Practical Secure Communication for Integrating Wireless Sensor Networks Into the Internet of Things

Fagen Li and Pan Xiong

***Abstract*— If a wireless sensor network (WSN) is integrated into the Internet as a part of the Internet of things (IoT), there will appear new security challenges, such as setup of a secure channel between a sensor node and an Internet host. In this paper, we propose a heterogeneous online and offline signcryption scheme to secure communication between a sensor node and an Internet host. We prove that this scheme is indistinguishable against adaptive chosen ciphertext attacks under the bilinear Diffie-Hellman inversion problem and existential unforgeability against adaptive chosen messages attacks under the q-strong Diffie-Hellman problem in the random oracle model. Our scheme has the following advantages. First, it achieves confidentiality, integrity, authentication, and non-repudiation in a logical single step. Second, it allows a sensor node in an identity-based cryptography to send a message to an Internet host in a public key infrastructure. Third, it splits the signcryption into two phases: i) offline phase; and ii) online phase. In the offline phase, most heavy computations are done without the knowledge of a message. In the online phase, only light computations are done when a message is available. Our scheme is very suitable to provide security solution for integrating WSN into the IoT.**

***Index Terms*— Wireless sensor network, Internet of things, security, signcryption, public key infrastructure, identity-based cryptography.**

I. Introduction

THE Internet of Things (IoT) is a novel paradigm that has received considerable attention from both academia and industry. The basic idea of IoT is the pervasive presence around us of a variety of things or objects-such as radio-frequency identification (RFID) tags, sensors, actuators, mobile phones, etc.-which, through unique addressing schemes, are able to interact with each other and cooperate with their neighbors to reach common goals [1]. Wireless sensor networks (WSNs) are ad hoc networks which usually consist of a large number of tiny sensor nodes with limited resources and one or more base stations. Usually, sensor nodes consist of a processing unit with limited computational power and limited capacity. On the other hand, the base station is a powerful trusted device that acts as an interface between the network user and the nodes. WSNs have many applications, including military sensing and tracking, environment monitoring, target tracking, healthcare monitoring, and so on. A user of the WSNs can read the data received from the sensors through the base station. If we hope to read the data anywhere in the world, we need to integrate the WSNs into the Internet as part of the IoT. There are three methods to accomplish this integration, front-end proxy solution, gateway solution and TCP/IP overlay solution [2]. In the front-end proxy solution, the base station acts as an interface between the WSNs and the Internet. There is no direct connection between the Internet and a sensor node. The base station parses all incoming and outgoing information. In the gateway solution, the base station acts as an application layer gateway that translates the lower layer protocols from both networks. In the TCP/IP overlay solution, sensor nodes communicate with other nodes using TCP/IP. The base station acts as a router that forwards the packets from and to the sensor nodes. In both gateway solution and TCP/IP overlay solution, the sensor nodes can communicate with the Internet hosts directly. However, new security challenges will appear, such as setup of a secure channel between a sensor node and an Internet host that supports end-to-end authentication and confidentiality services. Note that the computational power and storage of a sensor node are limited. But an Internet host has strong computational power and storage. So we hope to design a secure communication scheme that fits such a characteristic.

To support the authenticity of public keys in the public key cryptography, there are two main infrastructures called public key infrastructure (PKI) and identity-based cryptography (IBC) [3]. In the PKI, a certificate authority (CA) issues a certificate which provides an unforgeable and trusted link between the public key and the identity of a user by the signature of the CA. The drawback of the PKI is that we need to manage certificates, including revocation, storage and distribution. In addition, we need to verify the validity of certificates before using them. The PKI technique has been widely developed and applied in the Internet. In the IBC, a user's public key is derived directly from its identity information, such as telephone numbers, email addresses and IP addresses. Secret keys are generated for users by a trusted third party called private key generator (PKG). Authenticity of a public key is explicitly verified without requiring any certificate. The advantage of the IBC is that we eliminate the need for certificates and some of the problems associated with

Manuscript received January 30, 2013; revised April 14, 2013; accepted May 1, 2013. Date of publication June 21, 2013; date of current version August 28, 2013. This work was supported by the National Natural Science Foundation of China under Grants 60803133, 61073176, 61003230, 61003232, 61103207, 61272404 and 61272525. The associate editor coordinating the review of this paper and approving it for publication was Prof. Rose Qingyang Hu.

The authors are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: fagenli@uestc.edu.cn; 276878323@qq.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSEN.2013.2262271

them. On the other hand, the dependence on the PKG who can generate all users' secret keys inevitably causes the key escrow problem in the IBC. For the WSNs, IBC is the best choice because there is no certificates problem. However, IBC is only suitable for small networks. For the Internet security, we need PKI technique.

A. Motivation and Contribution

The motivation of this paper is to setup a secure channel between a sensor node and an Internet host that supports end-to-end confidentiality, integrity, authentication and non-repudiation services. In addition, we require that the IBC is used in the sensor node and that the PKI is used in the Internet host. We also require that the computational cost of sensor nodes is low. Our solution is heterogeneous online/offline signcryption (HOOSC). Concretely, we propose an efficient HOOSC scheme. We prove that the proposed scheme has the indistinguishability against adaptive chosen ciphertext attacks (IND-CCA2) under the bilinear Diffie-Hellman inversion problem (BDHIP) and existential unforgeability against adaptive chosen messages attacks (EUF-CMA) under the q-strong Diffie-Hellman problem (q-SDHP) in the random oracle model. Our scheme has the following characteristics: (i) It achieves confidentiality, integrity, authentication and non-repudiation in a logical single step. (ii) It allows a sensor node in the IBC to send a message to an Internet host in the PKI. (iii) It splits the signcryption into two phases: offline phase and online phase. In the offline part, most heavy computations are done without the knowledge of a message. In the online stage, only light computations are done when a message is known.

B. Related Work

Signcryption [4] is a new cryptographic primitive that fulfills both the functions of digital signature and public key encryption in a logical single step, at a cost significantly lower than that required by the traditional signature-then-encryption approach. That is, signcryption can simultaneously achieves confidentiality, integrity, authentication and non-repudiation at a lower cost. The performance advantage of signcryption over the signature-then-encryption method makes signcryption useful in many applications, such as electronic commerce, mobile communications and smart cards. Some PKI-based signcryption schemes [5]–[8] and IBC-based signcryption schemes [9]–[13] are proposed. However, these signcryption schemes are homogeneous and can not be used in heterogeneous communications.

In 2010, Sun and Li [14] proposed two heterogeneous signcryption schemes. The first scheme allows a sender in the PKI to send a message to a receiver in the IBC. The second scheme allows a sender in the IBC to send a message to a receiver in the PKI. But their schemes are only secure against outsider attacks (i.e. attacks made by an attacker who is neither the sender nor the receiver). Such signcryption schemes do not provide any kind of non-repudiation function. A stronger notion than outsider security is insider security [15]. The insider security means that (i) if a sender's secret key is exposed, an attacker is still not able to recover the message from the ciphertext and (ii) if a receiver's secret key is exposed, an attacker is still not able to forge a ciphertext. To achieve the insider security, Huang, Wong and Yang [16] proposed a heterogeneous signcryption scheme that allows a user in the IBC to send a message to a receiver in the PKI. Both [14] and [16] are not suitable for WSNs because the computational cost is high for signcrypting a message.

C. Organization

The rest of this paper is organized as follows. We introduce the preliminary work in Section II. We give the formal model of HOOSC in Section III. An efficient HOOSC scheme is proposed in Section IV. We analyze the proposed scheme in Section V. Finally, the conclusions are given in Section VI.

II. Preliminaries

In this section, we briefly describe the basic definition and properties of the bilinear pairings.

Let G_1 be a cyclic additive group generated by P, whose order is a prime p, and G_2 be a cyclic multiplicative group of the same order p. A bilinear pairing is a map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ with the following properties:

1) Bilinearity: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in G_1$, $a, b \in \mathbb{Z}_p^*$.
2) Non-degeneracy: There exists $P, Q \in G_1$ such that $\hat{e}(P, Q) \neq 1$.
3) Computability: There is an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in G_1$.

The modified Weil pairing and the Tate pairing are admissible maps of this kind. Please see [9]–[13] for more details. The security of our scheme relies on the hardness of the following problems.

Given two groups G_1 and G_2 of the same prime order p, a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ and a generator P of G_1, the q-bilinear Diffie-Hellman inversion problem (q-BDHIP) in $(G_1, G_2, \hat{e})$ is to compute $\hat{e}(P, P)^{1/\alpha}$ given $(P, \alpha P, \alpha^2 P, \ldots, \alpha^q P)$. We call bilinear Diffie-Hellman inversion problem (BDHIP) when $q = 1$.

Definition 1: The (ϵ, t)-BDHIP assumption holds if no t-polynomial time adversary $\mathcal{C}$ has advantage at least ϵ in solving the BDHIP problem.

Given two groups G_1 and G_2 of the same prime order p, a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ and a generator P of G_1, the q-strong Diffie-Hellman problem (q-SDHP) in $(G_1, G_2, \hat{e})$ is to find a pair $(w, \frac{1}{\alpha+w}P) \in \mathbb{Z}_p^* \times G_1$ given $(P, \alpha P, \alpha^2 P, \ldots, \alpha^q P)$.

Definition 2: The (ϵ, t)-q-SDHP assumption holds if no t-polynomial time adversary $\mathcal{C}$ has advantage at least ϵ in solving the q-SDHP problem.

III. Formal Model of HOOSC

In this section, we give the formal definition and security notions of HOOSC. This paper only discuss the case that senders belong to the IBC and receivers belong to the PKI.

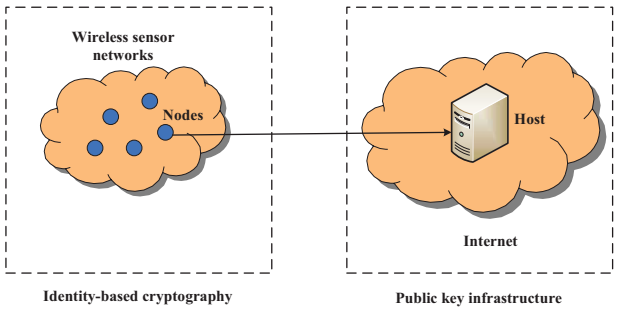

Fig. 1. Communication model for integrating WSNs into the Internet.

A. Syntax

A generic HOOSC scheme consists of the following five algorithms.

Setup: This is a probabilistic algorithm run by a PKG that takes as input a security parameter k, and outputs a master secret key msk and the system parameters *params* that includes a master public key mpk.

IBC-KG: This is a key generation algorithm for IBC users. The user submits an identity ID to its PKG. The PKG computes the corresponding secret key sk and transmits it to the user in a secure way. Note that the user's public key pk is identity ID. Such a public key does not need a digital certificate.

PKI-KG: This is a key generation algorithm for PKI users. The user chooses its secret key sk and publishes the corresponding public key pk. This public key needs a digital certificate that is sign by its CA.

Off-Signcrypt: This is a probabilistic offline signcryption algorithm run by a sender that takes as input the system parameters *param*, a sender's private key sk_s and a receiver's public key pk_r, and outputs an offline signcryption δ.

On-Signcrypt: This is a probabilistic online signcryption algorithm run by a sender that takes as input the system parameters *param*, a message m and an offline signcryption δ, and outputs a full signcryption ciphertext σ.

Unsigncrypt: This is a deterministic unsigncryption algorithm run by a receiver that takes as input a ciphertext σ, a sender's public key pk_s and the receiver's secret key sk_r, and outputs the plaintext m or the symbol $\perp$ if σ is an invalid ciphertext between the sender and the receiver.

These algorithms must satisfy the standard consistency constraint of HOOSC, i.e. if δ=*Off-Signcrypt*(sk_s, pk_r) and σ=*On-Signcrypt*(m, δ), then we have $m = Unsigncrypt(\sigma, pk_s, sk_r)$. Note that we omit *param* in *Off-Signcryptn*, *On-Signcryptn* and *Unsigncrypt* for simplicity.

For secure communication for integrating WSNs into the Internet, a sensor node is regarded as a sender and an Internet host is regarded as a receiver. HOOSC provides a secure channel between the sensor node and the Internet host that supports end-to-end confidentiality, integrity, authentication and non-repudiation services. Figure 1 shows the communication model for integrating WSNs into the Internet.

B. Security Notions

The standard security notions for signcryption are indistinguishability against adaptive chosen ciphertext attacks (IND-CCA2) (confidentiality) and existential unforgeability against adaptive chosen messages attacks (EUF-CMA) (unforgeability). We modify the notions in [9], [13] slightly to adapt for HOOSC.

For the IND-CCA2 security, we consider the following game played between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Initial: $\mathcal{C}$ runs *Setup* algorithm with a security parameter k and sends the system parameters and master secret key msk to $\mathcal{A}$. In addition, $\mathcal{C}$ also runs *PKI-KG* algorithm to get a receiver's public/secret key pair (pk_r^*, sk_r^*) and sends pk_r^* to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ performs a polynomially bounded number of unsigncryption queries in an adaptive manner. In an unsigncryption query, $\mathcal{A}$ submits a sender's identity ID_s and a ciphertext σ. $\mathcal{C}$ runs $Unsigncrypt(\sigma, ID_s, sk_r^*)$ algorithm and sends the result to $\mathcal{A}$.

Challenge: $\mathcal{A}$ decides when Phase 1 ends. $\mathcal{A}$ generates two equal length plaintexts m_0 and m_1 and a sender's identity ID_s^*. $\mathcal{C}$ first runs *IBC-KG* algorithm to generate the sender's secret key $sk_{ID_s^*}$. Then $\mathcal{C}$ takes a random bit $\gamma \in \{0, 1\}$ and computes δ^*=*Off-Signcrypt*$(sk_{ID_s^*}, pk_r^*)$ and σ^*=*On-Signcrypt*(m_γ, δ). Finally, $\mathcal{C}$ sends σ^* to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ can ask a polynomially bounded number of queries adaptively again as in the Phase 1. This time, it cannot make an unsigncryption query on (σ^*, ID_s^*) to obtain the corresponding plaintext.

Guess: $\mathcal{A}$ produces a bit γ' and wins the game if $\gamma' = \gamma$.

The advantage of $\mathcal{A}$ is defined as $Adv(\mathcal{A}) = |2\Pr[\gamma' = \gamma] - 1|$, where $\Pr[\gamma' = \gamma]$ denotes the probability that $\gamma' = \gamma$.

Definition 3: A HOOSC scheme is (ϵ, t, q_u)-IND-CCA2 secure if no probabilistic t-polynomial time adversary $\mathcal{A}$ has advantage at least ϵ after at most q_u unsigncryption queries in the IND-CCA2 game.

Note that the above definition catches insider security for confidentiality of signcryption since the adversary knows the master secret key msk and all senders' secret keys [15]. The insider security ensures the forward security of the scheme, i.e. confidentiality is preserved in case the sender's secret key becomes compromised.

For the EUF-CMA security, we consider the following game played between a challenger $\mathcal{C}$ and an adversary $\mathcal{F}$.

Initial: $\mathcal{C}$ runs *Setup* algorithm with a security parameter k and sends the system parameters to $\mathcal{F}$. In addition, $\mathcal{C}$ runs *PKI-KG* algorithm to get a receiver's public/secret key pair (pk_r^*, sk_r^*) and sends (pk_r^*, sk_r^*) to $\mathcal{F}$.

Attack: $\mathcal{F}$ performs a polynomially bounded number of key generation and signcryption queries in an adaptive manner. In a key generation query, $\mathcal{F}$ chooses an identity ID. $\mathcal{C}$ runs *IBC-KG* algorithm and sends corresponding secret key sk_{ID} to $\mathcal{F}$. In a signcryption query, $\mathcal{F}$ produces a sender's identity ID_s and a message m. $\mathcal{C}$ first runs *IBC-KG* algorithm to generate the sender's secret key sk_{ID_s}. Then $\mathcal{C}$ runs δ=*Off-*

Signcrypt(sk_{ID_s}, pk_r^*) and σ=*On-Signcrypt*(m, δ) algorithms. Finally, $\mathcal{C}$ sends σ to $\mathcal{F}$.

Forgery: $\mathcal{F}$ produces a sender's identity ID_s^* and a ciphertext σ^* and wins the game if the following conditions hold:

1) *Unsigncrypt*(σ^*, ID_s^*, sk_r^*) = m^*.
2) $\mathcal{F}$ has not made a key generation query on ID_s^*.
3) $\mathcal{F}$ has not made a signcryption query on (m^*, ID_s^*).

The advantage of $\mathcal{F}$ is defined as the probability that it wins.

Definition 4: A HOOSC scheme is (ϵ, t, q_k, q_s)-EUF-CMA secure if no probabilistic t-polynomial time adversary $\mathcal{F}$ has advantage at least ϵ after at most q_k key generation queries and q_s signcryption queries in the EUF-CMA game.

Note that the adversary knows the receiver's secret key sk_r^* in the above definition. That is, the definition also catches the insider security for unforgeability of signcryption [15].

IV. A HOOSC Scheme

In this section, we propose an efficient HOOSC scheme which is based on Barreto et al.'s signcryption scheme [13]. Our scheme allows a sender in the IBC to send a message to a receiver in the PKI. It consists of the following five algorithms.

Setup: Given a security parameter k, the PKG chooses groups G_1 and G_2 of prime order p (with G_1 additive and G_2 multiplicative), a generator P of G_1, a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$, and hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, $H_2 : \{0,1\}^n \times G_2 \times G_1 \rightarrow \mathbb{Z}_p^*$, and $H_3 : G_2 \rightarrow \{0,1\}^n$. Here n is the number of bits of a message to be signcrypted. The PKG chooses a master secret key $s \in \mathbb{Z}_p^*$ randomly and computes the master public key $P_{pub} = sP$. The PKG also computes $g = \hat{e}(P, P)$. The PKG publishes system parameters $\{G_1, G_2, n, \hat{e}, P, P_{pub}, g, H_1, H_2, H_3\}$ and keeps the master secret key s secret.

IBC-KG: A user in the IBC submits its identity ID to its PKG. The PKG computes the corresponding secret key $S_{ID} = \frac{1}{H_1(ID)+s}P$ and sends it to the user in a secure way. We will denote the sender by ID_s and its key pair by ($pk_s = ID_s$, $sk_s = S_{ID_s}$) in the following description.

PKI-KG: A user u in the PKI chooses a random number x_u from $\mathbb{Z}_p^*$ and computes $sk_u = \frac{1}{x_u}P$ as its secret key and $pk_u = x_u P$ as its public key. We will denote the receiver by $u = r$ and its key pair by ($pk_r = x_r P, sk_r = \frac{1}{x_r}P$) in the following description.

Off-Signcrypt: Given a sender's secret key S_{ID_s} and a receiver's public pk_r, this algorithm works as follows.

1) Choose x, β from $\mathbb{Z}_p^*$ randomly.
2) Compute $r = g^x$.
3) Compute $S = \beta S_{ID_s}$.
4) Compute $T = xpk_r$.

The offline signcryption is $\delta = (x, r, \beta, S, T)$.

On-Signcrypt: Given a message m and an offline signcryption δ, this algorithm works as follows.

1) Compute $c = m \oplus H_3(r)$.
2) Compute $h = H_2(m, r, S)$.
3) Compute $\theta = (x + h)\beta^{-1} \bmod p$.

The ciphertext is $\sigma = (c, \theta, S, T)$.

Unsigncrypt: Given a ciphertext σ, a sender's identity ID_s and a receiver's secret key sk_r, this algorithm works as follows.

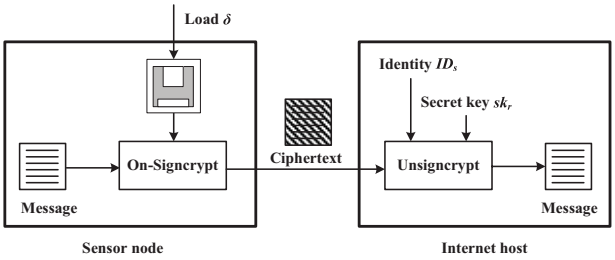

Fig. 2. Steps for secure communication using our scheme.

1) Compute $r = \hat{e}(T, sk_r)$.
2) Recover $m = c \oplus H_3(r)$.
3) Compute $h = H_2(m, r, S)$.
4) Accept the message if and only if $r = \hat{e}(\theta S, H_1(ID_s)P + P_{pub})g^{-h}$, return $\perp$ otherwise.

For secure communication for integrating WSNs into the Internet, a sensor node is regarded as a sender (the media access control address of the sensor node can be used for the identity ID) and an Internet host is regarded as a receiver. First, the sensor node is loaded with precomputed results $\delta = (x, r, \beta, S, T)$ of the offline phase from a more powerful device. When the sensor node wants to send a message m to the Internet host, the sensor node runs σ=*On-Signcrypt*(m, δ) algorithm and sends the ciphertext σ to the Internet host. In this process, the sensor node only does light computations, such as exclusive OR, hash function, modular multiplication and modular inverse. When receiving the ciphertext σ, the Internet host runs m = *Unsigncrypt*(σ, ID_s, sk_r) algorithm to obtain the message m. Our scheme simultaneously achieves confidentiality, integrity, authentication and non-repudiation. Figure 2 shows steps for secure communication using our scheme.

V. Analysis of the Scheme

In this section, we analyze the consistency, security and performance of the proposed scheme.

A. Consistency

Now we verify the consistency of the proposed scheme. First, since

$$\begin{aligned}\hat{e}(T, sk_r) &= \hat{e}(xpk_r, \frac{1}{x_r}P)\\ &= \hat{e}(xx_rP, \frac{1}{x_r}P)\\ &= \hat{e}(P, P)^x\\ &= g^x\\ &= r\end{aligned}$$

we can recover the message by computing $m = c \oplus H_3(r)$. Second, since $\theta S = (x + h)\beta^{-1}\beta S_{ID_s} = (x + h)S_{ID_s}$, we can

verify the signature by

$$\begin{aligned}
&\hat{e}(\theta S, H_1(ID_s)P + P_{pub})g^{-h} \\
&= \hat{e}((x+h)S_{ID_s}, (H_1(ID_s)+s)P)g^{-h} \\
&= \hat{e}((x+h)\frac{1}{H_1(ID_s)+s}P, (H_1(ID_s)+s)P)g^{-h} \\
&= \hat{e}((x+h)P, P)g^{-h} \\
&= \hat{e}(P, P)^{(x+h)}g^{-h} \\
&= g^{(x+h)}g^{-h} \\
&= g^{x} \\
&= r
\end{aligned}$$

B. Security

We prove that the proposed scheme satisfies IND-CCA2 and EUF-CMA by the following Theorems 1 and 2, respectively.

Theorem 1: In the random oracle model, if an adversary $\mathcal{A}$ has a non-negligible advantage ϵ against the IND-CCA2 security of the proposed scheme when running in a time t and performing q_u unsigncryption queries and q_{H_i} queries to oracles H_i $(i = 1, 2, 3)$, then there exists an algorithm $\mathcal{C}$ that can solve the BDHIP with an advantage $\epsilon' \geq \frac{\epsilon}{2q_{H_2}+q_{H_3}}(1-\frac{q_u}{2^k})$ in a time $t' \leq t + O(q_u)t_p + O(q_u q_{H_2})t_e$, where t_p denotes the cost for one pairing computation and t_e denotes the cost for an exponentiation computation in G_2.

Proof: We show how $\mathcal{C}$ can use $\mathcal{A}$ as a subroutine to solve a random given instance $(P, \alpha P)$ of the BDHIP.

Initial: $\mathcal{C}$ chooses a master secret key $s \in \mathbb{Z}_p^*$ randomly and sets master public key $P_{pub} = sP$. $\mathcal{C}$ also sets the receiver's public key $pk_r^* = \alpha P$ and $g = \hat{e}(P, P)$. Then $\mathcal{C}$ sends the system parameters and the receiver's public key pk_r^* to $\mathcal{A}$.

Phase 1: $\mathcal{C}$ simulates $\mathcal{A}$'s challenger in the IND-CCA2 game. $\mathcal{C}$ maintains three lists L_1, L_2 and L_3 to simulate the hash oracles H_1, H_2 and H_3, respectively. We assume that H_1 queries are distinct and that $\mathcal{A}$ will ask for $H_1(ID)$ before ID is used in any other queries.

- H_1 queries: For a $H_1(ID_i)$ query, $\mathcal{C}$ first checks if the value of H_1 was previously defined for the input ID_i. If it was, the previously defined value is returned. Otherwise, $\mathcal{C}$ returns a random $h_{1,i} \in \mathbb{Z}_p^*$ as the answer and inserts the pair $(ID_i, h_{1,i})$ into the list L_1.
- H_2 queries: For a $H_2(m_i, r_i, S_i)$ query, $\mathcal{C}$ first checks if the value of H_2 was previously defined for the input (m_i, r_i, S_i). If it was, the previously defined value is returned. Otherwise, $\mathcal{C}$ returns a random $h_{2,i} \in \mathbb{Z}_p^*$ as the answer. In addition, $\mathcal{C}$ simulates random oracle on its own to obtain $h_{3,i} = H_3(r_i) \in \{0,1\}^n$ and computes $c_i = m_i \oplus h_{3,i}$ and $\xi_i = r_i \cdot \hat{e}(P, P)^{h_{2,i}}$. Finally, $\mathcal{C}$ inserts the tuple $(m_i, r_i, S_i, h_{2,i}, c_i, \xi_i)$ into the list L_2.
- H_3 queries: For a $H_3(r_i)$ query, $\mathcal{C}$ first checks if the value of H_3 was previously defined for the input r_i. If it was, the previously defined value is returned. Otherwise, $\mathcal{C}$ randomly chooses $h_{3,i}$ from $\{0,1\}^n$, returns $h_{3,i}$ as an answer and inserts the tuple $(r_i, h_{3,i})$ into the list L_3.
- Unsigncryption queries: At any time $\mathcal{A}$ can perform an unsigncryption query for a ciphertext $\sigma = (c, \theta, S, T)$ and a sender's identity ID_i. $\mathcal{C}$ runs the H_1 simulation algorithm to find $h_{1,i} = H_1(ID_i)$ and computes the sender's private key $S_{ID_i} = \frac{1}{h_{1,i}+s}P$. For all valid ciphertexts, we have $\log_{S_{ID_i}}(\theta S - hS_{ID_i}) = \log_{pk_r^*} T$, where $h = H_2(m, r, S)$. Therefore, the following equation holds $\hat{e}(T, S_{ID_i}) = \hat{e}(pk_r^*, \theta S - hS_{ID_i})$. $\mathcal{C}$ first computes $\xi = \hat{e}(\theta S, pk_r^*)$ and then searches L_2 for the entries of the form $(m_i, r_i, S_i, h_{2,i}, c, \xi)$ indexed by $i \in \{1, \ldots, q_{H_2}\}$. If no an entry is found, σ is rejected. Otherwise, $\mathcal{C}$ further checks if the following equation holds for the corresponding indexes $\frac{\hat{e}(T, S_{ID_i})}{\hat{e}(pk_r^*, \theta S)} = \hat{e}(pk_r^*, S_{ID_i})^{-h_{2,i}}$. If the unique $i \in \{1, \ldots, q_{H_2}\}$ satisfying the above equation is found, the matching message m_i is returned. Otherwise, σ is rejected. It is easy to see that, for all queries, the probability to reject a valid ciphertext does not exceed $\frac{q_u}{2^k}$.

Challenge: $\mathcal{A}$ generates two equal length plaintexts m_0 and m_1 and the sender's identity ID_s^* on which it wishes to be challenged. $\mathcal{C}$ chooses $c^* \in \{0,1\}^n$, $\lambda, \theta^* \in \mathbb{Z}_p^*$, $S^* \in G_1$ randomly and computes $T^* = \lambda P$. $\mathcal{C}$ returns the ciphertext $\sigma^* = (c^*, \theta^*, S^*, T^*)$ to $\mathcal{A}$. $\mathcal{A}$ cannot recognize that σ^* is not a valid ciphertext unless it queries H_2 or H_3 on $\hat{e}(P, P)^{\lambda/\alpha}$.

Phase 2: $\mathcal{A}$ can ask a polynomially bounded number of queries adaptively again as in the Phase 1 with the restriction that it cannot make an unsigncryption query on σ^* to obtain the corresponding plaintext. $\mathcal{C}$ answer $\mathcal{A}$'s queries as in the Phase 1.

Guess: $\mathcal{A}$ produces a bit γ' which is ignored by $\mathcal{C}$.

$\mathcal{C}$ fetches a random entry $(m_i, r_i, S_i, h_{2,i}, c_i, \xi_i)$ or $(r_i, h_{3,i})$ from the lists L_2 or L_3. Since L_3 contains no more than $q_{H_2} + q_{H_3}$ records, the chosen entry will contain the right element $r_i = \hat{e}(P, P)^{\lambda/\alpha}$ with probability $1/(2q_{H_2} + q_{H_3})$. The BDHIP can be extracted by computing $(\hat{e}(P, P)^{\lambda/\alpha})^{\lambda^{-1}}$.

This completes the description of the simulation. It remains to analyze $\mathcal{C}$'s advantage. Define the event E as that $\mathcal{C}$ aborts in an unsigncryption query because of rejecting a valid ciphertext.

From the above analysis, we know that the probability of $\mathcal{C}$ not aborting is $\Pr[\neg\text{abort}] = \Pr[\neg E]$. We know that $\Pr[E] \leq q_u/2^k$. So we have $\Pr[\neg\text{abort}] \geq (1 - \frac{q_u}{2^k})$. In addition, $\mathcal{C}$ selects the correct element from L_2 or L_3 with probability $1/(2q_{H_2} + q_{H_3})$. Therefore, we have $\epsilon' \geq \frac{\epsilon}{2q_{H_2}+q_{H_3}}(1 - \frac{q_u}{2^k})$.

The bound on $\mathcal{C}$'s computation time derives from the fact that $\mathcal{C}$ needs $O(q_u)$ pairing calculations and $O(q_u q_{H_2})$ exponentiations in G_2 in the unsigncryption queries. ■

Theorem 2: In the random oracle model, if an adversary $\mathcal{F}$ has an advantage $\epsilon \geq 10(q_s+1)(q_s+q_{H_2})/2^k$ against the EUF-CMA security of the proposed scheme when running in a time t and performing q_s signcryption queries and q_{H_i} queries to oracles H_i $(i = 1, 2, 3)$, then there exists an algorithm $\mathcal{C}$ that can solve the q-SDHP for $q = q_{H_1}$ in expected time $t' \leq 120686 q_{H_1} q_{H_2} \frac{t+O(q_s t_p)}{\epsilon(1-1/2^k)(1-q/2^k)} + O(q^2 t_m)$, where t_p denotes the cost for one pairing computation and t_m denotes the cost for a scalar multiplication computation in G_1.

Proof: Similar to [13], we use the forking lemma [17] to prove the security of our scheme. To use the forking lemma, we need to show how the proposed scheme fits into the signature scheme described in [17], the simulation step

in which the signature can be simulated without the secret signcryption key of the sender, and how we can solve q-SDHP based on the forgery.

First, we observe that our scheme satisfies the requirement described in [17]. During the signcryption of message m, the tuple (σ_1, h, σ_2) is produced which corresponds to the required three-phase honest-verifier zero-knowledge identification protocol, where $\sigma_1 = r$ is the commitment of the prover, $h = H_2(m, r, S)$ is the hash value depending on m and σ_1 substituted for the verifier's challenge, and $\sigma_2 = \theta S$ is the response of the prover which depends on σ_1, h and the signcryption private key S_{ID_s}.

Next, we show $\mathcal{C}$ can provide a faithful simulation to $\mathcal{F}$ and solve the q-SDHP by interacting with $\mathcal{F}$. $\mathcal{C}$ takes as input $(P, \alpha P, \alpha^2 P, \ldots, \alpha^q P)$ and aims to find a pair $(w, \frac{1}{\alpha+w} P)$. $\mathcal{C}$ simulates $\mathcal{F}$'s challenger in the EUF-CMA game. $\mathcal{F}$ then adaptively performs key generation and signcryption queries as explained in the EUF-CMA game. We describe this process as follows.

Initial: First, $\mathcal{C}$ chooses $w_1, w_2 \ldots, w_{q-1} \in \mathbb{Z}_p^*$ randomly. $\mathcal{C}$ takes $(P, \alpha P, \alpha^2 P, \ldots, \alpha^q P)$ as input to compute a generator $Q \in G_1$ and another element $Q_{pub} = \alpha Q \in G_1$ such that it knows $q-1$ pairs $(w_i, V_i = \frac{1}{\alpha+w_i} Q)$ for $i \in \{1, \ldots, q-1\}$ as in the proof technique of [18]. To do so, $\mathcal{C}$ expands the polynomial $f(z) = \prod_{i=1}^{q-1}(z + w_i) = \sum_{j=0}^{q-1} c_j z^j$. A generator Q and an element Q_{pub} are then obtained as $Q = \sum_{j=0}^{q-1} c_j(\alpha^j P) = f(\alpha)P$ and $Q_{pub} = \sum_{j=1}^{q} c_{j-1}(\alpha^j P) = \alpha f(\alpha)P = \alpha Q$. As in [18], the pairs (w_i, V_i) are obtained by expanding $f_i(z) = \frac{f(z)}{z+w_i} = \sum_{j=0}^{q-2} d_j z^j$ and computing $V_i = \sum_{j=0}^{q-2} d_j(\alpha^j P) = f_i(\alpha)P = \frac{f(\alpha)}{\alpha+w_i} P = \frac{1}{\alpha+w_i} Q$. The PKG's public key is Q_{pub} and its corresponding master secret key is α.

$\mathcal{C}$ sends $\mathcal{F}$ the system parameters with the generator Q, $Q_{pub} = \alpha Q$ and $g = \hat{e}(Q, Q)$. $\mathcal{C}$ chooses a random challenge identity $ID_s^* \in \{0, 1\}^*$ and sends it to $\mathcal{F}$. In addition, $\mathcal{C}$ runs PKI-KG algorithm to get a receiver's public/secret key pair (pk_r^*, sk_r^*) and sends (pk_r^*, sk_r^*) $\mathcal{F}$.

Attack: $\mathcal{C}$ simulates $\mathcal{F}$'s challenger in the EUF-CMA game. $\mathcal{C}$ maintains three lists L_1, L_2 and L_3 to simulate the hash oracles H_1, H_2 and H_3, respectively. We also assume that H_1 queries are distinct and that $\mathcal{F}$ will ask for $H_1(ID)$ before ID is used in any other queries.

- H_1 queries: These queries are indexed by a counter ν that is initially set to 1. If $ID = ID_s^*$, $\mathcal{C}$ returns a random $w_s \in \mathbb{Z}_p^*$ as the answer. Otherwise, $\mathcal{C}$ returns w_ν as the answer and increments ν. In both cases, $\mathcal{C}$ puts the tuple (ID, w) (where $w = w_s$ or w_ν) into the list L_1.
- H_2 queries: For a $H_2(m_i, r_i, S_i)$ query, $\mathcal{C}$ first checks if the value of H_2 was previously defined for the input (m_i, r_i, S_i). If it was, the previously defined value is returned. Otherwise, $\mathcal{C}$ returns a random $h_{2,i} \in \mathbb{Z}_p^*$ as the answer.
- H_3 queries: For a $H_3(r_i)$ query, $\mathcal{C}$ first checks if the value of H_3 was previously defined for the input r_i. If it was, the previously defined value is returned. Otherwise, $\mathcal{C}$ randomly chooses $h_{3,i}$ from $\{0, 1\}^n$, returns $h_{3,i}$ as an answer and inserts the tuple $(r_i, h_{3,i})$ into the list L_3.
- Key generation queries: When $\mathcal{F}$ makes a key generation query on an identity ID_i, if $ID_i = ID_s^*$, then $\mathcal{C}$ fails and stops. Otherwise, $\mathcal{C}$ knows $H_1(ID_i) = w_i$ and returns $V_i = \frac{1}{\alpha+w_i} Q$ to $\mathcal{F}$.
- Signcryption queries: $\mathcal{F}$ chooses a plaintext m and a sender's identity ID_i. If $ID_i \neq ID_s^*$, then $\mathcal{C}$ knows the sender's private key $S_{ID_i} = V_i$ and can answer the query according to the steps of *Off-Signcrypt* and *On-Signcrypt*. If $ID_i = ID_s^*$, $\mathcal{C}$ does the following steps.
  1) Choose $\eta, \theta, h \in \mathbb{Z}_p^*$ randomly.
  2) Compute $S = \theta^{-1}\eta s k_r^*$.
  3) Compute $T = \eta(w_j Q + Q_{pub}) - h p k_r^*$.
  4) Compute $r = \hat{e}(T, sk_r^*)$.
  5) Patch the hash value $H_2(m, r, S)$ to h. $\mathcal{C}$ fails if H_2 is already defined but this only happens with probability $(q_s + q_{H_2})/2^k$.
  6) Compute $c = m \oplus H_3(r)$.
  7) Return $\sigma = (c, \theta, S, T)$ to $\mathcal{F}$.

Next, we coalesce the sender identity ID_s^* and the message m into a "generalized" forged message (ID_s^*, m) so as to hide the identity-based aspect of the EUF-CMA attacks, and simulate the setting of an identity-less adaptive-CMA existential forgery for which the forking lemma is proven.

From the forking lemma, if $\mathcal{F}$ is an efficient forger in the above interaction, then we can construct a Las Vegas machine $\mathcal{F}'$ that outputs two signed messages $((ID_s^*, m), h, \theta, S)$ and $((ID_s^*, m), h^*, \theta^*, S^*)$ with $h \neq h^*$ and the same commitment.

Finally, to solve the q-SDHP based on the machine $\mathcal{F}'$ derived from $\mathcal{F}$, we construct a machine $\mathcal{C}$ as follows.

1) $\mathcal{C}$ gets two distinct signatures $((ID_s^*, m), h, \theta, S)$ and $((ID_s^*, m), h^*, \theta^*, S^*)$ by running $\mathcal{F}'$.
2) $\mathcal{C}$ computes $V^* = (h - h^*)^{-1}(\theta S - \theta^* S^*) = \frac{1}{\alpha+w_s} Q = \frac{f(\alpha)}{\alpha+w_s} P$.
3) $\mathcal{C}$ uses long division and writes the polynomial f as $f(z) = \psi(z)(z+w_s)+\psi_{-1}$ for some polynomial $\psi(z) = \sum_{i=0}^{q-2} \psi_i z^i$ and some $\psi_{-1} \in Z_p^*$. Then $\frac{f(z)}{z+w_s}$ can be written as $\frac{f(z)}{z+w_s} = \psi(z)+\frac{\psi_{-1}}{z+w_s} = \sum_{i=0}^{q-2} \psi_i z^i + \frac{\psi_{-1}}{z+w_s}$. Hence, $\mathcal{C}$ can compute $\frac{1}{\alpha+w_s} P = \frac{1}{\psi_{-1}}(V^* - \sum_{i=0}^{q-2} \psi_i(x^i P))$.
4) $\mathcal{C}$ outputs $(w_s, \frac{1}{\alpha+w_s} P)$ as the solution of q-SDHP.

From the forking lemma and the lemma on the relationship between given-identity attack and chosen-identity attack [19], if $\mathcal{F}$ succeeds in a time t with probability $\epsilon \geq 10(q_s+1)(q_s+q_{H_2})/2^k$, then $\mathcal{C}$ can solve the q-SDHP in expected time $t' \leq 120686 q_{H_1} q_{H_2} \frac{t+O(q_s t_p)}{\epsilon(1-1/2^k)(1-q/2^k)} + O(q^2 t_m)$. ■

C. Performance

We compare the major computational cost, security, key size, ciphertext size and offline storage of our scheme with those of existing schemes [14], [16] in Table I. We assume that $|G_1| = 160$ bits, $|G_2| = 1024$ bits, $|p| = 160$ bits, $|m| = 160$ bits and $|ID| = 160$. In the "Schemes" column, SL is the original version in [14] that has not the non-repudiation since a receiver can generate the same ciphertext as a sender does. Of course, we can use a proper signature scheme [19], [20] to get

TABLE I
PERFORMANCE COMPARISON

Schemes	Computational cost		Security			Key size (bits)		Ciphertext size (bits)	Offline storage (bits)
	Signcrypt	Unsigncrypt	CCA2	CMA	IS	IBC users	PKI users		
SL [14]	1P	1P	Yes	No	No	320	320	640	0
HWY-I [16]	3M	2M+2P	Yes	Yes	Yes	480	320	960	0
HWY-II [16]	2M	4M	Yes	Yes	Yes	480	320	960	0
Ours	2M (offline)	2M+2P	Yes	Yes	Yes	320	320	640	1824

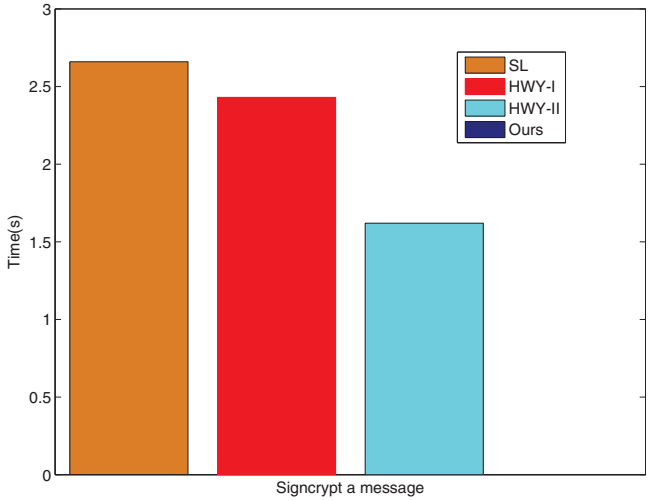

Fig. 3. Time for signcrypting a message.

the non-repudiation property. However, we need another two pairings computation. HWY-I and HWY-II are the first scheme and second scheme, respectively in [16]. For simplicity, we only consider the pairing and point multiplication operations since the two operations take the most running time of the whole algorithm [21]. The other operations are ignored. We denote by M the point multiplication in G_1 and P the pairing computation in the "Computational cost" column. In the "Security" column, CCA2, CMA and IS denotes IND-CCA2, EUF-CMA and insider security, respectively. In the "Key size" column, we consider the sum size of public key and secret key. From Table I, we can see that SL does not satisfy insider security. HWY-I, HWY-II and our scheme satisfy insider security. Our scheme splits the signcryption into two phases: offline phase and online phase. Two point multiplication operations have been precomputed offline. The online phase is very efficient and does not need any pairing and point multiplication operations. That is, our scheme can complete quickly the entire signcryption process when a message is available. Therefore, our scheme is very suitable to provide security solution for sensor nodes. The key size of PKI users in all schemes is 320 bits. For IBC users, SL and our scheme are the same and are 320 bits. HWY-I and HWY-II are 480 bits. For ciphertext length, SL and our scheme are 640 bits. HWY-I and HWY-II are 960 bits. Of course, our scheme needs an offline storage with 1824 bits.

We adopt the experiment in [22] on MICA2 that is equipped with an ATmega128 8-bit processor clocked at 7.3728 MHz, 4 KB RAM and 128 KB ROM. A point multiplication needs 0.81s using an elliptic curve with 160 bits p. According to [23], a pairing computation on MICA2 needs 2.66s. Figure 3 roughly shows the time for signcrypt a message in SL, HWY-I, HWY-II and our scheme. SL, HWY-I and HWY-II need 2.66s, 2.43s and 1.62s, respectively. The time of our scheme is negligible. As compared with SL, HWY-I, HWY-II, our scheme is most suitable for integrating WSNs into the Internet as part of the IoT. For energy consumption, a point multiplication uses 19.1mJ and a pairing uses 62.73mJ [22], [23]. To signcrypt a message, SL, HWY-I and HWY-II roughly uses 62.73mJ, 57.3mJ and 38.2 mJ, respectively. The computational energy consumption of our scheme is negligible. From [24], we know that MICA2 costs 4.12μJ to transmit a bit. For energy cost for communication, SL, HWY-I, HWY-II and our scheme costs 2.64mJ, 3.96mJ, 3.96mJ and 2.64mJ, respectively.

VI. CONCLUSION

In this paper, we proposed a heterogeneous online/offline signcryption scheme. It allows a sensor node in the IBC to send a message to an Internet host in the PKI. Our scheme adopted both online/offline technique and IBC technique to greatly reduce the computational cost of sensor nodes. The scheme setups a secure channel between a sensor node and an Internet host that supports end-to-end confidentiality, integrity, authentication and non-repudiation services. This method provides a new security solution for integrating WSNs into the Internet as part of the IoT.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
[2] R. Roman and J. Lopez, "Integrating wireless sensor networks and the Internet: A security analysis," *Internet Res.*, vol. 19, no. 2, pp. 246–259, 2009.
[3] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 2139. New York, NY, USA: Springer-Verlag, 2001, pp. 213–229.
[4] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost (signature) + cost (encryption)," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 1294. New York, NY, USA: Springer-Verlag, 1997, pp. 165–179.
[5] F. Bao and R. H. Deng, "A signcryption scheme with signature directly verifiable by public key," in *Public Key Cryptography* (Lecture Notes in Computer Science), vol. 1431. New York, NY, USA: Springer-Verlag, 1998, pp. 55–59.
[6] C. Gamage, J. Leiwo, and Y. Zheng, "Encrypted message authentication by firewalls," in *Public Key Cryptography* (Lecture Notes in Computer Science), vol. 1560. New York, NY, USA: Springer-Verlag, 1999, pp. 69–81.

[7] J. Malone-Lee and W. Mao, "Two birds one stone: Signcryption using RSA," in *Topics in Cryptology* (Lecture Notes in Computer Science), vol. 2612. New York, NY, USA: Springer-Verlag, 2003, pp. 211–226.
[8] C. K. Li, G. Yang, D. S. Wong, X. Deng, and S. S. M. Chow, "An efficient signcryption scheme with key privacy and its extension to ring signcryption," *J. Comput. Security*, vol. 18, no. 3, pp. 451–473, 2010.
[9] B. Libert and J. J. Quisquater, "A new identity based signcryption schemes from pairings," in *Proc. IEEE Inf. Theory Workshop*, Paris, France, 2003, pp. 155–158.
[10] S. S. M. Chow, S. M. Yiu, L. C. K. Hui, and K. P. Chow, "Efficient forward and provably secure ID-based signcryption scheme with public verifiability and public ciphertext authenticity," in *Information Security and Cryptology* (Lecture Notes in Computer Science), vol. 2971. New York, NY, USA: Springer-Verlag, 2004, pp. 352–369.
[11] X. Boyen, "Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 2729. New York, NY, USA: Springer-Verlag, 2003, pp. 383–399.
[12] L. Chen and J. Malone-Lee, "Improved identity-based signcryption," in *Public Key Cryptography* (Lecture Notes in Computer Science), vol. 3386. New York, NY, USA: Springer-Verlag, 2005, pp. 362–379.
[13] P. S. L. M. Barreto, B. Libert, N. McCullagh, and J. J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 3788. New York, NY, USA: Springer-Verlag, 2005, pp. 515–532.
[14] Y. Sun and H. Li, "Efficient signcryption between TPKC and IDPKC and its multi-receiver construction," *Sci. China Inf. Sci.*, vol. 53, no. 3, pp. 557–566, Mar. 2010.
[15] J.H. An, Y. Dodis, and T. Rabin, "On the security of joint signature and encryption," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 2332. New York, NY, USA: Springer-Verlag, 2002, pp. 83–107.
[16] Q. Huang, D. S. Wong, and G. Yang, "Heterogeneous signcryption with key privacy," *Comput. J.*, vol. 54, no. 4, pp. 525–536, Apr. 2011.
[17] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–396, Jan. 2000.
[18] D. Boneh and X. Boyen, "Short signatures without random oracles," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 3027. New York, NY, USA: Springer-Verlag, 2004, pp. 56–73.
[19] J. C. Cha and J. H. Cheon, "An identity-based signature from gap Diffie-Hellman groups," in *Public Key Cryptography* (Lecture Notes in Computer Science), vol. 2567. New York, NY, USA: Springer-Verlag, 2003, pp. 18–30.
[20] F. Hess, "Efficient identity based signature schemes based on pairings," in *Selected Areas in Cryptography* (Lecture Notes in Computer Science), vol. 2595. New York, NY, USA: Springer-Verlag, 2003, pp. 310–324.
[21] S. Cui, P. Duan, C.W. Chan, and X. Cheng, "An efficient identity-based signature scheme and its applications," *Int. J. Netw. Security*, vol. 5, no. 1, pp. 89–98, Jul. 2007.
[22] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs," in *Cryptographic Hardware and Embedded Systems* (Lecture Notes in Computer Science), vol. 3156. New York, NY, USA: Springer-Verlag, 2004, pp. 119–132.
[23] P. Szczechowiak, A. Kargl, M. Scott, and M. Collier, "On the application of pairing based cryptography to wireless sensor networks," in *Proc. 2nd ACM Conf. Wireless Netw. Security*, Zurich, Switzerland, 2012, pp. 1–12.
[24] D. Galindo1, R. Roman, and J. Lopez, "On the energy cost of authenticated key agreement in wireless sensor networks," *Wireless Commun. Mobile Comput.*, vol. 12, no. 1, pp. 133–143, Jan. 2012.

Fagen Li received the B.S. degree from the Luoyang Institute of Technology, Luoyang, China, in 2001, the M.S. degree from the Hebei University of Technology, Tianjin, China, in 2004, and the Ph.D. degree in cryptography from Xi'dian University, Xi'an, China, in 2007. He was a Post-Doctoral Fellow with Future University-Hakodate, Hokkaido, Japan, from 2008 to 2009, which is supported by the Japan Society for the Promotion of Science. He was a Research Fellow with the Institute of Mathematics for Industry, Kyushu University, Fukuoka, Japan, from 2010 to 2012. He is currently an Associate Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His current research interests include cryptography and network security. He has published more than 70 papers in the international journals and conferences.

Pan Xiong received the B.S. degree from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2009. She is currently pursuing the Master degree with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. Her current research interests include cryptography and network security.