A Scalable Anonymous Protocol for Heterogeneous Wireless Ad Hoc Networks

Vamsi Paruchuri\textsuperscript{a}, Arjan Durresi\textsuperscript{a,*}, Mimoza Durresi\textsuperscript{b}, Leonard Barolli\textsuperscript{b}, Makoto Takizawa\textsuperscript{b}

\textsuperscript{a}Department of Computer Science, Louisiana State University, 298 Coates Hall, Baton Rouge, LA, 7803, USA

\textsuperscript{b}Department of Information and Communication Engineering, Fukuoka Institute of Technology, 3-30-1 Wajiro-Higashi, Higashi-ku, Fukuoka 811-0295, Japan

*Corresponding author. Tel.: +1 225 578 3902; Fax: +1 225 578 1465; E-mail: durresi@csc.lsu.edu

Abstract— Ensuring anonymity in wireless and hoc networks is a major security goal. Using traffic analysis, the attacker can compromise the network functionality by correlating data flow patterns to event locations/active areas. In this paper we present a novel Scalable Anonymous Protocol that hides the location of nodes and obscure the correlation between event zones and data flow from snooping adversaries. We quantify the anonymity strength of our protocol by introducing a new anonymity metric: Degree of Exposure Index. Our protocol is designed to offer flexible tradeoffs between degree of anonymity and communication-delay overhead.

Keywords—Anonymity, privacy, sensor networks, wireless networks
1. Introduction

Wireless Ad Hoc Networks networks, applied to monitoring physical environments, have recently emerged as an important application resulting from the fusion of wireless communications and embedded computing technologies [19, 20, 21, 22]. Wireless ad hoc networks consist of hundred or thousands of sensor nodes, low power devices equipped with one or more sensors. Potential applications include monitoring remote or inhospitable locations, target tracking in battlefields, disaster relief networks, early fire detection in forests, and environmental monitoring.

With the growth and acceptance of the sensor networks, there has been increased interest in maintaining anonymity in the network. The mere fact that a sensor has sent some information to the base station can reveal extremely important information. For instance, consider a sensor network deployed for intruder detection in which a sensor keeps sensing for intruders. Thus, when an intruder, once in the network area, sees a transmission from a sensor close to his location, can rightly assume that his presence is sensed and might pursue evasive actions immediately. In general, interception of messages containing the physical locations of sensor nodes allows an attacker to locate the nodes and destroy them. The significance of hiding location information from an attacker lies in the fact that the sensor nodes have small dimensions and their physical location cannot be trivially traced. Thus, it is important to hide node locations. In the case of static nodes, the location information does not age and must be protected throughout the lifetime of the network. Moreover, it should be noted that adversaries can correlate data flow patterns to event locations/active areas using traffic analysis. Therefore, there is a strong need to develop anonymity mechanisms which hide the location of nodes and obscure the correlation between event zones and data flow from snooping adversaries.

Privacy International [8] defines four categories of privacy: information privacy, bodily privacy, communication privacy, and territorial privacy. Location privacy is a particular case of information privacy
and can be defined as the ability to prevent other parties from learning one’s current and past locations [3]. Anonymity can be defined as the state of being not identifiable within a set of subjects called the anonymity set [9].

Conventional protocols [5, 6, 7] proposed to ensure user anonymity in the Internet are based on the communication model in which any node can communicate with any other node. But, in a sensor network, most of the connections could be classified as node-to-base station and base station-to-node. In this communication model, just the information that a node has sent some information to the base station might reveal a lot. To the best of our knowledge, ours is the first protocol that deals with providing complete anonymity in sensor networks.

We present Hierarchical Anonymous Communication Protocol (HACP), a novel protocol that prevents traffic analysis from revealing node information including its location. We use token ring approach for achieving anonymity of communication between cluster heads. Routes are chosen and frames are scheduled to traverse these routes. Each frame is assigned a token and a node can send a message through a frame only if the token is free. For communication between sensors and cluster heads, we

The rest of the paper is organized as follows: Section II deals with related work, section III discusses our design goals and network model, section IV presents our protocol, section V discusses security and performance results of HACP and section VI concludes.

2. Related Work

In this section, we discuss some existing research efforts related to sensor networks, secure routing, anonymity and location privacy.

The problem of routing in sensor networks has been initially studied in a non-adversarial setting, and recently the focus of research shifted to the design of secure routing protocols; researchers have already devised a number of proposals to secure both reactive (on-demand) and proactive routing protocols and
identified a number of attacks [23, 24, 25, 26, 27]. There are several recent research efforts exploring different aspects of sensor net security, for example key management [13, 14, 15], secure multicast communication [12], authentication [16,17,18] and location privacy [1,2,3,4].

Anonymous communication for wired networks is a well-studied aspect. A seminal work in the domain of anonymity was notably reported by Chaum in [28]. In [7], Reiter and Rubin present Crowds, a scheme that enables anonymity of web transactions. The concept of a mix is introduced in [29]. A single processor in the network, called a mix, serves as a relay. Each processor P that wants to send a message m to a processor Q encrypts m using Q’s public key to obtain m’. Then P encrypts the pair (m’, q) using the public key of the mix. The mix decrypts the message and forward m’ to q. This scheme has been extended where several mixes are used to cope with the possibility of compromising the single mix. Another approach is to interpose an additional party (an anonymizer [30]) between the sender and receiver to hide sender’s identity from the receiver.

In [31], Gruteser and Grunwald propose an approach to enhance location privacy in wireless LANs based on disposable interface (MAC) identifiers. The Mist routing project [32] addresses the problem of routing a message to the user while keeping its location private. Mist operates by making use of a set of mist routers organized in a hierarchical structure that provides location privacy. In [33], Smailagic et al. present two location sensing systems and compare them to the existing location sensing proposals. They further perform a user privacy study and show that users expect two unique behaviors from the system: an introvert model, where privacy is preferred, and an extrovert model where availability is preferred. In [35], Jackson proposes a system that allows user control of the location information disclosure in systems like Active Badge [34]. An important work on IP private roaming has been reported in the framework of the Freedom Network [36, 37]. Recently, Kong and Hong have proposed a protocol for anonymous communication in mobile ad hoc networks [29]. To the best of our knowledge, we are not aware any work that provides anonymous communication in sensor networks.

In this paper, we consider a more general attacker model considered for the Internet [5, 6, 7] in which the
attacker (that may not be part of the network) has access to the entire networks traffic information. We also take into consideration that most of the traffic in a sensor network is either sensor node to the base station or base station to sensor node rather than between sensor nodes. The protocols described here are designed to be resilient to traffic analysis i.e., to make it difficult for observers to learn identifying information about the origin/destination of a connection. Also, we aim at hiding information about a node transmitting/receiving a message. Thus, the attacker would not be able to even figure out if a node is transmitting any data.

3. Design Goals and Network Models

3.1. Design Goals

We want to design a system that enables anonymous communication. Anonymity is the state of being not identifiable within a set of subjects called the anonymity set. Here, we define these terms more precisely in the context of hybrid ad hoc networks.

Anonymity is generally classified into source and destination anonymity. Source anonymity is defined as the property that a particular message is not linkable to any source, and vice-versa. A similar definition applies to destination anonymity. Unlinkability in this context means that the probability that a particular message was sent by a given source and/or received by the same destination is the same as imposed by the a priori knowledge. This means that the process of sending and/or receiving messages does not reveal any additional information about the identities of the source and/or destination that was not already known to the attacker prior to the message transmission.

3.2. Network Model

We consider clustered sensor networks because clustering allows for scalability of MAC and routing.
Cluster heads also serve as fusion points for aggregation of data, so that the amount of data that is actually transmitted to the base station is reduced. Clustering sensors into groups, so that sensors communicate information only to cluster heads and then the cluster heads communicate the aggregated information to the processing center, may save energy. Many clustering algorithms in various contexts have been proposed [43, 44, 45, 46]. These algorithms aim at generating the minimum number of clusters such that any node in any cluster is at most d hops away from the cluster head.

We use the communication graph $G(V_{\text{CH}}, E)$ to represent the network in terms of cluster heads. $V_{\text{CH}}$ is the set of cluster heads and $E$ is the set of communication edges (might be paths involving intermediate non-cluster heads) connecting the cluster heads. We assume that $G$ is connected.

We initially fix a spanning tree in the graph. Next, using an Euler tour (that is a DFS tour) of the spanning tree in the graph, we define a ring. Also, the ring formation can use the underlying routing protocol to achieve energy efficiency and load balancing.

We base our protocol on symmetric key cryptographic techniques because of infeasibility of implementation of public key protocols in sensor networks [10]. There exist a number of key pre-distribution schemes for sensor networks to set up secret keys among sensors [11, 13, 14, 15]. We assume that each sensor shares a secret key with its cluster head. Also, each cluster head shares a symmetric key with its neighboring cluster heads in the ring. We use $E(M, K_{ij})$ to represent encryption of message $M$ with $K_{ij}$, the secret key shared by nodes $i$ and $j$ and $D(M, K_{ij})$ to represent decryption of message $M$ with $K_{ij}$, the secret key shared by nodes $i$ and $j$.

**Tokens and Frames**

At anytime there can be only one frame traversing through the ring. For simplicity and illustration purposes, we presently consider only one frame. Later in Section 4.3., we present enhancements to deal with multiple frames. The nodes use a *token passing access mechanism* to access a frame passing through the network. A node wishing to send data should first receive permission. When it gets control of the
token, it may transmit data in that frame. Each frame is of fixed length and contains the status of the token itself. A token can be either in free status or occupied status. The format of the frame is as follows:

\[ <E(\text{Token} \parallel E(\text{Frame}_{\text{Header}}, K_{si}) \parallel E(\text{Frame}_{\text{Data}}, K_{sd})), K_s) > \]

where \( K_{si} \) is the secret key shared between the source node \( s \) and node \( i \) that is the upstream neighbor of sender \( s \) and \( K_{sd} \) is the secret key shared between the source node \( s \) and destination node \( d \).

The format of the Token is as follows:

\[ <\text{Redundancy predicate} \parallel \text{Status} > \]

Redundancy predicate is used for checking the validity of the frame. For the frame to be verified successfully by node \( i \), upon decryption the Redundancy predicate must be fulfilled. Status specifies if the token is occupied or free. If a token is free, a node can send data through that frame; else it cannot.

The format of the FrameHeader is as follows:

\[ <\text{Redundancy predicate} \parallel \text{Source Address} \parallel \text{Destination Address}> \]

Again Redundancy predicate is used for checking the validity of \( E(\text{Frame}_{\text{Header}}, K_{sd}) \).

The format of FrameData is as follows:

\[ <\text{Data length} \parallel \text{Data} \parallel \text{Padding}> \]

Data length specifies the length of the total data in the packet. This is crucial when the amount of data needed to be sent is not enough to fill the whole frame. In that case, data to be sent is padded with some random number to meet the constraint that the size of the frame is of fixed length.

4. Scalable Anonymous Protocol (SAP)
SAP provides two different mechanisms to achieve anonymity – one is based on introducing dummy messages for anonymity within a cluster and the other is based on ring-based approach for anonymous communication within cluster heads.

4.1. **Anonymous communication within a cluster**

Inserting dummy traffic in a network is a technique that hides the traffic patterns inside the network, making traffic analysis more difficult [40]. The generation of dummy traffic increases the anonymity of the messages sent through the mix network.

A dummy message is a *fake message* created by a sensor node. The final destination is its cluster head; the dummy message is discarded by the cluster head. Observers of the network and other nodes cannot distinguish the dummy from a real message.

In SAP, each sensor (including the cluster head) transmits messages at a Poisson rate $r_t$. Thus, on average each sensor would send a message every $1/r_t$ seconds. Let $r_s$ denote the sensing rate of each sensor. Thus, whenever there is sensed data to be sent, the sensor encrypts the data message with the secret key it shares with the cluster head and transmits it. Else, the sensor sends dummy messages. Hence, the dummy messages are sent at a rate of $(r_t - r_s)$.

Whenever a cluster head has a message to be sent to one of its cluster nodes, the cluster head simply encrypts the message with the secret key it shares with that sensor and sends.

Whenever a sensor senses a packet transmission, it receives the packet and decrypts it with its key and checks if it is a valid packet.

4.2. **Anonymous communication between cluster heads**
Whenever a node $i$ receives a frame, it decrypts the frame using the key shared with its downstream node in the ring and verifies the redundancy predicate. Once the *Redundancy predicate* is fulfilled, the following algorithm is executed.

1. If the node has no data to send, it just encrypts the resultant plain frame with the common key shared with its upstream node and retransmits the packet on to the ring.

2. If the *status* of the token is *free* and the node has some data to send to another node D, then $i$ constructs the frame as follows:
   - Node $i$ constructs $Frame_{Header}$ and $Frame_{Data}$ as explained earlier using key shared with the Destination.
   - Node $i$ sets the *status* field in the token to *occupied*.
   - Computes the following using its shared key with upstream node and transmits the packet on the ring.

   $$<E((Token||E(Frame_{Header},K_{sd})||E(Frame_{Data},K_{sd})),K_s)>$$

3. If the *status* of the token is set to *occupied*, the node checks if the data in the frame is destined to itself by decrypting $<E(Frame_{Header},K_{sd})>$ with the shared key and checking if the *Redundancy predicate* is fulfilled.
   - If the node is able to check the validity of the frame header, then it is addressed to node $i$, which makes a copy of it. It encrypts the whole frame with the key shared with its upstream node and transmits the frame on to the ring.
   - Else, if the node $i$ is not able to check the validity of the frame header, then it is not the destination and the node just encrypts the whole frame with the key shared with its upstream node and transmits the frame on to the ring.

Once the frame returns to the source, the source repeats the procedure as long as it has data to send. When it has no more data to send it sets the *status* field of the token to *free*, assigns the whole frame to
some randomly generated data. Then it encrypts the whole frame with its shared key with upstream node and transmits the frame on the ring.

4.3. Multiple Rings

In a network consisting of \( n \) nodes, the ring size is \( n \). Thus, a message needs to be transmitted along the whole ring and hence, each message is transmitted \( n \) times. To reduce the communication overhead (complexity), we divide the graph into sub-graphs and construct rings within each sub-graph. We choose the same partition mechanism presented in [10, 19]. An example partition is shown in Fig. 1. The dark circle indicates the base station to which all the nodes are communicating with.

Once we have the partition to sub-graphs, we have one ring in each sub-graph, which is formed by an Euler tour on the spanning tree of the sub-graphs. We call the nodes that are part of more than one ring as Junction nodes. There are at most \( \delta x \) nodes in each sub-graph, thus the time complexity is no more than \( \delta x \) within a sub-graph.

In order to enable communication with a node outside a sub-graph, we assign each ring a unique identifier, RID. Also, each node knows the RID of the ring to which the destination belongs. We introduce a new header \( E(\text{Frame}_{\text{RID}}, K_{sJ}) \) – in the frame in order to identify the destination’s RID, where \( K_{sJ} \) is the common key shared by the source with the Junction node that is also part of a ring that has to be traversed to reach the destination.

The modified format of the frame as follows:

\[
\langle E((\text{Token} \| E(\text{Frame}_{\text{RID}}, K_{sJ}) \| E(\text{Frame}_{\text{Header}}, K_{sJ}) \| E(\text{Frame}_{\text{Data}}, K_{sJ})), K_d)\rangle
\]

The format of Frame_{RID} is

\[
\langle \text{RIP} \| \text{RID}_D \rangle
\]
RIP is the redundancy predicate that has to be fulfilled so as to indicate successful decryption. RID_D is the Ring Identifier of the destination’s ring. The sender encrypts Frame_RID with the key shared with the Junction node that is part of ring that is on the way to the destination’s ring.

When a node in one ring has data to send to a node in another ring, then the frame need to be transferred from one ring to another until it reaches the ring of the destination. For this each Junction node maintains a forwarding routing table that specifies the ring a frame addressed to a particular destination ring has to be transferred to. A Junction Node upon successful decryption of E(Frame_RID, K_{sJ}) stores a copy of the frame and then retransmits the frame. The junction node based on the RID of the destination node, decides to which ring the frame has to be transferred. Then, it waits for a free token on the other ring it has to transmit the copied frame, encrypts the frame with the common key it shares with the next junction node on the way to the destination’s ring and transmits the frame. The process continues till the frame reaches the destination’s ring, where the Junction node that of RID_D that receives the frame just assigns some random string to E(Frame_RID, K_{sJ}) and transmits the frame on to the ring RID_D.

This mechanism prevents local traffic from traversing the whole network. Even if an adversary were able to compromise a Junction node, he would just be able to know the ring to which frame was destined to and no more. The attacker could not even figure out the originating ring of the frame. Thus, this mechanism does not reduce the anonymity provided by the protocol.

In some situations, only some nodes might have a need for anonymity in which case a ring has to be established only among those nodes. In such cases, the neighbors in a ring need not be physical neighbors in the network and these nodes can communicate using the shortest path available.

5. Performance of SAP
In this section we present the performance of SAP in terms of the overhead imposed and the anonymity provided. Initially, we describe the metrics we would be considering and present the performance of SAPP in terms of these metrics.

5.1. Metrics

Anonymity can be measured with various metrics, among which the most common is based on anonymity set. In our system, if the attacker holds the list of registered network nodes, the maximum degree of anonymity that the system can provide is proportional to the size of the list; in this case, the list corresponds to the anonymity set of the network. We will assume that the network has a sufficiently large anonymity set, so that it thus provides a reasonable anonymity to the users. In our protocol, the size of the anonymity set is same as the size of the ring. Thus, bigger the ring is more is the anonymity provided.

We present a new metric Data Exposure Index in section V.C. This metric effectively captures the probability with which an attacker can guess if a node is sending data. We consider the communication overhead imposed by the protocol and also discuss the average delay encountered by a packet before it reaches the destination.

5.2. Communication overhead

In SAP, whenever a node has data to send, it captures a free token and sends data in that frame. Else, it just forwards the idle frame. Thus, even if node has any data to send, at least one frame would be traversing the ring. We use the term communication overhead to represent the number of transmissions that correspond to idle frames. It should also be noted that the power consumption of a sensor can be derived from the average current drain [38] given by

$$I_{avg} = T_{on} * I_{on} + (1 - T_{on}) * I_{sby}$$

where

$T_{on}$ is fraction of time receiver or transmitter is on

$I_{on}$ is current drain from battery when receiver or transmitter is on and
$I_{\text{stby}}$ is current drain from the battery when both transmitter and receiver are off.

Thus, higher the communication overhead higher is the $T_{on}$, which implies higher is the power consumption. Thus, communication overhead also acts as a direct measure of power consumption.

Consider a ring with $N$ number of nodes out of which $N_a$ nodes have data to send at a rate of $R$ packets per unit time. Let us say, a frame can traverse the ring at a maximum of $t$ times in one unit of time. The value of $t$ depends on ring latency, which in turn depends on the transmission time of the frame ($T_{tr}$), ring traverse time delay ($T_r$) and processing delay at a node ($T_{proc}$). Here, we ignore the delay incurred at a node to process the frame before forwarding it.

$$t = \frac{1}{(N \cdot T_{tr} + T_{proc} + T_r)}$$

If $n$ tokens are present in the ring, then a maximum of $n \cdot t$ frames can be transmitted across the ring. Thus, ideally, we would like to have the following condition satisfied, so that no idle frame is transmitted:

$$\frac{N_a \cdot R}{N} = n \cdot t$$

Thus, the fraction of idle frames being transmitted over the ring is $1 - \frac{N_a \cdot R}{N \cdot n \cdot t}$. Thus, communication overhead i.e., number of transmissions corresponding to idle frames, is given by

Communication overhead

$= \text{number of idle frames} \cdot \text{number of nodes in the ring}$

$= N - \frac{N_a \cdot R}{n \cdot t}$

The communication overhead in rings for varying sizes and for different number of tokens is presented in Fig. 2. The communication overhead increases almost linearly as number of nodes in the ring increases.
This behavior is as expected because with more number of nodes in a ring more number of transmissions occur corresponding to each frame generated by any node.

5.3. Data Exposure Index

We introduce a new metric called Data Exposure Index (DEI) defined as follows:

\[
\text{DEI} = \frac{\text{Number of data generating nodes on the Ring}}{\text{Total Number of Nodes on the Ring}}
\]

The worst case scenario is when the DEI is equal to one. In this case all nodes on the ring generate data and the attacker’s assumption that data is being sent by some node is valid. The goal of hiding the information is a node is sending some data cannot be achieved in this case. On the other hand lower DEI is achieved by having few data source nodes on the ring. Less data sources or more the total nodes on the ring reduces the chances of the attacker to identify the data sources.

Fig. 3 shows the trade off between communication overhead and exposure degree. When the total number of nodes on the ring increases, while having the data sources the same, it can be observed that the DEI (right y axis) decreases but the bandwidth/power overhead (left y axis) increases. The user can get different tradeoffs by changing the number of data sources on the ring. For instance, for high anonymity, rings with high number of nodes have to be used, but which results in high communication overhead. Also, to keep the DEI low, ring formation should be such that only few nodes are transmitting at a given point of time. It should be noted most of the related works aim at hiding the communication pattern (i.e., who is talking to who) and not hiding the information if a node is transmitting or not. For these works, the DEI would be one as the attacker would be able to figure out who is transmitting and who is receiving, though he is not able to find out who is receiving from whom.
5.4. Mean waiting time

The mean waiting $E[W]$ for a frame normalized to $X$ is computed in [39] and is given as

$$E[W] = \frac{\rho(1+2a'+a'^2)+(1+\frac{\rho}{M}(1+a'))}{2\left(1-(1+a'(1+\frac{1}{M}))\rho\right)}$$

where,

- $M$ is number of nodes in the ring
- $X$ is frame transmission time
- $\rho$ is load of a station and defined as arrival rate at a station $X$ transmission time. This assumes exponential inter-arrival times
- $t'$ is the ring latency i.e., the propagation delay for a frame to traverse the ring
- $a' = t'/X$

Fig. 4 presents the waiting time in rings with different number of nodes. As it can be observed, the wait time increases very fast as the number of nodes in the ring increases. Fig. 5 shows the variation in wait time as the number of active nodes in the ring is varied. As expected, with increase in the number of nodes that have data to send, the wait time increases.

From Figs. 4 and 5, the trade off between number of nodes in the ring and anonymity degree is clear. For time sensitive data which require low latencies, rings with less number of nodes have to be formed which in turn results in lesser communication overhead and at the same time lesser anonymity.

6. Conclusions
The data-centric behavior of ad hoc and sensor networks leaves them vulnerable to traffic analysis and identification of event locations and active areas. Therefore, ensuring data anonymity is a crucial research area. We presented Scalable Anonymous Protocol (SAP) to achieve anonymous communications in a sensor network. We divide the network into rings and use the concept of tokens and rings to achieve anonymity.

We also present the tradeoffs between the overhead imposed and ring sizes. We show that higher anonymity comes at a cost – either higher communication/energy overhead or at higher latency. The choice of the parameters is left to the network administrator and depends on level of security needed and the type of traffic in the network.

References


30. www.anonymizer.com


FIGURE CAPTIONS

Fig. 1. A partition of a network into multiple rings

Fig. 2. Communication Overhead vs. number nodes in a ring
Fig. 3. Tradeoff between Communication overhead and the Data Exposure Index.

Fig. 4. Wait time in rings of different sizes.

Fig. 5. Wait time vs. number of active nodes in the ring. Total nodes = 32.
Cluster Head
Base Station
Region defining one ring
Number of nodes in the ring

communication overhead

0 5 10 15 20

0 5 10 15 20

1 Data node
3 Datanodes
5 Data
Number of Nodes in the Ring

Communication Overhead

Data Exposure Index

- Overhead - 1 Data Node
- Overhead - 2 Data Nodes
- Overhead - 3 Data Nodes
- DEI - 1 Data node
- DEI - 2 Data nodes
- DEI - 3 Data nodes
The graph shows the average wait time for different load levels with varying numbers of active nodes. The load is represented on the x-axis, and the average wait time is represented on the y-axis. The graph includes lines for 10 active nodes, 8 active nodes, 5 active nodes, and 3 active nodes, each distinguished by a different line style and marker.