This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2011 proceedings.

Towards Efficient Anonymous Communications in Sensor Networks

Juan Chen[1], Hongli Zhang[1], Binxing Fang[1,3], Xiaojiang Du[2], Lihua Yin[3], Xiangzhan Yu[1]

[1] School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China,
e-mail: janechen.hit@gmail.com, zhanghongli@hit.edu.cn

[2]Dept. of Computer and Information Sciences, Temple University, Philadelphia, PA, USA, e-mail: dux@temple.edu

[3]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, e-mail: fangbx@cae.cn

***Abstract*—Anonymous communication is a challenging task in resource constrained wireless sensor networks (WSN). However, anonymity is important for many sensor networks, in which we want to conceal the location and identify of important nodes (such as source nodes and base stations) from attackers. Existing WSN anonymous protocols either cannot achieve complete anonymity, or have large computation and/or storage overheads. In this paper, we present an efficient anonymous communication protocol for sensor networks. Our protocol can achieve sender/source anonymity, communication -relationship anonymity, and the base station anonymity simultaneously, while having small overheads on computation, storage and communication.**

Keywords - anonymous communication; wireless sensor networks; lightweight; security

I. INTRODUCTION

A wireless sensor network (WSN) typically consists of many low power and resource-constrained sensor nodes. There are many applications of WSNs ranging from military to civilian in nature. Due to its important, security in WSN has been a topic of intensive study in the last few years. However, an important security issue in WSN – anonymous communication has not been studied in great details.

An efficient anonymous communication protocol for WSN can prevent attackers from identifying and then capturing important nodes by hiding their real identities. A global attacker may locate sensor nodes using techniques such as angle of arrival, signal strength, and so on[4]. Moreover, if an attacker knows the identity and location of important nodes, he will select to compromise more important nodes, get much information from them and cause more damages to the network. Different from wired networks and other wireless networks such as ad hoc networks [2], WSN is a multi-sender and one receiver network, where all sensor nodes send data to the base station. Based on the communication relationship between neighboring nodes, attackers may be able to infer the location of a source node and the base station. A base station is the center of a WSN and its identity and location should be hided from attackers. Also, the location of the source node reveals the location of the event. Therefore, an efficient anonymous communication protocol is essential for WSN, and it should be able to provide sender anonymity, communication relationship anonymity and the base station anonymity. Efficiency is also important since sensor nodes have very limited resources.

Anonymity is an important security issue in wireless and wired networks. However, anonymity in WSN has not been studied in great details [2]. The anonymous protocols designed for ad hoc networks are not suitable for WSN due to the high computation and communication overheads [8]. Misra *et al.* [5] proposed two anonymous schemes for clustered wireless sensor networks, namely SAS and CAS. However, SAS and CAS can't achieve sender anonymity, because the end-to-end anonymous identity used by each source is unchanged and thus the attacker can trace the source only by compromising a node along the route.

Nezhad *et al.* [6] proposed DCARPS to provide the base station anonymity in WSN. A tree structure is used for communications for DCARPS. Neighbors of the base station use the same node ID to send messages. Different from the base station, neighbors of a sensor may use different node IDs to send messages. Thus, base station anonymity is destroyed. Denote C as the average number of neighbors of each node. If m of C ($m<C$) neighboring nodes of one node use the same ID to send messages, they share the same parent node. Their parent can be found in the overlap area within the communication ranges of the m nodes. Hence, the parent-child relationship is exposed and so does the communication-relationship among sensor nodes.

Sheu *et al.* [8] proposed an anonymous communication protocol in WSN, namely APR. In the routing setup stage, source node finds the path to the base station by broadcast. However, since the broadcast messages include two constant fields, attackers can easily identify broadcast messages. Then attackers can infer the location of the source node through the transmission timing of broadcast messages from different nodes.

In our research, we consider heavy-traffic sensor networks with frequent events happening. Heavy-traffic sensor networks are common, for example, those used in inventory tracking, environment monitoring [5], and people-centric sensing systems [7].

In this paper, we present an efficient Anonymous Communication (AC) protocol for sensor networks. The AC protocol consists of two schemes: anonymous one-hop communication and anonymous end-to-end communication scheme. Our schemes can be combined with any existing routing algorithm, to ensure that node discovery, route requests, and route replies use pseudonyms and the true node identity is kept secret. Theoretical analysis shows that AC

978-1-4244-9268-8/11/$26.00 ©2011 IEEE

can provide a wide range of anonymities including the sender anonymity, communication-relationship anonymity and base station anonymity while having small overheads on computation, storage and communication.

II. NETWORK AND ATTACK MODEL

We consider a network of many randomly distributed small wireless sensors. The base station looks the same as sensor nodes and it is placed randomly.

We assume attackers are more powerful than senor nodes. Attackers not only have the global eavesdropping ability but also can compromise some senor nodes and mount active attacks. Assume attackers have the following capabilities:

- **Resource-rich**: Attackers have sufficient energy resource, adequate computation capability and enough memory. They can thus locate a node by measuring the arrival angle of its packet or the strength of the signal.
- **Passive attack**: Several attackers may be scattered throughout the network and collaboratively eavesdrop on communications among sensor nodes.
- **Active attack**: Attackers may physically capture sensor nodes, control them and mount attacks such as replay, and selective forwarding. The compromised nodes may be located anywhere in the network. However, attackers won't be able to compromise a lot of nodes in a short time period without being detected, as several methods (e.g., [2]) could detect such attack and take defensive actions accordingly.

III. THE ANONYMOUS COMMUNICATION PROTOCOL

A. Overview

Each sensor node finds the link direction (towards the base station) between itself and its neighbors during the deployment stage by a broadcast from the base station. Then messages can be sent to the base station hop-by-hop by the link direction. After that, our AC protocol utilizes an anonymous one-hop communication scheme and an anonymous end-to-end communication scheme to achieve the source anonymity, the communication relationship anonymity and the base station anonymity.

B. Network Set-up Phase

As in several literatures (e.g., [10]), we assume the network is secure (e.g., no attacks) for a short time interval after sensor nodes are deployed. During this period the communication among sensor nodes are secure. We also assume that sensors may use a secure location discovery service (e.g., [4]) to estimate their locations, and no GPS receiver is required at each node.

Before deployment, each node i is preloaded with several parameters: random number α_i, hashing function H_1 and H_2, ID of the node ID_i, pair-wise key k_i (shared with the base station). We summarize the notations in Table I.

AC utilizes the base station broadcast scheme [1]. After broadcast, every node, say i, knows the smallest hops between itself and the base station. After that, i creates a global anonymous identity - AI_i for itself. AI_i is first computed by hashing the values of $ID_i \oplus \alpha_i$ using H_1, where $\oplus$ stands for EXCLUSIVE OR operation (refer to (1)).

$$AI_i = H_1(ID_i \oplus \alpha_i) \quad (1)$$

TABLE I. LIST OF NOTATIONS

Notation	Definition
C	The average number of neighbors for each node
D	raw data
N_{nei}^i	Neighbors of node i
$Hop_{i,bs}$	The smallest hops between node i and the base station
T_i	Neighboring information table of node i
$linkdir_{i\to j}$	Link direction from node i to node j
α_i	A random number shared between node i and the base station
$\alpha_{i\leftrightarrow j}$	A random number shared between node i and node j
k_i	A pair-wise key shared between node i and the base station
$k_{i\leftrightarrow j}$	A pair-wise key shared between node i and node j
$E_k(D)$	Data D encrypted by pair-wise key k
AI_i	The global anonymous identity of node i shared between i and the base station
$OHAI_{i\leftrightarrow j}$	One-hop anonymous identity shared between node i and node j
$AAI_{i\leftrightarrow j}$	An anonymous acknowledgement identity shared between node i and node j
S_{AJ}	The global identities obtained by decrypting messages at compromised nodes

Then, node i exchanges its information such as, ID_i, k_i, α_i and $Hop_{i,bs}$ with its neighbors. After that, j which is one of i's neighbors, calculates a new random number $\alpha_{j\leftrightarrow i}$ and a new pair-wise key $k_{j\leftrightarrow i}$ between node i and j as in (2) and (3). Node j also establishes one-hop anonymous identity, $OHAI_{j\leftrightarrow i}$ by (4). We set the link direction $linkdir_{i\to j}$ as follows: if $Hop_{i,bs}>Hop_{j,bs}$, then set $linkdir_{i\to j}$ as *uplink*; if $Hop_{i,bs}==Hop_{j,bs}$, then set $linkdir_{i\to j}$ as *randlink*; if $Hop_{i,bs}<Hop_{j,bs}$, then set $linkdir_{i\to j}$ as *downlink*.

$$\alpha_{j\leftrightarrow i} = H_1(ID_i \oplus ID_j) \quad (2)$$

$$k_{j\leftrightarrow i} = H_2(k_i + k_j + \alpha_i + \alpha_j) \quad (3)$$

$$OHAI_{j\leftrightarrow i} = H_1(\alpha_i \oplus \alpha_j) \quad (4)$$

Thereafter, node i creates a neighbor information table T_i which contains entries for links between itself and its one-hop neighboring nodes. Each entry has the fields of $OHAI_{i\leftrightarrow j}$, $AAI_{i\leftrightarrow j}$, $\alpha_{i\leftrightarrow j}$, $k_{i\leftrightarrow j}$ and $linkdir_{i\to j}$. In order to save storage space and remove unnecessary privacy information, each node i deletes the following information *of* every neighbor (say j): ID_j, k_j, $Hop_{j,bs}$.

C. The Anonymous end-to-end Communication Scheme

This scheme is performed right after the network set-up phase. The anonymous end-to-end communication scheme is used to provide the source anonymity and the base station anonymity. Each time a source node wants to send a message to the base station, it uses a global anonymous identity which can only be computed by itself and the base station. The global anonymous identity gets updated for every new

message from the same source node. After a message has been delivered to the base station, both the source node and the base station update the global anonymous identity of the source node.

Now we discuss the details of the source node anonymity scheme. When a source node wants to send a message to the base station, the source node uses a global anonymous identity to represent its real identity and changes it after sending every message. For example, if a source node *i* wants to send sensing data *D* to the base station, it first chooses a forwarding node using any existing secure routing protocol. If *j* is the forwarding node, then *i* sends to *j* a message with the form:

$$M_{i\to j} = OHAI_{i\leftrightarrow j} || E_{k_{i\leftrightarrow j}}(AI_i || E_{k_i}(D) || AAI_{i\leftrightarrow j})$$

where $AAI_{i\leftrightarrow j}$ is a temporal anonymous ACK identity randomly generated by *i*, $AAI_{i\leftrightarrow j}$ will be used in an ACK message in case of message loss and transmission errors. Different from other anonymous identities such as $OHAI_{i\leftrightarrow j}$, $AAI_{i\leftrightarrow j}$ is a temporal identity which is used for one time anonymous acknowledgement and there's no need for updating between neighboring nodes. Thus, $AAI_{i\leftrightarrow j}$ can be generated randomly. Afterwards, AI_i is updated by (5) and messages are sent hop-by-hop by an anonymous one-hop communication scheme (discussed in subsection D).

$$AI_i = H_1(AI_i \oplus \alpha_i) \qquad (5)$$

If the base station receives a message from its neighboring node, it checks the source node's identity AI_i and then decrypts the data *D* by the corresponding pair-wise key k_i. Then the base station updates AI_i as in (1). The source node will update AI_i before it wants to send out the next message. In order to behave like a sensor node, the base station also sends an ACK message back to its neighboring node.

D. The Anonymous one-hop Communication Scheme

This scheme is mainly used to conceal the data communication relationship between neighboring nodes. When a sensor node *j* receives a message with the form $M_{i\to j} = OHAI_{i\leftrightarrow j} || E_{k_{i\leftrightarrow j}}(AI_i || E_{k_i}(D) || AAI_{i\leftrightarrow j})$, *j* compares the $OHAI_{i\leftrightarrow j}$ with all the saved anonymous identities. If there is a match, then *j* decrypts this message with the corresponding key $k_{i\leftrightarrow j}$. As mentioned above, $AAI_{i\leftrightarrow j}$ is an anonymous identity for ACK and *j* uses it to send an anonymous ACK message back to *i* with the form $AAI_{i\leftrightarrow j} || D_{rand}$, where D_{rand} is a random padding that makes the length of ACK the same as that of a data message. On receiving the ACK message, *i* updates $OHAI_{i\leftrightarrow j}$ using (6). If node *i* does not receive a correct ACK message after a timeout period, *i* retransmits the message.

$$OHAI_{i\leftrightarrow j} = H_1(OHAI_{i\leftrightarrow j} \oplus \alpha_{i\leftrightarrow j}) \qquad (6)$$

After the communication from node *i* to *j*, node *j* chooses the next forwarding node *r* using the aforementioned routing protocol, encrypts the payload data by $k_{r\leftrightarrow j}$ and sends the message to *r* with the form $M_{j\to r} = OHAI_{j\leftrightarrow r} || E_{k_{j\leftrightarrow r}}(AI_i || E_{k_i}(D) || AAI_{j\leftrightarrow r})$. Afterwards, *j* updates $OHAI_{j\leftrightarrow r}$ in the same way as in (6).

IV. SECURITY ANALYSIS

In this section, we analyze the anonymity performance of the AC protocol under both global passive attacks and active attacks.

A. Sender Anonymity

In AC, as each source node uses different global anonymous identity every time when it sends a message to the base station, and the global anonymous identity is encrypted hop by hop, a global passive attacker can only observe a few transmissions in the network but they can't find the source node.

Active attackers are able to compromise some sensor nodes and then decrypt messages received by these nodes. After message decryption, they can get S_{AJ}, where S_{AJ} denotes the global identities from these messages. According to Theorem 1 (below), even with S_{AJ}, it is hard for an active attacker to find out the source node. Thus, sender anonymity is ensured.

Theorem 1. It is hard for an attacker to find the source node even under both passive and active attacks.

Proof. Suppose node *i* is the source node, and *i* sends out its message, say $M_{i\to j}$, to *j*. In order to protect the sender anonymity, *i* uses a global anonymous identity, say AI_i for end-to-end communication. If $M_{i\to j}$ is not forwarded by any compromised node before it arrives at the base station, it is impossible for an attacker to get AI_i without $k_{i\leftrightarrow j}$. Hence, an attacker can't trace the source node. Let N_{comp} denotes the set of compromised nodes. If $i \notin N_{comp}$ and $M_{i\to j}$ is forwarded by some compromised nodes before reaching the base station, then $AI_i \in S_{AJ}$. As node *i* uses different global identities each time *i* sends out a message, the attacker doesn't know whether one identity in S_{AJ} comes from the same node or different nodes. Hence, attackers can't idenfity the source node.

In the worst case, if $i \in N_{comp}$, attackers can find *i* is the source node by comparing *i*'s current global identity with the hashing results computed by equation (5). To be more precise, assume that the average number of messages captured by the attacker from a compromised node is γ and there are $|N_{comp}|$ compromised nodes. In order to get the global anonymous identities from these captured messages, the attacker need decrypt each captured message, and hence a total of Z_1 decryptions as listed in (7). If a compromised node was a source node, it must have used a global identity included in S_{AJ}. The attacker tries to find the source node with $|S_{AJ}|$ hashing operations for each node using (5). The total number of hashing operations is Z_2 as listed in (8).

$$Z_1 = |N_{comp}|\gamma \qquad (7)$$

$$Z_2 = |S_{AJ}||N_{comp}| \qquad (8)$$

Suppose a WSN has *N* sensor nodes, and among them δ nodes are source nodes. If the attacker randomly compromises sensor nodes in the WSN, it has a δ/N chance

of compromising a source node. If δ is large, the probability that a source node is compromised is large. However, γ is also large and the computation cost for the attacker is high. If δ is small, the attacker need less computations. However, the probability of compromising a source node is also small. To sum up, it is hard for an attacker to trace the source node. □

B. Communication Relationship Anonymity

Communication relationship should be protected, otherwise an attacker may infer the location of a source node or the base station [3]. In AC, no constant identity appears in any two messages after the network set-up stage. As each message uses different anonymous identity and is encrypted hop by hop, each message has different appearances after every hop. Passive attackers can only observe a lot of transmissions but they can't find out the communication relationship.

If node i receives a message and transmits it to a neighbor j with the form m_i at time t. Then node j transmits it to its neighbor r with a different form m_j, where $i \neq j \neq r$. Only if an attacker can tell that m_i and m_j are the same message, the communication relationship between i and j is discovered. Let $M=\{m_1,m_2,...,m_p\}$ denotes that messages sent by all the neighbors of node i during time interval t', where $t' \in (t, t+\varepsilon]$ and ε is the upper bound of transmission latency. According to the anonymous one-hop communication scheme, m_i includes anonymous identity and encrypted message body, say m_i *.identity* and m_i *.body*. If any of two nodes in $\{i, j, r\}$, say i and j, are not compromised, an attacker doesn't know $OHAI_{i\leftrightarrow j}$ and he can't decrypt the message transmitted from i to j because he doesn't know $k_{i\leftrightarrow j}$. Hence, an attacker can't find any $m_j \in M$ which satisfies that $D_{k_{i\leftrightarrow j}}(m_i.body) == D_{k_{j\leftrightarrow r}}(m_j.body)$, where $i \neq j \neq r$ and $D_k(m)$ denotes the decrypted data by key k. Thus, the communication relationship anonymity is satisfied.

C. Base Station Anonymity

As there is no information about the base station included in any message and all messages are indistinguishable, passive attackers can't find out the location of the base station by captured messages. Moreover, the base station behaves like normal sensor nodes. As for active attackers, they can compromise several nodes in a short time and get all the information from these compromised nodes and find out the communication relationship between neighboring nodes. However, due to the secure routing protocols such as probabilistic forwarding node selection scheme, it is hard for an attacker to find out who the base station is even if he can infer the communication relationship between two compromised neighboring nodes. Thus, the base station anonymity is satisfied.

We summarize the anonymous performance of our AC protocol and several existing anonymous schemes in Table II. It can be seen from Table II that only AC achieves complete anonymity, while other schemes cannot.

TABLE II. COMPARISON OF ANONYMITY PERFORMANCE

Anonymous communication protocols	Sender anonymity	Communication relationship anonymity	Base station anonymity
SAS	unsatisfied	satisfied	satisfied
CAS	unsatisfied	satisfied	satisfied
APR	unsatisfied	satisfied	unsatisfied
DCARPS	satisfied	unsatisfied	unsatisfied
AC	**satisfied**	**satisfied**	**satisfied**

V. PERFORMANCE ANALYSIS

In this section, we analyze the performance of the AC protocol, including the storage cost, computation cost and communication cost.

For anonymous end-to-end communication, each node i stores three parameters, AI_i, α_i and k_i for global anonymous identity generation and data encryption. For anonymous one-hop communication, for each of its neighbor j, i has to store $OHAI_{i\leftrightarrow j}$ and $\alpha_{i\leftrightarrow j}$ to generate the one-hop anonymous identity. Node i also has to $k_{i\leftrightarrow j}$ for one-hop message encryption. The random numbers and H_1 are n_1 bits. Keys and H_2 are n_2 bits. There are three possible link directions, so the link direction can be represented by 2 bits.

Hence, the total storage requirement at each node is $3n_1+n_2+(2n_1+n_2)C+2$bits, where C denotes the average number of neighbors for each node. If $n_1=n_2=n$, then the memory cost is $4n+3nC+2$bits. For instance, in a WSN with 1000 nodes, let n=128 (e.g., MD5 has 128-bit output) and each node has an average of 30 neighbors. The memory requirement shall be, 4*128+3*128*30+2=12,034bits ≈ 1.5 KB. For a WSN with 5,000 nodes and a neighbor size of 100 nodes, the memory requirement is: 4*128+3*128*100+2 =38,914bits=4.8 KB. Hence, it is feasible to implement AC on a TelosB mote [5], which has a flash memory of size 1MB and RAM of 48KB.

AC is a lightweight protocol using hashing function and symmetric cryptography. In order to accept and forward a message, each node needs two hashing operation for one-hop anonymous identities' updating. Table III compares the storage and computation costs of our AC protocol with several existing anonymous communication protocols. Note that we do not include the computation cost of data encryption as data encryption cost of AC is the same as the existing anonymous communication protocols, e.g., [5, 6, 8].

TABLE III. PERFORMANCE COMPARISON

Anonymous communication protocols	Storage cost (bits)	Computation cost
SAS	$2nN+4nC+16$	Generating anonymous IDs from pseudonym space
CAS	$6n+7nC+16$	Two hashing operations and two encryption operations
APR	$9n+7nC+2N-2C-2$	At least six hashing operations
DCARPS	$3n$	No extra computation cost with constant IDs
AC	$\mathbf{4n+3nC+2}$	**Two hashing operations.**

Table III shows that DCARPS [6] performs the best, in terms of storage and computation cost. However, DCARPS

has the worst anonymity and security performance. DCARPS can't achieve the base station anonymity and the communication relationship anonymity under global passive attacks. Moreover, DCARPS can't defend active attacks such as replay attacks since all nodes uses the same identity for message sending and forwarding. Table III also shows that SAS has low computation cost because SAS creates anonymous identities from the pseudonym space, which has light computations. However, SAS requires much more storage space than our AC protocol.

Below, we analyze the communication cost of AC. Under AC, each node sends out a one-hop broadcast message to exchange information with its neighbors for neighboring table establishment, and the communication cost in the entire network for this message exchange is N (messages). In addition, once a node forwards a message, it should send an anonymous ACK message. The communication cost of the ACK message is θ which depends on the adopted routing protocol. This is because the same message will be transmitted by different hop numbers under different routing protocols. Anonymous protocols without considering reliable communications (such as SAS, CAS and DCAPRS) have no such extra overhead. Both SAS and CAS establish pairwise keys for any two neighboring nodes and have an extra communication cost P. Table IV compares the communication cost of several anonymous protocols, and it shows that DCARPS has the smallest communication cost. This is because in DCARPS each node uses a constant ID for both receiving and forwarding messages. Hence, DCARPS doesn't need to exchange messages in the network initialization stage. However, because of using constant IDs, DCARPS has the worst anonymity and security performance. Note that Table IV does not include the communication cost of initial broadcasting as initial broadcasting under AC is the same as other anonymous communication protocols.

TABLE IV. COMPARISON OF COMMUNICATION COST

Anonymous protocols	Communication cost
SAS	$P+N$
CAS	$P+N+N*n$
APR	$N+\theta$
DCARPS	No extra computation cost due to constant IDs
AC	$\boldsymbol{N+\theta}$

To sum up, Table III and IV and the above discussions show that our AC protocol achieves all three anonymities with low computation, storage, and communication costs.

VI. CONCLUSION

In wireless sensor networks, especially heavy-traffic sensor networks, anonymous communication protocols can be used to conceal the identity of important nodes, such as source nodes and the base station. However, existing anonymity schemes either can't achieve all the anonymity requirements (including sender anonymity, communication relationship anonymity and the base station anonymity), or have large computation/storage costs. In this paper, we presented an efficient Anonymous Communication (AC) protocol for sensor networks. The AC protocol includes two schemes: anonymous one-hop communication and anonymous end-to-end communication. Our performance analysis showed that the AC protocol provides all three anonymities while having small computation, storage, and communication costs.

ACKNOWLEDGMENT

This research was supported in part by the China National Basic Research Program (973 Program) under grants 2011CB302605 and 2007CB311101, the China National High Technology Research and Development Program (863 Program) under grant 2010AA012504 and 2011AA010705; and by the US National Science Foundation under grants CNS-0963578, CNS-1002974, CNS-1022552, and CNS-1065444, as well as the US Army Research Office under grant W911NF-08-1-0334.

REFERENCES

[1] J. Chen, B. X. Fang, L. H. Yin and S. SU, "A Source-Location Privacy Preservation Protocol in Wireless Sensor Networks Using Source-Based Restricted Flooding," *Chinese Journal of Computers*, vol.33, no. 9, pp. 1736-1747, 2010.

[2] J. C. Kao and R. Marculescu, "Real-time anonymous routing for mobile ad hoc networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC'07)*, 2007.

[3] L. Kang, "Protecting location privacy in large-scale wireless sensor networks," in *Proc. of IEEE International Conference on Communications (ICC'09)*, 2009.

[4] L. Lazos and R. Poovendran, "SeRLoc:Secure range-independent localization for wireless sensor networks," in *Proc. of ACM Workshop Wireless Security*, 2004.

[5] S. Misra and G. Xue, "Efficient anonymity schemes for clustered wireless sensor networks," *International Journal of Sensor Networks*, vol.1, no. 1, pp.50-63, 2006.

[6] 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.

[7] J. Shi, R. Zhang, Y. Liu and Y. Zhang, "Prisense: privacy-preserving data aggregation in people-centric urban sensing systems," in *Proc. of the IEEE INFOCOM 2010*, San Diego, CA, March 2010.

[8] J. P. Sheu, J. R. Jiang and C. Tu, "Anonymous Path Routing in Wireless Sensor Networks," in *Proc. of IEEE International Conference on Communications (ICC'08)*, 2008.

[9] M. Shao, Y. Yang, S. Zhu, and G. Cao, "Towards Statistically Strong Source Anonymity for Sensor Networks," in *Proc. of IEEE INFOCOM 2008*, pp. 51-55, 2008.

[10] Y. Zhang, W. Liu, W. Lou and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol.24, no.2, pp.247-260, 2006.

[11] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar and G. Cao, "Towards event source unobservability with minimum network traffic in sensor networks," in *Proc. of the first ACM Conference on Wireless Network Security*, 2008.