Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 240610, 9 pages
doi:10.1155/2009/240610

Research Article

A Flexible and Efficient Key Distribution Scheme for Renewable Wireless Sensor Networks

An-Ni Shen,[1] Song Guo,[1] and Victor Leung[2]

[1] *School of Computer Science and Engineering, University of Aizu, Fukushima-Ken 965-8580, Japan*
[2] *Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada V6T 1Z4*

Correspondence should be addressed to Song Guo, sguo@u-aizu.ac.jp

Received 1 February 2009; Accepted 11 April 2009

Recommended by Yang Xiao

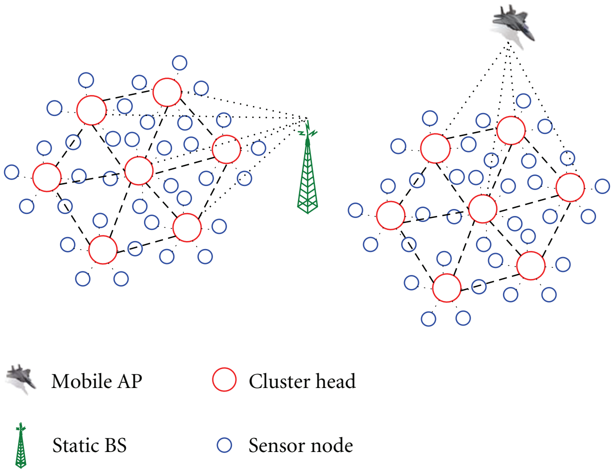
Many applications of wireless sensor network require secure data communications, especially in a hostile environment. In order to protect the sensitive data and the sensor readings, secret keys should be used to encrypt the exchanged messages between communicating nodes. Traditional asymmetric key cryptosystems are infeasible in WSN due to its low capacity at each senor node. In this paper, we propose a new key distribution scheme for hierarchical WSNs with renewable network devices. Compared to some of the existing schemes, our key establishment methods possess the following features that are particularly beneficial to the resource-constrained large-scale WSNs: (1) robustness to the node capture attack, (2) flexibility for adding new network devices, (3) scalability in terms of storage cost, and (4) low communication overhead.

1. Introduction

Wireless sensor networks (WSNs) have been envisioned to be very useful for a broad spectrum of emerging civil and military applications [1]. However, sensor networks are also confronted with many security threats such as node compromise, routing disruption, and false data injection, because they normally operate in unattended, harsh, or hostile environment. Among all these threats, the WSNs are particularly vulnerable to the node compromise because sensor nodes are not tamper-proof devices. An adversary might easily capture the sensor devices to acquire their sensitive data and keys and then abuse them to further compromise the communication between other noncaptured nodes. This typical threat is known as the *node capture attack*. In order to conquer such problem, it is desirable to design key distribution protocols to support secure and robust pairwise communication among any pair of sensors.

To prevent from the node capture attack is a challenging task in sensor networks that have scarce resources in energy, computation, and communication. Therefore, only lightweight energy efficient key distribution mechanisms are affordable. For example, the conventional asymmetric key cryptosystem, such as RSA [2] and Diffie-Hellman [3], cannot be implemented in sensor nodes due to their very limited capacities. As the first naive solution, all sensor devices are preloaded the same master key and thus any two nodes can use this master key for secure communication after deployment. However, if one sensor node is physically captured by an adversary, it would compromise the entire network secrecy. Another possible approach is to assign a distinct pairwise key for each pair of sensor nodes before they are deployed. Each sensor node needs to store $(n - 1)$ keys, where n is the size of the network. The solution provided secure against the node captured attack but not scalable. Moreover, addition of new sensors to a deployed network is extremely difficult.

WSNs can be broadly classified into flat WSNs and hierarchical WSNs. In a flat WSN, all senor nodes have the same computational and communication capacities. In a hierarchical WSN, however, some special sensor devices, called Cluster Head (CH), have much higher capacities than other sensor nodes. By applying some clustering algorithms like [4], the whole set of sensor devices could be partitioned into several distinct clusters such that each cluster has at least one CH. Under this arrangement, each sensor node forwards the generated packets to its local CH by short-range



(a) (b)

Figure 1: A three-tier hierarchical WSN.

transmissions, and the CH then performs a preprocessing for the raw data received from all other senor nodes in the cluster and finally forwards the aggregated data to the sink node, or Base Station (BS), by long-range transmissions. Key distribution protocols have already been studied comprehensively in flat WSNs, for example, in [5–8]. Recent research has more focused on the hierarchical architecture for large-scale resource-constrained WSNs, because it has been shown in [9] that a hierarchical architecture can provide better performance, in terms of communication overhead, than a flat architecture in such networks.

To solve the key agreement problem in hierarchical WSNs, Jolly et al. proposed a key predistribution scheme LEKM [10]. Before deployment, each CH stores a set of keys in its memory and each sensor node randomly selects a key from a CH and stores it with the CH's Id in its memory. After deployment, each sensor node establishes a securely link with the CH that has been selected. This is done at each sensor node by exchanging key information over the whole network. Such scheme has no computational cost at both sensor node and CH in key establishment phase and is robust against node capture attack after the key establishment phase. However, it has high storage and communication overhead at CHs.

Another proposal IKDM [9] is a polynomial-based protocol for hierarchical WSN. In the IKDM scheme, each sensor node or CH has fixed storage cost in predistribution phase. In order to improve the resilience against the node captured attack, the preloaded key of each sensor node is the exclusive-or result of ℓ ($\ell \geq 1$) number of bivariate polynomial keys which can be fetched by its CH from ℓ number of distinctive CHs all over the network. The parameter ℓ defines the tradeoff between the communication overhead and the robustness to the node capture attacks at the cluster heads. While the large ℓ can improve the security level of the network, it will also result in significant message exchanges for establishing secure links.

In real applications, new network devices need to be added into an already deployed network from time to time in order to replace the power-exhausted or compromised devices such that the performance of the whole network would not significantly degrade. However, most of schemes, for example, [9, 10], cannot provide a full solution to the key management for adding new cluster heads and sensor nodes in hierarchal renewable WSNs. In summary, the security and efficiency requirements in a WSN may include secrecy and authentication, robustness against node capture attack, dynamic membership management (including new network device addition), strong network connectivity, scalability to large-scale networks, and low complexities on memory, computation, and communication overhead. These challenges motivate us to propose scalable and robust pairwise key distribution mechanism between sensor devices in large-scale WSNs. In particular, our methods possess the following features that are particularly beneficial to the resource-constrained WSNs: (1) robustness to the node capture attack, (2) flexibility on key establishment for adding new network devices, (3) scalability in terms of storage cost, and (4) low communication overhead.

The rest of this paper is organized as follows. Section 2 presents our network model. Section 3 gives an overview of our proposal. Section 4 describes a group of protocols for our key distribution mechanism. Section 5 analyzes the security and evaluates the performance of our proposal. Section 6 summarizes our findings.

2. Network Model

As in other hierarchical models of sensor network [9–11], our system also assumes that a sensor network is divided into clusters, which are the minimum unit for detecting events. A cluster head coordinates all the actions inside a cluster and each pair of cluster heads in their transmission range can communicate directly with each other. Moreover, we assume a single base station (BS) or an access point (AP) in the network and works as the network controller to collect event data. As illustrated in Figure 1(a), the BS is a fixed infrastructure located in the network with virtually unlimited computational and communication power, unlimited memory storage capacity, and very large radio transmission range to ensure the full coverage of the whole network area. Another application scenario given in Figure 1(b) shows that the information collected by cluster heads from all its sensor nodes is retrieved by a mobile AP periodically. During the information retrieval operation, the AP broadcasts a beacon to activate cluster heads in its coverage area. Activated cluster heads then transmit their data to the AP through a common wireless channel. In the rest of paper, we use the general term BS for such network controller for describing our key distribution mechanism without discriminating the above two scenarios.

Our model has three different types of network devices: base station, cluster head, and normal sensor node. Each low-cost sensor node has low data processing capability, limited memory storage and battery power supplies, and

Table 1: Notations.

Symbol	Explanation
S_i	The Id of the sensor node $i\,(1 \leq\ i \leq n)$
CH_i	The Id of cluster head in cluster $i\,(1 \leq i \leq\ m)$
BS	The Id of the base station
$N_{\mathrm{S}}(\mathrm{CH}_a)$	The set of all sensor nodes in cluster a, that is, there is a pairwise key between CH_a and any sensor node $S_i \in N_{\mathrm{S}}(\mathrm{CH}_a)$
λ_{S}	The average number of sensor nodes in a cluster
$N_{\mathrm{CH}}(\mathrm{CH}_a)$	The set of all neighboring cluster heads of cluster a, that is, there is a pairwise key between CH_a and any cluster head $\mathrm{CH}_b \in N_{\mathrm{CH}}(\mathrm{CH}_a)$
λ_{CH}	The average number of neighboring cluster heads for a cluster head

short radio transmission range. Sensor nodes are restricted to direct communications with its CH only. The CHs are equipped with high power batteries, large memory storages, powerful antenna and data processing capacities, and thus can execute relatively complicated numerical operations. As the most powerful node in a WSN, the BS works as the central controller for data collect and key management. For the latter function, the BS maintains the topology of the whole network (the Ids of network devices and their connectivity information) and the method to generate keys for any secure link just based on Ids. In particular, we introduce two working modes for the BS: (1) on-line mode and (2) off-line mode.

In an on-line working mode, the key generation method at the BS can be requested from any cluster head and the BS should response in a timely manner. However, such on-line service is not always available at the BS. For example, the BS cannot response the request in certain period of time, in which it is already dedicated to some important and uninterruptable tasks as illustrated in Figure 1(a), or the requesting cluster head is not in its service area as illustrated in Figure 1(b). Under both cases, the BS is configured to work in the off-line mode, and the alternative methods for key generation relying on other network devices should be provided by the key distribution protocol.

A three-tier hierarchical wireless sensor network can thus be modeled as a simple graph G with a finite node set, including a base station, m cluster heads, and n sensor nodes. A secure wireless link corresponding to the wireless communication channel belongs to the arc set of G only if there exists a pairwise key between the transmission nodes of the link. In Table 1, we summarize the notations used in the rest of the paper.

3. Overview of Our Key Distribution Scheme

In this section, we present the foundations and basic idea of our key distribution scheme based on a three-tier hierarchal network model.

3.1. Key Distribution in Renewable WSNs. Specifics of wireless sensor networks, such as strict resource constraints and large network scalability, require a proposed security protocol to be not only secure but also efficient. Recent research shows that preloading symmetric keys into sensors before they are deployed is a practical method to deal with the key distribution and management problem in wireless sensor networking environments. After the deployment, if two neighboring nodes have some common keys, they can setup a secure link by the shared keys. As surveyed in [9], the existing schemes can be classified into the following three categories: random key predistribution schemes, polynomial-key predistribution schemes, and location-based key predistribution schemes.

In our key distribution scheme, a key distribution server (KDS) is available for both of the following cases. (1) KDS is installed in the base station, by which the keys can be delivered instantaneously when the BS is on-line to the requester. (2) It is available to the network deployer when the keys are required to be preloaded into network devices.

In many applications, new network devices need to be replenished into an already deployed network to replace the power-exhausted or compromised devices. The corresponding key management should be provided in order to setup the secure link between a new added network device and an existing one. To our best knowledge, there are no full solutions to the dynamic membership management for key distribution in hierarchal WSNs with renewable cluster head and sensor node. For example, some of them can only support the sensor node addition in the case when BS is on-line. The objective of our key distribution protocols is to provide a complete and flexible solution for such renewable WSNs. In particular, we will provide the key distribution protocols for both sensor node and cluster head when the BS is on-line or off-line.

3.2. Symmetric Polynomial Function. In our key distribution scheme, a bivariate symmetric polynomial function (s.p.f.) is used to generate the key for each link of the network. The t-degree bivariate symmetric polynomial function $f(x, y)$, introduced in [12], is defined as

$$f(x, y) = \sum_{i,j=0}^{t} a_{ij} x^i y^j. \tag{1}$$

The coefficients $a_{ij}\ (0 \leq i, j \leq t)$ are randomly chosen from a finite field $GF(Q)$, in which Q is a prime number that is large enough to accommodate a cryptographic key. As implied by its name, the symmetric property of a bivariate polynomial function satisfies $f(x, y) = f(y, x)$. In our key distribution scheme, the KDS maintains two bivariate polynomial functions:

(i) the s.p.f. $f_{\text{CH-NS}}(x, y)$ is used to establish the key between existing cluster head and new sensor node,

(ii) the s.p.f. $f_{\text{CH-NCH}}(x, y)$ is used to establish the key between existing cluster head and new cluster head.

After the pairwise key $K_{a,b}$ between network devices a and b is generated from the above polynomial functions by

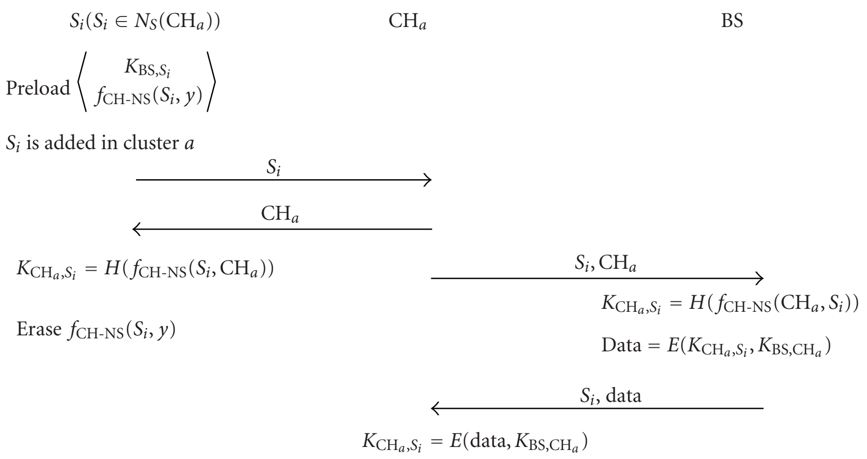

$S_i(S_i \in N_S(\mathrm{CH}_a))$
CH_a
BS
Preload $\left\langle \begin{matrix} K_{\mathrm{BS},S_i} \\ f_{\text{CH-NS}}(S_i, y) \end{matrix} \right\rangle$
S_i is added in cluster a
S_i
CH_a
$K_{\mathrm{CH}_a,S_i} = H(f_{\text{CH-NS}}(S_i, \mathrm{CH}_a))$
S_i, CH_a
$K_{\mathrm{CH}_a,S_i} = H(f_{\text{CH-NS}}(\mathrm{CH}_a, S_i))$
Erase $f_{\text{CH-NS}}(S_i, y)$
$\text{Data} = E(K_{\mathrm{CH}_a,S_i}, K_{\mathrm{BS},\mathrm{CH}_a})$
S_i, data
$K_{\mathrm{CH}_a,S_i} = E(\text{data}, K_{\mathrm{BS},\mathrm{CH}_a})$

Figure 2: Protocol illustration of adding a new sensor node when BS is on-line.

substituting the variables with Ids of the two communicating parties, the *data* over the link can therefore be securely transmitted as $E(data,\ K_{a,b})$, which is a symmetric encryption function using $K_{a,b}$ as the key.

By applying the symmetric property, a secure link can be easily built up by just exchanging the Ids of transmission nodes. However, such scheme suffers the t-security problem, which means a t-degree bivariate polynomial key scheme can only keep secure against coalitions of up to t compromised sensors. When the number of compromised nodes is less than t, the coefficients of the polynomial cannot be derived even all the compromised nodes put their stored information together. But once more than t nodes are compromised, the adversary can crack the coefficients of the polynomial such that all the pairwise keys in the entire group would be cracked. Although increasing the value of t can improve the security property of bivariate polynomial key scheme, it is not suitable for wireless sensor networks due to the limited memory size of sensors. In order to conquer this limitation, the pairwise key x calculated from the polynomials will be further scrambled by a one-to-one hash function $H(x)$.

4. Key Distribution Protocols

Our scheme supports new network device (sensor node and cluster head) addition for both BS on-line and off-line scenarios with the minimum assumption that the deployed network has completed its key establishment, that is, the key $K_{a,b}$ for any secure link (a, b) is already shared by both network devices a and b. Furthermore, our proposed scheme can provide forward secrecy as well as full prevention from the node capture attack for large-scale sensor networks.

4.1. BS is On-Line. Let S_i be the new sensor node to be added in the network. In order to calculate the key between S_i and its cluster head, the calculation can be done at the BS if it is working at the on-line mode. Suppose new sensor node S_i is randomly added into the network and eventually belongs to cluster CH_a. The following Protocol 1, as illustrated in Figure 2, is to establish a secure link between S_i and CH_a.

Protocol 1 (sensor addition when BS is on-line).

(1) The new sensor node S_i is randomly deployed to the existing network with preloaded information: the s.p.f. $f_{\text{CH-NS}}(S_i, y)$ and a key K_{BS,S_i}.

(2) After S_i is deployed, it exchanges Ids with its cluster head CH_a.

(3) S_i evaluates its stored s.p.f. $f_{\text{CH-NS}}(S_i, y)$ at $y = \mathrm{CH}_a$ to establish the key between itself and its cluster head as $K_{\mathrm{CH}_a,S_i} = H(f_{\text{CH-NS}}(S_i, \mathrm{CH}_a))$. After calculating the pairwise key, S_i erases the preloaded s.p.f. $f_{\text{CH-NS}}(S_i, y)$ immediately to avoid potential attacks.

(4) CH_a requests the new key between CH_a and S_i from BS by forwarding the Id of S_i and its own Id.

(5) BS then calculates the corresponding key using the s.p.f. $f_{\text{CH-NS}}$ as and returns the encrypted key $E(K_{\mathrm{CH}_a,S_i}, K_{\mathrm{BS},\mathrm{CH}_a})$ back to CH_a.

(6) CH_a decrypts the received date to recover K_{CH_a,S_i} using the key $K_{\mathrm{BS},\mathrm{CH}_a}$, which was already loaded at CH_a since its very initial deployment, that is, $K_{\mathrm{CH}_a,S_i} = E(E(K_{\mathrm{CH}_a,S_i}, K_{\mathrm{BS},\mathrm{CH}_a}), K_{\mathrm{BS},\mathrm{CH}_a})$.

Now we consider the addition of a new cluster head and the corresponding key distribution procedures when the BS is on-line. We assume the CH_a is to be replaced by a new cluster head $\mathrm{CH}_{a'}$, due to its low power level. Note that in the replacement phase of cluster head, the communication keys with existing network devices (i.e., cluster head and sensor node) are also renewed, not simply making use of the copies of the previous keys. This process avoids potential attack activities and achieves the forward secrecy. In other words, even the attacker could intercept packets and analysis data to compromise the key of old cluster head, it still cannot decrypt the secret data using the old keys.

The following Protocol 2, as illustrated in Figure 3, is to build up the keys between the new cluster head $\mathrm{CH}_{a'}$, and all existing sensor nodes S_i ($S_i \in N_S(\mathrm{CH}_a)$) in the same cluster as well as the keys between the new cluster head $\mathrm{CH}_{a'}$, and all its neighboring cluster heads CH_b ($\mathrm{CH}_b \in N_{\mathrm{CH}}(\mathrm{CH}_a)$).

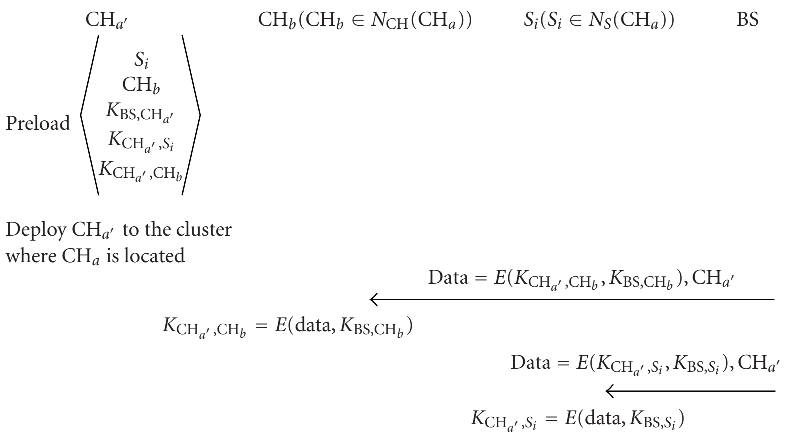

$\mathrm{CH}_{a'}$
$\mathrm{CH}_b(\mathrm{CH}_b \in N_{\mathrm{CH}}(\mathrm{CH}_a))$
$S_i(S_i \in N_S(\mathrm{CH}_a))$
BS
Preload
S_i
CH_b
$K_{\mathrm{BS},\mathrm{CH}_{a'}}$
$K_{\mathrm{CH}_{a'},S_i}$
$K_{\mathrm{CH}_{a'},\mathrm{CH}_b}$
Deploy $\mathrm{CH}_{a'}$ to the cluster where CH_a is located
Data $= E(K_{\mathrm{CH}_{a'},\mathrm{CH}_b}, K_{\mathrm{BS},\mathrm{CH}_b}), \mathrm{CH}_{a'}$
$K_{\mathrm{CH}_{a'},\mathrm{CH}_b} = E(\mathrm{data}, K_{\mathrm{BS},\mathrm{CH}_b})$
Data $= E(K_{\mathrm{CH}_{a'},S_i}, K_{\mathrm{BS},S_i}), \mathrm{CH}_{a'}$
$K_{\mathrm{CH}_{a'},S_i} = E(\mathrm{data}, K_{\mathrm{BS},S_i})$

Figure 3: Protocol illustration of adding a new cluster head when BS is on-line.

Protocol 2 (CH addition when BS is on-line).

(1) The following secret information is created and preloaded into $\mathrm{CH}_{a'}$:

(i) the pairwise key with base station $K_{\mathrm{BS},\mathrm{CH}_{a'}}$,

(ii) for each sensor node $S_i \in N_\mathrm{S}(\mathrm{CH}_a)$, its Id and the key $K_{\mathrm{CH}_{a'},S_i} = H(f_{\text{CH-NS}}(\mathrm{CH}_{a'}, S_i))$,

(iii) for each cluster heads $\mathrm{CH}_b \in N_{\mathrm{CH}}(\mathrm{CH}_a)$, its Id and key $K_{\mathrm{CH}_{a'},\mathrm{CH}_b} = H(f_{\text{CH-NCH}}(\mathrm{CH}_{a'}, \mathrm{CH}_b))$,

(2) The new cluster head $\mathrm{CH}_{a'}$ is then deployed physically to the cluster area where the old cluster CH_a is located.

(3) The base station transmits the encrypted key $E(K_{\mathrm{CH}_{a'},\mathrm{CH}_b}, K_{\mathrm{BS},\mathrm{CH}_b})$ to each neighboring cluster head CH_b of CH_a such that it can be decrypted as $K_{\mathrm{CH}_{a'},\mathrm{CH}_b}$ at the side of CH_b using the key $K_{\mathrm{BS},\mathrm{CH}_b}$, that is, $K_{\mathrm{CH}_{a'},\mathrm{CH}_b} = E(E(K_{\mathrm{CH}_{a'},\mathrm{CH}_b}, K_{\mathrm{BS},\mathrm{CH}_b}), K_{\mathrm{BS},\mathrm{CH}_b})$.

(4) Similarly, BS transmits the encrypted key $E(K_{\mathrm{CH}_{a'},S_i}, K_{\mathrm{BS},S_i})$ to each senor node S_i of CH_a such that it can be decrypted as $K_{\mathrm{CH}_{a'},S_i}$ at the side of S_i, that is, $K_{\mathrm{CH}_{a'},S_i} = E(E(K_{\mathrm{CH}_{a'},S_i}, K_{\mathrm{BS},S_i}), K_{\mathrm{BS},S_i})$, using the key K_{BS,S_i}.

4.2. BS is Off-Line

Protocol 3 (sensor addition when BS is off-line).

(1) The new sensor node S_i is randomly deployed to the existing network with the following preloaded information:

(i) the pairwise key K_{BS,S_i} shared with BS,

(ii) the Id of a cluster head CH_b, which is an arbitrary CH already in the network,

(iii) the key $K_{\mathrm{CH}_b,S_i} = H(f_{\text{CH-NS}}(S_i, \mathrm{CH}_b))$ shared with CH_b,

(iv) the encrypted key $E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{BS},\mathrm{CH}_b})$ of K_{CH_b,S_i} using $K_{\mathrm{BS},\mathrm{CH}_b}$,

(2) The added sensor node S_i sends the join-request message to the cluster head CH_a with the preloaded secret information CH_b and $E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{BS},\mathrm{CH}_b})$ and erases $E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{BS},\mathrm{CH}_b})$ afterwards.

(3) Based on CH_b, CH_a then knows to request the secret key from CH_b by providing information $E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{BS},\mathrm{CH}_b})$ and Id of S_i.

(4) After receiving the request message, CH_b uses $K_{\mathrm{BS},\mathrm{CH}_b}$ to decrypt $E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{BS},\mathrm{CH}_b})$ and obtain the pairwise key K_{CH_b,S_i}. CH_b then re-encrypts it using $K_{\mathrm{CH}_a,\mathrm{CH}_b}$ as the key and sends $E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{CH}_a,\mathrm{CH}_b})$ back to CH_a. Finally, CH_b deletes $E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{BS},\mathrm{CH}_b})$, $E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{CH}_a,\mathrm{CH}_b})$, and K_{CH_b,S_i} immediately.

(5) CH_a decrypts $E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{CH}_a,\mathrm{CH}_b})$ by $K_{\mathrm{CH}_a,\mathrm{CH}_b}$ to obtain the key K_{CH_b,S_i} with S_i.

Similar to the on-line case, we assume that the new sensor node S_i is randomly added into the network and eventually belongs to cluster CH_a. In order to create the key between S_i and CH_a, a cluster head CH_b is randomly assigned as the proxy of BS as illustrated in Figure 3. All required information to generate the key should be first forwarded to CH_b. The detailed process is described in Protocol 3.

We notice that the cluster head CH_b may be physically located far from CH_a due to the random deployment process of the sensor nodes, resulting in a relatively high communication overhead between CH_a due CH_b. In order to reduce such overhead, up to ℓ number of CHs are randomly chosen as potential proxies of BS and the corresponding keys are all generated and stored in S_i. CH_a will choose the closest one, for example, with minimum hops, as the selected proxy by looking up its routing table based on their Ids. Comparing to the on-line case, we also observe that the BS-on-line case is more efficient than the BS-off-line case in terms of communication and memory overhead when both are possible.

Finally, we consider the addition of a new cluster head when the BS is off-line. The same set of symbols as in the on-line case is used and the corresponding Protocol 4 is illustrated in Figure 5.

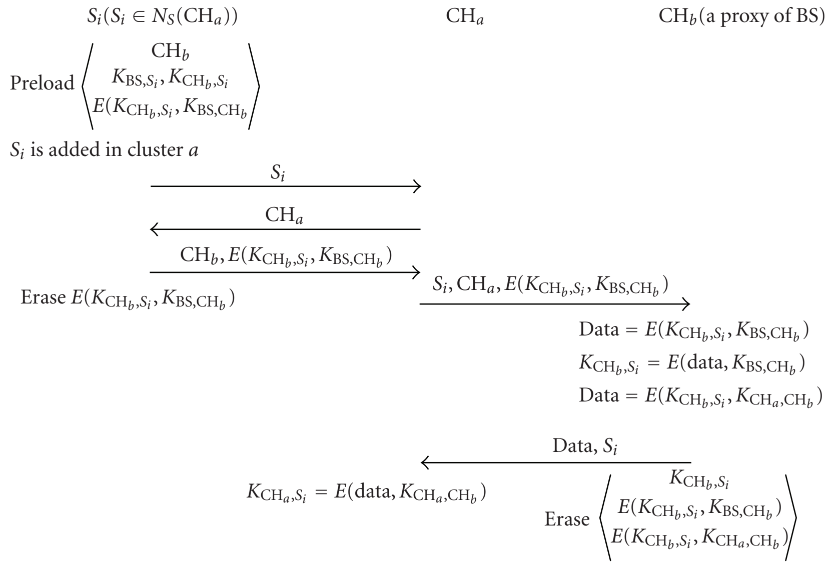

$S_i(S_i \in N_S(\mathrm{CH}_a))$
CH_a
CH_b(a proxy of BS)
Preload $\left\langle \begin{matrix} \mathrm{CH}_b \\ K_{\mathrm{BS},S_i}, K_{\mathrm{CH}_b,S_i} \\ E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{BS},\mathrm{CH}_b} \end{matrix} \right\rangle$
S_i is added in cluster a
S_i
CH_a
$\mathrm{CH}_b, E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{BS},\mathrm{CH}_b})$
Erase $E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{BS},\mathrm{CH}_b})$
$S_i, \mathrm{CH}_a, E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{BS},\mathrm{CH}_b})$
$\mathrm{Data} = E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{BS},\mathrm{CH}_b})$
$K_{\mathrm{CH}_b,S_i} = E(\mathrm{data}, K_{\mathrm{BS},\mathrm{CH}_b})$
$\mathrm{Data} = E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{CH}_a,\mathrm{CH}_b})$
Data, S_i
$K_{\mathrm{CH}_a,S_i} = E(\mathrm{data}, K_{\mathrm{CH}_a,\mathrm{CH}_b})$
Erase $\left\langle \begin{matrix} K_{\mathrm{CH}_b,S_i} \\ E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{BS},\mathrm{CH}_b}) \\ E(K_{\mathrm{CH}_b,S_i}, K_{\mathrm{CH}_a,\mathrm{CH}_b}) \end{matrix} \right\rangle$

Figure 4: Protocol illustration of adding a new sensor node when BS is off-line.

Protocol 4 (CH addition when BS is off-line).

(1) The following secret information is created and preloaded into $\mathrm{CH}_{a'}$:

  (i) the pairwise key with base station $K_{\mathrm{BS},\mathrm{CH}_{a'}}$,

  (ii) for each sensor $S_i \in N_\mathrm{S}(\mathrm{CH}_a)$, its Id, the key $K_{\mathrm{CH}_{a'},S_i} = H(f_{\text{CH-NS}}(\mathrm{CH}_{a'}, S_i))$ and encrypted key $E(K_{\mathrm{CH}_{a'},S_i}, K_{\mathrm{BS},S_i})$,

  (iii) for each cluster head $\mathrm{CH}_b \in N_\mathrm{CH}(\mathrm{CH}_a)$, its Id, the key $K_{\mathrm{CH}_{a'},\mathrm{CH}_b} = H(f_{\text{CH-NCH}}(\mathrm{CH}_{a'}, \mathrm{CH}_b))$ and the encrypted key $E(K_{\mathrm{CH}_{a'},\mathrm{CH}_b}, K_{\mathrm{BS},\mathrm{CH}_b})$,

(2) The new cluster head $\mathrm{CH}_{a'}$ is then deployed physically to the cluster area where the old cluster CH_a is located.

(3) $\mathrm{CH}_{a'}$ exchanges Ids with each sensor node $S_i \in N_\mathrm{S}(\mathrm{CH}_a)$ and then sends S_i the corresponding encrypted key $E(K_{\mathrm{CH}_{a'},S_i}, K_{\mathrm{BS},S_i})$. After that the new cluster head $\mathrm{CH}_{a'}$ erases $E(K_{\mathrm{CH}_{a'},S_i}, K_{\mathrm{BS},S_i})$ immediately. Each sensor node S_i then decrypts the received information to recover the key $K_{\mathrm{CH}_{a'},S_i}$.

(4) $\mathrm{CH}_{a'}$ exchanges Ids with each neighboring cluster head $\mathrm{CH}_b \in N_\mathrm{CH}(\mathrm{CH}_a)$ and then sends CH_b the corresponding encrypted key $E(K_{\mathrm{CH}_{a'},\mathrm{CH}_b}, K_{\mathrm{BS},\mathrm{CH}_b})$. After that the new cluster head $\mathrm{CH}_{a'}$ erases $E(K_{\mathrm{CH}_{a'},\mathrm{CH}_b}, K_{\mathrm{BS},\mathrm{CH}_b})$ immediately. Each cluster head CH_b decrypts the received information to recover the key $K_{\mathrm{CH}_{a'},\mathrm{CH}_b}$.

5. Security and Performance Evaluation

In this section, we will analyze the security and evaluate the performance of our proposed scheme by comparing with IKDM [9] and LEKM [10].

We note that neither of IKDM and LEKM protocols supports cluster head addition process. Regarding the sensor node addition process, we have the following observations. Recall that in the IKDM scheme, the polynomial functions to be used for key generation are stored in CHs all the time and thus no on-line BS is required. As we shall later, while it simplifies the process by avoiding the involvement of BS, potential security problem has been neglected. In the LEKM scheme, the preloaded key at each sensor node must be stored in some cluster head as well. If the key assigned to the new sensor node has not been preloaded to some CH at very initial deployment of the network, such key must be distributed to a CH as well by the on-line BS. Therefore, in the following evaluation, we only consider the off-line BS case and on-line BS case for the IKDM and LEKM protocols, respectively, in the senor node addition process.

5.1. Security Analysis. The security is analyzed in terms of the ability to defend from the node capture attack, which means the capture of some nodes may compromise the communication between other noncaptured nodes. This is recognized as the major threat in wireless sensor networks. In particular, we consider the security property of all these schemes in two typical scenarios: the fractions of compromised keys in noncaptured sensor nodes as a function of the number of compromised cluster heads and the number of sensor node, respectively.

Because only pairwise keys are remained in the sensor nodes for all schemes after deployment the network, that is, all security parameters that will not be used in the future have been already erased from the network, any sensor node's compromising will not endanger the secret communications of other noncaptured nodes. In other words, all these schemes have full ability to defense the node capture attack at sensor nodes.

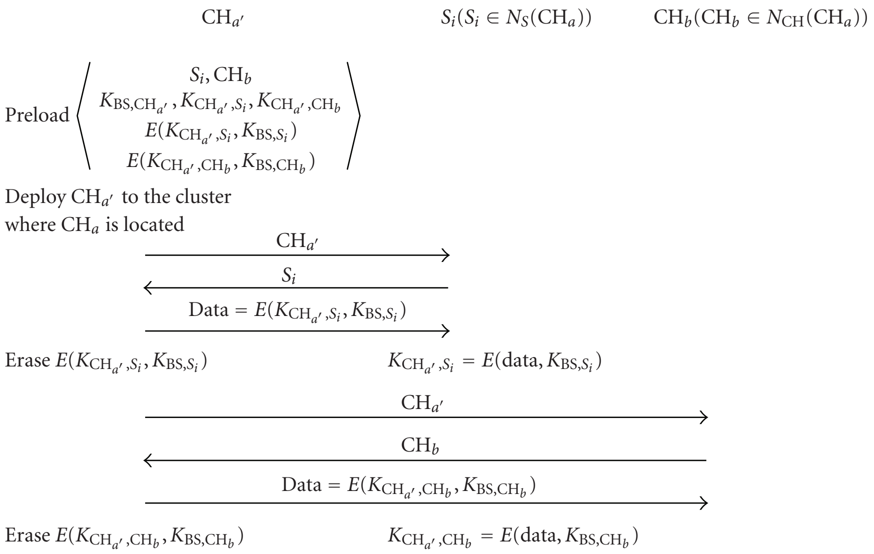

$\mathrm{CH}_{a'}$
$S_i(S_i \in N_S(\mathrm{CH}_a))$
$\mathrm{CH}_b(\mathrm{CH}_b \in N_{\mathrm{CH}}(\mathrm{CH}_a))$
Preload
S_i, CH_b
$K_{\mathrm{BS},\mathrm{CH}_{a'}}, K_{\mathrm{CH}_{a'},S_i}, K_{\mathrm{CH}_{a'},\mathrm{CH}_b}$
$E(K_{\mathrm{CH}_{a'},S_i}, K_{\mathrm{BS},S_i})$
$E(K_{\mathrm{CH}_{a'},\mathrm{CH}_b}, K_{\mathrm{BS},\mathrm{CH}_b})$
Deploy $\mathrm{CH}_{a'}$ to the cluster where CH_a is located
$\mathrm{CH}_{a'}$
S_i
$\mathrm{Data} = E(K_{\mathrm{CH}_{a'},S_i}, K_{\mathrm{BS},S_i})$
Erase $E(K_{\mathrm{CH}_{a'},S_i}, K_{\mathrm{BS},S_i})$
$K_{\mathrm{CH}_{a'},S_i} = E(\mathrm{data}, K_{\mathrm{BS},S_i})$
$\mathrm{CH}_{a'}$
CH_b
$\mathrm{Data} = E(K_{\mathrm{CH}_{a'},\mathrm{CH}_b}, K_{\mathrm{BS},\mathrm{CH}_b})$
Erase $E(K_{\mathrm{CH}_{a'},\mathrm{CH}_b}, K_{\mathrm{BS},\mathrm{CH}_b})$
$K_{\mathrm{CH}_{a'},\mathrm{CH}_b} = E(\mathrm{data}, K_{\mathrm{BS},\mathrm{CH}_b})$

Figure 5: Protocol illustration of adding a new cluster head when BS is off-line.

Table 2: Storage cost comparison over various distribution schemes.

Schemes		Our protocols	IKDM	LEKM
On-line	Cluster head	$\lambda_S + \lambda_{CH}$ Ids $\lambda_S + \lambda_{CH} + 1$ keys	N/A	$\lambda_S + m$ keys
	Sensor node	One key One s.p.f.		One Id Two keys
Off-line	Cluster head	$\lambda_S + \lambda_{CH}$ Ids $2\lambda_S + 2\lambda_{CH} + 1$ keys	One key Two s.p.f.	N/A
	Sensor node	ℓ Ids $2\ell + 1$ keys	ℓ Ids Two keys	

Now we consider the security property when some cluster heads are compromised. In our key distribution protocols, because the pairwise keys in CHs are unique and hashed, they cannot be used to obtain the corresponding polynomial, that is, all the coefficients of the polynomial, reversely. We conclude that our scheme has full ability to defense the node capture attack. This conclusion applies to LEKM as well because all unrelated keys are removed at CHs after network deployment. On the other hand, the IKDM scheme has the t-security problem because all preloaded t-degree polynomials at each CH will not be removed after network deployment. Once a group of CHs, exceeding t, are captured, all the keys in noncaptured nodes will also be compromised.

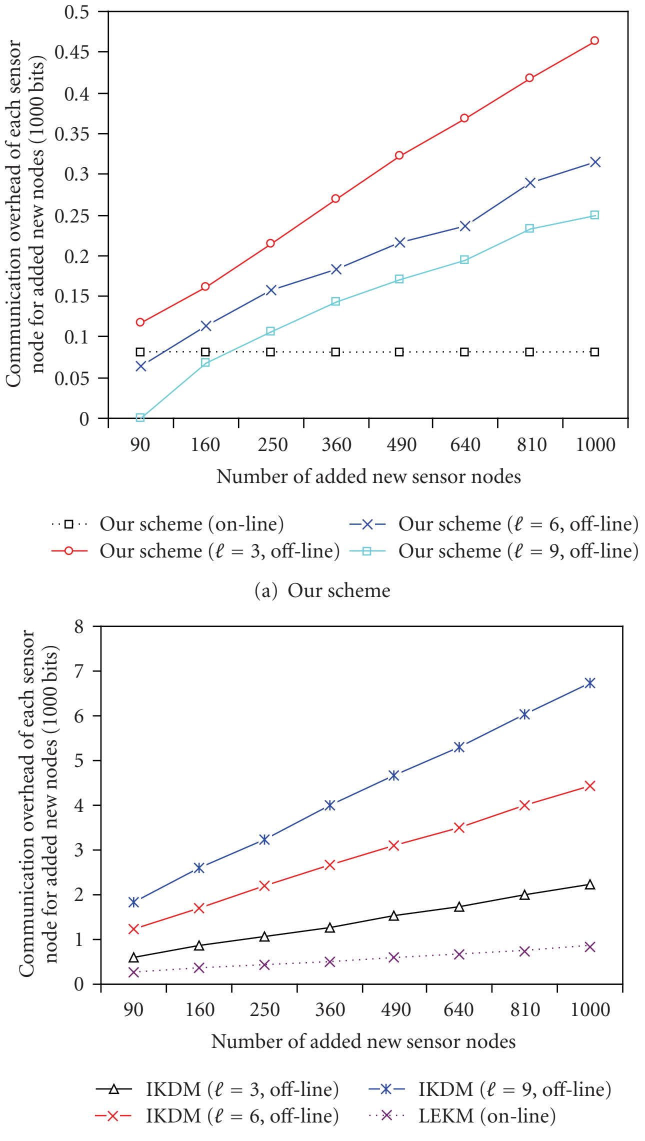
5.2. Performance Evaluation. Now we turn our attention to evaluate the performance of this group of key distribution schemes in hierarchical WSNs. The performance metrics are storage and communication overhead.

To supports a large-scale WSN, a feasible solution of key distribution should be scalable in terms of storage cost. In the scheme LEKM [10], the number of keys stored in each CH is linearly proportional to the number of clusters. The IKDM scheme has fixed storage overhead for sensor nodes and cluster heads. Our scheme has fixed storage cost for sensor nodes. The storage requirement $O(\lambda_S + \lambda_{CH})$ for cluster head is also reasonable because it requires to communicate with at least $\lambda_S + \lambda_{CH}$ number of nodes. The performance comparison in various network sizes is summarized in Table 2.

As shown in Figures 3 and 5 for the cluster head addition processes, the communication overhead of Protocols 2 and 4 is both fixed under the condition that λ_S and λ_{CH} are constant numbers, which is true for a uniform node deployment. This feature shows the scalability of our scheme in terms of message complexity. They are also the first solution for key management in WSNs with renewable cluster heads.

In the following, we conduct a simulation study on the communication overhead for the sensor node addition process. We have implemented a simulation tool using Java for the special purpose of evaluating the performance of this group of protocols while the lower MAC layer is assumed to be ideal.

A hierarchical wireless sensor network was simulated with different sizes of n sensor nodes and m clusters. In order to study the scalability of these protocols, we have considered the scenarios with a specified a cluster size m ($m = 9, 16, 25, 36, 49, 64, 81$, and 100) and a sensor node size n ($n = 100\,\mathrm{m}$). For each example, the whole network



(a) Our scheme

(b) Other schemes

Figure 6: Communication overhead comparison.

is regularly organized as $\sqrt{m} \times \sqrt{m}$ number of clusters, and there are exactly 100 sensor nodes in each $R \times R$ cluster. The transmission range of each cluster head is set as $\sqrt{5}R$, and the communications between CHs may be made in a multihop manner if they are separated far away from each other. To simulate the sensor node addition process, we consider 10 new sensor nodes to be added to each cluster. In each message interaction for all protocols, the length of each Id and key takes up 32 and 80 bits, respectively.

The performance comparison is made in terms of communication overhead. It is evaluated in the number of bits transmitted for key establishment between a sensor node and a cluster head. In all cases, that is, a sensor node size n, a cluster size m, and a specific key distribution scheme, we randomly generated 50 different instances and we present here the average over those 50 instances.

As shown in Figure 6(a), our scheme has the fixed and lowest communication overhead for the on-line scenario. The experimental results also comply with our protocol design for the off-line scenario, in which multiple candidate proxies can improve the performance, that is, the communication overhead is a decreasing function of ℓ under fixed network size. In summary, our scheme in both scenarios can significantly outperform other proposals as shown in Figure 6(b).

6. Conclusion

In this paper, we present an efficient and flexible key distribution scheme based on three-tier renewable wireless sensor networks. Our scheme can defend against node capture attack and support dynamic membership management. To our best knowledge, the solution of the key establishment for new cluster heads under both the BS off-line and on-line cases is proposed by the first time. Furthermore, our scheme is efficient and scalable in terms of communication and storage costs, which is particularly beneficial to support large-scale and resource constrained WSNs.

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] 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.

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

[4] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

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

[6] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS '03)*, pp. 52–61, Washington, DC, USA, October 2003.

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

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

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

[10] G. Jolly, M. C. Kuscu, P. Kokate, and M. Yuonis, "A low-energy management protocol for wireless sensor networks," in *Proceedings of the 8th IEEE International Symposium on Computers and Communication (ISCC '03)*, pp. 335–340, Kemer-Antalya, Turkey, June-July 2003.

[11] W. Zhang, H. Song, S. Zhu, and G. Cao, "Least privilege and privilege deprivation: towards tolerating mobile sink compromises in wireless sensor networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '05)*, pp. 378–389, Urbana-Champaign, Ill, USA, May 2005.

[12] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," Lecture Notes in Computer Science, pp. 471–486, 1993.