A Scalable ID-Based Pairwise Key Establishment Protocol for Wireless Sensor Networks

Huan-Chung Lin and Yuh-Min Tseng*

Department of Mathematics

National Changhua University of Education

Jin-De Campus, Chang-Hua 500, Taiwan, R.O.C.

ymtseng@cc.ncue.edu.tw

Received 27 March 2007; Revised 23 April 2007 ; Accepted 20 May 2007

Abstract. Wireless sensor networks (WSNs) have gained much attention due to large number of applications. The WSN systems are usually deployed in hostile environments where they encountered a wide variety of malicious attacks. In order to protect the transmitted messages between any two adjacent sensor nodes, a mutual authentication and key exchange protocol is required for wireless sensor networks. Because some nature restrictions of sensor nodes which include low power, less storage space, low computation ability and short communication range, most existing protocols attempt to establish a pairwise key between any two adjacent sensor nodes by adopting a key pre-distribution approach. However, this approach has some inherent drawbacks. With rapid growth of cryptographic techniques, recent results show that Elliptic Curve Cryptography (ECC) is suitable for resource-limited WSNs. In this paper, we propose a scalable ID-based pairwise key establishment protocol that allows a sensor node can establish session keys with all one-hop adjacent nodes by one broadcast message. Compared to other existing protocols, our protocol offers two important advantages: (1) each node requires only constant memory storage, so the proposed protocol is scalable; (2) it ensures that a sensor node can directly establish secure communications with other adjacent nodes, so that the entire connectivity of the network is guaranteed. We show that it can withstand the compromise attack with captured nodes. A performance analysis demonstrates that our protocol is well suited for resource-limited WSNs.

Keywords: Wireless sensor networks, Identity based, Elliptic Curve Cryptography, Key establishment.

1 Introduction

Recently, wireless sensor network (WSN) has gained much attention due to large number of applications. In early days, wireless sensor networks are used to military applications such as battle field surveillance and enemy tracking [1]. Afterwards, some science applications such as habitat monitoring, environment observing and calamity alert are also developed. Now, many kinds of civil applications have mushroomed. For example, remote patients monitoring, inventory systems, home automation and smart kindergarten are some of these applications [1], [2].

A typical WSN consists of one base station and a large set of sensor nodes. Sensor nodes have the ability to gather and process data, and then forward these data to the base station. However, WSNs are usually deployed in hostile environments where they encountered wide variety of malicious attacks. In which, how to protect the secure communication among sensor nodes is an important research issue, a mutual authentication and key exchange protocol is required for this requirement. Because some nature restrictions of sensor nodes which include low power, less storage space, low computation ability and short communication range, subsistent secure key exchange protocols used on wired networks are not suitable for WSNs.

Since sensor nodes are battery powered, researchers argued that traditional public key systems are unsuitable for WSNs. Therefore, some recently proposed protocols for WSNs attempt to establish a pairwise key between two adjacent sensor nodes by adopting a key pre-distribution approach. However, this approach has some inherent drawbacks. For example, in order to establish pairwise keys with other nodes, each node must be pre-distributed some sensitive data (or keys) which is the same with the partial data of other nodes. However, it incurs that the adversary may get some sensitive data of non-captured nodes by capturing some nodes. Another drawback is that the key pre-distribution approach usually can not guarantee entire connectivity even with high deployment density. With rapid growth of cryptographic techniques, recent results [3], [4], [5], [6], [7] show that Elliptic Curve Cryptography (ECC) is suitable for resource-limited WSNs because of its shorter key size, faster computation time and the same security level with traditional public key system.

* Correspondence author

In this paper, we first present an improved ECC protocol based on the two-party key exchange protocol [8]. Then, we propose a concrete protocol for secure communication in wireless sensor networks by using the improved ECC key agreement protocol. Compared to other existing protocols for WSNs, the proposed protocol has the following properties: (1) each node requires only constant memory storage, so our protocol offers perfect scalable property; (2) the proposed protocol ensures that a sensor node can directly establish secure communications with other adjacent nodes, so it can guarantee the entire connectivity of the network; (3) each node broadcasts only one message to establish session keys with each one-hop adjacent node; (4) it can withstand the compromise attack with captured nodes; (5) As adding new nodes into a sensor network, re-keying is achieved easily; (6) forward secrecy is offered by using the improved ECC key agreement protocol. Furthermore, we make a performance analysis to demonstrate that our protocol is well suited for resource-limited WSNs.

The rest of the paper is structured as follows. Section 2 discusses some existing protocols for WSNs. An improved ECC key agreement protocol is presented in Section 3. In Section 4, a concrete pairwise key establishment protocol for secure communication in wireless sensor networks is presented. Performance analysis is discussed in Section 5. Discussions and conclusions are given in Sections 6 and 7, respectively.

2 Related works

In this section, we review the recently proposed protocols [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21] for pairwise key establishment in wireless sensor networks. Most existing protocols are based on key pre-distribution approach to construct the session keys among sensor nodes. We divide these protocols into four categories: random key pre-distribution protocols [9], [10], [11], group-based key pre-distribution protocols [12], [13], [14], [15], hierarchical structure protocols [16], [17] and other protocols [18], [19], [20], [21].

2.1 Random key pre-distribution protocols

In 2002, Eschenauer and Gligor [9] proposed the first random key pre-distribution protocol which consists of three phases: the key pre-distribution phase, the shared-key discovery phase and the path-key establishment phase. Initially, the system generates a large global key pool which contains all keys used by the system. In the key pre-distribution phase, the system randomly selects a subset of keys from the key pool and loads them into a sensor node. After deployment, nodes perform the shared-key discovery phase. Each node exchanges key list with other adjacent nodes. Each pair of any two adjacent nodes discovers a common key to be their pairwise key or they should run the path-key establishment phase to establish the common key. In path-key establishment phase, any two nodes with no common key search a secure path and transmit a pairwise key through the path. All other nodes in the path can get the key because they must decrypt and encrypt the pairwise key hop-by-hop.

According to the random graph theory and the probability theory, Eschenauer and Gligor demonstrate that if the probability of any two nodes shared at least one common key over a critical value, the connectivity of the entire network can be reached with a high probability. In other words, it cannot guarantee the network's entire connectivity. Additionally, it encounters a network resilience problem that an adversary can get some sensitive data between non-captured nodes by capturing other nodes. We term such attack as "compromise attack with captured nodes". This is because the system pre-loads partial repetition data to each node in order to let nodes can establish pairwise keys with other nodes. In fact, most key pre-distribution protocols will suffer from this attack.

To resist this attack, Chan et al. [10] improved the Eschenauer-Gligor protocol [9] by "q-composite" method to enhance the capability of withstanding this attack. At least q ($q \geqq 2$) common keys are used to construct a pairwise key between two nodes in the shared-key discovery phase. On the other hand, Du et al. [11] proposed another protocol by using Blom's method [22]. They used a global matrices pool to replace the global key pool in [9]. In the key pre-distribution phase, each node randomly selects some matrices from the global matrices pool. And then loads a row of elements from each determined matrix into the node. In this case, any two adjacent nodes have a row of elements from the same matrix can establish a pairwise key. However, these two improved protocols [10], [11] do not offer scalable property. It means that a node needs to store many keys to keep the connection probability when the system is large. If the network has n nodes, each node requires $\Omega(n)$ storage requirement [19]. Furthermore, both protocols [10], [11] still need intermediate nodes to establish pairwise key between two adjacent nodes which can not discovery common keys. As the discussions above, it will degrade the network security.

In 2004, Wacker et al. [23] adopt a multi-path method to improve the security of path keys. In this case, a special connection structure needs to be maintained. Unfortunately, searching multi-disjoint paths is a difficult task for nodes. In 2005, Wacker et al. [24] presented an improved approach using a recursive manner to find the multi-path. But it requires a more complex structure to keep the recursive manner can work. However, the multi

paths require more intermediate nodes and communications. It will require more energy consumption and increase the adversary's attack chance.

According to the discussion above, random key pre-distribution protocols can not guarantee entire connectivity even with high deployment density [19], [25]. They require $\Omega(n)$ storage space so that they are not scalable. They can not resist the compromise attack with captured nodes. Furthermore, almost all (about *99%*) of the links are established by path keys [14].

2.2 Group-based key pre-distribution protocols

Generally, we assume that nodes are individually deployed to target area. In some scenarios, nodes are sprayed to target area to form several groups. For example, a packet of nodes is airdropped by a helicopter. In this case, the packet of nodes has high probability to fall into its communication range of each others. If a node can efficiently establish pairwise keys with the same group nodes, it will reduce the probability of performing the path-key phase.

Du et al. [12] first presented the deployment knowledge concept into the key pre-distribution approach. They assumed that each group of nodes is pre-determined as a deployment point. And the adjoining groups are determined in advance. In fact, this protocol is an improved version based on [9]. The global key pool is divided into many sub-key pools according to the deployment points. Each group is assigned to a specific sub-key pool which has a certain number of common keys with the other sub-key pools. Each node in the same group is assigned some keys from the same sub-key pool before deployment. By this key pre-distribution phase and group deployment approach, performance and security of the network are improved. In 2006, they proposed an improved protocol [15] on their original protocol [12] by using the global matrices pool [11].

On the other hand, Liu et al. [13] and Zhou et al. [14] respectively presented a group-based protocol in 2005. Liu et al. [13] assigned each node to be belonged to two kinds of groups: deployment group and cross group. Nodes in the same deployment group are supposed to be deployed to the same point at the same time. Nodes in the same cross group have the same modulo number which is computed from its ID. After deployment, if two adjacent nodes in the same deployment group or cross group, they can establish secure link with their direct key establishment phase. Otherwise, they should perform the path key establishment phase to obtain a pairwise key. Zhou et al. [14] adopted different approach to propose another protocol. They argued that a node stores pairwise keys with all nodes in the same deployment group. If adjacent nodes can not establish pairwise keys directly, i.e., they are in the different deployment groups, they can associate by relative agent groups.

Compared to the random key pre-distribution protocols, group-based key pre-distribution protocols have better performance, higher connection probability and better resilience ability. Unfortunately, most protocols based on random key pre-distribution approach require non-constant storage requirement so that these protocol are not scalable.

2.3 Hierarchical structure protocols

In hierarchical structure protocols [16], [17], there are several gateways between the base station and sensor nodes. Gateways are specific nodes with higher ability that are responsible for perform more tasks than general nodes, so these gateway nodes will lighten the overheads of other general nodes. In 2003, Jolly et al. [16] proposed a low-energy key management protocol for WSNs. In their protocol, each node stores only two pairwise keys. One is used to communicate with the base station, and the other is used to establish secure communication with the nearest gateway. Before deployment, each gateway stores a partial set of keys which are also stored into nodes. After the network is deployed, each node exchanges its own ID and the gateway number with the nearest gateway. Afterwards, these gateways interchange information each other and then they obtain gateway keys of nodes located in its range from other gateways. The proposed protocol offers the network's entire connectivity. But if gateways are captured, it will reveal the large amount of gateway keys and seriously endanger the security.

In 2007, Cheng and Agrawal [17] presented an improved key distribution mechanism (IKDM) by using the bivariate polynomial developed by Blundo et al. [26]. Each gateway does not directly store nodes' gateway keys, but it stores bivariate polynomial functions. After deployment, a node sends its ID and the gateway numbers to the nearest gateway. Then, the gateway asks other gateways with knowing the gateway numbers to obtain sub-keys, and the gateway can compute the gateway keys of neighboring nodes from these sub-keys. As a result, each node could communicate with its gateway. Roughly, hierarchical structure has better performance, but gateways will become hot points which the adversaries prior attack.

2.4 Other protocols

Cheng and Agrawal [18] proposed an efficient pairwise key establishment and management protocol (EPKEM). In their protocol, the system generates a large two-dimensional key matrix and then each node is pre-distributed a row and a column of the global matrix before deployment. After deployment, two adjacent nodes exchange the numbers of their own row and column to find intersecting elements, and then combine intersecting elements and their node ID to generate the pairwise key. Because all pairwise keys are distinct in this protocol, it can withstand the compromise attack with captured nodes and guarantee entire connectivity. But each node requires $O(\sqrt{n})$ keys so that it is not scalable.

Chan and Perrig [19] presented peer intermediaries for key establishment protocol (PIKE). Each node with an identity of the form *(x,y)*. A node solely shares a pairwise key with each node which has the same *x*-coordinate or *y*-coordinate. After deployment, two adjacent nodes have the pairwise key if their identities are half match, or they can route a key with an intermediary node. Although this protocol has some improved methods, it also needs to establish path-keys which have some weaknesses in random key pre-distributed protocols.

In 2003, Zhu et al. [20] proposed localized encryption and authentication protocol (LEAP). They artfully use $\{f_k\}$ a family of pseudo-random functions [27] to authenticate node and derive pairwise key. Each node *i* pre-load the common initial key K_I and then derives each master key $K_i = f_{K_I}(i)$ by a pseudo-random function f_{K_I}. After deployment, each node *i* broadcasts a HELLO message which has its identity *i* and a nonce. A node *j* who receives the HELLO message replies its identity *j* and a message authentication code (MAC) using its master key K_j. The transmitting node uses initial key K_I to derive the master key K_j and then authenticates the node *j*. Further, they use $K_{ij} = f_{K_j}(i)$ to be their pairwise key. Although they assume that an adversary compromising a sensor node needs more time than nodes complete neighbor discovery phase and all nodes remove the initial key K_I after neighbor discovery phase. It has a critical problem: if the initial key K_I is revealed, whole network will be broken. Therefore, they presented an extended scheme [21] which divides the system lifetime into several particular periods. A node which be deployed at period *k* is pre-loaded the initial key K_{I_k} and all master keys after that period. The initial key K_{I_k} is used to authenticate and establish pairwise keys with neighbors when the node is deployed. Extra master keys are used to establish pairwise keys at remaining periods. This improvement decreases the fraction of communication links when any initial key is revealed.

3 Preliminaries

In this section, we first list some notations that are used throughout the paper. Furthermore, we present the improved ECC protocol based on the two-party key exchange protocol [8] and then security analysis of the improved protocol is discussed.

3.1 Notations

Let EC be an additive cyclic group with a prime order *q*. EC is a subgroup of the group of points on an elliptic curve over a finite field $E(F_p)$. Let *G* be a generator of the group EC. We refer to [28], [29] for a fuller description of how the group and other parameters should be selected in practice for efficiency and security. In fact, ECC has shorter key size and faster computation time for the same secure level with other public key systems. The following system parameters and notations are used throughout the paper.

- *G*: a generator of the group EC.
- *q*: the order of *G*.
- *n*: the amount of sensor nodes in the system.
- s_{RS}: the master private key of the registration server RS.
- P_{RS}: the public key of the registration server RS such that $P_{RS} = s_{RS} \cdot G$.
- ID_i: the identity of sensor node *i*.
- s_i: the secret key of senor node *i*.
- P_i: the public key of sensor node *i*.
- δ_i: the number of all one-hop adjacent nodes of node *i*.
- δ: the average number of one-hop adjacent nodes of all nodes in the sensor network.
- *x(Q)*: the *x*-coordinate of a point *Q* on the EC.
- *y(Q)*: the *y*-coordinate of a point *Q* on the EC.
- *h()*: a one-way hash function.
- sk_{ij}: the established session key between node *i* and node *j*.

3.2 Improved ECC two-party key agreement protocol

Here, we present the improved ECC protocol which is based on the two-party key exchange protocol [8]. There are two phases in the improved protocol: the key issuing phase and the key agreement phase. Without loss of generality, let RS be a registration server. Initially, the registration server RS creates an elliptic curve group EC and chooses a generator G of EC. RS then selects the master private key s_{RS} and computes the corresponding public key $P_{RS}= s_{RS} \cdot G$. Each node i is assigned a pair of public and secret keys from RS. The detailed description for the protocol as follows:

Key Issuing Phase. In this phase, a node i submits its identity ID_i to RS and then RS performs following steps:

Generate a random integer $t_i \in Z_q^*$.

Computes $P_i=t_i \cdot G$ and $s_i=t_i+ s_{RS} \cdot h(ID_i||x(P_i))$ mod q.

RS issues G, q, P_{RS} , P_i and s_i to the node, where P_i and s_i are the public key and the secret key, respectively.

Key Agreement Phase. Figure 1 depicts the procedures of this protocol. Assumed that nodes i and j are two communication nodes. Thus, nodes i and j have the key pairs ($P_i=t_i \cdot G$ and $s_i=t_i+ s_{RS} \cdot h(ID_i||x(P_i))$ mod q) and ($P_j=t_j \cdot G$ and $s_j=t_j+ s_{RS} \cdot h(ID_j||x(P_j))$ mod q) ,respectively. Thus, nodes i and j can carry out the following steps to generate the session key shared between them.

Step 1: The node i generates a random integer $r_i \in Z_q^*$ and computes $V_i=r_i \cdot G$. Then i uses its secret key s_i to compute $w_i=r_i+s_i \cdot x(V_i)$ mod q, and then sends V_i, P_i and ID_i to node j.

Step 2: The node j also generates a random integer $r_j \in Z_q^*$ and computes $V_j=r_j \cdot G$. Then node j uses its secret key s_j to compute $w_j=r_j+s_j \cdot x(V_j)$ mod q, and then sends V_j, P_j and ID_j to node i.

After performing the above steps, nodes i and j can compute the session key shared between them. The node i computes K_{ij} as follows:

$$Z_j=P_j+h(ID_j||x(P_j)) \cdot P_{RS}=t_j \cdot G+[h(ID_j||x(P_j)) \cdot s_{RS}] \cdot G=[t_j+s_{RS} \cdot h(ID_j||x(P_j)] \cdot G=s_j \cdot G \quad (1)$$

and

$$K_{ij}=w_i \cdot (V_j+x(V_j) \cdot Z_j)=w_i \cdot (r_j \cdot G+(x(V_j) \cdot s_j) \cdot G)=w_i \cdot (r_j +s_j \cdot x(V_j)) \cdot G=(w_i \cdot w_j) \cdot G \quad (2)$$

Meanwhile, the node j also computes K_{ji} as follows:

$$Z_i=P_i+h(ID_i||x(P_i)) \cdot P_{RS}=t_i \cdot G+[h(ID_i||x(P_i)) \cdot s_{RS}] \cdot G=[t_i+s_{RS} \cdot h(ID_i||x(P_i)] \cdot G=s_i \cdot G \quad (3)$$

and

$$K_{ji}=w_j \cdot (V_i+x(V_i) \cdot Z_i)=w_j \cdot (r_i \cdot G+(x(V_i) \cdot s_i) \cdot G)=w_j \cdot (r_i +s_i \cdot x(V_i)) \cdot G=(w_j \cdot w_i) \cdot G \quad (4)$$

It is clear that nodes i and j compute the same point $K=K_{ij}=K_{ji}=(w_j \cdot w_i) \cdot G$. In this case, they can obtain the common session key as follows

$$sk_{ij}=h(x(K_{ij})||y(K_{ij}))=h(x(K_{ji})||y(K_{ji}))= sk_{ji}. \quad (5)$$

Node i		Node j
Randomly choose r_i $V_i=r_i \cdot G$ $w_i=r_i+s_i \cdot x(V_i)$ mod q	(V_i,P_i,ID_i) ——>	Randomly choose r_j $V_j=r_j \cdot G$ $w_j=r_j+s_j \cdot x(V_j)$ mod q
	<—— (V_j,P_j,ID_j)	
$Z_j=P_j+h(ID_j\|\|x(P_j)) \cdot P_{RS}$ $K_{ij}=w_i \cdot (V_j+x(V_j) \cdot Z_j)$ $sk_{ij}=h(x(K_{ij})\|\|y(K_{ij}))$		$Z_i=P_i+h(ID_i\|\|x(P_i)) \cdot P_{RS}$ $K_{ji}=w_j \cdot (V_i+x(V_i) \cdot Z_i)$ $sk_{ji}=h(x(K_{ji})\|\|y(K_{ji}))$

Fig. 1. Improved ECC key agreement protocol.

Security Analysis. In the following, we briefly discuss the security of the proposed protocol. The security of the proposed protocol is based on the Computational Diffie-Hellman (CDH) assumption, i.e., given G, $x \cdot G$, $y \cdot G$, finding $xy \cdot G$ is hard. Another is Elliptic Curve Discrete Logarithm Problem (ECDLP), i.e., given G and $x \cdot G$, finding x is hard. Based on the Computational Diffie-Hellman (CDH) assumption and Elliptic Curve Discrete Logarithm Problem (ECDLP), we show that the proposed protocol offers implicit key authentication, known-key security and full forward secrecy.

Implicit key authentication: A key agreement protocol offers implicit key authentication if a node i is believed that node j can compute the session key and no one other than node j can compute the session key. Any one who gets V_i, P_i, ID_i, V_j, P_j and ID_j which were transmitted between two nodes can easily compute $w_i \cdot G$ and $w_j \cdot G$. An adversary is hard to compute $(wj\,wi) \cdot G$ from $w_i \cdot G$ and $w_j \cdot G$, because only nodes i and j have the secret key s_i and s_j to compute w_i and w_j, respectively.

Known-key security: If the session key sk_{ij} is disclosed, attackers cannot find the inter-computed key K_{ij} because it is protected by a one-way hash function. Furthermore, even if the inter-computed key K_{ij} of the session key sk_{ij} is disclosed, the protocol still withstands the known-key attack. Suppose that the adversary has known a pre-key K_1 established between i and j. Since $K_1=(w_{j1}\,w_{i1}) \cdot G$, we have

$$K_1 = (w_{j1}\,w_{i1}) \cdot G = (r_{i1}+s_i\,x(V_{i1})) \cdot (r_{j1}+s_j\,x(V_{j1})) \cdot G = (r_{i1}\,r_{j1} + r_{i1}\,s_j\,x(V_{j1}) + r_{j1}\,s_i\,x(V_{i1}) + s_i\,x(V_{i1})\,s_j\,x(V_{j1})) \cdot G \quad (6)$$

Suppose that there is another value K_2 established between nodes i and j now. At the same reason, we have $K_2 = (r_{i2}\,r_{j2} + r_{i2}\,s_j\,x(V_{j2}) + r_{j2}\,s_i\,x(V_{i2}) + s_i\,x(V_{i2})\,s_j\,x(V_{j2})) \cdot G$. However, because K_1 consists of four items $(r_{i1}\,r_{j1}) \cdot G$, $r_{i1}\,s_j\,x(V_{j1}) \cdot G$, $r_{j1}\,s_i\,x(V_{i1}) \cdot G$ and $s_i\,x(V_{i1})\ s_j\,x(V_{j1})) \cdot G$, and each item's coefficient contains two unknown values, thus the adversary is unable to obtain the valid information from K_1. Therefore, the adversary does not find another session key K_2 from K_1, so the proposed scheme can withstand the known-key attack.

Full forward secrecy: A key agreement protocol offers full forward secrecy if the compromise of both nodes' secret keys cannot result in the compromise of previously established session keys. Suppose that the adversary has obtained both secret keys and tries to compute w_i or w_j in order to compute $K_{ij}=(w_i\,w_j) \cdot G$. To find w_i or w_j must require to know r_i or r_j from V_i or V_j, respectively. Thus, this will be equivalent to solving ECDLP. Moreover, because the session key K_{ij} includes the value of $(r_i\,r_j) \cdot G$, it is still unknown to the adversary. Therefore, the proposed protocol can provide full forward secrecy.

4 ID-based pairwise key establishment protocol for wireless sensor networks

In this section, we present an ID-based pairwise key establishment protocol for WSNs by adopting the improved ECC key agreement protocol described in Section 3. As we all know, sensor nodes communicate with each other by broadcasting messages like Ad hoc manner. Here, we assume that the network structure is a planar form and sensor nodes are static that it means nodes do not move after deployment.

In the improved ECC key agreement protocol, we requires a registration server RS which issues keys to each node. The sensor manufacturer can be viewed as the role of the registration server RS to issue keys to every node in the WSN. For convenience, we denote the sensor manufacturer as RS. Therefore, the RS creates an EC group and a generator G. Then the RS selects the master private key s_{RS} and computes the corresponding public key $P_{RS}= s_{RS} \cdot G$. In the following, we present our protocol which consists of four phases: (1) the manufacture phase (2) the network deploying phase (3) the re-keying phase (4) the adding new nodes phase.

Manufacture Phase. In this phase, RS performs the same procedures of "Key issuing phase" described in Section 3 to issue each sensor node's secret key s_i and public key P_i. Meanwhile, RS repeatedly performs Step 1 in "Key agreement phase" described in Section 3 to compute several pairs (r_i, V_i, w_i) into each sensor node i in order to efficiently reduce energy consumption for establishing session keys between sensor nodes. Note that the number of the pairs is dependent on the storage of nodes. Certainly, the number of the preloaded pairs is also dependent on the system requirement such as the usage time period of sensor nodes. According to the description of "Key agreement phase" described in Section 3, the computation cost of one pair (r_i, V_i, w_i) requires two point scalar multiplications on the EC and one point addition on the EC. Therefore, it will significantly reduce energy consumption of sensor nodes.

Network Deploying Phase. Without loss of generality, each node i needs to establish secure link with other one-hop adjacent nodes j ($1 \leqq j \leqq \delta_i$), where δ_i is the number of all one-hop adjacent nodes of node i. The details are described as follows:

(1) Node i randomly chooses a pair (r_i, V_i, w_i) from its storage.

(2) Node i broadcasts a message (V_i, P_i, ID_i) to all one-hop adjacent nodes j.

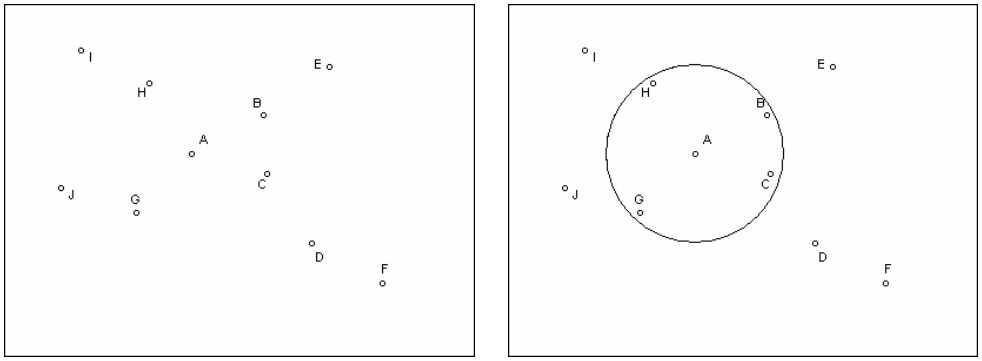
(3) After receiving every one-hop adjacent node j's broadcasting message (V_j, P_j, ID_j), the node i performs the following computing:

```
For j=1 to δi
    Zj:=Pj+h(IDj||x(Pj))·PRS
```

```
    Kij=wi·(Vj+x(Vj)·Zj)
    skij=h(x(Kij)||y(Kij))
end for.
```

After performing the above computations, each node *i* will establish all session keys sk_{ij} with each one-hop adjacent node *j*. Note that nodes are static after deployment, so each node can store other adjacent nodes' intermediate variable Z_j to reduce computation significantly when the system needs to execute the re-key phase.

For example, Figure 2(a) shows that some nodes are deployed in a scope and Figure 2(b) shows communication range of node *A*, i.e., nodes *B*, *C*, *G*, *H* can receive HELLO message which is broadcasted by node *A*. Node *A* can alike receive HELLO messages which are broadcasted by these nodes. Therefore, node *A* and its one-hop adjacent nodes can establish pairwise keys.



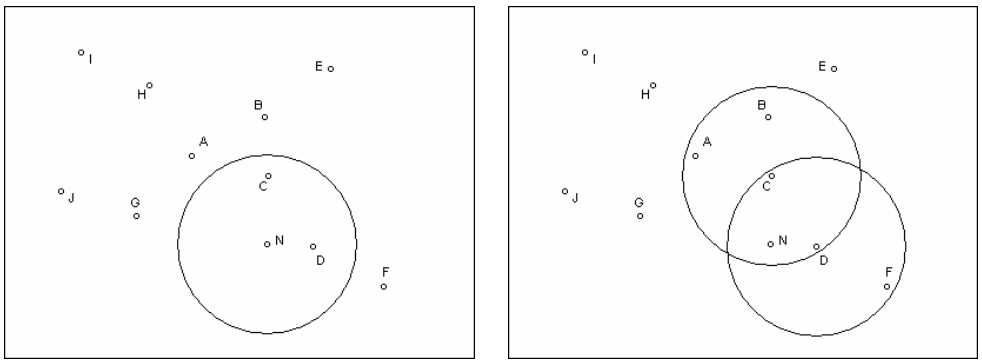
(a) A simple wireless sensor network. (b) Communication range of node A.

Fig. 2. Diagrams of a WSN

Re-keying Phase. By the development of advanced chip technique recently, the lifetime of sensor nodes is longer and longer and the nodes need to fresh their session keys to enhance security in some scenarios. In our protocol, we provide a simple re-keying mechanism. When the system periodically executes the re-keying phase, each node *i* only chooses another pair (r_i, V_i, w_i) from its storage and broadcasts (V_i, ID_i) to adjacent nodes *j*. After receiving all messages from adjacent nodes, the node *i* computes only K_{ij} and sk_{ij} for each adjacent node *j*. This is because each node has stored other adjacent nodes' intermediate variable Z_j in "Network deploying phase", it will reduce the computations of the re-keying phase.

Adding New Nodes Phase. In some applications, the system will add new nodes into the sensor network to replace the captured nodes or dead nodes to ensure the network working. Most protocols either do not provide this phase or require complex processes to handle this situation. In our protocol, only the added new nodes and its one-hop adjacent nodes perform the "Network deploying phase" to establish the secure communications between them.

For example, if node *N* is a new node which is deployed in the network. It broadcasts a HELLO message (V_n, P_n, ID_n) to nodes *C* and *D* such as Figure 3(a). Then nodes *C* and *D* reply HELLO message to node *N* to establish pairwise keys. Note that other nodes *A*, *B* and *F* will ignore HELLO message from *C* and *D*, because they are finished "Network deploying phase" and re-keying message is only two elements which without public key.



(a) Node N broadcasts HELLO message. (b) Nodes C and D reply to node N.

Fig. 3. Adding new node to the network

5 Performance analysis

As the performance analysis, we evaluate the computational complexity and the bit size of the message required in our protocol. Due to our protocol only requires one broadcast message, we omit the transmission round. And we ignore some light-weight operations which include modular addition in Z_q and concatenating strings. As we all know, they are much smaller than the following costly operations. For convenience, the following notations are used to analyze the computational cost.

- TG_{mul}: the time for point scalar multiplication on the EC.
- TG_{add}: the time for point addition on the EC.
- T_h: the time of executing the one way hash function $h()$.
- T_{mul}: the time for modular multiplication in Z_q.

Table 1 summarizes the performance result of the proposed protocol in terms of the network deploying phase and the re-keying phase. It decreases $2TG_{mul}+TG_{add}$ by pre-loading some pairs (r_i, V_i, w_i) of each node i. By remembering one-hop adjacent nodes' intermediate variable Z_j, it decreases $3TG_{mul}+2TG_{add}$ in the re-keying phase. From Table 1, we know that our protocol significantly reduces energy consumption because we adopt the improved ECC key agreement into our protocol by using characteristics of a wireless sensor network. Some previous implementations [3], [4], [5], [6], [7] of elliptic curve cryptographic primitives on microprocessors can give an evidence to demonstrate that our protocol is well suited for sensor nodes with limited computing capability. There is a simulation experiment on ECC by Wander et al. [5]. In their result, a node performs Elliptic Curve Diffie-Hellman (ECDH) key exchange algorithm needs *22.3mJ*. As we all know, ECDH key exchange algorithm includes two EC point scalar multiplications. Therefore, we use the double energy consumption, *44.6mJ*, to be the energy consuming upper bound of our protocol. Even if the average number of adjacent nodes, $\delta=50$, each node only needs around *2J* energy consumption for network deploying phase. Moreover, it requires less energy consumption at re-keying phase. In addition, performance can be significantly improved if sensor nodes can equip with a hardware coprocessor such as [30], [31].

Table 1. Performance evaluation of the proposed protocol

	Computational complexity	Bit size of the message
Network deploying phase	$\delta(3TG_{mul}+2TG_{add}+2T_h)$	$2\|G\|+\|ID\|$*
Re-keying phase	$\Delta(2TG_{mul}+TG_{add}+T_h)$	$\|G\|+\|ID\|$*

*$|G|$ and $|ID|$ denote the bit lengths of an EC point and a node identity, respectively.

6 Discussions

In this section, we compare our protocol with random key pre-distribution protocols [9], [10], [11], group-based key pre-distribution protocols [12], [13], [15], the IKDM [17] and the EPKEM protocol [18]. These previously proposed protocols have been reviewed in Section 2. We omit the discussions on other protocols because we

have reviewed them in Section 2. Table 2 summarizes the comparisons among our protocol and previously proposed protocols in terms of storage requirement, scalable property and entire connectivity property, as well as several security properties. From Table 2, it is obvious that our protocol has better properties that include requiring only constant memory storage, offering entire connectivity and scalable. Our protocol can withstand the compromise attack with captured nodes and the known-key attack. Although our protocol needs more energy consumption to perform secure operations, recently researches [3], [4], [5], [6], [7] have shown that sensor nodes are more and more capable to perform these tasks. In the following, we discuss the properties of our protocol in detail.

Storage requirement and scalable: Because sensor nodes have limited storage, the storage requirement of each node is a critical consideration for the secure protocol design in WSN. In our protocol, each node requires only constant memory storage. It is independent on others factors such as deploying density of network nodes, security level and the number of sensor nodes. That means that our protocol has perfect scalable property due to constant storage requirement.

Entire connectivity: Entire connectivity property means that each node can set up a secure link with every one-hop adjacent node. In our protocol, each node can establish a session key with every one-hop adjacent node by our improved ID-based key agreement protocol. This implies our protocol offers the entire connectivity property. Note that the session key of any two adjacent nodes is directly established without the assistances of others sensor nodes. It means that our protocol does not use the path key technique to establish the session key.

Against compromise attack with captured nodes: Compromise attack with captured nodes is a serious problem which popularly exists in the key pre-distribution protocols. It is defined as how much non-captured nodes' information the adversary can get from captured nodes. In other word, the adversary can compromise non-captured nodes by data obtained from captured nodes. Because our protocol is based on the ECC public key system, individual sensitive information does never store to other nodes. This means the adversary can not compromise non-captured nodes with captured nodes.

Table 2. Property comparisons among our protocol and the previously proposed protocols

	Random key protocols [9], [10], [11]	Group-based protocols [12], [13], [15]	IKDM [17]	EPKEM [18]	Our protocol
Storage requirement	Ω(n)	Conditional	Constant	$O(\sqrt{n})$	Constant
Scalable	No	No	Yes	No	Yes
Entire connectivity	No	No	Yes	Yes	Yes
Against compromise attack with captured nodes	Low	Medial	High	Absolute	Absolute
Withstanding known-key attack	No	No	No	No	Yes
Full forward secrecy	No	No	No	No	Yes

7 Conclusions

We have proposed a practical ID-based pairwise key establishment protocol for wireless sensor networks based on elliptic curve cryptography (ECC). Compared to the previously proposed protocols for WSNs, our protocol has the following merits: (1) each sensor node requires only constant memory storage, so our protocol offers scalable property; (2) by using ID-based ECC public key system, the proposed protocol ensures that each sensor node can directly establish secure communications with other adjacent nodes, so it guarantee the network's entire connectivity; (3) even the message broadcast by each node requires one message, any two adjacent nodes can still establish a session key without other nodes' assistances. Moreover, we also discuss the operations of adding new

nodes and re-keying phases. We have shown that the proposed protocol offers implicit key authentication and full forward secrecy. It withstands the known-key attack and the compromise attack with captured nodes.

Acknowledgement

This research was partially supported by National Science Council, Taiwan, R.O.C., under contract no. NSC94-2213-E-018-009 and NSC95-2221-E-018-010.

References

[1] N. Xu, "A survey of sensor network applications," *Survey Paper for CS694a, Computer Science Department, University of Southern California*, http://enl.usc.edu/~ningxu/papers/survey.pdf.

[2] I. Khemapech, I. Duncan and A. Miller, "A survey of wireless sensor networks technology," *Proceedings of the 6th Annual PostGraduate Symposium on the Convergence of Telecommunications, Networking & Broadcasting (PGNET'05)*, pp. 26-31, June 2005.

[3] D. J. Malan, M. Welsh and M. D. Smith, "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography," *Proceedings of the first IEEE communications society conference on Sensor and Ad Hoc Communications and Networks (SECON'04)*, pp. 71-80, October 2004.

[4] N. Gura, A. Patel, A. Wander, H. Eberle and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs," *Proceedings of Workshop on Cryptographic Hardware and Embedded Systems (CHES'04)*, pp. 119-132, August 2004.

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

[6] E. O. Bla and M. Zitterbart, "Towards acceptable public-key encryption in sensor networks," *Proceedings of the 2nd International Workshop on Ubiquitous Computing (IWUC'05)*, pp. 88-93, May 2005.

[7] K. Piotrowski, P. Langendoerfer and S. Peter, "How public key cryptography influences wireless sensor node lifetime," *Proceedings of the 4th ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'06)*, pp. 169-176, October 2006.

[8] Y. M. Tseng, "An efficient two-party identity-based key exchange protocol," *Informatica: International Journal*, Vol. 18, No. 1, pp. 125-136, 2007.

[9] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," *Proceedings of the 9th ACM Conference on Computer and Communication Security (CCS'02)*, pp. 41–47, November 2002.

[10] H. Chan, A. Perrig and D. Song, "Random key predistribution schemes for sensor networks," *Proceedings of IEEE Symposium on Security and Privacy (SP'03)*, pp. 197–213, May 2003.

[11] W. Du, J. Deng, Y.S. Han and P.K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)*, pp. 42-51, October 2003.

[12] W. Du, J. Deng, Y. S. Han, S. Chen and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," *Proceedings of the 23th IEEE Conference on Computer Communications (INFOCOM'04)*, pp. 586-597, March 2004.

[13] D. Liu, P. Ning and W. Du, "Group-based key pre-distribution in wireless sensor networks," *Proceedings of the 4th ACM Workshop on Wireless Security (WiSe'05)*, pp. 11-20, September 2005.

[14] L. Zhou, J. Ni and C. V. Ravishankar, "Efficient key establishment for group-based wireless sensor deployments," *Proceedings of the 4th ACM Workshop on Wireless Security (WiSe'05)*, pp.1-10, September 2005.

[15] W. Du, J. Deng, Y. S. Han and P. K. Varshney, "A key predistribution scheme for sensor networks using deployment knowledge," *IEEE Transactions on Dependable and Secure Computing*, Vol. 3, Issue 1, pp. 62-77, Jan-March 2006.

[16] C. Jolly, M. C. Kuscu, P Kokate and M. Younis, "A low-energy key management protocol for wireless sensor networks," *Proceedings of the 8th IEEE International Symposium on Computers and Communications (ISCC'03)*, pp. 335-340, June 2003.

[17] Y. Cheng and D.P. Agrawal, "An improved key distribution mechanism for large-scale hierarchical wireless sensor networks," *Ad Hoc Networks Journal*, Vol. 1, No. 1, pp. 35-48, January 2007.

[18] Y. Cheng and D. P. Agrawal, "Efficient pairwise key establishment and management in static wireless sensor networks," *Proceedings of the 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS'05)*, pp. 544-550, November 2005.

[19] H. Chan and A. Perrig, "PIKE: Peer intermediaries for key establishment in sensor networks," *Proceedings of the 24th IEEE Conference on Computer Communications (INFOCOM'05)*, pp. 524-535, March 2005.

[20] S. Zhu, S. Setia and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)*, pp. 62-72, October 2003.

[21] S. Zhu, S. Setia and S. Jajodia, "LEAP+: efficient security mechanisms for large-scale distributed sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, Vol. 2, Issue 4, pp. 500-528, November 2006.

[22] R. Blom, "An optimal class of symmetric key generation systems," *Proceedings of EUROCRYPT'84*, LNCS 209, pp. 335–338, 1985.

[23] A. Wacker, T. Heiber and H. Cermann, "A key-distribution scheme for wireless home automation networks," *Proceedings of IEEE Consumer Communications and Networking Conference (CCNC'04)*, pp. 47-52, January 2004.

[24] A. Wacker, M. Knoll, T. Heiber and K. Rothermel, "A new approach for establishing pairwise keys for securing wireless sensor networks," *Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys'05)*, pp. 27-38, November 2005.

[25] J. Hwang and Y. Kim, "Revisiting random key pre-distribution schemes for wireless sensor networks," *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04)*, pp. 43-52, October 2004.

[26] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung, "Perfectly-secure key distribution for dynamic conferences," *Proceedings of CRYPTO'92*, LNCS 740, pp. 471–486, 1993.

[27] O. Goldreich, S. Goldwasser and S. Micali, "How to construct random functions," *Journal of the ACM (JACM)*, Vol. 33, Issue 4, pp. 792-807, October 1986.

[28] V. S. Miller, "Use of elliptic curves in cryptography," *Proceedings of CRYPTO'85*, LNCS 218, pp.417-426, 1986.

[29] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, Vol. 48, No. 177, pp. 203-209, 1987.

[30] G. Bertoni, L. Breveglieri and M. Venturi, "Power aware design of an elliptic curve coprocessor for 8 bit platforms," *Proceedings of the 4th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*, pp. 337-341, March 2006.

[31] G. Bertoni, L. Breveglieri and M. Venturi, "ECC hardware coprocessors for 8-bit systems and power consumption considerations," *Proceedings of the third International conference on Information Technology: New Generations (ITNG'06)*, pp. 573-574, April 2006.