WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. (2011)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/wcm.1205

RESEARCH ARTICLE

An efficient anonymous communication protocol for wireless sensor networks

Juan Chen[1], Xiaojiang Du[2]* and Binxing Fang[1]

[1] School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
[2] Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, U.S.A.

ABSTRACT

Anonymous communication is very important for many wireless sensor networks, because it can be used to hide the identity of important nodes, such as the base station and a source node. In sensor networks, anonymous communication includes several important aspects, such as source anonymity, communication relationship anonymity, and base station anonymity. Existing sensor network anonymous schemes either cannot achieve all the anonymities or have large computation, storage, and communication overheads. In this paper, we propose an efficient anonymous communication protocol for sensor networks that can achieve all the anonymities while having small overheads on computation, storage, and communication. We compare our anonymous communication protocol with several existing schemes, and the results show that our protocol provides strong anonymity protection and has low overheads. Copyright © 2011 John Wiley & Sons, Ltd.

KEYWORDS

anonymous communication; sensor networks; security

***Correspondence**

Xiaojiang Du, Department of Computer and Information Sciences, Temple University Philadelphia, PA 19122, U.S.A.
E-mail: dux@temple.edu

1. INTRODUCTION

A wireless sensor network (WSN) consists of many sensor nodes spatially distributed in a certain area [1–3]. Sensor nodes sense environmental conditions, and the sensing data is sent hop by hop to a base station. There are many applications of WSNs ranging from military to civilian in nature. Security is an essential issue for WSNs deployed in hostile environments, such as military battlefields. In the past few years, WSN security has been a topic of intensive study. However, anonymity, as an important security issue in WSN, has not been well studied yet.

An effective anonymous communication protocol for WSN can prevent attackers from identifying (and then capturing) important nodes (such as source and base station). Global attackers may locate a node by using localization techniques such as triangulation, angle of arrival, and signal strength [4–6]. In WSN, sensor nodes use their identities for message receiving and forwarding. Besides, the base station knew where the event happened by the source nodes' identity. So, if each node uses its constant identity for communication, attackers can trace the source node or the base station by analyzing the identities. If an attacker knows the identity and location of each node, he will be able to selectively compromise more important nodes, which will allow him to get much more information and/or cause more damages to the network. Different from wired networks and many other types of wireless networks such as ad hoc networks [7–11], WSN is a many-to-one network, where all sensor nodes send data to one base station. With the communication relationship among neighboring nodes, attackers may be able to infer the location of a source node and the base station [12]. The graphical location of a sender, namely the source node, reveals the event occurrence. The base station is the center of a sensor network. It would cause much damage if the identity/location of an event source or the base station is revealed. Therefore, an effective anonymous communication protocol is essential for WSN security, and it should at least achieve the following three kinds of anonymities: sender anonymity, communication relationship anonymity, and the base station anonymity. Because typical sensor nodes have very limited resources in batteries, computational capabilities, and storage [13–16], the anonymous communication protocol should be efficient, that is, with small computation and storage requirements.

In this paper, we propose an efficient anonymous communication (EAC) protocol for sensor networks. Our contributions are threefolds:

(1) We show that none of the existing WSN anonymous communication protocols can achieve all three kinds of anonymities.
(2) We propose an efficient anonymous communication protocol, EAC, that guarantees the three kinds of anonymities: sender, communication relationship, and base station anonymity.
(3) EAC is lightweight and only uses hashing function and symmetric cryptography. Compared with existing anonymous communication protocols, EAC provides full anonymity while incurring low storage computation and communication costs.

The rest of the paper is organized as follows. In Section 2, we state the anonymity problem in WSN. After that, we discuss and analyze the anonymity properties of several related works in Section 3. In Section 4, we present our EAC protocol. We provide the security and performance analyses in Sections 5 and 6, respectively. Finally, we conclude this paper in Section 7.

2. PROBLEM STATEMENT

Anonymity in sensor networks means preventing a third party from knowing the identity of the two primary parties in a communication. Each node plays a different role in the network. A source node is a sensor close to the event spot and generates messages to the base station. Normal nodes on route are responsible for message relay. The base station is the controller of the network and carries out many tasks. An important node such as a source node or the base station plays a critical role in the network. A smart attacker may first try to identify important nodes and then compromise these nodes, which can cause great damage to the network.

Node identity anonymity and anonymous communication can prevent the aforementioned selectively attacks. Anonymity in the context of a sensor network includes sender anonymity, receiver anonymity, and unlinkability between the sender and the receiver. With the aforementioned anonymities, an adversary is not able to determine the sender and receiver's identities by reading a message intercepted from the network or through reading messages forwarded by a compromised sensor node. The adversary cannot determine whether two transmissions (from different nodes) are relaying the same message either.

In this paper, we study the important issue of node identity anonymity and anonymous communication. We propose several effective anonymous schemes that can hide node identity and relay messages between sensors and the base station. Our anonymous schemes can be used with any existing sensor routing protocols. We do not propose any new routing protocol in this paper.

3. RELATED WORK

A number of literatures (e.g., [17–19]) have studied anonymity in ad hoc networks. However, the anonymous protocols designed for ad hoc networks are not suitable for WSNs because of the large computation and communication overheads [5]. Misra and Xue [20] proposed two anonymous schemes for clustered WSNs, namely Simple Anonymity Scheme (SAS) and Cryptographic Anonymity Scheme (CAS). The former uses a pool of pseudonyms for anonymous identity generation, and the latter uses hashing function and symmetric cryptography. However, if a sensor node along the routing path is compromised, then neither SAS nor CAS achieves sender anonymity. The reason is given as follows.

Under SAS and CAS, the source node i sends a message to the base station through a neighbor node j with the form $A_{ij} \| \| E_k(A_{i,,\mathrm{BS}} \| \| \mathit{data})$, where A_{ij} and $A_{i,,\mathrm{BS}}$ represent the one-hop anonymous identity and the end-to-end anonymous identity, respectively. Although A_{ij} changes hop by hop, $A_{i,,\mathrm{BS}}$ does not change. Furthermore, the A_{ij} used by the first hop (i.e., from the source node i) is the same as $A_{i,,\mathrm{BS}}$. If a node (suppose j) along the pass is compromised, then the attacker knows $A_{i,,\mathrm{BS}}$, and he can find out the source node, which is the node whose one-hop anonymous identity is $A_{ij} = A_{i,,\mathrm{BS}}$.

Nezhad *et al*. [21] proposed Destination Controlled Anonymous Routing Protocol for Sensornets (DCARPS) to protect the base station anonymity in WSN. Under DCARPS, each node has two constant anonymous IDs, one for message receiving and another for message forwarding. However, later, we will show that DCARPS cannot provide the base station anonymity, nor the communication relationship anonymity.

In DCARPS, a broadcast tree structure is used for communications. The one-hop neighbors of the base station use the same anonymous ID to send messages to the base station, as the base station is the parent node of all these one-hop nodes. However, for any other sensor node (besides the base station), its neighbors may use different anonymous IDs to send out messages, because the probability of all neighbor nodes having the same parent node is very small. With the aforementioned differences, an eavesdropper can find out the base station.

In DCARPS, sensors use a broadcast tree to transmit data to the base station. Consider an example in Figure 1, node D is the parent node of nodes A, B, and C. When A, B, and C send data packets to D, they use the same anonymous receiving ID. A global observer can easily find out node D's location, which is an overlapping area of the three transmissions. Hence, the parent–child relationship is exposed and so is the communication relationship between the nodes.

Sheu *et al*. [5] proposed a new anonymous communication protocol for WSN, namely Anonymous Path Routing (APR), that includes rooting setup stage and anonymous communication stage. In the first stage, source node finds the path to the base station by broadcast. However, as

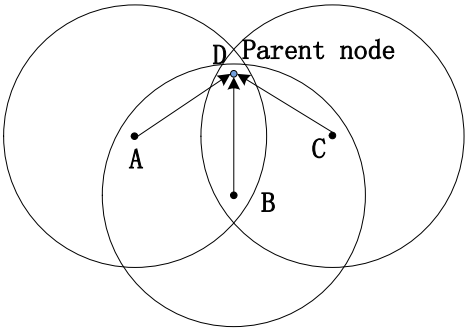

Figure 1. Localizing the parent node.

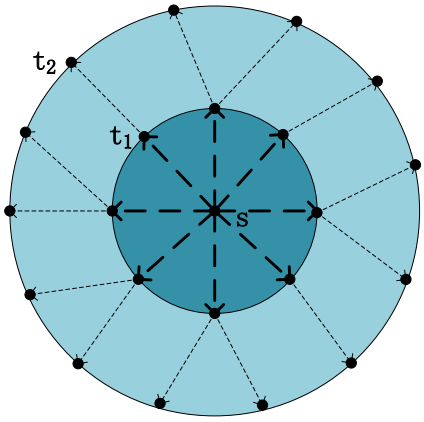

Figure 2. Clustering nodes according to the timing of broadcast messages.

the broadcast flag field and the receiver identity field of the broadcast message do not change during the broadcast, attackers can easily detect a broadcast message and trace the message. Figure 2 shows a broadcast initiated by node s. With the two invariant fields, an attacker can detect broadcast messages of the same session and record the location and time of each relay of the message. In Figure 2, the attacker could plot two circles according to the message timings, and the source node is located in the center of the circles. To sum up, without anonymous broadcast in the path discovery process, APR cannot achieve sender anonymity. As the network controller, the base station often broadcast messages, so without anonymous broadcast, the base station can be easily identified. Hence, APR cannot provide base station anonymity either.

4. THE EFFICIENT ANONYMOUS COMMUNICATION PROTOCOL

4.1. Network and attack models

We envision a network of hundreds of small wireless sensors that are randomly and uniformly distributed in a field. A base station collects sensing data from the sensors. The attacker nodes have much stronger capabilities than the senor nodes. The attacker nodes not only has global eavesdropping ability but also can compromise some senor nodes and launch active attacks. Specifically, the attacker nodes have the following capabilities:

- **Resource rich**: Attackers have sufficient energy supply, computation capability, and storage memory. They are able to locate a sensor node by measuring the arrival angle and the signal strength of its packets.
- **Passive attacks**: Several attacker nodes may be scattered throughout the network and collaboratively eavesdrop on communications among sensor nodes.
- **Active attacks**: Attackers can physically capture sensor nodes, control them, and mount attacks such as denial-of-service, replay, and forging attacks. The compromised nodes may be located anywhere in the network. However, attackers cannot compromise too many sensor nodes within a short period, as several methods [22–24] could detect this attack and take defensive actions accordingly [25].

4.2. Network predeployment

Before deployment, each sensor node i is preloaded with several parameters: random numbers β_i and α_i, hashing functions H_1 and H_2, node ID—ID_i, and pairwise keys k_i and k_b^i. For it to be simple, suppose that the random numbers, ID_i, and H_1 are n_1 bits. Keys and H_2 are n_2 bits. We summarize the notations in Table I.

4.3. Network initialization

As in several literatures (e.g., [25]), we assume the network is secure (e.g., no attacks) for a short period after sensor nodes are deployed. During this period, the communications among sensor nodes are secure. We also assume that sensors may use a secure location discovery service (e.g., [26]) to estimate their locations and that no GPS receiver is required at each node.

Our EAC protocol utilizes existing broadcast schemes originated from the base station [4]. After the broadcast process, each node (say, i) can find out the smallest hop count between itself and the base station. After that, i creates two anonymous identities: AI_i and BAI_i, namely the global anonymous identity and the anonymous broadcast identity, respectively. Initially, AI_i and BAI_i are computed according to Equation (1), where $\oplus$ stands for *exclusive or* operation.

$$\begin{cases} AI_i = H_1(ID_i \oplus \alpha_i) \\ BAI_i = H_1(ID_i \oplus \beta_i) \end{cases} \quad (1)$$

Then, i exchanges information with its neighbors by a one-hop broadcast message <BRO, $h = 1$, ID_i, k_i, k_b^i, α_i, β_i, $Hop_{i,\mathrm{bs}}$>, where BRO, h, and $Hop_{i,\mathrm{bs}}$ stand for the broadcast flag, the number of hops by broadcast, and the smallest hop count between i and the base station,

Table I. List of notations.

Notation	Definition
N	Total number of sensor nodes in the network
N^i_{nei}	Neighbors of node i
$Hop_{i,\text{bs}}$	The smallest hop count between node i and the base station
T_i	Node $i's$ neighbor information table
$linkdir_{i\to j}$	Link direction between node i to node j
α_i	A random number shared between node i and the base station
β_i	A random number shared between node i and all its neighbors
$\alpha_{i\leftrightarrow j}$	A random number shared between node i and node j
k_i	A pairwise key shared between node i and the base station
$k_{i\leftrightarrow j}$	A pairwise key shared between node i and node j
k^i_b	The broadcast key of node i
$E_k(D)$	Encrypting data D by key k
AI_i	The global anonymous identity of node i, only known by i and the base station
$OHAI_{i\leftrightarrow j}$	One-hop anonymous identity shared between node i and node j
BAI_i	An anonymous broadcast identity of node i
AAI_i	An anonymous acknowledgement identity generated by node i
S_{AJ}	Global identities obtained by decrypting messages from compromised nodes

respectively. On receiving a one-hop broadcast message (as previously) from its neighbor j, i calculates a new random number $\alpha_{i\leftrightarrow j}$ and a new pairwise key $k_{i\leftrightarrow j}$ between nodes i and j by hashing the values of $ID_i \oplus ID_j$ and $k_i + k_j + \alpha_i + \alpha_j$ using different hashing functions, respectively, as shown in Equations (2) and (3). Node i also establishes three anonymous identities, BAI_j, $OHAI_{i\leftrightarrow j}$, and AAI_i. The first BAI_j is calculated by Equation (1). The first value of $OHAI_{i\leftrightarrow j}$ and AAI_i are calculated by hashing the values of $\alpha_i \oplus \alpha_j$ and ID_i, respectively (refer to Equation (4)). On the basis of the smallest hop counts from nodes i and j to the base station, say, $Hop_{i,\text{bs}}$ and $Hop_{j,\text{bs}}$, $linkdir_{i\to j}$ is determined as follows: if $Hop_{i,\text{bs}} > Hop_{j,\text{bs}}$, then set $linkdir_{i\to j}$ as an uplink; if $Hop_{i,\text{bs}} == Hop_{j,\text{bs}}$, then set $linkdir_{i\to j}$ as a randlink; and if $Hop_{i,\text{bs}} < Hop_{j,\text{bs}}$, then set $linkdir_{i\to j}$ as a downlink.

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

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

$$\begin{cases} OHAI_{i\leftrightarrow j} = H_1\left(\alpha_i \oplus \alpha_j\right) \\ AAI_i = H_1(ID_i) \end{cases} \tag{4}$$

After one-hop broadcast, node i creates a neighbor information table T_i that contains entries for links between itself and its one-hop neighboring nodes. Each entry has the fields $BAI_j, OHAI_{i\leftrightarrow j}, AAI_j, \alpha_{i\leftrightarrow j}, \beta_i, k^i_b, k_{i\leftrightarrow j}$, and $linkdir_{i\to j}$. The initial neighboring information table T_i of node i is shown in Table II. In order to save storage space and remove privacy information, node i deletes the following information: $ID_i, Hop_{i,\text{bs}}$, and the one-hop broadcast message $<BRO, h = 1, ID_j, k_j, k^j_b, \alpha_j, \beta_j, Hop_{j,\text{bs}}>$ of each neighbor j.

Table II. The neighboring table at node i.

Anonymous broadcast identity	BAI_j	...
One-hop anonymous identity	$OHAI_{i\leftrightarrow j}$	...
Anonymous acknowledgement identity	AAI_j	...
Shared random number	$\alpha_{i\leftrightarrow j}$	...
Shared broadcast random number	β_j	...
Shared broadcast key	k^j_b	...
Shared one-hop key	$k_{i\leftrightarrow j}$	...
Link direction	$linkdir_{i\to j}$	...

4.4. Four efficient schemes for anonymous communications

Our EAC consists of four efficient schemes: anonymous data sending, anonymous data forwarding, anonymous broadcast, and anonymous acknowledgement (ACK). We present the details of the four schemes in Sections 4.4.1–4.4.4, respectively. We describe a secure node addition scheme in Section 4.4.5.

4.4.1. Anonymous data sending.

This scheme is performed right after the network is initialized. Considering the source node's anonymity, when a source node wants to send a message to the base station multihops away, the source node uses a global anonymous identity to represent its real identity and changes anonymous identity after every message sending. For example, if source node i wants to send its sensed data D to the base station, it first chooses a forwarding node on the basis of a probabilistic forwarding node selection scheme, which is described later. A node, say, i, classifies its neighbors into three sets according to their link direction values—$linkdir_{i\to j}$. Then i selects a forwarding node from these three sets with different probabilities. To ensure

that messages can be delivered to the base station, neighboring nodes whose link direction value is uplink should be selected with a high probability, more than 0.5. If j is the selected node, then i sends j a message with the form

$$M_{i\to j} = OHAI_{i\leftrightarrow j} \| E_{k_{i\leftrightarrow j}} \times (AI_i \| E_{k_i}(D) \| H(AI_i \| E_{k_i}(D))) \quad (5)$$

Afterwards, node i updates AI_i by hashing the value of $AI_i \oplus \alpha_i$ (refer to Equation (6)). Both i and j update $OHAI_{i\leftrightarrow j}$ by hashing the value of $OHAI_{i\leftrightarrow j} \oplus \alpha_{i\leftrightarrow j}$ (refer to Equation (6)). During the aforementioned process, a global eavesdropper only observes a broadcast transmission in the neighborhood of node i and j, but it cannot tell who the sender (or receiver) is because the aforementioned message $M_{i\to j}$ (Equation (5)) does not reveal any node identity information. If j is the base station, then j decrypts the payload by $k_{i\leftrightarrow j}$ and obtains AI_i. Only the base station knows which sensor node is the owner of AI_i. The base station knows k_i, and it can obtain the sensing data D. After that, the base station updates node i's global anonymous identity AI_i by Equation (5), which will be used for the next message from node i.

$$\begin{cases} AI_i = H_1(AI_i \oplus \alpha_i) \\ OHAI_{i\leftrightarrow j} = H_1(OHAI_{i\leftrightarrow j} \oplus \alpha_{i\leftrightarrow j}) \end{cases} \quad (6)$$

4.4.2. Anonymous data forwarding.

This scheme is used to conceal the data forwarding relationship among neighboring nodes. When a sensor node j receives a message with the form $OHAI_{i\leftrightarrow j} \| E_{k_{i\leftrightarrow j}}(AI_i \| E_{k_i}(D) \| H(AI_i \| E_{k_i}(D)))$, j compares the $OHAI_{i\leftrightarrow j}$ field with the anonymous identities in its table T_j. If there is no match, this means the message is not for j, and j drops the message. If the $OHAI_{i\leftrightarrow j}$ field matches an entry of T_j, then it means that this message is for node j, and j uses the corresponding shared one-hop key $k_{i\leftrightarrow j}$ to decrypt the message. Node j chooses the next forwarding node r by using the probabilistic forwarding node selection scheme (see Section 4.4.1), encrypts the payload data by $k_{j\leftrightarrow r}$, and sends the message to r with the form $OHAI_{j\leftrightarrow r} \| E_{k_{j\leftrightarrow r}}(AI_i \| E_{k_i}(D) \| H(AI_i \| E_{k_i}(D)))$. Afterwards, j updates both the one-hop anonymous identities $OHAI_{i\leftrightarrow j}$ and $OHAI_{r\leftrightarrow j}$.

4.4.3. Anonymous broadcasting.

This scheme can be applied to both unicast and multihop broadcast. It includes two subschemes: anonymous broadcast and probabilistic latency-based transmission. With anonymous broadcast, the attacker, even compromised some nodes, cannot distinguish broadcast messages from other (e.g., unicast) messages. If an attacker can identify broadcast messages, they can infer the location of the broadcast-originating node who originates the broadcast according to the transmission time order of different node. Usually, as the controller of the network, base station originates broadcast frequently. So, without anonymous broadcast, the base station can be located and thus can be attacked.

The anonymous broadcast scheme is presented in the following. First, a source node i broadcasts a message to its one-hop neighbors. Node i encrypts data with key k_b^i and uses an anonymous broadcast identity for the message, that is, $M_B = BAI_i \| k_b^i(D \| H(D))$. Then i updates its anonymous broadcast identity by hashing the value of $BAI_i \oplus \beta_i$ (refer to Equation (7)). To avoid accepting duplicate messages from the same broadcast, i also updates neighbors' anonymous broadcast identities according to Equation (7). When node j receives M_B, j, it checks if there is any entry in T_j matching BAI_i. If so, j adds a random delay before forwarding the message. The random delay is used to hide the timing order of transmissions. Then, j decrypts the payload using k_b^i and encrypts it by k_b^j. Afterwards, j replaces the anonymous broadcast identity by BAI_j and broadcasts to its one-hop neighbors a message in the following form: $BAI_j \| k_b^j(D \| H(D))$. Then j updates all its neighbors' anonymous broadcast identities as i does. Node j's neighbors will broadcast the message in a similar way. The message will be broadcasted in the entire network, one hop at a time.

$$BAI_i = H_1(BAI_i \oplus \beta_i) \quad (7)$$

4.4.4. Anonymous acknowledgement.

After a message is successfully transmitted, both the sender and the receiver update their anonymous identities. However, message loss and transmission errors may occur, which may cause the sender and the receiver out of synchronization. This will cause problems for future communications between the two nodes. To solve the aforementioned problem, we propose an anonymous ACK scheme as follows. When a node i wants to send a message, it generates an anonymous ACK identity (AAI) by hashing the value of $AAI_i \oplus \alpha_i$ using Equation (8), inserts it into table T_i, encrypts it as part of the payload, and sends j a message as shown in Equation (9):

$$AAI_i = H_1(AAI_i \oplus \alpha_i) \quad (8)$$

$$M_{i\to j} = D_{\text{rand}} \| OHAI_{i\leftrightarrow j} \| E_{k_{i\leftrightarrow j}} \times (AAI_i \| AI_i \| E_{k_i}(D) \| H(AAI_i \| AI_i \| E_{k_i}(D))) \quad (9)$$

where D_{rand} is a random padding that makes the length of message sent from a source node the same as that of a data message relayed by a normal node. On receiving this message, the receiver j decrypts the message, obtains the AAI, and adds it as part of the message to the next sensor r.

$$M_{j\to r} = AAI_i \| OHAI_{j\leftrightarrow r} \| E_{k_{j\leftrightarrow r}} \times (AAI_j \| AI_i \| E_{k_i}(D) \| H(AAI_j \| AI_i \| E_{k_i}(D))) \quad (10)$$

If the sender i overhears the message (from j to r) with the same AAI, it knows that the message has been received by j correctly. After that, both the sender i and receiver j update their shared anonymous identity. If node i does not overhear a message with the same AAI after a time-out period, i assumes j did not receive message $M_{i\to j}$ correctly (because of message loss or transmission errors), and i retransmits $M_{i\to j}$ to j. Node j can tell this is a retransmitted message according to AAI_i and, in this case, j will send to i an explicit ACK message in the form of $AAI_i \| D_{\text{rand}}$. In case an ACK message is lost, j also waits for a fixed period, say, T. After that, j updates its $OHAI_{i\leftrightarrow j}$ and is in synchronization with i's again. In order to behave like a sensor node, for each received message, the base station also sends an ACK message back to its neighboring sensor nodes.

4.4.5. Secure node addition.

Once a node is useless by an attack or out of energy, a new node, say i, should be added to the network. The secrets of i are either known by itself or shared between i and its neighbors. Node i's own secrets, such as $k_i, \alpha_i, \ldots$, can be loaded in advance. And other secrets of i can be generated by both node i and its neighbors after i is authenticated successfully by its neighbors. Node i can pass the authentication with the help of a trusted party such as the base station. The base station broadcasts a message with the form of $h \| E_{k_j}(k_s) | E_{k_r}(k_s) \| \ldots$, using the anonymous broadcasting scheme in Section 4.4.3, where h is a hop count field and increases when propagating. When a node is newly added to the network, the hop count of other nodes and the link directions may change. Each node gets its hop count by reading h in the broadcast message, and then each node updates its link directions. In the broadcast message, k_s is a one-time session key that has already been preloaded in node i. Nodes j and r are i's neighbors, and they can get k_s by decrypting the corresponding part of the broadcast message. Hence, node i will be able to pass the authentication from its neighbors by using k_s. Then, i can communicate with its neighbors and generate their sharing secrets securely by k_s. As all the information is encrypted by k_s, it is impossible for attackers to know the secrets shared by node i and its neighbors even if attackers know when node i is added to the network.

5. SECURITY ANALYSIS

In EAC, no two transmissions (either the same or different messages) use the same node identity. Recall that each message uses different anonymous identity and is encrypted hop by hop. A message has different appearance after every hop. Global passive attackers can only observe many transmissions but they cannot find out the source node, the communication relationship, and the base station. Later, we will mainly analyze the anonymity performance of EAC under active attacks.

5.1. Sender anonymity

Active attackers may compromise some sensor nodes and decrypt messages received by (and sent from) these nodes. Denotes S_{AJ} as the global identities obtained from the decrypted messages. In the following (Theorem 5.2), we will show that even with S_{AJ}, it is hard for active attackers to find out the source node. Thus, sender anonymity is ensured.

Lemma 5.1. *For $\forall i \in N$, it is impossible for node i to compute the pairwise key $k_{r\leftrightarrow j}$ shared between nodes j and r, where $i \neq r \neq j$.*

Proof. We discuss two cases depending on whether nodes j and r are neighbors of node i or not.

(1) If $j \in N^i_{\text{nei}}$ and $r \in N^i_{\text{nei}}$, then node i knows $k_{i\leftrightarrow j}$ and $k_{i\leftrightarrow r}$. By Equation (3), we have $k_{r\leftrightarrow j} = H_2(k_r + k_j + \alpha_r + \alpha_j)$. Inn order to calculate $k_{r\leftrightarrow j}$, one needs to know both k_r and k_j. However, k_r (k_j) is only known by node r (j). Hence, it is impossible for node i to compute $k_{r\leftrightarrow j}$.
(2) If $j \notin N^i_{\text{nei}}$ or $r \notin N^i_{\text{nei}}$, then node i knows even less information of nodes r and j. And it is impossible for node i to compute $k_{r\leftrightarrow j}$. □

Theorem 5.2. *It is hard for attackers to find out a source node under both passive and active attacks.*

Proof. Suppose node i is the source node, and i sends out a message to j with the form $M_{i\to j} = D_{\text{rand}} \| OHAI_{i\leftrightarrow j} \| E_{k_{i\leftrightarrow j}}(AAI_i \| AI_i \| E_{k_{i\leftrightarrow j}}(D))$. To provide the sender anonymity, i uses a global anonymous identity AI_i for end-to-end (from i to the base station) communication. If $M_{i\to j}$ is not relayed by any compromised node before it arrives at the base station, the attacker does not see the global anonymous identity AI_i. Hence, the attacker does not know who the source node is. Let N_{comp} denotes the set of compromised nodes. If $i \notin N_{\text{comp}}$ and $M_{i\to j}$ is relayed by some compromised nodes before it arrives at the base station, then $AI_i \in S_{AJ}$. Because node i uses a different global anonymous identity for sending out each message, the attacker cannot tell whether an identity in S_{AJ} belongs to the same node or not. Hence, the attacker still cannot find out the source node.

In the worst case, if $i \in N_{\text{comp}}$, then the attacker has access to all the information of node i and is able to find out that i is the source node. To be more precise, assume that the average number of messages captured by the attacker from a compromised node is γ and that there are $|N_{\text{comp}}|$ compromised nodes. In order to get all the global anonymous identities from these captured messages, attackers need Z_1 decryptions as shown in Equation (11). On the other hand, attackers try to find the source node by $|S_{AJ}|$ hashing operations on each node using (5). As if a compromised node was ever a source node, it must have used a

global identity included in S_{AJ}. And thus the total hashing operations is Z_2 as in Equation (12). If there are N nodes in a WSN and there are δ source nodes, then the attacker 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 large too, and then the computation cost for the attacker is high. If δ is small, attackers need less computation. However, the probability of compromising a source node is also small.

$$Z_1 = |N_{\text{comp}}|\gamma \tag{11}$$

$$Z_2 = |S_{AJ}\|N_{\text{comp}}| \tag{12}$$

□

5.2. Communication relationship anonymity

Communication relationship should be protected, otherwise an attacker may infer the identity of a source node or the base station [12]. If node i receives a message and then 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 $i \neq j \neq r$. According to the anonymous data forwarding scheme (in Section 4.4.2), m_i includes anonymous identity and encrypted message body, denoted as $m_i.identity$ and $m_i.body$, respectively. Only if an attacker can tell that m_i and m_j are the same message (i.e., they have the same message body after decryption), the communication relationship between i and j is discovered. Let $M = m_1, m_2, \ldots, m_p$ denote the messages sent by all the neighbors of node i during a time interval $t' \in (t, t+\varepsilon]$, where ε is the upper bound of transmission latency. As long as two (out of the three) nodes are not compromised, then the communication relationship anonymity is guaranteed. Suppose nodes i and j are not compromised. The attacker cannot decrypt the message body $m_i.body$ transmitted from i to j because he does not know $k_{i\leftrightarrow j}$. Hence, the attacker cannot tell if any $m_j \in M$ satisfies that $D_{k_{j\leftrightarrow r}}(m_j.body) == D_{k_{i\leftrightarrow j}}(m_i.body)$ (i.e., if they have the same message body after decryption), where $D_k(m)$ means decrypting m by key k. Thus, the communication relationship anonymity is ensured.

5.3. Base station anonymity

Because there is no information about the base station included in any message and all messages are indistinguishable, passive attackers cannot find out which node the base station is on the basis of captured messages. Moreover, the base station behaves like a normal sensor node. As for active attackers, they may compromise several sensor nodes in a short period, get information from these compromised nodes, and find out the communication relationship among the neighboring nodes. However, because of the 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. Although an attacker can identify a broadcast message within the transmission range of each compromised node, he cannot find out the base station by the timing order of the broadcast transmissions, because a random delay is added for each broadcast message. Hence, the base station anonymity is provided.

In all, sender anonymity is guaranteed by the anonymous data sending scheme. As each source node uses a global anonymous identity instead of its real identity and changes anonymous identity after every message sending, it is difficult for attackers to trace the source node by analyzing anonymous identities from captured messages. Communication relationship anonymity is achieved by the anonymous data forwarding scheme. By this scheme, one node forwards a message to one of its neighboring node by a hidden identity that is only shared between them. So, the message sender and receiver are unlinkable. And the anonymous broadcasting scheme is used to conceal the broadcast-originating node that may be the source node or the base station. Different from the aforementioned three anonymous schemes, the anonymous ACK scheme is not designed for any of the three anonymities. It is used to deal with problems such as message loss and transmission errors that may cause anonymous identities' updating between two neighboring nodes out of synchronization. So, with the anonymous ACK scheme, the other three anonymous schemes can provide anonymous communication with reliability.

We compare the anonymous performance of our EAC protocol with several existing anonymous schemes in Table III. It can be seen from Table III that only EAC achieves all the anonymities, whereas the other schemes cannot.

6. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the EAC protocol, including the storage, computation, and communication costs. For anonymous end-to-end communications, 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 $\alpha_{i\leftrightarrow j}$ for generating the one-hop anonymous identity. Node i also has to store AAI_i, α_i, and $k_{i\leftrightarrow j}$ for generating the AAI and one-hop message encryption. For anonymous broadcast communication, i stores BAI_i, β_i, and k_b^i to create anonymous broadcast identity and encrypt the broadcast message. Node i also stores BAI_j, β_j, and k_b^j for broadcast message authentication and decryption. For forwarding node selection, i needs to store the $linkdir_{i\rightarrow j}$ for each of its neighbor j.

There are three possible link directions, so the link direction can be represented by 2 bits. Hence, the total memory

Table III. 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
EAC	**satisfied**	**satisfied**	**satisfied**

SAS, Simple Anonymity Scheme; CAS, Cryptographic Anonymity Scheme; APR, Anonymous Path Routing; DCARPS, Destination Controlled Anonymous Routing Protocol for Sensornets; EAC, Efficient Anonymous Communication.

requirement for one node is $4n_1 + 2n_2 + (5n_1 + 2n_2)C +$ 2 bits, where C denotes the average number of neighbors for each node. If $n = n_1 = n_2$, then the memory cost is $6n+7nC+2$ bits. For instance, in a WSN with 1000 nodes, let $n = 128$ (e.g., MD5 has 128-bit hash code), and each node has an average neighbor size of 30 nodes. The memory requirement shall be $6 * 128 + 7 * 128 * 30 + 2 = 27,650$ bits $= 3,456$ bytes $= 3.38$ kB. For a WSN with 5000 nodes and a neighborhood size of 100 nodes, the memory requirement shall be $6 * 128 + 7 * 128 * 100 + 2 = 90,370$ bits $= 11.03$ kB. A TelosB mote (Crossbow Technology Inc., Milpitas, CA, USA) [20] has 1-MB flash memory and 48-kB RAM. Hence, it is feasible to implement our EAC protocol in today's sensor nodes.

Our EAC is a lightweight protocol because it only uses hashing functions and symmetric cryptography. In order to accept and forward a message, each node needs two hashing operations for one-hop anonymous identities updating and one hashing operation for AAI updating. Besides, each node needs one hashing operation for message digest. Table IV compares the storage and computation costs of our EAC protocol with several existing sensor anonymous communication protocols. Note that we do not include the computation cost of data encryption, as data encryption operation by EAC is the same as the existing anonymous communication protocols [5,20,21].

Table IV shows that DCARPS [21] has the smallest storage and computation cost. However, DCARPS has the worst anonymity and security performance. DCARPS cannot achieve the base station anonymity and the communication relationship anonymity under global passive attacks. Moreover, DCARPS cannot defend active attacks such as replay attacks because all nodes uses the same identity for message sending and forwarding. Table IV also shows

Table IV. 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
EAC	$\boldsymbol{6n+7nC+2}$	**four hashing operations**

SAS, Simple Anonymity Scheme; CAS, Cryptographic Anonymity Scheme; APR, Anonymous Path Routing; DCARPS, Destination Controlled Anonymous Routing Protocol for Sensornets; EAC, Efficient Anonymous Communication.

Table V. Performance comparison.

Anonymous communication protocols	Communication cost (number of messages)
SAS	$P + N$
CAS	$P + N + N * n$
APR	$N + \gamma$
DCARPS	No extra communication cost with constant IDs
EAC	$\boldsymbol{N+\gamma}$

SAS, Simple Anonymity Scheme; CAS, Cryptographic Anonymity Scheme; APR, Anonymous Path Routing; DCARPS, Destination Controlled Anonymous Routing Protocol for Sensornets; EAC, Efficient Anonymous Communication.

that SAS has low computation cost because SAS creates anonymous identities from the pseudonym space, which has light computations. However, EAC uses much less storage than SAS.

Our EAC is independent from any routing protocols. So without considering the specific routing protocol, the communication cost of EAC is $N+\gamma$. As each node initializes a one-hop broadcast message to exchange information among its neighbors for neighboring table establishment, the communication cost of the whole network for message exchange is N. Besides, according to the anonymous ACK scheme (see Section 4.4.4), once a node receives a message, it should send an anonymous ACK message in case of message loss. So, γ is the communication cost of ACK messages. Anonymous protocols that are not considering reliable communication such as SAS, CAS, and DCAPRS have no such extra overhead. Both SAS and CAS establish pairwise keys for any two nodes and have extra communication cost P. Table V shows that DCARPS has the smallest communication cost. This is because each node uses its constant IDs for message receiving and forwarding respectively in DCARPS. So, DCARPS does not have to exchange messages in the network initialization stage. However, without using anonymous identities, DCARPS has the worst anonymity and security performance. Note that we do not include the communication cost of initial broadcasting, as initial broadcasting operation by EAC is the same as the other existing anonymous communication protocols.

To sum up, the previous discussions show that our EAC protocol achieves all three anonymities with low storage and computation costs.

7. CONCLUSION

Anonymous communication is very important in WSNs, because it can be used to conceal the identities of important nodes, such as source nodes and the base station. Existing sensor anonymity schemes cannot achieve all the three kinds of anonymities. In this paper, we presented an EAC protocol for sensor networks, and it consists of four schemes: anonymous sending, anonymous forwarding, anonymous broadcasting, and anonymous ACK. Performance analysis and comparison showed that EAC can provide all three anonymities: the sender anonymity, the communication relationship anonymity, and the base station anonymity while incurring small storage, computation, and communication costs.

ACKNOWLEDGEMENTS

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

REFERENCES

1. Shi J, Zhang R, Liu Y, Zhang Y. Prisense: privacy-preserving data aggregation in people-centric urban sensing systems, In *Proceedings of the IEEE INFOCOM 2010*, San Diego, CA, 2010.
2. Chen X, Makki K, Yen K, Pissinou N. Sensor network security: a survey. *IEEE Communications Surveys and Tutorials* 2009; **11**(2): 52–73.
3. Akyildiz IF, Stuntebeck EP. Wireless underground sensor networks. *IEEE Journal on Selected Areas in Communications* 2008; **4**(6): 669–686.
4. Chen J, Fang BX, Yin LH, SU S. A source-location privacy preservation protocol in wireless sensor networks using source-based restricted flooding. *Chinese Journal of Computers* 2010; **33**(9): 1736–1747.
5. Sheu JP, Jiang JR, Tu C. Anonymous path routing in wireless sensor networks, In *Proceedings of IEEE International Conference on Communications (ICC '08)*, 2008.
6. Patwari N, Ash JN, Kyperountas S, Hero AO, Moses RL, Correal NS. Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine* 2005; **22**(4): 54–69.
7. Chen H, Xiao Y, Hong X, Hu F, Xie J. A survey of anonymity in wireless communication systems. *Security and Communication Networks* 2009; **2**(5): 427–444.
8. Takahashi D, Hong X, Xiao Y. On-demand anonymous routing with distance vector protecting traffic privacy in wireless multi-hop networks, In *Proceedings of the 4th International Conference on Mobile Ad-hoc and Sensor Networks (MSN '08),* 2008.
9. Asadpour M, Sattarzadeh B, Movaghar A. Anonymous authentication protocol for GSM networks. *International Journal of Security and Networks* 2008; **3**(1): 54–62.
10. Tsai K, Hsu C, Wu T. Mutual anonymity protocol with integrity protection for mobile peer-to-peer networks. *International Journal of Security and Networks* 2010; **5**(1): 45–52.
11. Chen Y, Susilo W, Mu Y. Convertible identity-based anonymous designated ring signatures. *International Journal of Security and Networks* 2006; **1**(3): 218–225.

12. Kang L. Protecting location privacy in large-scale wireless sensor networks, In *Proceedings of IEEE International Conference on Communications (ICC '09)*, 2009.
13. Yang Y, Shao M, Zhu S, Urgaonkar B, Cao G. Towards event source unobservability with minimum network traffic in sensor networks, In *Proceedings of the first ACM Conference on Wireless Network Security*, 2008.
14. Chen T, Zhong S. INPAC: An enforceable incentive scheme for wireless networks using network coding, In *Proceedings of the 29th Conference on Information Communications (INFOCOM '10)*, 2010.
15. Kermarrec AM, Tan G. Greedy geographic routing in large-scale sensor networks: a minimum network decomposition approach, In *Proceedings of the eleventh ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2010.
16. Liu J, Xiao Y. Temporal accountability and anonymity in medical sensor networks. *ACM/Springer Mobile Networks and Applications (MONET), Special Issue on Ubiquitous Body Sensor Networks* 2010: 1–18.
17. Kao JC, Marculescu R. Real-time anonymous routing for mobile ad hoc networks, In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '07)*, 2007.
18. Zhang Y, Liu W, Lou W. Anonymous communications in mobile ad hoc networks, In *Proceedings of the 24th Conference on Information Communications (INFOCOM '05)*, 2005.
19. Law YW, Doumen J, Hartel P. Survey and benchmark of block ciphers for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)* 2006; **2**(1): 65–93.
20. Misra S, Xue G. Efficient anonymity schemes for clustered wireless sensor networks. *International Journal of Sensor Networks* 2006; **1**(1): 50–63.
21. Nezhad AA, Miri A, Makrakis D. Location privacy and anonymity preserving routing for wireless sensor networks. *Journal of Computer Networks* 2008; **52**(18): 3433–3452.
22. Li T, Song M, Alam M. Compromised sensor nodes detection: a quantitative approach, In *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops*, 2008.
23. Song H, Xie L, Zhu S, Cao G. Sensor node compromise detection: the location perspective, In *Proceedings of the 2007 International Conference on Wireless Communications and Mobile Computing*, 2007.
24. Lu R, Lin X, Zhang C, Zhu H, Ho PH, Shen X. AICN: an efficient algorithm to identify compromised nodes in wireless sensor network, In *Proceedings of the IEEE International Conference on Communications (ICC '08)*, 2008.
25. Zhang Y, Liu W, Lou W, Fang Y. Location-based compromise-tolerant security mechanisms for wireless sensor networks. *IEEE Journal on Selected Areas in Communications* 2006; **24**(2): 247–260.
26. Lazos L, Poovendran R. SeRLoc:secure range-independent localization for wireless sensor networks, In *Proceedings of ACM Workshop Wireless Security*, 2004.
27. Shao M, Yang Y, Zhu S, Cao G. Towards statistically strong source anonymity for sensor networks, In *Proceedings of IEEE INFOCOM 2008*, 2008; 51–55.

AUTHORS' BIOGRAPHIES

Juan Chen is a Ph.D. candidate in the School of Computer Science and Technology at the Harbin Institute of Technology, China. She received her B.E. degree and her M.S. degree, both in Computer Science, in 2005 from South West University and in 2008 from Harbin Institute of Technology, respectively. Her research interests include wireless network security and privacy.

Xiaojiang (James) Du is currently an associate professor in the Department of Computer and Information Sciences at Temple University. Du received his B.S. degree in Electrical Engineering from Tsinghua University, Beijing, China, in 1996 and his MS and Ph.D. degree in Electrical Engineering from the University of Maryland College Park in 2002 and 2003, respectively. Between August 2004 and July 2009, Du was an assistant professor in the Department of Computer Science at North Dakota State University, where he received the Excellence in Research Award in May 2009. His research interests are wireless networks, security, computer networks, and systems. He has published over 80 journal and conference papers in these areas and has been awarded more than $2M research grants from the US National Science Foundation (NSF) and Army Research Office. He serves on the editorial boards of four international journals. Du is the Chair of the Computer and Network Security Symposium of the IEEE/ACM International Wireless Communication and Mobile Computing conference 2006–2010. He is a Technical Program Committee member of several premier ACM/IEEE conferences such as INFOCOM (2007–2012), IM, NOMS, ICC, GLOBECOM, WCNC, BroadNet, and IPCCC. Du is a Senior Member of IEEE and a Life Member of ACM.

Binxing Fang received his B.S. degree in Computer Science from the Harbin Institute of Technology of China in 1981. He received his M.S. and Ph.D degrees in Computer Science from the Tsinghua University and Harbin Institute of Technology of China in 1984 and 1989, respectively. He is a member of Chinese Academy of Engineering. His research interests include information security, information retrieval, and distributed systems. Fang is the director of the National Computer Network Emergency Response Technical Team in China, an expert of the National 863 High-Tech Project in the information security technology field. Fang is the principal investigator of over 30 projects from the state and ministry/province in China. He has published over 200 papers.