Hindawi Publishing Corporation
Journal of Sensors
Volume 2015, Article ID 534657, 14 pages
http://dx.doi.org/10.1155/2015/534657

Research Article

A Bilinear Pairing-Based Dynamic Key Management and Authentication for Wireless Sensor Networks

Chin-Ling Chen,[1] Tzay-Farn Shih,[1] Yu-Ting Tsai,[1] and De-Kui Li[2]

[1]*Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan*
[2]*Department of Logistics Management, Wuhan Technology and Business University, Wuhan, Hubei 430065, China*

Correspondence should be addressed to Tzay-Farn Shih; tfshih@cyut.edu.tw

Received 3 June 2014; Accepted 22 October 2014

Academic Editor: Gyuhae Park

In recent years, wireless sensor networks have been used in a variety of environments; a wireless network infrastructure, established to communicate and exchange information in a monitoring area, has also been applied in different environments. However, for sensitive applications, security is the paramount issue. In this paper, we propose using bilinear pairing to design dynamic key management and authentication scheme of the hierarchical sensor network. We use the dynamic key management and the pairing-based cryptography (PBC) to establish the session key and the hash message authentication code (HMAC) to support the mutual authentication between the sensors and the base station. In addition, we also embed the capability of the Global Positioning System (GPS) to cluster nodes to find the best path of the sensor network. The proposed scheme can also provide the requisite security of the dynamic key management, mutual authentication, and session key protection. Our scheme can defend against impersonation attack, replay attack, wormhole attack, and message manipulation attack.

1. Introduction

In recent years, wireless sensor networks have been used in a variety of environments; a wireless network infrastructure, established to communicate and exchange information in a monitoring area, has also been applied in different environments, including disaster relief operations, seismic data collecting, monitoring wildlife, and collecting battlefield information.

Due to their small size, the sensors can be spatially scattered to form an ad hoc network. The sensors have an inherent limitation. The wireless sensor network requires an appropriate encryption or decryption system to protect the collected information [1]. The high cost of an encryption/decryption mechanism (e.g., Diffie and Hellman key management [2] or Rivest et al. encryption [3]) is unsuitable for use in a wireless sensor network.

In addition, the topology of the network environment is another important issue. The hierarchical predistribution protocol [4] allows some of the cluster nodes to aggregate the events of the sensor nodes to communicate with the base station. The hierarchical predistribution protocol includes several cluster nodes, sensor nodes, and base station; the most common hierarchical networks are two-level, and the two classes of sensor are sensor node and cluster node. The advantage of this scheme is the easy management of the data aggregation [5–7]. The process of aggregating the data from multiple nodes involves eliminating redundant transmission and providing fused data to the base station. It is also considered as an effectual technique for wireless sensor networks to save energy [8]. The most popular data aggregation algorithms are cluster-based data aggregation algorithms, in which the nodes are grouped into clusters: each cluster consists of a cluster node and some sensors; each sensor transmits data to its cluster node; and each cluster node aggregates the collected data; it then transmits the fused data to the base station.

The key management scheme is divided into four types: the Random Key Predistribution Protocol (RKP), the Group-Based Key Predistribution Protocol (GKP), the Hierarchical Key Predistribution Protocol (HKP), and the Pairing-Based Protocol (PBC). In 2003, Chan et al. proposed a Random Key Predistribution scheme [9]. Since each node randomly picks keys from a large key pool such that any two sensor

nodes share at least one common key, ensuring adequate storage space and the range of the network is a challenge. The PIKE scheme [10] addressed the problem of high density deployment requirements in RKP. But in this scheme the session key is segmented into many key fragments. Therefore, the combination of the session key is complex. However, the PIKE solved the storage problem of RKP. Cheng and Agrawal proposed an improved key distribution mechanism [11]. The IKDM established a session key which used the exchange information between sensors; it can easily generate a session key by the polynomial function. In recent years, the pairing-based cryptography [12], TinyPBC, is a tiny pairing-based protocol and its computation cost is lower than other corresponding bilinear pairing-based schemes. The pairing-based mechanism was used in the sensor network to accomplish the key management of the sensor's session key. It can use the sensors' identity for sensors to send data to each other via the sensor network. After the identity exchange, the sensors key can easily compute the session key via the bilinear pairing. In such design, the security can also be enhanced.

In this paper, we propose using bilinear pairing to design dynamic key management and authentication scheme of the hierarchical sensor network. We use the dynamic key management mechanism [13, 14] and the pairing-based cryptography (PBC) [12, 15, 16] to establish the session key. We also use the hash message authentication code (HMAC) [15, 17] to offer mutual authentication between the sensors and the base station. Moreover, we also involve the capability of the Global Positioning System (GPS) [18, 19] to cluster nodes, in order to find the best path of the sensor network.

The remainder of this paper is organized as follows. The preliminaries are presented in Section 2. The proposed scheme is described in Section 3. The security analysis of our scheme is given in Section 4. And the discussions are offered in Section 5. Finally, conclusions are presented in the last section.

2. Preliminaries

2.1. Sensor Network Architecture. Categories of sensor networks significantly affect key establishment design [4]. The relative capabilities of different sensors are divided into the following two classes:

(1) homogeneity: all sensors have the same capabilities;

(2) heterogeneity: there is an inherent hierarchy of sensors with respect to their capabilities (with fewer sensors at higher, more "powerful" levels). The most common hierarchical networks are two-level, where there are two classes of sensors.

We choose the hierarchical sensors network's model, and the architecture is described as follows: a small number high class sensors (cluster node), large number low class sensors (sensor node), and a sink node (base station). High class sensors have more powerful ability, they have been equipped with tamper-resistant hardware and GPS capability, the cluster node with powerful ability can plan routing table and achieve more security of sensor network, and the low class sensors have not been equipped with tamper-resistant hardware and GPS capability.

2.2. Bilinear Pairing. The bilinear map can be constructed on elliptic curves. Each operation for computing $e(P, Q)$ is a pairing operation [8]. Let G be a cyclic additive group, and let G_T be a cyclic multiplicative group. Both groups G and G_T have the same prime order q. Groups G and G_T are called bilinear groups. The security of the bilinear pairing-based scheme relies on the difficulty of the Discrete Logarithm Problem (DLP); that is, given the point $Q = aP$, no efficient algorithm exists to obtain a given P and Q. The mapping $e : G \times G \rightarrow G_T$ is called a bilinear map if it satisfies the following properties:

(1) bilinear:

$\forall P, Q, R \in G, \forall a, b \in Z$,

$e(Q, P + R) = e(P + R, Q) = e(P, Q) \cdot e(R, Q)$,

$e(aP, bP) = e(P, bP)^a = e(aP, P)^b = e(P, P)^{ab}$,

(2) nondegenerate:

$P, Q \in G$ exists such that $e(P, Q) \neq 1_{G_T}$,

(3) computable:

an efficient algorithm exists to compute $e(P, Q)$ for any $P, Q \in G$.

2.3. Hash Massage Authentication Code (HMAC). We combine the message authentication code [20, 21] and the bilinear pairing key to accomplish the hash-based message authentication code (HMAC); this is a specific construction for computing a message authentication code (MAC) using a cryptographic hash function in combination with a secret key. Both data integrity and authenticity of a message can be achieved by using a hash-based message authentication code in such a technique. We note HMAC (i.e., $H_K(\cdot)$ is a HMAC which signifies a one-way hash function with pairing key K).

2.4. Pairing-Based Cryptography (PBC). Since pairing-based cryptography (PBC), based on the identity-based cryptography (IBC) [22, 23], is used in many environments of cryptographic protocols and applications [12], the IBC has some drawbacks; this method needs a private key generator (PKG); it is a trusted entity in charge of generating and escrowing user's private keys. In wireless sensor networks, if the sensors need to be deployed in an unattended environment, a sensor node should be a PKG, and this is difficult in a wireless sensor network. If we can easily generate a session key via a simple mechanism, it can reduce the complexity. PBC technology does not need a PKG and the sensors can authenticate themselves in the wireless sensor network. Therefore, the PBC is the best technology for key management.

3. The Proposed Scheme

In this paper, we propose a bilinear pairing-based scheme to design a dynamic key management for wireless sensor network. We first introduce the proposed protocol architecture as in Figure 1.

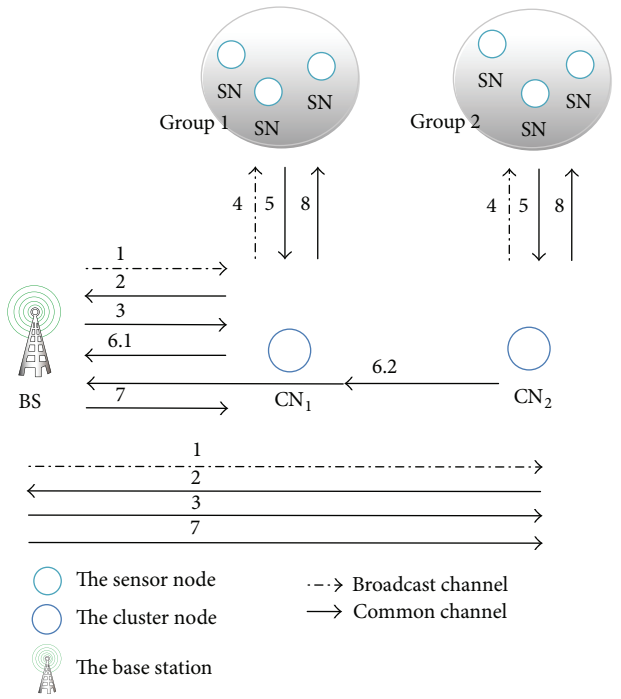

Figure 1: The architecture of wireless sensor network.

(1) Base station broadcasts the starting message to cluster nodes.

(2) Cluster nodes respond the message authentication code to the base station.

(3) After authentication, the base station sends a response message to allow cluster nodes to rule its group members of the sensor nodes.

(4) Cluster node broadcasts the request message to find the members from the neighboring sensor nodes.

(5) Sensor nodes reply the request and respond the message authentication code to the cluster node.

(6) In order to get the sensor nodes' session key, if the cluster node can transmit to the base station, enter into Step 6.1; else if the cluster node needs to transmit the collected information via the next neighboring, enter into Step 6.2.

(7) After authentication, the base station sends the corresponding session key of sensor nodes to the cluster nodes.

(8) After receiving the session keys, the cluster nodes can verify the message authentication code from Step 5. After that, the cluster nodes send the updated identities to the sensor nodes.

3.1. Initialization Phase. In this phase, the base station computes the parameters to predistribute into the sensor nodes and the cluster node. The overview of the initialization phase is shown in Figure 2.

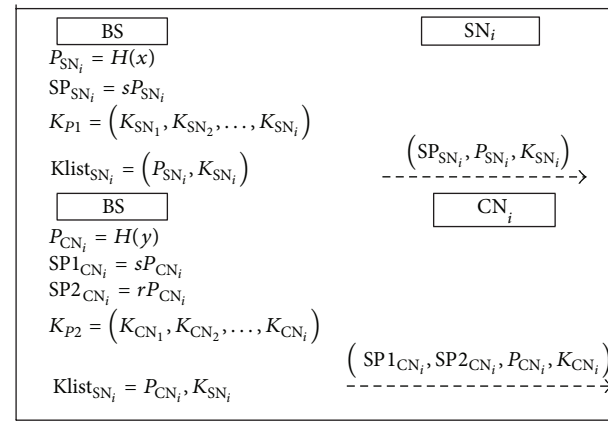

Figure 2: The overview of the initialization.

Step 1. First, the base station selects a random number x and computes the sensor node identity P_{SN_i}:

$$P_{SN_i} = H(x). \tag{1}$$

Then, the base station randomly selects a secret parameter s and uses the secret parameter s and sensor node identity P_{SN_i} to compute the secret parameter $SP1_{SN_i}$:

$$SP1_{SN_i} = sP_{SN_i}. \tag{2}$$

The base station randomly computes a key pool K_{P1}, where $K_{P1} = (K_{SN_1}, K_{SN_2}, \ldots, K_{SN_i})$, and distributes a session key K_{SN_i} to the ith sensor node. It then stores the sensor node identity P_{SN_i} and the K_{SN_i} in the key list $Klist_{SN_i}$:

$$Klist_{SN_i} = (P_{SN_i}, K_{SN_i}). \tag{3}$$

After that, the base station sends the parameters ($SP1_{SN_i}$, P_{SN_i}, K_{SN_i}) to the corresponding sensor node.

Step 2. The base station selects a random number y and computes the cluster node identity P_{CN_i}:

$$P_{CN_i} = H(y). \tag{4}$$

Then the base station randomly selects a secret parameter r and uses the random secret parameters (s, r) to compute the secret parameters ($SP1_{CN_i}$, $SP2_{CN_i}$), respectively:

$$\begin{aligned} SP1_{CN_i} &= sP_{CN_i} \\ SP2_{CN_i} &= rP_{CN_i}. \end{aligned} \tag{5}$$

The base station randomly computes a key pool K_{P2}, where $K_{P2} = (K_{CN_1}, K_{CN_2}, \ldots, K_{CN_i})$, and distributes a session key K_{CN_i} to the ith cluster node. It then stores the cluster node identity P_{CN_i} and the K_{CN_i} in the key list $Klist_{CN_i}$:

$$Klist_{CN_i} = (P_{CN_i}, K_{CN_i}). \tag{6}$$

The base station sends the parameters ($SP1_{CN_i}$, $SP2_{CN_i}$, P_{CN_i}, K_{CN_i}) to the cluster node.

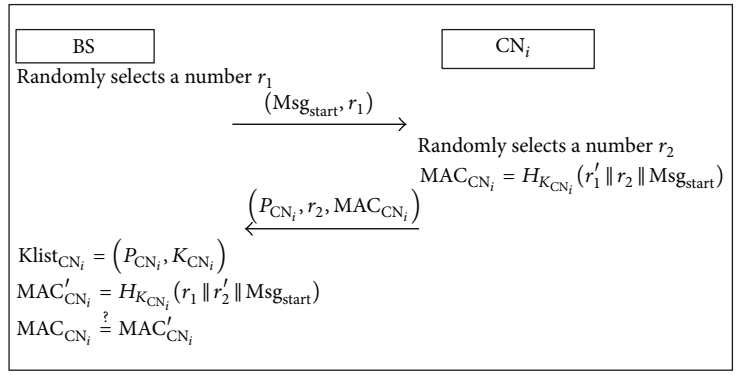

Figure 3: The overview of the starting cluster node.

3.2. Location-Based Routing Determination Phase

3.2.1. The Starting Cluster Node Process. After the sensors are deployed, we must start the cluster node and get the path routing. In Figure 3, we authenticate the cluster node to confirm the legality of the cluster node. Next, the cluster node can rely on the location-based routing to find the best routing path.

Step 1. First, the base station selects a random number r_1 and broadcasts the message $(\mathrm{Msg}_{\mathrm{start}}, r_1)$ to the sensor network.

Step 2. When the cluster node CN_i receives the message, it can select a random number r_2 and compute the message authentication code with key K_{CN_i}:

$$\mathrm{MAC}_{\mathrm{CN}_i} = H_{K_{\mathrm{CN}_i}}\left(r_1' \| r_2 \| \mathrm{Msg}_{\mathrm{start}}\right). \quad (7)$$

The cluster node CN_i then sends the message $(P_{\mathrm{CN}_i}, r_2, \mathrm{MAC}_{\mathrm{CN}_i})$ to the base station.

Step 3. Upon receiving the message, the base station can use identity P_{CN_i} to find key K_{CN_i} from the $K\mathrm{list}_{\mathrm{CN}_i}$:

$$K\mathrm{list}_{\mathrm{CN}_i} = \left(P_{\mathrm{CN}_i}, K_{\mathrm{CN}_i}\right). \quad (8)$$

It then computes the message authentication code $\mathrm{MAC}'_{\mathrm{CN}_i}$

$$\mathrm{MAC}'_{\mathrm{CN}_i} = H_{K_{\mathrm{CN}_i}}\left(r_1 \| r_2' \| \mathrm{Msg}_{\mathrm{start}}\right) \quad (9)$$

and checks if it is equal to $\mathrm{MAC}_{\mathrm{CN}_i}$:

$$\mathrm{MAC}_{\mathrm{CN}_i} \stackrel{?}{=} \mathrm{MAC}'_{\mathrm{CN}_i}. \quad (10)$$

3.3. Location-Based Routing Phase. The cluster nodes can establish the best route on the basis of receiving the broadcast location message in a monitoring area.

Step 1. After the initialization phase, the sensor nodes and cluster nodes store the operating parameters and then distribute the associated messages within their monitoring environment.

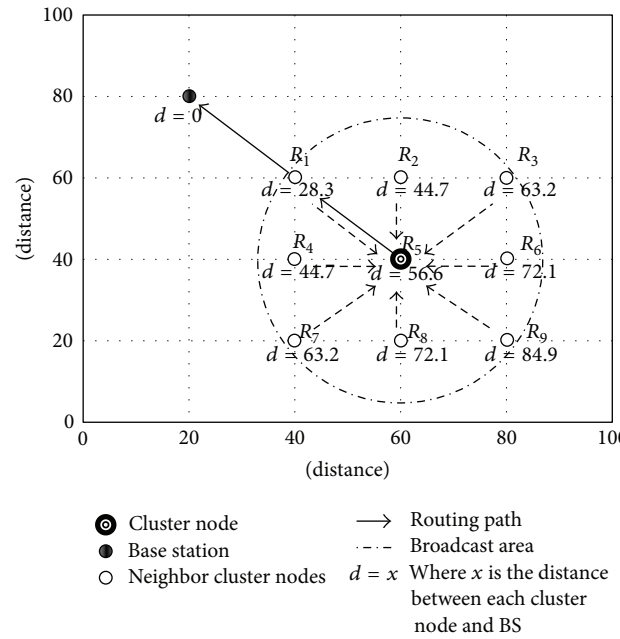

Figure 4: Each cluster node broadcasts its location to its neighbor cluster nodes.

Step 2. The base station broadcasts the starting message $\mathrm{Msg}_{\mathrm{start}}$ to the cluster nodes.

Step 3. Upon receiving the starting message, the cluster node (equipped with a GPS receiver) broadcasts the message $\mathrm{Msg}_{\mathrm{location}}$ concerning its location to the neighbor cluster nodes.

Step 4. After receiving the message $\mathrm{Msg}_{\mathrm{location}}$, the cluster nodes know the location of the source of the neighboring cluster such that it can transmit the monitoring data to the cluster node which is the nearest node to the base station.

For example, in Figure 4, cluster nodes R_1, R_2, R_3, R_4, R_5, R_6, R_7, and R_8 can receive the nearest distance message to the base station from the neighbor cluster nodes R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, and R_9. It can compare the received location

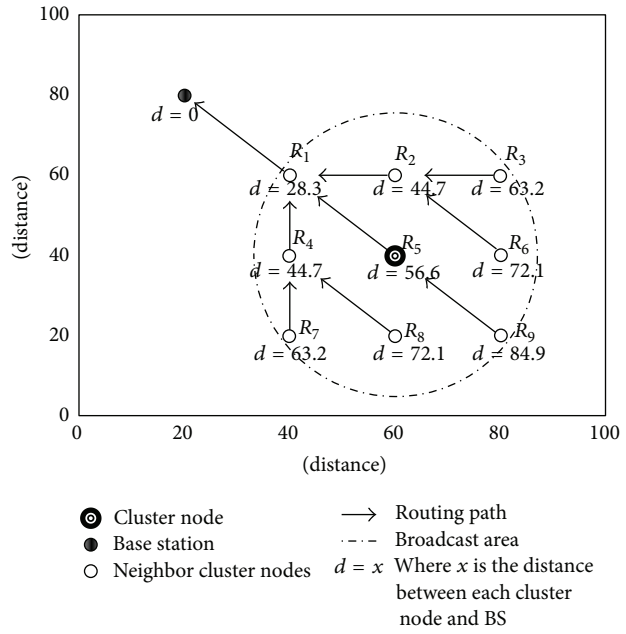

Figure 5: Establishing the best routing overview.

messages to select the nearest node from the base station and establish the multihop routing path to the cluster node R_1. The cluster node R_1 will be used to relay communications to the base station, so the best path of the cluster node R_5 will be established as follows: $R_5 \rightarrow R_1 \rightarrow$ BS. On the basis of the shortest distance between the cluster node and the base station, each cluster node will establish the best routing path.

In Figure 5, the cluster node R_9 can determine that the neighbor cluster node on the best path is R_5, and the cluster node R_5 can determine the R_1 and BS, respectively. The best path for the cluster node R_9 can be established as follows: $R_9 \rightarrow R_5 \rightarrow R_1 \rightarrow$ BS. In the same way, the cluster node R_3 can determine the best path: $R_3 \rightarrow R_2 \rightarrow R_1 \rightarrow$ BS. Every pair of nodes along the resulting multihop path can establish a pairwise key for encrypted communication in such a way that each intermediate node can relay data towards the base station in a totally secure way. Location awareness also increases the probability that the geographically closest node pairs establish a pairwise session key along the best path to the BS, with the effect of saving energy on all the nodes involved in multihop routing.

3.4. The Authentication Phase of the Cluster Node and the Sensor Node. The base station sends the broadcast message $\mathrm{Msg}_{\mathrm{start}}$ to the cluster nodes; when the cluster node receives the message, it will broadcast the request message $\mathrm{Msg}_{\mathrm{req}}$ to find the neighboring sensor node to join the group. The overview of the authentication phase of the cluster node and the sensor node is shown in Figure 6.

Step 1. When the cluster node CN_i receives the starting message $\mathrm{Msg}_{\mathrm{start}}$, the cluster node CN_i selects a nonce n_1 and sends $(P_{\mathrm{CN}_i}, n_1, \mathrm{Msg}_{\mathrm{req}})$ to the neighboring sensor nodes.

Step 2. Upon receiving the message, the sensor node SN_i selects a nonce n_2 and uses $(n_1 \| n_2 \| \mathrm{Msg}_{\mathrm{rep}})$ to compute the message authentication code MAC_1:

$$\mathrm{MAC}_1 = H_{K_{\mathrm{SN}_i}}\left(n_1' \| n_2 \| \mathrm{Msg}_{\mathrm{rep}}\right). \tag{11}$$

The sensor node SN_i sends $(P_{\mathrm{SN}_i}, n_2, \mathrm{Msg}_{\mathrm{rep}}, \mathrm{MAC}_1)$ to the ith cluster node CN_i.

Step 3. The cluster node CN_i adds the sensor node's identity P_{SN_i} into the identity list $\mathrm{SNID}_{\mathrm{list}}$:

$$\mathrm{SNID}_{\mathrm{list}} = \left(P_{\mathrm{SN}_1}, P_{\mathrm{SN}_2}, \ldots, P_{\mathrm{SN}_i}\right). \tag{12}$$

It then sends the cluster node's identity and the sensor node's identity list $\mathrm{SNID}_{\mathrm{list}}$ to the base station. If the cluster node is the nearest base station, then it directly enters into Section 3.4, the Authentication Phase of the Base Station and the Cluster Node. Otherwise, the cluster node needs to transmit the collected information via the next neighboring cluster node and enters into Section 3.5, the Authentication Phase of the Cluster Node and the Cluster Node.

After the authentication and obtaining the session key of the sensor node, the cluster node CN_i computes the message authentication code MAC_1' and checks if it is equal to MAC_1 or not:

$$\begin{aligned} \mathrm{MAC}_1' &= H_{K_{\mathrm{SN}_i}}\left(n_1 \left\| n_2' \right\| \mathrm{Msg}_{\mathrm{rep}}\right) \\ \mathrm{MAC}_1' &\stackrel{?}{=} \mathrm{MAC}_1. \end{aligned} \tag{13}$$

Then the cluster node CN_i selects a random integer number α, computes the new parameter $(\alpha P_{\mathrm{CN}_i}, \alpha P_{\mathrm{SN}_i})$, and updates into $(P_{\mathrm{CN}_i}, P_{\mathrm{SN}_i})$, respectively:

$$\begin{aligned} P_{\mathrm{CN}_i} &= \alpha P_{\mathrm{CN}_i} \\ P_{\mathrm{SN}_i} &= \alpha P_{\mathrm{SN}_i}. \end{aligned} \tag{14}$$

The cluster node CN_i uses the session key to encrypt the new parameter P_{SN_i} of the sensor node SN_i:

$$C_1 = E_{K_{\mathrm{SN}_i}}\left(P_{\mathrm{SN}_i}\right). \tag{15}$$

The cluster node CN_i randomly selects a nonce n_3 and computes the message authentication code MAC_2:

$$\mathrm{MAC}_2 = H_{K_{\mathrm{SN}_i}}\left(n_3 \left\| n_2' \right\| C_1\right). \tag{16}$$

Then the cluster node CN_i sends the message $(P_{\mathrm{CN}_i}, C_1, n_3, \mathrm{MAC}_2)$ to the sensor nodes.

Step 4. The sensor node SN_i computes the message authentication code MAC_2' and checks if it is equal to MAC_2:

$$\begin{aligned} \mathrm{MAC}_2' &= H_{K_{\mathrm{SN}_i}}\left(n_3' \| n_2 \| C_1'\right) \\ \mathrm{MAC}_2' &\stackrel{?}{=} \mathrm{MAC}_2. \end{aligned} \tag{17}$$

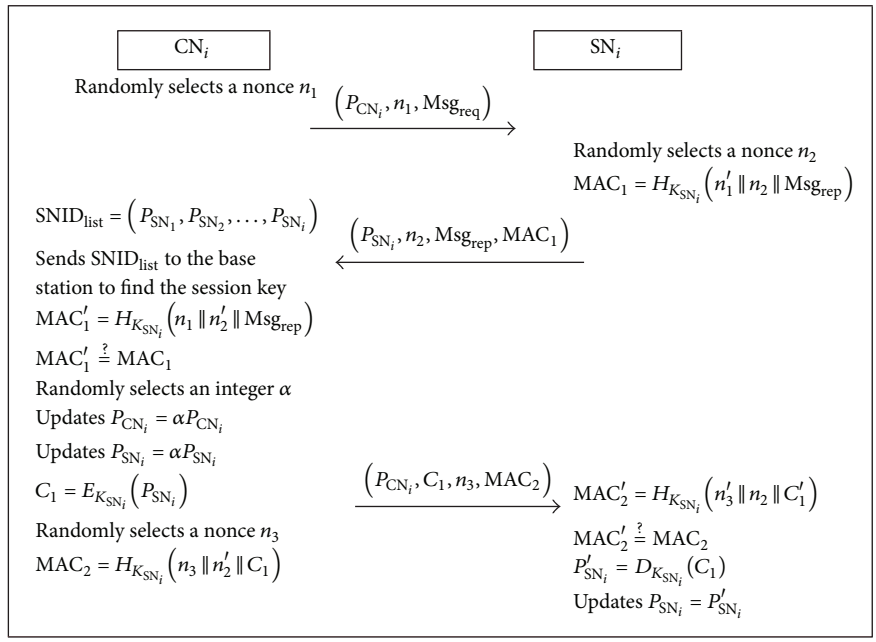

Figure 6: The overview of the authentication phase of the cluster node and the sensor node.

After authentication, the sensor node SN_i decrypts the encrypted message C_1:

$$P_{SN_i}' = D_{K_{SN_i}}(C_1). \tag{18}$$

Then the sensor node updates parameter P_{SN_i}:

$$P_{SN_i} = P_{SN_i}'. \tag{19}$$

3.5. The Authentication Phase of the Base Station and the Cluster Node. In this phase, the cluster node sends the message to the base station to find the corresponding sensor node's session key. The overview of the authentication phase of the base station and the cluster node is shown in Figure 7.

Step 1. First, the cluster node CN_i collects the sensor nodes identity P_{SN_i} into the identity list $SNID_{list}$:

$$SNID_{list} = (P_{SN_1}, P_{SN_2}, \ldots, P_{SN_i}). \tag{20}$$

Then the cluster node CN_i uses the pairing function to compute the pairing session key K_{CN_i-BS}:

$$K_{CN_i-BS} = e(SP2_{CN_i}, P_{BS}). \tag{21}$$

It computes the message authentication code MAC_{CN_i-BS}:

$$MAC_{CN_i-BS} = H_{K_{CN-BS}}(P_{CN_i} \| n_{CN_i} \| SNID_{list}). \tag{22}$$

Then the cluster node CN_i sends the message (P_{CN_i}, n_{CN_i}, $SNID_{list}$, MAC_{CN_i-BS}) to the base station.

Step 2. After receiving, the base station uses the pairing function to compute the pairing session key K_{BS-CN_i}:

$$K_{BS-CN_i} = e(rP_{BS}, P_{CN_i}). \tag{23}$$

The base station computes the message authentication code MAC'_{CN_i-BS}:

$$MAC'_{CN_i-BS} = H_{K_{BS-CN_i}}(P'_{CN_i} \| n'_{CN_i} \| SNID'_{list}). \tag{24}$$

It checks if it is equal to MAC_{CN_i-BS}:

$$MAC'_{CN_i-BS} \stackrel{?}{=} MAC_{CN_i-BS}. \tag{25}$$

After authentication, the base station uses the identity list $SNID_{list}$ to find the corresponding session key K_{SN_i}, makes the key list $Klist_{SN_i}$, and enters it into SNK_{CN_i}:

$$\begin{aligned} Klist_{SN_i} &= (P_{SN_i}, K_{SN_i}) \\ SNK_{CN_i} &= (Klist_{SN_1}, Klist_{SN_2}, \ldots, Klist_{SN_i}). \end{aligned} \tag{26}$$

The base station randomly selects a nonce n_{BS} and computes the message authentication code MAC_{BS-CN_i}:

$$MAC_{BS-CN_i} = H_{K_{BS-CN_i}}(n_{BS} \| n'_{CN_i} \| P_{BS}). \tag{27}$$

Then, the base station uses the pairing session key to encrypt the sensor node's session key list SNK_{CN_i}:

$$C_2 = E_{K_{BS-CN_i}}(SNK_{CN_i}). \tag{28}$$

The base station sends the message (P_{BS}, n_{BS}, C_2, MAC_{BS-CN_i}) to the corresponding cluster node CN_i.

Step 3. When the cluster node CN_i receives the message, it computes the message authentication code MAC'_{BS-CN_i}:

$$MAC'_{BS-CN_i} = H_{K_{CN_i-BS}}(n'_{BS} \| n_{CN_i} \| P'_{BS}). \tag{29}$$

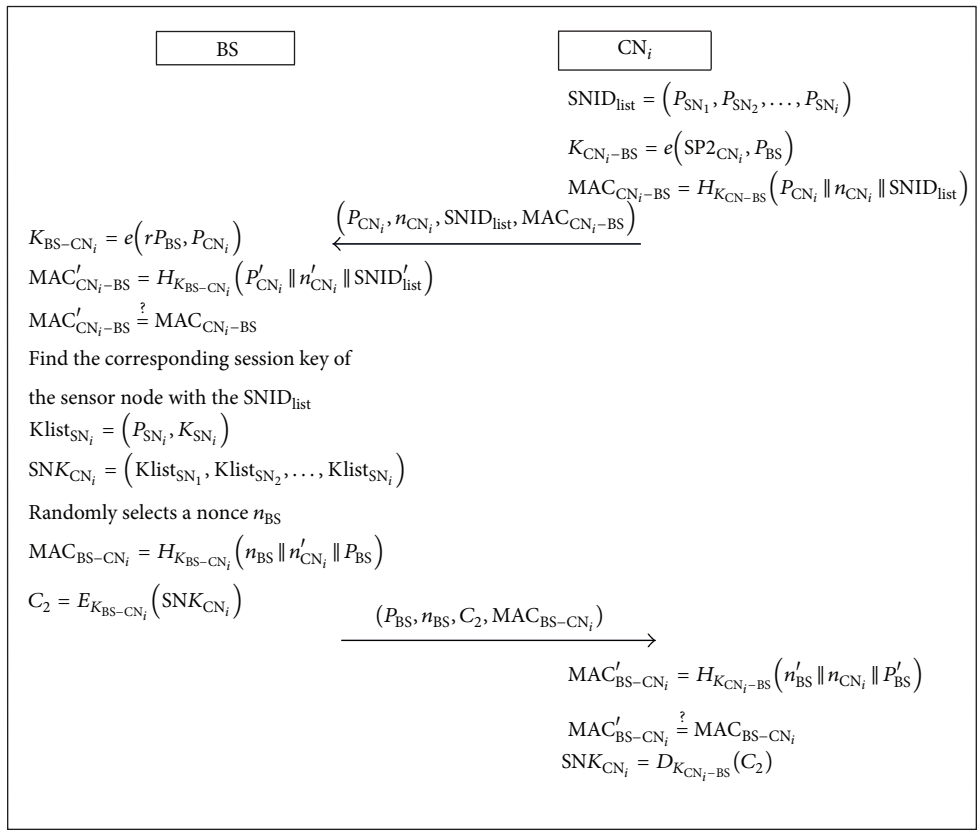

Figure 7: The overview of the authentication phase of the base station and the cluster node.

It checks if it is equal to $\text{MAC}_{\text{BS}-\text{CN}_i}$:

$$\text{MAC}'_{\text{BS}-\text{CN}_i} \stackrel{?}{=} \text{MAC}_{\text{BS}-\text{CN}_i}. \tag{30}$$

Then, the cluster node CN_i decrypts the encrypted message C_2:

$$\text{SN}K_{\text{CN}_i} = D_{K_{\text{CN}_i-\text{BS}}}(C_2). \tag{31}$$

In this phase, the cluster node CN_i obtains the sensor node's session key K_{SN_i} and finishes the mutual authentication with the base station.

3.6. The Authentication Phase of the ith Cluster Node and the jth Cluster Node. When the cluster cannot directly transmit the message to the base station, it will enter into this phase. The overview of the authentication of the cluster node and the cluster node is shown in Figure 8.

Step 1. The cluster node CN_i computes the pairing session key $K_{\text{CN}_i-\text{CN}_j}$ and randomly selects a nonce n_1 to compute the message authentication code $\text{MAC}_{\text{CN}_i-\text{CN}_j}$:

$$\begin{aligned} K_{\text{CN}_i-\text{CN}_j} &= e\left(\text{SP2}_{\text{CN}_i}, P_{\text{CN}_j}\right) \\ \text{MAC}_{\text{CN}_i-\text{CN}_j} &= H_{K_{\text{CN}_i-\text{CN}_j}}\left(P_{\text{CN}_i} \| n_1 \| \text{Msg}_{\text{collect}}\right). \end{aligned} \tag{32}$$

Then, the cluster node CN_i sends the message $(P_{\text{CN}_i}, n_1, \text{Msg}_{\text{collect}}, \text{MAC}_{\text{CN}_i-\text{CN}_j})$ to the cluster node CN_j.

Step 2. When receiving the message, the cluster node CN_j computes the pairing session key $K_{\text{CN}_j-\text{CN}_i}$:

$$K_{\text{CN}_j-\text{CN}_i} = e\left(\text{SP2}_{\text{CN}_j}, P_{\text{CN}_i}\right). \tag{33}$$

Then, the cluster node CN_j uses the pairing session key $K_{\text{CN}_j-\text{CN}_i}$ to compute the message authentication code $\text{MAC}'_{\text{CN}_i-\text{CN}_j}$ and checks if it is equal to $\text{MAC}_{\text{CN}_i-\text{CN}_j}$:

$$\begin{aligned} \text{MAC}'_{\text{CN}_i-\text{CN}_j} &= H_{K_{\text{CN}_j-\text{CN}_i}}\left(P'_{\text{CN}_i} \| n'_1 \| \text{Msg}'_{\text{collect}}\right) \\ \text{MAC}'_{\text{CN}_i-\text{CN}_j} &\stackrel{?}{=} \text{MAC}_{\text{CN}_i-\text{CN}_j}. \end{aligned} \tag{34}$$

The cluster node CN_j randomly selects a nonce n_2 and computes the message authentication code $\text{MAC}_{\text{CN}_j-\text{CN}_i}$:

$$\text{MAC}_{\text{CN}_j-\text{CN}_i} = H_{K_{\text{CN}_j-\text{CN}_i}}\left(P_{\text{CN}_j} \| n'_1 \| n_2\right). \tag{35}$$

The cluster node CN_j sends the message $(P_{\text{CN}_j}, n_2, \text{MAC}_{\text{CN}_j-\text{CN}_i})$ to the cluster node CN_i.

Step 3. After receiving the message $(P_{\text{CN}_j}, n_2, \text{MAC}_{\text{CN}_j-\text{CN}_i})$ the cluster node CN_i computes the message authentication

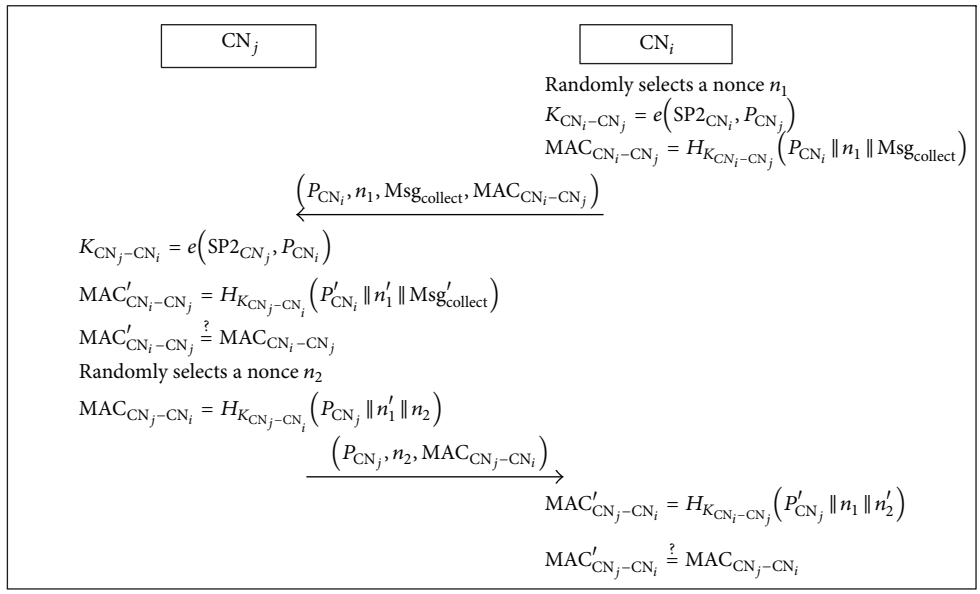

Figure 8: The overview of the authentication phase of the ith cluster node and the jth cluster node.

code $\mathrm{MAC}'_{\mathrm{CN}_j-\mathrm{CN}_i}$ and checks if it is equal to $\mathrm{MAC}_{\mathrm{CN}_j-\mathrm{CN}_i}$ as follows:

$$\begin{aligned}\mathrm{MAC}'_{\mathrm{CN}_j-\mathrm{CN}_i} &= H_{K_{\mathrm{CN}_i-\mathrm{CN}_j}}\left(P'_{\mathrm{CN}_j}\|n_1\|n'_2\right)\\ \mathrm{MAC}'_{\mathrm{CN}_j-\mathrm{CN}_i} &\stackrel{?}{=} \mathrm{MAC}_{\mathrm{CN}_j-\mathrm{CN}_i}.\end{aligned} \tag{36}$$

4. Security Analysis

4.1. Mutual Authentication

4.1.1. The Authentication between the Cluster Node and the Sensor Node

(1) The Cluster Node Authenticates Sensor Node. In the authentication phase of the cluster node and the sensor node, when the sensor node SN_i receives the message $(P_{\mathrm{CN}_i}, n_1, \mathrm{Msg}_{\mathrm{req}}, \mathrm{Msg}_{\mathrm{location}})$, the sensor node SN_i selects a nonce n_2 and uses $(n_1\|n_2\|\mathrm{Msg}_{\mathrm{rep}})$ to compute the message authentication code MAC_1:

$$\mathrm{MAC}_1 = H_{K_{\mathrm{SN}_i}}\left(n'_1\|n_2\|\mathrm{Msg}_{\mathrm{rep}}\right). \tag{37}$$

After the authentication of the base station and cluster node, the cluster node CN_i obtains the session key of the sensor nodes. The cluster node CN_i computes the message authentication code MAC'_1 and checks if it is equal to MAC_1:

$$\begin{aligned}\mathrm{MAC}'_1 &= H_{K_{\mathrm{SN}_i}}\left(n_1\|n'_2\|\mathrm{Msg}_{\mathrm{rep}}\right)\\ \mathrm{MAC}'_1 &\stackrel{?}{=} \mathrm{MAC}_1.\end{aligned} \tag{38}$$

(2) The Sensor Node Authenticates the Cluster Node. The cluster node CN_i uses the session key to encrypt the new parameter P_{SN_i} of the sensor node SN_i:

$$C_1 = E_{K_{\mathrm{SN}_i}}\left(P_{\mathrm{SN}_i}\right). \tag{39}$$

The cluster node CN_i randomly selects a nonce n_3 and computes the message authentication code MAC_2:

$$\mathrm{MAC}_2 = H_{K_{\mathrm{SN}_i}}\left(n_3\|n'_2\|C_1\right). \tag{40}$$

Then the cluster node CN_i sends the message $(P_{\mathrm{CN}_i}, C_1, n_3, \mathrm{MAC}_2)$ to the sensor nodes.

When the sensor node SN_i receives the message, it computes the message authentication code MAC'_2 and checks if it is equal to MAC_2:

$$\begin{aligned}\mathrm{MAC}'_2 &= H_{K_{\mathrm{SN}_i}}\left(n'_3\|n_2\|C'_1\right)\\ \mathrm{MAC}'_2 &\stackrel{?}{=} \mathrm{MAC}_2.\end{aligned} \tag{41}$$

Therefore, our schemes achieve the mutual authentication between the cluster node and sensor node.

4.1.2. The Authentication between the ith Cluster Node and the jth Cluster Node

(1) The jth Cluster Node Authenticates the ith Cluster Node. In the authentication phase of the cluster node and the cluster node, the cluster node CN_i computes the pairing session key $K_{\mathrm{CN}_i-\mathrm{CN}_j}$ and randomly selects a nonce n_1 to compute the message authentication code $\mathrm{MAC}_{\mathrm{CN}_i-\mathrm{CN}_j}$ as follows:

$$\begin{aligned}K_{\mathrm{CN}_i-\mathrm{CN}_j} &= e\left(\mathrm{SP2}_{\mathrm{CN}_i}, P_{\mathrm{CN}_j}\right)\\ \mathrm{MAC}_{\mathrm{CN}_i-\mathrm{CN}_j} &= H_{K_{\mathrm{CN}_i-\mathrm{CN}_j}}\left(P_{\mathrm{CN}_i}\|n_1\|\mathrm{Msg}_{\mathrm{collect}}\right).\end{aligned} \tag{42}$$

Then it sends the message $(P_{CN_i}, n_1, Msg_{collect}, MAC_{CN_i-CN_j})$ to the sensor nodes.

Upon receiving the message $(P_{CN_i}, n_1, Msg_{collect}, MAC_{CN_i-CN_j})$, the cluster node CN_i computes the pairing session key $K_{CN_j-CN_i}$:

$$K_{CN_j-CN_i} = e\left(SP2_{CN_j}, P_{CN_i}\right). \tag{43}$$

Then, the cluster node CN_j uses the pairing session key $K_{CN_j-CN_i}$ to compute the message authentication code $MAC'_{CN_i-CN_j}$ and checks if it is equal to $MAC_{CN_i-CN_j}$ or not:

$$\begin{aligned} MAC'_{CN_i-CN_j} &= H_{K_{CN_j-CN_i}}\left(P'_{CN_i} \| n'_1 \| Msg'_{collect}\right) \\ MAC'_{CN_i-CN_j} &\stackrel{?}{=} MAC_{CN_i-CN_j}. \end{aligned} \tag{44}$$

(2) The ith Cluster Node Authenticates the jth Cluster Node. Upon receiving the message $(P_{CN_i}, n_1, Msg_{collect}, MAC_{CN_i-CN_j})$, the cluster node CN_j randomly selects a nonce n_2 and computes the message authentication code $MAC_{CN_j-CN_i}$:

$$MAC_{CN_j-CN_i} = H_{K_{CN_j-CN_i}}\left(P_{CN_j} \| n'_1 \| n_2\right). \tag{45}$$

Then it sends the message $(P_{CN_j}, n_2, MAC_{CN_j-CN_i})$ to the cluster node CN_i.

When the cluster node CN_i receives the message, it computes the message authentication code $MAC'_{CN_j-CN_i}$ and checks if it is equal to $MAC_{CN_j-CN_i}$:

$$\begin{aligned} MAC'_{CN_j-CN_i} &= H_{K_{CN_i-CN_j}}\left(P'_{CN_j} \| n_1 \| n'_2\right) \\ MAC'_{CN_j-CN_i} &\stackrel{?}{=} MAC_{CN_j-CN_i}. \end{aligned} \tag{46}$$

Therefore, our scheme achieves mutual authentication among the cluster nodes.

4.1.3. The Authentication between the Base Station and the Cluster Node

(1) The Base Station Authenticates the Cluster Node. In the authentication phase of the base station and the cluster node, when the cluster node CN_i receives the message $(P_{CN_i}, n_{CN_i}, SNID_{list}, MAC_{CN_i-BS})$, the cluster node CN_i uses the pairing function to compute the pairing session key K_{CN_i-BS}:

$$K_{CN_i-BS} = e\left(SP2_{CN_i}, P_{BS}\right). \tag{47}$$

It computes the message authentication code MAC_{CN_i-BS}:

$$MAC_{CN_i-BS} = H_{K_{CN-BS}}\left(P_{CN_i} \| n_{CN_i} \| SNID_{list}\right). \tag{48}$$

Then it sends the message $(P_{CN_i}, n_{CN_i}, SNID_{list}, MAC_{CN_i-BS})$ to the base station.

When the base station receives the message, the base station uses the pairing function to compute the pairing session key K_{BS-CN_i}:

$$K_{BS-CN_i} = e\left(rP_{BS}, P_{CN_i}\right). \tag{49}$$

The base station computes the message authentication code MAC'_{CN_i-BS}:

$$MAC'_{CN_i-BS} = H_{K_{BS-CN_i}}\left(P'_{CN_i} \| n'_{CN_i} \| SNID'_{list}\right). \tag{50}$$

It checks if it is equal to MAC_{CN_i-BS}:

$$MAC'_{CN_i-BS} \stackrel{?}{=} MAC_{CN_i-BS}. \tag{51}$$

(2) The Cluster Node Authenticates the Base Station. When the base station receives the message $(P_{CN_i}, n_{CN_i}, SNID_{list}, MAC_{CN_i-BS})$, the base station randomly selects a nonce n_{BS} and computes the message authentication code MAC_{BS-CN_i}:

$$MAC_{BS-CN_i} = H_{K_{BS-CN_i}}\left(n_{BS} \| n'_{CN_i} \| P_{BS}\right). \tag{52}$$

Then it sends the message $(P_{BS}, n_{BS}, C_2, MAC_{BS-CN_i})$ to the base station.

Then, the cluster node CN_i receives the message and computes the message authentication code MAC'_{BS-CN_i}:

$$MAC'_{BS-CN_i} = H_{K_{CN_i-BS}}\left(n'_{BS} \| n_{CN_i} \| P'_{BS}\right). \tag{53}$$

It checks if it is equal to MAC_{BS-CN_i}:

$$MAC'_{BS-CN_i} \stackrel{?}{=} MAC_{BS-CN_i}. \tag{54}$$

Therefore, we complete the mutual authentication.

4.2. Dynamic Key Management. Our scheme offers random pairwise keys predistribution. After completing the information transmission, the cluster nodes and the sensor nodes update the session key for each session. It can prevent the replay attack. We divide it into two parts to analyze this process: the cluster node to sensor node and the cluster node to cluster node.

(1) The Cluster Node to Sensor Node. For example, if the sensor node SN_i wants to communicate with the cluster node CN_i, it computes the dynamic key $K_{SN_i-CN_i}$:

$$K_{SN_i-CN_i} = e\left(SP_{SN_i}, P'_{CN_i}\right). \tag{55}$$

Then, it encrypts the collected data with the key $K_{SN_i-CN_i}$ and sends the encrypted message $E_{K_{SN_i-CN_i}}(M)$ to the cluster node CN_i.

Upon receiving the message, the cluster node CN_i computes the session key $K_{CN_i-SN_i}$:

$$K_{CN_i-SN_i} = e\left(SP_{CN_i}, P'_{SN_i}\right). \tag{56}$$

Then, it decrypts the message $E_{K_{SN_i-CN_i}}(M)$ and gets the collected data.

After the transaction, the cluster node computes a new integer parameter α_{new} to update the identity of the cluster node and sensor nodes:

$$P_{CN_i} = \alpha_{new} P_{CN_i}$$
$$P_{SN_i} = \alpha_{new} P_{SN_i}. \quad (57)$$

So, our scheme updates a new session key in each section.

(2) The Base Station to Cluster Nodes. If the base station wants to update the session key, it can compute a new integer r_{new} to generate a new secret parameter $SP2_{CN_i}^{new}$:

$$SP2_{CN_i}^{new} = r_{new} P_{CN_i}. \quad (58)$$

Then, it encrypts the new secret parameter $SP2_{CN_i}^{new}$ with key $K_{BS-CN_i} = e(rP_{BS}, P_{CN_i})$ and sends to the corresponding cluster node CN_i. This mechanism can prevent the cluster node be captured. If a perceptible attacker gets the cluster node and intercepts the secret parameters in the sensor network, we can change the secret parameter via the base station.

4.3. Providing Session Key Protection (Elliptic Curve Discrete Logarithm Problem). The security of our scheme relies on the difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP) concerning bilinear groups. We compute parameter SP_{SN_i}; given the point $SP_{SN_i} = sP_{SN_i}$, it is difficult to obtain the secret parameter s by giving the secret parameter SP_{SN_i} and the P_{SN_i}. If an attacker steals the transferred traffic information, the attacker cannot crack the session key to decrypt the ciphertext.

4.4. Impersonation Attack. In the impersonation attack, if the attacker tries to steal the information between the sensors' communications, our scheme can defend against the information being used to conduct falsification, modification, replacement, and retransmission. In order to prevent the impersonation attack, the session key is generated by using mutual authentication. In the mutual authentication phase, we use $H_{K_{CN_i-CN_j}}(\cdot)$ and a one-way hash function with key $K_{CN_i-CN_j}$ to implement message authentication; the key is difficult to crack and calculate. The related information is shown as follows:

$$K_{CN_i-CN_j} = e\left(SP2_{CN_i}, P_{CN_j}\right)$$
$$MAC_{CN_i-CN_j} = H_{K_{CN_i-CN_j}}\left(P2_{PUBCN_j} \| n_1 \| Msg_{collect}\right). \quad (59)$$

So, the attacker cannot accomplish the impersonation attack.

4.5. Replay Attack. For the reply attack, we use dynamic key management to update the session key in each transaction, and we change the message authentication code in each section as follows. If an attacker tries to steal information to resend the same information to the target sensor node, it is impossible to pass the authentication.

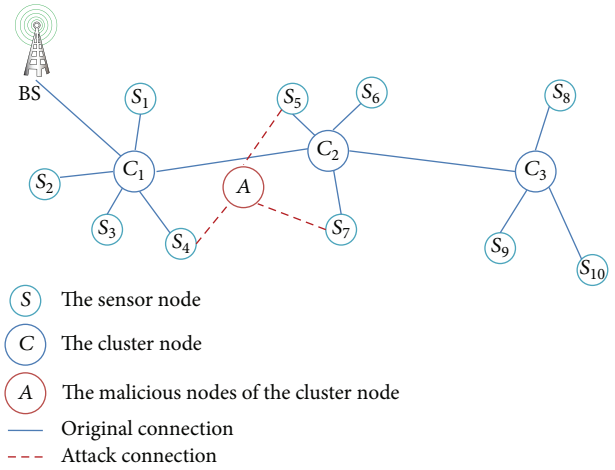

Figure 9: The scenario of the wormhole attack.

For example, in each section, the sensor node uses n_1^{new}, n_2^{new}, and Msg_{rep} to compute the new message authentication code MAC_1^{new}; the cluster node uses n_3^{new}, n_2^{new}, and C_1 to compute a new message authentication code MAC_2^{new}:

$$MAC_1^{new} = H_{K_{SN_i}}\left(n_1^{new} \| n_2^{new} \| Msg_{rep}\right)$$
$$MAC_2^{new} = H_{K_{SN_i}}\left(n_3^{new} \| n_2^{new} \| C_1\right). \quad (60)$$

If the attacker uses the message authentication code, the verifiers can verify the legality as follows:

$$MAC_1 \stackrel{?}{=} MAC_1^{new}$$
$$MAC_2 \stackrel{?}{=} MAC_2^{new}. \quad (61)$$

Therefore, the attacker cannot successfully achieve the replay attack.

4.6. Wormhole Attack. In a wormhole attack, an attacker records a packet in one location of the network and sends it to another location, creating a tunnel between the attacker's nodes. The packet is retransmitted to the network under the attacker's control [24, 25]. In the location-based routing determination phase of our scheme, the cluster nodes can establish the best route on the basis of the received broadcast location message in a monitoring area.

In Figure 9, if an attacker deploys a malicious cluster node A, it can collect the message from the sensor nodes S_4, S_5, S_7 and successfully intercept the messages. But, in our schemes, we involve the starting cluster node process in the location-based routing determination phase to build the communication connections initially. Afterward, we also involve the mutual authentication between the cluster nodes and sensor nodes. So, the attacker cannot successfully complete the wormhole attack.

Table 1: The comparisons of the prevention attacks.

	IKDM [11]	TinyPBC [12]	KMTD [13]	Ours
Against impersonation attack	Yes	Yes	Yes	Yes
Against replay attack	No	No	Yes	Yes
Against wormhole attack	No	No	No	Yes
Against message manipulation attack	No	No	No	Yes

Table 2: The cost comparison of the stored key and method used.

	IKDM [11]	TinyPBC [12]	KMTD [13]	Ours
Stored cost (cluster node)	1 SK + 2 IDs	N/A	2 SKs + 1 ID	1 SK + 2 IDs
Stored cost (sensor node)	2 SKs + 1 ID	1 SK + 2 IDs	2 SKs + 1 ID	1 SK + 2 IDs
New node algorithm	Yes	Yes	Yes	Yes
Detailed security analysis	No	No	Yes	Yes
Group-based protocol	Yes	No	Yes	Yes
Sensor's homogeneity	Homogeneity	Homogeneity	Hierarchical	Hierarchical
GPS capability (cluster node)	No	No	No	Yes
Path planning agreements	No	No	No	Yes
Dynamic key management	No	No	Yes	Yes

SK: the session key.
ID: the identity.

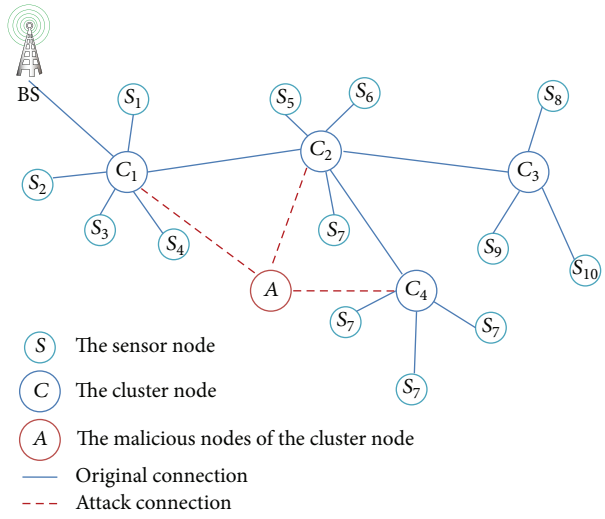

Figure 10: The scenario of the message manipulation attack.

4.7. Message Manipulation Attack. In a message manipulation attack, an attacker may drop, modify, or even forge exchanged messages in order to interrupt the communication process [15].

In Figure 10, an attacker deploys a malicious cluster node A and forges a fake cluster node; the malicious node A can receive messages from the cluster nodes, and the attacker may drop, modify, or even forge exchanged messages in order to interfere with the normal communication process. If a malicious cluster node A wants to interfere with a path among C_1, C_2, C_4, the cluster node C_4 communicates with the malicious cluster node A, and it cannot pass the mutual authentication successfully, because it is difficult to compute the HMAC's key. Moreover, the routing path is established in the location-based routing determination phase. It is impossible for an attacker to interfere with the routing path and message.

5. Discussions

In Table 1, our scheme can prevent more attacks than other related schemes. In Table 2, the cluster node only needs to store 1 session key and 2 identity parameters; we can use the bilinear pairing function to calculate the session key between the clusters or the session key between the cluster and the base station. We use the GPS to support the path planning agreements and use the location-based routing determination to build the network routing path. The dynamic key management protocol can update the session key to enhance the security.

The proposed scheme provides complete authentication. In Table 3, we make the computation cost of the session key agreement according to four stages.

(1) Sensor Node to Sensor Node. TinyPBC is a tiny pairing-based protocol and the computation cost is lower than the bilinear pairing-based protocol. In this stage, the scheme TinyPBC can use the cost $2T_h + 2T_P + 1T_E + 1T_D + 2T_n$ to generate a session key. We have more cost $2T_h + 2T_A$ than the TinyPBC scheme does. Our scheme inherits the advantage of the TinyPBC: we use the sensor level to build the hierarchical sensor network; that is, we use TinyPBC's topology (sensor node to sensor node) to our scheme (cluster node to cluster node), and it provides more powerful key management in WSN. It can also easily carry out message data aggregation and generate the session key between the cluster node and sensor nodes. So, our scheme can prevent more attacks, such as wormhole and message manipulation attacks. The computation cost of the pairing-based cryptography is

Table 3: The computation cost of the session key.

	IKDM [11]	TinyPBC [12]	KMTD [13]	Ours
Sensor node to sensor node	N/A	$2T_h + 2T_P + 1T_E + 1T_D + 2T_n$	N/A	N/A
Cluster node to sensor node	N/A	N/A	$7T_h + 2T_E + 2T_D + 2T_n$	$4T_h + 1T_a + 1T_E + 1T_D + 3T_n + 2T_A$
Cluster node to cluster node	$4T_h + 2T_{\text{pf}} + nT_d + 1T_E + 1T_D$	N/A	N/A	$4T_h + 2T_P + 1T_E + 1T_D + 2T_n + 2T_A$
Cluster node to base station	N/A	N/A	$6T_h + 3T_E + 4T_D + 2T_n + 2T_A$	$4T_h + 2T_P + 1T_E + 1T_D + 2T_n + 2T_A$

T_P: the time cost of a pairing operation.
T_a: the time cost of an additive group G.
T_h: the time cost of a hash operation.
T_E: the time cost of an encryption.
T_D: the time cost of a decryption.
T_n: the time cost of generating a nonce.
T_{pf}: the time cost of the polynomial function.
T_d: the time cost of the combining key fragment degree.
T_A: the time cost of the authentication.

the same as the TinyPBC scheme, but our scheme has better performance and security.

(2) Cluster Node to Sensor Node. According to the comparison of the KMTD, the sensor network is more convenient, complete, and secure. In order to achieve more security and easy key management, we use the bilinear pairing to generate the session key. We need not use the encryption and decryption to generate the session key or the base station's help. The computation cost is reduced and $3T_h + T_E + T_D$ and $2T_A + T_a + T_n$ are added to help the session key generation. This method can defend against more attacks and also has the path planning agreement.

(3) Cluster Node to Cluster Node. According to the comparison of the IKDM, the computation cost is reduced to $2T_{pf} + nT_d$ and added the bilinear pairing cost $2T_p + 2T_A + 2T_n$. The polynomial function easily generates the session key between the cluster nodes. However, the IKDM generates a session key which is unsuitable for large scale sensor network.

The construction methods of the session key need more key material of the cluster node to combine, so we chose the bilinear pairing to generate the session key in the cluster nodes and enhance security. It can more easily complete the session key.

(4) Cluster Node to Base Station. According to the comparison of the KMTD, the computation cost is reduced to $2T_h + 2T_E + 3T_D$ and $2T_p$ is added. We combine the message authentication code and the bilinear pairing key to accomplish the HMAC. The security of our scheme relies on ECDLP; the attacker cannot compute the secret key, and this increases the security between the base station and the cluster node. The session key and the mutual authentication are generated by the bilinear pairing function. It has the characteristic of ECDLP; the attacker cannot compute the secret key and pass verification.

Based on these concepts, we use the hierarchical topology which has more power and can easily implement key management. We combine the message authentication code and the bilinear pairing key to accomplish the message authentication.

6. Conclusion

We used bilinear pairing to design a dynamic key management and authentication of the hierarchical sensor network. We used the dynamic key management, pairing-based cryptography, hash message authentication code, and the GPS capability's cluster nodes to establish the secure agreement of the wireless sensor network. Our scheme achieves the following goals:

(1) proposing the dynamic key management to update the session key;

(2) overcoming the sensor node inherent limitations. We use the hierarchical network protocol in the wireless sensor network. It is more suitable for the large monitoring range in a wireless sensor network;

(3) providing the mutual authentication among the sensor nodes, cluster nodes, and the base station;

(4) using the characteristics of the Discrete Logarithm Problem to generate the session key, so that its security could be enhanced.

Notations

SN_i: The ith sensor node
CN_i: The ith cluster node
BS: The base station
G: A cyclic additive group which has the same prime order q
G_T: A cyclic multiplicative which has the same prime order q
$e(,)$: Pairing operation $e : G \times G \rightarrow G_T$
P_{SN_i}: The identity of the ith sensor node
P_{CN_i}: The identity of the ith cluster node

P_{BS}: The identity of the base station
s, r: An integer number of secret parameters generated by the base station
P_{PUB}: A public parameter, $P_{PUB} = s \cdot P$
$SP1_{SN_i}$, $SP1_{CN_i}$: A secret parameter using a secret number s to compute the secret parameter for the ith sensor node and cluster node, respectively
$SP2_{CN_i}$, $SP2_{CN_j}$: A secret parameter using a secret number r to compute the secret parameter for the ith cluster node and the jth cluster node, respectively
α, β: An integer of the secret parameter generated by the cluster node and the base station, respectively
K_{SN_i}: A session key of the ith sensor node
K_{P1}: A key pool generated by the base station, $K_{P1} = (K_{SN_1}, K_{SN_2}, \ldots, K_{SN_i})$
K_{P2}: A key pool generated by the base station, $K_{P2} = (K_{CN_1}, K_{CN_2}, \ldots, K_{CN_i})$
$K\text{list}_{SN_i}$: A key list, $K\text{list}_{SN_i} = (P_{SN_i}, K_{SN_i})$, for $i = 1$ to n
$K\text{list}_{CN_i}$: A key list, $K\text{list}_{CN_i} = (P_{CN_i}, K_{CN_i})$, for $i = 1$ to n
$SNID_{list}$: The cluster node collects the sensor's identity to send to the base station
n_A: A nonce generated by A
MAC: The message authentication code
$X \stackrel{?}{=} Y$: determine if X is equal to Y
$H(\cdot)$: A one-way hash function
$H_K(\cdot)$: A one-way hash function with key K
$E_K(M)$: Using an asymmetric key K to encrypt message M
$D_K(M)$: Using an asymmetric key K to decrypt message M
C_i: The ith encrypted message
Msg_{start}: The starting message which is used to start the cluster node which is dominated by the base station
$Msg_{location}$: The location message
Msg_{req}: The request message generated by the cluster node to find the sensor node
Msg_{rep}: The response message generated by the sensor node to respond to the cluster node request
Msg_{finish}: The finished message
$\dashrightarrow$: A secure channel
$\rightarrow$: An insecure channel.

Conflict of Interests

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

Acknowledgments

This research was supported by the Ministry of Science and Technology, Taiwan, under Contracts nos. MOST 103-2221-E-324-023, MOST 103-2632-E-324-001-MY3 and MOST 103-2622-E-212-009-CC2 and Collaborative Innovation Center for Modern Logistics and Business of Hubei (Cultivation).

References

[1] F. Xia, X. Yang, H. Liu, D. Zhang, and W. Zhao, "Energy-efficient opportunistic localization with indoor wireless sensor networks," *Computer Science and Information Systems*, vol. 8, no. 4, pp. 973–990, 2011.

[2] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[3] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the Association for Computing Machinery*, vol. 21, no. 2, pp. 120–126, 1978.

[4] K. M. Martin and M. Paterson, "An application-oriented framework for wireless sensor network key establishment," *Electronic Notes in Theoretical Computer Science*, vol. 192, no. 2, pp. 31–41, 2008.

[5] M. Ye, C. F. Li, G. H. Chen, and J. Wu, "EECS: an energy efficient clustering scheme in wireless sensor networks 10a.2," in *Proceedings of the 24th IEEE International Performance, Computing, and Communications Conference (IPCCC '05)*, pp. 535–540, April 2005.

[6] Z. X. Liu, Q. C. Zheng, L. Xue, and X. P. Guan, "A distributed energy-efficient clustering algorithm with improved coverage in wireless sensor networks," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 780–790, 2012.

[7] J. Yue, W. M. Zhang, W. D. Xiao, D. Q. Tang, and J. Y. Tang, "Energy efficient and balanced cluster-based data aggregation algorithm for wireless sensor networks," *Procedia Engineering*, vol. 29, pp. 2009–2015, 2012.

[8] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[9] 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, Washington, DC, USA, May 2003.

[10] J.-P. Sheu and J.-C. Cheng, "Pair-wise path key establishment in wireless sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2365–2374, 2007.

[11] Y. Cheng and D. P. Agrawal, "An improved key distribution mechanism for large-scale hierarchical wireless sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 35–48, 2007.

[12] L. B. Oliveira, D. F. Aranha, C. P. L. Gouvêa et al., "TinyPBC: pairings for authenticated identity-based non-interactive key distribution in sensor networks," *Computer Communications*, vol. 34, no. 3, pp. 485–493, 2011.

[13] C.-L. Chen, Y.-T. Tsai, and T.-F. Shih, "A novel key management of two-tier dissemination for wireless sensor network," in *Proceedings of the 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS '12)*, pp. 576–579, Palermo, Italy, July 2012.

[14] C. Blundo, A. de Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly secure key distribution for dynamic conferences," *Information and Computation*, vol. 146, no. 1, pp. 1–23, 1998.

[15] S. M. Mizanur Rahman and K. El-Khatib, "Private key agreement and secure communication for heterogeneous sensor networks," *Journal of Parallel and Distributed Computing*, vol. 70, no. 8, pp. 858–870, 2010.

[16] K.-A. Shim, Y.-R. Lee, and C.-M. Park, "EIBAS: an efficient identity-based broadcast authentication scheme in wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 1, pp. 182–189, 2013.

[17] T. W. Chim, S. M. Yiu, L. C. K. Hui, and V. O. K. Li, "MLAS: multiple level authentication scheme for VANETs," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1445–1456, 2012.

[18] Q. Qian, X. Shen, and H. Chen, "An improved node localization algorithm based on DV-hop for wireless sensor networks," *Computer Science and Information Systems*, vol. 8, no. 4, pp. 953–972, 2011.

[19] X. Wang, J. Ma, S. Wang, and D. Bi, "Distributed energy optimization for target tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 73–86, 2010.

[20] N. Komninos, D. D. Vergados, and C. Douligeris, "Authentication in a layered security approach for mobile ad hoc networks," *Computers and Security*, vol. 26, no. 5, pp. 373–380, 2007.

[21] N. Komninos, D. D. Vergados, and C. Douligeris, "Authentication in a layered security approach for mobile ad hoc networks," *Computers & Security*, vol. 26, no. 5, pp. 373–380, 2007.

[22] K.-A. Shim, "An ID-based aggregate signature scheme with constant pairing computations," *Journal of Systems and Software*, vol. 83, no. 10, pp. 1873–1880, 2010.

[23] T.-Y. Chang, "An ID-based group-oriented decryption scheme secure against adaptive chosen-ciphertext attacks," *Computer Communications*, vol. 32, no. 17, pp. 1829–1836, 2009.

[24] M.-Y. Su, "WARP: a wormhole-avoidance routing protocol by anomaly detection in mobile ad hoc networks," *Computers & Security*, vol. 29, no. 2, pp. 208–224, 2010.

[25] R. Stoleru, H. Wu, and H. Chenji, "Secure neighbor discovery and wormhole localization in mobile ad hoc networks," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1179–1190, 2012.