International Journal of Innovative Computing, Information and Control
Volume **x**, Number **0x**, x **2008**

pp. **0–0**

AN EFFICIENT SENSOR-TO-SENSOR AUTHENTICATED PATH-KEY ESTABLISHMENT SCHEME FOR SECURE COMMUNICATIONS IN WIRELESS SENSOR NETWORKS

Chun-Ta Li[1,2]

Department of Information Management[1]
Tainan University of Technology
529 Jhong Jheng Road, Yongkang, Tainan, Taiwan 710, R.O.C.
Department of Computer Science and Engineering[2]
National Chung Hsing University
250 Kuo Kuang Road, Taichung, Taiwan 402, R.O.C.
th0040@mail.tut.edu.tw

Min-Shiang Hwang[3]

Department of Management Information Systems[3]
National Chung Hsing University
250 Kuo Kuang Road, Taichung, Taiwan 402, R.O.C.
mshwang@nchu.edu.tw

Yen-Ping Chu[4]

Department of Computer Science and Information Engineering[4]
Tunghai University
181 Section 3, Taichung Harbor Road, Taichung, Taiwan 407, R.O.C.
ypchu@nchu.edu.tw

Abstract. *Path-key establishment has become accepted as a commonly used solution in wireless sensor networks (WSNs) for protecting node-to-node communications from malicious attacks. Unfortunately, traditional security approaches are not well suited to WSNs due to their limited computational/communication abilities and memory, and their vulnerable-to-attack structure. Moreover, to extend lifetime and usability of sensor networks, power conservation and scalability are required in the design of sensor network schemes. In this paper, we propose an efficient sensor-to-sensor authenticated path-key establishment (ES2S-APKE) scheme for wireless sensor networks. ES2S-APKE accomplishes node authentication and pairwise key establishment by applying well-known Elliptic Curve Cryptography (ECC) and using cluster-based sensor groups. In clustered sensor networks, a back-end system creates a view of the credential authority (CA) and provides credential update service for all involved nodes in the network, including sink nodes and sensor nodes. A ticket scheme is introduced to provide efficient S2S path-key establishment service. Finally, the security and performance of our proposed ES2S-APKE is compared with Lee's* [19] *and Varadharajan's* [32] *schemes.*

Keywords: Ad hoc networks, elliptic curve cryptography, mutual authentication, path key establishment, security, wireless sensor networks.

1. Introduction.

Wireless sensor networks are formed dynamically by a number of sensor nodes. In a wireless sensor network, when sensors deploy in a designated area, they must pass an identity authentication examination by their corresponding sink nodes in order to identify both trustworthy and unreliable nodes from a security standpoint. Through this identity authentication process, the controller node (also called sink node) can determine if the sensor information can be trusted and unauthorized nodes can be isolated from networks during the identity authentication procedure. After a sensor passes the identity authentication check of a sink node, the packets transmitted between a sensor and the sink node must be kept secret while a sensor sends its data. They must establish a session key to be used between them for securing their subsequent communications.

As sensors have limited resources and are expensive to install, computational and communication overhead must be kept at a minimum. Hence the proposed scheme must be devised with a view towards minimizing power consumption and ensuring that the sensor nodes in the network are dormant. For example, some sensor nodes in a period of dormancy might not reply to the sensed messages in order to extend their usage life. Moreover, compared with traditional communication networks, some characteristic considerations surrounding wireless sensor networks are discussed and addressed in the design of WSNs, including: non-centralized architecture, self-organization, and dynamic topology etc.

In the following, we identify general considerations regarding WSNs currently of interest to researchers, introduce four main security threats to WSNs caused by active intruders, and finally state several key goals of our proposed ES2S-APKE scheme.

1.1. General Considerations of WSNs.

In this section, we will briefly introduce the essential criteria that a wireless sensor network should satisfy the following considerations [1, 10, 14, 23, 25, 30]:

- **Hardware constraints**: These regard the physical properties and many constraints on proposed hardware; for example, that of limited energy. In addition, due to limited volume of the sensor, some sensors have physical constraints on the amount of storage, bandwidth, energy, and limited computational ability they can provide.
- **Communication types**: The existing routing protocols show that there are three main types of communications in WSNs; including direct, clustering-based, and multi-hops communication. In direct communication, every sensor node transmits its sensor data to a sink node and the sink node is responsible for collecting these data to back-end manager node for further processing. In clustering communication, all sensor nodes are divided into several groups and each cluster head node is responsible for collecting data within its group. Multi-hops communication is used because the communication range of a sensor is assumed to be limited and the intermediate nodes maybe used for transmitting the communication packets to each other on their path between the source node and the destination node.
- **Scalability**: Another consideration is the scalability of sensor networks. In this case, networking must continue functioning regardless of the number of sensors placed.
- **Fault tolerance**: Due to the environmental influence on sensors, many contingencies have been addressed in sensor networks; for example, sensors may crash or power failures might occur. Such problems must be avoided by fault tolerance strategies to maintain functionality.
- **Power saving**: When sensors are distributed to monitor environments of interest, these sensors may be required to function over a span of several weeks, even several months. Therefore, how to provide a mechanism of saving power (such as intermittent rest periods) to extend lifespan is highly important. In general, too much power is consumed during the message transmitted phase.
- **Production costs**: Depending on the orientation of the sensor network, a large number of sensors might be scattered randomly over a given environment; such as in the case of weather monitoring. If the overall cost is appropriate for sensor networks and it will be more acceptable and successful to users which need careful consideration. As a result, tamper protection is usually unavailable for sensor nodes and malicious persons may extract compromising cryptographic information from a sensor node.

1.2. **Security Threats of WSNs.** In this subsection, we introduce the four main security threats to WSNs, including: node replication, impersonation, wormhole, and compromised node attacks. Since wireless sensor networks are easily compromised by intruders due to their lack of tamper resistance, four types of attacks are anticipated.

- **Node replication attack**: In node replication attacks, when a sensor node is compromised by an intruder, he/she can directly place many replicas of this compromised node at different areas within the networks. Thus, the intruder may use these compromised nodes to subvert the network functionalities, for example by injecting false data.
- **Impersonation attack**: These kinds of attacks are also called sybil attcks [22]. In sybil attacks, a sensor node can illegitimately claim multiple identities by either directly forging false identities, or else impersonating legal identities. This harmful attack may lead to serious threats to distributed storage, routing algorithms, data aggregation, and voting protocols.
- **Wormhole attack**: For wormhole attacks [26], the malicious node may be located within transmission range of legitimate nodes while legitimate nodes are not themselves within transmission range of each other. Thus, the malicious node can tunnel control traffic between legitimate nodes and nonexistent links which in fact are controlled by the malicious node. Later, the malicious node can drop tunnelled packets or carry out attacks on routing protocols.
- **Compromised node attack**: For the internal attacks, due to the lack of tamper resistance in sensor nodes, an intruder may compromise a sensor node and use it to establish communication channels with non-compromised sensors to launch other more serious attacks within the sensor network.

1.3. **The Goals of Our ES2S-APKE Scheme.** According to the above-mentioned requirements of WSNs, in the following, some of the general goals and security issues that should be treated in WSNs. Our scheme aims to achieve the following requirements for evaluating new path-key establishment schemes in WSNs:

- **Mutual authentication**: The scheme can achieve mutual authentication during the proposed sensor identification phase and two paradigms of path-key establishment phases. Not only can sink nodes and valid sensors verify each other every session, but two communication sensors can verify each other.
- **Availability**: The availability of the network should not be affected even if sensors can only provide limited storage, limited power, and limited computational ability. Therefore, a mechanism regulating of sleep patterns is necessary for a sensor to extend its lifetime.
- **Location awareness**: The damage cannot be spread from the victimized area to the entire network by security attacks even if the sensor node is compromised. A secure communication scheme must limit the damage's scope caused by the intruders; the mechanism of location awareness is used in our scheme for this purpose.
- **No verification table**: The verification tables are not required to be stored inside the sink nodes to prevent stolen-verifier attacks [31].
- **Confidentiality**: Path-key establishment in every session must be secure against malicious intruders even if those intruders collect transmission packets.
- **Perfect forward secrecy**: In a two-party path-key establishment, a scheme is said to have perfect forward secrecy if revealing of the secret key to an intruder cannot help him/her derive the session keys of past sessions.
- **Key revocation**: When the back-end system or the sink node decides to terminate a sensor utilizing task, or when a sensor is lost, the sensor must not be allowed to make use of the credential which it stores to connect to networks.
- **Re-keying**: By introducing a re-keying mechanism, a sink node can conveniently update a sensor's credential without the intervention of back-end system for the purpose of reducing the communication interactions and management burden on that back-end system.

- **Prevention of four main security threats**: As described in Section 1.2, the proposed scheme must resist four major security threats (Node replication/Impersonation/Wormhole/Compromised node attacks) in securing WSNs.

1.4. **Outline.** The remainder of this paper is organized as follows. Section 2 reviews related works dealing with path-key establishment in WSNs. Details of the proposed ES2S-APKE scheme are described in Section 3. A performance evaluation of the proposed approach and comparisons between related cluster-based schemes and our scheme are conducted in Section 4. Finally, some conclusions are given in Section 5.

2. **Related Works.** Recently, several path-key establishment schemes have been proposed to secure communications in wireless sensor networks [6, 8, 15, 28, 33, 34, 35]. A naive solution to this problem would be store a master key in all deployed sensor nodes and the storage overhead is light for each sensor node as it needs only one master key regardless of the network size. However, this solution fails to prevent security breaches and thus is impracticable for WSNs whose sensor nodes lack tamper resistance and are easy for an attacker to compromise, leaving all the secrets in those networks known to the attacker. In addition, the attacker may manipulate an internal compromised node to launch other malicious attacks, and thus the network is rendered useless by poor security. The other extreme solution is to store a set of $n-1$ pairwise keys in each sensor node before deployment in such a way that it shares a unique pairwise key with all other nodes in the network, where n is the number of sensor nodes in the network. However, this solution leads to a large amount of key storage that grows linearly with network size, and is therefore not scalable and impracticable due to the very limited local memory of a sensor node. Therefore, many key pre-distributed schemes have been proposed to reduce this storage requirement. Any two sensors will share some pre-loaded key information in order to establish a common path-key. However, if a small number of sensor nodes are compromised by an attacker and this problem may expose a large fraction of common path-keys between non-compromised nodes [20, 28]. As a result, unlike the conventional key pre-distribution schemes in [2, 3, 9, 12, 28] that attempt to reduce key storage before sensor node deployment. We propose a scalable scheme for key management in dynamic sensor networks without pre-loading key information of other sensor nodes to reduce the overhead of key storage and the damage resulting from compromised node attacks.

Also, security in a wireless sensor network is very different from that in classical mobile ad hoc networks. In order to reduce the power consumption of limited-resource sensors and enhance scalability [1, 10], cluster-based path-key establishment schemes [4, 11, 16, 24, 29] have been proposed for WSNs in particular. In clustered (hierarchical) WSNs, sensor nodes are typically organized into many clusters, with cluster controllers collecting sense data from ordinary sensor nodes in the managed cluster to the back-end system. Furthermore, compared to mobile ad hoc networks, when sensor nodes are randomly deployed in a designated area, they only infrequently move from one cluster to another, and thus mobility is not a critical issue in WSNs. However, after several weeks or months of deployment, some sensor nodes in the network may exhaust their power due to continued operation. Therefore, the sleep patterns are required to extend the availability of WSNs and re-key freshness should guarantee that each shared path-key (session key) is fresh [10]. This also implies that a sensor node joined in WSNs has been authenticated. The proposed scheme issue each sensor node a preloaded certificate which includes lifetime to set the bootstrapping time and identity information to prove itself, thus allowing it to be joined into the sensor network. As a result, our credential-based scheme can reduce the key storage requirements of limited-resource sensors and prevent invalid sensor nodes from joining the sensor networks at the very beginning. Moreover, the back-end system can arrange sensor nodes with different periods of bootstrapping time by updating the credential, leaving sensors operated or slept into the networks by turns. Finally, the problem related to lifetime extension of WSNs is solved by introducing a credential update procedure.

3. **ES2S-APKE: An Efficient S2S-APKE Scheme.** In our proposed ES2S-APKE scheme, a sample of system architecture for securing communications over a wireless sensor network (WSN) is given in Figure 1. A distributed structure of WSN consists of three types of participants, namely: a powerful

back-end system, sink nodes, and sensor nodes. In Figure 1, a sense field has 9 hexagonal grids of sensor nodes based on geographical and deployment knowledge. A number of sensor nodes are uniformly distributed into hexagonal grids and each grid has a designated grid controller called sink node, which plays a privileged role for sensor-to-sensor path-key establishment and is responsible for identifying the validity of sensor nodes in its managed grid. The back-end system will generate a grid-secret key for all sink nodes, and each sink node has its own secret key which is also issued by back-end system. Then, each sink node is responsible for collecting all sensed data of its managed grid to the back-end system for further processing from sensor nodes under the grid for which it is responsible. In addition, all sensor data transmitted between participants should be verified and protected in the scheme; and thus even if an intruder eavesdrops on the communications between nodes or injects illegal sensor data into networks, the scheme still provides an adequate level of security. Furthermore, the ES2S-APKE scheme also takes into account the hardware and resource constraints of sensor nodes by minimizing computational overhead without resorting to conventional asymmetric cryptographic solutions. We largely succeed in ensuring that the ES2S-APKE scheme maintains low computational overhead using Elliptic Curve Cryptography (ECC) [17, 21] for WSNs, as can been seen in a detailed description of our proposed ES2S-APKE scheme.

Motivated by the types of security threats mentioned in Section 1, our ES2S-APKE scheme accomplishes two main tasks. One is sensor node identification and the other is sensor-to-sensor path-key establishment. For sensor node identification, a deployed sensor node will prove its identity to its designated sink node and convinces the sink node that it is legal to participate in the sensor network during the sensor node identification phase. For sensor-to-sensor path-key establishment, a path-key is established by two communication nodes to protect communications. There are two types of sensor-to-sensor path-key establishment, namely: intra-grid communication and inter-grid communication. As a result, the ES2S-APKE scheme consists of five phases, namely: a predeployment phase, a sensor node identification phase, a credential update phase, a intra-grid path-key establishment phase, and a inter-grid path-key establishment phase. The goal of the sensor node identification phase is to identify whether or not a sensor node is allowed to participate the sensor network. Thus, an invalid outsider would be unable to send malevolent data into the networks and the sink node can confirm that received sensor data has come from a valid sensor node, not from malicious outsiders. In addition, the goal of the credential update phase is that a sink node can quickly update a sensor's credential without the help of a back-end system to reduce the communication steps between the back-end system, and to alleviate the burden of back-end system. Finally, the proposed ES2S-APKE scheme allows two sensor nodes to agree on a common path-key to protect later communications without adapting heavy computations; thus reducing power consumption on sensor nodes. In Figure 2, two general paradigms for intra-grid and inter-grid communications are presented and the details of ES2S-APKE scheme are described as follows.

Figure 1. The structure of WSN

3.1. **Notations and Assumptions.** Before presenting the details of our ES2S-APKE scheme, we must outline some basic notations and assumptions seen in Tables 1 and 2 and used throughout this paper. During sensor-to-sensor path-key establishment phase, an attacker may launch Sybil, node replication, wormhole, node compromised, or eavesdropping attacks in WSNs. However, we assume that the sink nodes and the back-end system in ES2S-APKE scheme are trusted entities and not captured by attackers.

3.2. **Predeployment Phase.** Before a number of sensor nodes are deployed, the BS preloads each sensor node with a set of sensor node parameters which stores the elliptic curve $E(\mathbf{F}_q)$, the cyclic group

Table 1. Notations

$(\mathbf{G}, P)$	A cyclic group $\mathbf{G} = \langle P \rangle$ of pints over the elliptic curve $E(\mathbf{F}_q)$ and its generator point of order n, where $\mathbf{F}_q$ is a finite field, q is a large odd prime of at least 160 bits, and $n > 4\sqrt{q}$.
ID_i	The identity of sensor node i.
BS	The back-end system.
SID_j	The identity of sink node j.
GID_j	The identity of sense grid j.
C_i	The credential of sensor node i which is issued by BS.
SK_{ij}	The common path-key established by node i and j.
GSK	The grid-secret key shared among all sink nodes in the network.
K_i	The secret key of sink node i.
L_i	A lifetime which the sensor node i allowed to join the sensor network.
T_i	The bootstrapping time of sensor node i.
n_i	A nonce which is generated by node i.
$Ticket_{ij}$	A ticket which is generated by sink node for one sensor node i to communicate with another sensor node j.
$H(.)$	A public and collision-free one-way hash function.
$MAC_{i,j}$	The message authentication code which is generated by node i and it would be verified by node j and is defined by $MAC = H(k; m)$, where m denotes the message under the protection key of k.
$a\|\|b$	Concatenation of message a and b.
$E_K[.]$	The symmetric encryption function with key K.
$D_K[.]$	The symmetric decryption function with key K.

Table 2. Assumptions

A-1	All sensor nodes have the same transmission range and the links between connected nodes are bidirectional wireless links. Each sensor or sink node has a unique, non-zero, and integer-valued ID. The grid-secret key is only shared and keep secret between sink nodes that participate in the networks. Similarly, the secret key of sink node is undisclosed to any others.
A-2	Each node is capable of executing $E_{SK}[.]$, $D_{SK}[.]$, and $H(.)$ algorithms.
A-3	The proposed protocol provides perfect forward secrecy meaning if the sensor's credential, grid-secret key and two communication nodes' session key are simultaneously revealed to an attacker, the attacker is still unable to derive the session keys of previous sessions because to do this is as difficult as solving the Diffie-Hellman problem over ECDLP.
A-4	We assume that ES2S-AKE scheme does not provide a mechanism to against a Denial-of-Service (DOS) attack. For this attack, an attacker can simply disrupt, subvert, or destroy a network and this kind of attack is common though to all schemes, which is not our research focus in this paper.
A-5	A sink node has more computational and communication ability than normal sensor nodes and it is a trusted and non-compromised node at which all legal sensors have to identify by it in advance. Moreover, the back-end system must update some system parameters and privately send them for involved sink nodes.

FIGURE 2. General paradigms of intra-grid (a) and inter-grid (b) communications

$\mathbf{G}$ over $E(\mathbf{F}_q)$, the generator P, the sensor identity ID_i, the identity of designated sink node SID_j, the identity of sensing grid GID_j, the bootstrapping time T_i, the lifetime L_i, and the credential $C_i = H(K_j||ID_i||SID_j||GID_j||T_i||L_i)$ for further authentication during the sensor node identification phase, where K_j is its designated sink node's secret key.

For each sink node S_j, the BS preloads it with a set of sink node parameters, including $E(\mathbf{F}_q)$, the cyclic group $\mathbf{G}$ over $E(\mathbf{F}_q)$, the generator P, the sink node identity SID_j, the identity of responsible grid GID_j, the shared grid-secret key GSK, and the sink node S_j's secret key K_j. Note that S_j only protects its own secret key K_j and it does not need to store a verification table in the database; therefore taking advantage to prevent stolen-verifier attacks.

3.3. **Sensor Node Identification Phase.** In Figure 3, the flowchart of the sensor node identification phase is provided. When a sensor node bootstraps itself with its designated sink node to join a wireless sensor network, it can be accepted only if it has a legal credential C_i issued by back-end system, and its bootstrapping time T_i must be within a tolerance period of current time. The sensor node identification phase will perform the following steps:

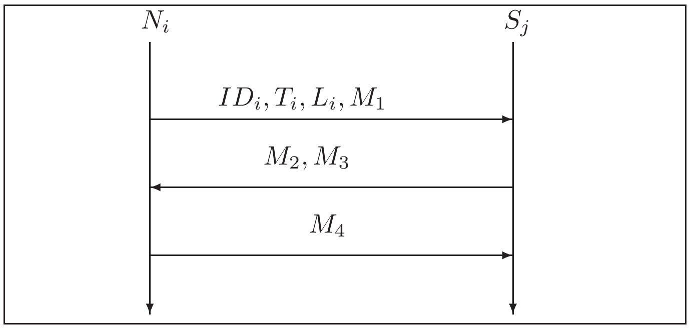

FIGURE 3. The flowchart of sensor node identification phase.

1. At the beginning, a sensor node N_i generates a random number $r_i \in \mathbf{Z}_q^*$, computes $M_1 = C_i \cdot r_i P$, and sends (ID_i, T_i, L_i, M_1) to its designated sink node S_j.
2. When the sink node S_j receives the packet, S_j first checks T_i and L_i. If above are invalid, stop; otherwise, S_j computes $C_i' = H(K_j||ID_i||SID_j||GID_j||T_i||L_i)$ and $M_1' = C_i'^{-1} M_1 = (C_i'^{-1} \cdot C_i \cdot r_i)P = r_i P$. Moreover, S_j generates a random number $r_j \in \mathbf{Z}_q^*$ and computes $M_2 = r_j P$, $SK_{ij} = r_j M_1' = r_i r_j P$, and $M_3 = H(SID_j||ID_i||M_1'||M_2||SK_{ij})$. Finally, S_j sends (M_2, M_3) to N_i.
3. When the sensor node N_i receives the packet, N_i computes $r_i P$ and $SK_{ij}' = r_i M_2 = r_i r_j P$ and checks whether $H(SID_j||ID_i||r_i P||M_2||SK_{ij}') = M_3$ holds or not. If it holds, it means that S_j is authenticated by N_i and N_i will compute $M_4 = H(ID_i||SID_j||SK_{ij}')$ and send it to S_j; otherwise, N_i discards the packet and stops.
4. When the sink node S_j receives the packet, S_j computes $M_4' = H(ID_i||SID_j||SK_{ij})$ and checks whether $M_4' = M_4$ holds or not. If it does not hold, stop; otherwise, N_i is authenticated by S_j and mutual authentication is accomplished between N_i and S_j. Finally, N_i is allowed to join

the sensor network and the common path-key SK_{ij} is used as a session key for securing further communications.

3.4. **Credential Update Phase.** If the lifetime of a sensor's credential is expired or its designated sink node wants to change their secret key from K_j into K'_j, the following procedure is performed.

1. The sink node S_j calculates new credential C'_i as $C'_i = H(K'_j||ID_i||SID_j||GID_j||T'_i||L'_i)$ and sends $E_{SK_{ij}}[C'_i, T'_i, L'_i]$ to the sensor node N_i, where K'_j is S_j's new secret key, T'_i and L'_i are new bootstrapping time and new lifetime, respectively.
2. When the sensor node N_i receives the packet, N_i reveals (C'_i, T'_i, L'_i) by computing $D_{SK_{ij}}[E_{SK_{ij}}[C'_i, T'_i, L'_i]]$ and updates the old credential (C_i, T_i, L_i) on the memory to set the new credential (C'_i, T'_i, L'_i).
3. At the time of bootstrapping, N_i bootstraps itself to S_j and the authentication procedures are the same as in the previous sensor node identification phase mentioned in Section 3.3.

3.5. **Intra-Grid Path-Key Establishment Phase.** After a sink node S_j verifies the validity of all sensor nodes in its management grid GID_j, two legal sensor nodes, N_a and N_b, in GID_j can calculate a common path-key to ensure secure communications with S_j's help. This is due to the fact that there is no additional deployment information or preloaded keys that need to be stored in sensor nodes for reducing storage overhead. As a result, S_j will issue a ticket for N_a and N_b to establish a path-key; the handshake for intra-grid path-key establishment between a sink node S_j and two sensor nodes N_a and N_b is depicted in Figure 4.

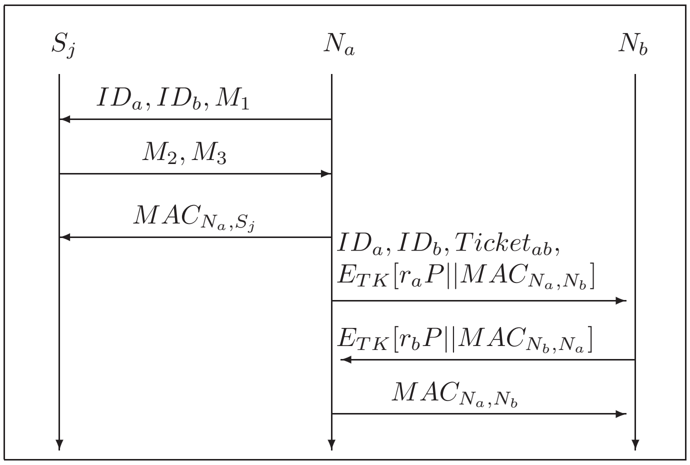

FIGURE 4. Handshake between a sink node S_j and two sensor nodes N_a and N_b for intra-grid path-key establishment phase.

The ES2S-APKE scheme for intra-grid path-key establishment phase works as follows:

1. N_a chooses a random number r_x from $\mathbf{Z}_q^*$, computes $M_1 = E_{SK_{aj}}[r_xP, n_a]$ and sends (ID_a, ID_b, M_1) to S_j, where SK_{aj} is a pre-computed session key between S_j and N_a, and n_a is a nonce.
2. S_j obtains r_xP and n_a by decrypting M_1, chooses $r_y \in \mathbf{Z}_q^*$ randomly, and computes $M_2 = E_{SK_{aj}}[r_yP, n_a + 1]$ and $M_3 = E_K[SID_j||ID_a||ID_b||r_xP||$ $Ticket_{ab}||TK||T_L]$, where $K = r_xr_yP$, TK is a temporary secret key, T_L is a lifetime for $Ticket_{ab}$, and $Ticket_{ab} = E_{SK_{bj}}[SID_j||ID_a||ID_b||TK||T_L]$. Then, S_j sends (M_2, M_3) to N_a.
3. Upon receiving (M_2, M_3) from S_j, N_a obtains r_yP and verifies the validity of $n_a + 1$ by decrypting M_2. If it holds, N_a computes $K = r_x \cdot r_yP$ and reveals $(SID_j||ID_a||ID_b||r_xP||Ticket_{ab}||TK||T_L)$ by decrypting M_3. N_a also checks whether r_xP is correct or not. If it holds, N_a believes that it really is communicating with S_j. Then, N_a sends an additional message authentication code $MAC_{N_a,S_j} = H(K; r_yP)$ to S_j. S_j checks whether K and r_yP are correct or not. If it succeeds, S_j also believes that it really is communicating with N_a.

4. N_a generates a random value $r_a \in \mathbf{Z}_q^*$ and makes $E_{TK}[r_aP||MAC_{N_a,N_b} = H(TK; r_aP)]$. Then, N_a sends ID_a, ID_b, $Ticket_{ab}$, and $E_{TK}[r_aP||MAC_{N_a,N_b}]$ to N_b.
5. N_b obtains TK by decrypting $Ticket_{ab}$ using a pre-computed session key SK_{bj}. Then, N_b further reveals (r_aP, MAC_{N_a,N_b}) by decrypting $E_{TK}[r_aP||MAC_{N_a,N_b}]$ using TK. Then, N_b examines the validity of r_aP by checking MAC_{N_a,N_b}. If the validation processes succeed, N_b chooses $r_b \in \mathbf{Z}_q^*$ randomly, computes $E_{TK}[r_bP||MAC_{N_b,N_a} = H(SK_{ab}; r_bP)]$ and sends it to N_a, where MAC_{N_b,N_a} is used for key confirmation and $SK_{ab} = r_b \cdot r_aP$ is a common path-key established by N_a and N_b and it is used for securing later communications.
6. Upon receiving the message from N_b, for key confirmation, N_a reveals r_bP, computes $SK_{ab} = r_a \cdot r_bP$, and checks the validity of $H(SK_{ab}; r_bP)$. If the key confirmation succeed, N_a convinces that SK_{ab} is established by N_a and N_b and sends $MAC_{N_a,N_b} = H(SK_{ab}; r_aP)$ to N_b for mutual authentication of SK_{ab}. Finally, N_b checks the validity of MAC_{N_a,N_b}. If it holds, N_b also believes that SK_{ab} is established by them.

3.6. **Inter-Grid Path-Key Establishment Phase.** In this phase, the communications of two sensor nodes are in different sense grids. Figure 5 shows the handshake between two sink nodes, S_i and S_j, and two sensor nodes, N_a and N_b, during the inter-grid path-key establishment phase, where N_a belongs to GID_i (sink node S_i) and N_b belongs to GID_j (sink node S_j). In Figure 5, we assume that node N_a wants to establish a path-key with node N_b, and that the transmission path of the N_b is known. The ES2S-APKE scheme for inter-grid path-key establishment phase works as follows:

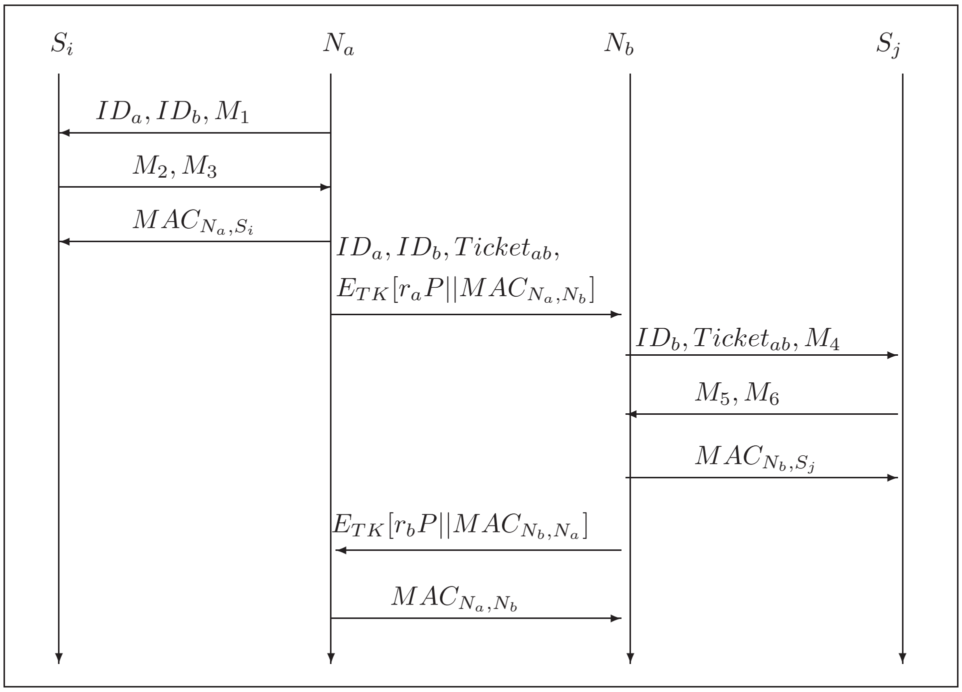

FIGURE 5. Handshake between two sink nodes S_i and S_j and two sensor nodes N_a and N_b for inter-grid path-key establishment phase.

1. N_a generates a random number r_x from $\mathbf{Z}_q^*$, makes $M_1 = E_{SK_{ai}}[r_xP, n_a]$ and sends (ID_a, ID_b, M_1) to S_i.
2. S_i obtains r_xP and n_a by decrypting M_1, chooses $r_y \in \mathbf{Z}_q^*$ randomly, and calculates $M_2 = E_{SK_{ai}}[r_yP, n_a + 1]$ and $M_3 = E_K[SID_i||SID_j||ID_a||ID_b||r_xP||$ $Ticket_{ab}||TK||T_L]$, where $K = r_xr_yP$, $Ticket_{ab} = E_{GSK}[SID_i||SID_j||ID_a||ID_b||TK||T_L]$, and

GSK is a grid-secret key shared among all sink nodes in the network. Then, S_i sends (M_2, M_3) to N_a.

3. On receiving the message (M_2, M_3) from S_i, N_a first reveals (r_yP, n_a+1) by decrypting M_2 and checks the validity of n_a+1 for freshness checking. If it holds, N_a can derive corresponding session key $K = r_x \cdot r_yP$ and decrypt M_3 to get $(SID_i||SID_j||ID_a||ID_b||r_xP||Ticket_{ab}||TK||T_L)$. Then, N_a checks the authenticity of r_xP. If it is true, N_a believes that it really communicates with S_i and responses $MAC_{N_a,S_i} = H(K; r_yP)$ to S_i. On receiving the message from N_a, S_i verifies whether K and r_yP are correct or not. If it succeeds, S_i also believes that it really is communicating with N_a.
4. N_a generates a random value $r_a \in \mathbf{Z}_q^*$ and makes $E_{TK}[r_aP||MAC_{N_a,N_b} = H(TK; r_aP)]$. Then, N_a sends ID_a, ID_b, $Ticket_{ab}$, and $E_{TK}[r_aP||MAC_{N_a,N_b}]$ to N_b.
5. On receiving the message from N_a, N_b generates a random value $r_i \in \mathbf{Z}_q^*$, makes $M_4 = E_{SK_{bj}}[r_iP, n_b]$, and forwards $(ID_b, Ticket_{ab}, M_4)$ to S_j.
6. On receiving the message from N_b, S_j decrypts M_4 to get (r_iP, n_b) using the pre-computed path-key SK_{bj}, generates a random value $r_j \in \mathbf{Z}_q^*$, computes $K' = r_j \cdot r_iP$, and makes $M_5 = E_{SK_{bj}}[r_jP, n_b + 1]$. In addition, S_j decrypts $Ticket_{ab}$ to get $(SID_i||SID_j||ID_a||ID_b||TK||T_L)$ using the common grid-secret key GSK and makes $M_6 = E_{K'}[SID_i||SID_j||ID_a||$ $ID_b||r_iP||TK||T_L]$. Finally, S_j sends (M_5, M_6) to N_b.
7. On receiving the message from S_j, N_b decrypts M_5 to get (r_jP, n_b+1) and checks the authenticity of n_b+1. If it is true, N_b is convinced that it really is communicating with S_j and computes $K' = r_i \cdot r_jP$. For mutual authentication, N_b sends $MAC_{N_b,S_j} = H(K'; r_jP)$ to S_j and S_j will check the authenticity of MAC_{N_b,S_j}. If it is true, S_j also believes that it really is communicating with N_b.
8. When K' is derived, N_b can use it to decrypt M_6 to get $(SID_i||SID_j||ID_a||$ $ID_b||r_iP||TK||T_L)$. Moreover, N_b uses TK to decrypt $E_{TK}[r_aP||MAC_{N_a,N_b}]$ to get $(r_aP||MAC_{N_a,N_b})$ and checks the authenticity of MAC_{N_a,N_b} by determining whether $H(TK; r_aP) = MAC_{N_a,N_b}$ is true or not. If it is true, N_b generates a random value $r_b \in \mathbf{Z}_q^*$ and derives the common path-key $SK_{ab} = r_b \cdot r_aP$. For key establishment, N_b makes $MAC_{N_b,N_a} = H(SK_{ab}; r_bP)$ and sends $E_{TK}[r_bP||MAC_{N_b,N_a}]$ to N_a.
9. On receiving the message from N_b, N_a decrypts it to get $(r_bP||MAC_{N_b,N_a})$ and computes $SK_{ab} = r_a \cdot r_bP$ to check if $H(SK_{ab}; r_bP) = MAC_{N_b,N_a}$ is true or not. If it is true, N_a is convinced that the path-key SK_{ab} is really established with N_b and sends $MAC_{N_a,N_b} = H(SK_{ab}; r_aP)$ to N_b for mutual authentication. Finally, N_b checks the authenticity of MAC_{N_a,N_b}. If it is true, N_b also is convinced that SK_{ab} is really established with N_a and it is used for securing later communications between N_a and N_b.

4. **Analysis of ES2S-APKE Scheme.** In the section, we analyze the essential requirements and security of the ES2S-APKE scheme and show a performance evaluation of the proposed ES2S-APKE scheme in terms of the computational and communicative costs.

4.1. **Properties Analysis.** In this subsection, we describe how our proposed ES2S-APKE scheme achieves the security-related properties listed in Section 1.3.

Mutual authentication: The ES2S-APKE scheme copes with this requirement by introducing a challenge-response mechanism, as described in Section 3. During the sensor node identification phase, the sensor node is authenticated based on its authorized credential in the sense that the sink node verifies that the sensor node is indeed legal and authorized. In addition, the sink node authenticates itself to the sensor node by showing the knowledge of its own secret key. Moreover, for two general paradigms of intra-grid and inter-grid sensor-to-sensor path-key establishment, the ES2S-APKE scheme provides mutual authentication not only between two communication sensors but also between the sensor node and the corresponding sink node.

Sleep pattern: The sleep pattern is highly necessary in WSNs to extend the availability of the networks. Therefore, in the ES2S-APKE scheme, the sink node can set fresh bootstrapping times for live sensors while other sensor nodes can shut down to save power. Different sensor nodes are operated according to the bootstrapping times to which they belong and the lifetime of WSNs is therefore extended in a differentiated way.

Location awareness: If an attacker is able to compromise a sensor node's secret information somehow and tries to impersonate the identity of this sensor to replicate to other sensor grids, he/she will fail. This is due to the fact that he/she cannot start using the compromised sensor without a legal credential. No one can tamper with the location information of compromised credentials, and thus our proposed location-aware mechanism can be used to control the damage for spreading out over the whole network from the compromised node if that node is unable to sustain compromise for a preset time interval.

No verification table: Since there are not any passwords or verification tables stored in the sink nodes, stolen-verifier attacks can not work in ES2S-APKE scheme. Moreover, our scheme can reduce the burden of the sink node's management overhead and the sink node S_j's secret key K_j is protected by the secure one-way hash function $H(.)$.

Security of path-key establishment: The common path-key $SK_{ab} = r_a \cdot r_b P$ is known to no one but the N_a and N_b since the random values r_a and r_b are only known to N_a and N_b, respectively. Moreover, the common path-key is established according to the Diffie-Hellman algorithm [7] over ECDLP (Elliptic Curve Discrete Logarithm Problem), and two sensors can negotiate a common path-key to provide secure communications in our ES2S-APKE scheme after they authenticate each other.

Perfect forward secrecy: For the inter-grid path-key establishment phase discussed in Section 3.6, we assume that the temporary key TK is known by an attacker. Then the attacker can decrypt $E_{TK}[r_aP||MAC_{N_a,N_b}]$ and $E_{TK}[r_bP||MAC_{N_b,N_a}]$ in steps 4 and 8 to obtain r_aP and r_bP, respectively. However, he cannot derive $SK_{ab} = r_a \cdot r_bP = r_b \cdot r_aP$ because the difficulty is similar to solving the Diffie-Hellman problem over ECDLP. Thus, the common path-key remains secure and our ES2S-APKE scheme provides perfect forward secrecy.

Key revocation: Unlike the addition of a sensor node to WSNs, the revocation of a sensor node is much more complicated. When a sensor node is lost or it exhausts its power, it must not be allowed to make use of the key information stored in local memory to connect to the network. In our proposed ES2S-APKE scheme, the sink node has all the information of each sensor node of its managed grid, making revocation easy. However, if the sensor node moves to another grid, the corresponding sink node has no information about which nodes have been revoked. A simple solution to this problem is to share a revocation list with each sink node. Despite the fact that the lifetime of a credential has not expired, the sensor node will not be allowed to join the network and the back-end system will reclaim this sensor to reset its credential so that it can prevent it being used for compromised node attacks.

Re-keying: For the credential update phase discussed in Section 3.4, the old credential could be updated freely between the sensor node and the corresponding sink node and the back-end system does not interfere with them to reduce the burden of the back-end system. In addition, the back-end system is still able to change the secret key stored in a sink node and the re-keying procedure is performed by updating an old secret key. Thus, the sink node can use the new secret key to generate new credentials for sensors in its managed grid without reclaiming them.

4.2. **Security Analysis.** In this part, we show how our scheme resists node replication, impersonation, wormhole, and compromised node attacks as follows:

Node replication attack: By sensor identification, our ES2S-APKE scheme can prevent attackers from directly deploying unauthorized nodes into the networks due to the impossibility of obtaining the secret key of the sink node. Therefore, attackers are unable to forge certificates for malicious sensors and the scheme ensures security by preventing malicious sensors from injecting false reports to disrupt the network function.

Impersonation attack: In the sensor identification phase, if an attacker wants to impersonate a sink node, he/she can compute a response message $M_1' = C_i'^{-1} M_1 = r_i P$ to match the sensor contribution r_iP. However, without knowing the secret key of the sink node K_j, it is impossible for the attacker to compute a legal sink node response message due to the difficulty of inverting the one-way hashing function $H(.)$ and solving the Diffie-Hellman algorithm over ECDLP. On the other hand, the naive way to impersonate a sensor is to intercept a valid login request, and then replay it again. However, since the bootstrapping time is adopted to ensure the freshness of the login request and such replaying attacks can be easily overcome. As a result, an attacker could never impersonate someone to establish a session key with the sink node and he/she cannot request any valid ticket to perform the sensor-to-sensor path-key establishment phase from the sink node.

Wormhole attack: In this attack, assume that two malicious nodes are working collusively for the purpose of tunnelling an implicit channel to attack routing protocols. Let us consider the proposed scheme assuming, an attacker wants to deploy two malicious sensors into the vicinities of honest sensors. Corresponding secret channels should be first established between a malicious sensor and its honest neighbors. However, without sending a valid ticket, two malicious sensors could not perform sensor-to-sensor path-key establishment with the other honest nodes, and thus we conclude that the ES2S-APKE scheme is secure against wormhole attacks.

Compromised node attack: In a wireless sensor network it is unavoidable that attackers will compromise deployment sensors to participate in the protocol. Subsequently, attackers can obtain the secret information stored in a compromised node to affect other non-compromised nodes. Although we do not prevent this, our credential-based scheme can prevent attackers from damaging the entire network because they cannot alter the identity of sensor grid GID_j or derive the secret key K_j of sink node S_j stored in the compromised node's credential. As a result, this kind of attack is limited to the vicinity of the compromised node in our scheme.

In the event that the attackers compromise a sensor node, it stays at the designated sensor area without attacking the networks. Since new sensors are deployed into the vicinity of this compromised node, the compromised node can establish communication channels with them. Currently no solutions can defeat these attacks [35]. As a consequence, one possible solution to such attacks is to preset new paths reserved for new sensors only, and to store the preset knowledge in the corresponding sink node. The purpose of presetting new paths is to disallow old compromised nodes from establishing communications with new non-compromised nodes, as it would be impossible for the managed sink node to issue valid tickets to old nodes. This ensures new nodes are safe from compromised attacks caused by old nodes.

4.3. **Performance Analysis.** In this subsection, we evaluate the performance of the proposed ES2S-APKE scheme in terms of the total number of cryptographic operations performed during the identification and intra/inter path-key establishment phases. To evaluate performance, we define several computational parameters as follows:

- T_{Exp}: The time of modular exponentiation.
- T_{Ha}: The time of hashing operation.
- T_{Mul}: The time of multiplication operation.
- T_{Emul}: The time of multiplication of an integer and an elliptical curve point.
- T_{Sym}: The time of symmetric encryption/decryption operation.
- T_{Asym}: The time of asymmetric encryption/decryption operation.

Computational overhead: For the communication round, multiple messages can be sent in a single round between communicating parties. Owing to the requirement of scalability, too many interactions are not scalable in sensor networks. Therefore, sensor-to-sensor or sensor-to-controller must take the communication rounds into account during the entirety of the interactions. In Tables 3, 4, and 5, we compare the performances of the involved phases of our scheme with the schemes proposed in [19, 32] in terms of computational costs. Under the conditions introduced in [18], one T_{Exp} operation is approximately equal to 240 T_{Mul} operations, and one T_{Emul} operation is approximately equal to 29 T_{Mul} operations. Moreover, as introduced in [5, 27], one T_{Asym} operation is approximately equal to five-thirds T_{Exp} operation and one T_{Ha} operation is at least 10 times faster

Table 3. Comparisons of identification phase with related schemes

	[19]	[32]	Ours
Sensor N_i	$1T_{Exp}+2T_{Sym}+2T_{Ha}+2T_{Mul}$	$3T_{Asym}+1T_{Ha}$	$1T_{Mul}+2T_{Ha}+2T_{Emul}$
Sink node S_j	$1T_{Exp}+2T_{Sym}+2T_{Ha}+2T_{Mul}$	$3T_{Asym}+1T_{Ha}$	$2T_{Ha}+3T_{Emul}$
Rough estimation	$484T_{Mul}+44T_{Ha}$	$2400T_{Mul}+2T_{Ha}$	$146T_{Mul}+4T_{Ha}$
Mutual authentication	No	No	Yes
Forward secrecy	No	No	Yes
Location aware	No	No	Yes
Re-keying	No	No	Yes

Table 4. Comparisons of intra-grid communications with related schemes

	[19]	[32]	Ours
Sensor N_a	$2T_{Sym}+1T_{Ha}$	$8T_{Sym}+4T_{Ha}$	$5T_{Sym}+4T_{Ha}+4T_{Emul}$
Sensor N_b	$2T_{Sym}+1T_{Ha}$	$8T_{Sym}+4T_{Ha}$	$3T_{Sym}+3T_{Ha}+2T_{Emul}$
Sink node S_j	$4T_{Sym}$	$8T_{Sym}+4T_{Ha}$	$3T_{Sym}+1T_{Ha}+2T_{Emul}$
Rough estimation	$82T_{Ha}$	$252T_{Ha}$	$232T_{Mul}+118T_{Ha}$
Mutual authentication	No	Yes	Yes
Forward secrecy	No	No	Yes

Table 5. Comparisons of inter-grid communications with related schemes

	[19]	[32]	Ours
Sensor N_a	$2T_{Sym}+1T_{Ha}$	$4T_{Sym}+3T_{Ha}+4T_{Asym}$	$5T_{Sym}+4T_{Ha}+4T_{Emul}$
Sensor N_b	$2T_{Sym}+1T_{Ha}$	$2T_{Ha}+4T_{Asym}$	$5T_{Sym}+4T_{Ha}+4T_{Emul}$
Sink node S_i	$4T_{Sym}$	$7T_{Sym}+4T_{Ha}+1T_{Asym}$	$4T_{Sym}+1T_{Ha}+2T_{Emul}$
Sink node S_j	$4T_{Sym}$	$3T_{Sym}+2T_{Ha}+1T_{Asym}$	$4T_{Sym}+1T_{Ha}+2T_{Emul}$
Rough estimation	$122T_{Ha}$	$4000T_{Mul}+152T_{Ha}$	$348T_{Mul}+190T_{Ha}$
Mutual authentication	No	No	Yes
Forward secrecy	No	No	Yes

than one T_{Sym} operation. Therefore, we learn that

$$1\ T_{Asym} \approx 5/3\ T_{Exp}, 1\ T_{Exp} \approx 240\ T_{Mul},\ and\ 1\ T_{Asym} \approx 400\ T_{Mul}. \tag{1}$$

$$1\ T_{Emul} \approx 29\ T_{Mul}\ and\ 1\ T_{Sym} \approx 10\ T_{Ha}. \tag{2}$$

with respect to software. Table 3 shows that the identification phase of our proposed scheme is extremely lightweight despite many additional security properties. The computational loads of the involved parties of our scheme are approximately 25% and 6% of [19] and [32], respectively.

In addition, Tables 4 and 5 show that the computational loads of two intra/inter communications of our proposed scheme are heavier than Lee-Chang's scheme [19]. However, the proposed ES2S-APKE scheme is well suited to sensor networks as it achieves the basic security goals of mutual authentication and perfect forward secrecy, and Lee-Chang's scheme does not achieve these security requirements. Also, it has been shown that ECC or symmetric computations need less computation time than modular exponentiation computations, and ECC with a 160-bit key size can be instead 1024-bit key size in RSA and ElGamal solutions [13]. On the other hand, although Varadharajan et al.'s scheme [32] achieves non-repudiation, it will have great trouble in performing asymmetric operations for every sensor in each ongoing session due to its constrained computational capability. Hence, our scheme employing ECC can achieve much higher saving in energy and bandwidth. As a result, we believe that the performance of our proposed ES2S-APKE scheme is acceptable for sensor nodes and can be practically applied over WSNs.

Communication overhead: Any two parties in the proposed scheme require three rounds to accomplish mutual authentication and session key establishment. Note that three rounds is the minimum number needed for any authenticated key establishment scheme with mutual authentication to fulfill its goal. As a result, the proposed ES2S-APKE scheme is highly efficient in the sense of communication overhead.

Storage overhead: While sensors are bootstrapping and authenticating themselves to the sink node S_j, the sink node S_j only stores three values $(ID_i,\ L_i,\ SK_{ij})$ for each currently in-use sensor. In addition, each sensor node stores one nonce and two contributions for each ongoing session. Once two sensors agree on the common path-key which is established and only shared between them, each sensor maintains no permanent partial key information and only stores one path-key per session.

Management overhead: The proposed ES2S-APKE scheme achieves low management overheads because the sink node does not need to maintain one certificate per sensor and each credential is still secure against malicious attacks as discussed in the previous subsection. On the other side, each sensor only needs to maintain its own certificate it belongs to and it does not require to maintain a member list that contains the identities and the corresponding credentials of current members of participating grid.

5. **Conclusions.** In this paper, a hierarchical and efficient sensor-to-sensor authenticated path-key establishment (ES2S-APKE) scheme based on elliptic curve cryptography for wireless sensor networks is proposed. The primary features of the ES2S-APKE scheme are: (1) mutual authentication and perfect forward secrecy are achieved in three rounds of message exchange; (2) the scheme can defend against most of the well-recognized attacks presented in Section 1.2, even if the sensor node is compromised; (3) the sink node has no need to store verification tables; (4) the mechanisms of location awareness and key revocation are used to control the damage from malicious attacks; (5) re-keying and sleep patterns are introduced for back-end system to update sensor's lifetime to increase the availability of WSNs. Compared with other related schemes, we show that the proposed ES2S-APKE scheme can be efficiently implemented in limited-resource sensor devices due to the usage of cryptology based on Elliptic Curve Cryptography that is more efficient than schemes based on the RSA or ElGamal solutions.

Acknowledgment. This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the grants NSC 96-2219-E-001-001, and NSC 96-2219-E-009-013.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, Wireless sensor networks: a survey, *Computer Networks*, vol.38, no.4, pp.393-422, 2002.
[2] H. Chan and A. Perrig, PIKE: peer intermediaries for key establishment in sensor networks, *Proceedings of IEEE INFOCOM*, pp.13-17, 2005.
[3] H. Chan, A. Perrig and D. Song, Random key pre-distribution schemes for sensor networks, *Proceedings of IEEE Symposium on Security and Privacy*, pp.197-213, 2003.
[4] Michael Chorzempa, Jung-Min Park and Mohamed Eltoweissy, Key management for long-lived sensor networks in hostile environments, *Computer Communications*, vol.30, no.9, pp.1964-1979, 2007.
[5] R. Cramer and V. Shoup, A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack, *Advances in Cryptology, CRYPTO'98*, pp.13-25, 1998.
[6] Ashok Kumar Das, An identity-based random key pre-distribution scheme for direct key establishment to prevent attacks in wireless sensor networks, *International Journal of Network Security*, vol.6, no.2, pp.129-139, 2008.

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

[8] Falko Dressler, Authenticated reliable and semi-reliable communication in wireless sensor networks, *International Journal of Network Security*, vol.7, no.1, pp.62-69, 2008.

[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*, pp.41-47, 2002.

[10] Fei Hu and Neeraj K. Sharma, Security considerations in ad hoc sensor networks, *Ad Hoc Networks*, vol.3, no.1, pp.69-89, 2005.

[11] Fei Hu, Waqaas Siddiqui and Krishna Sankar, Scalable security in wireless sensor and actuator networks (WSANs): Integration re-keying with routing, *Computer Networks*, vol.51, no.1, pp.285-308, 2007.

[12] D. Huang, M. Mehta, D. Medhi and H. Lein, Location aware key management scheme for wireless sensor networks, *Proceedings of ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp.29-42, 2004.

[13] Min-Shiang Hwang and Ting-Yi Chang, Threshold signatures: Current status and key issues, *International Journal of Network Security*, vol.1, no.3, pp.123-137, 2005.

[14] Minoru Ito and Masahiro Tanaka, Localization of a Moving Sensor by Particle Filters, *International Journal of Innovative Computing, Information and Control*, vol.4, no.1, pp.165-173, 2008.

[15] Yixin Jiang, Chuang Lin, Minghui Shi and Xuemin (Sherman) Shen, Self-healing group key distribution with time-limited node revocation for wireless sensor networks, *Ad Hoc Networks*, vol.5, no.1, pp.14-23, 2007.

[16] Issa Khalil, Saurabh Bagchi and Ness Shroff, Analysis and evaluation of SECOS, a protocol for energy efficient and secure communication in sensor networks, *Ad Hoc Networks*, vol.5, no.3, pp.360-391, 2007.

[17] K. Koblitz, Elliptic curve cryptosystems, *Mathematics of Computation*, vol.48, no.177, pp.203-209, 1987.

[18] Neal Koblitz, Alfred Menezes and Scott Vanstone, The state of elliptic curve cryptography, *Designs, Codes and Cryptography*, vol.19, no.2-3, pp.173-193, 2000.

[19] Jung-San Lee and Chin-Chen Chang, Secure communications for cluster-based ad hoc networks using node identities, *Journal of Network and Computer Applications*, vol.30, no.4, pp.1377-1396, 2007.

[20] D. Liu and P. Ning, Establishing pair-wise key establishments in distributed sensor networks, *Proceedings of 10th ACM Conference on Computer and Communications Security*, pp.52-61, 2003.

[21] Victor S. Miller, Use of elliptic curves in cryptography, *Advances in Cryptology, CRYPTO'85*, Lecture Notes in Computer Science, vol. 218, pp.417-426, 1985.

[22] J. Newsome, E. Shi, D. Song and A. Perrig, The sybil attack in sensor networks: analysis & defenses, *The 3rd International Symposium on Information Processing in Sensor Networks*, California, USA, pp.259-268, 2004.

[23] Y. Obashi, H. Chen, H. Mineno and T. Mizuno, An energy-aware routing scheme with node relay willingness in wireless sensor networks, *International Journal of Innovative Computing, Information and Control*, vol.3, no.3, pp.565-574, 2007.

[24] L. B. Oliveira, A. Ferreira, M. A. Vilaça, H. C. Wong, M. Bern, R. Dahab and A. A. F. Loureiro, SecLEACH - on the security of cluster sensor networks, *Signal Processing*, vol.87, no.12, pp.2882-2895, 2007.

[25] Hector Benitez Perez, F. Garcia-Nocetti and H. Thompson, Fault Classification Based upon Self Organizing Feature Maps and Dynamic Principal Component Analysis for Inertial Sensor Drift, *International Journal of Innovative Computing, Information and Control*, vol.3, no.2, pp.257-276, 2007.

[26] Asad Amir Pirzada and Chris McDonald, Detecting and evading wormholes in mobile ad-hoc wireless networks, *International Journal of Network Security*, vol.3, no.2, pp.191-202, 2006.

[27] B. Schneier, *Applied Cryptography, Protocols, Algorithms, and Source Code in C*, John Wiley and Sons Inc., second edition, 1996.

[28] Jang-Ping Sheu and Jui-Che Cheng, Pair-wise path key establishment in wireless sensor networks, *Computer Communications*, vol.30, no.11-12, pp.2365-2374, 2007.

[29] Wei-Tsung Su, Ko-Ming Chang and Yau-Hwang Kuo, eHIP: An energy-efficient hybrid intrusion prohibition system for cluster-based wireless sensor networks, *Computer Networks*, vol.51, no.4, pp.1151-1168, 2007.

[30] Qian Tian and Noriyoshi Yamauchi, A New Method of Noise Removal for Body Vibration Signals in Wireless Sensor Networks, *International Journal of Innovative Computing, Information and Control*, vol.2, no.6, pp.1259-1270, 2006.

[31] Chwei-Shyong Tsai, Cheng-Chi Lee and Min-Shiang Hwang, Password authentication schemes: Current status and key issues, *International Journal of Network Security*, vol.3, no.2, pp.101-115, 2006.

[32] Vijay Varadharajan, Rajan Shankaran and Michael Hitchens, Security for cluster based ad hoc networks, *Computer Communications*, vol.27, no.5, pp.488-501, 2004.

[33] Nen-Chung Wang and Shian-Zhang Fang, A hierarchical key management scheme for secure group communications in mobile ad hoc networks, *Journal of Systems and Software*, vol.80, no.10, pp.1667-1677, 2007.

[34] Yang Xiao, Venkata Krishna Rayi, Bo Sun, Xiaojiang Du, Fei Hu and Michael Galloway, A survey of key management schemes in wireless sensor networks, *Computer Communications*, vol.30, no.11-12, pp.2314-2341, 2007.

[35] Yun Zhou, Yanchao Zhang and Yuguang Fang, Access control in wireless sensor networks, *Ad Hoc Networks*, vol.5, no.1, pp.3-13, 2007.