Privacy-Aware Probabilistic Sampling for Data Collection in Wireless Sensor Networks

João Guerreiro, Edith C.-H. Ngai and Christian Rohner
Department of Information Technology
Uppsala University, Sweden

***Abstract*— The rising popularity of web services and their applications to sensor networks enables real-time data collection and queries by users. Unlike traditional periodic data collection, the traffic patterns generated from real-time data collection may expose the interests of users or the locations of unusual events to the attackers. To provide privacy in data collection, we propose a novel *probabilistic sampling* mechanism that can hide user queries and unusual events in the network, while supporting both routine and on-demand data reporting. Our goal is to prevent attackers from locating the unusual events and identifying interests of users by eavesdropping and analyzing the network traffic. In our probabilistic sampling scheme, the data are carefully reported at random times in order to mask the unusual events and user queries. In the meantime, our scheme can provide users with high data accuracy at minimized communication overheads. Extensive simulations have been conducted to evaluate the security strength, data accuracy and communication overheads of the proposed scheme.**

***Index Terms*— Wireless sensor network, privacy, sampling, data collection**

I. Introduction

The initial proposal of Web Services [1] comes from the need of companies to standardize the way of exchanging data between different applications. Recently, Web Services are also suggested for intercommunication between sensor networks and the Web which enables users to make queries and collect data more easily. Furthermore, it has been shown that having web services on sensors is feasible [2], especially with ease using RESTful [1][3] interfaces to request resources.

Data collection from a sensor network can be achieved periodically, which we call *periodic sampling*. This mechanism is popular in many environmental monitoring applications in which data are reported to the gateway periodically for processing and storage [4]. Unfortunately this method does not satisfy a user who has soft real time requirements. Instead of waiting for the next periodic data report, the user should be able to query the sensors immediately in query-driven or event-driven applications [5]. In this paper, we consider data reports that could be triggered by the requests from the gateway or generated autonomously by the sensors both routinely or on-demand. However, the real-time feature also opens up vulnerabilities for attackers to identify the interests of users or locate the unusual events in the sensing field.

Although various encryption schemes [6] can be applied to keep the content of messages secret, the user activity also needs to be hidden in order to discourage an eavesdropper from performing traffic analysis. An attacker might be interested in discovering the user's patterns such as times for requesting data and which sensors are the most solicited. In this paper, we propose a novel sampling mechanism, called *probabilistic sampling*, which can hide the traffic patterns and user behaviors in data collection for wireless sensor networks. This method will enable the blending of user requests and events that need to be noticed, with the routine traffic, so that they become indistinguishable to an attacker.

The paper is organized as follows. In section II we review the related work that has been done and inspired us to come up with this problem. In Section III we define more clearly the problem at hand and set some objectives for our work. Section IV is dedicated to the explanation of the probabilistic sampling mechanism that we propose to solve our problem. In section V we evaluate the performance of our proposed method with extensive simulations. Finally we conclude the paper in section VI.

II. Related Work

Privacy issues in computer networks and wireless communications have attracted a lot of attention. Related work has been done on how to keep intact the privacy of users who access remote resources on the web in order to hide their interests [7]. In this paper, we focus on providing privacy in the vicinity of the sensor network that uses web services as an easy and general way for data collection. Focusing on the sensor network itself, we are interested in understanding the traffic analysis techniques that could be applied by the eavesdroppers or attackers [8] and their counter measures. In [9], a security mechanism has been proposed to perform traffic anonymity with dummy traffic synthesis. We are also concerned about hiding user activity, but the difference is that the dummy traffic that we generate actually contains useful information. This information can improve the data accuracy without invading the privacy in data collection. Equally interesting is the work done in [10] where sensor messages are buffered in intermediary nodes to keep the temporal privacy of sensed events, so that an eavesdropper cannot infer accurately the moment when an event of interest was sensed. Different from the above work, our work aims at masking the data reports completely, rather than delaying their delivery to the sink.

Besides temporal privacy, location privacy have become a major concern in wireless sensor networks [11], [12], [13]. The

978-1-4577-9538-2/11/$26.00 ©2011 IEEE

random walk based phantom flooding scheme [14] has been proposed to defend against an external adversary who attempts to trace back to the data source in sensor networks and provide source location privacy. A path perturbation algorithm [15] has also been proposed to cross paths in areas where at least two users meet which intends to make the attackers confuse the paths of different users. Other schemes, like ConstRate and ProbRate, which introduce dummy traffic to hide the real event sources, are proposed to provide source event unobservability in the network [16], [17]. The above work either injects dummy packets at the sources or prolongs the original paths for routing which will increase the traffic and energy consumption of the nodes. On the contrary, all traffic generated in our approach is carrying useful data that can improve the data accuracy for users. Moreover, our work considers not only the location privacy of the sources, but we aim at hiding the interests of our users by generating autonomous data reports adaptively to the user queries.

III. Problem Descriptions

A. System Model

We consider a system model that enables users to make queries and receive data reports conveniently from the sensor network through the Internet. A typical and efficient way of relaying sensor data to the Internet using web services is illustrated in figure 1. More specifically, a RESTful interface [1] is adequate for the task due to its simplicity. REST (Representational State Transfer) suggests that each unique URL is a representation of an object. The advantages of REST web services are lightweight, easy to build and the results are human readable [3].

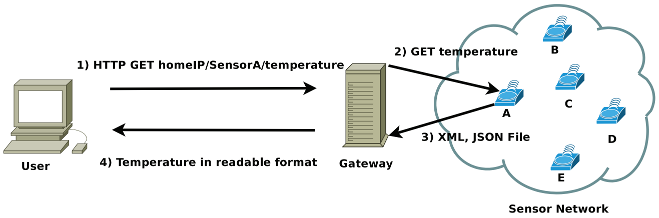
1) HTTP GET homeIP/SensorA/temperature
2) GET temperature
3) XML, JSON File
4) Temperature in readable format
User
Gateway
A
B
C
D
E
Sensor Network

Fig. 1. System model with web services.

As we can see in the figure 1, in the first step, the user requests the temperature of sensor A with his web browser. This request is then passed through the gateway to sensor A. In the third step, the sensor replies with an XML or a JSON file that contains the answer. A typical JSON document containing the answer would look like :

```
[{
"sensorName": "A",
"currentTemperature": 17.5,
"currentTime": "13:53:04"
}].
```

The presence of a gateway can also protect the sensor network from the outside world. It is used for restricting access to the sensor network, so that only authenticated users can use the system.

In our system model, we consider exclusively two communication scenarios.

- In the first scenario, a query is initiated by the user and is passed to the sensors through the gateway. It is followed by the replies from the sensors in a JSON document.
- The second scenario consists of automatic data reports from the sensors, without any request from the gateway. This situation may occur either on routine basis or when an unusual event is detected. An unusual event is defined by the system as abnormal situation that needs to be reported urgently. For example, an extremely elevated temperature could mean that there is a fire inside a house.

We consider passive attackers that could eavesdrop the network traffic and identify the areas of interest from the users by analyzing the traffic patterns [18]. For instance, the attackers could locate the unusual events and the solicited sensors by observing the increased network traffic in particular areas.

Our probabilistic sampling method leverages the data reports generated by the sensors and triggered by the gateway to mask the user activities and unusual events. Both the sensors and the gateway will generate automatic data reports with probabilities adaptive to the query rates and the unusual events in the network. These randomly generated data reports could hide the on-demand data reports triggered by the user queries or unusual events. The automatically generated data reports and the on-demand data reports are hard to be distinguished by eavesdropping and network traffic analysis. Note that the automatically generated data reports will replace the routine traffic periodically generated for continuous monitoring, so that they will not give extra communication overheads.

B. Objectives

In this paper, our primary goal is to protect the sensor network from passive attacks. Indeed an attacker listening to the network traffic may observe a sudden increase of network traffic and identify unusual events and user activities in particular areas of the network by performing traffic analysis.

The envisioned probabilistic sampling mechanism that we propose have the following main features:

- We aim at providing users with ability to collect sensing data both on routine basis and on demand. It means that the system can monitor the environment continually at low data rate. In the meantime, users should be able to make real-time queries without waiting for the next period of data collection.
- Our proposed data collection mechanism should provide security against passive attacks. It should be able to prevent attackers from identifying the interests of users and the unusual events in the sensing field.
- Besides protecting the privacy, our probabilistic sampling method should not introduce too much network traffic. An easy way to mask user activities is by collecting data at all times, which is not efficient in terms of energy

consumption. Hence, we need to find a smarter way of doing it.

- Although automatic reports for continuous monitoring could be generated at random times to hide the user queries, we want to make sure that our probabilistic sampling method can provide satisfactory data accuracy for users when they consult the sensor readings of a whole day.

The process of masking user queries and unusual events is similar, except that unusual events are rarer which makes them even easier to handle.

On the other hand, the number of user queries is varying along time, which requires a dynamic mechanism for obfuscation. This is a non-trivial problem since the users' behaviors are unpredictable, so we have to adapt our sampling method to the number of user queries in order to obfuscate the user behaviors. When we try to minimize the data reports, we also need to keep the samples well distributed across time. For example, if the data are mostly collected at the beginning of the day, the errors will be large for the remaining of the day due to the lack of data.

IV. Probabilistic Sampling

A. Design Principles

The main idea of probabilistic sampling is to confuse the attacker by reporting the data in a non-deterministic way, such that the attackers could not identify the user queries or unusual events by observing the traffic pattern. The data reports for masking the user queries and unusual events will be generated probabilistically either triggered by the gateway or directly from the sensors.

More formally, we consider that the sensor readings are taken throughout the day and we divide the day into time intervals with equal number of time slots. If the size of time interval is five, it means that each interval contains five time slots. A time slot is a short period of time, where at most one data report can be made to the gateway. If a user query was made but the sensing data had already been collected in that time slot, then the user will retrieve a cached value from the gateway.

Our method possesses three main policies:

- The use of probabilities for reporting data.
- If too many time slots passed without any data report, then take appropriate action.
- Allow at most one user query per time interval.

The first policy implies that there will be a certain probability of reporting data in every time slot, either triggered by the gateway or from an autonomous sensor report. Both the gateway and the sensors will have the same probability of initiating a data report, such as $\frac{1}{Interval\ size \times 2}$, so that there is a good chance that at least one data report will occur in total for each time interval. By doing so, the data achieved for continuous monitoring can keep a level of accuracy similar to periodic sampling at the same time interval. The interval size is a parameter chosen by the user. The greater the interval size, the less data reports there will be.

The second policy also aims at increasing the accuracy of the data. Even though we manage to report data at least once in each time interval, there might still be too many time slots between two reports. Hence, once we observe this phenomenon, we increase the probability of reporting immediately and we keep the high probability until the report actually occurs. After this happens, we can reset the probability to its original value. Note that we increase the probability, but do not force a report with a 100% chance. Otherwise, after some traffic observations, the attacker will be able to completely differentiate some of the generated reports from the user generated ones.

The third policy advocates that a user should only be able to query once every time interval. It would not be beneficial for a user to query more, otherwise an attacker can identify a higher activity in that interval by comparing it to the others. It is reasonable to adopt this kind of policy, since the user does not gain much insight by collecting successive temperature readings in very short time. The quality of the information will not increase significantly anyway.

As a consequence from the last policy, after a user query occurs, we will set the probability of report generation to zero in the remaining of that time interval. This can avoid situations where after a user query has been made, several generated reports follow up, thus raising the suspicion of an attacker.

B. Probabilistic Sampling Algorithm

Algorithm 1 shows the pseudo-code of how our probabilistic algorithm is implemented. The algorithm will determine the data report probability, $Proba\_Report$, from the gateway and the sensors according to the three policies presented above.

In the first phase of the algorithm, we initialize the variables. $Proba\_Report$ represents the probability of generating an automatic data report. This probability is the same for both the data reports triggered by the gateway and the individual sensors. The variable $zerosCounter$ is used to keep track of how many time slots have passed without any report, so that we can elevate the reporting probability if necessary.

For each time slot, the algorithm determines the probability $Proba\_Report$ according to the three data report generation policies. Initially the algorithm checks if the current time slot is at the beginning of a new time interval. If so, the values of variables $Proba\_Report$ and $zerosCounter$ will be reset. Afterwards, it will decide if some sort of activity will occur during the current time slot. If data is not collected, our algorithm will increase the variable $zerosCounter$. It will also increase the value of $Proba\_Report$ if there have been too many time slots without data collection.

As a final remark, though it is not presented in this algorithm, we take into account the timing issues. For example, if the gateway and the individual sensors both decide to report data during the same time slot, sometimes one might be faster than the other or they might report at the exact same time.

V. Evaluations

A. Simulation Settings

To conduct an evaluation for our algorithm, we stress the importance of using realistic data. We used the data collected

Algorithm 1 Probabilistic Sampling Algorithm

```
Require: $Interval\ Size$

  % Variables initialization
  $Proba\_Report = \frac{1}{IntervalSize \times 2}$
  $zerosCounter = 0$

  % For each time slot of the day
  for $i = 1$ to $\#TimeSlots$ do
    if Time slot is at beginning of a new Interval then
      Reset value of $Proba\_Report$
    end if

    % First policy
    Schedule sensor report with probability $Proba\_Report$
    Schedule gateway query with probability $Proba\_Report$

    if No data report is scheduled for current time slot then
      $zerosCounter = zerosCounter + 1$
      % Second policy
      if $zerosCounter \geq IntervalSize$ then
        $Proba\_Report = \frac{1}{2}$
      end if
    else
      Generate the data report
      Save the sensed data
      $zerosCounter = 0$
      Reset value of $Proba\_Report$
    end if

    % Third policy
    if A user query was generated then
      $Proba\_Report = 0$
    end if
  end for
```

from the sensors (station ID 1) deployed by EPFL[1] in their campus. These data are collected on 17/06/2008, covering the readings of the whole day. There are 675 collected data samples, which are separated in time intervals between 2 and 3 minutes. The time differences between each sample are probably due to the latency of communications. We consider that each collected sample is valid for a time between 2 and 3 minutes. We consider this period of time as a time slot. Only one data report can be collected during a time slot.

We run our simulations based on the real sensing data using Matlab with user queries randomly added. There is a probability $\frac{Number\ of\ User\ Queries}{Total\ Number\ of\ Time\ Slots}$ of generating a user query in each time slot. We also implement and evaluate our probabilistic sampling algorithm, together with the periodic sampling algorithm for comparison. We run these algorithms for ten thousand times each, since we find that it is long enough for the results to converge to a certain value. In the following evaluations, we collect the data samples from the sensors and reconstruct the continuous data by interpolations for measuring data accuracy. The interpolation technique applied here is third degree interpolation. Third degree interpolation gives more accurate results in exchange for extra computation time. We consider this not to be a problem, since all the interpolation calculations are performed in the gateway which has more processing power than the sensors.

B. Data Accuracy

In the first experiment, we measure the MAE (Mean Absolute Error) for quantifying how much the interpolated data deviate from the real data in probabilistic sampling and periodic sampling respectively. A lower MAE is desirable, since it implies a higher accuracy of the reconstructed data.

The definition of MAE is

$$MAE = \frac{\sum_{t=1}^{N} |r_t - i_t|}{N},$$

where r_t is the real data at time t and i_t is the interpolated results, and N is the total number of data samples that can be collected during a whole day.

In figure 2, we examine how the MAE is affected by the choice of the size of time intervals and the number of user queries made to the system. A good result we find beforehand is that the MAEs are quite low for both methods.

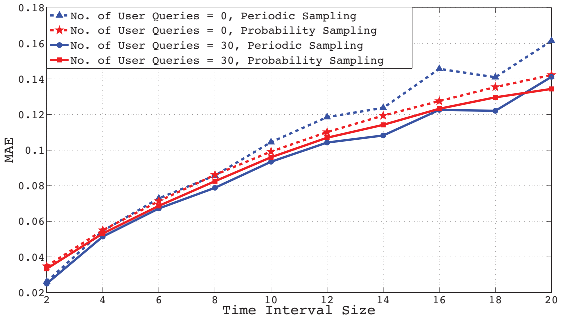

Fig. 2. MAE comparison varying size of time interval.

From the figure, we can see that with zero user queries the probabilistic sampling method has a lower MAE than the periodic sampling method, even though the data collected by probabilistic sampling are not uniformly spread. Nevertheless, the two methods still achieve very close MAEs.

Once user queries are made, the accuracy of periodic sampling becomes slightly better than our probabilistic method. Indeed from 0 to 30 user queries in a day, the MAE of probabilistic sampling changes very little. This is due to the fact that less automatic reports (controlled by $Proba\_Report$) will be generated in our algorithm if the number of user queries increases. Thus the total number of collected data will not increase as much as that in periodic sampling.

It should also be noted that the higher the interval size the less data reports are made, so it is normal that the MAE increases with a higher interval size.

[1] http://sensorscope.epfl.ch/index.php/SensorScope_Deployments

C. Communication Overheads

In this experiment we will evaluate the communication overheads of both methods for data collection.

In figure 3, we show the average number of transmissions in periodic sampling varying the total number of user queries in a day. We set the size of time interval to five. We find that the number of autonomously generated sensor reports remains the same independent of the user queries. The total number of transmissions increases with the number of user queries.

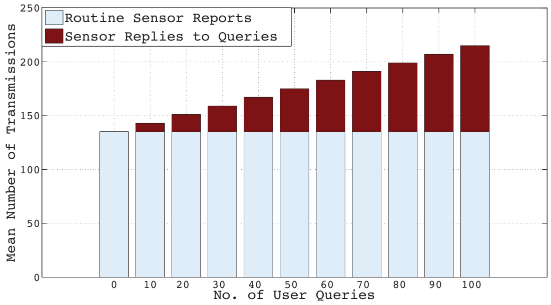

Fig. 3. Mean number of transmissions in periodic sampling varying total number of user queries in a day.

In figure 4 we show the results of the same experiment with probabilistic sampling. We can see that the more sensors replies to the queries generated by the users or our method, the less our algorithm generates automatic reports from the sensors. Overall, the total number of transmissions does not change much with the number of user queries. This observation also explains the small MAE variation in figure 2 when the number of user queries is increased.

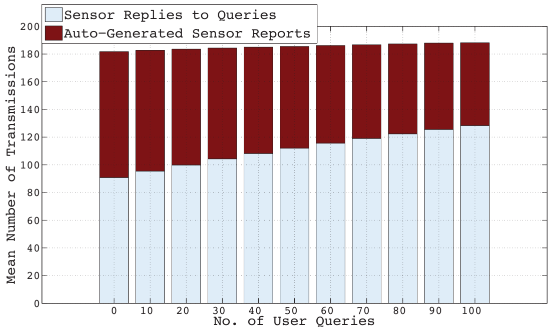

Fig. 4. Mean number of transmissions in probabilistic sampling varying total number of user queries in a day.

If we compare the two figures, we can see that probabilistic sampling maintains a quite constant number of transmissions even with different number of user queries. It implies that our proposed method can save more energy of the sensors when the number of user queries is high.

D. Security Strength

An attacker attempts to explore the user activities by observing the network traffic. He will try to identify the time intervals that a user query is present. To characterize how successful an attacker can be in determining those time intervals, we define the attacker's success rate, $Succ$, as follows

$$Succ = \frac{Correctly\ Suspected\ Intervals}{Intervals\ With\ Activity + Wrong\ Guesses}.$$

The success rate is calculated by the correct guesses he makes, divided by the sum of total number of time intervals containing at least one user query and the number of wrong guesses the attacker makes. Note that the number of wrong guesses is added in the denominator as a penalty of extensive random guesses.

To determine which time intervals are suspicious of possessing user activity, the attacker will go through all the intervals and check which contain more than one data report. The attacker will then mark those intervals as suspicious of involving user activities.

Note that the attacker can determine which intervals contain unusual events in the same way. Due to the low probability of unusual events, the successful rate for the attacker to identify the unusual events is expected to be lower than that of identifying user queries.

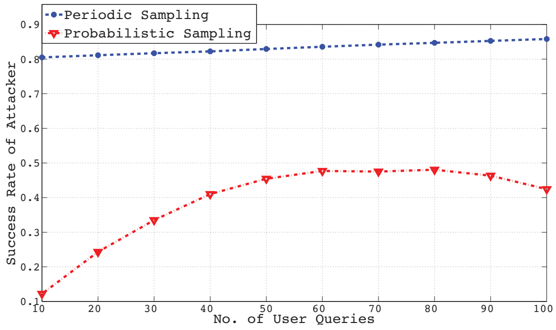

Fig. 5. Attacker's success rate on detecting user activities.

In figure 5, we set the number of unusual events to zero and the interval size to five, while varying the number of user queries. We compare the successful rate of attackers on identifying the user queries in periodic sampling and probabilistic sampling. In periodic sampling, the only intervals that are not identified are those with a user query that occurred in the same time slot as the periodic data report.

By comparing the two methods, we can see that probabilistic sampling can mask the user activities much better than periodic sampling. The success rate of attacker is quite low in probabilistic sampling especially in situations with small number of user queries.

VI. Conclusions and Future Work

In this paper, we proposed a novel probabilistic sampling algorithm to protect the privacy of user activities and unusual events in wireless sensor networks. Our scheme can protect the networks against passive attackers who intend to identify the time and location of user queries and unusual events by eavesdropping and performing network traffic analysis. We jointly considered the automatic data reports generated by the sensors or triggered by the gateway, together with the on-demand data reports triggered by the user queries or unusual events. By carefully adjusting the generating probability of automatic data reports, we can effectively mask the user queries and unusual events in the network. In the meantime, our scheme keeps the number of transmissions reasonably low and it will not increase with user queries. We evaluated the data accuracy, communication overheads, and security strength of probabilistic sampling and compared with traditional periodic sampling mechanism by extensive simulations. The results demonstrated that probabilistic sampling can achieve much better security strength than traditional periodic sampling, while providing comparable data accuracy and communication overheads.

In the future, we are interested in improving the data accuracy and security strength by adding randomness in the spatial domain. Sensors located in particular area may share the data reporting duties. By combining their results, the data accuracy from interpolation could be improved with reduced communication overheads, while the attackers would also find it harder to identify patterns by traffic analysis.

Acknowledgement

This work was supported by the VINNOVA VINNMER program funded by the Swedish Governmental Agency for Innovation Systems.

References

[1] L. Richardson and S. Ruby, *RESTful Web Services*, O'Reilly Media, Inc. May 2007.

[2] D. Yazar and A. Dunkels, "Efficient application integration in IP-based sensor networks", *Proceedings of ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 43-48. 2009.

[3] R. T. Fielding and R. N. Taylor, "Principled design of the modern Web architecture", *Proceedings of International Conference on Software Engineering*, pages 407-416. 2000.

[4] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy efficient communication protocol for wireless microsensor networks", *Proceedings of Hawaii International Conference on System Sciences*, 2000.

[5] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks", *Proceedings of SIGMOD Record*, 2002.

[6] J. Zhang and V. Varadharajan, "Wireless sensor network key management survey and taxonomy", *Journal of Network and Computer Applications 33(2)*, pages 63-75. 2010.

[7] Y. Elovici, B. Shapira and A. Maschiach, "A new privacy model for hiding group interests while accessing the Web", *Proceedings of the ACM Conference on Computer and Communications Security*, pages 63-70. 2002.

[8] G. Danezis and R. Clayton, "Introducing traffic analysis", Available: http://research.microsoft.com/en-us/um/people/gdane/papers/TAIntro-book.pdf, 2007. [Accessed August 14, 2010].

[9] W. M. Shbair, A. R. Bashandy and S. I. Shaheen, "A new security mechanism to perform traffic anonymity with dummy traffic synthesis". *Proceedings of International Conference on Computational Science and Engineering*, vol. 1, pages 405-411. 2009.

[10] P. Kamat, W. Xu, W. Trappe and Y. Zhang, "Temporal privacy in wireless sensor networks", *Proceedings of IEEE ICDCS*, pages 23-23, 2007.

[11] M. Gruteser, G. Schelle, A. Jain, R. Han and D. Grunwald, "Privacy-aware location sensor netwroks", *Proceedings of USENIX Workshop on Hot Topics in Operation Systems (HotOS IX)*, 2003.

[12] Ying Jian, Shigang Chen, Zhan Zhang and Liang Zhang, "Protecting receiver-location privacy in wireless sensor networks", *Proceedings of IEEE Infocom*, pages 1955-1963, 2007.

[13] E. Ngai and, I. Rodhe, "On providing location privacy for mobile sinks in wireless sensor networks", *Proceedings of ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 116-123, 2009.

[14] P. Kamat, Y. Zhang, W. Trappe and C. Ozturk, "Enhancing source-location privacy in sensor network routing", *Proceedings of IEEE ICDCS*, 2005.

[15] B. Hoh and M. Gruteser, "Protecting location privacy through path confusion", *Proceedings of IEEE/CreateNet International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm)*, 2005.

[16] M. Shao, Y. Yang, S. Zhu and G. Cao, "Towards statistically strong source anonymity for sensor networks", *Proceedings of IEEE Infocom*, 2008.

[17] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar and G. Cao, "Towards event source unobservability with minimum network traffic in sensor network", *Proceedings of ACM WiSec*, 2008.

[18] J. Deng, R. Han and S. Mishra, "Countermeasures against trace analysis attacks in wireless sensor networks", *Proceedings of IEEE/CreateNet International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm)*, 2005.