International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.9 (2016), pp.187-202
http://dx.doi.org/10.14257/ijmue.2016.11.9.20

An Secure Hierarchical Key Agreement Scheme for Wireless Sensor Networks

Kefei Mao*, Jie Chen and Jianwei Liu

School of Electronic and Information Engineering, Beihang University, Beijing, China
**Corresponding Author: Kefei Mao. E-mail:owen.buaa@gmail.com*

Abstract

In Wireless Sensor Networks (WSN), a key agreement scheme is an essential task for secure communications. Recently, Lee and Kim proposed a hierarchical key agreement scheme for the fresh key establishment in WSN. This scheme achieves a secure session key agreement. In this paper, we analyze the security of the scheme and illustrate that their scheme is unconfident against the insider attack in practice. Moreover, it is also difficult to resist the replay attack in this scheme. Then, we proposed a novel scheme inspired by their scheme. The entities of our scheme include a Sink besides the sensor nodes, which interrupts the construction of the insider attack. Our scheme adopts the timestamp mechanism to resist replay attack, which could decrease the storing requirement of the sensor nodes. Thus, it is more practical and realistic. We illustrate that our proposal can provide stronger security than Lee and Kim's scheme.

***Keywords**: Security, Key Agreement Scheme, Wireless Sensor Networks (WSN), Authentication*

1. Introduction

The more wireless sensor technology is being developed, the more benefits are brought to civilian and military requirements by using the Wireless Sensor Networks (WSN). WSN has become a network of integrated sensor technologies, and supports multiple extensible applications, such as emergency response, medical monitoring and battlefield management [1-3]. At the same time, WSN has also attracted various attacks due to the significance of its data [4]. Therefore, how to make the session key agreement securely and efficiently between any two leaf nodes in the open networks becomes a primary security issue [5-7]. The essence of the problem is to implement a hierarchical key agreement scheme [8-10].

In recent years, many security schemes have been proposed for WSN likely environments [6]. In the early days, the public key cryptography (PKC) is the primary solution in WSN as well as other cryptography system [11-14]. Due to the reliability and the credibility of the traditional pair-wise key establishment techniques, the methods are widely cognitive. In particular, the schemes were proposed by using elliptic curve cryptography (ECC) because of the storage and computing cost advantages [15-17]. With the development of identity-based cryptography (IBC) [18] and applications [19, 20], some papers [21-23] have used IBC and pairing-based cryptography for key distribution in WSN.

Recently, Inspired by the Guo *et al.*'s research [8], Kim proposed a hierarchical key agreement protocol applicable to WSN [24]. The scheme is also based on IBC, and claims that it resists the corruption of any sensor nodes in the pyramid. Unfortunately, Lee and Kim found that the scheme in [24] fail to achieve freshness of the session key. Also, Lee and Kim proposed an improved scheme to satisfy the freshness by using the nonce [7] and inherit the security advantages from the paper [24].

In this paper, we show that Lee and Kim's scheme [7] failed to resist the replay attack and the insider attack in practice. We start from constructing a realistic security model. Then, we analyze that the scheme is vulnerable to the replay attack in our security model because the security assumption is too hard to achieve in practice. Moreover, the scheme is hard to resist the insider attack. An adversary, as a cluster head (CH) or a cluster member (CM), could legally acquire the private key, and launch an attack, successfully. In order to resist the attacks, we propose a novel hierarchical key agreement scheme in WSN. Our scheme keeps the quality of key freshness in Lee and Kim's scheme and overcomes the weaknesses of the scheme in our security model.

The remainder of this paper is organized as follows. Section 2 gives the problem characteristics and notations. Section 3 briefly reviews the Lee and Kim's scheme. Section 4 illustrates the drawbacks of their scheme in our security model. Section 5 provides a novel scheme. Section 6 gives a security and performance analysis of our proposed scheme. Finally, the conclusion is presented in Section 7.

2. Preliminaries

In this section, we describe the basic system model in WSN and the mathematical backgrounds in our paper. Basic notations are provided at the end of the section.

2.1. Basic System Model

A typical WSN configuration involves three parties [25], namely a Sink, the Cluster Heads (CH), and the Cluster Members (CM). We assume there are n CHs, namely $\{CH_i\}_{i=1}^{n}$. Every CH whose identity is $\{CH_i\}_{i=1}^{n}$ manages m CMs, namely $\{CM_{ij}\}_{j=1}^{m}$. All CMs communicate with others through their CH and the Sink. Figure 1 illustrates the basic system model of the WSN.

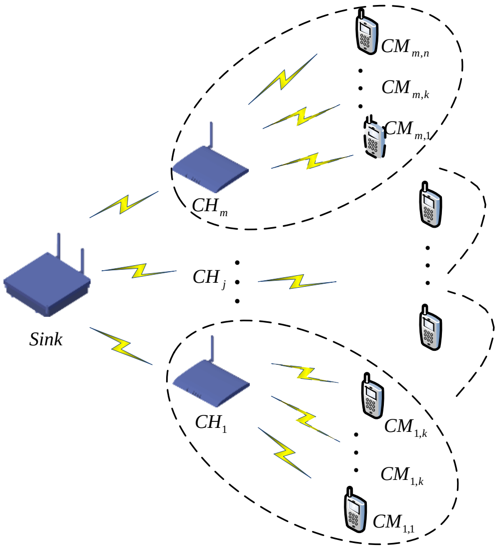


Figure 1. Basic System Model of WSN

2.2. Mathematical Backgrounds

2.2.1. Bilinear Maps

Let G_1 and G_2 be two cyclic groups of prime order p, and Let P be a generator of G_1. The bilinear pairing is a map $\hat{e}: G_1 \times G_1 \to G_2$ with following properties.

Bilinearity. For all $P,Q \in G_1$ and $a,b \in Z_q^*$, $\hat{e}(aP,bQ) = \hat{e}(P,Q)^{ab}$.

Non-degeneracy. $\hat{e}(P,P) \neq 1$.

Computability. For all $P,Q \in G_1$, we can find an efficient algorithm to compute $\hat{e}(P,Q)$.

2.2.2. Discrete Logarithm Problem

Given a randomly chosen $h = aP$ and P in an additive cyclic group G_1, with $a \in Z_q^*$, compute $a \in Z_q^*$. We say that the $(t-\varepsilon)$-DLP assumption holds in G_1 if on t-time algorithm has non-negligible advantage ε in solving the Discrete Logarithm Problem (DLP) in G_1.

2.3. Notations

To provide a quick reference, the basic notations used in this paper are listed in Table 1.

Table 1. Basic Notations

Notations	Descriptions
CH_i	The cluster head i
CM_{ij}	The cluster member node j in the cluster head i
ID_i	The identity of i
AD_i	The amplified identity of i
$H_1(\cdot)$	The hash function, which maps $\{0,1\}^* \to G_1$
$H_2(\cdot)$	The hash function, which maps $G_2 \times G_1^2 \times \{0,1\}^n \to \{0,1\}^n$
$H_3(\cdot)$	The hash function, which maps $G_1^4 \times \{0,1\}^{2n} \to \{0,1\}^n$
T_i	The $i-th$ timestamp

3. Review of the Scheme in Paper

In this section, we briefly review the Lee and Kim's freshness consideration key agreement scheme in paper [7], which consist of two phases: Hierarchical Key Settlement Phase, Session Key Agreement and Secure Communication Phase. It is assumed that each entities shares two groups G_1 and G_2 of prime order p with a bilinear map $\hat{e}: G_1 \times G_1 \to G_2$ and a cryptographic hash function $H:\{0,1\}^* \to G_1$. The basic transmission of the scheme is shown in Figure 2.

3.1. Hierarchical Key Settlement Phase

Step K1. Sink picks three random numbers $s_1, s_2, s_3 \leftarrow Z_q^*$ as the master private keys. Then, Sink computes an amplified identity $AD_S = H(ID_S)$ and a public key $s_1 AD_S$, where ID_S is the real identity of Sink. Then, Sink secure stores the master private key (s_1, s_2, s_3) and the amplified identity AD_S . Finally, Sink sends the data package $\{(s_1 AD_S, s_2, s_3), AD_S\}$ to the CH via a secure way.

Step K2. After CH (ID_{CH_i}) receives the package, it computes $AD_{CH_i} = H(ID_{CH_i})$ and $s_2 AD_{CH_i}$. Then, CH secure stores the private key $(s_1 AD_S, s_2, s_3)$ and the amplified identities (AD_S, AD_{CH_i}) . Finally, CH sends the data package $\{(s_1 AD_S, s_2 AD_{CH_i}, s_3), AD_S, AD_{CH_i}\}$ to its member nodes $\{CM_{ij}\}_{j=1}^{m}$ via a secure way.

Step K3. After the CM ($ID_{CM_{ij}}$) receives the package, it computes $AD_{CM_{ij}} = H(ID_{CM_{ij}})$ and $s_3 AD_{CM_{ij}}$. Then, CM secure stores the private keys $(s_1 AD_S, s_2 AD_{CH_i}, s_3 AD_{CM_{ij}})$ and the amplified identity $(AD_S, AD_{CH_i}, AD_{CM_{ij}})$, respectively.

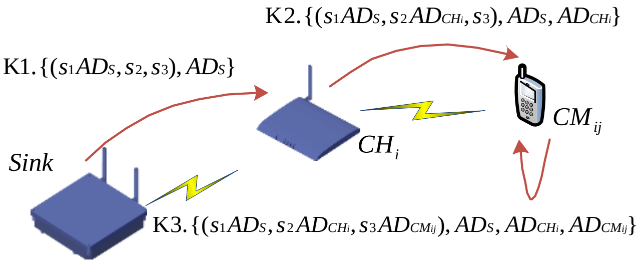

K2. $\{(s_1 AD_S, s_2 AD_{CH_i}, s_3), AD_S, AD_{CH_i}\}$
K1. $\{(s_1 AD_S, s_2, s_3), AD_S\}$
CM_{ij}
CH_i
$Sink$
K3. $\{(s_1 AD_S, s_2 AD_{CH_i}, s_3 AD_{CM_{ij}}), AD_S, AD_{CH_i}, AD_{CM_{ij}}\}$

Figure 2. Hierarchical Key Settlement Phase of the Scheme

3.2. Session Key Agreement and Secure Communication Phase

Two CMs (CM_{ij} and CM_{kl}) compute a session key as follows.

Step C1. The CM (CM_{ij}) chooses a random number r_1, and computes $R_1 = r_1 AD_{CM_{ij}}$. The session key sk is given by the following formula. Here $AD_S{}' = H(ID_S)$, $AD_{CH_k}{}' = H(ID_{CH_k})$ and $AD_{CM_{kl}}{}' = H(ID_{CM_{kl}})$.

$$sk = \hat{e}(s_1 AD_S, AD_S{}') \cdot \hat{e}(s_2 AD_{CH_i}, AD_{CH_k}{}') \cdot \hat{e}(s_3 AD_{CM_{ij}}, AD_{CM_{kl}}{}')^{r_1} \tag{1}$$

Then, CM_{ij} computes the verifier $V_1 = H(sk, R_1)$, and sends the data package $\{R_1, V_1\}$ to the other CM (CM_{kl}).

Step C2. After CM_{kl} receives the package, it computes the session key sk by the following formula, where $AD_S{}' = H(ID_S)$ and $AD_{CH_i}{}' = H(ID_{CH_i})$.

$$sk^* = \hat{e}(s_1 AD_S, AD_S{}') \cdot \hat{e}(s_2 AD_{CH_k}, AD_{CH_i}{}') \cdot \hat{e}(s_3 AD_{CM_{kl}}, R_1) \tag{2}$$

Then, CM_{kl} also computers the verifier $V_1^* = H(sk^*, R_1)$. Only if V_1^* is equal to V_1 , CM_{kl} assures the correctness of sk^* .

Step C3. After CM_{kl} assures the session key sk^* , it encrypts the plain data $DATA$ to get the encrypt data $EDATA$ by using the key. Then, CM_{kl} computes the verifier $V_2 = H(sk^*, EDATA)$, and sends the data package $\{EDATA, V_2\}$ to CM_{ij} .

Step C4. After CM_{ij} receives the package, it also computes the verifier $V_2^* = H(sk, EDATA)$. Only if V_2^* is equal to V_2 , CM_{ij} assures the correctness of the encrypt data $EDATA$ and the session key sk .

4. Cryptanalysis of Lee and Kim's Scheme

In this section, we propose a more practical security model. Moreover, we point out Lee and Kim's scheme [7] suffers two attacks in our model. The details are explained in the following section.

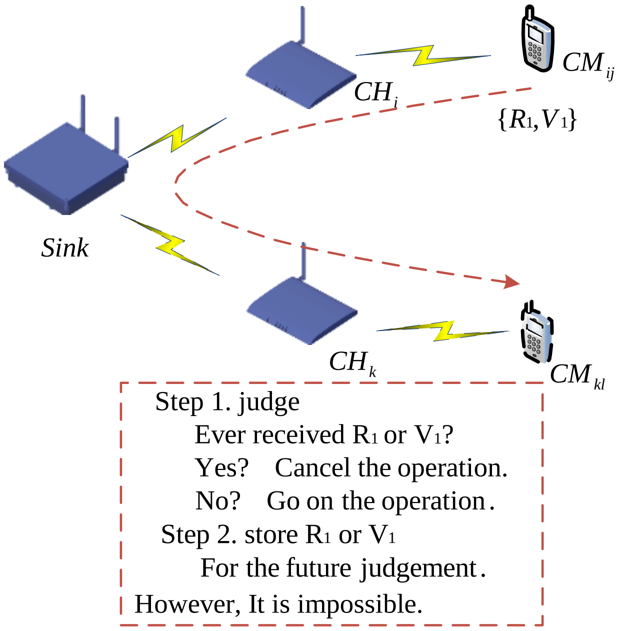
4.1. Our Security Model

The authors of the paper proposed a hierarchical key agreement scheme in WSN. The authors assume the insider's secret values are reliable and security, such as the private key of the CHs and the CMs. The adversary can only implement the active and passive attack by controlling the insecurity channel. However, it is more useful to carefully consider the corruption risk of the internal nodes. In practice, the WSN is easily attacked from inside. For example, the adversary can capture some nodes, and then use the side channel attacks to get the secret values of the nodes. Therefore, we propose a more practical security model.

Inspired by the papers, we enhance the ability of the adversary in our model. Same as others, the adversary can totally control over the communication channel. More precisely, the adversary may eavesdrop, intercept, modify, replay or inject the communication between any entities in the WSN (*e.g.* the channel between CM_{ij} and CM_{kl}). Furthermore, the adversary can also corrupt some the secret parameters from the CHs and the CMs except those of the entities who are attacking by the adversary. This state imitates that the insider is corrupted in the WSN. Under above conditions, the scheme should have the ability to resist various kinds of attacks, and achieve the following security goals. (1) **Key security**. Any adversary cannot obtain the current session key. (2) **Known-key security**. Any adversary compromised a shared session key cannot obtain the current session key.

4.2. Weaknesses

According to Lee and Kim's paper [7], we find that the scheme has the following disadvantages in our security model.



Step 1. judge
Ever received R_1 or V_1?
Yes? Cancel the operation.
No? Go on the operation.
Step 2. store R_1 or V_1
For the future judgement.
However, It is impossible.

Figure 3. Cannot Resist the Replay Attack

4.2.1. It Cannot Resist the Replay Attack

The scheme will has security problems if the adversary launches the replay attack in real world. As we know, the nonce R_1 is used to prevent the replay attack in the paper. However, the scheme cannot resist replay attack in practice. More precisely, it is essential that the CM should judge whether the nonce R_1 and verifier V_1 have ever received. In order to make a judgment, the CMs should store the received nonce R_1, interminably. However, it is obviously impossible to sensor nodes because of the limited resources. Thus, it cannot resist the replay attack to use the method in Lee and Kim's scheme. The more details are described as follows.

As is shown in Figure 3, when the adversary has captured the data packet $\{R_1, V_1\}$ from CM_{ij} to CM_{kl}, he/she can store it. Then he/she can replay the nonce R_1 and the verifier V_1 whenever he/she wants. If the victim CM (CM_{kl}) receives the data packet $\{R_1, V_1\}$, it is obviously that it can believe the package from the real node CM_{ij} because V_1 is equal to $H(sk^*, R_1)$. In order to avoid the attack, CM_{kl} should be stored the nonce R_1 received before and judge whether the current nonce R_1 is a replay attack. However, there is no enough resource to support the storing and querying nonce in CM. Thus, Lee and Kim's scheme cannot resist the replay attack in practice.

4.2.2. It Cannot Resist the Insider Attack (the Cluster Head Disguise)

We now demonstrate that the Lee and Kim's scheme is vulnerable to the one kind of insider attack. The adversary who has registered as a legal CH (CH_A) as shown in Figure 4. According to the definition of our security model, the adversary can intercept the communication data in the WSN because the wireless channel is openness. The adversary who disguises a CH successfully attacks a communication between CM_{ij} and CM_{kl} as follows.

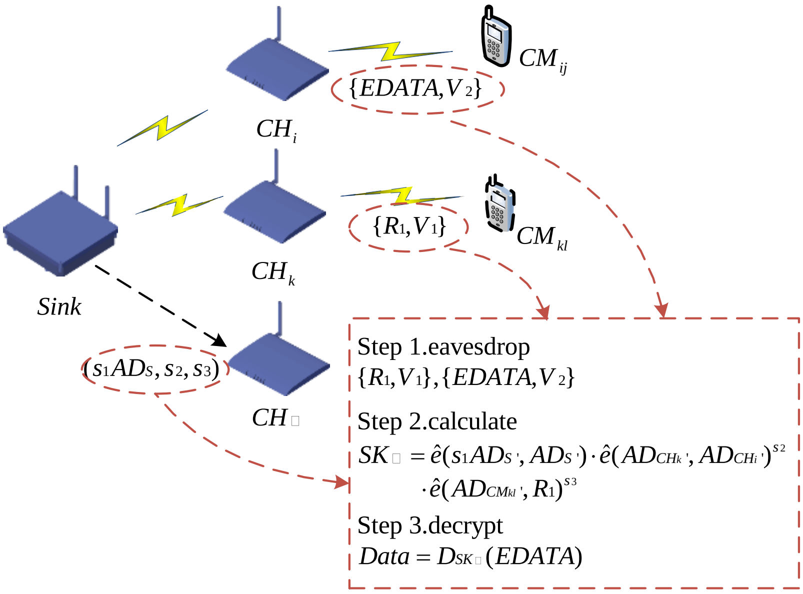


Figure 4. Cannot Against Insider Attack

Step H1. Assume that CH_A is an adversary who has registered as a CH, and then it can legally receive a private key set (s_1AD_S, s_2, s_3) from the Sink (**Step K1**).

Step H2. Suppose CM_{ij} and CM_{kl} are victim CMs who send the data packages through their CHs (CH_i and CH_k) and Sink. When CM_{ij} runs the **Step C1**, the adversary can intercept the data package $\{R_1, V_1\}$ because channel is unsecure between CM_{ij} and CM_{kl}.

Step H3. When CM_{kl} sends back the encrypt data $EDATA$ to CM_{ij} at the Step C3, the adversary can also intercept the data package $\{EDATA, V_2\}$.

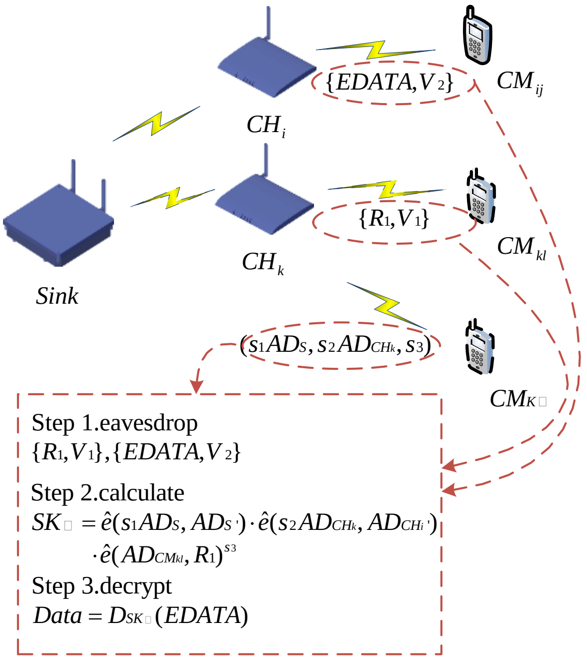
Step H4. After the above steps, the adversary CH_A can compute the key sk_A by the following formula. Here, the values $AD_S' = H(ID_S)$, $AD_{CH_k}' = H(ID_{CH_k})$, $AD_{CH_i}' = H(ID_{CH_i})$ and $AD_{CM_{kl}}' = H(ID_{CM_{kl}})$.

$$\begin{aligned} sk_A &= \hat{e}(s_1AD_S, AD_S') \cdot \hat{e}(AD_{CH_k}', AD_{CH_i}')^{s_2} \cdot \hat{e}(AD_{CM_{kl}}', R_1)^{s_3} \\ &= \hat{e}(s_1AD_S, AD_S') \cdot \hat{e}(s_2AD_{CH_k}, AD_{CH_i}') \cdot \hat{e}(s_3AD_{CM_{kl}}, R_1) \\ &= sk \end{aligned} \tag{3}$$

Step H5. The adversary CH_A can decrypt $EDATA$ to obtain the encrypt data between CM_{ij} and CM_{kl} by using the session key sk_A because the keys sk_A , sk and sk^* are equal. Thus, Lee and Kim's scheme [7] cannot resist the insider attack when the adversary disguises a CH.

4.2.3. It Cannot Resist the Insider Attack (the Cluster Member Disguise)

Moreover, there is another insider attack as follows. The adversary who has registered as a legal CM (CM_{kA}) as shown in Figure 5. The adversary has the same CH (CH_k) with the one victim CM_{kl}. Therefore, the adversary is a neighbor node of the victim. The adversary launches an attack to the communication between CM_{ij} and CM_{kl} as follows.



Step 1.eavesdrop
$\{R_1, V_1\}, \{EDATA, V_2\}$

Step 2.calculate
$$SK_{\square} = \hat{e}(s_1AD_S, AD_S') \cdot \hat{e}(s_2AD_{CH_k}, AD_{CH_i}') \cdot \hat{e}(AD_{CM_{kl}}, R_1)^{s_3}$$

Step 3.decrypt
$$Data = D_{SK_{\square}}(EDATA)$$

Figure 5. Cannot Against Insider Attack

Step M1. Assume that CH_A is an adversary who has registered as a CM in a CH (CH_k), and then it legally receives a private key set $(s_1AD_S, s_2AD_{CH_k}, s_3)$ from CH_k (**Step K2**).

Step M2. Suppose two CMs (CM_{ij} and CM_{kl}) are victim nodes. It is same above describe that the adversary can intercept the data $\{R_1, V_1\}$ and $\{EDATA, V_2\}$.

Step M3. After the above steps, The adversary CH_A can compute the session key as follows. Here, the values $AD_S{}' = H(ID_S)$, $AD_{CH_i}{}' = H(ID_{CH_i})$ and $AD_{CM_{kl}}{}' = H(ID_{CM_{kl}})$.

$$\begin{aligned} sk_A &= \hat{e}(s_1AD_S, AD_S{}') \cdot \hat{e}(s_2AD_{CH_k}, AD_{CH_i}{}') \cdot \hat{e}(AD_{CM_{kl}}{}', R_1)^{s_3} \\ &= \hat{e}(s_1AD_S, AD_S{}') \cdot \hat{e}(s_2AD_{CH_k}, AD_{CH_i}{}') \cdot \hat{e}(s_3AD_{CM_{kl}}, R_1) \\ &= sk \end{aligned} \tag{4}$$

Step M4. It is obviously that the keys sk_A, sk and sk^* are equal. Therefore, the adversary CH_A can decrypt $EDATA$ to obtain the plain text $DATA$ using the key sk_A. Thus, Lee and Kim's scheme cannot resist the insider attack when the adversary disguises a neighbor CM with one of two victims.

5. Our Enhanced Scheme

In this section, we propose an improved scheme based on identity-based cryptography, which can overcome the weaknesses of Lee and Kim's scheme in Section 4. Our scheme construction is inspired by the papers [7-8, 19]. Our scheme consists of three operational phase: System Parameter Generation Phase, Initial Phase, Authentication and Transmission Phase. The details of our scheme are described as follows.

5.1. System Parameter Generation Phase

Similar to the papers [7-8], we use the identity-based encryption (IBE) and the hierarchical structure in our scheme. More specifically, the identity of a sensor node is his/her public key, and his/her private key generates by the Private Key Generator (PKG) in Sink. The parent nodes calculate and release the private key set for their descendants.

Step G1. Sink generates two addition groups G_1 and G_2 of prime order p with a bilinear map $\hat{e}: G_1 \times G_1 \to G_2$. Then, Sink chooses three hash functions satisfied $H_1: \{0,1\}^* \to G_1$, $H_2: G_2 \times G_1^2 \times \{0,1\}^n \to \{0,1\}^n$ and $H_3: G_1^4 \times \{0,1\}^{2n} \to \{0,1\}^n$. After that, it randomly chooses $3+m$ random numbers $\{s_1, s_2, s_3, sr_i \leftarrow Z_q^* \,|\, i \in (1, \cdots, m)\}$ as a master key of Sink and a random generator $P_0 \in G_1$. Here, m is the number of CHs in WSN. Finally, Sink secure stores the master key $MK = \{s_1, s_2, s_3, sr_i \leftarrow Z_q^* \,|\, i \in (1, \cdots, m)\}$ and publishes the public parameters $params = \{q, G_1, G_2, P_0, \hat{e}, H_1, H_2, H_3, s_1P_0, s_2P_0, s_3P_0, sr_iP_0 \,|\, i \in (1, \cdots, m)\}$.

5.2. Initial Phase

Step I1. When a CH (CH_i) with identity ID_{CH_i} wants to register in WSN. It sends his identity ID_{CH_i} to Sink. Sink computes the amplified identities (AD_S, AD_{CH_i}) and the private key $(s_1sr_i\,AD_s, s_2sr_i\,AD_{CH_i}, s_3sr_i)$ .Here, the values $AD_s = H_1(ID_s)$ and $AD_{CH_i} = H_1(ID_{CH_i})$. Then, the Sink sends the data package $\{(s_1sr_i\,AD_s, s_2sr_i\,AD_{CH_i}, s_3sr_i), (AD_s, AD_{CH_i})\}$ to CH_i via a secure channel. Finally, CH_i keeps the received information in its secure memory.

Step I2. When a CM (CM_{ij}) with identity ID_{CH_i} wants to register in WSN. It sends his identity $ID_{CM_{ij}}$ to its CH (CH_i). CH_i computes $AD_{CM_{ij}} = H(ID_{CM_{ij}})$ and $s_3 sr_i AD_{CM_{ij}}$. Then, the CH (CH_i) sends the private key set $\{(s_1 sr_i AD_s, s_3 sr_i AD_{CM_{ij}}), (AD_s, AD_{CM_{ij}})\}$ to its CM (CM_{ij}), securely. Finally, CM_{ij} keeps the received information in its secure memory.

5.3. Authentication and Key Agreement Phase

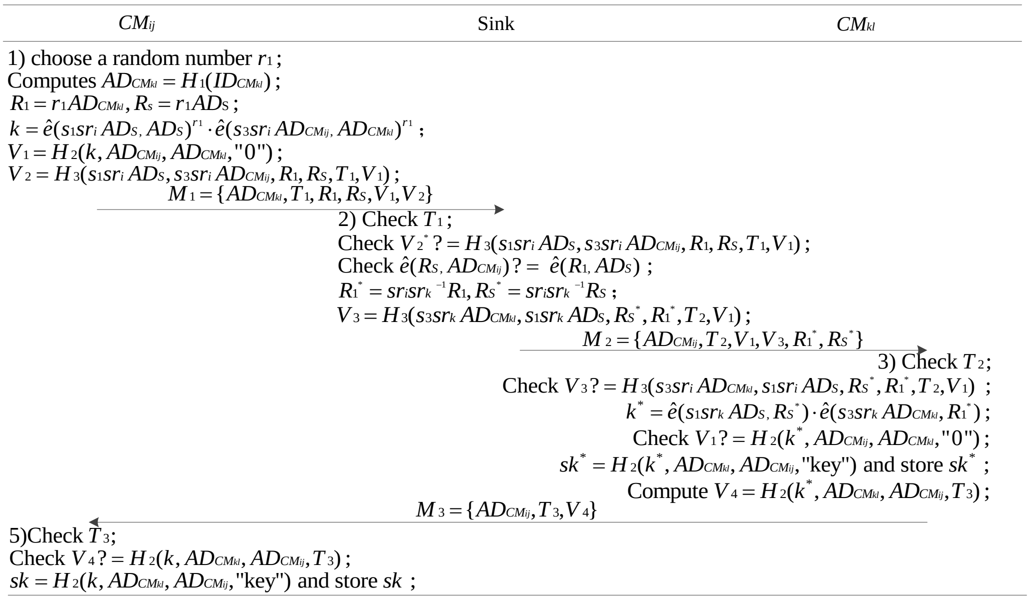
When CM_{ij} wants to establish a session key sk with CM_{kl} by the helping of Sink, the following steps are executed among CM_{ij}, CM_{kl} and Sink in Figure 6.

Step A1. CM_{ij} chooses a random number r_1. Then it computes an amplified identity $AD_{CM_{kl}} = H_1(ID_{CM_{kl}})$, $R_1 = r_1 AD_{CM_{ij}}$ and $R_s = r_1 AD_S$. The temporary key k is calculated by using the above parameters and the private key set $(s_1 sr_i AD_s, s_3 sr_i AD_{CM_{ij}})$.

$$k = \hat{e}(s_1 sr_i AD_{S}, AD_S)^{r_1} \cdot \hat{e}(s_3 sr_i AD_{CM_{ij}}, AD_{CM_{kl}})^{r_1} \tag{5}$$

Then, CM_{ij} computes the verification values $V_2 = H_2(s_1 sr_i AD_s, s_3 sr_i AD_{CM_{ij}}, R_1, R_S, T_1, V_1)$ and $V_1 = H_2(sk, AD_{CM_{ij}}, AD_{CM_{kl}}, "0")$ .Then it sends $M_1 = \{AD_{CM_{kl}}, T_1, R_1, R_S, V_1, V_2\}$ to Sink. Here, and T_1 is a timestamp.

Step A2. After received the data package M_1, Sink verifies the timestamp T_1 whether it is within the valid time for communication. If it is invalid, the program terminates. Otherwise, Sink judges whether the received V_2 is equal to $H_2(s_1 sr_i AD_s, s_3 sr_i AD_{CM_{ij}}, R_1, R_S, T_1, V_1)$ and computes whether $\hat{e}(R_S, AD_{CM_{ij}}) = \hat{e}(R_1, AD_S)$. Only if they are all equal, Sink convince the two values R_1 and R_2, then it computes $R_1^* = sr_i sr_k^{-1} R_1$ and $R_S^* = sr_i sr_k^{-1} R_S$. Otherwise, it ends the processing. Sink computes the value $V_3 = H_3(s_1 sr_k AD_s, s_3 sr_k AD_{CM_{kl}}, R_S^*, R_1^*, T_2, V_1)$, and sends $M_2 = \{AD_{CM_{ij}}, T_2, V_1, V_3, R_1^*, R_S^*\}$ to CM_{kl}.



CM_{ij} | Sink | CM_{kl}

1) choose a random number r_1;
Computes $AD_{CM_{kl}} = H_1(ID_{CM_{kl}})$;
$R_1 = r_1 AD_{CM_{kl}}, R_s = r_1 AD_S$;
$k = \hat{e}(s_1 sr_i AD_{S}, AD_S)^{r_1} \cdot \hat{e}(s_3 sr_i AD_{CM_{ij}}, AD_{CM_{kl}})^{r_1}$;
$V_1 = H_2(k, AD_{CM_{ij}}, AD_{CM_{kl}}, "0")$;
$V_2 = H_3(s_1 sr_i AD_S, s_3 sr_i AD_{CM_{ij}}, R_1, R_S, T_1, V_1)$;

$M_1 = \{AD_{CM_{kl}}, T_1, R_1, R_S, V_1, V_2\}$

2) Check T_1;
Check $V_2^* ?= H_3(s_1 sr_i AD_S, s_3 sr_i AD_{CM_{ij}}, R_1, R_S, T_1, V_1)$;
Check $\hat{e}(R_S, AD_{CM_{ij}}) ?= \hat{e}(R_1, AD_S)$;
$R_1^* = sr_i sr_k^{-1} R_1, R_S^* = sr_i sr_k^{-1} R_S$;
$V_3 = H_3(s_3 sr_k AD_{CM_{kl}}, s_1 sr_k AD_S, R_S^*, R_1^*, T_2, V_1)$;

$M_2 = \{AD_{CM_{ij}}, T_2, V_1, V_3, R_1^*, R_S^*\}$

3) Check T_2;
Check $V_3 ?= H_3(s_3 sr_i AD_{CM_{kl}}, s_1 sr_i AD_S, R_S^*, R_1^*, T_2, V_1)$;
$k^* = \hat{e}(s_1 sr_k AD_{S}, R_S^*) \cdot \hat{e}(s_3 sr_k AD_{CM_{kl}}, R_1^*)$;
Check $V_1 ?= H_2(k^*, AD_{CM_{ij}}, AD_{CM_{kl}}, "0")$;
$sk^* = H_2(k^*, AD_{CM_{kl}}, AD_{CM_{ij}}, "key")$ and store sk^*;
Compute $V_4 = H_2(k^*, AD_{CM_{kl}}, AD_{CM_{ij}}, T_3)$;

$M_3 = \{AD_{CM_{ij}}, T_3, V_4\}$

5)Check T_3;
Check $V_4 ?= H_2(k, AD_{CM_{kl}}, AD_{CM_{ij}}, T_3)$;
$sk = H_2(k, AD_{CM_{kl}}, AD_{CM_{ij}}, "key")$ and store sk;

Figure 6. Authentication and Key Agreement Phase in our Scheme

Step A3. After received the data package M_2, CM_{kl} checks the validity of the timestamp T_2. If it has grown stale, CM_{kl} ends the session. Otherwise, it judges whether the verifier V_3 is equal to the hash value $H_3(s_1 sr_k AD_s, s_3 sr_k AD_{CM_{kl}}, R_S^*, R_1^*, T_2, V_1)$. Only if they are equal, CM_{kl} computes the temporary key k^* as follows.

$$k^* = \hat{e}(s_1 sr_k AD_S, R_S^*) \cdot \hat{e}(s_3 sr_k AD_{CM_{kl}}, R_1^*). \tag{6}$$

Otherwise, it ends the processing. Then, CM_{kl} checks whether the received value V_1 is equal to the hash value $H_2(k^*, AD_{CM_{ij}}, AD_{CM_{kl}}, "0")$. Only if they are equal, CM_{kl} authenticates CM_{ij} and assures the temporary key k^*. Then, CM_{kl} computes $sk^* = H_2(k^*, AD_{CM_{kl}}, AD_{CM_{ij}}, "key")$, and a value $V_4 = H_2(k^*, AD_{CM_{kl}}, AD_{CM_{ij}}, T_3)$. Finally, it sends $M_3 = \{AD_{CM_{kl}}, T_3, V_4\}$ to CM_{ij}. Here, T_3 is a current timestamp.

Step A4. After received the data package M_3, CM_{ij} checks the validity of the timestamp T_3. If it is invalid, it terminates the processing. Otherwise, CM_{ij} computes a hash value $V_4^* = H_2(k, AD_{CM_{kl}}, AD_{CM_{ij}}, T_3)$. Only if $V_3 = V_3^*$, CM_{ij} computes the session key sk is equal to $H_2(k, AD_{CM_{kl}}, AD_{CM_{ij}}, "key")$ with CM_{kl}.

6. Correctness and Security Analysis

In this section, we present the correctness of our improved scheme. Then, we analyze our enhancement scheme regarding security and overcome the weaknesses analyzed in Section 3.

6.1. Correctness

We verify the correctness of key agreement in our scheme as follows.

$$\begin{aligned} k &= \hat{e}(s_1 sr_i AD_S, AD_S)^{r_1} \cdot \hat{e}(s_3 sr_i AD_{CM_{ij}}, AD_{CM_{kl}})^{r_1} \\ &= \hat{e}(s_1 sr_i AD_S, r_1 AD_S) \cdot \hat{e}(s_3 sr_i AD_{CM_{kl}}, r_1 AD_{CM_{ij}}) \\ &= \hat{e}(s_1 sr_i sr_k AD_S, sr_k^{-1} R_S) \cdot \hat{e}(s_3 sr_i sr_k AD_{CM_{kl}}, sr_k^{-1} R_1) \\ &= \hat{e}(s_1 sr_k AD_S, sr_i sr_k^{-1} R_S) \cdot \hat{e}(s_3 sr_k AD_{CM_{kl}}, sr_i sr_k^{-1} R_1) \\ &= \hat{e}(s_1 sr_k AD_S, R_S^*) \cdot \hat{e}(s_3 sr_k AD_{CM_{kl}}, R_1^*) \\ &= k^* \end{aligned} \tag{7}$$

The session key, computed by CM_{ij} and CM_{kl}, $sk = H_2(k, AD_{CM_{kl}}, AD_{CM_{ij}}, "key")$ and $sk^* = H_2(k^*, AD_{CM_{kl}}, AD_{CM_{ij}}, "key")$ are equal.

6.2. Security Analysis

6.2.1. Authentication

CM_{ij} sends the data packages $M_1 = \{AD_{CM_{kl}}, T_1, R_1, R_S, V_1, V_2\}$ to Sink. Here, the value $V_2 = H_2(s_1 sr_i AD_s, s_3 sr_i AD_{CM_{ij}}, R_1, R_S, T_1, V_1)$. Without the private keys $s_1 sr_i AD_s$ and $s_3 sr_i AD_{CM_i}$, the adversary cannot generate a legal data package M_1 due to the nature of the hash function. Therefore, Sink could authenticate CM_{ij} by checking the correctness of the value V_2. Similarly, CM_{kl} could authenticate Sink by judging the accuracy of the V_3. A sensor node CM_{kl} sends the message $M_3 = \{AD_{CM_{kl}}, T_3, V_4\}$ to another node CM_{ij}.

Here, the value $V_4 = H_2(k^*, AD_{CM_{kl}}, AD_{CM_{ij}}, T_3)$. Since only CM_{ij} and CM_{kl} can compute the temporary key k^*. Therefore, CM_{ij} could authenticate CM_{kl} through checking the correctness of the V_4. It is essential to note that the scheme should ensure adequate entropy of the keys in the System Parameter Generation Phase in order to prevent the offline guessing attack.

6.2.2. Security Key Establishment

Key security is a critical requirement for a key agreement scheme. Assume that the adversary succeed get the session key $sk = H_2(k, AD_{CM_{kl}}, AD_{CM_{ij}}, \text{"key"})$ described in Section 5. Since the hash function has the one-way characteristic, the adversary should obtain the temporary k or k^* for computing the key sk. However, the temporary key k is established by CM_{ij} in the Step A1. Since the adversary does not know the private values of CM_{ij}, he/she cannot compute the temporary key $k = \hat{e}(s_1 sr_i AD_S, AD_S)^{r_1} \cdot \hat{e}(s_3 sr_i AD_{CM_{ij}}, AD_{CM_{kl}})^{r_1}$. Moreover, In the Step A3, the temporary k^* is established by CM_{kl}. Since the adversary does not know the values of $s_1 sr_k AD_S$ and $s_1 sr_k AD_{CM_{kl}}$ together, he/she cannot calculate the temporary key k^* by formula $k^* = \hat{e}(s_1 sr_k AD_S, R_S^*) \cdot \hat{e}(s_3 sr_k AD_{CM_{kl}}, R_1^*)$.

Our scheme also achieves perfect forward secrecy. Since the numbers of R_1 and R_S are randomly computed in every communication, the values of R_1^* and R_S^* are freshly in every stage. Therefore, all the historical session keys will still secure even if the long-term private keys are disclosed in future. Base on the one-way function and the freshness of R_1 and R_S, the adversary cannot obtain any information about the future session key even if the current session key sk or sk^* is compromised.

6.2.3. Security Against Replay Attacks

Our scheme can resist replay attack because we used the timestamps. If an adversary replays the eavesdropped message, then the sensor nodes will detect the attack when examining the current time. During the authentication and key agreement phase, when Sink receives a data package $M_1 = \{AD_{CM_{kl}}, T_1, R_1, R_S, V_1, V_2\}$, it verifies the timestamp T_1 with the current time. If the message is a replay message, then Sink will find it. If adversary change the timestamp T_1 in data package M_1, however, it cannot know the values of $s_1 sr_i AD_S$, $s_1 sr_i AD_{CM_{ij}}$, Sink will find the replay when it check the value V_2. Similarly, when CM_{kl} receives the data package $M_2 = \{AD_{CM_{ij}}, T_2, V_1, V_3, R_1^*, R_S^*\}$, it judges the timestamp T_2 to against the message replay. Similarly, when CM_{ij} receives the data package $M_3 = \{AD_{CM_{kl}}, T_3, V_4\}$, it resist the message replay by judging the timestamp T_3 and the session key sk. It is should admit that the synchronization problem is the main vulnerability to timestamp, However, the situation has been released with the development of the sensor technologies [26] such as using the linear regression to achieve long-term synchronization and using the time base signal in the Global Position System (GPS). The timestamp is the economic and effective means to resist the replay attack in WSN.

6.2.4. Security Against Insider Attacks

In the proposed scheme, Sink computes and distributes different private keys $(s_1sr_i AD_s, s_2sr_i AD_{CH_i}, s_3sr_i)$ for various CHs. Thus, the adversary CH cannot get the information including the private keys of the other CHs. Furthermore, if we assume that an adversary CM who has registered as CM_{ij} can obtain $s_1sr_i AD_s$ and $s_3sr_i AD_{CM_i}$ from CH_i. However, he/she cannot calculate s_3sr_i from $s_3sr_i AD_{CMij}$ except he/she can solve the DLP assumption. Thus, our scheme destructs the attack conditions as **Step H1** and **Step M1**. As a result, the scheme prevents the adversary to generate the temporary key k and the session key sk in sequence. Therefore, the proposal could withstand the insider attack. It should be noted that Sink has an enough computing power as a data concentrator node. Although our proposal increases the computation cost of Sink, the advice is feasible to WSN.

6.2.5. Security Against Impersonation Attacks

To impersonate CM_{ij} to Sink, an adversary has to generate a legal message $M_1 = \{AD_{CM_{kl}}, T_1, R_1, R_S, V_1, V_2\}$, where $V_2 = H_2(s_1sr_i AD_s, s_3sr_i AD_{CM_{ij}}, R_1, R_S, T_1, V_1)$. The adversary cannot compute V_2 because the values $s_1sr_i AD_s$ and $s_3sr_i AD_{CM_i}$ are private key of CM_{ij}. Therefore, the adversary cannot impersonate CM_{ij} to Sink. To attack Sink to CM_{kl}, an adversary has to generate a data $M_2 = \{AD_{CM_{ij}}, T_2, V_1, V_3, R_1^*, R_S^*\}$, where $V_3 = H_3(s_1sr_k AD_s, s_3sr_k AD_{CM_{kl}}, R_S^*, R_1^*, T_2, V_1)$. Without the knowledge of $s_1sr_k AD_s$ and $s_3sr_k AD_{CM_{kl}}$, an adversary cannot generate V_3. Thus, the adversary cannot impersonate Sink to CM_{kl}. To impersonate CM_{kl} to CM_{ij}, an adversary has to generate a legal message $M_3 = \{AD_{CM_{kl}}, T_3, V_4\}$ .Here the authentication value $V_4 = H_2(k^*, AD_{CM_{kl}}, AD_{CM_{ij}}, T_3)$. The adversary cannot compute V_4 since he/she does not know the temporary key k^*. Therefore, the adversary cannot impersonate CM_{ij} to CM_{kl}.

6.3. Security Comparison

We compared the proposed scheme with Lee and Kim's scheme and Guo *et al.*'s scheme in terms of security properties. Similar to the Kim's work, P1, P2, P3, P4, P5 and P6 denote the key agreement, the authentication, the impersonation attack, the key freshness, the replay attack and the insider attack, separately. According to Table 2, we can conclude that the proposed scheme delivers a higher level of security compared to related works.

Table 2. The Security Comparison

	Ours	Lee and Kim's [7]	Kim's[24]	Guo *et al.*'s [8]
P1	Provide	Provide	Provide	Provide
P2	Provide	Provide	Provide	Provide
P3	Resistance	Resistance	Resistance	Resistance
P4	Resistance	Provide	N/A	N/A
P5	Resistance	N/A	N/A	N/A
P6	Resistance	N/A	N/A	N/A

7. Conclusion

We discuss several security weaknesses in the paper under a new security model. There is an adversary who can legally get the private parameter of CM or CH and control the channel between two CMs in the attack model. After that, we point out the two weaknesses of Lee and Kim's scheme [7] in our model. Then we propose an enhanced scheme based on the bilinear pair to overcome these weaknesses. Our proposal increases its security strength by using the Sink nodes. The forthcoming work is to achieve a formal proof method of the key agreement scheme in WSN. Then we achieve it in the real sensor hardware environments.

Acknowledgements

The authors would like to thank the anonymous reviews for their valuable comments. This paper is supported by the National Key Basic Research Program (973 program) through project 2012CB315905, by the Natural Science Foundation through projects 61370190, 61272501, 61173154, 61402029 and 61003214, and by the Beijing Natural Science Foundation through projects 4132056 and 4122041. The authors also thank China Aeronautical Radio Electronics Research Institute for its funding.

References

[1] 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, **(2002)**, pp. 521-534.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A survey on sensor networks", IEEE Communications magazine, vol. 40, no. 8, **(2002)**, pp. 102-114.

[3] L. Y. Li and C. D. Liu, "An improved algorithm of LEACH routing protocol in wireless sensor networks", Journal of Harbin University of Science and Technology, vol. 20, no. 2, **(2015)**, pp. 75-79.

[4] A. Perrig, J. Stankovic and D. Wagner, "Security in wireless sensor networks", Communication of the ACM, vol. 47, no. 6, **(2004)**, pp. 53-57.

[5] R. D. Pietro, L. V. Mancini and S. Jajodia, "Providing secrecy in key management protocols for large wireless sensors networks", Ad Hoc Networks, vol. 1, no. 4, **(2003)**, pp. 455-468.

[6] C. Y. Chen and H. C. Chao, "A survey of key distribution in wireless sensor networks", Security and Communication Networks, vol. 7, no. 12, **(2014)**, pp. 2495-2508.

[7] S. Lee and H. Kim, "Freshness Consideration of Hierarchical Key Agreement Protocol in WSNs", International Journal of Security and Its Applications, vol. 8, no. 1, **(2014)**, pp. 81-91.

[8] H. Guo, Y. Mu, Z. Li and X. Zhang, "An efficient and non-interactive hierarchical key agreement protocol", Computers & Security, vol. 30, no. 1, **(2011)**, pp. 28-34.

[9] X. He, M. Niedermeier and H. De Meer, "Dynamic key management in wireless sensor networks: A survey", Journal of Network and Computer Applications, vol. 36, no. 2, **(2013)**, pp. 611-622.

[10] O. K. Sahingoz, "Large scale wireless sensor networks with multi-level dynamic key management scheme", Journal of Systems Architecture, vol. 59, no. 9, **(2013)**, pp. 801-807.

[11] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn and P. Kruus, "TinyPK: securing sensor networks with public key technology", Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks, Washington, DC, USA, **(2004)**, pp. 56-64.

[12] G. Gaubatz, J. Kaps and B. Sunar, "Public key cryptography in sensor networks-revisited", in Security in Ad-hoc and Sensor Networks, Edited C. Castelluccia, H. Hartenstein, C. Paar and D. Westhoff, Springer, Heidelberg, vol. 3313, **(2005)**, pp. 2-18.

[13] W. Du, R. Wang and P. Ning, "An efficient scheme for authenticating public keys in sensor networks", Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing, Urbana-Champaign, IL, USA, **(2005)**, pp. 58-67.

[14] C. P. Gouvêa, L. B. Oliveira and J. López, "Efficient software implementation of public-key cryptography on sensor networks using the MSP430X microcontroller", Journal of Cryptographic Engineering, vol. 2, no. 1, **(2012)**, pp. 19-29.

[15] W. Shi and P. Gong, "A new user authentication protocol for wireless sensor networks using elliptic curves cryptography", International Journal of Distributed Sensor Networks, vol. 2013, no. 730831, **(2013)**, pp. 1-7.

[16] A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks", International Conference on Information Processing in Sensor Networks 2008, St. Louis, MO, USA, **(2008)**, pp. 245-256.

[17] N. Gura, A. Patel, A. Wander, H. Eberle and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs", Proceedings of the 6th International Workshop Cambridge, MA, USA, **(2004)**, pp. 119-132.

[18] A. Shamir, "Identity-based cryptosystems and signature schemes", in Advances in Cryptology, Edited G. R. Blakley and D. Chaum, Springer, Heidelberg, vol. 196, **(1985)**, pp. 47-53.

[19] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing", SIAM Journal on Computing, vol. 32, no. 3, **(2003)**, pp. 586-615.

[20] L. Chen and C. Kudla, "Identity based authenticated key agreement protocols from pairings", Proceedings on the 16th Computer Security Foundations Workshop, Pacific Grove, CA, USA, **(2003)**, pp. 219-233.

[21] L. B. Oliveira, D. F. Aranha, C. P. Gouvêa, M. Scott, D. F. Câmara, J. López and R. Dahab, "TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks", Computer Communications, vol. 34, no. 3, **(2011)**, pp. 485-493.

[22] S. Sung and J. Ryou, "ID-based sensor node authentication for multi-layer sensor networks", Journal of Communications and Networks, vol. 16, no. 4, **(2014)**, pp. 363-370.

[23] L. B. Oliveira and R. Dahab. "Pairing-based cryptography for sensor networks", Proceedings of the 5th IEEE International Symposium on Network Computing and Applications, Cambridge, MA, USA, **(2006)**.

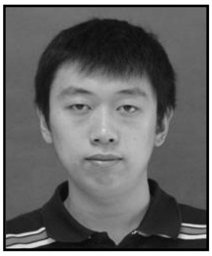
[24] H. Kim, "Efficient and non-interactive hierarchical key agreement in WSNs", International Journal of Security and Its Applications, vol. 7, no. 1, **(2013)**, pp. 159-170.

[25] H. Lu, J. Li and M. Guizani, "Secure and efficient data transmission for cluster-based wireless sensor networks", IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 3, **(2014)**, pp. 750-761.

[26] J. J. P. Solano and S. F. Castell, "Adaptive time window linear regression algorithm for accurate time synchronization in wireless sensor networks", Ad Hoc Networks, vol. 24, no. A, **(2015)**, pp. 92-108.

Authors

Kefei Mao, He received the M.Sc. degree in computer science from the University of Chinese Academy of Sciences, Beijing, China, in 2007. He is currently pursuing the doctoral degree with Beihang University, Beijing, China. His current research interests include optimization algorithms design and analysis, computational complexity, and network schemes.



Jie Chen, He received the M.Sc. degree in Electronic engineering from the Northwest Polytechnic University, Xi'an, China, in 2010. He is currently pursuing the doctoral degree with Beihang University, Beijing, China. His current research interests include Ad hoc network security, Future Network Design and Software Define Network security.

Jianwei Liu, He received the B.S. and M.S. degrees in Electronic and Information from Shandong University, Shandong, China in 1985 and 1988, respectively. He received his Ph.D. degree in Communication and Electronic System from Xidian University, Shanxi, China in 1998. Now, he is a Professor of Electronic and Information Engineering at Beihang University, Beijing, China. His current research interests include wireless communication network, coding theory, and information security.