An RSA-Like Scheme for Multiuser Broadcast Authentication in Wireless Sensor Networks

Chen-Yang Cheng,1 Iuon-Chang Lin,2,3 and Shu-Yan Huang2

1Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei 106, Taiwan
2Department of Management Information Systems, National Chung Hsing University, Taichung 402, Taiwan
3Department of Photonics and Communication Engineering, Asia University, Taichung 413, Taiwan

Correspondence should be addressed to Iuon-Chang Lin; iclin@nchu.edu.tw

Received 13 March 2015; Revised 3 August 2015; Accepted 24 August 2015

Academic Editor: Alessandro Nordio

Copyright © 2015 Chen-Yang Cheng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For network users roaming in a wireless sensor network (WSN), they can broadcast queries to WSNs to obtain the latest sensed data from sensor nodes using their mobile devices. In such a scenario, each sensor node has to verify the validity of every query sent from users. In this paper, RSA-like public key cryptography is employed to design a mechanism for multiuser broadcast authentication in WSNs. Within the proposed scheme, the use of certificates becomes optional. When users broadcast queries to WSNs, each sensor node can verify every query immediately without buffering any one. As a result, the energy cost for verifying a query by a sensor node is very small. Furthermore, our scheme provides enough scalability and security. The quantitative analyses show that our scheme is efficient in terms of storage and computational overheads.

1. Introduction

Wireless sensor networks (WSNs) are widely used in various applications, such as building automation, mobile object tracking, and habitat monitoring [1, 2]. A WSN usually consists of one or more base stations and a large number of sensor nodes. Because the sensor nodes are resource-constrained and usually deployed in hostile environments, they are vulnerable to malicious attacks. Thus, broadcast authentication becomes a critical issue in WSNs, which can prevent adversaries from injecting bogus queries. Traditional schemes [3–8] applied the key pools, space pools of matrix or polynomials to establish the pairwise key between any two neighboring sensor nodes. Although these schemes can establish a secure communication channel, they cannot withstand injecting bogus queries.

Thus, three scenarios are considered in this paper. The first scenario is that users broadcast messages to WSNs using their own mobile device dynamically, and then each sensor node will verify these messages. The second scenario is that once malicious users are revoked by the base station, the action of rekeying will be triggered by the base station. The third scenario is that when new users join WSNs, the base station will take some appropriate actions for these users.

In this paper, an RSA-like scheme is employed to construct a multiuser broadcast authentication mechanism. Although it is usually thought that RSA is expensive for sensor nodes in terms of computational overhead, however, the authors in the work in [9] showed that the cost of the RSA signature verification is not really expensive. Because each sensor node only needs to verify the RSA signature, the computational cost for a sensor node to verify a message is acceptable. As reported in [10], public key cryptography is viable on an Atmel ATmega128 at 8 MHz for resource-constrained sensor nodes. The authors tested and implemented elliptic curve point multiplication and RSA operations on two 8-bit microcontrollers with assembly languages. Elliptic Curve Cryptography (ECC) is more computationally efficient than RSA, but RSA can still be implemented for sensor nodes, such as Crossbow MICA Motes. For example, in the implementation of [10], it requires 0.81 s for 160-bit ECC point multiplication and 0.43 s and 10.99 s for RSA-1024 public key operation and private key operation, respectively. Furthermore, the Chinese Remainder Theorem (CRT) can
accelerate RSA private key operations, namely, decryption and signature generation.

Broadcast authentication enables each sensor node to verify the received messages which are originated with the authorized source and were not modified. In our scheme, when a user broadcasts a query message to WSNs, each sensor node only has to verify the signature attached in the message using the public key computed by the base station. At the beginning of network initialization, the base station generates RSA public key and private key for each sensor node and each user, respectively. Once a user is revoked, the base station has to regenerate a new RSA public key for each sensor node. At the same time, other users do not have to change their own private key. So sensor nodes do not have to store a revocation list in their own memory. The main contributions of this paper are described as follows:

1. We propose an RSA-like scheme to secure the multiuser broadcast. Our scheme provides enough security with 1024-bit RSA and great scalability.

2. In our scheme, each sensor node does not buffer any message, and it can verify every message immediately. Thus, the impact of DoS attacks can be mitigated.

3. Once illegal network users are revoked by the base station, all the current users do not have to obtain new authentication information from the base station. On the other hand, there is no need for each current user to reobtain his/her private key from the base station after he/she has participated in WSN.

4. Each sensor node only has to store one network public key, which is 1024 bits. No matter how many network users are there in WSNs, each sensor node can verify these messages. Therefore, our scheme is more efficient in terms of storage overhead as compared with the previous schemes.

5. A quantitative energy consumption analysis on computational cost for verifying a message shows that our scheme indeed outperforms the previous schemes.

The rest of this paper is organized as follows. In Section 2, the related work will be introduced. In Section 3, the network and adversary models used in this paper are presented. Subsequently, RSA cryptosystem and the concept of RSA master-key will be described in Section 3. In Section 4, the proposed multiuser broadcast authentication scheme is presented. Section 5 is the discussions of our proposed scheme. Section 6 is the performance evaluation in terms of communication, storage, and computational overheads. The conclusion is made in Section 6.

2. Related Work

In order to prevent adversaries from injecting bogus queries, the authors in the work in [11] first proposed a scheme, called µTESLA, to overcome this problem. They employed a one-way hash function to generate a key chain for the authentication of broadcast messages. However, a source requires maintaining a long chain of keys for the long-term uses. In addition, µTESLA suffers from serious DoS attacks. Each sensor node has to buffer all received messages within a time interval, and then it can verify these messages by using the delayed disclosure key broadcasted by the base station at the next time interval. The base station and sensor nodes are assumed to be loosely time synchronized.

Furthermore, the authors in the work in [12] proposed a novel protocol called BABRA to address the problem of broadcast authentication in WSNs. Unlike µTESLA, BABRA can support broadcast for infinite rounds. At the same time, it eliminates the requirement of key chain. Nevertheless, BABRA also suffers from serious DoS attacks, since each sensor node has to buffer all messages before the corresponding key is disclosed.

In [13], the authors proposed a broadcast source authentication mechanism based on multiple MACs (Message Authentication Codes). The scheme requires sensor nodes to have different overlapping set of keys. When the source wants to broadcast a query, it uses its keys to compute multiple MACs and appends them to the message. Then, the recipient can verify the message based on the MACs by using the common keys shared with the source. In comparison with the above schemes, each recipient could verify a message immediately. Therefore, the impact of DoS attacks can be mitigated. However, the key predistribution under a hierarchical structure results in scalability issues. The authors in the work in [14, 15] proposed broadcast authentication schemes using one-time signature. As compared with the above schemes, each sensor node can verify a query immediately without buffering others. However, the number of signatures is limited when a lot of queries are signed by the source.

The authors in the work in [16] first proposed a protocol for multiuser broadcast authentication, in which any unauthorized user cannot broadcast queries to a WSN arbitrarily. Each authorized user may be equipped with a powerful mobile device, and then he/she can broadcast queries to WSNs for the purpose of obtaining the latest sensed data from sensor nodes in WSNs. Whenever a WSN processes a query, sensor nodes are able to verify the query. However, the user's public key certificate incurs additional communication and computational overheads.

In [17–19], the main idea of these schemes is to preload each sensor node/network user with some secret information. After that, sensor nodes can compute session keys shared between them and users. Hence, the authenticity of users can be verified through these session keys. All the above schemes are based on challenge-response protocols. Although the above schemes have been proposed for user authentication, most of them do not provide adequate efficiency. By contrast, some schemes in [20–22] focus on the mechanism, in which each sensor node can verify every query directly without challenging any nonce. These schemes provide adequate efficiency for multiuser broadcast authentication. However, it is still difficult to deal with the resource-constrained problem and sensor nodes compromise attack. An efficient scheme is proposed to address the problems without incurring much overhead.

The authors in the work in [16] proposed the first solution to the problem called authenticated querying. They utilized
Elliptic Curve Cryptography (ECC) to construct the user authentication scheme, which only considered the situation that a user’s query involves a single sensor node. Besides, this scheme incurs additional communication overhead because the user’s certificate needs to be transmitted. Furthermore, each sensor node has to verify the user’s certificate and signature. Obviously, it also incurs additional computational overhead. A fully symmetric key based solution was proposed for authenticated querying [17]. The authors used a bivariate polynomial to establish shared keys between the user and the sensor nodes that should process the user’s query. Then, these sensor nodes can verify the authenticity of the user by using the shared keys between them and the user. The scheme is effectively tolerant of the sensor node compromise attack, but it still incurs additional communication overhead because the collection of MACs needs to be transmitted. In particular, when there are a large number of sensor nodes that should process the user’s query, the collection of MACs will be big.

The authors in the work in [19] proposed a distributed user access control scheme, which includes local authentication and remote authentication. Unfortunately, this scheme incurs significant communication overhead, especially when the user’s access control list is heavy. The reason is that the access control list needs to be transmitted. In [18], the authors proposed a user authentication scheme with the self-certified key (SCK) cryptosystem. The main idea is to establish pairwise keys between the user and his/her local sensor nodes. Then, these sensor nodes can verify the authenticity of the user. Because each sensor node is preloaded with a public/private key pair, the scheme suffers from serious sensor node compromise attack. An adversary may utilize the keying material of a compromised sensor node to impersonate a legal user to destroy the WSNs.

In [22], the authors initially proposed two basic schemes called CAS and DAS. In CAS, each user is equipped with a public/private key pair and his/her public key certificate signed by the base station, and then he/she signs every broadcast message with his/her private key. Upon receiving the message, each sensor node can verify the public key certificate of the user by using the public key of the base station. Finally, each sensor node can verify the message the user broadcasts in the WSNs. However, the certificate has to be transmitted and verified by each sensor node. CAS introduces additional communication and computational overheads. In DAS, each sensor node has to store all the users’ ID information and their corresponding public keys. However, the storage of DAS is neither efficient nor scalable. This scheme is not suitable for storage-constrained sensor nodes when there are a large number of users.

Subsequently, the authors proposed two advanced schemes called BAS and HAS. In BAS, each sensor node is required to store a Bloom filter and \( k \) hash functions. Upon receiving a message, each sensor node can check whether the user’s ID and his/her corresponding public key are authentic by using the Bloom filter and \( k \) hash functions. However, the probability of a false positive \( f_{\text{req}} \) may happen even though it is very small. It may suggest that an illegal user is authentic. Besides, BAS supports up to 1,000 users when \( f_{\text{req}} = 2.03 \times 10^{-17} \) for a Bloom filter of 9.8 KB. It incurs a large amount of storage overhead, and the maximum supported number of users is limited by the storage limit and \( f_{\text{req}} \). Therefore, an improved scheme called HAS was proposed to support more users. To achieve this goal, HAS utilizes the Bloom filter and Merkle hash tree. Nevertheless, it still incurs a large amount of storage overhead and lacks scalability.

The authors in the work in [21] proposed three broadcast authentication schemes. The first scheme is CAS as mentioned before, and the second scheme is based on Merkle hash tree. The base station first constructs a Merkle hash tree, in which each leaf node contains the hash value of a user’s ID and his/her corresponding public key. Then, each sensor node has to store the value of the final root node of the hash tree. At the same time, each user has to obtain his/her auxiliary authentication information (AAI). When a user broadcasts a message to the WSNs, he/she signs the message and appends his/her AAI to the message. Upon receiving this message, each sensor node can verify the user’s public key by using AAI. If the final hash value is equal to the value of the final root node, each sensor node can verify the user’s public key. The Merkle hash tree based scheme does not require the users’ public key certificate to be transmitted. In addition, this scheme can be improved by increasing the number of stored hash values in each sensor node. Thus, the size of AAI can be reduced. However, once a user is revoked by the base station, each current user has to obtain his/her updated AAI from the base station. It is impractical for the current users.

The third scheme is an ID-based authentication scheme. The concept of ID-based cryptography originated from [23]. A user’s ID is just like the user’s public key. In this scheme, the user’s public key is \( U_{\text{ID}} \parallel v_{i} \), where \( U_{\text{ID}} \) is the user’s ID and \( v_{i} \) is the current time interval. Each sensor node can verify the message broadcasted by the user using \( U_{\text{ID}} \parallel v_{i} \) and the network public key. However, this scheme requires each sensor node to perform two expensive pairing operations. In addition, each user has to obtain a new private key from the base station at the beginning of each time interval. Once some users are revoked, each sensor node needs to store a revocation list only within the current time interval.

An ID-based signature scheme called BNN-IBS [24] is based on Schnorr signature [25], and the authors in the work in [20] proposed a variant of BNN-IBS called vBNN-IBS with a smaller signature size. The proposed scheme called IMBAS [20] is also used to secure the multiuser broadcast. When a user wants to broadcast a message to WSNs, he/she signs the message using vBNN-IBS signature. The base station can also broadcast a message to WSNs with a smaller message size. Upon receiving a message broadcasted by the user (or the base station), each sensor node can verify it immediately. Furthermore, each sensor node also has to store the IDs of the revoked users as a revocation list infinitely when the number of revoked users increases unceasingly. As a result, it can be found that the public key cryptography is easier to be used for multiuser broadcast authentication than the symmetric key cryptography.

In [20–22], the authors used Elliptic Curve Cryptography (ECC) based schemes to secure multiuser broadcast authentication. \( E/F_{p} \) used in [20–22] denotes the elliptic curve over
the finite field $F_p$, where $p$ is a large odd prime. The elliptic curve $E/F_p$ is defined by the equation $y^2 = x^3 + ax + b$, where $a, b \in F_p$, $G_1$ is a $q$-order subgroup of the additive group of points of $E/F_p$ [21].

### 3. Preliminaries

First, the concepts of the proposed network model and adversary model are introduced. Then, the review of RSA public key cryptosystem and the concept of RSA master-key are presented in the section.

#### 3.1. The Network Model

In this model, the sensor network consists of a base station and a large number of sensor nodes. The base station is assumed to be powerful, while sensor nodes are resource-constrained. Furthermore, there are a large number of network users. These users who roam in WSNs can use their mobile devices to broadcast queries to WSNs for the purpose of obtaining the latest sensed data. The mobile devices of the users are more powerful than resource-constrained sensor nodes in terms of computation, communication, storage, and energy abilities. The number of network users may be dynamic. In this paper, the WSNs time is assumed to be loosely synchronized.

#### 3.2. The Adversary Model

We assume that the base station is always trustworthy, but sensor nodes may be compromised by an adversary. Therefore, there may be some malicious sensor nodes in the WSNs. The adversary is able to compromise or capture not only sensor nodes but also users’ mobile devices, and then all the secret information (e.g., keying material or secret data) held by them is known by the adversary. In addition, the adversary may impersonate these captured users to broadcast bogus messages to WSNs. So these users of WSNs have to be revoked by the base station to prevent them from destroying WSNs. Furthermore, the adversary can flood bogus messages into WSNs to exhaust the precious energy of sensor nodes. Note that the adversary can also eavesdrop and resend the messages.

#### 3.3. RSA Public Key Cryptosystem

We give an introduction to RSA public key cryptosystem. In RSA cryptosystem, each participant holds a public/private key pair, which is generated by a certificate authority (CA). The steps of generating the public/private key pair are described as follows:

1. Two large primes $p$ and $q$ are randomly chosen, and then $N = pq$ can be computed.
2. To choose a parameter $e$, (1) have to be satisfied:
   \[
   \text{GCD}(\phi(N), e) = 1,
   \]
   \[
   1 < e < \phi(N),
   \]
   where $\phi(N) = (p - 1)(q - 1)$ is Euler’s totient function of $N$. Consequently, a parameter $d$ can be computed through Extended Euclidean algorithm, such that
   \[
   e \times d \equiv 1 \mod \phi(N).
   \]
3. Finally, $(N, e)$ is the public key and $d$ is the private key.

Now, assume that $A$ wants to send a message $M$ to $B$. If $A$ wants to prove the confidentiality of $M$, he/she can use $B$’s public key $(e_B, N_B)$ to encrypt the message $M$. Then, $A$ sends the encrypted message $C$ to $B$, where $C = M^{e_B} \mod N_B$.

Upon receiving the encrypted message, $B$ can use his/her private key $d_B$ to recover the message $M$ by computing
\[
M = C^{d_B} \mod N_B.
\]

On the other hand, if $A$ wants to prove the origin and integrity of $M$, he/she can use his/her private key $d_A$ to sign the message $M$. Then, $A$ sends the message $M$ with the signature $S$ to $B$, where $S = M^{d_A} \mod N_A$. Upon receiving the message $M$ and the signature $S$, $B$ can use $A$’s public key $(e_A, N_A)$ to verify the signature by computing
\[
M' = S^{e_A} \mod N_A.
\]
And then $B$ checks whether $M = M'$ or not. The signature is accepted if so and rejected otherwise.

#### 3.4. RSA Master-Key

The authors in the work in [26] proposed an RSA master-key scheme, which is built on RSA cryptosystem. Suppose that there are $n$ entities $f_1, f_2, \ldots, f_n$:

1. CA randomly chooses large primes $p_1, q_1, p_2, q_2, \ldots, p_n, q_n$ and a positive integer $e_H$ which is relatively prime to $L$, where $L = \text{LCM}(L_1, L_2, \ldots, L_n)$ and $L_i = \phi(p_i \times q_i)$ for $i = 1, 2, \ldots, n$.
2. Let $m_i = p_i \times q_i$ for $i = 1, 2, \ldots, n$. CA computes the corresponding $d_{Hi}$ such that $e_{Hi} \times d_{Hi} \equiv 1 \mod L_i$. Then, let $e_i \equiv e_{Hi} \mod L_i$ and $d_i \equiv d_{Hi} \mod L_i$ for each $i$.
3. For $i = 1, 2, \ldots, n$, let $(m_i, e_i, d_i)$ be defined as the $i$th entity’s RSA system with elementary encryption key $e_i$ and decryption key $d_i$. The sequence $(e_{Hi}, d_{Hi}, p_1, q_1, p_2, q_2, \ldots, p_n, q_n)$ is called the master-key of the system. This master-key is only known by CA.

Therefore, each $f_i$ can be encrypted into $\bar{f}_i$, where $\bar{f}_i \equiv f_i^{e_i} \mod m_i$. It can also be $f_i \equiv \bar{f}_i^{d_i} \mod m_i$. In fact, we have $f_i \equiv \bar{f}_i^{d_i} \mod m_i$ for all $i$ due to the equation $e_{Hi} \times d_{Hi} \equiv 1 \mod L_i$.

### 4. The Proposed Multiuser Broadcast Authentication Scheme

In this section, a scheme for multiuser broadcast authentication, which is based on RSA cryptosystem and the master-key scheme [26], is presented. The detailed steps of our scheme are described in the following sections.

#### 4.1. Our Scheme

We assume that there are $m$ network users in WSNs, and the base station is the highest authority. The
task of the base station is to generate a private key for each user and assign a public key for each sensor node. We use the RSA-like scheme to secure the multiuser broadcast. When the event which a user joins or leaves happens, our scheme can cope with the situation. Furthermore, once a user is compromised, the base station will take an appropriate action to cope with such a situation. The steps of our scheme are described as follows.

4.1.1. The Setup Phase. First, the base station randomly chooses \( 2m + 2 \) distinct large primes \( p_i, q_i \), \( 0 \leq i \leq m \), and it also chooses two distinct large primes \( p_{bs}, q_{bs} \) for itself. Second, the base station computes \( N_0 = p_0 \times q_0 \) for each sensor node, \( N_{bs} = p_{bs} \times q_{bs} \) for itself, and \( N_1 = p_1 \times q_1, N_2 = p_2 \times q_2, \ldots, N_m = p_m \times q_m \) for users \( User_1, User_2, \ldots, User_m \). Note that \( 2m + 4 \) distinct large primes are only known by the base station, but \( N_1, N_2, \ldots, N_m \) and \( N_{bs} \) are public.

4.1.2. The Key Generation Phase. The base station computes the least common multiple \( L_0 \) of \( m + 2 \) integers \( \phi(N_0), \phi(N_1), \ldots, \phi(N_m), \phi(N_{bs}) \), such that

\[ L_0 = \text{LCM} \{ \phi(N_0), \phi(N_1), \ldots, \phi(N_m), \phi(N_{bs}) \}, \quad (5) \]

where \( \phi(N_i) \) is Euler’s totient function of \( N_i \) and \( \phi(N_i) = (p_i - 1) \times (q_i - 1) \). The base station chooses a parameter \( d_0 \) which is relatively prime to \( L_0 \) \( (1 < d_0 < L_0) \). Then, it can compute \( e_0 \) through Extended Euclidean algorithm, such that

\[ d_0 \times e_0 \equiv 1 \mod L_0, \quad (6) \]

Note that the bit-length of \( e_0 \) can be first chosen as short as possible because each sensor node uses \( e_0 \) to verify signatures. The notations \( d_0 \) and \( e_0 \) used here are opposite to the common notations of RSA. The reason is that we use \( d_0 \) to sign messages and \( e_0 \) to verify signatures. After generating \( d_0 \), the base station uses it to generate the private keys \( d_i, d_2, \ldots, d_m \) for users \( User_1, User_2, \ldots, User_m \) and \( d_{bs} \) for itself by computing

\[ d_i = d_0 \mod \phi(N_i), \quad d_{bs} = d_0 \mod \phi(N_{bs}), \quad (7) \]

where \( i = 1, 2, \ldots, m \).

4.1.3. The Key Assignment Phase. After finishing the above key generation, the base station can preload/broadcast each sensor node with \( e_0 \) prior to the WSNs deployment or during WSNs operation time. This method is similar to the method used in [21]. For users \( User_1, User_2, \ldots, User_m \), the base station delivers \( (d_i, N_i), (d_2, N_2), \ldots, (d_m, N_m) \) to them through a secure channel. The base station keeps its private key \( d_{bs} \). Hence, each user has two parameters \( (d_i, N_i) \), where \( d_i \) must be kept secret and public \( N_i \) can be used to represent his/her ID. The two parameters \( (d_{bs}, N_{bs}) \) have the same meaning as above for the base station.

4.1.4. Multiuser Broadcast Authentication. Assume that a user \( N_i \) wants to broadcast a message \( M \) to the WSNs; he/she uses his/her private key \( d_i \) to sign \( M \) and broadcasts the following message:

\[ \{M, TS, N_i, \text{Sign} \}, \quad (8) \]

where \( TS \) is the timestamp and \( \text{Sign} = h(M, TS, N_i)^{d_i} \mod N_i \). \( h(*) \) is a one-way hash function which maps arbitrary inputs to fixed length outputs. After receiving the message, a sensor node first checks whether \( TS \) is fresh. If so, the sensor node will verify the signature \( \text{Sign} \) by computing

\[ h(M, TS, N_i) = \text{Sign}^{e_i} \mod N_i, \quad (9) \]

where \( e_i \) is the public key generated by the base station. If the above equation holds, the signature will be accepted. Otherwise, it will be rejected. For the sake of simplicity, we refer to \( h(M, TS, N_i) \) as \( P \). Because \( L_0 \) is a multiple of \( \phi(N_i) \), we can show that

\[ e_0 \times d_i \mod L_0 = e_0 \times (d_0 \mod \phi(N_i)) \mod L_0 = e_0 \times d_0 \mod \phi(N_i) = 1. \quad (10) \]

Therefore, \( e_0 \times d_i = 1 + t \times \phi(N_i) \), where \( t \in N \). According to Euler’s Theorem, \( P^{\phi(N_i)} \mod N_i = 1 \), we can show that the above equation is correct as the following equation:

\[ \text{Sign}^{e_i} \mod N_i = (P^{d_i} \mod N_i)^{e_0} \mod N_i = P^{\phi(N_i) \times e_0} \mod N_i = P \times P^{t \times \phi(N_i)} \mod N_i = P. \quad (11) \]

The structure can be described according to Figure 1. The relation between users and sensor nodes belongs to a two-layer structure. Every sensor node can verify the signatures signed by users. Besides, the messages broadcasted by the base station can still be verified by every sensor node using the same manner, so we do not describe it again.

4.2. User Revocation. In our scheme, once a user is revoked, the base station has to regenerate a corresponding public key \( e_i \) for sensor nodes. Suppose that a user \( User_i \) is compromised, and then the base station takes the steps as follows.

4.2.1. The LCM Regeneration. The base station computes the least common multiple \( L_0' \) of \( m + 1 \) integers \( \phi(N_0), \phi(N_1), \ldots, \phi(N_m), \phi(N_{bs}) \), which do not include \( \phi(N_i) \), such that

\[ L_0' = \text{LCM} (\phi(N_0), \phi(N_1), \ldots, \phi(N_m), \phi(N_{bs})). \]

4.2.2. The Public Key Regeneration. To update \( e_0 \) with \( e_0' \), the base station uses the equation \( e_0' \times e_0 \equiv 1 \mod L_0' \) to regenerate \( e_0' \). Note that \( d_0 \) is unchanged. The reason is that \( d_0 \) is relatively prime to \( L_0' \). Once \( L_0 \) is changed into \( L_0' \), \( d_0 \) is still relatively prime to \( L_0' \).
4.2.3. The Public Key Broadcast. After regenerating $e'_0$, the base station has to broadcast $e'_0$ to WSNs. First, $e'_0$ is signed by the base station with the base station’s private key $d_{bs}$ for the purpose of proving its authenticity. Second, the base station broadcasts $e'_0$ with its signature, TS, and $N_{bs}$ to WSNs. Third, upon the receipt of this message, a sensor node can use the public key $e_0$ to verify the authenticity of $e'_0$. Finally, the sensor node updates $e_0$ with $e'_0$, if the signature verification succeeds. Note that $d_0$ is unchanging, so the values $d_1, d_2, \ldots, d_m$ of legal users are still unchanging, too. Suppose that an adversary forges the message broadcasted by the base station; this attack will be detected by sensor nodes because the private key $d_{bs}$ is only known by the base station.

4.3. User Join. When a new user wants to join WSN, the steps are similar to the process of user revocation. Once a new user $User_{ms+1}$ wants to join WSN, the base station chooses two distinct large primes $p_{ms+1}$ and $q_{ms+1}$ and computes $N_{ms+1} = p_{ms+1} \times q_{ms+1}$. Then, $L_0$ can be recomputed by (5), and $d_{ms+1} = d_0 \mod o(N_{ms+1})$. Finally, the base station delivers $(d_{ms+1}, N_{ms+1})$ to $User_{ms+1}$ through a secure channel.

5. Discussion

In this section, the scheme in terms of security, DoS attacks, and scalability is analyzed.

(1) The Problem of Factoring $N_i$. The security of our scheme is based on the difficulty of factoring $N_i$, into $p_i$ and $q_i$. Suppose an adversary knows $N_1, N_2, \ldots, N_m$, but he/she cannot compute $L_0$ such that $L_0 = L \cdot \text{LCM}(o(N_0), o(N_1), \ldots, o(N_m))$. The adversary cannot obtain $d_0$ from the equation $d_0 \equiv 1 \mod L_0$ and use $d_0$ to compute $d_i$ from the equation $d_i = d_0 \mod o(N_i)$. Hence, we believe that it is extremely difficult to obtain the private key $d_i$ of user $N_i$.

(2) The Problem of Preventing the Unauthorized User. When an adversary impersonates a legal user $N_i$ to sign a message $M$ with the key $d_i$, sensor nodes will fail to verify the signature attached in the message. We show the result as the following equation:

$$\text{Sign}^{e_0} \mod N_i = \left( P^{d_i} \mod N_i \right)^{e_0} \mod N_i$$

$$= P^{d_i \cdot e_0} \mod N_i$$

$$= P^{d_i \cdot e_0 \mod o(N_i)} \mod N_i \neq P,$$

where $P = h(M, TS, N_i)$. Because $d_{c} \neq d_0 \mod o(N_i)$, the signature verification fails. Therefore, the adversary cannot impersonate any legal user to inject bogus messages into WSNs successfully. Furthermore, the timestamp used in the message can also prevent the replay attack.

(3) Denial of Service (DoS) Attacks. According to [21, 27], μTESLA suffers from serious DoS attacks because each sensor node has to buffer all the messages received within one time interval. This problem can be mitigated by the immediate verification of messages. In our scheme, each sensor node can verify messages sent by legal or illegal users immediately, so our scheme can mitigate the impact of DoS attacks. If an adversary wants to broadcast forged messages to WSNs, this attack will be detected by the sensor nodes. And then they may notify the base station of such a situation.

(4) Scalability. When old sensor nodes exhaust their energy or the sensing region of a WSN has to be enlarged, new sensor nodes have to be deployed. Our scheme can deal with this problem by preloading these sensor nodes with the network public key $e_0$. If a new user wants to join WSNs, the base station will generate two distinct large primes $p_{new}$ and $q_{new}$ for him/her. Subsequently, the new user’s ID $N_{new} = p_{new} \times q_{new}$ is computed by the base station. At the same time, the base station has to regenerate the new network public key $e'_0$. Then, it broadcasts $e'_0$ and its signature to the WSNs. Users can join WSNs without much overhead.

6. Performance Evaluations

In this section, the performance of our scheme in terms of communication, storage, and computational overheads is evaluated. Moreover, the scheme is compared with the previous schemes that were proposed for multiuser broadcast authentication.

6.1. Communication Overhead. In this section, our scheme and the previous schemes in terms of the communication overhead are evaluated. In [20–22], the authors used Elliptic Curve Cryptography (ECC) based schemes to secure multiuser broadcast authentication. To provide the same level of security strength as 1024-bit RSA, ECC requires $p$ of 160 bits, if $G_q$ is a $q$-order subgroup of the multiplicative group of the finite field $F_p^*$.

Like the assumptions used in [21], $|p|$ and $|q|$ are assumed to be 512 bits and 160 bits, respectively. And the embedded degree of $E/F_p$ is equal to 2. According to [21], suppose that a point is over $E/F_p$, and only one of its $X$ and $Y$ coordinates needs to be transmitted because the other can be
Table 1: Comparisons of communication overhead for the related schemes.

| Schemes   | Message size               |
|-----------|----------------------------|
| CAS [21]  | 148 bytes                  |
| MAS [21]  | 84 + 20 × \log_2 m bytes  |
| IAS [21]  | 108 bytes                  |
| HAS [22]  | (74 + 20 × \log_2 m)/1000 bytes |
| IMBAS [20]| 84 bytes (1)              |
|           | 54 bytes (2)               |
| Our scheme| 278 bytes                  |

which is a popular platform for WSNs. The MICA2DOT mote is equipped with the Atmel ATmega128L 8-bit microcontroller at 4 MHz and the Chipcon CC1000 low-power wireless transceiver [9]. As reported in [9], the energy consumption of MICA2DOT mote in active and power-down modes is 13.8 and 0.0075 mW, respectively. Furthermore, the MICA2DOT mote consumes 59.2 and 28.6 \( \mu J \) to transmit and receive one byte, respectively.

In our scheme, the broadcasting message is \{M, TS, N, Sign\}, and the size is 20 bytes (message M) + 2 bytes (timestamp TS) + 128 bytes (N) + 128 bytes (Sign) = 278 bytes. The payload for each packet is 32 bytes, since we require 9 packets to transmit the broadcasting message. And each packet requires a 9 bytes' header, ensuing 8 bytes' preamble, which consists of source, destination, length, packet ID, CRC, and control bytes [9]. Therefore, it requires transmitting 441 bytes ((32 + 9 + 8) bytes × 9 = 441 bytes). In addition, energy consumption on transmitting and receiving a broadcast message of our scheme is 441 × 59.2 \( \mu J \) = 26.11 and 441 × 28.6 \( \mu J \) = 12.61 mJ, respectively. To estimate the energy consumption of broadcasting a message to the entire network, each sensor node has to retransmit once and receive \( v \) times the same message, where \( v \) is the number of neighbors for a sensor node. Eventually, the total energy consumption of our scheme is \( V \times (26.11 + v \times 12.61) \) mJ, where \( V \) is the total number of sensor nodes. Similarly, the energy consumption of CAS, MAS, IAS, and IMBAS schemes on message broadcast can also be estimated. In order to have a consistent comparison environment, the parameters are chosen as the same as [21]. Figure 3 shows energy consumption of the schemes on message broadcast, where \( v = 20, p \) is the probability of the user broadcast, the probability of the base station broadcast is \( 1 - p \), and there are 1024 users \( (m = 1024) \). Figure 4 shows the energy consumption of the schemes on message broadcast for a sensor node when \( v \) varies. Therefore, we can find that the other schemes except for MAS outperform our scheme, since our scheme has larger message size.
6.2. Storage Overhead. In the comparison, the same settings are used as in Section 6.1. In CAS, each sensor node is required to store the ECC's parameters \((p, q, E/F_p, G_1, h, P, \text{and } P_{\text{pub}})\) if ECDSA is used. \(h\) is a one-way hash function such as SHA-1, \(P\) is a generator of \(G_1\), and \(P_{\text{pub}}\) is the network public key of the base station. Note that \(P_{\text{pub}}\) and \(P\) are two points over \(E/F_p\), among which only one of their \(X\) and \(Y\) coordinates needs to be stored.

In MAS, it requires each sensor node to keep the value \(h_i\) which is the final root node of the hash tree. Furthermore, each sensor node has to store the ECC's parameters if an updated \(h_i\) should be signed by the base station to prove its authenticity.

### Table 2: Comparisons of storage overhead for the six schemes.

| Schemes       | Memory size | Quantitative result |
|---------------|-------------|---------------------|
| CAS [21]      | \(p, q, a, b, P, P_{\text{pub}}\) | \(\approx 120 \text{ bytes} \) |
| MAS [21]      | \(p, q, a, b, h_i, P, P_{\text{pub}}\) | \(\approx 140 \text{ bytes} \) |
| IAS [21]      | \(p, q, a, b, P, P_{\text{pub}}\) | \(\approx 120 \text{ bytes} \) |
| HAS [22]      | A Bloom filter, \(p, q, a, b, P, P_{\text{pub}}\) | \(\approx 10,155 \text{ bytes} \) |
| IMBAS [20]    | \(p, q, a, b, P, P_{\text{pub}}\) | \(\approx 120 \text{ bytes} \) |
| Our scheme    | \(e_0\) | 128 bytes |

In IAS, each sensor node also has to be preloaded with the ECC's parameters \((p, q, E/F_p, G_1, G_2, \hat{e}, H, h, P, \text{and } P_{\text{pub}})\), where \(H\) is the MapToPoint hash function which can map strings to nonzero elements in \(G_1\). When HAS supports up to 1000 users, each sensor node is required to store 9.8 KBytes for a single Bloom filter [22] \((f_{\text{req}} = 2.03 \times 10^{-17})\).

The parameters used by IMBAS [20] are similar to IAS's parameters not including \(G_2, \hat{e},\) and \(H\), so we do not describe them again. For our scheme, only the public key \(e_0\) has to be stored by each sensor node. For the sake of simplicity, we only quantify the parameters \((p, q, a, b, P, \text{and } P_{\text{pub}})\) for the ECC based schemes to compare our scheme with them.

The comparisons of the different schemes are shown in Table 2. Clearly, it can be observed that most of the schemes outperform HAS in terms of storage overhead for each sensor node. In addition, from Figure 5, it can be observed that the storage size of our scheme is much less than HAS. Moreover, the storage size of HAS increases significantly when the number of network users varies from 0 to 1000. Therefore, it can be inferred that the scalability of HAS is not good as compared with the other schemes.

6.3. Storage Overhead on Revocation List. An adversary may capture the devices of legal users, and then he/she can
impersonate these legal users to broadcast bogus messages to WSNs for the purpose of exhausting the precious energy of sensor nodes. So the base station has to revoke these illegal users in WSNs. The previous schemes require each sensor node to store the IDs of revoked users infinitely when the number of revoked users increases unceasingly. For example, CAS, HAS, and IMBAS require each sensor node to store the revoked IDs. Therefore, the three schemes may incur a large amount of storage overhead.

In IAS, each sensor node stores the revoked IDs only within one time interval and removes them at the beginning of the next time interval because each user has to obtain a new private key from the base station at the beginning of each time interval. Instead, MAS and our scheme do not require each sensor node to store the revoked IDs. Therefore, the three schemes may incur a large amount of storage overhead.

In IAS, each sensor node stores the revoked IDs only within one time interval and removes them at the beginning of the next time interval because each user has to obtain a new private key from the base station at the beginning of each time interval. Instead, MAS and our scheme do not require each sensor node to store the revoked IDs. Therefore, the three schemes may incur a large amount of storage overhead.

Figure 6 shows the comparisons of the storage size for the six schemes with respect to the number of revoked users. In addition, once an illegal user is revoked, MAS requires current users to obtain the updated AAs regenerated by the base station. This may be impractical for the current users roaming in WSNs. In IAS, each current user also has to obtain a new private key from the base station at the beginning of each time interval, so it is still impractical for the current users. Table 3 illustrates whether or not current users have to obtain new authentication information from the base station.

Table 3: A table illustrating whether or not current users have to obtain new authentication information from the base station for the six schemes.

|                | CAS [21] | MAS [21] | IAS [21] | HAS [22] | IMBAS [20] | Our scheme |
|----------------|----------|----------|----------|----------|------------|------------|
| Yes or No      | No       | Yes      | Yes      | No       | No         | No         |

In CAS, each sensor node has to perform two ECDSA signature verification procedures and two hash function operations. In MAS, to verify the user’s public key, it requires a chain of hash function operations, and we assume that there are λ hash function operations to be performed. In IAS, it requires one modular exponentiation operation and one hash function operation. In addition, IMBAS takes one ECDSA signature verification if the message is broadcasted by the base station. In our scheme, each sensor node is required to perform one hash function operation and one RSA signature verification.

According to [9], the energy cost to verify an ECDSA signature is 45.09 mJ in a MICA2DOT mote. Thus, we can estimate the time to perform an ECDSA signature verification is $45.09 \text{ mJ}/13.8 \text{ mW} = 3.267 \text{ s}$. We recall that 13.8 mW is the energy consumption of the MICA2DOT mote in active mode. Besides, assume that SHA-1 is used, and its energy consumption is 5.9 $\mu$J/byte [9]. Therefore, the time to perform the SHA-1 hash function is 0.01 s when the input size is 24 bytes. Obviously, the time is very small, so it can be ignored.

For a MICA2 mote at 8 MHz, it takes 0.81 s to perform one point multiplication operation over an elliptic curve [10]. Thus, MICA2DOT mote roughly needs $0.81 \times 8/4 = 1.62$ s to perform such an operation. As reported in [29], the time to perform the Tate pairing on a 32-bit ST22 smartcard microprocessor at 33 MHz is 0.752 s. Thus, for the MICA2DOT mote, it roughly needs $6.204 \text{ s} (0.752 \times 33/4)$ to perform the Tate pairing.

6.4. Computational Overhead. In this section, the computational overhead of each sensor node will be evaluated by us. We focus on the computational overhead of verifying a message broadcasted by a user for each sensor node. Note that we still use the same assumptions as mentioned before for our evaluations. In order to compare our scheme with the other schemes in terms of computational overhead, we define seven notations as follows:

(i) $T_m$ is the time to perform one point multiplication operation over an elliptic curve.

(ii) $T_p$ is the time to perform one pairing operation.

(iii) $T_f$ is the time to perform one signature verification (ECDSA-160).

(iv) $T_h$ is the time to perform one one-way hash function operation.

(v) $T_H$ is the time to perform one MapToPoint hash function operation.

(vi) $T_e$ is the time to perform one modular exponentiation operation in $G_2$.

(vii) $T_{RSA}$ is the time to perform one signature verification (RSA-1024).
Table 4: Comparisons of computational overhead for the six schemes.

| Schemes | Computational overhead | Quantitative result |
|---------|------------------------|---------------------|
| CAS [21] | $2T_E + 2T_h$ | $\approx 6.534$ s |
| MAS [21] | $T_E + (\lambda + 1)T_h$ | $\approx 3.267$ s |
| IAS [21] | $2T_p + T_H + T_e$ | $\approx 13.209$ s |
| HAS [22] | $T_E + \lambda T_h$ | $\approx 3.267$ s |
| IMBAS [20] | $p(3T_m + T_h) + (1 - p)T_E$ | $\approx 4.228$ s ($p = 0.6$) |
| Our scheme | $T_{RSA} + T_h$ | $\approx 0.862$ s |

According to [21], the MapToPoint hash function takes 3.0 ms, a modular exponentiation operation in $G_2$ takes 3.13 ms, and a Tate pairing operation takes 47.40 ms on a Pentium IV 2.26 GHz processor with 256 M RAM. Similarly, for the MICA2DOT mote, the times of performing the MapToPoint hash function and a modular exponentiation operation in $G_2$ are $6.204 \times 3.0/47.40 = 0.392$ and $6.204 \times 3.13/47.40 = 0.409$ s, respectively. In addition, the energy consumption of the MICA2DOT mote to sign and verify a signature of 1024-bit RSA is 304 and 11.9 mJ, respectively [9]. We can also estimate the time to perform RSA signature verification is 11.9 mJ/13.8 mW = 0.862 s. In 160-bit ECDSA, we can also estimate the time to perform RSA signature verification is 11.9 mJ/13.8 mW = 0.862 s. In 160-bit ECDSA, the energy consumption to sign and verify a signature is 22.82 and 45.09 mJ, respectively [9].

The cost of generating an RSA signature is higher than that of verifying an RSA signature, but it does not matter in our scheme. The reason is that signatures are generated by users who are equipped with mobile devices or the powerful base station. In our scheme, each sensor node is required to perform only one RSA signatures verification.

Table 4 shows the computational time of the six schemes for verifying a message. We can observe that the computational time of our scheme is significantly less than that of the remaining schemes. Furthermore, it can also be inferred that IAS introduces a much higher computational overhead as compared to the remaining schemes because it requires each sensor node to perform two pairing operations.

Figure 7 also shows the energy consumption of the schemes for verifying a message when the total number of sensor nodes varies from 0 to 10,000. Obviously, our scheme is better than previous schemes.

7. Conclusions

In this paper, we propose an RSA-like multiuser broadcast authentication scheme in WSNs. Although it is assumed that RSA is unsuitable for resource-constrained sensor nodes, the energy cost to verify an RSA signature is still acceptable for the sensor nodes. The quantitative analysis on computational cost for verifying a message shows that our scheme is more efficient than the previous schemes. Moreover, the main cost of computation always falls in the base station which is trustworthy and powerful. In our scheme, the storage cost for each sensor node takes a network public key only. No matter how many users are compromised in WSNs, the storage cost for each sensor node is still invariable because it only needs to update the network public key. On the other hand, each sensor node is able to verify every message immediately, so our scheme can mitigate the impact of DoS attacks. As a result, although our scheme has bigger message size, it is still adoptable for the WSNs in terms of storage overhead, computational overhead, scalability, and security requirements.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] H.-W. Tsai, C.-P. Chu, and T.-S. Chen, “Mobile object tracking in wireless sensor networks,” Computer Communications, vol. 30, no. 8, pp. 1811–1825, 2007.
[2] W.-L. Yeow, C.-K. Tham, and W.-C. Wong, “Energy efficient multiple target tracking in wireless sensor networks,” IEEE Transactions on Vehicular Technology, vol. 56, no. 2, pp. 918–928, 2007.
[3] H. Chan, A. Perrig, and D. Song, “Random key predistribution schemes for sensor networks,” in Proceedings of the IEEE Symposium on Security And Privacy, pp. 197–213, May 2003.
[4] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, “A pairwise key pre-distribution scheme for wireless sensor networks,” in Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS ’03), pp. 42–51, ACM, Denver, Colo, USA, October 2003.
[5] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, “A pairwise key predistribution scheme for wireless sensor networks,” ACM Transactions on Information and System Security, vol. 8, no. 2, pp. 228–258, 2005.
[6] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” in Proceedings of the 9th ACM
Conference on Computer and Communications Security (CCS '02), pp. 41–47, ACM, Washington, DC, USA, November 2002.

[7] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03), pp. 52–61, ACM, Washington, DC, USA, October 2003.

[8] D. Liu, P. Ning, and L. I. Rongfang, "Establishing pairwise keys in distributed sensor networks," ACM Transactions on Information and System Security, vol. 8, no. 1, pp. 41–77, 2005.

[9] A. S. Wandert, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom '05), pp. 324–328, March 2005.

[10] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs," in Proceedings of the 6th International Workshop on Cryptographic Hardware and Embedded Systems, pp. 119–132, Boston, Mass, USA, August 2004.

[11] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: security protocols for sensor networks," in Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom '01), pp. 189–199, Rome, Italy, July 2001.

[12] Y. Zhou and Y. Fang, "WSN09-1: BABRA: batch-based broadcast authentication in wireless sensor networks," in Proceedings of the 49th Annual IEEE Global Telecommunications Conference (GLOBECOM '06), pp. 1–5, San Francisco, Calif, USA, December 2006.

[13] T. Wu, Y. Cui, B. Kusy et al., "A fast and efficient source authentication solution for broadcasting in wireless sensor networks," in Proceedings of the IFIP International Conference on New Technologies, Mobility and Security (NTMS '07), pp. 53–63, Paris, France, May 2007.

[14] S.-M. Chang, S. Shieh, W. W. Lin, and C.-M. Hsieh, "An efficient broadcast authentication scheme in wireless sensor networks," in Proceedings of the ACM Symposium on Information, Computer and Communications Security (ASIACCS '06), pp. 311–320, Taipei, Taiwan, March 2006.

[15] A. Perrig, "The BiBa one-time signature and broadcast authentication protocol," in Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS '01), pp. 28–37, Philadelphia, Pa, USA, November 2001.

[16] Z. Benenson, N. Gedicke, and O. Raivio, "Realizing robust user authentication in sensor networks," in Proceedings of the 1st Workshop on Real-World Wireless Sensor Networks (REALWSN '05), Stockholm, Sweden, June 2005.

[17] S. Banerjee and D. Mukhopadhyay, "Symmetric key based authenticated querying in wireless sensor networks," in Proceedings of the 1st International Conference on Integrated Internet Ad hoc and Sensor Networks, pp. 213–229, Nice, France, May 2006.

[18] C. Jiang, B. Li, and H. Xu, "An efficient scheme for user authentication in wireless sensor networks," in Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops/Symposia (AINAW '07), vol. 1, pp. 438–442, Ontario, Canada, May 2007.

[19] H. Wang and Q. Li, "Distributed user access control in sensor networks," in Proceedings of the IEEE International Conference Distributed Computing in Sensor Systems, pp. 305–320, Francisco, Calif, USA, June 2006.

[20] X. Cao, W. Kou, L. Dang, and B. Zhao, "IMBAS: identity-based multi-user broadcast authentication in wireless sensor networks," Computer Communications, vol. 31, no. 4, pp. 659–667, 2008.

[21] K. Ren, W. Lou, K. Zeng, and P. J. Moran, "On broadcast authentication in wireless sensor networks," IEEE Transactions on Wireless Communications, vol. 6, no. 11, pp. 4136–4144, 2007.

[22] K. Ren, W. Lou, and Y. Zhang, "Multi-user broadcast authentication in wireless sensor networks," in Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '07), pp. 223–232, San Diego, Calif, USA, June 2007.

[23] A. Shamir, "Identity-based cryptosystems and signature schemes," in Advances in Cryptology: Proceedings of CRYPTO '84, vol. 196 of Lecture Notes in Computer Science, pp. 47–53, Springer, Berlin, Germany, 1985.

[24] M. Bellare, C. Namprempre, and G. Neven, "Security proofs for identity-based identification and signature schemes," in Proceedings of the International Conference on Advances in Cryptology (EUROCRYPT '04), Interlaken, Switzerland, May 2004, pp. 268–286, Springer, 2004.

[25] C. P. Schnorr, "Efficient signature generation by smart cards," Journal of Cryptology, vol. 4, no. 3, pp. 161–174, 1991.

[26] C.-C. Chang and C.-W. Chan, "A database record encryption scheme using the RSA public key cryptosystem and its master keys," in Proceedings of the International Conference on Computer Networks and Mobile Computing (ICCNMC '03), pp. 345–348, Shanghai, China, October 2003.

[27] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," Wireless Networks, vol. 8, no. 5, pp. 521–534, 2002.

[28] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in Advances in Cryptology—CRYPTO 2001: 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19–23, 2001 Proceedings, vol. 2139 of Lecture Notes in Computer Science, pp. 213–229, Springer, Berlin, Germany, 2001.

[29] G. M. Bertoni, L. Chen, P. Fragneto, K. A. Harrison, and G. Pelosi, "Computing tate pairing on smartcards," White Paper, STMicroelectronics, 2005.
