A Hierarchical Anonymous Communication Protocol for Sensor Networks

Arjan Durresi[1], Vamsi Paruchuri[1], Mimoza Durresi[2], and Leonard Barolli[2]

[1] Department of Computer Science, Louisiana State University,
298 Coates Hall, Baton Rouge, LA, 70803, USA
{durresi,paruchuri}@csc.lsu.edu
http://www.csc.lsu.edu/∼durresi
[2] Department of Information and Communication Engineering
Faculty of Information Engineering, Fukuoka Institute of Technology
3-30-1 Wajiro-Higashi, Higashi-ku, Fukuoka 811-0295, Japan
durresim@franklin.edu, barolli@fit.ac.jp

Abstract. Ensuring anonymity in sensor 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 hierarchical anonymous communication 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.

1 Introduction

Wireless sensor networks, applied to monitoring physical environments, have recently emerged as an important application resulting from the fusion of wireless communications and embedded computing technologies. Sensor 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. 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.

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* [1]. Anonymity can be defined as the state of being not identifiable within a set of subjects called the anonymity set [2].

Conventional protocols [3],[4] proposed to ensure user anonymity in the Internet are not suitable for sensor networks. 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. Dummy messages are used fora Anonymity inside clusters.

The rest of the paper is organized as follows: Section 2 deals with related work, Section 3 discusses our design goals and network model, Section 4 presents our protocol, Section 5 discusses security and performance results of HACP and Section 6 concludes.

2 Related Work

Anonymous communication for wired networks is a well-studied aspect. A seminal work in the domain of anonymity was notably reported by Chaum in [5]. In [6], Gruteser and Grunwald propose an approach to enhance location privacy in wireless LANs based on disposable interface (MAC) identifiers. The Mist routing project [7] addresses the problem of routing a message to the user while keeping its location private. In [8], Smailagic et al. present two location sensing systems and compare them to the existing location sensing proposals. In [9], Jackson proposes a system that allows user control of the location information disclosure in systems like Active Badge [10]. An important work on IP private roaming has been reported in the framework of the Freedom Network [11], [12]. Recently, Kong and Hong have proposed a protocol for anonymous communication in mobile ad hoc networks [13].

3 Design Goals and Network Model

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 [14],[15],[16],[17]. 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(VCH, E) to represent the network in terms of cluster heads. VCH 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 [18]. There exist a number of key pre-distribution schemes for sensor networks to set up secret keys among sensors [19]. 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. 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:

$$< E((Token||E(Message_{Header}, K_{sd})||E(Message_{Data}, K_{sd})), K_{si}) > \quad (1)$$

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: $< Redundancypredicate||Status >$

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 $Frame_{Header}$ is as follows:
$< Redundancy\ predicate||Source\ Address||Destination\ Address >$

The format of $Frame_{Data}$ is as follows: $< Data\ length||Data\ Padding >$

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 Hierarchical Anonymous Communication Protocol (HACP)

HACP provides two different mechanisms to achieve anonymity - one is based on introducing dummy messages for anonymity with in a cluster and the other is based on ring based approach for anonymous communication with in cluster heads.

4.1 Anonymous communication with in a cluster

Inserting dummy traffic in a network is a technique that hides the traffic patterns inside the network, making traffic analysis more difficult [20]. 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 HACP, each sensor (including the cluster head) transmits messages at a Poisson rate r_t. Thus, on an 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 Equation (1) using its shared key with upstream node and transmits the packet on the ring.
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 with in each sub-graphs. We choose the same partition mechanism presented in [18]. 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 x nodes in each sub-graph, thus the time complexity is no more than x within a sub-graph.

In order to enable communication with 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 - $E(Frame_{RID}, K_{sJ})$ - 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: $< E((Token||E(Frame_{RID}, K_{sJ})||E(Frame_{Header}, K_{sd})|| E(Message_{Data}, K_{sd})), K_{si}) >$

The format of $Frame_{RID}$ is: $< RIP||RIDD >$. 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 HACP

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

5.1 Communication overhead

In HACP, 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. 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 [21] given by: $I_{avg} = T_{on} * I_{on} + (1 - T_{on}) * I_{stby}$, 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_{stby} is current drain from the battery when both transmitter and receiver are off. Therefore, higher the communication overhead higher the T_{on}, which implies higher is the 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_t) and processing delay at a node (T_{proc}). Here, we ignore the delay incurred at a node to process the frame before forwarding it. Therefore, $t = \frac{1}{N*T_{tr}+T_{proc}+T_t}$.

If n tokens are present in the ring, then a maximum of $n * t$ 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: $\frac{N_a}{N} * R = n * t$.

Thus, the fraction of idle frames being transmitted over the ring is: $1 - \frac{N_a * R}{N * n * t}$. Thus, communication overhead i.e., number of transmissions corresponding to idle frames, is given by: number of idle frames $\times$ number of nodes in the ring $= N - \frac{N_a * R}{n * t}$. The communication overhead in rings for varying sizes and for different number of tokens is presented in Figure 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.

5.2 Data Exposure Index

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

$$DEI = \frac{Number\ of\ data\ generating\ nodes\ on\ the\ ring}{Number\ of\ total\ nodes\ on\ the\ ring} \tag{2}$$

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

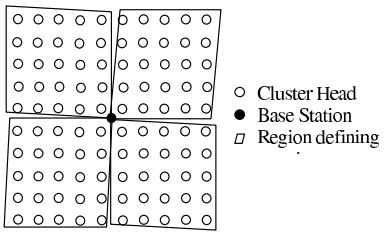

Fig. 1. Partition of a network into multiple rings

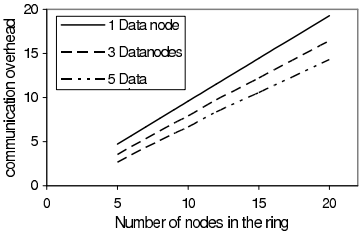

Fig. 2. Communication Overhead Vs number nodes in a ring

by some node is valid. 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 tradeoff 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.

5.3 Mean Waiting Time

Fig. 4 presents the waiting time calculated as shown in [22] 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 tradeoff 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 less communication overhead but at the same time in less anonymity.

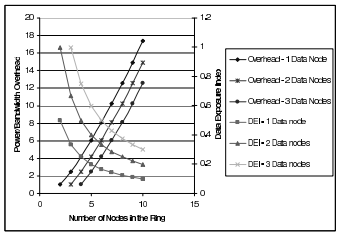

Fig. 3. Tradeoff between Communication overhead and the Data Exposure Index

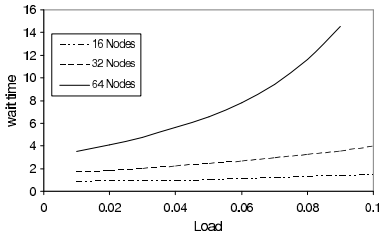

Fig. 4. Wait time in rings of different sizes

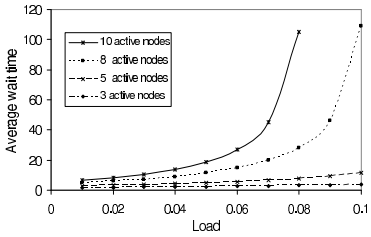

Fig. 5. Wait time vs. number of active nodes in the ring. Total nodes = 32

6 Conclusion

The data-centric behavior of 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 Hierarchical Anonymous Communication Protocol (HACP) 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

1. Alastair R. Beresford and Frank Stajano: Location Privacy in Pervasive Computing. IEEE Pervasive Computing, 2(1):46-55, 2003.
2. A. Pfitzmann and M. Kohntopp: Anonymity, unobservability, and pseudonymity - a proposal for terminology. In Proceedings of International Workshop on Design Issues in Anonymity and Unobservability, 2000
3. Goldschlag D., Reed M. and Syverson P.: Onion routing for anonymous and private Internet connections. Communications of the ACM 42, 2 (Feb. 1999), 39-41.

4. Amos Beimel and Shlomi Dolev: Buses for anonymous message delivery. In Second International Conference on FUN with Algorithms, pages 1-13, Elba, Italy, May 2001. Carleton Scientific.
5. D. Chaum: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM, 4(2), February 1981.
6. M. Gruteser and D. Grunwald: Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis. In Proceedings of WMASH, 2003.
7. J. Al-Muhtadi, R. Campbell, A. Kapadia, D. Mickunas, and S. Yi: Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments. In Proceedings of the International Conference of Distributed Computing Systems (ICDCS), 2002.
8. A. Smailagic, D. P. Siewiorek, J. Anhalt, D. Kogan, and Y. Wang: Location Sensing and Privacy in a Context Aware Computing Environment. Pervasive Computing, 2001.
9. I. W. Jackson: Anonymous Addresses and Confidentiality of Location. In Proceedings of International Workshop on Information Hiding, 1996.
10. R. Want, A. Hopper, V. Falcao, and J. Gibbons: The active badge location system. ACM Transactions on Information Systems, January 1992
11. http://www.zeroknowledge.com
12. P. Boucher, I. Goldberg, and A. Shostack: Freedom System 2.0 Architecture. Zero-Knowledge Systems Inc. white paper, December 2000.
13. J. Kong and X. Hong: ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks. In Proceedings of MobiHoc, 2003.
14. S. Bandyopadhyay and E.J. Coyle: An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003), San Francisco, CA, March 30 - April 3, 2003.
15. C.F. Chiasserini, I. Chlamtac, P. Monti and A. Nucci: Energy Efficient design of Wireless Ad Hoc Networks. In Proceedings of European Wireless, February 2002
16. Ossama Younis and Sonia Fahmy: Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach. Proceedings of IEEE INFOCOM, volume 1, pp. 629-640, March 2004.
17. Ossama Younis and Sonia Fahmy: HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-hoc Sensor Networks. IEEE Transactions on Mobile Computing, volume 3, issue 4, pp. 366-379, Oct-Dec 2004.
18. Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, and Doug Tygar: SPINS: Security Protocols for Sensor Networks. Proceedings of Seventh Annual International Conference on Mobile Computing and Networks MOBICOM 2001, July 2001.
19. 11. Wenliang Du, Jing Deng, Yunghsiang S. Han, Shigang Chen and Pramod Varshney: A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge. Proc. of IEEE INFOCOM'04, March 2004.
20. Claudia Daz and Bart Preneel: Reasoning about the Anonymity Provided by Pool Mixes that Generate Dummy Traffic. Appeared in Information Hiding, 6th International Workshop, Lecture Notes in Computer Science, Springer-Verlag, 16 pages, 2004
21. Edgar H. Callaway, Jr., Wireless Sensor Networks: Architectures and Protocols. New York: Auerbach Publications (an imprint of CRC Press). 2003.
22. Alberto Leon-Garcia and Indra Widjaja: Communication Networks: Fundamental Concepts and Key Architectures. McGraw Hill, 2000.