Accountable and Privacy-enhanced Access Control in Wireless Sensor Networks

Daojing He, *Member, IEEE*, Sammy Chan, *Member, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

***Abstract*—In general, owners and users of wireless sensor networks (WSNs) are different entities. A user may want to hide his/her data access privacy from anyone else including the network owner and, at the same time, users who misbehave need to be identified. Such requirements necessitate privacy-preserving and accountable access control. In this paper, we develop a novel protocol, named *APAC*, to satisfy this need. First, APAC can enforce strict access control so that the sensed data is only accessible by the authorized users. Second, APAC offers sophisticated user privacy protection. Third, misbehaving users or owners can be audited and pinpointed. Last but not least, it does not rely on the existence of a trusted third party, and thus is more feasible in practice. The feasibility of the APAC is demonstrated by experiments on resource-limited mobile devices and sensor platforms.**

***Index Terms*—Access Control, Wireless Sensor Network, User Privacy, Accountable.**

I. Introduction

Wireless sensor networks (WSNs) have been widely deployed in physical world for various applications such as environmental monitoring [1]. Often, the environment in which a WSN is operated is vulnerable to various security attacks. One of the most important security measures is users access control, which ensures only legitimate users can access the data collected by sensor nodes. Data access control in WSNs mainly follows two approaches, namely, centralized and distributed approaches. In the centralized case, sensed data are collected from individual nodes and then transmitted to a central location, usually the sink, for access from authorized users. In the distributed approach, each authorized user enters the sensor field to directly access data on nodes without involving the sink. The distributed approach [2]–[5] can avoid the weaknesses of the centralized approach such as single point of failure, performance bottleneck, and many potential security vulnerabilities along the long communication paths from sensor nodes to the sink. Note that the work presented in this paper is applicable to both centralized and distributed access control.

The traditional trust model of WSNs assumes the owner of the network to be both the collector and consumer of sensor readings. While this makes sense for small, experimental networks, this is not likely to be the case for large scale WSNs. A large WSN could be operated by multiple owners to provide services to users belonging to different organizations. For example, large scale WSNs are developed in projects such as GEOSS [6] and NOPP [7], in which 61 countries are involved in GEOSS and various government departments are involved in NOPP. Users from business sector, government agencies and academia would be interested to access the data collected in these networks. Under such a circumstance, there is no trust among owners and users due to their diverse and possibly conflicting interests.

While it is important to enforce network access control to handle adversaries including dishonest users, it is critical to be able to avoid invasion of user privacy. Privacy means not only hiding the user's true identity, but also the linkage among the transactions of the same unknown user. Thus, an adversary should be prevented from linking the communication activities of a particular user to establish the user's profile. Furthermore, user accountability must be provided since bad user behaviors and insider attacks should be audited and pinpointed. Also, security and privacy of WSNs is usually achieved through a trusted third party such as key distribution center or trusted authority, but the establishment and maintenance of this entity in such a distributed environment is neither feasible nor pragmatic, particularly in privacy-aware context (e.g., once it is compromised, the user privacy would be exposed). Due to these reasons, there is a growing demand for adequate provision of secure, privacy-preserving and accountable access control protocols without involving any trusted third party.

However, past research on WSN security mainly focused on communication security and data security [2]–[5], [8]–[13]. Privacy-preserving access control has only gained limited attention so far, and no privacy-preserving and accountable access control protocol for sensor networks has been proposed. More importantly, when considering this research issue, we observe that none of the available privacy-aware cryptographic primitives can be applied directly. The detailed analysis to arrive at these conclusions will be given in Sections II and III.B.

To resolve the above challenges, this paper makes three main contributions:

(1) We identify the characteristics of a multi-owner-multi-user sensor network and then introduce the secure, privacy-preserving and accountable access control problem in WSNs for the first time in literature.

(2) We propose a novel protocol to meet the requirements of

Manuscript received 28 Jan. 2014; revised 06 May 2014 and 21 Jul. 2014; accepted 23 Jul. 2014. This research is supported by a strategic research grant from City University of Hong Kong [Project No. 7004054], the Pearl River Nova Program of Guangzhou (No. 2014J2200051), and the Specialized Research Fund for the Doctoral Program of Higher Education. D. He is the corresponding author of this article.

D. He is with the Software Engineering Institute, East China Normal University, Shanghai, P.R. China, and also with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, P.R. China (e-mail: hedaojinghit@gmail.com).

S. Chan is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR, P.R.China (e-mail: eeschan@cityu.edu.hk).

M. Guizani is with Qatar University, Qatar (e-mail: mguizani@ieee.org).

this new kind of protocols, namely *APAC*, which exploits the group signature technique, tailors, and adapts it for WSNs. Moreover, APAC is efficient even in a large scale network with many nodes, many users and many revoked users. Furthermore, it supports dynamic participation. New users can easily join the network, and users can easily be revoked when their subscriptions expire. Since the group signature technique is not originally designed for access control, a direct application of this technique simply cannot meet the four requirements, i.e., privacy-preservation, accountability, high efficiency, and no trusted third party. To address these challenges, we employ the *separation of duties* principle to design new key generation and tracing phases for the existing ways of group signature construction. Besides an enhanced group signature algorithm, some newly designed mechanisms are incorporated into APAC, such as a novel user revocation mechanism which is suitable for the resource-limited sensor nodes, and a hybrid membership maintenance approach which can improve the network performance.

(3) We implement the proposed protocol in real resource-poor mobile devices and two common sensor platforms (i.e., TelosB and Imote2). Evaluation results demonstrate the efficiency of APAC in practice. To the best of our knowledge, this is also the first implemented privacy-preserving and accountable access control on the WSN platform.

The rest of this paper is organized as follows. In Section II, we review the related work. Section III presents the network, trust and adversary models. Section IV discusses the drawbacks of available privacy-aware cryptographic primitives and presents the key management scheme used by APAC. Section V describes APAC in detail. Section VI analyses the security properties of APAC. Section VII describes the implementation and experimental results of APAC via real sensor platforms. Finally, Section VIII concludes this paper.

II. Related Work

In the literature, various mechanisms have been proposed to address different aspects of securing WSNs. However, past research mainly focused on communication security and data security, such as secure data access [2]–[5], secure data discovery and dissemination [8], key distribution [9], intrusion detection [10], secure time synchronization [11], secure reprogramming [12], and secure routing [13]. For the secure data access schemes of [2]–[5], the authors just focus on how to control the access to the nodes, but they do not consider the user privacy aspect in the data access procedure. For example, some studies [2]–[5] using attribute-based cryptographic primitives have achieved fine-grained access control over sensor data.

Recently, some techniques have been proposed to provide privacy protection over the data collected by and transmitted through WSNs [14]. For example, based on the ring signature technique, a novel distributed privacy-preserving access control named *Priccess* is proposed [15]. Also, based on the blind signature algorithm, a distributed privacy-preserving access control named *DP*^{2}*AC* is presented [16]. These two techniques not only control distributed access to the nodes, but also hide users' identities and thereby guarantees query privacy. However, there are some security weaknesses and efficiency problems in these protocols. For example, it has been reported that DP^{2}AC is not fine-grained, since each anonymous user has the same access privilege. Additionally, for Priccess, both the computation complexity on a node and the anonymity strength depend on the size of the chosen signing group, thus a balance is needed between anonymity strength and overhead. More importantly, they are not designed with user accountability in mind. Also, the network owner can impersonate any network user.

III. Network, Trust and Adversary Models

In this section, we introduce the network, trust and adversary models for this work.

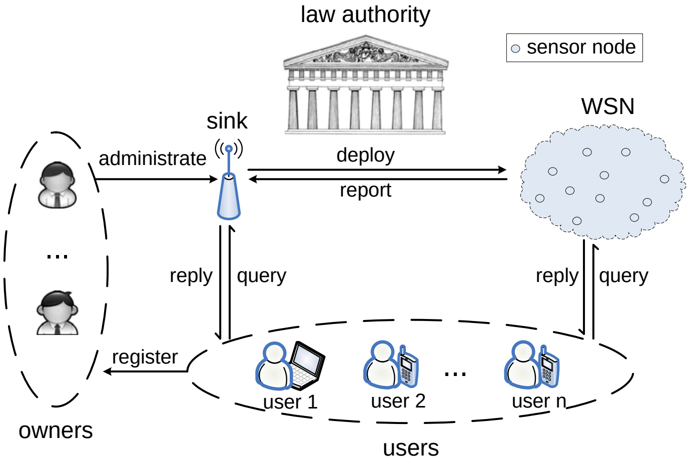


Fig. 1. WSN network architecture.

A. Network Model

Fig. 1 depicts a WSN which consists of many users, a large number of resource-constrained sensors and a sink[1], one or more network owners and an off-line law authority. An example of possible law authority is a local police department. The sensors report their sensed data to the sink (administrated by one or more network owners) and users in response to queries. After registering to one or more network owners, network users use access devices such as smart phones or Laptop PCs to access the sensed data by sending queries to the sink or the targeted nodes. The network owners bootstrap the keying materials for access devices to enforce the access control policy. According to the agreement, each network owner has specific access privilege of his/her own. For instance, a network owner just supports the access service for temperature reading, another owner supports accessing CO_2 reading. Also, since time synchronization is commonly available in WSNs to support various applications, we assume the nodes are synchronized with the help of some existing secure time synchronization schemes [11].

[1] Without loss of generality, we consider one sink in this paper, but the proposed protocol is also applicable to networks with multiple sinks.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2014.2347311, IEEE Transactions on Wireless Communications

B. Trust and Adversary Models

It is assumed that users need to pay according to the amount of data they retrieve from the network. Therefore, network owners would enforce strict access control. In the extreme cases such as severe attacks, the law authority, which acts as an independent third party, will be asked to track the particular responsible attacker. However, both the law authority and network owners may, for some reasons, probe users' access profile such as data type and access time. That is, trust in network owners or the law authority is limited. Thus, both of them should be prevented from compromising user privacy.

An adversary could be either an outsider or insider. As an outsider, it could eavesdrop or replay messages transmitted in the network, inject bogus messages into the network, or launch wormhole attacks. In insider attacks, the adversary can compromise and control a number of users and nodes subject to his/her choice.

IV. Key Management in Accountable and Privacy-enhanced Access Control

A. Overview of Accountable and Privacy-enhanced Access Control

As shown in Fig. 1, accountable and privacy-enhanced access control involves five kinds of participants, network owners, sensor nodes, the sink, network users and the off-line law authority. This new kind of protocols should have the following features.

(1) **Security**: Network owners can delegate their access privilege to network users. Users who want to access the network first need to register to at least one network owner.

(2) **Privacy-preservation**: Each authorized user wants to send a query to the sink (rsp. the nodes) in such a way that it remains anonymous, and the sink is (rsp. the nodes are) convinced that the query command is indeed from a legitimate user authorized by a particular network owner. From the perspective of the network, this is required and sufficient for limiting user access privileges. Even though network owners have the ability to delegate network access privilege to network users, they cannot determine the actual source of the query command.

(3) **Accountability**: Network owners can help the law authority to track the particular attacker who is responsible for a certain network access activity. In other words, given a query submitted by a user, neither network owners nor the law authority can determine the user identity unless they collaborate and combine their knowledge. Also, both the law authority and network owners cannot impersonate any legitimate user.

(4) **No trusted third party**: The trust of all entities should be limited. Therefore, the mechanism can avoid single point of failure.

(5) **High efficiency**: Due to limited energy, processing and storage resources of sensor nodes, a cryptographic mechanism should be efficient.

Without the help of any trusted third party, security, privacy, and accountability are three seemingly contradictive objectives in the case of access control. First, a user has to reveal his/her identity in order to be verified for authentication purposes. However, the identity of a user serves as a unique identifier that an adversary can make use of to filter out a particular user's network access transactions, and trace his/her data access pattern, which may leak sensitive user privacy information. The linkability among a user's network access transactions may also reveal a user's profile without the user's consent. Second, a user hopes to protect his/her data access privacy from his/her network owner, although the network owner controls network access. Therefore, neither network owners nor the law authority should be able to trace a particular user. However, in case of service disputes or frauds, the corresponding network owner needs to help the law authority to identify the actual source of a query.

B. Drawbacks of Available Cryptographic Primitives

We observe that none of the available privacy-aware cryptographic primitives (e.g., standard digital signature, group signature, blind signature, ring signature) can be applied to achieve the goal discussed above. The detailed analysis is given as follows.

A naive approach to enforce access control is to require each authorized user to send the targeted nodes a standard digitally signed message. A standard digital signature algorithm such as RSA allows a user to sign a message with his/her private key such that any verifier can verify the message originated from an authorized user. However, such a signature message will directly reveal the user's identity.

Also, blind signature and ring signature suffer from degraded security protection due to the lack of user accountability. That is, they provide irrevocable anonymity. Technically there is no way to revoke the anonymity of the user even if he/she decides to expose himself/herself.

Group signature is another signer-ambiguous signature scheme that is suitable for user privacy protection due to the k-anonymity property it possesses. A group signature scheme allows a member of a group to sign a message on behalf of the group, without revealing which member generated the signature. However, different from blind signature and ring signature, in exceptional cases such as a legal dispute, a designated group manager can use the group private key to open a group signature to reveal unambiguously the identity of the signature's originator.

Group signature is an attractive cryptographic primitive to support accountability because it has the capability to revoke a user's anonymity. However, the revocation capability of group signature algorithms also degrades user privacy protection because the network owners who usually serve as the group managers, will always be able to track each user. Moreover, a network owner can impersonate any user as he/she is responsible for generating and distributing member secret keys. This means that group signature cannot be directly applied to our problem, since network owners are not trustworthy.

According to the above analysis, it is clear that achieving privacy-preserving and accountable access control is still an open challenge in WSNs. The problem is further exacerbated by the fact that sensor nodes usually have limited resources.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2014.2347311, IEEE Transactions on Wireless Communications

When designing suitable cryptographic techniques, computation efficiency and storage overhead should be given priority to cope with the resource-constrained nature of WSNs.

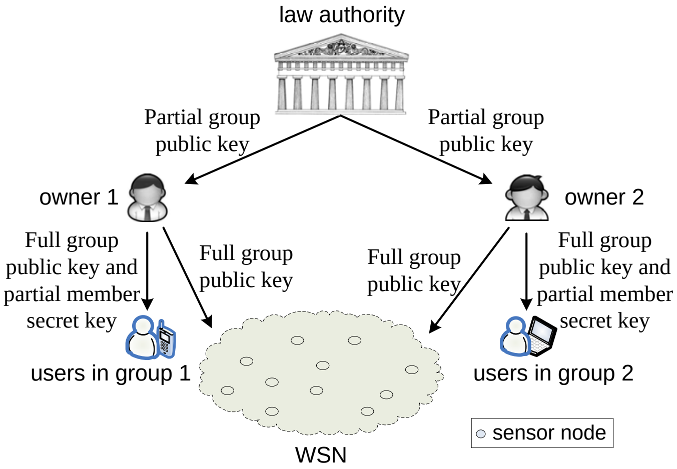


Fig. 2. Trust and key management model of APAC.

C. Trust and Key Management Model of APAC

The trust and key management model adopted in this paper is shown in Fig. 2. Here users who have registered to the same network owner are organized in a group, among which the network owner acts as the group manager. Our protocol does not assume the existence of a trusted third party, and is therefore more practical. In this case, trust between entities is rather limited. Before accessing the network, each user has to enroll in at least one user group whose manager thus knows the identity of the user. For the whole network, the law authority generates partial group public key and partial group private key, and then allocates the former one to each group manager and keeps the latter one secretly. With partial group public key from the law authority, the group manager generates the full group public key and the other partial group private key, and keeps the latter one secretly. To access the network, each user generates partial member secret key and then requests the other partial member secret key and group public key from his/her group manager. Each group manager distributes the corresponding group public key to the sink and sensor nodes before network deployment. Additionally, according to the agreement among network owners, the access privilege of each group is pre-loaded on each node.

The above key management scheme is based on the separation of duties principle and has several salient features. First, from network access control point of view, each authorized user is assigned a member secret key to generate a legitimate access credential, i.e., the signature of a fresh query. The legitimacy of this access credential is verified by the sink and each node through group public key. Thus, centralized and distributed access security is guaranteed. Second, it divides group private key and the identities of users among two autonomous entities: the group manager and the law authority. The law authority knows the partial group private key, but not the identities of users; the group manager knows the identities of users, but not the full group private key. The system is designed in such a way that given an access credential generated by a user, neither the group manager nor the law authority can determine the user identity or compromise his/her privacy unless both of them collude. Further, without the help of the group manager, the law authority cannot compromise the privacy of any user. User privacy is enhanced in this way. Also, without the full member secret key, both the law authority and network owners cannot impersonate any legitimate user. Finally, in case of service disputes or frauds, the law authority can collect the partial group private key and the identities of users from the corresponding network owner to pinpoint the responsible user. Thus user accountability is achieved. It should be noted that the whole key management procedure can be finished during the system setup phase, thus it does not incur any computation and communication overhead subsequently.

We observe that in order to revoke a user, there are two approaches employed in existing group signature techniques: one is “verifier-local revocation”, where the trace key of a user is added into the latest revocation list (RL) received by each verifier (e.g., sensor nodes in this paper) once the user is revoked. In this case, not only the transmission and storage costs of RL, but also the verification cost of each signature is linearly proportional to the number of revoked users, which can potentially grow fairly large as time elapses. Obviously, this approach cannot be employed in resource-limited sensor nodes. The other is through updating the group public key and member secret keys at all unrevoked users, where the verification cost on each signature is constant. In APAC, we assume the latter approach is used.

V. APAC: The Protocol

In this section, APAC is presented in detail. Before giving the detailed description, we first give an overview of APAC.

A. Overview of APAC

As described above, in order to achieve the goal of the proposed protocol, we need to re-design the key generation and tracing phases of the existing group signature schemes. However, signing and verification algorithms remain the same as the original group signature construction. Here we choose the group signature proposed by Camenisch and Groth [17] as an example because the computation complexity of its signature generation and verification is usually lower than other group signature algorithms. However, as described in Section IV.C, any other efficient group signature schemes can just as easily be applied in APAC. Moreover, some newly designed mechanisms such as novel user revocation and key exchange mechanisms are incorporated into the design of our protocol.

APAC consists of seven phases: system setup, new user joining, user query generation, receiver verification, key establishment, user revocation and user tracing. In the system setup phase, the law authority and each network owner generate the partial group public key and partial group private key, respectively. The full group public key of each network owner is pre-loaded on each node and the sink. The new user joining phase is invoked whenever a user wants to join the network while the user revocation phase runs whenever a user is to be revoked. In the user query generation phase, if a user has a new query, he/she will need to construct the query and the group signature and then send them to the sink or sensor nodes. In the receiver verification phase, if the query verification passes then the sink or sensor nodes respond to the user's query. In the key establishment phase, the user and the sink (rsp. the node) establish a shared session key for the subsequent secure communication. In the user tracing phase, the corresponding network owner helps the law authority to track a particular user who is responsible for a certain network access activity.

B. System Setup Phase

In the early stage of system setup, the law authority is responsible for the generation operation of the partial group private key and partial group public key of the whole network. Specifically, the law authority proceeds as follows:

1. Randomly select an l_Q-bit prime Q and an l_P-bit prime P such that $Q|P-1$. Let F be an element of order Q in $\mathbb{Z}_P^*$.

2. Randomly choose $X_G, X_H \in \mathbb{Z}_Q$ and set $G = F^{X_G} \bmod P$, $H = F^{X_H} \bmod P$.

3. Send partial group public key $\{Q, P, F, G, H\}$ to network owners, possibly via an open wireless channel, and keep partial group private key X_G secretly.

Additionally, as the group manager, each network owner prepares the full group public key and partial group private key as follows.

1. Choose an l_n-bit RSA modulus $n = pq$ as a product of two safe primes p and q. Select at random $a, g, h, w \in QR_n$. Here QR_n denotes the set of all quadratic residues of $\mathbb{Z}_n^*$.

2. Keep the partial group private key (p, q) secretly. The above operation generates the group public key $gpk = \{n, a, g, h, w, Q, P, F, G, H\}$ and the group private key $gsk = \{X_G, p, q\}$. The network owner notifies the sink of gpk, and distributes gpk into each node before the network deployment.

C. New User Joining Phase

Before accessing the network, a user i with the identity UID_i has to authenticate himself/herself to his/her user group manager[2], say j. Specifically, user i generates the partial member secret key as follows:

1. Select a random number $x_i \in \mathbb{Z}_Q$ and compute $Y_i = G^{x_i} \bmod P$.

2. Form a commitment to x_i, $g^{x_i} h^{r_i^{'}} \bmod n$ with $r_i^{'} \in_R \mathbb{Z}_n$ and prove knowledge of x_i, $r_i^{'}$ fitting the above. Here the partial member secret key is $\{x_i, r_i^{'}\}$.

3. Send $(Y_i, g^{x_i} h^{r_i^{'}} \bmod n)$ and the proof to owner j via a secure channel (i.e., using a secure transmission protocols, such as the wired Transport Layer Security protocol).

Upon receipt of the message, owner j prepares user i's the other partial member secret key as follows.

1. Choose a random l_e-bit number e_i such that $E_i = 2^{l_E} + e_i$ is prime. It must be the case that $l_Q + l_c + l_s + 1 < l_E$. A suggestion for parameters (l_c, l_s) is $l_c = 160, l_s = l_e = 60$.

2. Compute $w_i = w^{E_i^{-1}} \bmod n$.

3. Choose a random number $r_i^{''} \in \mathbb{Z}_e$ and set $y_i = (a g^{x_i} h^{r_i^{'} + r_i^{''}})^{E_i^{-1}} \bmod n$.

4. Transmit $\{grp_j, w_i, y_i, E_i, r_i^{''}\}$ back to user i via a secure channel, where grp_j indicates the identity of the group. Finally, user i gets his/her member secret key $msk_i = \{gpk, w_i, x_i, r_i, y_i, e_i\}$, where $r_i = r_i^{'} + r_i^{''}$. Owner j stores the pairing between Y_i and UID_i. Note that this phase does not involve any node and thus does not incur any processing load to it.

D. User Query Generation Phase

For sake of simplicity, in the following procedures, we just use a one-owner-multi-user WSN as an example. Obviously, the proposed protocol can be applied easily in the case of two or more owners. After system setup, each user i can obtain sensed data through accessing the sink or targeted nodes.

User i firstly constructs an appropriate query req, and then generates a group signature σ on $h(req \| grp_j)$ through member secret key msk_i, where $h(.)$ is a collision-resistant hash function. Here the query req mainly indicates the identity of the node to which the query is made, and which sensed data a user wants to access. The procedure of generating σ is specified by Camenisch's scheme [17]. For completeness, the steps carried out by user i are described as follows.

1. Select a random number $r \in \{0,1\}^{l_n/2}$ and $R \in \mathbb{Z}_Q$.

2. Compute $u = h^r y_i w_i \bmod n$, $U_1 = F^R \bmod P$, $U_2 = G^{R+x_i} \bmod P = G^R Y_i \bmod P$, and $U_3 = H^{R+e_i} \bmod P$.

3. Choose $r_x \in \{0,1\}^{l_Q + l_c + l_s}$, $r_r \in \{0,1\}^{l_n/2 + l_c + l_s}$, $r_e \in \{0,1\}^{l_e + l_c + l_s}$ and $R_R \in \mathbb{Z}_Q$ and compute $v = u^{r_e} g^{-r_x} h^{r_r} \bmod n$, $V_1 = F^{R_R} \bmod P$, $V_2 = G^{R_R + r_x} \bmod P$, $V_3 = H^{R_R + r_e} \bmod P$.

4. Generate a challenge $c = h(gpk, u, v, U_1, U_2, U_3, V_1, V_2, V_3, h(req \| grp_j))$ and set $z_x = r_x + c x_i$, $z_r = r_r + c(-r_i - r E_i)$, $z_e = r_e + c e_i$, and $Z_R = R_R + cR$. Obtain the group signature σ as $(c, u, U_1, U_2, U_3, z_x, z_r, z_e, Z_R)$.

Additionally, timestamp T_i is also added into the query req by user i to resist replay attacks. Finally, in centralized access control, user i sends access request $Que = \{req, grp_j, \sigma\}$ to the sink. On the other hand, in distributed access control, user i sends access request Que to the targeted nodes. Note that there are two different cases for user i to access the nodes. One case is that user i wants to access one or more particular nodes, say $\{S_1, ..., S_k\}$, with identities $\{SID_1, ..., SID_k\}$. Here $k \geq 1$. We assume that sensor nodes do not know their geographical locations. Obviously, this assumption makes APAC more applicable in the real world. In this case, the identities are added into the query req. The other case is that user i wants to access the nodes in a specific region. We assume that the nodes know their geographical locations which can be acquired via deployment knowledge or many existing secure localization schemes (e.g., [18]). In this case, user i needs to add the information about the specific region into the query req.

[2] Such an authentication is based on the pre-established trust relationship between the user and owner and may be done through in-person contact.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2014.2347311, IEEE Transactions on Wireless Communications

E. Receiver Verification Phase

Here we consider the targeted nodes as the receiver of the access request message. The following procedure can be simply applied in the case of the sink. Upon receiving an access request Que, a node first checks whether the timestamp T_i included in Que is within some allowable range compared with its current time. If the decision is negative, the query is rejected. Otherwise, the node extracts the group identity grp_j and the query req from Que. The node then checks the validity of grp_j and req according to the access privilege of the group stored in it. If they are invalid, the message Que is rejected; otherwise, the node extracts the group public key gpk of owner j and then verifies such a group signature σ as follows:

1. Check that $z_e \in \{0,1\}^{l_e+l_c+l_s}$ and $z_x \in \{0,1\}^{l_Q+l_c+l_s}$.
2. Set $v = (aw)^{-c} g^{-z_x} h^{z_r} u^{c \times 2^{l_E}+z_e} \bmod n$, $V_1 = {U_1}^{-c} F^{Z_R} \bmod P$, $V_2 = {U_2}^{-c} G^{Z_R+z_x} \bmod P$, $V_3 = {U_3}^{-c} H^{Z_R+z_e} \bmod P$.
3. Check if the challenge c is correct: $c \overset{?}{=} h(gpk, u, v, U_1, U_2, U_3, V_1, V_2, V_3, h(req \| grp_j))$

If the above check succeeds, the node gives a response to user i; otherwise, the message Que is rejected.

F. Key Establishment Phase

In some application scenarios, a session key should be established between a user and the sink (rsp. the targeted nodes) to protect data communication against attacks. For key establishment, the protocol implements an Elliptic Curve Diffie-Hellman (ECDH) based mutually authenticated key exchange. Here we consider the targeted nodes as an example, but the same procedure can be applied to the sink. Let $\mathbb{G}$ be a cyclic group (with generator K) of large prime order v, here the bit length of v is set to 160. In the system setup phase, the network owner generates a public/private key pair for each node and loads it to sensor nodes before their deployment. To issue a public/privacy key pair for each node, say S_k with identifier SID_k, the network owner picks a random number $d_k \in \mathbb{Z}_v^*$ and computes $G_k = d_k \times K$, where d_k is the private key assigned to node S_k and G_k is the public key. Thus, the network owner manages the mapping of the public keys of sensor nodes to the identities of the nodes (i.e., $< G_k, SID_k >$), and this mapping is distributed to each user. In the user query generation phase, if user i wants to access the node, say S_k, the user randomly chooses $b \in \mathbb{Z}_v^*$, computes $G_i = b \times K$. Also, G_i is added into the query req. As described in Section V.E, a node, say S_k, first needs to confirm the validity of Que from user i. Then node S_k does the following.

Node S_k computes $sk = d_k \times G_i$ as the session key between itself and user i. Thus, the node can use the key sk to encrypt the required sensor data $sensor\_data$ of user i. Subsequently, the node gives a response $\{G_i, E_{sk}(sensor\_data), h(sensor\_data, sk)\}$ to user i, where $E_{sk}(X)$ denotes encrypting a message X using a symmetric key sk. Upon receiving this response, user i can pick G_k from the mapping and generates $sk = b \times G_k$ and then decrypts the message to obtain $sensor\_data$. After that, user i uses sk to compute $h(sensor\_data, sk)$ and then compares it with the received $h(sensor\_data, sk)$. If the result is positive, user i believes this message is from the node and has never been modified by adversaries. Thus, a secret session key between user i and node S_k is established so that the same key can be used for subsequent secret communication session. This session is uniquely identified through $\{grp_j, G_i\}$. In this key establishment phase, the user authenticates a node and a sensor node also authenticates the user; mutual authentication is thus provided between the user and the node.

All session keys are only for per-session use in the proposed protocol (In this paper, a session refers to the message flow resulting from a single triggering access event). Thus, this solution is scalable and equipped with efficient key management because the number of managed keys is linearly proportional to the number of principals. For each single triggering access event, if a user and the sink (or the targeted node) communicate for a long period of time or there is a need of strong security in a specific application, the user and the sink (or the targeted node) periodically update the session keys to limit the amount of private communication information can be recovered in case the keys are compromised.

G. User Revocation Phase

In most cases, a network owner, say j, hopes to limit the time period for which each user i can access the network. This can be done as follows. In the new user joining phase, when user i registers to owner j, the network owner sets the expiration time of user i, denoted as $TEXP_i$, which results in the mapping between $TEXP_i$ and (E_i, w_i). Thus, once the subscription period of user i has expired, the network owner can publish E_i, and replace in gpk the element w with w_i. Then the network owner needs to sign a *User Revocation Message* and then advertises it to all users and the sink, by say, e-mail or web site announcement. Here an example of *User Revocation Message* is “Owner j asks to delete a user with (E_i, w_i)”. Any unrevoked user, say m, updates his/her member secret key msk_m as follows. User m chooses α, β such that $\alpha E_i + \beta E_m = 1$. Then user m computes the new $w_m = {w_i}^{\beta E_i} w_m^{\alpha E_m} \bmod n$. At the same time, the network owner broadcasts the *User Revocation Message* to all nodes. Upon receipt of the message, each node replaces in gpk the element w with w_i. For small WSNs, a revocation message can reach all nodes by broadcast from the network owner. On the other hand, for large-scale WSNs, a two-tier architecture is commonly used and hence the revocation message could be relayed to all nodes via the cluster gateway nodes.

Obviously, if the network owner frequently sends the *User Revocation Messages*, APAC may not achieve good performance. To solve the above problem, we observe that user revocations are mainly due to two reasons: one is expiration of service subscription, the other is violation of network access policy. Due to the nature of the network access service, user revocations due to the former reason usually happen periodically and are pre-scheduled; and this is the major reason causing frequently updating the group public key. On the other hand, user revocations due to the latter is often random and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2014.2347311, IEEE Transactions on Wireless Communications

sporadic. Based on this observation, a hybrid membership maintenance approach can be employed in APAC to minimize the frequency of broadcasting the *User Revocation Message*. In the system setup phase, each network owner can set the minimum subscription period of the network service as δ time unit, for example, as one month. Thus, each user subscribes the network service for $x \times \delta$ time units, where x is a positive integer. Upon the expiration of each minimum subscription period, each unrevoked user and sensor node will await the *User Revocation Message* from the network owner. Now the frequency of updating the group public key will be decreased, since it does not involve the revocation of the users whose service subscription are expired.

H. User Tracing Phase

When a network owner observes certain network access being disputable and suspicious, it finds the corresponding access request message $Que = \{req, grp_j, \sigma\}$ from the network log file on the sink, to which each node submits their received access requests. Alternatively, there are also hybrid WSNs where the set of nodes, in addition to resource-poor sensor nodes, includes some small number of resource-rich collector nodes, each serving as a temporary repository of the user access records for their constituent sensor nodes. In this case, the network owner can find the corresponding access request message from the collector nodes. Obviously, through grp_j, a responsible network owner j is found from the perspective of the network.

When the law authority decides to track the particular attacker that is responsible for a certain network access, the following procedure is taken:

1. Given the link and the session identifier, the law authority finds the corresponding access request $Que = \{req, grp_j, \sigma\}$ from the network log file on the sink or sensor nodes.

2. The law authority verifies that the group signature is valid, and then uses the partial group private key X_G to compute $Y_i = (U_2 \times {U_1}^{-X_G} \bmod P)$ since

$$U_2 \times {U_1}^{-X_G} \bmod P = G^R Y_i \times (F^R)^{-X_G} \bmod P$$
$$= G^R Y_i \times G^{-R} \bmod P = Y_i.$$

Note that in [17], for the user tracing phase (called *Open* algorithm in a group signature technique), the authors just mention using X_G to decrypt $({U_1}^{\frac{P-1}{Q}} \bmod P, {U_2}^{\frac{P-1}{Q}} \bmod P)$ to get $(G^{\frac{P-1}{Q} x_i} \bmod P)$ and return i (we think here i should be Y_i because i is not defined in [17]). However, the authors do not provide any details on how to decrypt $({U_1}^{\frac{P-1}{Q}} \bmod P, {U_2}^{\frac{P-1}{Q}} \bmod P)$ to get $(G^{\frac{P-1}{Q} x_i} \bmod P)$ and how to derive Y_i from $(G^{\frac{P-1}{Q} x_i} \bmod P)$. Here we assume that the former process is: $(G^{\frac{P-1}{Q} x_i} \bmod P) = ({U_2}^{\frac{P-1}{Q}} \times ({U_1}^{\frac{P-1}{Q}})^{-X_G} \bmod P)$, and the latter one is: $Y_i = ((G^{\frac{P-1}{Q} x_i})^{\frac{Q}{P-1}} \bmod P)$ since

$$(G^{\frac{P-1}{Q} x_i})^{\frac{Q}{P-1}} \bmod P = G^{x_i} \bmod P = Y_i.$$

Compared to the *Open* algorithm of the Camenisch and Groth scheme which requires four modular exponentiation operations on the law authority, our proposed *Open* algorithm mainly requires one modular exponentiation operation on the law authority, and thus is more efficient.

3. Then the law authority reports Y_i to owner j via a secure channel. Owner j can look up the record (Y_i, UID_i) to find the corresponding identity UID_i, and then replies UID_i to the law authority via a secure channel. At this point, the law authority and only the law authority gets to know about which particular user is responsible for the network access in audit.

VI. Security Analysis

We evaluate the security of our work by analyzing its fulfillment of the security requirements described in Section I. As described in Section IV.C, the key management scheme upon which our protocol is built is a variation of group signature. Thus it inherits the security properties of group signature, i.e., correctness, unforgeability, anonymity, unlinkability, traceability. For more information about these properties, the reader is referred to [19].

A. User Authentication

By correctness, in APAC, a group signature σ generated by a legitimate user can surely be identified by the aforementioned verification procedure. By unforgeability, only a group member can sign a fresh query on behalf of the group. Thus, in order to pass the signature verification of the sink or sensor nodes, each user has to register to at least one network owner, then the network owner distributes him/her partial member secret key. Thus, the network owner enforces strict access control by user registration.

B. Integrity Protection of Query Command

In APAC, an authorized user uses a group signature technique to authenticate the query $\{req, grp_j\}$. The sink or sensor nodes know the group public key of the corresponding network owner, and thus can verify the message $\{req, grp_j, \sigma\}$ as well as $\{req, grp_j\}$. Therefore, an adversary cannot modify the query command and then pass the verification of the sink or sensor nodes.

C. Resistance to Node and User Compromised Attacks

As described in Section V.B, only the group public keys of the network owners and the access privilege of each group are pre-loaded on every node. Obviously, no matter how many sensor nodes are compromised, a benign node and sink will not grant the adversary any access privilege. Also, as described in Section V, even if some users are compromised, a benign node or sink will not grant the adversary any access privilege that is beyond the privileges of the compromised users.

D. Limiting User Access Privileges

Each user's activities can be restricted by group division. As described in Section V.E, to pass the verification of the sink or sensor nodes, the query req included in every access request should be set according to the access privilege of the group.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2014.2347311, IEEE Transactions on Wireless Communications

E. Privacy Preservation against the Adversary and the Network Owner

In APAC, users of a group and their group manager (i.e., the corresponding network owner) have no knowledge of the full group private key nor can they compute it. Thus, by anonymity and unlinkability, both the adversary (even by compromising sensor nodes and other users) and the network owner can neither link a group signature to the corresponding user who is responsible, nor link two different group signatures to the same particular user. Moreover, each communication session in APAC is identified only through one fresh random number, which again discloses nothing regarding the user identity information.

F. Privacy Preservation against the Law Authority

The law authority has no knowledge regarding to whom a member secret key is assigned because APAC allows a late binding between member secret keys and network users. Further, it is the network owners' sole responsibility to generate partial member secret keys and then assign them to each user without any involvement of the law authority. Because no other entities except the key holder himself has the knowledge of the corresponding member secret key, and can therefore, generate the given signature, key holder must be a member of the user group grp_j. This audit result satisfies our two requirements. First, the result only reveals nonessential attribute information of the user (i.e., the user group identity grp_j) and still protects user privacy. Second, the result is sufficient for user accountability and limiting user access privileges from the perspective of the network. In other words, the sink and sensor nodes can know which group a responsible entity is from. Note that the access privilege of each group is pre-loaded on the sink and each node.

G. User Accountability

In the cases of attacks and disputes, the responsible users and/or network owners can be audited and pinpointed. As described in Section V.H, on the one hand, the verifiers (i.e., the sink, the nodes, and the owners) can know which network owner is responsible for a particular network access. This is sufficient for user accountability and limiting user access privileges purpose from the perspective of the network. On the other hand, the law authority could track any particular user through the cooperation from the corresponding network owner. Moreover, since the law authority and the network owners do not know any full member secret key, both the law authority and the network owners cannot impersonate any legitimate user.

VII. Implementation and Performance Evaluation

We evaluate APAC by implementing all components on an experimental test-bed.

A. Implementation and Experimental Setup

Here we assess the costs incurred by APAC on two kinds of common sensor platforms, i.e., Imote2 and TelosB. The Imote2 has the Intel PXA271 XScale 32-bit processor running at 13 to 416 MHz. Also, the TelosB mote has an 8-MHz CPU, 10 kB of RAM, and 48 kB of ROM. Our implementation has the authority, network owner, network user, sink, and sensor node side programs. The protocols operated by the first four entities have been implemented in C (using OpenSSL [20]) and executed in Laptop PCs (with 2-GB RAM) under Ubuntu 11.04 environment with different computational power. In addition, the sensor node side programs are written in nesC. Our motes run TinyOS [21] 2.x. In our experiment, we set $l_n = l_P = 1024, l_Q = 282$, and $l_E = 404$. For key establishment, we set the bit length of c and d as 160. Such key lengths provide security equivalent to 1024-bit RSA, which is considered secure enough for now and immediate future. Also, we choose to use SHA-1 function as our base hash function. The Elliptic Curve Scalar Multiplication (ECSM) operation of William&Mary (WM)-ECC library [22] and exponentiation operation of WM-RSA library are employed in APAC. Throughout this paper, unless otherwise stated, all experiments on Laptop PCs (resp., sensor nodes) were repeated ten thousand times (resp., one thousand times) for each measurement in order to obtain accurate average results.

B. Evaluation Results

We use the following four metrics to evaluate APAC, namely, memory overhead, execution time of each operation of APAC, message overhead, and energy overhead. The memory overhead refers to the amount of data space consumed by the real implementation.

As referred by [23], the size of the group signature with the Camenisch's method [17] is about 812 bytes. The length of each access request is $|Que| = |req| + |grp_j| + |\sigma| = 852$ btyes, where the lengths of the query req and the identity of a group grp_j are set to 32 bytes and 8 bytes, respectively. As mentioned in Section V.A, in order to improve the communication efficiency due to receiving access request message, other group signature algorithms with shorter group signature (e.g., 200-byte group signature with the Boneh *et al.*'s method [24]) can just as easily by applied in our proposed protocol.

Table I presents the measured running time for some phases of our proposed protocol, which is tested on Laptop PCs with different computational power. For example, the execution time of the law authority and a network owner for the system setup phase are 361.117 ms and 76.698 ms on a 1.6-GHz Laptop PC, respectively. Also, the execution time of a user for the user query generation phase is 7.761 ms on a 1.8-GHz Laptop PC.

Our experiments show that for distributed access control, the signature verification time on an Imote2 mote is 39 ms while that on a TelosB mote is 1.61 seconds. Thus, the computation complexity of the proposed protocol is comparable to that of those secure data access control methods (e.g., [2]–[5], [15]). Table I also shows the signature verification time on the sink side for centralized access control. For example, the signature

TABLE I
RUNNING TIME FOR SOME PHASES OF OUR PROPOSED PROTOCOL.

	System setup (authority)	System setup (owner)	New user joining (user)	New user joining (owner)	User query generation	Signature verification (sink)
Time (CPU = 1.6 GHz) (ms)	361.117	76.698	2.055	53.721	8.663	10.218
Time (CPU = 1.8 GHz) (ms)	322.170	68.129	1.864	47.116	7.761	9.061
Time (CPU = 2 GHz) (ms)	285.880	61.227	1.752	41.593	6.945	8.144
Time (CPU = 2.2 GHz) (ms)	263.682	55.741	1.623	38.307	6.368	7.340
Time (CPU = 2.4 GHz) (ms)	239.663	50.444	1.440	35.304	5.867	6.808
Time (CPU = 2.6 GHz) (ms)	220.151	47.081	1.265	33.150	5.396	6.276
Time (CPU = 3.1 GHz) (ms)	182.278	39.518	1.072	27.307	4.476	5.228

TABLE II
RUNNING TIME FOR THE REMAINING PHASES OF OUR PROPOSED PROTOCOL.

	User tracing (authority)	User revocation (unrevoked user)	Key establishment (the sink or a network user)
Time (CPU = 1.6 GHz) (ms)	12.015	0.891	0.6515
Time (CPU = 1.8 GHz) (ms)	10.730	0.776	0.591
Time (CPU = 2 GHz) (ms)	9.652	0.689	0.521
Time (CPU = 2.2 GHz) (ms)	8.683	0.691	0.473
Time (CPU = 2.4 GHz) (ms)	8.031	0.629	0.425
Time (CPU = 2.6 GHz) (ms)	7.387	0.537	0.4085
Time (CPU = 3.1 GHz) (ms)	6.213	0.463	0.3285

verification time for the sink is about 6.276 ms on a 2.6-GHz Laptop PC.

The energy consumption on each node can be estimated by the formula $E = U * I * t$, where E denotes the power in millijoules (mJ), U is the voltage in volts (V), I is the current draw in milliamps (mA) and t is the time in seconds. From the Crossbow data sheet for TelosB, when a TelosB is in active mode, $V = 3$ V and $I = 1.8$ mA. In an example calculation using a signature verification time of 1.61 seconds on a TelosB, the amount of energy required to execute is as follows:

$$E = 3 * 1.8 * 1.61 = 3.6888\ mJ.$$

From the Crossbow data sheet for Imote2, when an Imote2 is in active mode (13 MHz, radio off), $V = 4.5$ V and $I = 31$ mA. Thus, the consumed energy of signature verification on an Imote2 is estimated to be 5.4405 mJ.

The energy consumption on receiving a message of x bytes is $E_r = U*I_r*x*8/d_r$, where I_r is the current draw in receiving mode and d_r is the data rate (bits per second). According to the date sheet of each platform, d_r is 250 kbps for TelosB and Imote2, I_r is 23 mA for TelosB, and I_r is 44 mA for Imote2 (with 13 MHz and radio Tx/Rx). Thus, we can calculate the energy consumption for the receipt of each access request message on TelosB and Imote as 3*23*852*8/250000=1.8812 mJ and 4*44*852*8/250000=4.7985 mJ, respectively. Thus, for our protocol, the total energy consumption due to the receipt of each access request message from the users and the signature verification are 3.6888 mJ + 1.8812 mJ =5.57 mJ and 5.4405 mJ + 4.7985 mJ =10.239 mJ on TelosB and Imote2, respectively. Assuming two (rsp., three) new AA alkaline batteries with a capacity of 2850 mAh, we can perform around 3.684 million (rsp., 3.006 million) instances of the receipt of an access request with a signature verification of the proposed protocol on TelosB (rsp., Imote2).

As described above, user tracing and user revocation phases just incur computation overhead on the law authority and an unrevoked user, respectively. Table II presents the measured running time for the remaining phases of our proposed protocol. For example, the execution time of the law authority for the user tracing phase is 7.387 ms on a 2.6-GHz Laptop PC. Additionally, for the user revocation phase, each unrevoked user on 2.4-GHz Laptop PC takes 0.629 ms. Also, the sink (or a user) on a 1.6-GHz Laptop PC consumes about 0.6515 ms to generate a session key (i.e., an ECSM operation).

We implement the interface of TinyOS 2.1.1 [21] to provide symmetric key cryptography using the hardware security support in IEEE 802.15.4 radio components (e.g., CC2420). Table III shows the execution time of hardware AES encryption (including encryption and successful transmission) for TelosB with the plaintext (i.e., the sensor data $sensor\_data$ in this paper) of different lengths (from 4 bytes to 60 bytes in increments of 16 bytes). In our implementation, we choose Cipher Block Chaining Message Authentication Code mode (leveraging the same underlying 128-bit AES encryption), and the maximum length of packet payload is set to be 120 bytes. The experiments were repeated seventy thousand times for each measurement in order to get fairly accurate average results. From Table III, it can be seen that the time consumed by hardware encryption implementation is extremely small. For example, even with 100-byte plaintext as input, the encryption and successful transmission procedures take an average of 27.70524 ms on TelosB motes. The implementation of hardware AES encryption on a TelosB mote uses 301 bytes of RAM and 11,616 bytes of ROM, respectively. The resulting size of our implementation corresponds to 2.94% and 23.63% of the RAM and ROM capacities of TelosB, respectively. Also, it takes about 1.55 s for a TelosB mote to generate a session key (i.e., an ECSM operation). It has been reported [25] that it takes about 11.8 ms for an Imote2 mote to generate a session key.

Since only signature verification, key establishment, and data encryption phases could possibly be executed on resource-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2014.2347311, IEEE Transactions on Wireless Communications

TABLE III
THE EXECUTION TIME OF HARDWARE AES ENCRYPTION FOR TELOSB WITH THE PLAINTEXT OF DIFFERENT LENGTHS.

Length of the plaintext (byte)	4	20	36	52	68	84	100	116
Time(ms)	20.67262	21.40111	22.1121	22.1405	23.4911	24.77953	27.70524	29.3201

constrained sensor nodes, the above experimental results show the efficiency of the proposed protocol in practice.

VIII. CONCLUSION

In this paper, we have proposed APAC, which, to the best of our knowledge, is the first attempt to establish an accountable access control framework with a sophisticated user privacy protection model tailored for WSNs. The security analysis has demonstrated APAC can achieve the requirements of the protocol of this kind. We have implemented the protocol on real mobile devices and sensor platforms with limited-resource. Experimental results have shown that our approach is feasible for real-world applications.

REFERENCES

[1] H. Guo and Z. Sun, "Channel and energy modeling for self-contained wireless sensor networks in oil reservoirs," *IEEE Trans. Wireless Commun.*, vol. 13, no. 4, pp. 2258-2269, April 2014.

[2] S. Ruj, A. Nayak, and I. Stojmenovic, "Distributed fine-grained access control in wireless sensor networks," in *Proc. IEEE IPDPS*, pp. 352-362, 2011.

[3] J. Hur, "Fine-grained data access control for distributed sensor networks," *Wireless Networks*, vol. 17, no. 5, pp. 1235-1249, July 2011.

[4] S. Yu, K. Ren, and W. Lou, "FDAC: Toward fine-grained distributed data access control in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 4, pp. 673-686, April 2011.

[5] G. Bianchi, A. T. Capossele, C. Petrioli, and D. Spenza, "AGREE: exploiting energy harvesting to support data-centric access control in WSNs," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2625-2636, Nov. 2013.

[6] Taking the Pulse of the Planet: EPA's Remote Sensing Information Gateway. http://www.epa.gov/geoss/.

[7] NOPP, http://www.nopp.org/.

[8] D. He, S. Chan, S. Tang, and M. Guizani, "Secure data discovery and dissemination based on hash tree for wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4638-4646, Sept. 2013.

[9] S. Zhu, S. Setia, and S. Jajodia, "LEAP+: Efficient security mechanisms for large-scale distributed sensor networks," *ACM Transactions on Information and System Security*, vol. 2, no. 4, 2006.

[10] Y. Keung, B. Li, and Q. Zhang, "The intrusion detection in mobile sensor network," in *Proc. ACM MobiHoc '10*, pp. 11-20, 2010.

[11] S. Rahman, N. Nasser, and T. Taleb, "Secure timing synchronization for heterogeneous sensor network using pairing over elliptic curve," *Wireless Communications and Mobile Computing*, vol. 10, no. 5, pp. 662-671, 2010.

[12] J. Deng, R. Han, and S. Mishra, "Secure code distribution in dynamically programmable wireless sensor networks," in *Proc. IPSN '06*, 2006.

[13] H. Lu, J. Li, and H. Kameda, "A secure routing protocol for cluster-based wireless sensor networks using ID-based digital signature," in *Proc. GLOBECOM*, pp. 1-5, 2010.

[14] P. Gupta and M. Chawla, "Privacy preservation for WSN: a survey," *International Journal of Computer Applications*, vol. 48, no. 3, pp. 11-16, June 2012.

[15] D. He, J. Bu, S. Zhu, S. Chan, and C. Chen, "Distributed access control with privacy support in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3472-3481, Oct. 2011.

[16] R. Zhang, Y. Zhang, and K. Ren, "Distributed privacy-preserving access control in sensor networks," *IEEE Trans. Parallel and Distrib. Syst.*, vol. 23, no. 8, pp. 1427-1438, Aug. 2012.

[17] J. Camenisch and J. Groth, "Group signatures: better efficiency and new theoretical aspects," in *Proc. SCN '04*, 2004.

[18] T. Zhang, J. He, and H. Yu, "Secure localization in wireless sensor networks with mobile beacons," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 732381, 2012.

[19] E. Bresson and J. Stern, "Efficient revocation in group signatures," in *Proc. PKC*, 2001.

[20] OpenSSL, http://www.openssl.org.

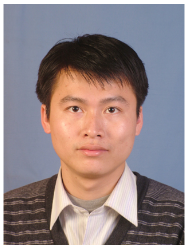
[21] Tiny OS. http://www.tinyos.net.

[22] H. Wang, B. Sheng, C. C. Tan, and Q. Li, "WM-ECC: An elliptic curve cryptography suite on sensor motes," College of William and Mary, Computer Science, Williamsburg, VA, Tech. Rep. WM-CS-2007-11, 2007.

[23] N. Rabadi and S. Mahmud, " A broadcast protocol with drivers anonymity for vehicle-to-vehicle communication networks," *International Journal of Vehicle Information and Communication Systems*, vol. 2, no. 1-2, pp. 1-26, 2009.

[24] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Proc. Crypto '04*, 2004.

[25] A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks," in *Proc. IEEE IPSN*, pp. 245-256, 2008.



Daojing He (S'07-M'13) received the B.Eng.(2007) and M. Eng. (2009) degrees from Harbin Institute of Technology (China) and the Ph.D. degree (2012) from Zhejiang University (China), all in Computer Science. He is currently a Professor in the Software Engineering Institute, East China Normal University, P.R. China. His research interests include network and systems security. He is an associate editor or on the editorial board of some international journals such as *IEEE Communications Magazine* and IEEE/KICS *Journal of Communications and Networks*.

Sammy Chan (S'87-M'89) received his B.E. and M.Eng.Sc. degrees in electrical engineering from the University of Melbourne, Australia, in 1988 and 1990, respectively, and a Ph.D. degree in communication engineering from the Royal Melbourne Institute of Technology, Australia, in 1995. From 1989 to 1994, he was with Telecom Australia Research Laboratories, first as a research engineer, and between 1992 and 1994 as a senior research engineer and project leader. Since December 1994, he has been with the Department of Electronic Engineering, City University of Hong Kong, where he is currently an associate professor.

Mohsen Guizani (S'85-M'89-SM'99-F'09) is currently a Professor and the Associate Vice President for Graduate Studies at Qatar University, Qatar. He received his B.S. (with distinction) and M.S. degrees in Electrical Engineering; M.S. and Ph.D. degrees in Computer Engineering in 1984, 1986, 1987, and 1990, respectively, from Syracuse University, Syracuse, New York. His research interests include Computer Networks, Wireless Communications and Mobile Computing, and Optical Networking. He currently serves on the editorial boards of six technical Journals and the Founder and EIC of "Wireless Communications and Mobile Computing" Journal published by John Wiley (http://www.interscience.wiley.com/jpages/1530-8669/). Dr. Guizani is an IEEE Fellow and a Senior member of ACM.