Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2011, Article ID 893592, 12 pages
doi:10.1155/2011/893592

Research Article

Secure Clustering and Symmetric Key Establishment in Heterogeneous Wireless Sensor Networks

Reza Azarderskhsh and Arash Reyhani-Masoleh

Department of Electrical and Computer Engineering, The University of Western Ontario, London, ON, Canada N6A 5B9

Correspondence should be addressed to Reza Azarderskhsh, razarder@uwo.ca

Received 1 June 2010; Revised 10 August 2010; Accepted 2 October 2010

Academic Editor: Damien Sauveron

Information security in infrastructureless wireless sensor networks (WSNs) is one of the most important research challenges. In these networks, sensor nodes are typically sprinkled liberally in the field in order to monitor, gather, disseminate, and provide the sensed data to the command node. Various studies have focused on key establishment schemes in homogeneous WSNs. However, recent research has shown that achieving survivability in WSNs requires a hierarchy and heterogeneous infrastructure. In this paper, to address security issues in the heterogeneous WSNs, we propose a secure clustering scheme along with a deterministic pairwise key management scheme based on public key cryptography. The proposed security mechanism guarantees that any two sensor nodes located in the same cluster and routing path can directly establish a pairwise key without disclosing any information to other nodes. Through security performance evaluation, it is shown that the proposed scheme guarantees node-to-node authentication, high resiliency against node capture, and minimum memory space requirement.

1. Introduction

The extensive rise of using wireless sensor networks (WSNs) in diverse applications such as hostile, unattended, and inaccessible environments mandates the users to be more assured about the security compared to the survivability. The inherent nature of wireless sensor nodes, such as being subject to resource constraints (power, processing, and communication), easily captured, and possibly tampered with, causes other security schemes developed for infrastructure-based wireless networks to be infeasible for WSNs [1, 2]. An example of these sensor nodes is the reduced function devices (RFDs) defined in the IEEE 802.15.4-2006 standard [3].

As long as security schemes provide confidentiality, authentication, and integrity, which are critical for such applications, a secure and survivable infrastructure is always desired. Network survivability has been defined as the ability of the network to fulfill its mission in the presence of attacks and/or failures in a timely manner [4]. As a standard criteria to enhance scalability and survivability in the WSNs, clustering sensor nodes into some groups is considered in the literature, see, for example, [5–9]. Due to the energy constraint nature of wireless sensor nodes and their limited transmission range, establishing multihop routing toward the gateway is more efficient than having direct transmission [7]. Moreover, data transmission consumes the most energy in comparison with data computation. Consequently, sending signals in an optimal power level is very crucial. From the security point of view, through compromising a sensor node by an adversary in a multi-hop path, the information on the node is exposed, and an attacker might be able to control the operation of the captured node. Therefore, for the purpose of securing communication links in WSNs, every message should be encrypted and authenticated by any two individual sensor nodes [10].

The secure clustering and key establishments are challenging problem in the WSNs. Therefore, an efficient key management scheme should be designed in order to distribute the cryptographic keys amongst the sensor nodes. It is noted that using a single traditional symmetric key is not secure; because sensor nodes are not tamper proof and upon being captured by an adversary, all information will be exposed to the adversaries [11]. Recently, incorporating pairwise keys for secure communication amongst sensor nodes in the heterogeneous WSNs has been considered in [12, 13].

In this paper, we investigate secure clustering of wireless sensor nodes with evaluating their survivability concurrently. To date, numerous key establishment schemes have been proposed for homogeneous WSNs incorporating symmetric keys, that is, what is mentioned in [1, 11, 14–17]. In these schemes, the secure connectivity is based on the probability of sharing some symmetric keys and key materials among sensor nodes. Note that these schemes not only suffer from high computation cost, communication overhead, and large memory requirements, but also there is no guarantee for secure key establishment among all sensor nodes. Moreover, due to the resource constraint nature of sensor nodes, employing asymmetric and public key cryptography in WSNs using these schemes is slow, complex, and infeasible [18].

Recently, Malan et al., [19], demonstrated that a lightweight type of public key cryptography called elliptic curve cryptography (ECC) is computationally feasible for resource-constrained sensor nodes in WSNs. In [20], a public key cryptography scheme called TinyECC is presented. This scheme is based on software implementation of ECC on TinyOS for sensor nodes. To have an acceptable security level, it has been demonstrated that ECC requires considerably less resources compared with RSA [21] depending on the key size. In [22], it has been shown that even RSA can be feasible for sensor nodes under certain conditions, such as employing a dedicated hardware accelerator for cryptographic computations. Furthermore, recent works such as those in [23, 24] have presented the use of ECC public key cryptography for WSNs.

In clustered WSNs, there is a hierarchy among the nodes regarding their capabilities. Gateways are more powerful and have greater resources while sensor nodes are limited in resources. In these networks, gateways form a virtual infrastructure and sensor nodes connect to the gateways in a direct or multi-hop routes [25]. The gateways are assumed to be tamper proof and can be used to distribute cryptographic keys to the sensor nodes. Recent research (see, e.g., [11, 12, 26–29]) has assumed that the adversary is present after node deployment and key establishment phases. Consequently, the adversary is unable to compromise the links without actually capturing a sensor node. However, in situations such as enemy battle fields, borderline monitoring, and autonomous networks with high-security requirements, it is not practical to assume that the adversary does not exist in the field during deployment and the exchanged information may be recorded/altered by the adversary. Therefore, a security mechanism should be proposed to solve this problem.

In this paper, we capitalize on the strength of public key cryptography to establish secure communication in clustered WSNs. Since gateways in clustered WSNs are assumed to be powerful and tamper proof, they can operate as a key distribution center (KDC) within each cluster. We present a deterministic pairwise key establishment scheme for the clustered WSNs using public key cryptography. In comparison with the previous works available in the literature, the proposed scheme has the following contributions.

(i) We propose a new secure clustering scheme for the heterogeneous WSNs incorporating ECC. The key management scheme is performed in the early phase of clustering and bootstrapping with the assumption that the adversary exists in the environment.

(ii) Instead of preloading large number of keys into each sensor node, we embed the public key of the gateways into each sensor node before deployments. Therefore, any broadcast from the gateways can be authenticated easily by the legitimate sensor nodes using elliptic curve digital signature algorithm (ECDSA) [30].

(iii) The memory complexity and the overall communication overhead of the presented scheme are analyzed in terms of the number of neighbor nodes available for each sensor node. Consequently, the number of symmetric keys required to be stored in each sensor node is obtained efficiently. It is shown that the memory requirements of the proposed scheme are less than its counterparts.

(iv) We investigate the node/link compromise probability regarding the number of hops. Note that when a node is captured by the adversary, the pairwise nature of the proposed scheme exposes no information from other communication links.

In the proposed scheme, all messages broadcasted from the gateways should be authenticated. Therefore, the messages from illegitimate users or compromised sensor nodes can be easily rejected by the other nodes.

The organization of this paper is as follows. In Section 2, we review the related work. The preliminaries and network model are stated in Section 3. The proposed secure clustering scheme is presented in Section 4. Section 5 shows an analysis on node degree in the proposed network model for clustered WSNs. The performance analysis and simulation results are reported in Section 6. Finally, we conclude the paper in Section 7.

2. Related Work

In this section, we review the related works that have been previously proposed for key management in WSNs. To be more specific and to improve the comparison, we focus on the hierarchical/heterogeneous networks rather than distributed and homogeneous WSNs.

The idea of using a pairwise key scheme to secure communication links in WSNs is proposed by Chan et al., [14]. In this scheme, each node stores pairwise keys between other nodes in the entire network. This scheme allows node-to-node authentication; however, upon node capture all the keys in the WSN are revealed. Furthermore, the scheme is not scalable for large networks. In [26], a low-energy key management protocol for clustered WSNs is presented, where all sensor nodes of the cluster are randomly assigned to each gateway within the clusters before deployment.

Recently, a probabilistic unbalanced and distributed scheme is presented for heterogeneous WSNs in [31]. Their scheme leverages the existence of a small percentage of powerful (more capable) sensor nodes beyond the low-power sensor nodes. The powerful nodes are equipped with additional keys and act as gateways within the network. These nodes are assumed to be tamper proof if they are captured by an adversary. It has been shown that their scheme, which is based on the work proposed entirely in [11], not only provides an equal level of security but also reduces the effects of both single and multiple node capture attacks.

A uniform framework for random key management in the distributed peer-to-peer WSNs with heterogeneous sensor nodes is proposed in [12]. Indeed, similar to [31], the deployment of some heterogeneous sensor nodes (called high-class nodes) amongst the low-class sensor nodes has been studied. In this heterogeneous WSN, the connectivity between a low-class node and a high-class node is more important than the connectivity between two low-class sensor nodes. In [31], a hybrid security mechanism is proposed that can work with or without the presence of KDC. Here, all the sensor nodes are preloaded with a random set of keys drawn from a pool before deployment. Whenever KDC is available, each gateway shares a public and private key combination with KDC. The authors evaluate connectivity, reliability, and resiliency of their scheme, but the memory requirement may not be scalable in certain situations.

In [18], the concept of incorporating deployment knowledge for key establishments in heterogeneous WSNs is presented. This scheme relies on prior deployment knowledge and location information. It should be noted that in some applications such information is not available.

An efficient public key-based heterogeneous sensor network key distribution scheme is proposed in [32]. This scheme provides facilities for in-network processing, which helps optimize usage of sensor resources incorporating a certificate generation using the private key of the base station. The authors of [2] proposed a key predistribution scheme for heterogeneous WSNs based on symmetric key techniques. Note that they do not provide a prefect tradeoff between resiliency against node capture and memory storage requirements.

In [33], an identity and pairing-based secure key management scheme for heterogeneous sensor networks is presented. In this scheme, sensor nodes do not need to store any key of the other nodes, rather it computes secret sharing key using pairing and identity properties. In [34], a multiuser broadcast authentication is presented that emphasizes the use of public key cryptography in heterogeneous WSNs. The scheme is of interest but is applicable for special kind of WSNs with many user nodes.

3. Preliminaries

In this section, we describe the notations and network model used for the clustered WSNs.

Table 1: Notations and their definitions.

Notation	Definition
N	Number of sensor nodes in the network
A	Area that sensor nodes are deployed
G	Number of gateways in the network
n	Number of neighbor nodes
r	Transmission range of each sensor node
R	Largest radius of a cluster covered by each gateway
n_i	Sensor node n_i, $i \in \{1,\ldots,N\}$
S	Area covered by each sensor node
G_j	Gateway G_j, $j \in \{1,\ldots,G\}$
$K_{n_i}^{n_{i'}}$	Symmetric key between sensor node n_i and $n_{i'}$
$P^u_{n_i}$, $P^r_{n_i}$	Public and private key of sensor node n_i, $1 \le i \le N$
x_i	Probability of node n_i to be compromised
$P^u_{G_j}$, $P^r_{G_j}$	Public and private key of gateway G_j, $1 \le j \le G$
$E_K(\cdot)$	The encryption function using the key K
$D_K(\cdot)$	The decryption function using the key K
$\deg n_i$	Number of links connected to the node n_i

3.1. Notations and Definitions. Let n_i and G_j denote the senor node i, $i \in \{1,\ldots,N\}$ and the gateway j, $j \in \{1,\ldots,G\}$, in the network, respectively. We assume that each sensor node and gateway are identified by a unique ID number i and j, respectively, where N and G are the largest ID numbers. We use $\deg n_i$ to represent the number of edges connected securely to a sensor node n_i. The transmission ranges of all sensor nodes and all the gateways are noted by r and R, respectively, where $R > r$. Therefore, a sensor node and a gateway can communicate with each other if they are within the distance r of each other.

Definition 1. A set of sensor nodes $\mathcal{N}$ is a covering set of area A if and only if for each point, say $P \in A$, there is $n_i \in \mathcal{N}$ that n_i covers P. The senor node n_i covers point P if it falls into the transmission range of the node n_i, that is, r [8].

The largest radius of a cluster was covered by a gateway G_j, defined by R, and approximated by multiplying the range of each sensor node, r, with the number of hops to the gateway, h, that is, $R_{G_j} = h \times \mathrm{r}$.

Definition 2. Minimum spanning tree [35]: given a connected weighted graph $\mathcal{G} = (V, E)$,a minimum spanning tree covers all the vertices$V$(contains $|V| - 1$ edges) of $\mathcal{G}$that has minimal total edge weight.

Definition 3. Shortest path tree [35]: a shortest path tree of a connected weighted graph $\mathcal{G} = (V, E)$ is a spanning tree of

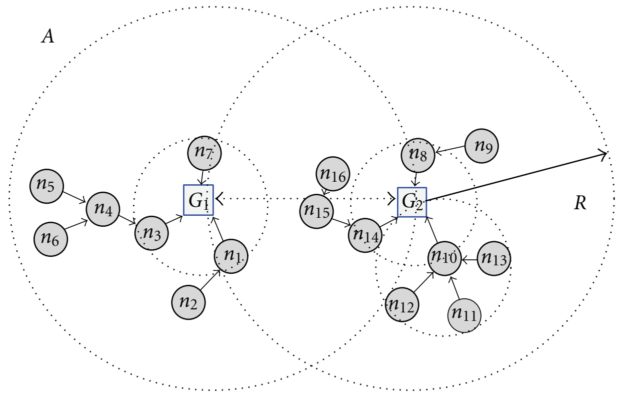

Figure 1: A simple clustered WSN with two gateways and 16 sensor nodes deployed in the area A.

$\mathcal{G}$, consisting of a root node s, that the distance between s and all other vertices in $\mathcal{G}$ is minimal.

The goal of a minimum spanning tree is minimum weight, while the goal of a shortest path tree is to preserve distances from the root [35].

Definition 4. Digital signature [30]: a digital signature algorithm is a mathematical scheme and a cryptographic tool for demonstrating nonrepudiation, authenticating the integrity and origin of a signed message. A private key is used by the signer to generate the digital signature for the message, and the public key is used by anyone to verify the signature. Note that ECDSA and RSA are popular digital signature algorithms.

All other notations used in this paper with their definition are summarized in Table 1.

3.2. Network Model. In this section, an explanation regarding secure operation of the clustered WSNs is presented. Then, an elaboration on how to establish security in the initial phase of bootstrapping and clustering of these networks is given. In this model, it is assumed that the number of gateways is relatively small in comparison with the number of sensor nodes, that is, $G \ll N$, and the gateways are aware of their location information and can communicate with each other and the base station (BS) securely. An illustration of a typical clustered WSNs is shown in Figure 1. To meet the coverage requirements, we assume that all sensor nodes are distributed uniformly and randomly in the monitoring area A. Note that sensor nodes have no knowledge about their geographic location information.

In this model, two phases of operations, namely preloading and deployment, are proposed. In what follows, these phases are explained.

3.2.1. Prior Deployment and Preloading Phase. Before sensor nodes are randomly deployed in an environment, a server is used to generate and preload required keys based on ECC into sensor nodes and gateways. As illustrated in Figure 2(a), a sensor node, say n_i, $1 \leq i \leq N$, is preloaded with its own public key, that is, $P^u_{n_i}$, private key, that is, $P^r_{n_i}$, and the public key of all existing gateways in the network, that is, $\{P^u_{G_j} \mid 1 \leq j \leq G\}$. Consequently, the gateway G_j is preloaded with the public key of all gateways (including its own) $\{P^u_{G_j} \mid 1 \leq j \leq G\}$, its private key $P^r_{G_j}$, and the public keys of all sensor nodes $\{P^u_{n_i} \mid 1 \leq i \leq N\}$ in the network. These keys are embedded in the sensor nodes and the gateways.

3.2.2. Deployment Phase. In clustered WSNs, sensor nodes are deployed randomly and uniformly in a manner similar to distributed WSNs as explained entirely in [11, 36]. The gateways are deployed within the field, such that each sensor node can hear from at least one gateway. This is achieved by varying the transmission range of gateways, R, in the network during the initial communication setup. We assume that the gateways know the location of the BS and communicate with the BS directly or in a multi-hop manner securely.

4. Proposed Secure Clustering

Sensor nodes in clustered WSNs should be securely partitioned into clusters. Therefore, we assume that if the adversaries exist in the field, they are unable to comprehend the exchanged information. In Figure 1, a simple network with two gateways (G_1 and G_2) and 16 sensor nodes (n_1 to n_{16}) is illustrated. The gateway G_j in each cluster should securely discover all the sensor nodes which belong to it. Additionally, sensor nodes should be aware of their assigned gateway/cluster.

As depicted in Figure 2(b), each gateway G_j broadcasts the message B_{G_j} to all sensor nodes with a random delay, that is,

$$\begin{gathered} G_j \longrightarrow n_i : \\ B_{G_j} = \left\langle \mathrm{ECDS}_{P^r_{G_j}}\left\{h\left(M\|\mathrm{ID}_{G_j}\right)\right\}, P^u_{G_j}, M, \mathrm{ID}_{G_j}\right\rangle. \end{gathered} \tag{1}$$

Here, M denotes the broadcast message and as presented in (1) G_j calculates B_{G_j} as follows. First, a one-way hash function $h(\cdot)$ is executed over the $(M\|\mathrm{ID}_{G_j})$, where "$\|$" denotes the concatenation operator. Second, an elliptic curve digital signature [30] is calculated over the hash results using the private key of the gateway G_j, that is, $\mathrm{ECDS}_{P^r_{G_j}}$. The final message should be accompanied by the public key of the gateway G_j, that is, $P^u_{G_j}$, message M, and ID_{G_j}. This broadcast will be repeated several times to ensure that the maximum number of sensor nodes receives it.

For the purpose of message authentication, upon receiving the broadcast message, the sensor node n_i makes a list for all the received messages from the gateways as $\ell = \{B_{G_1}, B_{G_2}, \ldots, B_{G_k}\}$, where k, $1 \leq k \leq G$, is the number of gateways from which a sensor node received a broadcast message. Priority of the generated list is based on signal-to-noise ratio (SNR) of the received message, that is, $P_{B_{G_1}} > P_{B_{G_2}} > \ldots > P_{B_{G_k}}$, where the $P_{B_{G_k}}$ is the received signal power from the gateway G_k for $1 \leq k \leq G$. Afterwards, each sensor node n_i will verify the message

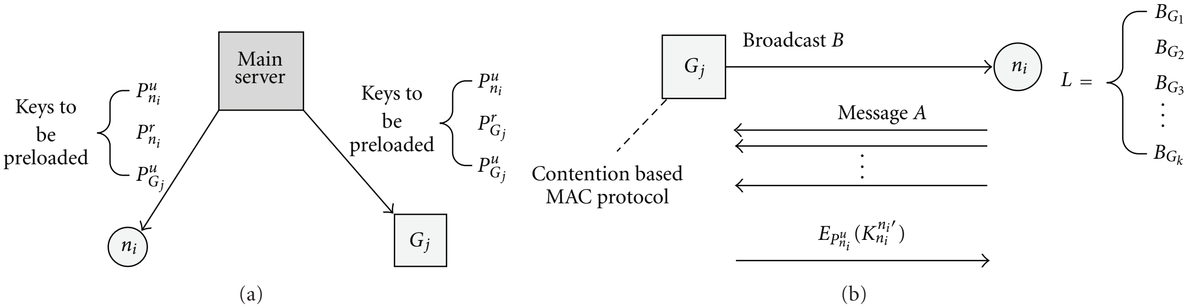

Figure 2: An illustration of information exchange prior to and after deploying sensor nodes and gateways: (a) embedding keys into gateways and sensor nodes, (b) information exchange between sensor nodes and gateways during secure clustering.

integrity using ECDSA with public key of the gateways and compares the received public key with its pre-loaded one. Note that verifying the authenticity of the public key of a gateway is finding out whether the attached public key of the gateway is the same as the one embedded in the memory of a sensor node. If the received public key does not match the pre-loaded one, sensor node n_i will reject the broadcast message. This prevents sensor nodes from performing expensive verification on the fake signatures broadcasted from the adversaries [37].

Furthermore, each sensor node n_i can determine the distance d_{n_i} from the desired gateway G_j incorporating received signal strength indicator (RSSI) [38]. The minimum distance from the gateway G_j is called one-hop distance as $d = \min\{d_{n_i}, 1 \le i \le N\}$, in which sensor nodes in this distance can communicate with the gateway directly. Using a global positioning system (GPS) for location finding [36] and time distance calculation [15] requires extra hardware costs and tight time synchronization, respectively. Furthermore, it has been shown in [38] that employing RSSI is more reliable in determining connectivity compared to the location information, as the location information is not available in various applications.

The Breadth-First search algorithm [39] is used by the gateway in each cluster to find which sensor nodes select the gateway G_j as their cluster head. Note that a similar algorithm is used in [6]. The gateway G_j broadcasts a message requesting sensor nodes to notify the gateway if they are within the communication distance d from the gateway. In this case, each sensor node n_i encrypts its ID concatenated with its public key using the public key of the desired gateway. This message is transmitted by a sensor node at maximum power to acknowledge the desired gateway in the top of its list ℓ as follows:

$$n_i \longrightarrow G_j : \quad A = E_{P^u_{G_j}}\left(\mathrm{ID}_{n_i} \| P^u_{n_i}\right), \tag{2}$$

where $E_{P^u_{G_j}}(\cdot)$ denotes the encryption function using the public key of gateway G_j. Then, the gateway G_j decrypts this message by using its private key as follows:

$$G_j : \; D_{P^r_{G_j}}(A) = \mathrm{ID}_{n_i} \| P^u_{n_i}. \tag{3}$$

In this case, the gateway G_j compares the received public key from the sensor nodes with the ones that are embedded in its memory prior to deployment. This helps to prevent an adversary from throwing illegitimate nodes into a cluster and mounting a denial-of-service (DoS) attack.

As a large number of sensor nodes will respond to a gateway, avoiding contention is difficult. Since contention causes collisions, this affects the survivability of the network. Therefore, a suitable medium access control (MAC) protocol is required to be installed in each sensor node. It is noted that assuming sensor nodes to be time synchronized is infeasible because of the large number of nodes. To overcome this problem, the contention-based and self-stabilizing MAC protocol presented in [40] is incorporated here. Eventually, each gateway will compile a list of all the sensor nodes in its cluster along with their IDs and public keys.

At this point, the public keys of sensor nodes and gateways are authenticated. Now, each gateway G_j will ask its one-hop sensor nodes n_{1i} (e.g., n_8, n_{10}, and n_{14} of cluster 2 in Figure 1) within the cluster to broadcast a message to ask its one-hop neighbors in the cluster to report to n_{1i}. In this case, sensor node n_{1i} acts as the parent node to the nodes in its one-hop neighborhood. Similarly, the other neighbors ask their one-hop neighbors to report themselves. Therefore, every node within the cluster will connect to the gateway in a single or multi-hop route, that is, $n_{1i}, n_{2i}, n_{3i}, \ldots, n_{hi}$, where h is the number of hops from a node n_i to the gateway G_j. All these sensor nodes send their information to the n_{1i} node, and n_{1i} notifies the gateways about these sensor nodes.

Every sensor node which has selected G_j as the gateway and is within the preferred cluster will be discovered by the gateway G_j. Note that a unique path exists from each node to the gateway as each node has just one parent. For routing the information to the gateway in each cluster, an appropriate routing algorithm is required. It defines the path that the packets can be forwarded to the gateway. Therefore, a minimum cost path algorithm can be used to find the optimal spanning tree rooted at the given node.

Theorem 5. *The nodes that immediately follow the root node n_i in the minimum cost tree constitute the minimum neighborhood of node n_i. The minimum cost routes between the node n_i and the gateway G_j are all contained in the minimum neighborhoods of the nodes [25].*

4.1. Secure and Survivable Routing. In this subsection, we present the routing algorithm for the sensor nodes to forward data toward the gateway in each cluster. If data from neighborhoods are highly correlated, then the minimum spanning tree (MST) is beneficial in terms of survivability and network lifetime [41]. However, in the case of low correlation amongst sensor nodes, shortest path tree (SPT) should be incorporated to achieve survivability and better network lifetime [41]. Additionally, shorter paths are more secure than the longer paths (as we explain more in Section 6.1). Note that using the shortest path limits the number of paths which can be used to relay data toward the gateway. In [42], a shortest cost path routing algorithm for maximizing network lifetime based on link costs is presented. The costs reflect both the communication energy consumption rates and the residual energy level.

Here, the use of link estimation and parent selection (LEPS) scheme was employed as proposed in [43] as a routing algorithm. In this method, each node monitors all traffic received within the one-hop range, including route updates from the neighbor nodes. Using the least cost path, it manages the nearest available neighbor node and decides the next hop. To find a least cost path, one needs to calculate the costs of all edges between each sensor node then obtain a set of least cost paths. To accomplish this, we use the cost function as formulated in [5].

(i) $f(E_{n_i})$: the function of remaining energy of the sensor node n_i, for all $i \in \{1, \dots, N\}$.

(ii) $d_{n_i,n_{i'}}$: the distance between sensor nodes n_i and $n_{i'}$.

(iii) $F(e_{n_i,n_{i'}})$: the error function between sensor node n_i and $n_{i'}$.

Then, the cost function for a link between sensor node n_i and $n_{i'}$ can be estimated as

$$C_{n_i,n_{i'}} = (d_{n_i,n_{i'}})^{\alpha} + f(E_{n_i}) + F(e_{n_i,n_{i'}}), \tag{4}$$

where α is free space loss exponent and typically $\alpha \geq 2$. The error function is related to the maximum data buffered in sensor node b and the distance between sensor nodes n_i and $n_{i'}$. Then one can write it as

$$F(e_{n_i,n_{i'}}) = c_0 \cdot \frac{d_{n_i,n_{i'}}}{b}, \tag{5}$$

where c_0 is a constant coefficient. To find the least cost path from a sensor node n_i to the gateway G_j, the number of hops should be considered as well [5].

4.2. Symmetric Key Establishment. After secure clustering, broadcast authentication, and determining the desired routing algorithm among sensor nodes and gateways, sensor nodes should establish secure communication between each other to reach the gateway securely in a multi-hop path. Since gateways are aware of the one-hop neighbors of the sensor nodes and have enough information to control sensor nodes, they send pairwise keys to each sensor node and its potential one-hop neighbors. To achieve this, gateway G_j will send the pairwise key to the sensor node n_i which is common between its neighbors $n_{i'}$ regarding the least-cost path routing algorithm.

First, the symmetric key generated for the sensor node n_i and $n_{i'}$, that is, $K_{n_i}^{n_{i'}}$, should be encrypted using the public key of the sensor node n_i, that is, $E_{P_{n_i}^u}(K_{n_i}^{n_{i'}})$, for $1 \leq i, i' \leq N$. Then, each gateway G_j unicasts this message to the sensor node n_i. Each sensor node decrypts this message using its own private key $P_{n_i}^r$ and obtains the symmetric key $K_{n_i}^{n_{i'}}$. Since this message should be encrypted by the public key (based on ECC) of every individual sensor node, then disclosing symmetric key is not possible to the adversary. As an example, in Figure 1, the sensor node n_4 will receive the symmetric keys for nodes n_3, n_5, and n_6 as $K_{n_4}^{n_3}$, $K_{n_4}^{n_5}$, and $K_{n_4}^{n_6}$, respectively.

In the proposed scheme, we do not consider unicast authentication for performance reasons. However, the following explains unicast authentication mechanism for the proposed symmetric key establishment method.

Unicast Authentication. The question is how sensor node n_i ensures that the encrypted symmetric key, that is, $E_{P_{n_i}^u}(K_{n_i}^{n_{i'}})$, is originated from gateway G_j and not from the adversary?

To address this issue, ECDSA authentication can be incorporated as follows. To ensure that the message, that is, $E_{P_{n_i}^u}(K_{n_i}^{n_{i'}})$, is unicasted from the gateway G_j, the elliptic curve digital signature can be calculated by the gateway on the message. Therefore, sensor node n_i can verify the signature using the public key of gateway G_j, and this assures that the message is coming from a legitimate gateway, and not from an adversary. This scheme requires N times signature generation by the gateways, and all the sensor nodes should verify and decrypt the unicasted message. Note that this increases the computation cost as the verification of a signature is an expensive operation. However, a one-time digital signature generation can reduce some of the overheads.

Another scheme is to allow each sensor node and its corresponding gateway to obtain a shared symmetric key during the first broadcast authentication (secure clustering) incorporating elliptic curve Diffie-Hellman (ECDH) method. Then, using symmetric key, the unicast authentication can be performed by generating a message authentication code (MAC). Therefore, any unicast from the gateway can be authenticated by the sensor nodes.

Authentication methods imply overheads in computation and communication times. Therefore, a trade-off must be achieved between the required level of security in the authentication and the time costs, otherwise the arising overheads could be against the survivability of the network.

Message Freshness. Beyond guaranteeing confidentiality and authentication, it is important to ensure that data is recent, fresh, and no adversary replayed old messages. A sensor node n_i can achieve this through a nonce (which is a unpredictable random number). In the proposed scheme, before unicasting the symmetric keys by the gateways, sensor node n_i can send a key request message to the gateway G_j accompanying with a random nonce, i.e., N_{n_i} and encrypted by $P_{G_j}^u$.

Therefore, when a gateway wants to unicast the symmetric key (encrypted by $P^u_{n_i}$) to node n_i, gateway G_j includes its random nonce, that is, $N_{G_j}$and N_{n_i} to the unicast message. After this exchange, node n_i ensures that the message is recently initiated and is not a replay of old messages.

4.3. Survivable-Secure Connectivity. To better present the connectivity in each cluster of the proposed infrastructure for a WSN, we define a graph $G = (V, E)$ to model the connectivity between a set of sensor nodes. Each sensor node is represented by a vertex in V, $V = \{n_1, \dots, n_{N_c}\}$, where N_c represents the number of sensor nodes within each cluster (In Section 5.1, we study the average number of sensor nodes inside a cluster.). For any two nodes n_i and $n_{i'}$ in V, the edge $(n_i, n_{i'}) \in E$ exists if and only if the nodes are within communication range of each other. The node degree is defined as the number of edges connected to the node. For example, in Figure 1, $\deg n_4 = 3$. Now, let us assume that node n_i wishes to send information to the node $n_{i'}$, and let $P(n_i, n_{i'})$ be the received power at $n_{i'}$. In this case, gateway G_j compares the SNR with the environment noise threshold, and if it is more than the noise threshold, then n_i can send a message to the $n_{i'}$. In this situation, these nodes have achieved survivable connectivity and the edge $(n_i, n_{i'})$ exists. To obtain the $P(n_i, n_{i'})$ in each cluster, the following steps should be completed.

(1) The gateway broadcasts a start message.

(2) Each sensor node n_i transmits a message with its ID_{n_i}.

(3) All the sensor nodes record the received signal strength.

(4) The gateways request each sensor node to report (the recorded information) to the gateway.

To achieve secure connectivity, in addition to the above conditions for survivable connectivity, sensor nodes should have previously established a symmetric/secret common key $K^{n_{i'}}_{n_i}$ for each edge in E. In this case, the proposed graph is securely connected. Finally, the gateway G_j will be aware of the degree of each sensor node within its cluster. Note that $\deg n_i$ determines the amount of symmetric keys which should be loaded from the gateway G_j to each sensor node.

5. Node Degree Analysis in the Proposed Scheme

The proposed scheme for establishing security for clustered WSNs is based on using PKC. The required symmetric key for each sensor node depends on the node degree and routing algorithm. In the proposed scheme, each sensor node has one secure path to the gateway across multiple hops. Therefore, the degree of connectivity of each sensor node may be different. Our routing algorithm is based on minimum neighborhood path, but some sensor nodes may have a higher neighborhood degree. Therefore, it is interesting to see how many neighbors a sensor can have related to the proposed scheme.

The question is *what is the number of nodes in a certain area S in the environment of A?* Since sensor nodes have a random and uniform deployment, one can assume a Poisson distribution [11]. Therefore, the probability mass function can be defined for the random deployment as

$$P(n \mid S) = \text{Probability of } n \text{ nodes is in area } S. \tag{6}$$

From the Poisson process and node density as $\rho = N/A$, one can write

$$P(n \mid S) = \frac{(\rho S)^n}{n!} \cdot e^{-\rho S} = \frac{((N/A)S)^n}{n!} . e^{-(N/A)S}. \tag{7}$$

Then, the average number of nodes in the radius of r and area of $S = \pi r^2$ can be obtained by

$$\overline{n} = \sum_{n=0}^{N} nP(n \mid S) = \rho \cdot S = \frac{N}{A} S = \frac{N}{A} \pi r^2. \tag{8}$$

To determine the probability of having average number of sensor nodes in neighborhood of a sensor node, one can write

$$\Pr(n = \overline{n} \mid S) = \frac{(\rho \cdot S)^{\rho \cdot S}}{(\rho \cdot S)!} \cdot e^{-\rho \cdot S}. \tag{9}$$

As the $\rho \cdot S >> 1$ regards the Sterling's formulas, one can simplify that

$$\Pr(n = \overline{n} \mid S) = \frac{1}{\sqrt{2\pi\rho \cdot S}}. \tag{10}$$

It is interesting to note that the density of sensor nodes after the clustering will be the same because the deployment of sensor nodes is randomly uniform.

To calculate the probability that each sensor node has at least n neighbors, the minimum node degree can be written as follows:

$$\Pr(d \geq n) = \left(1 - \sum_{D=0}^{n-1} P(D \mid S)\right)^N. \tag{11}$$

As an example, assume that $N = 1000$ nodes are to be deployed randomly in an area of $A = 1000 \times 1000 \text{ m}^2$ and the transmission range of each sensor node $r = 100$ m. From (8), the average number of neighbor nodes is found as $\overline{n} \approx 32$, and the probability of having this as neighbor degree is about 7.2% (10). Note that the number of neighbor nodes defines the $\deg n_i$ and the number of symmetric keys that should be stored dynamically in each sensor node consequently.

As shown in Figure 1, the one-hop neighbors for gateways G_1 and G_2 are $\{n_1, n_3, n_7\}$ and $\{n_8, n_{10}, n_{14}\}$, respectively. To establish secure communication between nodes in routing path, the gateway G_1 sends secret keys to the sensor node within its cluster by encrypting them with the public key of the given node. For example, one-hop neighbors of sensor node n_{10} are $\{n_{11}, n_{12}, n_{13}\}$, then it receive these $\{K^{n_{10}}_{n_{11}}, \ K^{n_{10}}_{n_{12}}, \ K^{n_{10}}_{n_{13}}\}$ symmetric keys encrypted with $P^u_{n_{10}}$. All the sensor nodes in the network will get the secret key shared with their neighborhood nodes similarly.

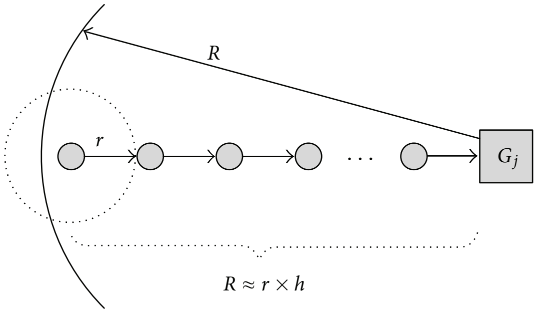

Figure 3: Approximating the cluster size from the number of hops and average node degree of each sensor node.

5.1. Average Number of Sensor Nodes and Number of Hops Inside a Cluster. Since we assumed the sensor nodes to be uniformly deployed in the field, we propose the following approximation for the average number of nodes per cluster and cluster size. Let N_c be the number of the sensor nodes inside a cluster with radius R. It is clear that, N_c follows the Poisson distribution similar to the node degree analysis introduced before (7). Then, N_c can be calculated as

$$\overline{N_c} = \frac{N}{A}\pi R^2, \tag{12}$$

where $\overline{N_c}$ is the average number of sensor nodes inside the cluster. Employing $R = h \times r$,

$$\overline{N_c} = \frac{N}{A}\pi h^2 r^2, \tag{13}$$

where h is the maximum number of hops between a node and the gateway as shown in Figure 3. From (8), then

$$\overline{N_c} = \overline{n}h^2, \tag{14}$$

and the number of hops can be approximated as

$$h = \left\lceil \sqrt{\frac{\overline{N_c}}{\overline{n}}} \right\rceil. \tag{15}$$

It should be noted that in a real scenario with a fixed range of gateway, R, increasing the range of each sensor node, r, should be accompanied by decreasing the number of hops for energy saving purposes and node lifetime. Therefore, the average number of sensor nodes inside a cluster remains unchanged. As illustrated in Table 2, we vary the range of sensor nodes from 25 m up to 100 m and obtain the relevant maximum number of hops.

6. Performance Analysis

Here, we analyze the memory storage, communication overhead, and resiliency for the proposed scheme.

6.1. Link Compromise Probability. The previously proposed schemes based on probabilistic key pre-distribution, and there is a known trade-off between the secure connectivity, memory storage, and resiliency against node capture. Here, we adopted the definition of resiliency as proposed entirely in [14].

Table 2: Analytical number of hops with various sensor node transmission ranges for a fixed gateway range $R = 200$.

r	$\overline{n}$	$\overline{N_c}$	h
25	2	128	8
50	8	128	4
75	18	128	3
100	32	128	2

Definition 6. Let us assume that x nodes are randomly captured within a cluster. Then, the probability that the link between two fixed noncompromised nodes is not affected is defined as resiliency. The inverse of resiliency also called the fraction of the network that can be compromised.

In multi-hop routing, it is commonly well known that choosing short multi-hop paths instead of long multi-hop paths is beneficial. This is because as the length of a multi-hop path (number of hops) increases, the probability of path compromise increases as well. Therefore, for the proposed scheme, we calculate the probability of the link between sensor node n_i and gateway G_j to be compromised without capturing them directly. Let us assume the following:

(i) x_i: the probability of node n_i to be compromised.

(ii) h: the number of hops from a sensor node n_i to reach the gateway G_j.

Therefore, the probability that the given path being compromised $P(l)$, given that the sensor node n_i and the gateway G_j are not compromised, is

$$\begin{aligned} P(l) &= \Pr\Big[\text{the link between sensor node } n_i \text{ and} \\ &\qquad \text{the gate way } G_j \text{ is compromised}\Big] \\ &= 1 - \Pr[\text{no node in between is compromised}] \\ &= 1 - \prod_{\text{i}=1}^{\text{h}-1}(1 - x_i). \end{aligned} \tag{16}$$

After establishing the routing algorithm, because the number of sensor nodes in neighborhood is different, the probability of node compromise directly or indirectly will be different. This compromise probability depends on the attacker model. In Figure 4, the effect of increasing, number of hops on link compromise probability is illustrated in terms of node compromise probability x_i. Since our routing algorithm is based on minimum neighborhood degree, we try to reduce the degree of each node to decrease the indirect link compromise probability and have better resiliency against node capture attack.

6.2. Simulations. We assume a network with $N = 1000$ sensor nodes is randomly and uniformly deployed in an area

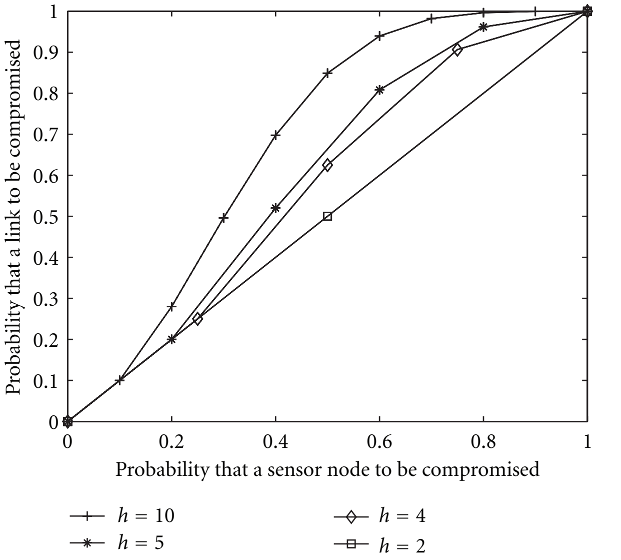

FIGURE 4: The impact of number of hops on link compromise probability.

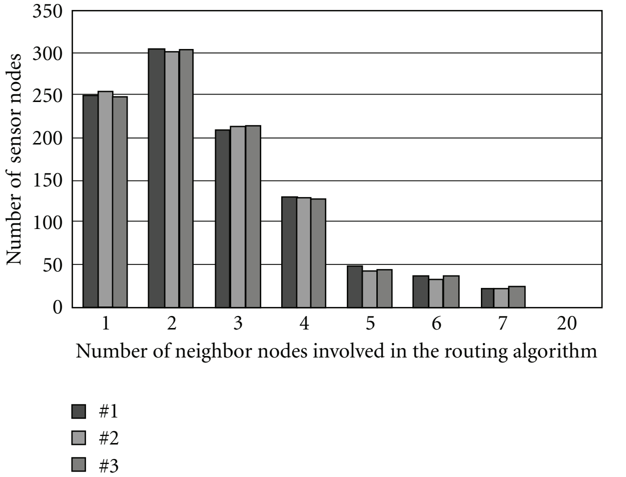

FIGURE 5: Number of neighbor nodes involved in the routing algorithm toward the gateway with $N = 1000$; $G = 10$; $r = 100$ m.

of $A = 1000 \times 1000\,\text{m}^2$. We choose the number of the gateways $G = 10$ to cover a considerable area of sensor nodes. The transmission range is varied for each sensor node from 25 m to 100 m to achieve different average node degree $\overline{n}$, ranging from 2 to 32. The maximum range of each gateway is set to $R = 200$ m. The simulations are performed using QualNet, scalable wireless network simulator [44].

Through simulations, we observe the number of neighbor nodes which are involved in the routing algorithm and are communicating securely (using allocated symmetric keys). In Figure 5, the secure neighborhood degree is plotted for each sensor node for the proposed network model. About 300 nodes are communicating with just two sensor nodes and about 25 sensor nodes are communicating with 7 other neighbor nodes securely. We run the simulations three times, and the results are almost the same. Therefore, the maximum number of symmetric keys which are required to be dynamically loaded to the sensor nodes is always less than the average number of nodes $\overline{n}$ for the proposed scheme.

6.3. Measuring Storage Saving. In this section, the memory storage requirements for sensor nodes and the gateways are analyzed. In the proposed network model, the number of gateways is much less than the number of sensor nodes, that is, $G \ll N$. As each gateway is pre-loaded with $\{P^u_{G_j}, P^r_{G_j}, P^u_{n_i}\}$, consequently the memory storage requirement for each gateway is obtained as

$$M_G = (2 + N) \times B^u, \tag{17}$$

where B^u is the key size for public key cryptography.

On the other hand, each sensor node n_i is pre-loaded with $\{P^u_{n_i}, P^r_{n_i}, P^u_{G_j}\}$. After deployment, each sensor node stores additional symmetric keys to communicate with their neighbors, that is, $\{K^{n_{i'}}_{n_i}\}$, then

$$M_n = (G + 2) \times B^u + d_m \times B^k, \tag{18}$$

where B^k denotes the size of symmetric key cryptography, and d_m is the maximum neighborhood degree.

It should be noted that since the gateways are tamper proof, the number of keys stored in each sensor node can be further reduced by incorporating the same pair of public and private keys for all the gateways, that is, P^u_G and P^r_G. Therefore, the total memory storage requirement for each sensor node can be written as

$$M_n = 3 \times B^u + d_m \times B^k. \tag{19}$$

TABLE 3: Number of encryption/decryption during secure clustering and pairwise key establishment.

Operation	No. of computations
Secure clustering	
ECDS generation, and broadcast $G_j \to n_i$	G
ECDS verification by n_i	N
Encryption $E_{P^u_{G_j}}(\cdot)$, $n_i \to G_j$	N
Decryption $D_{P^r_{G_j}}(\cdot)$ by G_j	N
Pairwise key establishment	
ECDS and encryption by $E_{P^u_{n_i}(\cdot)}$, $G_j \to n_i$	G
ECDS verification and decryption by $D_{P^r_{n_i}(\cdot)}$	N

The proposed scheme requires less memory space than probabilistic schemes based on the work proposed in [11, 14], where those schemes require $m \times B^k$ bits. As an example, assume that ECC (163-bit) is used for the communication between sensor nodes and the gateway and the SKIPJACK (83-bit) cryptography is used in the communication between each sensor node and its neighbors. Therefore, from (19), the worst case memory requirement for each sensor node is

Table 4: Comparison of the proposed scheme with recent existing works.

Property	Proposed	[31]	12]
Resiliency	High	Low	Low
Key establishment	Guaranteed	Not guaranteed	Not guaranteed
Scalability	Scalable	Authentication problem	Not scalable
Memory requirement	Efficient	Less efficient	Efficient

$M_n = (3) \times 163 + 7 \times (83) = 1,070$ bits. As shown in our simulation results in Figure 5, the maximum node degree in the proposed scheme is 7. However, in the probabilistic schemes, the storage requirement is $(200) \times 83 = 16,600$ bits. The scheme proposed in [31] requires $54 \times 83 = 4,482$ bits to be stored in each sensor node for the balanced scheme and $30 \times 83 = 2,490$ bits for the unbalanced scheme with connectivity of 67%. Therefore, the proposed approach saves almost 57% of memory storage in comparison with the scheme presented in [31]. Note that the proposed scheme is deterministic and completely connected. As one can deduce from (17), the number of keys stored in each gateway is 1,002 keys. Note that in this work as well as [12, 31] and several previous works reviewed in this paper, it is assumed that gateways are more powerful than the sensor nodes in terms of memory, computation, and communication capabilities. In Table 4, the proposed scheme is qualitatively compared with its counterparts.

6.4. Communication and Computation Overheads. Inherently, randomized key predistribution schemes (including the basic scheme and its extended schemes reviewed in this paper) suffer from lack of structure because the key ring k is chosen randomly from a key pool. Consequently, the communication complexity is $\Theta(k)$, and increasing k results in a dramatic increase in communication overhead. The number of messages passed in the network is a metric related to the power consumption and communication overhead. It is well known that transmitting is the most costly operation on a sensor node (e.g., the cost of transmitting one bit of data using MICA mote sensor node is approximately equivalent to processing 1000 CPU instructions) [45]. We define the communication overhead as the sum of packets sent and received per cluster in the network. The average number of packets can be estimated as the sum of the following.

(i) Packets sent from G_j to n_i as a message B in each cluster.

(ii) Packets sent by each sensor node toward the gateway within the cluster as a message A.

(iii) Unicast encrypted messages (pairwise secret keys) that each gateway sent to the nodes within its cluster $(K_{n_i}^{n_{i'}})$.

6.4.1. Cost of Secure Clustering and Pairwise Key Establishment. In Table 3, the number of encryptions and decryptions during the secure clustering and pairwise key establishment is reported. Therefore, the cost of secure clustering, i.e., C_{SC}, can be formulated as follows

$$\begin{aligned} C_{\text{SC}} = {} & G \times C_{\text{ECDS}_{P^r_{G_j}}} + N \times C_{\text{ECDSV}_{P^u_{G_j}}} \\ & + N \times C_{E_{P^u_{G_j}}}(\cdot) + N \times C_{D_{P^r_{G_j}}}(\cdot), \end{aligned} \tag{20}$$

where $C_{\text{ECDS}_{P^r_{G_j}}}$ is the cost of generating an elliptic curve digital signature using private key of gateway G_j, $C_{\text{ECDSV}_{P^u_{G_j}}}$ is the cost of verifying the signature using the public key of gateway G_j by sensor node n_i, $C_{E_{P^u_{G_j}}}(\cdot)$ is the cost of an encryption using public key of gateway G_j by sensor node n_i, and $C_{D_{P^r_{G_j}}}(\cdot)$ is the cost of a decryption using the private key of the gateway G_j performed by the gateway G_j.

6.5. Compromise Analysis and Key Revocation. Sensor nodes are deployed physically in insecured environments; hence, they are prone to be compromised. When a sensor node is captured, we assume that all information and stored key materials will be exposed to the adversary. In the proposed key management scheme, each sensor node stores the pairwise keys between its potential neighbors. After an adversary captures one of its neighbor nodes, she will be able to decrypt the information coming from other neighbor nodes directly. But other links which are not involved directly in this communication will remain secure. Therefore, the resiliency of the scheme is high because of its deterministic nature.

The problem which remains is the injection of false data into the network by the adversary. In this case, an efficient malicious behavior detection scheme is required to identify the misbehaving nodes and revoke them and their keys from the network. In the distributed and homogeneous WSNs, the resource constraint nature of sensor nodes limits the memory, computation, and communication resources which can be used for revocation. In [46], an efficient misbehaving detection scheme based on artificial immune system (AIS) for distributed sensor networks has been presented.

In clustered WSNs using public key infrastructure, a gateway as a certificate authority (CA) can issue a certificate revocation list (CRL) containing a list of keys to be revoked. Since, in the proposed scheme, node-to-node authentication is considered with the pairwise key allocation, then detecting and reporting misbehaved nodes is possible.

Upon detection of a misbehaving node by the gateway, a digital signature including the IDs of all the pairwise keys stored in that node can be generated and broadcast within the entire cluster as follows:

$$\left\langle K_{n_i}^{n_{i'}} \| \mathrm{ECDSA} \right\rangle, \quad \text{for } i, i' \in \{1, \dots, N\}. \tag{21}$$

Note that in the scheme presented in [31] for heterogeneous and hierarchical WSNs, key revocation is not considered. In Table 4, resiliency of the proposed scheme is compared with the counterparts.

6.6. Scalability Analysis of the Proposed Scheme. The main drawback of the pairwise scheme proposed previously for distributed and homogeneous WSNs is scalability. In those networks, if the size of the network increases, the number of keys required to be stored in each sensor node will increase. Note that in the proposed scheme, adding new nodes to the network can be achieved easily by forwarding the required session *key-request* message to its potential neighbors and then toward the gateway. Upon authenticating the gateway, it can join the network via other nodes securely. Therefore, the proposed scheme is scalable.

7. Conclusions and Future Work

In this paper, we have proposed a new secure clustering scheme for clustered WSNs incorporating public key cryptography. We take advantage of gateway nodes which are powerful and tamper proof to establish/revocate the symmetric keys in each cluster. This key establishment is completed during the bootstrapping and clustering phase assuming that the adversary is present in the field. We have presented an approximation to determine the number of neighbor nodes for each sensor node obtained from the average number of neighbor nodes involved in the routing algorithm toward the gateway. Consequently, we have analyzed the number of keys which are required to be dynamically loaded to each sensor node, and a considerable saving in memory requirements is achieved. High resiliency against node capture and node-to-node authentication is accomplished by the proposed scheme. We note that we have not considered the overhead of the broadcasts from the gateways, as we assumed that they are powerful. However, applying network coding schemes will be considered to reduce these overheads in future works.

Acknowledgments

The authors would like to thank the anonymous reviewers for the careful reading and valuable comments that helped to improve this paper. This research is supported in part by the NSERC Discovery grant awarded to A. Reyhani-Masoleh.

References

[1] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, "A survey of key management schemes in wireless sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2314–2341, 2007.

[2] L. Oliveira, H. Wong, A. Loureiro, and R. Dahab, "On the design of secure protocols for hierarchical sensor networks," *International Journal of Security and Networks*, vol. 2, no. 3, pp. 216–227, 2007.

[3] *IEEE Std.802.15.4 for Information Technology—Telecommunication and Information Exchage between Systems—Local and Metropolitan Area Networks Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks*, IEEE, 2006.

[4] V. Westmark, "A definition for information system survivability," in *Proceeding of the 37th Hawaii Internal Conference on System Sciences (HICSS '04)*, pp. 2086–2096, IEEE press, 2004.

[5] K. Akkaya and M. F. Younis, "Energy and QoS aware routing in wireless sensor networks," *Cluster Computing*, vol. 8, no. 2-3, pp. 179–188, 2005.

[6] C. P. Low, C. Fang, J. M. Ng, and Y. H. Ang, "Efficient load-balanced clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 31, no. 4, pp. 750–759, 2008.

[7] M. F. Younis, K. Ghumman, and M. Eltoweissy, "Location-aware combinatorial key management scheme for clustered sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 8, pp. 865–882, 2006.

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

[9] Q. Tian and E. J. Coyle, "Optimal distributed detection in clustered wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3892–3904, 2007.

[10] J. Deng and Y. S. Han, "Multipath key establishment for wireless sensor networks using just-enough redundancy transmission," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 3, Article ID 4378397, pp. 177–190, 2008.

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

[12] K. Lu, Y. Qian, M. Guizani, and H.-H. Chen, "A framework for a distributed key management scheme in heterogeneous wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 2, pp. 639–647, 2008.

[13] W. Zhang, M. Tran, S. Zhu, and G. Cao, "A random perturbationbased scheme for pairwise key establishment in sensor networks," in *Proceedings of the 8th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing, (MobiHoc '07)*, E. Kranakis, E. M. Belding, and E. Modiano, Eds., pp. 90–99, ACM, 2007.

[14] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the IEEE Symposium on Security And Privacy*, pp. 197–213, May 2003.

[15] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 2, pp. 228–258, 2005.

[16] D. Liu, P. Ning, and L. I. Rongfang, "Establishing pairwise keys in distributed sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 1, pp. 41–77, 2005.

[17] Y. Xue, H. Jürgensen, R. Azarderakhsh, and A. Reyhani-Masoleh, "Key management for wireless sensor networks using trusted neighbors," in *Proceedings of IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC '08)*, C.-Z. Xu and M. Guo, Eds., vol. 2, pp. 228–233, IEEE Computer Society, 2008.

[18] C. Ma, Z. Shang, H. Wang, and G. Geng, "An improved key management scheme for heterogeneity wireless sensor networks," in *Proceedings of the 3rd International Conference on Mobile Ad-Hoc and Sensor Networks (MSN '07)*, pp. 854–865, 2007.

[19] D. Malan, M. Welsh, and M. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," in *Proceedings of 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON '04)*, pp. 71–80, 2004.

[20] A. Liu, P. Kampanakis, and P. Ning, "Tinyecc: elliptic curve cryptography for sensor networks (version 0.3)," 2007, http://discovery.csc.ncsu.edu/software/TinyECC.

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

[22] W. Shih, W. Hu, P. Corke, and L. Overs, "A public key technology platform for wireless sensor networks," in *Proceedings of the 6th ACM conference on Embedded Network Sensor Systems*, pp. 447–448, ACM, 2008.

[23] B. Panja and S. K. Madria, "An energy and communication efficient group key in sensor networks using elliptic curve polynomial," in *Proceedings of the 6th International Conference on Ad-Hoc, Mobile and Wireless Networks (ADHOC-NOW '07)*, pp. 153–171, Springer, 2007.

[24] P. Szczechowiak, L. Oliveira, M. Scott, M. Collier, and R. Dahab, "Nanoecc: testing the limits of elliptic curve cryptography in sensor networks," in *Proceedings of European Conference on Wireless Sensor Networks (EWSN '08)*, pp. 305–320, Springer, 2008.

[25] M. Čagalj, J.-P. Hubaux, and C. C. Enz, "Energy-efficient broadcasting in all-wireless networks," *Wireless Networks*, vol. 11, no. 1-2, pp. 177–188, 2005.

[26] G. Jolly, M. C. Kusçu, P. Kokate, and M. F. Younis, "A lowenergy key management protocol for wireless sensor networks," in *Proceedings of the 8th IEEE Symposium on Computers and Communications (ISCC '03)*, pp. 335–340, 2003.

[27] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 24–34, 2007.

[28] F. Kausar, S. Hussain, L. T. Yang, and A. Masood, "Scalable and efficient key management for heterogeneous sensor networks," *Journal of Supercomputing*, vol. 45, no. 1, pp. 44–65, 2008.

[29] R. Azarderakhsh, A. Reyhani-Masoleh, and Z.-E. Abid, "A key management scheme for cluster based wireless sensor networks," in *Proceedings of IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC '08)*, C.-Z. Xu and M. Guo, Eds., pp. 222–227, 2008.

[30] D. Johnson, A. Menezes, and S. A. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, 2001.

[31] P. Traynor, R. Kumar, H. Choi, G. Cao, S. Zhu, and T. La Porta, "Efficient hybrid security mechanisms for heterogeneous sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 663–676, 2007.

[32] J. Brown, X. Du, and K. Nygard, "An efficient public-key-based heterogeneous sensor network key distribution scheme," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 991–995, November 2007.

[33] SK. MD. M. Rahman, N. Nasser, and K. Saleh, "Identity and pairing-based secure key management scheme for heterogeneous sensor networks," in *Proceedings of the 4th IEEE International Conference on Wireless and Mobile Computing, Networking and Communication (WiMob '08)*, pp. 423–428, October 2008.

[34] K. Ren, S. Yu, W. Lou, and Y. Zhang, "Multi-user broadcast authentication in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, pp. 4554–4564, 2009.

[35] S. Khuller, B. Raghavachari, and N. Young, "Balancing minimum spanning trees and shortest-path trees," *Algorithmica*, vol. 14, no. 4, pp. 305–321, 1995.

[36] W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *Proceedings of 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 4, pp. 586–597, IEEE, 2004.

[37] P. Ning, A. Liu, and W. Du, "Mitigating DoS attacks against broadcast authentication in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 1, article no. 1, pp. 1–35, 2008.

[38] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *Proceedings of the Conference on Computer Communications (ACM SIGCOMM '04)*, pp. 121–131, usa, September 2004.

[39] B. Awerbuch and R. G. Gallager, "A new distributed algorithm to find breadth first search trees," *IEEE Transactions on Information Theory*, vol. 33, no. 3, pp. 315–322, 1987.

[40] C. Busch, M. Magdon-Ismail, F. Sivrikaya, and B. Yener, "Contention-free MAC protocols for asynchronous wireless sensor networks," *Distributed Computing*, vol. 21, no. 1, pp. 23–42, 2008.

[41] R. Cristescu and B. Beferull-Lozano, "Lossy network correlated data gathering with high-resolution coding," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2817–2824, 2006.

[42] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609–619, 2004.

[43] A. Woo, T. Tong, and D. E. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (Sensys '03)*, I. F. Akyildiz, D. Estrin, D. E. Culler, and M. B. Srivastava, Eds., pp. 14–27, ACM, 2003.

[44] I. Scalable Network Technologies, "QualNet Simulator," http://www.scalable-networks.com/products/qualnet.

[45] M. Wireless Sensor Networks, "Crossbow," http://www.xbow.com/Products.

[46] M. Drozda, S. Schaust, and H. Szczerbicka, "AIS for misbehavior detection in wireless sensor networks: performance and design principles," in *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 3719–3726, IEEE, 2007.