Distributed Access Control with Privacy Support in Wireless Sensor Networks

Daojing He, *Student Member, IEEE*, Jiajun Bu, *Member, IEEE*, Sencun Zhu, Sammy Chan, *Member, IEEE*, and Chun Chen, *Member, IEEE*

***Abstract*—A distributed access control module in wireless sensor networks (WSNs) allows the network to authorize and grant user access privileges for in-network data access. Prior research mainly focuses on designing such access control modules for WSNs, but little attention has been paid to protect user's identity privacy when a user is verified by the network for data accesses. Often, a user does not want the WSN to associate his identity to the data he requests. In this paper, we present the design, implementation, and evaluation of a novel approach, *Priccess*, to ensure distributed privacy-preserving access control. In Priccess, users who have similar access privileges are organized into the same group by the network owner. A network user signs a query command on behalf of his group and then sends the signed query to the sensor nodes of his interest. The signature can be verified by its recipient as coming from someone authorized without exposing the actual signer. In addition to the theoretical analysis that demonstrates the security properties of Priccess, this paper also reports the experimental results of Priccess in a network of Imote2 motes, which show the efficiency of Priccess in practice.**

***Index Terms*—Authentication, privacy, distributed access control, wireless sensor network.**

I. Introduction

The primary purpose of deploying a wireless sensor network (WSN) is to monitor the physical world and provide observations for various applications. As WSNs are usually deployed in an environment that is vulnerable to many security attacks, it is critical to control the access to the sensor nodes (e.g., reading sensor data), especially when there are many users in the system. Additionally, different users may have different access privileges. For example, in the case of a WSN deployed in a battlefield, a soldier only needs to access the data related to his task, but a higher rank officer often requires information gathering for an overall manoeuvre and therefore should have more information access privileges than a soldier [1]. The application will be compromised if access control is not properly enforced. Access control can be executed by two approaches, namely, centralized and distributed. A centralized access control approach requires a base station to be involved whenever a user requests to get authenticated and access the information stored in the sensor nodes. Unfortunately, it is inefficient, not scalable, and vulnerable to many potential attacks along the long communication path [2]. For example, for sensor networks deployed in extreme and hazardous environments such as oceans and animal habitats, it may be impossible or prohibitive to maintain a stable communication connection between an in-network base station and the outside network. Therefore, a centralized approach makes sense only for small, experimental networks, but not for large scale sensor networks. On the other hand, in distributed access control, the authorized users can enter the sensor field to directly access data on sensor nodes without involving a base station. This approach can avoid weaknesses such as single point of failure, performance bottleneck, which are inevitable in the centralized case. These advantages together have led to recent increasing popularity of distributed data access control [1]-[7].

In a large scale WSN, owners and users are normally from very different groups and, sometimes, even have conflicting interests with each other. Therefore, a user might not want the WSNs or others to associate his identity to the data he requests, or the time he accesses the network, etc. For example, some current projects including GEOSS [8] and NOPP [9] are constructing large-scale WSNs to adaptively observe the earth-ocean-atmosphere system. The sensed data may be of interest to numerous users from both public and private sectors, ranging from individual users to universities, government research centers, and business companies. The network owner and users may not fully trust each other, due to diversified interests (e.g., GEOSS [8] involves 61 countries, NOPP [9] involves the Defense Advanced Research Projects Agency, the Department of State and the Department of Homeland Security among others). Therefore, there is a growing demand for protecting users' data access privacy (e.g., [10]). In particular, a user may wish to keep confidential whether he accessed the sensor data, the data types he was interested in, or from which nodes he obtained the data, since the disclosure of such information may be used against his interest. For example, an oil company interested in the data of an ocean sensor network [8], [9] may want to hide its access privacy from both the network owner and other network users who might potentially be its business competitors [10]. The above cases necessitate distributed privacy-preserving access control.

A privacy-preserving access control in WSNs should satisfy

Manuscript received December 23, 2010; revised March 27, 2011 and May 18, 2011; accepted July 10, 2011. The associate editor coordinating the review of this paper and approving it for publication was Z. Han.

This work was supported by the National Science Foundation of China (Grant No. 61070155), the Program for New Century Excellent Talents in University (NCET-09-0685), grant NSF CAREER 0643906, and a grant from the Research Grants Council of the Hong Kong SAR, China [Project No. City U 111208].

D. He, J. Bu, and C. Chen are with the College of Computer Science, Zhejiang University, P.R. China (e-mail: hedaojinghit@gmail.com).

S. Zhu is with the Department of Computer Science and Engineering and the College of Information Sciences and Technology, Pennsylvania State University, USA (e-mail: szhu@cse.psu.edu).

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).

Digital Object Identifier 10.1109/TWC.2011.072511.102283

the following requirements: (1) **User Authentication**: user authentication needs to be enforced for sensor data in WSNs so that the information will not be obtained by unauthorized entities; (2) **User Privacy-Preserving**: a network user may want to hide his data access privacy from anyone else including the network owner and other network users. More specifically, anyone else should be prevented from either knowing who is the sender of the query command, or whether two query commands originate from the same (unknown) sender; (3) **Integrity Protection of Query Commands**: the adversary may try to modify the query command constructed by a user, and a secure access control method should support the integrity protection of the query command; (4) **Node Compromise Tolerance**: the adversary cannot impersonate any network user by compromising nodes; (5) **Scalability**: the protocol should be efficient even in a large scale WSN with many users and many nodes. (6) **Freshness**: to defend against replay attacks, a node should have the capability of freshness checking for any query message; (7) **Limits of Access Privileges**: access restriction may be enforced for users with different access privileges; (8) **Dynamic Participation**: new users can easily join the network, and users can easily be revoked when they are expired. (9) **Availability of Secure Channels between a Network User and Sensor Nodes**: In some application scenarios, it is necessary to establish secure channels between a network user and the targeted nodes. (10) **Efficiency**: Due to the limited energy, processing and storage resources of sensor nodes, a cryptographic technique should be efficient.

Obviously, designing a distributed privacy-preserving access control in WSNs is a non-trivial task because wireless networks are vulnerable to attacks and sensor nodes are resource constrained. In particular, a network user hopes to protect his data access privacy from the network owner, although the network owner controls the whole network. Despite significant progress in WSNs security [5], distributed privacy-preserving access control has drawn attention only very recently. The only distributed privacy-preserving access control protocol we know of is *DP^{2}AC* [6], which employs the blind signature technique. However, a recent study [11] reports that DP^{2}AC is not fine-grained, since each anonymous user has the same access privilege. Further, we observe that DP^{2}AC has a number of security weaknesses and efficiency problems (more detailed description will be given in Section II). Therefore, to the best of our knowledge, until now no secure protocol for privacy-preserving access control has been proposed.

This paper makes two main contributions:

(1) We first identify the characteristics of a single-owner multi-user sensor network and present the requirements of distributed privacy-preserving access control. Then we propose a novel approach to ensure distributed privacy-preserving access control, called *Priccess*, which is built on a ring signature technique. Since a ring signature scheme was not originally designed for privacy-preserving access control, a direct application of the method cannot satisfy requirements (3), and (6)-(9), which are very challenging for ensuring secure, efficient and robust distributed privacy-preserving access control. To address these issues, some additional mechanisms are incorporated into the design of the proposed protocol. Finally, Priccess satisfies all of the above requirements. In addition, our theoretical analysis demonstrates the security properties of Priccess.

(2) We also implement the proposed protocol in a network of Imote2 motes. Evaluation results show the efficiency of Priccess in practice. To the best of our knowledge, this is also the first implemented privacy-preserving access control on the WSN platform.

The rest of the paper is structured as follows. In Section II, we first survey and analyze the related work, and then discuss their security weaknesses. Section III presents the network, trust and adversary models. Section IV describes our proposed scheme. Then in Section V, some important issues about our scheme are given. Section VI provides theoretical analysis of the security properties of Priccess. Section VII describes the implementation and experimental results of Priccess via real sensor platforms. Finally, Section VIII concludes this paper and points out future research directions.

II. Related Work

In the literature, some schemes have been proposed for achieving access control in sensor networks [1]-[7]. Digital signatures have been demonstrated to be feasible for resource-constrained sensor nodes [1], [2]. Some approaches in [3], [4] make use of the simple operations such as one-way hash functions and exclusive-OR operations to enable efficient access control. In addition, the least privilege scheme in [5] can be used to achieve a specific type of access control, in which a user can only access the sensor data at a predetermined physical path in the field. We observe that all these works [1]-[5] just focus on designing access control modules for WSNs, but do not pay attention to protecting user's identity privacy when a user is verified by the network for data accesses.

Recently, a novel privacy-preserving access control protocol named DP^{2}AC has been proposed [6]. In order to achieve privacy-preserving access control, the use of blind signatures in token generation ensures that tokens are publicly verifiable yet unlinkable to user identities. However, as reported in a recent study [11], DP^{2}AC is not fine-grained, since each anonymous user has the same access privilege. Further, due to the use of blind signatures, each query command cannot be signed by a network user. More specifically, the protocol overlooks the authentication of query commands constructed by the network users. As a result, an adversary can easily intercept the token and impersonate any authorized network users to modify the query command and then obtain the responses from sensor nodes. Therefore, their approach is not practical for securing real world applications. Moreover, DP^{2}AC is not efficient in three aspects. First, network-wide flooding is required once token-reuse detection runs. Second, for token reuse detection, the protocol needs to store tokens in local memory of every node. Once a token has been used (i.e., a user query command is sent), it is permanently stored and will not be released from the memory. Since a node has limited memory capacity, this makes DP^{2}AC rather impractical. Also, as each token only allows the user to access the nodes once, the number of user queries allowed in DP^{2}AC is thus limited. Clearly, a practical access control protocol should not limit the

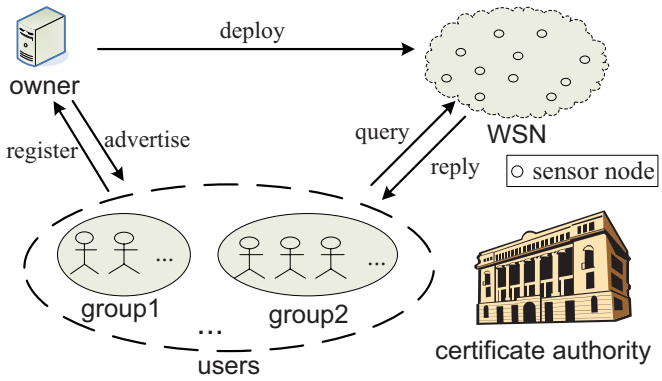

Fig. 1. A system overview of distributed privacy-preserving access control in WSNs.

number of user queries. Third, users can only access one node at a time, while majority of actual access requests are targeted to many nodes via broadcast. Very recently, in [7], we have considered using ring signature to achieve distributed privacy-preserving access control, but do not discuss it in much great detail.

III. Network, Trust and Adversary Models

A. Network Model

As shown in Fig. 1, a WSN consists of a large number of resource-constrained sensor nodes, many sensor network users, a single network owner and an offline certificate authority (CA). The sensor nodes are used to sense conditions in their local surroundings and report their observations to network users based on various query commands. The network users (e.g., soldiers) use access devices such as PDAs or laptop PCs to access the sensed data. The network owner bootstraps the keying materials for access devices to enforce the access control policy and it cannot be compromised. Additionally, to achieve fine-grained data access control, users who have similar access privileges should be organized into the same group by the network owner. The security requirements of distributed access control can be achieved by employing digital signatures, where each entity has a public/private key pair. For entity authentication, the public key of each entity must be authentic to all the entities in the network. Therefore, the network model requires the CA. Everyone trusts the public-key certificates issued by the CA, and the CA has a public key which is publicly known (e.g., built into all the web browsers). Such sensor networks are under construction by many multi-sponsor projects [8], [9].

B. Trust Model

Malicious network owner model: We suppose that the network owner charges users for accessing sensor data, thus enforcing strict access control. The network owner is trusted to provide the appropriate amount of data commensurate with users' payments. This coincides with the typical assumption about service providers. However, the network owner may for various purposes be interested in users' identities and data access patterns (e.g., who are interested in what kinds of data at what locations and times).

Network users are assumed to be selfish, privacy-sensitive, curious and rational. By selfish, we mean that users always try to pay less for more data given any possible opportunity. By privacy-sensitive and curious, we mean that users are reluctant to disclose their own data access privacy but are interested in knowing others'. For example, a user does not leak his data access privacy to the network owner. Also, users are rational, meaning that they would misbehave only when benefiting from doing so. For example, we assume that users do not launch Denial-of-Server (DoS) attacks on the sensor network because this is against their interest in acquiring useful sensor data.

C. Adversary Model

We assume that an adversary can launch both outside and inside attacks. In outside attack, the adversary may eavesdrop, copy and replay the transmitted messages in the WSN. Therefore, for a practical threat model we consider an adversary that is able to eavesdrop all network communications, as well as inject bogus messages or forge non-existing links in the network by launching a wormhole attack. As an inside attack, we assume that the adversary may compromise and control a number of sensor nodes subject to his choice. Additionally, we consider two sybil attacks: one is that the network owner could add a user to a group in which other users are impersonated by the network owner, this would remove the anonymity of this user. The other is that by presenting multiple identities, a malicious user can control a substantial fraction of the system and thereby undermine the security.

In this paper, we mainly focus on access control on sensor nodes side. More specifically, this paper studies how to ensure that only authorized users can gain access to authorized data from the nodes. However, the adversary can impersonate one or more nodes to respond to users' query commands, and thereby introduce arbitrary false information to network users. To address this challenge, many effective mechanisms have been proposed (e.g., [12], [13]). For the network owner and users, they can make use of some automatic diagnosis schemes to identify the malicious nodes [12]. For the nodes, they can use spatial correlation to detect the malicious nodes [13].

IV. Priccess: The Protocol

In this section, Priccess is presented in detail. Before giving the detailed description, we first give an overview of distributed privacy-preserving access control.

A. Overview of Distributed Privacy-preserving Access Control

Fig. 1 shows a system overview of distributed privacy-preserving access control in WSNs. It mainly involves three kinds of participants, the network owner, all sensor nodes, and the network users. The users who want to access the network firstly register to the network owner. Then the network owner divides all users into groups. Network users in the same group have the same access privilege. At the same time, the network owner maintains a group access list pool, which contains the identity and other information (e.g., access privileges) of each group. Here we assume a user, say U_j, belongs to the group with the identity gid. The number of group members is assumed to be N. The network owner

advertises the group access list pool to all users. In addition, the group access list pool is pre-loaded on each node. U_j wants to send a query command to the nodes in such a way that it remains anonymous, yet the nodes are convinced that the query command is indeed from a member of the group with the identity gid. Even though the network owner controls the whole network and has the privilege to divide users into groups, it cannot determine the actual source of the query command. Without loss of generality, we just consider the users who belong to the same group with the identity gid in the following unless specified otherwise.

A viable approach is for U_j to send the nodes a standard digitally signed message. A digital signature scheme allows a user to sign a message with his private key such that any verifier can verify that the message originated from an authorized user. However, such a signature message will directly reveal U_j's identity. Therefore, a standard digital signature cannot be used to achieve privacy-preserving access control.

A more suitable approach is for U_j to send the query command to the sensor nodes through a standard group signature technique. A group signature scheme allows one member of the group to sign a message such that any verifier can just verify that the message originated from a group member without knowing the identity of the actual sender. Unfortunately, a standard group signature algorithm does not solve the "privacy-preserving" access control problem, because it requires the prior cooperation of the other group members to set up, and leaves a user U_j vulnerable to later identification by the group manager (i.e., the network owner).

According to the above analysis, it can be concluded that designing a distributed privacy-preserving access control protocol for WSNs is a difficult task due to the following two reasons. First, due to the limited energy, processing and storage resources of sensor nodes, many cryptographic techniques which incur heavy resource consumption cannot be employed. Second, as described in Section I, the design goal of distributed privacy-preserving access control should include those nine properties. In particular, a network user hopes to protect his data access privacy from the network owner, although the network owner controls the whole network.

B. Overview of Priccess

In this paper, a ring signature technique is introduced to the design of Priccess. In Particular, U_j sends a query command to the sensor nodes through a ring signature algorithm. The ring signature allows a user U_j from a set of possible signers (i.e., a subset of the group) to convince the verifier (i.e., the sensor nodes) that the signer of the signature belongs to the set but the identity of the signer is not disclosed. It protects the anonymity of a signer since the verifier knows only that the signature comes from a member of a ring, but does not know exactly who the signer is. There is no way to revoke the anonymity of the signer. Obviously, the ring signature technique can remedy the security issues of the group signature application. The ring signature is signer-ambiguous in the sense that the verifier is unable to determine the identity of the actual signer in a ring of size r with probability greater than $1/r$. But we notice that because a ring signature scheme was not originally designed for distributed privacy-preserving access control, a direct application of ring signature technique is still unable to meet requirements (3) and (6)-(9). To address these issues, some additional mechanisms are incorporated into the design of the proposed protocol. Detailed description of these mechanisms will be given in Section IV and V.

Priccess consists of six phases: system initialization, user query generation, sensor node verification, establishing secure channels between the network user and sensor nodes, new user joining phase, and user revocation phase. In the system initialization phase, the network owner and all users create their public and private keys. Then the network owner divides all users into groups and maintains a group access list pool. The group access list pool is pre-loaded on the corresponding sensor nodes before they are deployed. In the user query generation phase, if a user has a new query, he will need to construct the query command and the ring signature and then send them to the sensor nodes. In the sensor node verification phase, if the query verification passes then the sensor nodes respond to the user's query command. The new user joining phase is invoked whenever a user wants to join the network while user revocation phase runs whenever a user is to be revoked. In this paper, we just focus on the access control on sensor networks, the secure storage on sensor nodes is out of our scope. Additionally, in Priccess, we choose Elliptic Curve Cryptography (ECC) because ECC has a significant advantage over RSA due to its computational efficiency, small key size, and compact signatures [14].

C. System Initialization

Priccess is based on an efficient ring signature [15]. However, any other ring signature scheme can just as easily be applied in our scheme. Let q denote a large prime number. Two field elements $a, b \in \mathbb{Z}_q$ are chosen such that $4a^3 + 27b^2 \neq 0 (mod\, q)$ in order to define the equation of a non-supersingular elliptic curve $E : y^2 = x^3 + ax + b (mod\, q)$ over $\mathbb{Z}_q$. We define $E(\mathbb{Z}_q)$ as a group for the set of solutions $(x, y) \in \mathbb{Z}_q \times \mathbb{Z}_q$ to the congruence $y^2 = x^3 + ax + b (mod\, q)$ together with a special point ∂ called the point at infinity. Also, a generator point $P = (x_q, y_q)$ is chosen such that its order is a large prime number p over $E(\mathbb{Z}_q)$, where $P \neq \partial$. In such a way, a subgroup G of the elliptic curve group $E(\mathbb{Z}_q)$ with order p is constructed. We choose cryptographic hash functions $H1 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ and $H2 : G \times G \rightarrow \mathbb{Z}_p^*$. Here p, q, E, P, G are public parameters while $H1$ and $H2$ are public functions. In this phase, the network owner (resp. every network user, say U_j) creates its private key $x_{owner} \in \mathbb{Z}_p^*$ (resp. $x_j \in \mathbb{Z}_p^*$) and the corresponding public key $Y_{owner} = x_{owner} \cdot P$ (resp. $Y_j = x_j \cdot P$).

(1) The network owner submits its public key Y_{owner} and identity information (e.g., the owner's social security number or passport) to the CA. And the CA verifies the identity of the network owner upon registration and then generates a public-key certificate for it. The certificate consists of the network owner's public key and identity information plus other information (e.g., an indication of the period of validity of the certificate, some information about the CA), with the whole block signed by the CA.

(2) As shown in Fig. 2, every network user, U_j also registers with the CA through submitting its public key Y_j and its

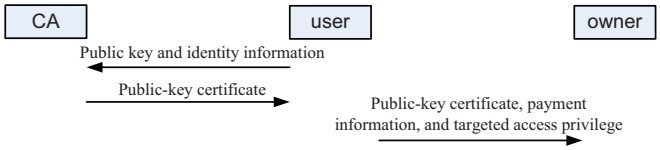

Fig. 2. The partial protocol run of the initiation phase.

targeted region (6)	gid (1)	request (4)	cP	reserved (1)	timestamp (4)

Fig. 3. An example format of the query command Que. The byte size of each field is indicated below the label.

identity information. The CA verifies the identities of each user and then generates a public-key certificate for each user. Each certificate consists of the network user's public key plus other information (e.g., an indication of the period of validity of the certificate, some information about the CA), with the whole block signed by the CA.

(3) U_j delivers its public-key certificate, payment information and targeted access privilege (e.g., the targeted parameters set may include temperature, light, etc) to the network owner.

(4) Upon receiving this message, the network owner puts U_j into a group according to its access privileges. The network owner maintains a group access list pool. Every group access list is composed of gid, $group\ access\ privilege\ mask$, $group\ members'\ public\ keys$ and $public\ keys'\ expiration\ times$. Here gid is a unique number to identify the group. The bit length of gid is assumed to be 8. The information stored at the sensor nodes is divided into multiple group access privilege levels. Additionally, $group\ access\ privilege\ mask$ is a bit map, and each bit represents a specific information or service. For example, if the first bit represents the light parameter, a '1' in this bit indicates that the light parameter is available for all members of this group. Also, $group\ members'\ public\ keys$ consists of the public keys of all users which belong to this group. Besides, $public\ keys'\ expiration\ times$ contains the expiration time of every public key. In order to reduce the communication overhead, the group members' public keys are numbered in consecutive order. The group access list pool is pre-loaded on the corresponding sensor nodes before they are deployed.

(5) The network owner signs the group access list pool by its private key x_{owner} and then advertises the pool plus the digital signature and its public-key certificate to all network users by, say, e-mail or web site announcement. Upon receiving the advertisement message, every user U_j uses his public key to search the access list of the group to which he belongs.

D. User Query Generation

After system initialization, network users can enter the network to access the senor nodes. As discussed before, U_j can obtain his group access list, which is composed of gid, $group\ access\ privilege\ mask$, $group\ members'\ public\ keys$ and $public\ keys'\ expiration\ times$. Here gid is the identity of the group to which U_j belongs. We assume that $group\ members'\ public\ keys$ consists of $Y_1, Y_2, \ldots, Y_N$.

U_j firstly constructs an appropriate query command Que. An example format of Que is depicted in Fig. 3. We consider a scenario where U_j wants to access the nodes in a specific region. For example, U_j is interested in temperature readings from sensors placed in the rectangle of coordinates [10..30,40..60]. All sensor nodes know their geographical locations which can be acquired via deployment knowledge or many existing secure localization schemes (e.g., [16]). Hence U_j needs to add the information about the specific region to the *targeted region* field of Que. Additionally, timestamp T_j is also added into Que by U_j to resist replay attacks. In this case, an existing secure clock synchronization technique (e.g., [17]) is employed in each sensor node. Note that if no such technique is used in a WSN, instead of timestamp, the nonce can be applied to prevent from replay attack. The detailed description about this will be given in Section VI. Here the *request* field represents request detail. An example is "Return the current temperature". Obviously, the *targeted region* and *request* fields are set according to $group\ access\ privilege\ mask$ of U_j's group. In general, in order to support a variety of applications, the format and length of Que in Priccess should be set according to the specified application scenario. For example, in some WSNs, the network users are not aware of the region information. Thus, the *targeted region* field of Que should be omitted.

With the Que, U_j computes $H1(Que)P$. To sign $H1(Que)P$ on behalf of the subgroup $\mho$ from the group $\{U_1, U_2, \ldots, U_N\}$, where $|\mho| = m \leq N$, the signer $U_j \in \mho$ carries out the following steps:

(1) For all $i \in \{1, \ldots, m\}$ and $U_i \neq U_j$, U_j randomly chooses $a_i \in \mathbb{Z}_q^*$ and for which the a_i are pairwise different. Compute $R_i = a_i P (i \neq j)$

(2) Choose a random number $a \in \mathbb{Z}_q^*$.

(3) Compute R_j, where

$$R_j = aP - \sum_{i=1, i \neq j}^{m} H2(H1(Que)P, R_i) Y_i$$

If $R_j = \partial$ or $R_j = R_i$ for some $i \neq j$, then go to step (2). Here $-$ denotes subtraction of points over $E(\mathbb{Z}_q)$.

(4) Compute α, where

$$\alpha = a + \sum_{i=1, i \neq j}^{m} a_i + x_j H2(H1(Que)P, R_i) \bmod p$$

(5) The signature of Que made by the subgroup $\mho$ from the group $\{U_1, U_2, \ldots, U_N\}$ is given by $\sigma = \{R_1, \ldots, R_m, Y_1, \ldots, Y_m, \alpha\}$. Here we use the notation $\{C, D\}$ to denote the concatenation of C and D. After that, U_j sends the message $\{Que, \sigma\}$ to the sensor nodes. To reduce the communication overhead, $\{Y_1, \ldots, Y_m\}$ included in σ can be replaced by its number.

It is worth to note that in Priccess, only a subset of group members in the group have been chosen to generate the ring signature. Thus, the anonymity strength of Priccess varies with m, the size of the chosen signing group. The larger the size of the signing group is, the more anonymous Priccess can be. However, a large size of a signing group incurs large signature and verification overheads. Therefore, in Priccess, it is flexible for each network user to choose the value of m to balance the anonymity strength and overhead.

When a sensor node wants to return the results securely to U_j, it can first establish a secure channel between itself and U_j. The details about setting up such a secure channel will be discussed in Section IV.F.

E. Sensor Node Verification

Upon receiving the message $\{Que, \sigma\}$, each node firstly checks whether the timestamp T_j 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 U_j's group identity gid and $request$ from Que. The node then checks the validity of gid and $request$ according to the group access list pool stored in it. If they are invalid, the message $\{Que, \sigma\}$ is rejected. Otherwise, the node gets $\{Y_1, \ldots, Y_m\}$ from its memory. After that, the node verifies such an alleged signature σ on the query Que as follows:

(1) Compute h_i, where $h_i = H2(H1(Que)P, R_i)$ for all $1 \leq i \leq m$

(2) Check the equation

$$\sigma P = \sum_{i=1}^{m} (R_i + h_i Y_i)$$

The above check should succeed for a valid signature because

$$\sum_{i=1}^{m} (R_i + h_i Y_i) = R_j + h_j Y_j + \sum_{i=1, i \neq j}^{m} (R_i + h_i Y_i)$$

$$= aP + h_j Y_j + \sum_{i=1, i \neq j}^{m} R_i = \sigma P$$

If so, the node gives a response to user U_j; otherwise, the message $\{Que, \sigma\}$ is rejected. Here $+$ denotes addition of points over $E(\mathbb{Z}_q)$.

F. Establishing Secure Channels between the Network User and Sensor Nodes

In some application scenarios, a network user wishes to secretly communicate with the targeted sensor nodes. For example, the sensor nodes wish to send a confidential response to the user. For key establishment, here the protocol implements an Elliptic Curve Diffie-Hellman (ECDH) key exchange or digital envelopes. In the user query generation phase, user U_j randomly chooses $c \in \mathbb{Z}_q^*$ and computes cP. cP is added into the query command Que, where the bit length of cP is 160. As before, a sensor node confirms the validity of $\{Que, \sigma\}$ from U_j. Then the sensor node does the following.

For ECDH key exchange, the node randomly chooses $d \in \mathbb{Z}_q^*$ and generates dP, whereafter it can compute $sk = d(cP)$ as the session key between itself and U_j. Thus, the node can use the key sk to encrypt the required sensor data $sensor\_data$ of U_j. Subsequently, the node gives a response $\{cP, dP, E_{sk}(sensor\_data)\}$ to U_j, where $E_K(X)$ encrypts a message X using a symmetric key K. Upon receiving this response, U_j can generate $sk = c(dP)$ and then obtain $sensor\_data$. Upon establishment of the secure channel, the network user and the sensor node have a shared symmetric key used for the subsequent communication session. This session is uniquely identified through $\{cP, dP\}$.

For digital envelopes, the node generates a random symmetric key sk as the session key between itself and U_j. Subsequently, it uses sk to encrypt $sensor\_data$ through symmetric encryption. At the same time, it encrypts sk using public-key encryption with U_j's public key cP. Then the node sends the encrypted session key plus the encrypted sensor data to U_j. With the private key c, only user U_j is capable of decrypting the session key sk and therefore of recovering the original sensor data.

G. New User Joining Phase

This phase is invoked when a new user, say U_j, hopes to access a WSN after the network is deployed. U_j creates its private key $x_j \in \mathbb{Z}_p^*$ and the corresponding public key $Y_j = x_j \cdot P$. As described in Section IV.C, U_j registers with the CA and then registers to the network owner. Once the network owner accepts the request from U_j, it will assign U_j into one group with the identity gid according to his access privilege. The network owner updates the group access list pool. Then the network owner advertises *New User Joining Message* to the network users, who belong to the group with the identity gid. Obviously, it has minimal impact on the efficiency. Here an example of *New User Joining Message* is "Add a public key Y_j and its expiration time to the group with the identity gid". At the same time, the network owner uses its private key $\{x_{owner}\}$ to sign such a *New User Joining Message* and then advertises it to all sensor nodes. Upon receiving this message, the nodes update the group access list pool stored at themselves respectively. Due to the severe energy constraint of sensor nodes, it is unacceptable that the radio works in a high power level. However, the network owner does not have this energy constraint and can make the radio work in a high power level. For example, 2.4G radio like cc2500 can reach about 1200 meters. Therefore, it is practical that the network owner advertises the new user joining message to all nodes.

H. User Revocation Phase

In most cases, the network owner hopes to limit the time period for which each user U_j can access the network. As described in Section IV.C, to achieve this goal, the owner sets the expiration time for the public key of every user. Obviously, a general access device (e.g., PDA, Laptop PC) can easily achieve clock synchronization. Thus, each user can directly delete a public key from the group access list pool once it is expired. There are two cases for each node to revoke a user as follows. (1) If a secure clock synchronization technique is employed in each node, each node can directly delete the public key from the group access list pool once it is expired. (2) Otherwise, once the public key of a user is expired, the network owner needs to sign a *User Revocation Message* and then advertises it to all nodes by using its private key $\{x_{owner}\}$. Here an example of *User Revocation Message* is "Delete a public key Y_j from the group with the identity gid".

V. DISCUSSION

So far, we have elaborated the procedures of our protocol. By the protocol, we can achieve privacy-preserving access control. Besides, some important issues need to be considered.

A. Group Division

As described in Section IV.C, the network owner divides all network users into groups. We need to choose the group size carefully to balance the security and efficiency of Priccess. In the following description, we consider the scenario where a user U_j constructs a query command Que and signs it, then sends it to the sensor nodes. Here we assume that U_j belongs to the group with the identity gid. We also assume that the size of the group is N. The size of a group is directly related to the maximal anonymity strength of its member. As mentioned in Section IV.B, the verifier (including all sensor nodes, the network owner, and all users except U_j) is unable to determine the identity of the actual signer in a group of size N with probability greater than $1/N$. In the system initialization phase, when a user receives the advertisement about his group access list, he obtains the size N of his group. In other words, he has the knowledge that the verifier is unable to determine his identity with probability greater than $1/N$ in the case that he sends a query command. Based on such a strength of anonymity, a user can decide whether to send a query (i.e., whether to accept the network service). Thus, as the service provider, the network owner is stimulated to keep the size of each group large enough. On the other hand, as described in Section VII, not only the maximal execution time for user query generation phase but also the maximal node verification delay is directly proportional to the size of the group.

B. User Colluding Attacks

As described in Section III.B, the users are assumed to be privacy-sensitive and curious. Hence a user does not leak his data access privacy and patterns to other users. On the other hand, if multiple users of the same group collude, the strength of user anonymity will be decreased. We assume that p users of a group with size N collude. In this case, the strength of user anonymity is decreased. More exactly, Priccess can ensure that the adversary is unable to determine the identity of a normal user with probability greater than $1/(N-p)$ in the case that the user sends a query command.

C. Sybil Attacks

A direct application of ring signature technique will make Priccess vulnerable to the two sybil attacks described in Section III.C. To defeat these sybil attacks, some remedies for Priccess are suggested as follows. As described in Section IV.C, while registering to the CA, a user needs to submit his public key and identity information to the CA. In system initialization phase, all users' public-key certificates plus the group access list pool, with the whole block signed by the network owner are advertised to all users, by say, e-mail or web site announcement. With such a message, U_j can use the public key of the CA to check whether any user in the same group has registered to the CA. If the result is negative, U_j will reject the service of accessing the WSN. Since the CA ensures that no one (e.g., a user or the network owner) can register with multiple identities, these two attacks cannot be launched. As the service provider, the network owner is stimulated to transmit public-key certificates of all users to each user. If not, the users can reject the network service.

D. Scalability and Dynamic Participation

Considering the memory constraint on sensor nodes, only the public key of the network owner and the group access list pool are pre-loaded in each sensor node. Here we assume that the total number of network users is num and the length of the public key of each network user U_j is 160 bits. We also assume the length of the expiration time of each public key is 2 bytes. The size of the group access list pool is about $22*num$ bytes. For example, the flash memory of Imote2 can store a group access list pool with about 1500 network users. Thus, we believe that the storage overhead of Priccess will not be a problem for sensor nodes. In addition, we consider the computation complexity on sensor nodes. As before, the computation complexity on sensor nodes (or network users) depends only on the size of each ring (specifically, the size of the chosen signing group) rather than the total number of sensor nodes or network users. In Section VII.B, we will show that Priccess can be efficiently applied on sensor nodes. According to the above analysis, it is concluded that Priccess can ensure scalability. As described in Section IV.G and IV.H, Priccess can provide dynamic participation.

VI. Security Analysis

In the following, we will analyze the security of Priccess to verify whether the security requirements mentioned in Section I have been satisfied. Note that requirement (5), (8) and (9) have been analyzed in Section V.D and IV.F.

User Authentication: As described in IV, in order to pass the signature verification of sensor nodes, each user has to register to the CA and the network owner, then the network owner relegates him to a group according his access privilege. To send a valid message $\{Que, \sigma\}$, a network user needs to sign the query command Que with his private key and the public keys of all chosen group members. Therefore, the network owner enforces strict access control by user registration.

User Privacy-Preserving: As before, the use of ring signature can ensure user privacy-preserving. More specifically, ring signature can lead to desirable user privacy-preserving property: We assume that m group members in the group have been chosen to generate the ring signature. The ring signature is signer-ambiguous in the sense that the verifier (e.g., the network owner, sensor nodes, and other network users) is unable to determine the identity of the actual signer with probability greater than $1/m$. Further, based on ring signature, Priccess can achieve the following two properties. (1) No one can know the actual signer of a ring signature, even if all of the private keys of the parties of the ring are known. (2) An actual signer can deny that he has made a ring signature, even if his private key is known. In addition, as described in Section V, the joining of a new user strengthens the maximal privacy of the group members.

Integrity Protection of Query Command: In Priccess, an authorized network user uses a ring signature technique to authenticate the query command Que. The sensor nodes know the public keys of all group members, and thus can verify the message $\{Que, \sigma\}$ as well as Que. Therefore, an adversary cannot modify the query command and then pass the verification of the sensor nodes.

Node Compromise Tolerance: As described in Section IV.C, only the public key of the network owner and the group access list pool are pre-loaded on every node. Therefore, even if an adversary compromises some nodes, the adversary just obtains the public key of the network owner and the group access list pool. Without the private key of a network user, the adversary cannot impersonate any network user by compromising nodes.

Freshness: The use of the timestamp included in the query Que can ensure the freshness of the message $\{Que, \sigma\}$. In addition, nonces can be used to prevent replay attacks instead of timestamp. A feasible mechanism is suggested as follows. A network user, say U_j, sends the request for nonce to the sensor nodes. Then the targeted sensor nodes, say $\{S_1, \ldots, S_i\}$, return random or pseudo-random numbers $\{nonce_1, \ldots, nonce_i\}$ to U_j, respectively. Here $i \geq 1$. Subsequently, U_j adds $\{h(ID_1, nonce_1), \ldots, h(ID_i, nonce_i)\}$ to the query command Que. We assume that every targeted node, say S_i, receives the message $\{Que, \sigma\}$. S_i checks whether $h(ID_i, nonce_i)$ is included in the query command Que before S_i verifies the signature σ. If it is invalid, S_i simply drops the message $\{Que, \sigma\}$. Otherwise, the node verification procedure proceeds to the next step.

Limits of Access Privileges: The network owner can restrict each network user's activities by group division. As described in Section IV.E, to pass the verification of sensor nodes, the *targeted node identities set* and *request* fields included in every query should be set according to *group access privilege mask*.

VII. Implementations and Performance Evaluation

As sensor nodes are usually resource constrained, they may not be able to execute expensive cryptographic operations efficiently and thus become the bottleneck of a security protocol. Thus, we evaluate Priccess by implementing key components on an experimental test-bed, including user query generation, sensor node verification, and establishing secure channels between the user and sensor nodes. For new user joining phase, the communication among the CA, users, and network owner depends on WiFi or wired networks. Additionally, the computing and storage resources of them are rich. As described in Section IV.G, the radio of the network owner can work in a high power level. Note that in this phase, the nodes just need to receive the broadcast message from the owner. For user revocation phase, each entity in the network just needs to directly delete the public key from the pool once it is expired. Even if there is no secure clock synchronization technique employed in each node, each node just needs to receive the broadcast message from the owner. Thus, we believe that the overhead of such two phases is not a problem.

A. Experimental Test-bed and Implementation Setup

We have implemented Priccess on a real world experimental test-bed. Our implementation has the network owner, network user and sensor side programs. The network owner side programs are C programs using OpenSSL [18] running on a 3.2 GHz desktop PC. In addition, the network user side programs are C programs using OpenSSL running on a 1.8 GHz laptop PC. The sensor side program is written in nesC and runs on Imote2 motes. The Imote2 has the Intel PXA271 XScale 32-bit processor running at 13 to 416 MHz. Our Imote2 motes run TinyOS [19]. We use SHA-1 as the one-way hash function and AES as the data encryption algorithm. In addition, the key size of ECC is set to 160 and 192 bits, respectively. Note that 160-bit ECC key length is considered secure enough for now and immediate future. In our experiment, we assume that m members in the group have been chosen to generate the ring signature.

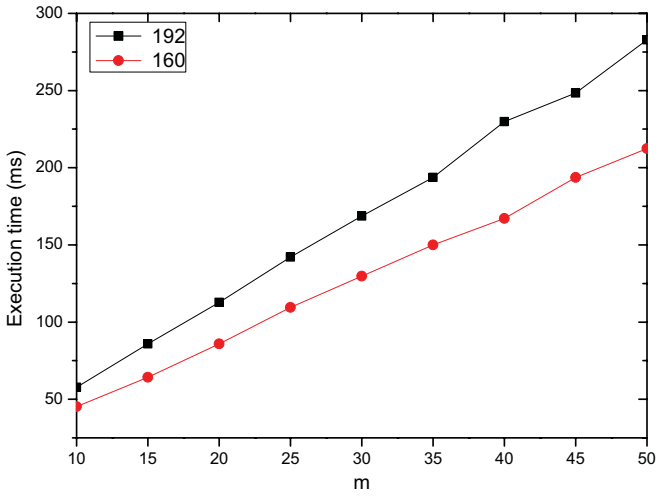

Fig. 4. The execution time for user query generation.

B. Evaluation Results

In this subsection, we present evaluation results of Priccess. We use the following four metrics: message overhead, the execution time, message complexity, and energy consumption. The execution time measures the time duration for each operation of our scheme. The message complexity then shows the amount of messages transmitted during the user query generation phase.

Firstly, we consider the message overhead of Priccess without considering packet headers. There are two cases as follows. One case is that we do not need to establish secure channels between the network user and sensor nodes. The message overhead of Priccess is $(20 \times m + 36)$ bytes. For the second case that we need to establish secure channels between the network user and sensor nodes, The message overhead of Priccess is $(20 \times m + 56)$ bytes.

Fig. 4 shows the execution time for user query generation phase when the number of chosen group members (m) and the key size of ECC (l) vary. For example, the execution time is 45.3 ms in the case that $m = 10$ and $l = 160$. The execution time is 212.4 ms (resp. 282.9 ms) in the case that $m = 50$ and $l = 160$ (resp. $l = 192$). Note that user query generation runs on a 1.8 GHz laptop PC. Considering the clock frequency of a typical PDA is more than 1 GHz, our scheme is efficient for most user access devices (e.g., laptop PCs and PDAs). Note that for the cryptographic operations on desktop PC and laptop PC, we perform the same experiment one thousand times and take an average over them.

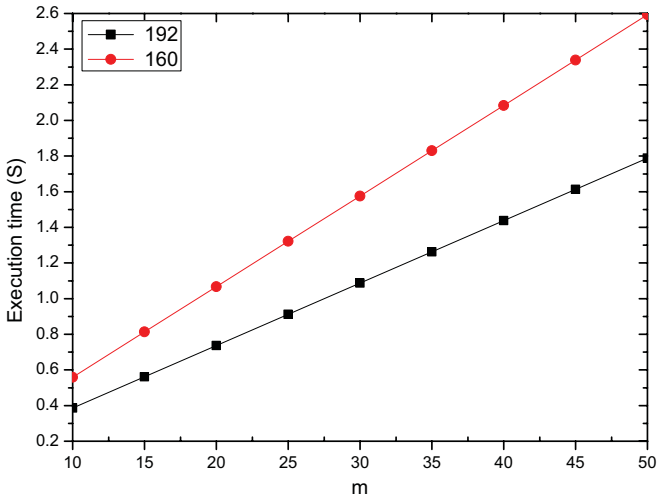

Fig. 5. The execution time for node verification.

TABLE I
THE ENERGY CONSUMPTION FOR PRICCESS

m	10	20	30	40	50
Node verification cost(160)(J)	0.35	0.66	0.98	1.29	1.45
Node verification cost(192)(J)	0.50	0.96	1.42	1.88	2.33

Fig. 5 shows the execution time for node verification phase when the number of chosen group members (m) and the key size of ECC (l) vary. For example, our experiments show that the verification time on Imote2 mote is 0.385 second in the case that $m = 10$ and $l = 160$. The execution time is 1.437 seconds (resp. 2.084 seconds) in the case that $m = 40$ and $l = 160$ (resp. $l = 192$). As described in Section IV.B and IV.E, the verification time on sensor nodes is independent of the scale of a WSN. That is, even when a WSN scales up to thousands of motes, the node verification delay does not increase. It is directly proportional to the number of chosen group members (m).

Here we consider the execution time for establishing secure channels between the network user and the sensor nodes. For ECDH key exchange, Imote2 consumes about 0.034 second to generate such a session key. For digital envelopes, ECC public-key encryption takes about 0.271 second on Imote2. Here the key size of ECC is set to 160 bits. Our implementation shows that it takes 0.984 μs and 0.39 ms for AES to encrypt 64 bytes data on a network user and Imote2, respectively. As described above, the message complexity of Priccess is independent of the size of a WSN (i.e., the total number of sensor nodes). More specifically, in a particular sensor network, the network user only needs to send the message $\{Que, \sigma\}$ and then the targeted sensors give some responses.

We use the formula $E = U * I * t$ to estimate the energy consumption of signature verification, where U is the voltage, I is the current and t is the time duration. Thus, it is linear to the execution time. Imote2 motes are powered by three AA batteries, so U is approximately equal to 4.5 volts. From the Imote2 data sheet, the current is 200mA in active mode. Table I lists the energy consumption of Priccess when l and m vary. For example, node verification on a Imote2 mote consumes roughly 0.663 J energy when $m = 20$ and $l = 160$.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a novel protocol to achieve privacy-preserving access control for WSNs. The security analysis and experimental results show that our approach is feasible for real applications. To the best of our knowledge, until now this is the first secure privacy-preserving access control scheme for WSNs. Our experiment shows that the system overhead of the proposed protocol is reasonable in practical scenarios.

To achieve privacy-preserving access control, the network owner in Priccess cannot determine the identity of the actual signer of a query command. Thus, without the assumption that users are rational, a dishonest user may launch DoS attacks without exposing his identity. For example, a dishonest network user can exploit the benefits of distributed privacy-preserving access control and keeps sending query commands to sensor nodes for preventing its competing users from accessing sensor data. One of our future works is to investigate how to defend such an attack by using some non-cryptographic techniques such as rate-limiting mechanisms. In general, the network owner should have measures to identify dishonest users and defend against their attacks. To achieve this, we can also rely on reports from sensor nodes. For example, if a dishonest user launches some unknown attacks on sensor nodes, the network owner can identify which particular user group (the one including the dishonest users) launched the attack by analyzing the access records periodically submitted by each node. Then, the network owner can defeat users' misbehavior through some ways (e.g., revoking the user group).

REFERENCES

[1] H. Wang, B. Sheng, and Q. Li, "Elliptic curve cryptography based access control in sensor networks," *Int'l J. Security and Networks*, vol. 1, no. 3-4, pp. 127–137, 2006.
[2] H. Wang and Q. Li, "Distributed user access control in sensor networks," in *Proc. IEEE/ACM DCOSS*, pp. 305–320, 2006.
[3] M. Das, "Two-factor access control in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1086–1090, 2009.
[4] D. He, Y. Gao, S. Chan, C. Chen, and J. Bu, "An enhanced two-factor access control scheme in wireless sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 10, no. 4, pp. 361–371, 2010.
[5] H. Song, S. Zhu, W. Zhang, and G. Cao, "Least privilege and privilege deprivation: towards tolerating mobile sink compromises in wireless sensor networks," *ACM Trans. Sensor Networks*, vol. 4, no. 4, pp. 1–34, 2008.
[6] R. Zhang, Y. Zhang, and K. Ren, "DP^{2}AC: distributed privacy-preserving access control in sensor networks," in *Proc. IEEE INFOCOM*, 2009.
[7] D. He, J. Bu, S. Zhu, M. Yin, Y. Gao, H. Wang, S. Chan, and C. Chen, "Distributed privacy-preserving access control in a single-owner multi-user sensor network," in *Proc. IEEE INFOCOM Mini-Conference*, 2011.
[8] Taking the Pulse of the Planet: EPA's Remote Sensing Information Gateway. http://www.epa.gov/geoss/.
[9] NOPP, http://www.nopp.org/.
[10] B. Carbunar, Y. Yu, L. Shi, M. Pearce, and V. Vasudevan, "Query privacy in wireless sensor networks," in *Proc. IEEE SECON*, pp. 203–212, 2007.
[11] M. Li, W. Lou, and K. Ren, "Data security and privacy in wireless body area networks," *IEEE Wireless Commun.*, vol. 17, no. 1, pp. 51–58, 2010.
[12] K.-F. Ssu, C.-H. Chou, H. Jiau, and W.-T. Hu, "Detection and diagnosis of data inconsistency failures in wireless sensor networks," *Computer Networks*, vol. 50, no. 9, pp. 1247–1260, 2006.
[13] D. Janakiram, V. A. Reddy, and A. V. U. P. Kumar, "Outlier detection in wireless sensor networks using Bayesian belief networks," *Commun. Syst. Software and Middleware (Comsware)*, pp. 1–6, 2006.
[14] A. Liu and P. Ning, "TinyECC: a configurable library for elliptic curve cryptography in wireless sensor networks," in *Proc. ACM/IEEE IPSN*, 2008.

[15] X. Lin, R. Lu, H. Zhu, P.-H. Ho, X. Shen, and Z. Cao, "ASRPAKE: an anonymous secure routing protocol with authenticated key exchange for wireless ad hoc networks," in *Proc. IEEE ICC*, 2007.
[16] D. He, L. Cui, H. Huang, and M. Ma, "Design and verification of enhanced secure localization scheme in wireless sensor networks," *IEEE Trans. Parallel and Distrib. Syst.*, vol. 20, no, 7, pp. 1050–1058, 2009.
[17] K. Sun, P. Ning, and C. Wang, "TinySeRSync: secure and resilient time synchronization in wireless sensor networks," in *Proc. ACM CCS*, pp. 264–277, 2006.
[18] OpenSSL, http://www.openssl.org.
[19] Tiny OS. http://www.tinyos.net.

Daojing He received his B.Eng. and M.Eng. degrees in Computer Science from Harbin Institute of Technology in 2007 and 2009, respectively. He is currently a Ph.D. student in Zhejiang University, P.R. China. His research interests include network and systems security with focuses on wireless security. He serves as TPC for IEEE Globecom 2011, IEEE PIMRC 2011, etc.

Jiajun Bu received the BS and PhD degrees in computer science from Zhejiang University, China, in 1995 and 2000, respectively. He is currently a professor in the College of Computer Science and the deputy dean of the Department of Digital Media and Network Technology at Zhejiang University. His research interests include embedded system, mobile multimedia, and data mining.

Sencun Zhu is an associate professor at Department of Computer Science and Engineering and College of Information Sciences and Technology, the Pennsylvania State University. He received the PhD degree in Information Technology from George Mason University in 2004. Prior to that, he received the M.S. degree in Signal Processing from University of Science and Technology of China in 1999 and the B.S. degree in Precision Instruments from Tsinghua University in 1996. His research interests include network and systems security with focuses on wireless security, online social network security and privacy, and software security and protection. His publications can be found from http://www.cse.psu.edu/ szhu

Sammy Chan 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.

Chun Chen received the bachelor degree in mathematics from Xiamen University, China, in 1981, and the masters and PhD degrees in computer science from Zhejiang University, China, in 1984 and 1990, respectively. He is a professor in the College of Computer Science, and the director of the Institute of Computer Software at Zhejiang University. His research activity is in image processing, computer vision, and embedded system.