Private Cluster Head Election in Wireless Sensor Networks

Levente Buttyán and Tamás Holczer
Laboratory of Cryptography and Systems Security (CrySyS)
Budapest University of Technology and Economics
Email: {buttyan, holczer}@crysys.hu

Abstract

Clustering is a useful mechanism in wireless sensor networks that helps to cope with scalability problems and, if combined with in-network data aggregation, may increase the energy efficiency of the network. At the same time, by assigning a special role to the cluster head nodes, clustering makes the network more vulnerable to attacks. In particular, disabling a cluster head by physical destruction or jamming may render the entire cluster inoperable temporarily until the problem is detected and a new cluster head is elected. Hence, the cluster head nodes may be attractive targets of attacks, and one would like to make it difficult for an adversary to identify them. The adversary can try to identify the cluster head nodes in various ways, including the observation of the cluster head election process itself and the analysis of the traffic patterns after the termination of the cluster head election. In this paper, we focus on the former problem, which we call the private cluster head election *problem. This problem has been neglected so far, and as a consequence, existing cluster head election protocols leak too much information making the identification of the elected cluster head nodes easy even for a passive external observer. We propose the first private cluster head election protocol for wireless sensor networks that is designed to hide the identity of the elected cluster head nodes from an adversary that can observe the execution of the protocol.*

1. Introduction

Wireless sensor networks (WSNs) are often envisioned to be deployed in large scale monitoring applications. In those applications, the WSN may consists of hundreds or potentially thousands of sensor nodes. Such a large network usually cannot efficiently operate without some structuring. For this reason, clustering has been proposed as a way to introduce some hierarchical structure in the network (see [1] for a survey on proposed clustering protocols for sensor networks). Besides alleviating the scalability problem, clustering combined with in-network data aggregation is also seen as a useful mechanism to increase the energy efficiency of the system, another important design criterion of WSNs.

While clustering has many advantages regarding the operation of the WSN, we argue that it has some drawbacks with respect to security. In particular, each cluster usually has a controller node, called the cluster head, that has a distinguished role. For instance, the cluster head may be responsible for controlling the operation of the sensor nodes in the cluster by setting their configuration parameters, and for aggregating the sensor readings collected from the cluster and storing the result or sending it to the sink or some higher level cluster head. If such a cluster head node is disabled by physical destruction or jamming, then the entire cluster becomes inoperable temporarily until the problem is detected and a new cluster head is elected. Hence, when clustering is used, the adversary can focus its effort and resources on attacking the cluster head nodes, which constitute only a small subset of the nodes[1].

In order to address this problem, one would like to make it difficult for the adversary to identify the cluster head nodes. For this, in turn, one needs to understand when and how the adversary may attempt such an identification. Basically, the adversary can try to identify the cluster heads either (i) during the cluster head election process itself, or (ii) after the cluster head election during the regular operation of the network. In case (i), the adversary may passively observe the execution of the cluster head election protocol and learn, just like the cluster members, which nodes became cluster heads, or it may actively interfere with the execution of the protocol and try to manipulate its outcome such that it becomes easier for the adversary to figure out who the cluster heads are. In case (ii), the adversary may eavesdrop the messages and try to figure out from the addressing information who the cluster heads are, or, if such addressing information is hidden from the adversary, it may try to analyze the traffic patterns and identify the cluster heads as the sinks of the local traffic flows. The adversary may also try to tamper with an arbitrary sensor node and read from its stored state information (e.g., its routing table) who may be the cluster head that the node is associated with.

In this paper, we focus on the prevention of the identification of the cluster head nodes during the execution of the cluster head election protocol, because this seems to be a completely neglected problem. Indeed, most existing

1. Similarly, base stations may be attractive targets of attacks. However, base stations may be assumed to be better protected (more difficult to compromise) than cluster head nodes, which are elected from the set of regular sensor nodes.

978-1-4244-5113-5/09/$25.00

cluster head election protocols use cleartext cluster head announcement messages to broadcast the identifier of the cluster head within the cluster, and hence, trivially leak the identity of the elected cluster heads even to a passive observer. We propose the first private cluster head election protocol for wireless sensor networks that minimizes the information leaked out about the elected cluster heads to an external observer. We address both eavesdropping attacks and traffic analysis attacks against the cluster head election protocol. In addition, if the adversary tampers with an arbitrary sensor node, our protocol ensures that the adversary can identify only those cluster heads that the compromised sensor node can be potentially associated with but no other cluster heads in the network.

The remainder of the paper is organized as follows: In Section 2, we introduce our system and adversary models. In Section 3, we present our basic private cluster head election protocol, and in Section 4, we propose some extensions to the basic protocol and fine tune its parameters. In Section 5, we briefly discuss the problem of identifying the cluster heads after the execution of the cluster head election protocol, and give an overview of some related work on anonymous routing protocols for wireless ad hoc and sensor networks. However, the full treatment of this problem is left for future work. Finally, in Section 6, we conclude the paper.

2. System and attacker models

The sensor network consists of the sensor nodes that communicate with each other via wireless channels. Each node can directly communicate with the nodes within its radio range; those nodes are called the (one-hop) neighbors of the node. In order to communicate with distant nodes (outside the radio range), the nodes use multi-hop communications.

The nodes may be aware of their geographical locations, and they may already be partitioned into well defined geographical regions. In this case, these regions are the clusters, and the objective of the cluster head election protocol is to elect a leader within each geographical region. We call this approach location based clustering; an example would be the PANEL protocol [2].

Alternatively, the nodes may be unaware of their locations, and know only their neighbors. In this case, the clusters are not pre-determined, but they are dynamically constructed parallel to the election of the cluster heads. Basically, any node may announce itself as a cluster head, and the nodes within a certain number of hops on the topology graph may join that cluster head as cluster members. We call this approach topology based clustering; an example would be the LEACH protocol [3].

The location based and the topology based approaches are illustrated in Figure 1.

Both approaches may use controlled flooding of broadcast messages. In case of location based clustering, the scope of a flood is restricted to a given geographic region. Nodes within that region re-broadcast the message to be flooded when they receive it for the first time. Nodes outside of the region simply drop the message. In case of topology based clustering, we assume that the broadcast messages has a Time-to-Live field that controls the scope of the flooding. Any node that receives a broadcast message with a positive TTL value for the first time will automatically decrement the TTL value and re-broadcast the message. Duplicates and messages with TTL smaller than or equal to zero are silently discarded. When we say that a node broadcasts a message, we mean such a controlled flooding (either location based or topology based, depending on the context).

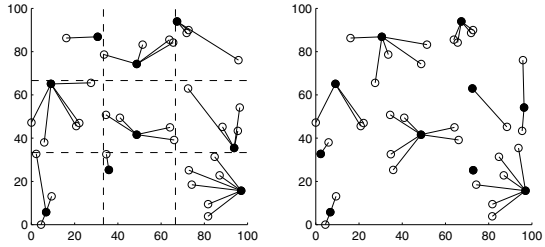

Figure 1. Result of a location based (left), and topology based (right) one-hop cluster head election protocol. Solid dots represent the cluster heads, and empty circles represent cluster members.

We call the set of nodes which are (in the location based case) or can potentially be (in the topology based case) in the same cluster as a node S the *cluster peers* of S. Hence, in the location based case, the cluster peers of S are the nodes that reside within the same geographic region as node S. In the topology based case, the set of cluster peers of S usually consists in its n-hop neighborhood, for some parameter n. The nodes may not explicitly know all their cluster peers.

Regarding the adversary, we assume that it is a global external observer, which can eavesdrop messages and perform traffic analysis everywhere in the network, but it cannot perform active attacks on the cluster head election protocol.

Against such an adversary, a rather simple solution could be based on a commonly shared global key. Using that shared global key as a seed of a pseudo random number generator, every node can construct locally (without any communications) the same pseudo randomly ordered list of all nodes. These lists will be identical for every node because all nodes use the same seed and the same pseudo random number generator. Then, the first A nodes of the list are elected cluster heads such that every node can communicate with a cluster head and no subset of A covers the whole system. An illustration of the result of this algorithm can be seen in Figure 1 for location based and topology based

cluster head election.

The problem with this solution is that it is not robust: compromising a single node would leak the common key, and the adversary could compute the identifier of all cluster heads. While we do not want to fully address the problem of compromised nodes in this paper, we still aim at a more robust solution than the one described above. In particular, the system should not collapse by compromising just a single or a few node.

3. Basic protocol

In this section, we describe the basic protocol that we propose for private cluster head election. An important extension to this basic protocol will be presented in Section 4, where we also describe how to set the parameters of the protocol. We first give a brief overview of the basic principles of our protocol, and present the details later.

We assume that the nodes are synchronized (see [4] for a survey on time synchronization mechanism for sensor networks), and each node starts executing the protocol roughly at the same time. The protocol terminates after a predefined fix amount of time. During the execution of the protocol, any node that has not received any cluster head announcement yet may decide to become a cluster head, in which case, it broadcasts a cluster head announcement message announcing itself as a cluster head. This message is broadcasted among the cluster peers of the node sending the announcement (see previous section). Upon reception of a cluster head announcement, any node that has neither announced itself as a cluster head nor received any such announcement yet will consider the sender of the announcement as its cluster head. In order to prevent an external observer to learn the identity of the cluster heads, all messages sent in the protocol are encrypted such that only the nodes to whom they are intended can decrypt them. For this, we assume that each node shares a common key with all of its cluster peers (an overview of available key establishment mechanisms for sensor networks can be found in [5]). In addition, in order to avoid that message originators are identified as cluster heads, the nodes that will be cluster members are required to send dummy messages that cannot be distinguished from the announcements by the external observer (i.e., they are encrypted and disseminated in the same way as the announcements).

The pseudo-code of the protocol is given in Algorithm 1, and a more detailed explanation of the protocol's operation is presented below. The protocol consists of two rounds, where the length of each round is τ. The nodes are synchronized, they all know when the first round begins, and what the value of τ is. At the beginning, each node starts two random timers, T1 and T2, where T1 expires in the first round (uniformly at random) and T2 expires in the second round (uniformly at random). Each node also initializes at random

Algorithm 1 Basic private cluster head election algorithm

```
start T1, expires in rand(0,τ) //timer, expires in round 1
start T2, expires in rand(τ,2τ) //timer, expires in round 2
announFirst = (rand(0,1) ≤ γ)
CHID = -1 // ID of the cluster head of the node
while T1 NOT expired do
  if receive ENC(announcement) AND (CHID = -1) then
    CHID = ID of sender of announcement
  end if
end while
// T1 expired
if announFirst AND (CHID = -1) then
  broadcast ENC(announcement);
  CHID = ID of node itself;
else
  broadcast ENC(dummy);
end if
while T2 NOT expired do
  if receive ENC(announcement) AND (CHID = -1) then
    CHID = ID of sender of announcement
  end if
end while
// T2 expired
if (NOT announFirst) AND (CHID = -1) then
  broadcast ENC(announcement);
  CHID = ID of node itself;
else
  broadcast ENC(dummy);
end if
```

a binary variable, called announFirst, that determines in which round the node would like to send a cluster head announcement. The probability that announFirst is set to the first round is γ, which is a system parameter. The setting of γ will be elaborated in Section 4.

In the first round, every node S waits for its first timer T1 to expire. If S receives an announcement before T1 expires, then the sender of the announcement will be the cluster head of S. When T1 expires, S broadcasts a message as follows: If announFirst is set to the first round and S has not received any announcement yet, then S sends an announcement, in which it announces itself as a cluster head. Otherwise, S sends a dummy message. In both cases, the message is encrypted (denoted by ENC() in the algorithm) such that only the cluster peers of S can decrypt it.

The second round is similar to the first round. When T2 expires S broadcasts a message as follows: If announFirst is set to the second round and S has not received any announcement yet, then S sends an announcement, otherwise, S sends a dummy message. In both cases, the message is encrypted.

It is easy to see that at the end of the second round each node is either a cluster head or it is associated with a cluster

head whose ID is stored in variable CHID. Without the second round, a node can remain unassociated, if it sends and receives only dummy messages in the first round. In addition, a passive observer only sees that every node sends two encrypted messages, one in each round. This makes it difficult for the adversary to identify who the cluster heads are (see also more discussion on this in the next section). In addition, if a node is compromised, the adversary learns only the identity of the cluster heads whose announcements have been received by the compromised node.

4. Extensions and fine tuning

Thanks to the dummy messages and the encryption in the basic protocol, an external observer cannot trivially identify the cluster heads; however, it can still use side information and suspect some nodes to be cluster heads with higher probability than some other nodes. Such a side information is the number of the cluster peers of the nodes. Indeed, the probability of becoming a cluster head depends on the number of the cluster peers of the node. For instance, if a node does not have any cluster peers (except itself), it will be a cluster head with probability one. On the other hand, if the node has a larger number of cluster peers, then the probability of receiving an announcement from a cluster peer is larger, and hence, the probability that the node itself becomes cluster head is smaller. Note also that the number of cluster peers can be deduced from the topology of the network, which may be known to the adversary.

In the sequel, we call the number of cluster peers of a node (including the node itself) the node's *degree* for short. The probability of becoming a cluster head is approximately inversely proportional to the degree:

$$\Pr(\text{CH(S)}) \cong \frac{1}{D(S)} \tag{1}$$

where $CH(S)$ is the event of S being elected cluster head, and $D(S)$ is the degree of node S. Figure 2 illustrates this proportionality where the curve belongs to Equation 1 and the plotted dots correspond to simulation results (100 nodes, random deployment, one hop communication, topology based clustering).

An efficient approach to mitigate this problem is to modify the nodes' degree such that it becomes a common value α for all of them. In theory, this common value can be anything between 1 and the total number N of the nodes in the network. In practice, it should be around the average degree, which can be estimated locally by the nodes. For example, assuming one-hop communications (meaning that the cluster peers are the radio neighbors), the following formula can be used:

$$\alpha = E(D(S)) = (N-1)\frac{R^2\pi}{A} + 1 \tag{2}$$

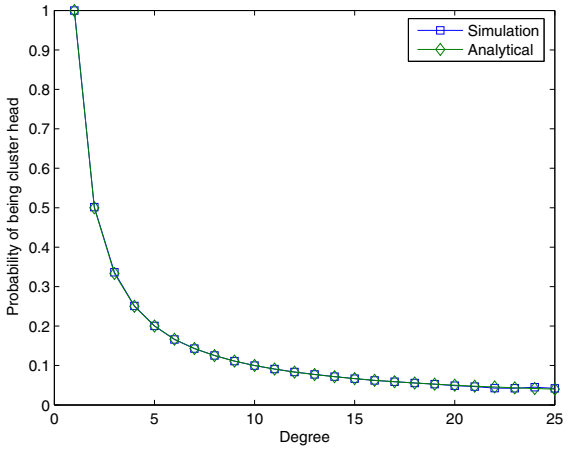

Figure 2. Probability of being cluster head as a function of the degree.

where R is the radio range, and A is the size of the total area of the network. The formula is based on that the degree is proportional to the radio coverage, total area ratio. Similar formulae can be derived for the general case of multi-hop communication.

If a node S has more than α cluster peers it can simply discard the messages from $D(S) - \alpha$ randomly chosen cluster peers. If S has less than α cluster peers it must get new cluster peers by the help of its actual cluster peers (if S has not got any cluster peers originally, then it will always become a cluster head). The new cluster peers can be selected from the set of cluster peers of the original cluster peers. To explore the potential new cluster peers, every node can broadcast its list of cluster peers within its few hop neighborhood before running the basic protocol. From the lists of the received cluster peers, every node can select its $\alpha - D(S)$ new cluster peers uniformly at random. Then, the basic cluster head election protocol can be executed using the balanced set of cluster peers. An example for this degree balancing is shown in Figure 3 (70 nodes, random deployment, one hop communication, topology based clustering).

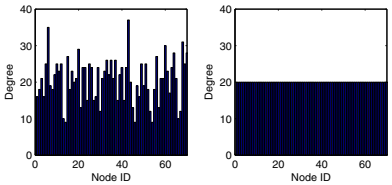

Figure 3. Result of degree balancing. The 70 nodes are represented on the x axis. The degrees before (left), and after (right) the balancing are represented on the y axis.

Another important side information an attacker can use

is the *order* in which the nodes send messages in the first round of the protocol. Indeed, the sender of the i-th message will be cluster head if none of the previous $i-1$ messages are announcements (but dummies) and the i-th message is an announcement. Thus, the probability P_i that the sender of the i-th message becomes cluster head depends on i and parameter γ:

$$P_i = (1-\gamma)^{i-1}\gamma, 1 \leq i \leq n$$

The $(n+1)$-st element of the distribution is the probability that no announcement is sent in the first round:

$$P_{n+1} = (1-\gamma)^n$$

in which case the sender of the first message of the second round must be a cluster head.

The entropy of this distribution characterizes the uncertainty of the attacker who wants to identify the cluster head using the order information. Assuming that the degree has been already balanced, this entropy can be calculated as follows:

$$H = -\sum_{i=1}^{n+1} P_i \log P_i = \tag{3}$$

$$-\sum_{i=1}^{n}\left((1-\gamma)^{i-1}\gamma\log\left((1-\gamma)^{i-1}\gamma\right)\right)-$$

$$-(1-\gamma)^n\log(1-\gamma)^n$$

where γ is the probability of sending an announcement in the first round and n is the balanced degree.

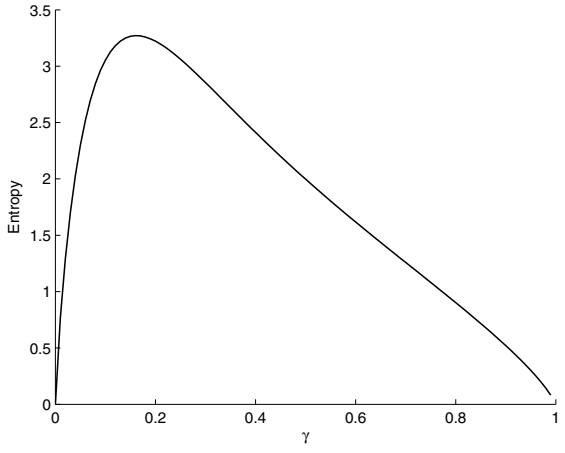

Figure 4. Entropy of the attacker as a function of sending announcement in the first round (γ). Number of nodes in one cluster: 10.

In Figure 4, we plotted formula (3). If γ is large, then the uncertainty of the attacker is low, because one of the first few senders will become the cluster head with very high probability. If γ is very small, then the uncertainty of the attacker is small again, because no cluster head will be elected in the first round with high probability, and therefore, the first sender of the second round will be the cluster head. The ideal γ value corresponds to the maximum entropy, which can be easily computed by the nodes locally from formula (3). For instance Table 1 shows some ideal γ values for different number of nodes in one cluster.

Table 1. Optimal γ values ($\hat{\gamma}$) for different number of nodes in one cluster. Achieved entropy ($H(\hat{\gamma})$) and maximal entropy ($H_{\max} = \log_2 n$)

n	10	25	50	100
$\hat{\gamma}$	0.167	0.082	0.049	0.027
$n\hat{\gamma}$	1.67	2.05	2.45	2.7
$H(\hat{\gamma})$	3.281	4.410	5.312	6.218
$H_{\max}$	3.322	4.644	5.644	6.644

5. Further discussions

So far, we addressed the problem of preventing an adversary from identifying the cluster heads during the execution of the cluster head election protocol. However, as we mentioned earlier, the adversary may also attempt to identify the cluster heads once they have been elected. Even if the addressing information in the messages of the data communication protocol used between the cluster members and the cluster heads are encrypted, the adversary can still perform traffic analysis, and identify the cluster heads based on the observed traffic patterns. For instance, the cluster head usually collects sensor readings from the cluster members and then sends those readings or their aggregated value to the sink. This results in a very particular traffic pattern within the cluster, namely, the cluster head always sends after the cluster members and it may send its messages to a larger distance in case of a remote sink. Similarly, if the cluster head does not send anything to the sink but stores the sensor readings of the cluster members locally (e.g., to be fetched later by a mobile sink), then the node that only receives messages can be suspected by the adversary to be the cluster head. Obviously, the details may depend on the particular protocol used for communications, but the general idea should be clear.

In order to cope with this problem, the communication protocol used in the network for sending sensor readings must provide receiver anonymity. The cluster members could broadcast their messages in an encrypted form such that all other nodes in the cluster receive them but only the cluster head can decrypt them. Such broadcast communication does in effect provide receiver anonymity, but it is very costly in terms of overhead and energy consumption.

An alternative, less secure but more efficient approach would be to use an anonymous routing protocol such as

those proposed in [6], [7], [8], [9], [10]. These protocols, however, are general purpose routing protocols, and their applicability for intra-cluster communications still needs to be investigated in terms of efficiency achieved, level of security provided, and the potential integration with the cluster head election protocol and the in-network aggregation scheme.

A very relevant paper related to the problem at hand is [11], where the location privacy of the sink in WSNs is investigated with respect to an adversary that performs traffic analysis. The authors describe a set of basic decorrelation techniques to mislead the attacker such as packet reencryption, introducing random delays, and more sophisticated routing mechanisms such as selection of random paths and random path forking. The investigation of these and similar receiver anonymity techniques within the context of cluster based communications and in-network aggregation is on our future research agenda.

6. Conclusion and future work

This paper has two main contributions: First, we introduced the problem of private cluster head election, i.e., the problem of preventing an external global observer from identifying the cluster head nodes elected during the cluster head election process. We argued that if the cluster head nodes can be identified by an adversary, then it can efficiently attack the network by focusing its effort and resources on disabling the cluster heads. Hence, a private cluster head election protocol is highly desirable. Yet, the existing cluster head election protocols do not have this desirable property. In order to remedy this situation, and as a second contribution of the paper, we proposed the first private cluster head election protocol. Our protocol is rather simple and it is suitable for both location based and topology based clustering. We also proposed a useful extension to our basic protocol and showed how to fine tune its parameters such that the amount of information leaked by the protocol about the identity of the cluster heads is minimized.

While our protocol can resist against a passive observer, it is vulnerable to active attacks, in particular, to physical tampering of the nodes. More specifically, if an adversary physically compromises a sensor node, then it can learn all its state information, including the identifier of the cluster head node with which the compromised node is associated. We are currently working on the design of a private cluster head election protocol that can resist against even physical tampering.

We also briefly touched upon the problem of preventing the identification of the cluster head nodes after the termination of the cluster head election process, i.e., during the regular operation of the network. We suggested that communication protocols that ensure receiver anonymity may be useful building blocks for the solution to that problem. However, the full treatment of this problem is left for future work.

Acknowledgments. The work described in this paper is based on results of the WSAN4CIP project (`http://www.wsan4cip.eu`), which receives research funding from the European Community's 7th Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. Levente Buttyán has also been partially supported by the Hungarian Academy of Sciences through the Bolyai János Research Fellowship.

References

[1] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[2] L. Buttyán and P. Schaffer, "PANEL: Position-based Aggregator Node Election in Wireless Sensor Networks," in *Proceedings of the 4th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2007.

[3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference onSystem Sciences.*, 2000.

[4] Y. R. Faizulkhakov, "Time synchronization methods for wireless sensor networks: A survey," *Programming and Computing Software*, vol. 33, no. 4, pp. 214–226, 2007.

[5] J. Lopez and J. Zhou, *Wireless Sensor Network Security*. Cryptology and Information Security Series, IOS Press, 2008.

[6] S. Seys and B. Preneel, "ARM: Anonymous routing protocol for mobile ad hoc networks," in *20th International Conference on Advanced Information Networking and Applications, AINA*. IEEE, 2006, pp. 133–137.

[7] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Mask: Anonymous on-demand routing in mobile ad hoc networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 9, pp. 2376–2385, 2006.

[8] H. Choi, P. McDaniel, and T. La Porta, "Privacy Preserving Communication in MANETs," in *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2007, pp. 233–242.

[9] T. Rajendran and K. V. Sreenaath, "Secure anonymous routing in ad hoc networks," in *Proceedings of the 1st Bangalore Annual Computer Conference*. ACM New York, 2008.

[10] A. A. Nezhad, A. Miri, and D. Makrakis, "Location privacy and anonymity preserving routing for wireless sensor networks," *Computer Networks*, vol. 52, no. 18, pp. 3433–3452, 2008.

[11] J. Deng, R. Han, and S. Mishra, "Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks," *Pervasive and Mobile Computing*, vol. 2, no. 2, pp. 159–186, 2006.