Secure Access Control Method for Wireless Sensor Networks

Cheng-Yeh Chen  Te-Cheng Hsu  John Y. Chiang  Wen-Shyong Hsieh
Department of Computer Science  Department of Computer Science  Department of Computer Science
Computer Science and Engineering, National Sun Engineering, Shu-Te National Sun and Information University University
80424, Kaohsiung, 82445, Kaohsiung, 80424, Kaohsiung, University, Taiwan, R.O.C. Taiwan, R.O.C. Taiwan, R.O.C.
88675254332  88675254332  88675252000-4343 Yat-sen University, 1. Department of Computer Science
zengya.chen@gmail.com hsugoo@gmail.com Chiang@cse.nsysu.edu.tw
2. Department of Computer Science
Wen-Shyong Hsieh
Yat-sen University, 80424, Kaohsiung, Yat-sen University, 80424, Kaohsiung, Taiwan, R.O.C.
88675254332 wshsieh@stu.edu.tw

The author(s) declare(s) that there is no conflict of interests regarding the publication of this article.

Abstract

Wireless sensor networks (WSNs) contain a large number of resource-constrained and energy-limited sensor nodes that are generally deployed in an open environment. Those sensor nodes communicate with each other or a base station via wireless channels. Therefore, secure access control is an important issue in WSNs because sensor nodes are susceptible to various malicious attacks during the authentication and key establishment phase, and the new node addition phase. In this study, we propose a new access control method based on elliptic curve cryptography and the chameleon hash function. This method addresses the security problems in existing research. It also has additional advantages since it does not require time synchronization between communication nodes, nor does it require node verification tables. In addition to our proposal, the correctness, security, resistance to possible attacks, and the performance of the proposed method are analyzed and evaluated. The results of our study demonstrate that the proposed method has outstanding performance and fulfills all the requirements for secure communication in WSNs.

KEYWORDS

Wireless sensor networks (WSNs), access control, and Chameleon hash function.
1. INTRODUCTION

Wireless sensor networks (WSNs) consist of base stations and a large number of resource-constrained and energy-limited sensor nodes that are typically deployed in various environments. Since the base stations and sensor nodes communicate with each other via wireless channels [1], the WSN system is threatened by malicious security attacks.

In WSNs, after a prolonged period of operation, the power of some of the sensor nodes gets exhausted. This occurs due to uneven distribution of radio transmission load, or damage caused by unpredictable events. The base station must dynamically add new nodes to the WSN to ensure the coverage and connectivity of sensor nodes. Following the addition of a node, the new node is authenticated by the neighboring nodes, and it establishes shared session keys with them for secure communication.

In order to attack or access communication information, malicious nodes may modify received messages, eavesdrop on transmission messages, insert false messages, or provide misleading information to legitimate sensor nodes. Therefore, access control security is a major concern in WSNs. Zhou et al. [2] proposed an access control protocol to improve the performance of traditional public-key-based encryption protocols. This protocol was based on the elliptic curve. However, it assumed that each sensor node could sustain for a tolerable time interval before it was compromised, rendering the scheme unsuitable for certain practical applications. Huang et al. [3] proposed an efficient access control protocol based on the elliptic curve and hash chains. This scheme could easily add new nodes and resist various attacks. Kim et al. [4] proposed an enhanced access control protocol based on the scheme proposed by Huang et al. Their research indicated that the method proposed by Huang et al. could not resist replay as well as active attacks. Further, it lacked hash chain renewability after the authentication and key establishment phase. In 2010, Jian et al. [5] and Peng et al. [6] demonstrated that the scheme presented by Kim et al. was vulnerable to masquerade attacks executed by new as well as legal nodes because it lacked hash chain renewability. In 2012, P. Zeng et al. [7] proposed a practical access control scheme, still based on elliptic curve and hash chain. However, such large number of key distribution scheme [2-7], is vulnerable to various adversary attacks and had huge storage overhead at the sensors node.

In 2013, Molavi et al. [8] use ECDLP method to solve security problem, but still need time synchronization and had more compute time.

Our goal to solve the problems mentioned above, we propose a secure and flexible access control protocol. The idea of proposed scheme based on the chameleon hash function [9-12] and Diffie–Hellman key exchange. This proposed method can dynamically and securely add new nodes to existing networks. Moreover, it overcomes the existing security problems, and had fewer number of transmission and does not require time synchronization or the verification table.

The remainder of this paper is organized as follows. Section 2 describes the preliminary information relevant to our scheme, i.e., the chameleon hash function based on the elliptic curve. In Section 3, the proposed scheme is introduced, followed by the correctness and performance analysis in Section 4. Finally, we draw our conclusions in Section 5.
2. PRELIMINARIES

This section introduces the properties of chameleon hash functions based on elliptic curves [7,8]. The base station initially chooses a large prime number \( q \), an elliptic curve \( E_q \), a point \( P \) of order \( n \) over \( E_q \), a subgroup \( G \) with order \( P \), and a cryptographically secure hash function \( f: \mathbb{Z}_q \times G \to \mathbb{Z}_q \). Here, \( M \) is a message space, and \( R \) is a finite space. Two random elements are chosen \( k, x \in \mathbb{Z}_q^* \), and then \( K = kP \) and \( Y = xP \) are computed. The public hash key is \( HK = (K,Y) \), and the private trapdoor key is \( TK = (k,x) \).

The proposed chameleon hash function is as follows:

Given the hash key \( HK \), \( CH(m,r): \mathbb{Z}_q \times \mathbb{Z}_q \to G \) is defined as

\[
CH(m,r) = f(m,kP)kP + rY
\]

(1).

According to [10], the chameleon hash function has the following properties:

1. Anyone that knows the public key \( HK = (K,Y) \) can compute the associated hash function \( CH(m,r) \) in polynomial time.
2. If the private trapdoor key is not known, it is not possible to find two inputs \( (m_1, m_2) \) that are mapped to the same output, such that \( CH(m_1, r_1) = CH(m_2, r_2) \).
3. The holder of the secret key information can easily find collisions for a given input, i.e., two messages \( m_1 \) and \( m_2 \), where \( CH(m_1,r_1) = CH(m_2,r_2) \).

Note that in signature schemes that use the chameleon hash function, there is a signer and a recipient, and two phases of execution. In the off-line phase, the signer pre-computes the chameleon hash values \( CH \) and the corresponding signatures \( \sigma \) on the hash values. In the online phase, the signer computes a random number \( r \) of the chameleon hash function for the given message \( m \) such that \( CH(m,r)=CH \).

The signature for the message \( m \) is \( r, \sigma \). When the recipient receives \( m, r, \sigma \), it decrypts \( \sigma \) and verifies \( CH(m,r)=CH \).

3. THE PROPOSED SCHEME

This section presents the proposed access control method based on the chameleon hash function [9-12] and used twice Diffie–Hellman key exchange. This study used signature parameter to archives robust authentication and key exchange. This method consists of two phases: the initialization phase, and the node authentication and key establishment phase. The basic concepts and operations are as follows:

1. Initialization phase: The base station sets the necessary values of the sensor nodes before they are deployed.
2. Node authentication and key establishment phase: A deployed node engages in mutual authentication with one of its neighboring nodes, and then establishes a common session key with
the neighboring node for secure communication.

In order to depict the proposed method more clearly, the notations used in this paper are listed in Table 1.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_i )</td>
<td>The identity of ( i )th node</td>
</tr>
<tr>
<td>( BS )</td>
<td>The base station</td>
</tr>
<tr>
<td>( q )</td>
<td>A large prime number</td>
</tr>
<tr>
<td>( E_q )</td>
<td>An elliptic curve</td>
</tr>
<tr>
<td>( p )</td>
<td>A generator of a group ( G )</td>
</tr>
<tr>
<td>( f )</td>
<td>A secure hash function ( f: Z_q \times G \rightarrow Z_q )</td>
</tr>
<tr>
<td>( Q^i )</td>
<td>The x-coordinate of elliptic curve point ( \bar{Q} )</td>
</tr>
<tr>
<td>( r )</td>
<td>A random number ( r \in R )</td>
</tr>
<tr>
<td>( CH )</td>
<td>The chameleon hash function ( CH(m,r) = f(m,K) \cdot K + rY )</td>
</tr>
<tr>
<td>( Mav )</td>
<td>The mutual authentication value</td>
</tr>
</tbody>
</table>

### 3.1 Initialization phase

To set up the network system parameters, the base station initially chooses a large prime number \( q \), an elliptic curve \( E_q \), a point \( P \) of order \( n \) over \( E_q \), a subgroup \( G \) with order \( P \), and a cryptographically secure hash function \( f: Z_q \times G \rightarrow Z_q \). It then completes the initialization phase by carrying out the following steps.

Step 1: Chooses random elements \( x, \alpha \in \mathbb{Z}_q^* \) as the secret key, and then computes the public key

\[
Y = xP \quad \text{as well as the chameleon hash value of the base station} \quad CH_{BS} = \alpha Y .
\]

Step 2: Select random numbers \( k_i \in \mathbb{Z}_q^* \) as the secret key, where \( i = 1, 2, 3, \ldots, n \), computes the trapdoor/hash key pair \((k, x^{-1}, k, P)\) and loads \( k, P \) to node \( N_i \).

Step 3: Computes a security key \( r_i \) of \( CH_{BS} \) for the given \( m \), where

\[
m = N_i \parallel k, P^x \quad \text{of node} \quad N_i , \quad \text{such that} \quad CH_{BS} = f(m,k,P) \cdot k, P + r_i Y .
\]

Here,

\[
r_i = \alpha - f(m,k,P)k, x^{-1} \mod q .
\]

Loads \((N_i, k_i, P, r_i, CH_{BS})\) to node \( N_i \), and publishes \( E_q, q, n, P, Y \) to all the sensor nodes.

These three steps of the initialization phase are shown in Figure 1.
The node \( N_i \) and the base station \( \text{BS} \)

<table>
<thead>
<tr>
<th>Preloads</th>
<th>Choose secret key ( x, \alpha \in \mathbb{Z}_q^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_i, k, P, r, CH_{BS} )</td>
<td>Computes public key ( Y = xP )</td>
</tr>
<tr>
<td></td>
<td>and chameleon hash value ( CH_{BS} = aP )</td>
</tr>
<tr>
<td></td>
<td>Generates ( k_r \in \mathbb{Z}_q^* )</td>
</tr>
<tr>
<td></td>
<td>Computes ( k_r x^{-1}, k_r P )</td>
</tr>
<tr>
<td></td>
<td>( r_r = \alpha - f(m, k) P) k_r x^{-1} \mod q</td>
</tr>
<tr>
<td></td>
<td>( m = N_i \parallel k_r P^x )</td>
</tr>
</tbody>
</table>

3.2 Authentication and key establishment phase

After all the sensor nodes have been deployed, if node \( N_i \) wants to communicate with another node \( N_j \), they must implement the following steps to authenticate each other. Subsequently, they must establish a shared session key for securing their communication.

Step 1: Node \( N_i \) chooses a random integer \( a \in \mathbb{Z}_q^* \) to compute the public key \( aP \) and \( r_i Y \), and then sends \( (N_i, aP, r_i Y, k_r P) \) to the node \( N_j \).

Step 2: Node \( N_j \) computes the chameleon hash value \( CH'_{BS} \) of node \( N_i \) based on the received message \( (N_i, aP, r_i Y, k_r P) \). \( m = N_i \parallel k_r P^x \), according to equation (1), where

\[
CH'_{BS} = f(m, k_r P) k_r P + r_i Y.
\]

Then \( N_j \) compares \( CH'_{BS} \) with the chameleon hash value of the base station \( CH_{BS} \).

If \( CH'_{BS} = CH_{BS} \), then node \( N_j \) chooses a random number \( b \in \mathbb{Z}_q^* \) to compute the public key \( bP \) and the shared session key \( baP(abP) \) between \( N_i \) and \( N_j \).

Then, \( N_j \) uses \( r_j \) to compute the mutual authentication value \( Mav_{N_j} \) for the given values of \( r_j \) and \( abP^x \), where \( Mav_{N_j} = f((abP^x \parallel r_j r_i Y^X), k_r P) \). It subsequently delivers the message \( (N_j, bP, r_i Y, k_r P, Mav_{N_j}) \) to node \( N_i \).

Step 3: When node \( N_i \) receives the message from \( N_j \), it also computes the chameleon hash value \( m = N_i \parallel k_r P^x \), \( CH'_{BS} = f(m, k_r P) k_r P + r_i Y \) of node \( N_j \), according to the received message

\( (N_j, bP, r_i Y, k_r P, Mav_{N_j}) \) from \( N_j \). It then compares \( CH'_{BS} \) with the chameleon hash value.
of the base station \( CH_{BS} \). If \( CH'_{BS} = CH_{BS} \), then node \( N_i \) computes the shared session key \( abP \), the mutual authentication value \( Mav'_{Nj} \) for the given \( r_j \), and \( abP^X \), where

\[
Mav'_{Nj} = f((abP^X \parallel r_jY^X), k_jP).
\] (3)

If \( Mav'_{Nj} = Mav_{Nj} \), then the node \( N_j \) is legitimized, and node \( N_i \) sends back a mutual authentication value \( Mav_{Nj} \) for the given \( r_i \) and \( abP^X \), where

\[
Mav_{Nj} = f((abP^X \parallel r_jY^X), k_jP).
\] (4)

Step 4: When \( N_j \) receives \( Mav_{Nj} \), it also computes the value for \( Mav'_{Nj} = f((abP^X \parallel r_jY^X), k_jP) \). If \( Mav'_{Nj} = Mav_{Nj} \), then node \( N_j \) is authenticated. Otherwise, the value \( Mav_{Nj} \) is discarded.

The steps of the authentication and the key establishment phase are shown in Figure 2.

Figure 2. Authentication and key establishment phase

The Node \( N_i \) Generates \( a \) and \( aP, rY \)

Check \( CH_{BS} ? = CH_{BS}' \)

Compute \( abP \) and \( r_jY^X \)

\( Mav'_{Nj} = f((abP^X \parallel r_jY^X), k_jP, Mav_{Nj}) \)

The Node \( N_j \) Generates \( b \) and \( bP, baP, rY \)

Compute \( abP \) and \( r_jY^X \)

\( Mav_{Nj} = f((abP^X \parallel r_jY^X), k_jP) \)

Check \( Mav'_{Nj} ? = Mav_{Nj} \)

Note that after the sensor network has operated for a certain period of time, new nodes must be deployed in the network to extend its life. For example, for the deployment of a new node \( N_{i+1} \), the base station adds the new node by following steps 2 and 3 of the initialization phase outlined in section 3.1.

4. ANALYSIS OF THE PROPOSED SCHEME

In this section, we analyze the correctness, the security, and the performance of the proposed access control protocol. The security analysis also shows how the proposed method can resist known security attacks.
4.1 Correctness

In the authentication and key establishment phase, node \( N_j \) authenticates node \( N_i \) based on the chameleon hash value of the node \( N_i \), i.e., the computed result of \( CH_{bs} = f(m, k, P) \cdot k, P + rY \).

Based on the received message \((N_i, aP, rY, kP)\) from node \( N_i \), and the published message of the base station \((P, Y)\), the Eq. (2) can be rewritten as

\[
CH_{bs} = f((N_i, k, P^x), k, P) \cdot k, P + (\alpha - f((N_i, k, P^x), k, P)k, x^{-1}) \cdot xP
\]

\[
= f((N_i, k, P^x), k, P) \cdot k, P - f((N_i, k, P^x), k, P)k, P + \alpha xP
\]

\[
= \alpha Y
\]

All nodes of the WSN have the same chameleon hash value as the base station, i.e., \( CH_{bs} = \alpha Y \). This is preloaded by the base station. Therefore, node \( N_i \) can verify the received identification and the hash key \( k, P \) of node \( N_j \) based on the result of comparison of \( CH_{bs} \) with \( CH_{bs} \). Moreover, node \( N_j \) also can ensure whether node \( N_i \) is valid by applying the same method that is used by node \( N_j \). In this case, the received message is from node \( N_j \).

If node \( N_i \) confirms that node \( N_j \) is valid, a shared session key \( abP \) can be established by using the mutual authentication value \((Mav_N)\) of the received message from node \( N_j \). Subsequently, \( Mav_N \) can be computed, where

\[
Mav'_N = f((abP^x \parallel r, rY^x), k, P)
\]

Eq. (3) can be rewritten as

\[
Mav'_N = f((abP^x \parallel r, rY^x), k, P)
\]

\[
= f((baP^x \parallel r, rY^x), k, P)
\]

\[
= Mav_N
\]

On the other hand, the random values \( a \) and \( b \) are chosen by nodes \( N_i \) and \( N_j \), respectively. The point \( P \) of the Elliptic curve and the public key \( Y \) are initially published by the base station. Therefore, node \( N_i \) can obtain the value of \( abP^x \) using the aforementioned parameters. \( r_i \) is \( N_i \)'s security key. Node \( N_i \) then compares \( Mav'_N \) with \( Mav_N \), which are obtained by Diffie–Hellman key exchange. If they are equal, the shared session key has been established. In a similar manner, node \( N_j \) can apply the method used by node \( N_i \) to confirm the mutual authentication value.

4.2 Security Analysis

The proposed scheme utilizes the chameleon hash function and twice Diffie–Hellman key exchange produce the mutual authentication value to establish mutual authentication between communication nodes. This study construct of security chameleon hash function, further proof in
For example, if nodes $N_i$ and $N_j$ require mutual authentication, $N_j$ will first compare the preloaded chameleon hash value of the base station, $CH_{BS}$, with the chameleon hash value of node $N_i$, $CH_{BS}^i$, obtained by using Eq. (1). However, the calculation of $CH_{BS}^i$ requires the ID, the hash key $k_i$, and the value $r_iY$ of node $N_i$. Thus, the process can authenticate the ID and hash key because computing $CH_{BS}^i$ is an elliptic curve discrete logarithm problem (ECDLP) if the attacker does not have any information about the ID and hash key. Furthermore, $r_i$ is protected by the secret key $\alpha$ and $k_i^{-1}$.

After node $N_i$ has been authenticated, node $N_j$ applies Eq. (1) and Eq. (2) to compute the mutually authenticated value $Mav_{N_j}$. Then, $Mav_{N_j}$ is compared with $Mav_{N_i}$. $Mav_{N_j}$ is obtained by using the shared session key $abP$ and security key $r_j$. However, only the communication nodes recognize the session key $abP$, and only node $N_i$ and the base station can have the secret key $r_j$.

In addition to security analysis, the following paragraphs explain how the proposed method can resist attacks such as legal node masquerading attacks, forgery attacks, new node masquerading attacks, replay attacks, and man-in-the-middle attacks.

A. Legal node masquerading attack

In this study, the authentication of node $N_j$ depends on the comparison of chameleon hash values in the authentication and key establishment phase.

First, the node to be authentication node $N_j$ checks $CH_{BS}$. $N_j$ can provide a legitimate value of $r_jY$ only if it is legitimate, and only the base station know the secret keys $\alpha$ and $k_j^{-1}$ can provide corresponding $r_j$ for given message.

Secondly, only a legitimate node $N_j$ can use $r_j$ to compute the mutual authentication value and get authenticated successfully.

An attacker may obtain the commutation values by eavesdropping on the communication between nodes $N_i$ and $N_j$. However, even if the attacker obtains the values of $aP$ and $bP$ from the authentication and key establishment phase, deriving the legalized session key $abP$ is extremely difficult because of the ECDLP. Therefore, the proposed scheme can prevent a legal node masquerading attack.

B. Forgery attack

Consider the case where an adversary has obtained the commutation values by eavesdropping on the communication channel.

The adversary may attempt to create a legitimate authentication message using the following equation.

$$r'_i = r_i + k_jx^{-1}(f(m', k_i) - f(m, k_i)) \mod q.$$  \hspace{1cm} (5)

However, it is not possible for the adversary to create a forged message because the value of $r'_i$ cannot be computed without secret key $k_jx^{-1}$. 
C. New node masquerading attack

When some of the sensor nodes in the wireless sensor network are damaged or have exhausted their power, new sensor nodes must be added to the network to extend its life. In the new node addition phase, the base station preloads the \( k_{i_1}P, CH_{BS}, N_{i_1} \) and \( r_{i_1} \) values to a new node. Owing to the non-availability of the secret keys \( x \) and \( \alpha \), the attacker cannot compute the value of \( r_{i_1} \); hence, it cannot use \( r_{i_1} \) to verify the ID and the hash key of the new node. Therefore, the proposed scheme can prevent new node masquerading attacks.

D. Replay attack

A replay attack is one in which an attacker captures the transmitted messages of a legitimate node, and later replays them on the network in an attempt to imitate legitimate authentication messages. For example, if an attacker transmits the captured message \( (N_i, aP, rY, k_1P) \) to another node \( N_j \), the attacker must then provide \( r_i \) for establishing a mutual authentication value \( M_{av} \). The \( M_{av} \) is required for a shared session key with the node to be connected. However, it is not possible for the attacker to obtain \( r_i \) without the secret key \( k_i x^{-1} \).

In addition, if the attacker sends the value \( M_{av} \) to connect node \( N_j \), node \( N_j \) can use the shared session key to authenticate whether the connecting node is legitimate or not. Another point that must be noted is that every connection between two nodes uses the up-to-date session key values \( a \) and \( b \). By means of these different strategies, the proposed method can resist replay attacks.

E. The man-in-the-middle attack

In the proposed scheme, the communication nodes can mutually authenticate and establish session keys between users and servers. Although an attacker may launch a man-in-the-middle attack, the attacker can only know the values of \( aP \) and \( bP \). It still has to resolve the ECDLP. Moreover, even if the attacker obtains user information (such as \( N_i, k_1P, rY \) ), the attacker still cannot pass the authentication and key establishment phase because it cannot compute the mutual authentication value \( M_{av} \) of the session key. Therefore, the proposed scheme can resist man-in-middle attacks.

4.2.1 Session key security

The proposed scheme uses a session key. Only the communicating parties know the session key \( abP \) when the user verifies the message from receiving party. The session key \( SK = abP \) is not known by anyone other than the communicating parties because the random values \( aP \) and \( bP \) are protected by the ECDLP. Therefore, the proposed scheme provides session key security.
4.3 Performance Analysis

Table 2 shows the proposed scheme compared with related works Huang et al. [3], and Kim et al. [4], H. Lee et al. [6] and B. Molavi et al. [8] in terms of the function of access control method. The computation cost and the transmission cost are two popular benchmarks for evaluation of the efficiency of the WSNs. Table 3 shows the computation costs of the proposed method during different phases. When compared to the computation cost of the hash chain and the elliptic curve-based schemes, the computation cost of the proposed method is slightly higher. However, our proposed method does not require time synchronization or the verification table for authentication. Hence, it is more efficient than the hash chain and elliptic curve-based methods.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Resist various attacks</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Easily add new node</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Do not need time synchronization</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>After authentication do not need broadcast</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Do not had verify table</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>After add new node do not need broadcast</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 3: The computation costs of proposed method during different phases

<table>
<thead>
<tr>
<th>Phase</th>
<th>Computation cost</th>
<th>Node</th>
<th>$N_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialization</td>
<td>$2T_{e_m} + n(2T_{e_m} + 1T_{mul})$</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>Authentication and key establishment</td>
<td>n/a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New node adding</td>
<td>$2T_{e_m} + 1T_{mul}$</td>
<td>n/a</td>
<td></td>
</tr>
</tbody>
</table>

$T_{e_m}$ : time cost for one multiplication computation over an elliptic curve.

$T_{h}$ : time for cryptographic secure hash function.

$T_{mul}$ : time cost of performing the modulus multiplication operation.

The energy cost of communication and cryptography in wireless sensor networks is an important consideration. In [13] indicate energy cost of cryptographic protocols, both from a communication and a computation. According to [13] measure practical costs of computation and communication in WSN
used TelosB sensors run Elliptic Curve Diffie-Hellman key exchange with authentication energy costs is 130(mJ), communication costs is 58.3(mJ), and computation cost is 77(mJ). As a result, we can realize communication cost is an important consideration same with computation cost. Table 4 presents the computation and the transmission costs incurred by using the methods put forward by Zhou et al. [2], Huang et al. [3], Kim et al. [4] and H. Lee et al. [7]. It also displays the costs incurred by using our proposed method. The protocol proposed by Huang et al. requires seven transmissions and updates the broadcast hash chain after each authentication, key establishment, and new node addition phase. Our method requires only five transmissions. Consequently, it needs less bandwidth than the schemes proposed by Zhou et al. and Huang et al. Therefore, the proposed scheme can be applied to applications from the WSNs criteria that had less number of transmissions and had lower data rate of communication. However, every broadcast has resulted in poor throughput of WSNs. Our proposed do not need broadcast to update node list and verify table. Furthermore, it can be implemented by using any elliptic curve and cryptographic secure hash function.

Table 4. The comparison of computation costs and number of transmissions

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Each node achieves authentication and establishes shared session key</th>
<th>Total number of transmissions to establish shared session key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhou et al.(2007)</td>
<td>$3T_{e_m} + T_{mul} + T_h$</td>
<td>21</td>
</tr>
<tr>
<td>Kim and Lee(2009)</td>
<td>$2T_{e_m} + 8T_h$</td>
<td>14</td>
</tr>
<tr>
<td>Huang et al.(2009)</td>
<td>$2T_{e_m} + 5T_h$</td>
<td>10</td>
</tr>
<tr>
<td>H. Lee et al.(2012)</td>
<td>$2T_{e_m} + 5T_h$</td>
<td>8</td>
</tr>
<tr>
<td>Our proposed</td>
<td>$5T_{e_m} + T_h$</td>
<td>5</td>
</tr>
</tbody>
</table>

$T_{e_m}$: point multiplication computation over an elliptic curve.
$T_h$: time for cryptographic secure hash function.
$T_{mul}$: time cost of performing the modulus multiplication operation.

5. CONCLUSION

This study demonstrated a new access control scheme for WSNs that has three characteristics. First, the proposed scheme can establish mutual authentication between two communication nodes, and resist possible attacks when the base station adds a new node to the WSN. Second, the proposed scheme can establish a common session key between two nodes in an efficient manner. Moreover, it does not require time synchronization or the verification table. Third, the proposed scheme requires fewer transmissions and less bandwidth than that of existing schemes.
References