buffer;
16: Forward packet to $next_{hop}$;
17: Set timer $\Delta t = \frac{D}{d_{next_{hop}}} \times p_t$;
18: **while** $\Delta t = true$ **do**
19: Signature remains in buffer;
20: **end while**
21: Signature removed from buffer;

When an intermediate node (Algorithm 2) receives the packet (either from the source node or from another en-route node), it will first check whether the packet is new or old (Line 3). If it is new, then the node will first check the availability of the trusted node from the forward direction set (M_F) excluding the $prev_{hop}$ node if it belongs to forward set (Line 13). If trusted nodes exists in the forward set then the node will randomly select any one trusted node as a next hop (Line 14) and forward the packet towards it (Line 45). If there is no trusted node available in the forward direction, then it will check to which set the sender of the packet belongs to. For example, If the packet, forwarded by a node, belongs to the right backward set (Line 16), then it will first check whether the left or middle backward sets contain any trusted nodes (Lines 17:18). If so, it will randomly select one node from those sets (Line 19) and forward the packet towards it (Line 45). If there is no trusted node in those two sets, then the node will randomly select a trusted node from the right backward set ($M(t_{B_r})$) excluding the one from which the node received the current packet (Lines 20:21) and forward the packet towards it (Line 45). Similar operations will be performed, if the packet, forwarded by a node, belongs to the left (Lines 25:33) and middle backward or forward (Lines 34:43) sets. An example IRL routing scenario is shown in Figure 3.

Algorithm 2 IRL - Routing at Intermediate Node.

1: $next_{hop} \leftarrow \emptyset$;
2: $M_{temp} = \emptyset$
3: **if** Signature of new packet already exists in buffer **then**
4: $M_{temp} = \{M_{temp}\} + LasttimePrev_{hop}$
5: $M_{temp} = \{M_{temp}\} + LasttimeNext_{hop}$
6: Set $counter$ = $timesReceviedBefore$ + 1;
7: Remove signature from buffer;
8: **if** $counter$ = 3 **then**
9: Drop packet and exit;
10: **end if**
11: **end if**
12: $M_{temp} = \{M_{temp}\} + prev_{hop}$
13: **if** $(M(t_F) - \{M(t_F) \bigcap M_{temp}\}) \neq \emptyset$ **then**
14: $next_{hop}(k)$ = Rand$(M(t_F) - \{M(t_F) \bigcap M_{temp}\})$;
15: **else**
16: **if** packet came from B_r **then**
17: $M_{temp1} = M(t_{B_l}) \cup M(t_{B_m})$
18: **if** $M_{temp1} \neq \emptyset$ **then**
19: $next_{hop}(k)$ = Rand(M_{temp1});
20: **else if** $M(t_{B_r}) \neq \emptyset$ **then**
21: $next_{hop}(k)$ = Rand$(M(t_{B_r}) - \{M(t_{B_r}) \bigcap M_{temp}\})$;
22: **else**
23: Drop packet and Exit;
24: **end if**
25: **else if** packet came from B_l **then**
26: $M_{temp2} = M(t_{B_r}) \cup M(t_{B_m})$
27: **if** $M_{temp2} \neq \emptyset$ **then**
28: $next_{hop}(k)$ = Rand$(M_{temp2} - \{M_{temp2} \bigcap M_{temp}\})$;
29: **else if** $M(t_{B_l}) \neq \emptyset$ **then**
30: $next_{hop}(k)$ = Rand$(M(t_{B_l}) - \{M(t_{B_l}) \bigcap M_{temp}\})$;
31: **else**
32: Drop packet and Exit;
33: **end if**
34: **else**
35: $M_{temp3} = M(t_{B_r}) \cup M(t_{B_l})$
36: **if** $M_{temp3} \neq \emptyset$ **then**
37: $next_{hop}(k)$ = Rand$(M_{temp3} - \{M_{temp3} \bigcap M_{temp}\})$;
38: **else if** $M(t_{B_m}) \neq \emptyset$ **then**
39: $next_{hop}(k)$ = Rand$(M(t_{B_m}) - \{M(t_{B_m}) \bigcap M_{temp}\})$;
40: **else**
41: Drop packet and Exit;
42: **end if**
43: **end if**
44: **end if**
45: Rest is same as Algorithm 1 from lines 13:21;

Figure 3. Sample routing scenario of IRL scheme.

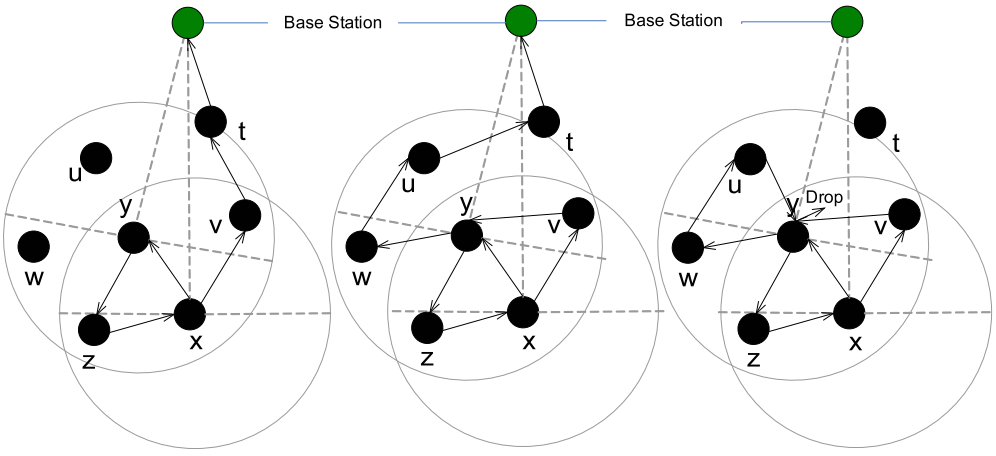


This routing strategy may result in the creation of a cycle (loop). However, due to the randomness in the selection of the next-hop and the presence of the different four direction sets, the probability of creation of any cycle is very low. Nevertheless, in order to fully avoid the occurrence of the cycles, each node (prior to forwarding of a packet) will save the signature of the packet in the buffer for the δt time, that is:

$$\delta t = 2\left(\frac{D}{d} \times p_t\right) \tag{6}$$

where D is the distance between the forwarding node and the base station, d is the distance between the forwarding node and the next hop, and p_t is the propagation transfer time between the forwarding node and the next hop. This signature consists of two fields: (1) sequence number of the packet, and (2) the payload. The potential of the signature to compare and identify the same packet is detailed in the later section. Corresponding to this signature, three more fields are also stored in the buffer: (1) previous hop identity, (2) next hop identity where the packet is forwarded, and (3) counter, that tells how many times the same packet is received by the node. This information will later be used to get rid of any cycle. The size of the buffer is mainly dependent on the network traffic conditions. However, it is expected to be low due because the sensor nodes sent data either in periodic intervals or upon the occurrence of some event.

Figure 4. Three sample cycle detection and prevention scenarios.



If the node received the packet whose signature exists in the buffer (Algorithm 2, Lines 3:11), then including the previous hop node (Line 12), two other nodes will also be excluded from the selection of

the next hop process: 1) the node from which last time the packet was received (Line 4) and 2) the node from which last time the packet was forwarded (line 5). If the same packet is received three times by the same node (Line 8) then the packet will be dropped (Line 9). Three sample scenarios of the loop creation, detection and prevention are shown in Figure 4. Creation of loops and traversing of the packets in the backward direction is not a completely negative effect. Rather, it provides positive effects in terms of strengthening the route and source location privacy, because these effects will helps to increase the safety period [3], which is the time for an adversary to reach at the source node.

Identity Privacy: Whenever a node receives the packet p from the source node or en-route node then the receiving node will replace the previous hop's identity $prev_{hop}$ contained in the packet with its own (Algorithm 1, Line 13). After that, the node will get the next forwarding node $next_{hop}$ (as described earlier) and update the header of the packet $p = \{prev_{hop}, next_{hop}, payload\}$ (Line 14). After modification of the two header fields, the node will forward the packet (Line 16). In this way, all the intermediate forwarding nodes replace the source and next hop's identity contained in the packet p. This process will go on until the packet reaches the base station.

Location Privacy: The neighboring nodes which are in each other's radio range can easily approximate the location of each other by measuring the received signal strength and the angle of arrival [28]. If the adversary is within the range of the source node, then adversary can easily estimate the location of the source. Once the packet has crossed the radio range of the original source node, then becomes very difficult for an attacker to estimate the location of the node either in terms of the physical distance or in terms of the number of hops of an original source node. The main reason for this is that the path selection is random and packets are forwarded by only trusted nodes which only contain the information of the last and the next hop.

4.3. Reliable Identity, Route, and Location Privacy (r-IRL)

It is also possible that some applications require more reliability in terms of packet reach-ability; and the packet could be dropped due to either network congestion or malicious behavior of an en-route node. Thus, in order to achieve more reliability, the packet should be forwarded from multiple paths simultaneously, which will give trustworthiness in the sense that at least the packet should reach the base station by any one of the paths, although, this may increase some communication overhead.

Our reliable IRL (r-IRL) algorithm is the extended version of our proposed IRL algorithm, in which we introduce one more parameter, reliability r. The source node i will multi-cast a packet to all r randomly selected neighboring trusted nodes that are in the forward direction. If there are no adequate trusted nodes present in the forward direction, then it will select the remaining trusted nodes from the backward direction. The rest of the mechanism of the r-IRL algorithm is the same as the IRL algorithm.

4.4. Data Privacy

The payload contains the identity of the source node (ID_x) and the actual data (d). Identity is encrypted with the public key (k_{bs}^{+}) of the base station and data is encrypted with the secret key ($k_{x,bs}$) shared between the sender node and the BS. Both are appended with the payload as shown below:

$$payload = [E(ID_x, k_{bs}^{+}), E(d, k_{x,bs})]$$

If we assume that the adversary knows the range of identities assigned to the sensor nodes, public key of the base station and information about cipher algorithm used in the network, an adversary can then successfully obtain the identity of the source by performing simple brute-force search attack [29] by comparing the pattern of encrypted identity with a known range of identities. Therefore in order to provide protection against brute-force search attack, we append a random number (R_n) (equivalent to the size of identity) with the identity of a node and then perform encryption. Now the payload is:

$$payload = [E(ID_x||R_n, k_{bs}^{+}), E(d, k_{x,bs})]$$

where $||$ is the append operation. Inclusion of random number may introduce additional computational overhead. However, the amount of overhead is mainly dependent on random number generation technique. Recently, very nice random generation techniques have been specially designed for low power sensor networks, such as [30, 31]. These techniques could be used to generate random number for each packet. Also, overall computational overhead is dependent on the number of packets generated by the sensor nodes. Mostly, sensor nodes are event driven or query driven [32]. Therefore, amount of traffic is usually kept low as compared to traditional networks.

Our proposed data privacy approach provides several benefits. Firstly, data secrecy is achieved in the presence of identity anonymity. This feature is not available in earlier proposed privacy schemes. Secondly, the base station will receive both the identity of the actual source node and message authentication. If the packet has been successfully decrypted with the shared secret key, it means that packet is received from genuine sensor node.

5. Analysis and Evaluation

5.1. Security Resiliency Analysis

Suppose we have an adversary $\mathcal{A}$ who strives to defeat our privacy protocols and guess the original source node. We will distinguish between two kinds of nodes. A source node is the original sender of a packet q and a forwarding node is the node that forwards a packet to another node until it reaches the destination. Hence the source node is also a forwarding node. The adversary's goal is to find out the source node. This analysis assumes that we are using IRL algorithm including our proposed data privacy mechanism. So if the adversary sees a packet, it will trivially know the identity of the last forwarding node (which could possibly be the sender node).

We will deal with separate cases. Case 1 is when the adversary is close to the base station and can eavesdrop on any packet received by the base station. Case 2 deals with the case when the adversary can see any packet within the radio range of a particular node. Case 3 extends this into two or more nodes.

An adversary will try to solve the following problem: Given a packet q and a subset of nodes N', find out the sender node s. In other words, the algorithm for the adversary takes two inputs and outputs a node s'; Namely $\mathcal{A}(q, N') = s'$. If $s' = s$, the adversary succeeds in defeating our protocol. We have to find: $\Pr[\mathcal{A}(q, N') = s]$, which is the probability for an adversary to find out the sender node. Our assumption is that, from an adversarial perspective, all nodes are equally likely to be senders of a packet. This does not necessarily mean that the network traffic is uniformly distributed. Notice that if the adversary knows beforehand which nodes are more likely to send packets, then no privacy preserving method can

prevent the adversary from guessing the most likely senders, since this constitutes the adversary's a priori knowledge.

Notations and definitions: Denote a generic node by m. The set of neighbors of m is denoted by N_m, which also includes m itself. The number of forward and backward nodes of m is denoted by m_f and m_b respectively. If a node a is a backward node of m, then we denote it as $a \rightarrow m$. We say that a node a is in the backward set of node m, if $a \rightarrow a_1 \rightarrow \ldots a_r \rightarrow m$, for some nodes $a_1, \ldots a_r$ where $r \geq 0$. For compact notation we will denote this as $a \rightarrow^r m$, if the IDs of the intermediate nodes are not significant. We will also use the notation $\rightarrow^r m$ to denote a generic node, who is r links (hops) away from m. Define the backward set C_m of m as $C_m = \{a | a \rightarrow^r m, r \geq 0\}$, that is the set of all the possible nodes such that they have a forward link to m. Denote the base station as B. It will also be seen as another node. Let the total number of nodes in the network excluding the base station be N. We will use the term "adversary is in possession of a node" to indicate that the adversary can passively listen to any communication within the radio range of that node.

Claim 1: Suppose $\mathcal{A}$ is in possession of B. Let B_b be the number of backward nodes of the base station (nodes one hop away from the base station). Then for any packet q received by B and for large enough N:

$$\Pr\left[\mathcal{A}\left(q, N\right) = s\right] \approx \frac{B_b + 1}{N} \tag{7}$$

Proof: The adversary can always know the ID of the last forwarding node. Let B_b be the number of backward nodes to the base station. The packet could only have come from one of the nodes in $N_B - \{B\}$ (which only contains backward nodes to B). Since the nodes are just a hop away from the BS, so they will not send the packet to another node. Hence for large N we have:

$$\begin{aligned}
\Pr\left[\mathcal{A}\left(q, N\right) = s\right] &= \Pr\left[\mathcal{A}\left(q, N\right) = s | s \in N_B - \{B\}\right] \times \Pr\left[s \in N_B - \{B\}\right] + \\
&\quad \Pr\left[\mathcal{A}\left(q, N\right) = s | s \notin N_B - \{B\}\right] \Pr\left[s \notin N_B - \{B\}\right] \\
&= 1 \cdot \frac{B_b}{N} + \frac{1}{N - B_b - 1}\left(1 - \frac{B_b}{N}\right) \\
&\approx \frac{B_b}{N} + \frac{1}{N - B_b}\left(1 - \frac{B_b}{N}\right) = \frac{B_b + 1}{N}
\end{aligned}$$

■

Now let us assume that $\mathcal{A}$ is in possession of a node m in the network. The following probability estimate gives an upper bound of the probability of success of the adversary. It is an upper bound since it does not include the possibility of a packet sent backwards. When a packet is sent backwards over one or many hops, the probability of success of the adversary decreases since there would be more possible nodes. Thus in this scenario our result would be like an upper bound on the adversary's limitations.

Claim 2: Suppose $\mathcal{A}$ is in possession of a node m. Let $c = |C_{\rightarrow^2 m}|$ denote the number of backward nodes in backward set $C_{\rightarrow^2 m}$ of some node $\rightarrow^2 m$. Then,

$$\Pr\left[\mathcal{A}\left(q, N\right) = s\right] \leq \frac{m_f + m_b + 1}{N} + \frac{1}{c+1}\left(1 - \frac{m_f + m_b + 1}{N}\right) \tag{8}$$

Proof: Since the adversary is in possession of a node m, it knows its backward and forward nodes. Furthermore, if any of these nodes including the node m itself is the sender of a packet q, then the

adversary will know. This is true since the adversary can see all incoming packets to the node m and to its neighbor nodes (the forward and the backward nodes). Thus it can see if the payload of q is not equal to the payload of any q' being received by these nodes in a given interval of time. If this is the case, then the adversary will know the sender.

Now if none of the nodes in N_m are the senders, then the packet was forwarded by a node i that is two hops away from m. The adversary knows the ID of that node through the packet q. Thus the adversary makes a list of all the possible backward nodes in the backward set of i. Let that number be denoted by c. Notice that node i could also be the possible sender. Hence the total number of possible senders would be $c + 1$. We have:

$$\begin{aligned}\Pr\left[\mathcal{A}\left(q,N\right)=s\right] &= \Pr\left[\mathcal{A}\left(q,N\right)=s|s\in N_m\right]\Pr\left[s\in N_m\right]+\\ &\quad \Pr\left[\mathcal{A}\left(q,N\right)=s|s\notin N_m\right]\Pr\left[s\notin N_m\right]\\ &\leq \frac{m_f+m_b+1}{N}+\frac{1}{c+1}\left(1-\frac{m_f+m_b+1}{N}\right)\end{aligned}$$

■

Now, suppose the adversary is in possession of two nodes at the same time; m_1 and m_2. We can safely assume that $N_{m_1}\cap N_{m_2}=\varphi$, since it would be more advantageous to the adversary to cover nodes with non-overlapping radio ranges. The adversary will always know whenever any node in N_{m_1} or N_{m_2} is the sender of a packet. How about the case when they are not the senders? There could be two possible cases: without loss of generality, first assume that $m_2\in C_{m_1}$. If the packet q was received by some node in N_{m_1} and was received by some node in N_{m_2} before, then the adversary had already checked it when the packet was sent to a node in N_{m_1}. Thus the adversary need only check packets received in N_{m_1} that were not received by N_{m_2}. In this case, the sender cannot be in N_{m_2}. In any case, the adversary has to find out the backward sets of $\rightarrow^2 m_1$ or $\rightarrow^2 m_2$, depending on where the packet was received. Since, in the adversary's knowledge, all nodes are equally likely to be senders, the probability of a packet being received at the two sets is the same. In case $m_2\notin C_{m_1}$, then the adversary has no real advantage except that it can see packets at two disjoint locations in the network. Thus we only state the case when $m_2\in C_{m_1}$. We have the following result:

Claim 3: Suppose the adversary is in possession of two nodes m_1 and m_2. Assume further that $m_2\in C_{m_1}$. Let $c_1=|C_{\rightarrow^2 m_1}|$ and $c_2=|C_{\rightarrow^2 m_2}|$ then:

$$\Pr\left[\mathcal{A}\left(q,N\right)=s\right]=\frac{\left|N_{m_1}\right|+\left|N_{m_2}\right|}{N}+\frac{1}{2}\left(\frac{1}{c_1+1-\left|N_{m_2}\right|}+\frac{1}{c_2+1}\right)\left(1-\frac{\left|N_{m_1}\right|+\left|N_{m_2}\right|}{N}\right) \tag{9}$$

In general, we have:

Claim 4: Let us assume that A is in possession of k nodes $m_k\rightarrow^{r_1}\cdots\rightarrow^{r_{k-2}} m_2\rightarrow^{r_{k-1}} m_1$ and let m_f and m_b denote the average number of forward and backward nodes averaged over all the k nodes. Let $t=m_f+m_b+1$. Let for $1\leq i\leq k$, $c_i=|C_{\rightarrow^2 m_i}|$, then:

$$\Pr\left[\mathcal{A}\left(q,N\right)=s\right]=\frac{kt}{N}+\frac{1}{k}\left(\frac{1}{c_1+1-(k-1)t}+\frac{1}{c_2+1-(k-2)t}\cdots+\frac{1}{c_k+1}\right)\left(1-\frac{kt}{N}\right) \tag{10}$$

Observations: The probability is lowest when the adversary is actually at the base station. If the adversary has more nodes in possession, the probability increases linearly, with more success rate when

the nodes are actually connected. This also shows that if a packet originates from any node that does not have a backward node, the adversary will always know the sender. This drawback can be avoided by requiring all nodes to have backward nodes. In other words, avoid a tree topology.

The above security resiliency analysis description is for route and location privacy. The security strength of identity and data privacy is mainly dependent on the encryption schemes. If encryption scheme is strong then we can achieve stronger identity and data privacy. If encryption scheme is weak then we have weak identity and data privacy.

5.2. Memory Consumption Analysis

Each sensor node needs to maintain one table that contains the list of neighboring nodes, their direction and their trust states as shown in Table 2. Node identity can be represent in two bytes [15, 33]. Four sets of directions can be easily represented in 2 bits. Trust calculation is based on time-based past interaction only. Therefore, the total size required to calculated trust value is $4\Delta t$ bytes [22]. Here, Δt represents size of time window and 4 bytes are required to store number of successful (2 bytes) and unsuccessful (2 bytes) interactions. Trust value can be represented in one byte. Therefore the size of each record is $3.25 + 4\Delta t$ bytes ($26 + 32\Delta t$ bits). If we assume that the node has M neighboring nodes then the total size of the table will be $M(26 + 32\Delta t)$ bits.

Table 2. Neighbor list table at sensor node.

Neighbor nodeID (Integer)	Direction	Past interactions based on time window						Trust value
		Successful interactions ($S_{x,y}$)			Unsuccessful interactions ($U_{x,y}$)			
1	F (00)	10	...	5	4	...	1	90
2	B_R (01)	2	...	4	8	...	2	25
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
M	B_L(11)	5	...	7	0	..	3	70

In order to achieve data privacy in the presence of identity anonymity, our proposed scheme uses two keys: one Public key of the base station k_{bs}^{+} and other is shared secret key k_{x-bs}. Therefore, total memory required at the sensor node for our proposed scheme is: $M(26 + 32\Delta t) + k_{bs}^{+} + k_{x,bs}$.

Table 3 shows the memory requirement of various privacy schemes, in which M represents the neighborhood size, K represents pseudonym space, $4\Delta t$ represents size of time window, and N is the total number of nodes in the network.

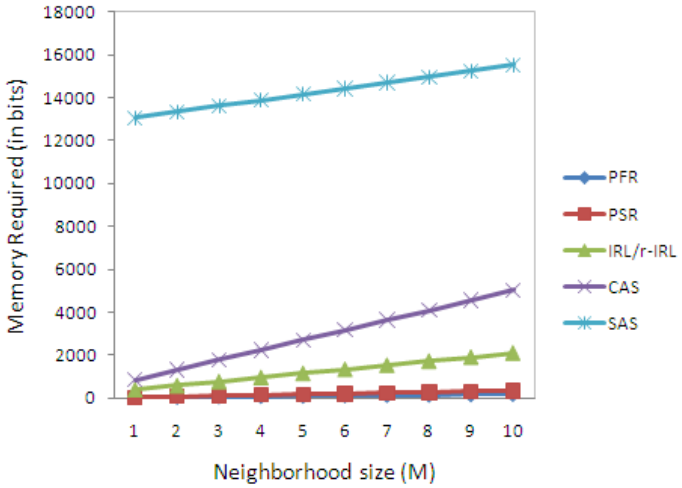
Table 3. Memory requirement in bits.

PFR [3]	(16+1)M bits
PSR [4]	(16+16+1)M bits
SAS [5]	K(4M+2N)+16M bits
CAS [5]	K(6+7M)+16M bits
IRL / r-IRL	$M(26 + 32\Delta t) + k_{bs}^{+} + k_{x,bs}$bits

In the Phantom Flood Routing (PFR) [3] scheme, each sensor node needs to maintain the list of neighbor nodes and these neighbor nodes are divided into two sets. Here we assume that identity of a node is represented by two bytes, and set is distinguished by a single bit. So the total memory required by each node in the PFR scheme is (16+1)M bits. In the Phantom Single-path Routing (PSR) [4] scheme, each node maintain the list of neighbor nodes, hop count (2 bytes), and set identification (1 bit). Therefore, the total memory required by each node in the PSR scheme is (16+16+1)M bits. In the SAS scheme, each node needs K(4M+2N)+16M bits of memory. Here M represents the neighborhood size, K represents pseudonym space and N is the total number of nodes in the network. For the CAS scheme, each node requires K(6+7M)+16M bits of memory. (See [5] for more details about the SAS and CAS schemes.)

Let us assume that the sensor node has ten neighbor nodes, then the total memory required by the sensor node by the PFR, PSR, IRL, CAS and SAS is 21.25, 41.25, 260.5, 628 and 1940 bytes respectively, as shown in Figure 5.

Figure 5. Memory consumption analysis: N= 100; K=8 bytes; Δt = 5; k_{bs}^{+} = 20 bytes; k_{x-bs} = 8 bytes.



Additionally, cycle prevention strategy (Section 4.2) requires some short term memory to store signature of the packet for short period of time (δt). In our proposed schemes, signature of the packet comprises of six fields: 1) Sequence number (2 bytes), 2) previous hop identity (2 bytes), 3) next hop identity (2 bytes), 4) payload (variable size), 5) counter (2 bits), and 6) δt time (4 bytes). So, for each packet sensor node requires $10.25 + Size$(payload) bytes of memory. The packet signature will be

removed from the buffer after δt time. For example, sensor node x received 20 packets of equal size of payload (e.g., 10 bytes). Then, the total memory required by the sensor node is $20 \times (10.25 + 10)$ = 405 bytes. This additional overhead does not make sensor nodes overloaded because of following reasons:

1. Generally, wireless sensor networks are event driven [34–36] or sensor nodes generate packets in periodic intervals [37, 38]. Therefore, the amount of overall traffic usually remains low.
2. In our proposed schemes, packets always follow different routes. Therefore, the probability of a single node to be overloaded is very low.

Let us assume that the amount of traffic is very high and single sensor node needs to store large amount of packets at a time. Then in order to reduce the size of memory, we can use the technique of signature generation code [39]. This technique allows us to represent single signature code in few bytes. However, this technique is based on Bloom filters [40] that require the computation of multiple hash values. This may increase the computational cost. Therefore, we need to trade off between memory and computation cost.

5.3. Energy Consumption Analysis

In this section, we will show the efficiency of our routing strategies with existing schemes. Energy is computed based on the communication overhead (including transmission and reception cost, path length) introduced by our proposed routing protocols and compared it with other existing schemes.

Table 4. Simulation parameters.

Network specific	Number of nodes	300
	Distance b/w nodes	50 units
	Mobility of nodes	zero
Node specific	Sensor node's Initial battery	1×10^6J
	Power consumption for trans.	1.6W
	Power consumption for recv.	1.2 W
	Idle power consumption	1.15W
	Carrier sense threshold	$3.65e^{-10}$W
	Receive power threshold	$1.55e^{-11}$W
	Frequency	$9.14e^8$
	Trans. & Recv. antenna gain	1.0
Protocol & Application specific	Application	CBR
	Reliability param. r for r-IRL	3
	h_{walk} param. for PFR & PSR	10

We have implemented our IRL and r-IRL routing schemes on Sensor Network Simulator and Emulator (SENSE) [41]. At the application layer we used constant bit rate component (CBR) that generate

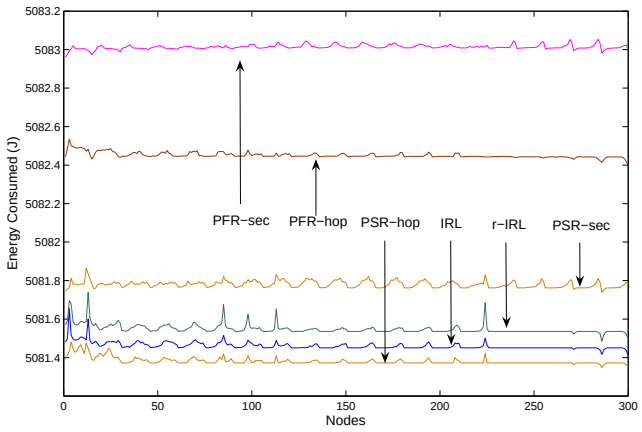
constant traffic during simulation between randomly selected source node(s) and the base station. For the simplicity, we assumed that both sensor nodes and the base station are static. Network consists of 300 sensor nodes that are organized into 15 by 20 grid manner. Other simulation parameters are given in Table 4.

We have compared our proposed IRL and r-IRL algorithms with the four variations of phantom routing schemes [3, 4] that are:

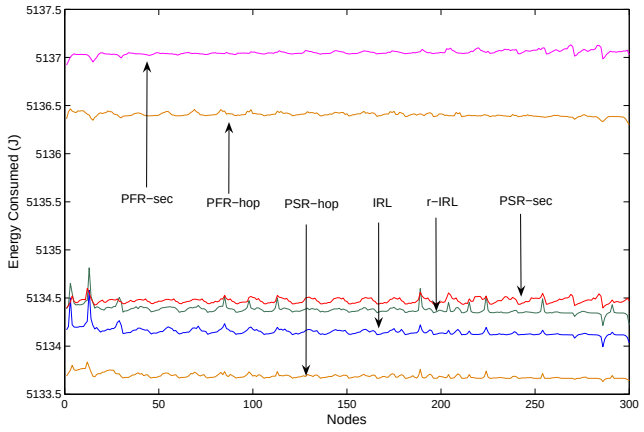
1. Phantom single path routing scheme with hop-based approach (PSR-hop).
2. Phantom single path routing scheme with sector-based approach (PSR-sec).
3. Phantom flood routing scheme with hop-based approach (PFR-hop).
4. Phantom flood routing scheme with sector-based approach (PFR-sec).

We did not compared our schemes with the SAS and CAS [5] schemes because the authors did not propose any routing strategy.

Figure 6. Energy consumption analysis: simulation time: 5,000.



(a) Source nodes:5



(b) Source nodes:10

The energy consumption analysis with different scenarios are shown in Figure 6. For the r-IRL scheme we select $r = 3$, which means a single packet will reach the destination via three different routes simultaneously. For phantom routing schemes, we select parameter h_{walk}=10 (as recommended

in [3]). Figure 6 clearly indicates that, the IRL and r-IRL schemes consume less energy as compared to the PSR-sec, PFR-hop and PFR-sec schemes but slightly consume higher energy as compared to the PSR-hop scheme. This is due to the fact that the IRL and r-IRL algorithms provides more path diversity and packets sometimes took longer paths.

5.4. Path Diversity Analysis

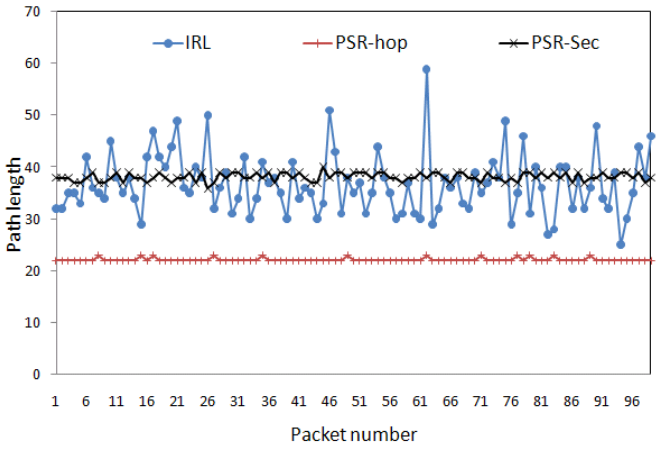
Strength of route privacy is dependent on path diversity. High path diversity provides strong route privacy and low path diversity provides weak rout privacy. Path diversity can be categorized into two types.

1. Length variation: Path could be long or short and mainly dependent on routing scheme. For example, packets always reach to the destination via shortest path. In this scheme, packets may reach to the destination via longer path if any node is not working properly within the shortest available path. With respect to the route privacy, length variation provides minimum route privacy. If we have longer paths, then it will increase time for an adversary to find out actual source node or vice versa. So, the longer path increases safety time.

2. Path variation: Each packet may follow different route. It is also dependent on routing strategy. For example, routing scheme make decision about next hop based on the energy level of neighboring nodes. With this approach, one can achieve limited path variation. With respect to the route privacy, if we have more path variation, then it will become clueless for an adversary to guess from where next packet will come.

Our proposed routing strategies (IRL and r-IRL) have both features. Because of the concept of *direction* (Section 4.1), proposed schemes provide more length variation and because of the *randomness* (Section 4.2) proposed schemes provide high path variation. Incorporation of both features offer high path diversity.

Figure 7. Path diversity of privacy schemes.



In order to analyze the path diversity behavior, we have organized 300 sensor nodes in a 10 by 30 grid manner. The rest of simulation parameters are given in Table 4. In the simulation, a single source node (ID: 224) generates 100 data packets for the base station. Figure 7 shows the path diversity (in terms

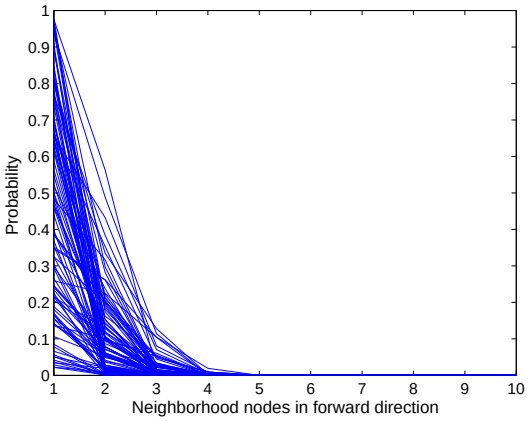
of path length) of the IRL, PSR-hop and PSR-sec schemes. The average path taken by the PSR-hop, IRL and PSR-sec is 22.12, 36.81 and 38.17, respectively. It indicates that the IRL scheme incurs more delay as compared with the PSR-hop scheme and less delay as compared with the PSR-sec scheme. This figure also indicates that the IRL scheme has more path variation as compared with the other schemes, which creates more difficulties for the adversary to trace back the source from the captured packets.

Figure 7 also shows that some packets took longer paths in the IRL scheme as compared with others. This is due to the fact that the source or en-route node did not find any trusted node in its forward direction, so the packet is relayed back in the backward direction. If we assume that each node has p probability to be trusted and all probabilities are independent of each other, then the total probability P_b for a node i to relay the packet in the backward direction is:

$$P_b(i) = \prod_{k=1}^{m_f} (1 - p_k) \tag{11}$$

where m_f represents the number of nodes in the forward direction. Figure 8 shows the result of 100 simulation runs in which we have assumed that each node has equal probability to be trusted and un-trusted. It shows that, as the neighborhood size increases, the probability of the packet to move in the backward direction decreases sharply.

Figure 8. Probability of a packet to move in the backward direction.



5.5. Discussion

From the memory, energy and path diversity analysis, we see that our solution is optimal especially with respect to the PSR-hop scheme. However, at a modest cost of memory and energy, our solutions provide full network level privacy as compared with the other existing schemes. This cost is justifiable because we have additionally achieved trustworthiness and reliability (in terms of packet reach-ability). With this level of resource consumption, our solutions can easily be used on real sensor nodes, for example, MICA2 sensor node has ATMega 128L micro controller (8 MHz @ 8 MIPS), 128 Kbyte program flash memory, 512 Kbyte measurement (serial) flash, and 4 Kbyte EEPROM [42].

6. Conclusions and Future work

Existing privacy schemes of WSNs only provides partial network level privacy. Providing full network level privacy is a critical and challenging issue due to the constraints imposed by the sensor nodes (e.g.,

energy, memory and computation power), sensor network (e.g., mobility and topology) and QoS issues (e.g., packet reach-ability and timeliness). Therefore, in this paper we proposed the first full network level privacy solution that is composed of two new identity, route and location privacy algorithms and data privacy mechanism. Our solutions provide additional trustworthiness and reliability at modest cost of energy and memory. We also proved analytically that our solutions provides protection against an adversary who is capable of performing privacy disclosure attacks such as eavesdropping and hop-by-hop trace backing.

In our future work, we will evaluate our proposed schemes from the perspective of computation cost that is required to perform encryption and random number generation.

Acknowledgments

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2009-(C1090-0902-0002)). This work was also supported by KOSEF grant funded by the Korean government (MEST, No.2008-1342), and was supported by Basic Science Research Program funded by National Research Foundation (2009-0076798).

References and Notes

1. Xi, Y.; Schwiebert, L.; Shi, W. Preserving Source Location Privacy in Monitoring-Based Wireless Sensor Networks. In *Proceedings of Parallel and Distributed Processing Symposium (IPDPS 2006)*, Rhodes Island, Greece, 2006.
2. Habitat monitoring on Great Duck Island (Maine, USA), 2002. Available online: http://ucberkeley.citris-uc.org/research/projects/great_duck_island (accessed on 21 August, 2009).
3. 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*, Washington, DC, WA, USA, 2004; pp. 88–93.
4. Kamat, P.; Zhang, Y.; Trappe, W.; Ozturk, C. Enhancing Source-Location Privacy in Sensor Network Routing. In *Proceedings of the 25th IEEE International conference on Distributed Computing Systems*, Columbus, OH, USA, 2005; pp. 599–608.
5. Misra, S.; Xue, G. Efficient Anonymity Schemes for Clustered Wireless Sensor Networks. *Int. J. Sens. Netw.* **2006**, *1*, 50–63.
6. Wood, A.D.; Fang, L.; Stankovic, J.A.; He, T. SIGF: A Family of Configurable, Secure Routing Protocols for Wireless Sensor Networks. In *Proceedings of the 4th ACM Workshop on Security of Ad Hoc and Sensor Networks*, Alexandria, VA, USA, 2006; pp. 35–48.
7. Ouyang, Y.; Le, Z.; Chen, G.; Ford, J.; Makedon, F. Entrapping Adversaries for Source Protection in Sensor Networks. In *Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)*, Niagara-Falls, Buffalo, NY, USA, 2006; pp. 23–34.
8. Zorzi, M.; Rao, R.R. Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Multihop Performance. *IEEE Tran. Mob. Comput.* **2003**, *2*, 337–348.

9. Zorzi, M.; Rao, R.R. Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance. *IEEE Tran. Mob. Comput.* **2003**, *2*, 349–365.
10. Capone, A.; Pizziniaco, L.; Filippini, I.; de la Fuente, M.G. SiFT: An Efficient Method for Trajectory Based Forwarding. In *Proceedings of International Symposium on Wireless Communication Systems*, Siena, Italy, 2005; pp. 135–139.
11. Blum, B.; He, T.; Son, S.; Stankovic, J. *IGF: A State-Free Robust Communication Protocol for Wireless Sensor Networks*; Technical Report CS-2003-11; Department of Computer Science, University of Virginia, USA, 2003.
12. RYU, J.; Kim, S.G.; Choi, H.H.; An, S.S.; Ahn, S.Y.; Kim, B.J. Method and System for Locating Sensor Node in Sensor Network Using Transmit Power Control. U.S. Patent Application: 2009/0128298 A1, 2009.
13. Barbeau, M.; Kranakis, E.; Krizanc, D.; Morin, P. Improving Distance Based Geographic Location Techniques in Sensor Networks. In *Proceedings of 3rd International Conference on Ad Hoc Networks and Wireless*, Vancouver, British Columbia, 2004; pp. 197–210
14. Perrig, A.; Szewczyk, R.; Tygar, J.D.; Wen, V.; Culler, D.E. SPINS: Security Protocols for Sensor Networks. *Wirel. Netw.* **2002**, *8*, 521–534.
15. Karlof, C.; Sastry, N.; Wagner, D. TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, MD, USA, 2004; pp. 162–175.
16. Lopez, J. Unleashing Public-Key Cryptography in Wireless Sensor Networks. *J. Comput. Security* **2006**, *14*, 469–482.
17. Gaubatz, G.; Kaps, J.-P.; Sunar, B. Public Key Cryptography in Sensor Networks-Revisited. *Lect. Note. Comput. Sci.* **2006**, *3313*, pp. 2–18.
18. Armenia, S.; Morabito, G.; Palazzo, S. Analysis of Location Privacy/Energy Efficiency Tradeoffs in Wireless Sensor Networks. In *IFIP-Networking 2007, LNCS 4479*, Atlanta, GA, USA, 2007; pp. 215–226.
19. Shaikh, R.A.; Jameel, H.; d'Auriol, B.J.; Lee, H.; Lee, S.; Song, Y.-J. Intrusion-Aware Alert Validation Algorithm for Cooperative Distributed Intrusion Detection Schemes of Wireless Sensor Networks. *Sensors* **2009**, *9*, 5989–6007.
20. Shaikh, R.A.; Jameel, H.; Lee, S.; Song, Y.J.; Rajput, S. Trust Management Problem in Distributed Wireless Sensor Networks. In *Proceedings of 12th IEEE International Conference on Embedded Real Time Computing Systems and its Applications (RTCSA 2006)*; IEEE Computer Society: Sydney, Australia, 2006; pp. 411–415
21. Jiang, P. A New Method for Node Fault Detection in Wireless Sensor Networks. *Sensors* **2009**, *9*, 1282–1294.
22. Shaikh, R. A.; Jameel, H.; J. d'Auriol, B.; Lee, H.; Lee, S.; Song, Y.-J. Group-based Trust Management Scheme for Clustered Wireless Sensor Networks. *IEEE Trans. Parallel Dist. Sys.* **2009**, *20*, 1698–1712.
23. Buchegger, S.; Boudec, J.-Y.L. A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks. In *Proceedings of P2PEcon*, Cambridge, MA, USA, 2004.

24. Gupta, S. Automatic Detection of DOS Routing Attacks in Wireless Sensor Networks. Master thesis, University of Houston, Houston, TX, USA, 2006.
25. Du, X.; Guizani, M.; Xiao, Y.; Chen, H.H. Two Tier Secure Routing Protocol for Heterogeneous Sensor Networks. *IEEE Trans. Wirel. Commun.* **2007**, *6*, 3395–3401.
26. Karlof, C.; Wagner, D. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. In *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications (WSNA03)*; IEEE Computer Society: Anchorage, Alaska, USA, 2003; pp. 113–127.
27. Srinivasan, A.; Teitelbaum, J.; Liang, H.; Wu, J.; Cardei, M. Reputation and Trust-based Systems for Ad Hoc and Sensor Networks. In *Algorithms and Protocols for Wireless Ad Hoc and Sensor Networks*; Boukerche, A., Ed.; Wiley & Sons: New Jersey, NJ, USA, 2006.
28. Durresi, A.; Paruchuri, V.; Durresi, M.; Barolli, L. Anonymous routing for mobile wireless ad hoc networks. *Int. J. Distrib. Sens. Netw.* **2007**, *3*, 105–117.
29. Pfleeger, C.P.; Pfleeger, S.L. *Security in Computing*, 4th ed.; Prentice Hall: New Jersey, NJ, USA, 2006.
30. Latif, R.; Hussain, M. Hardware-Based Random Number Generation in Wireless Sensor Networks(WSNs). In *Lect. Not. Comput. Sci.* **2009**, *5576*, 732–740.
31. Seetharam, D.; Rhee, S. An Efficient Pseudo Random Number Generator for Low-Power Sensor Networks. In *Proceedings of Annual IEEE International Conference on Local Computer Networks (LCN'04)*, Tampa, FL, USA, 2004.
32. Tilak, S.; Abu-Ghazaleh, N.B.; Heinzelman, W. A Taxonomy of Wireless Micro-Sensor Network Models. *SIGMOBILE Mob. Comput. Commun. Rev.* **2002**, *6*, 28–36.
33. Shaikh, R.A.; Lee, S.; Khan, M.A.U.; Song, Y.J. LSec: Lightweight Security Protocol for Distributed Wireless Sensor Network. *Lect. Not. Comput. Sci.* **2007**, *4217*, 367–377.
34. Jamieson, K.; Balakrishnan, H.; Tay, Y.C. Sift: A MAC Protocol for Event-Driven Wireless Sensor Networks. *Lect. Not. Comput. Sci.* **2006**, *3868*, 260–275.
35. Lee, S.H.; Cho, B.-H.; Choi, L.; Kim, S.-J. Event-Driven Power Management for Wireless Sensor Networks. *Lect. Not. Comput. Sci.* **2007**, *4761*, 419–428.
36. Figueiredo, C.M.S.; Nakamura, E.F.; Loureiro, A.A.F. A Hybrid Adaptive Routing Algorithm for Event-Driven Wireless Sensor Networks. *Sensors* **2009**, *9*, 7287–7307.
37. Al-Karaki, J.N.; Kamal, A.E.; Routing Techniques in Wireless Sensor Networks: A Survey. *IEEE Wirel. Commun.* **2004**, *11*, 6–28.
38. Chenyang Lu; Blum, B.M.; Abdelzaher, T.F.; Stankovic, J.A.; He, T. RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks. In *Proceedings of 8th IEEE Real-Time and Embedded Technology and Applications Symposium*, San Jose, CA, USA, 2002; pp. 55–66 .
39. Amin, S.O.; Siddiqui, M.S.; Hong, C.S.; Lee, S. RIDES: Robust Intrusion Detection System for IP-Based Ubiquitous Sensor Networks. *Sensors* **2009**, *9*, 3447–3468.
40. Bloom, B.H. Space/Time Trade-Offs in Hash Coding with Allowable Errors. *Commun. ACM* **1970**, *13*, 422–426.

41. Szymanski, B.K. SENSE: Sensor Network Simulator and Emulator. Available Online: http://www.ita.cs.rpi.edu/sense/index.html (accessed on 25 April, 2009).
42. Crossbow Inc. Wireless Sensor Networks, MICA2 Series. Available Online: http://www.xbow.com (accessed on 21 June, 2008).