Hindawi Publishing Corporation
Mobile Information Systems
Volume 2015, Article ID 627548, 10 pages
http://dx.doi.org/10.1155/2015/627548

Hindawi

Research Article

Enhanced Key Management Protocols for Wireless Sensor Networks

Baojiang Cui,[1] Ziyue Wang,[1] Bing Zhao,[2] Xiaobing Liang,[2] and Yuemin Ding[3]

[1]*School of Computer, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*State Grid Metering Center, Beijing 100192, China*
[3]*Department of Electronic Systems Engineering, Hanyang University, Ansan 426791, Republic of Korea*

Correspondence should be addressed to Baojiang Cui; cuibj@bupt.edu.cn

Received 29 August 2014; Accepted 1 September 2014

Academic Editor: David Taniar

With rapid development and extensive use of wireless sensor networks (WSNs), it is urgent to enhance the security for WSNs, in which key management is an effective way to protect WSNs from various attacks. However, different types of messages exchanged in WSNs typically have different security requirements which cannot be satisfied by a single keying mechanism. In this study, a basic key management protocol is described for WSNs based on four kinds of keys, which can be derived from an initial master key, and an enhanced protocol is proposed based on Diffie-Hellman algorithm. The proposed scheme restricts the adverse security impact of a captured node to the rest of WSNs and meets the requirement of energy efficiency by supporting in-network processing. The master key protection, key revocation mechanism, and the authentication mechanism based on one-way hash function are, respectively, discussed. Finally, the performance of the proposed scheme is analyzed from the aspects of computational efficiency, storage requirement and communication cost, and its antiattack capability in protecting WSNs is discussed under various attack models. In this paper, promising research directions are also discussed.

1. Introduction

Wireless sensor networks (WSNs) have been extensively used in various applications, such as homeland security, battlefield surveillance, environmental monitoring, and health care. Through collection and processing of the sensing data from the coverage area, WSNs enable users to access detailed and reliable information at any time and any place, which is a ubiquitous sensing technology.

WSNs have two salient characteristics: (i) it uses wireless communication and anyone within the range of the network can attack it; (ii) it may be deployed in unattended environments or even hostile regions, such as battlefield, where it can be physically attacked or captured [1]. Thus, how to ensure the security of WSNs becomes a significant issue.

Security researches of WSNs mainly focus on key distribution, secure routing protocols, secure transmission, and security defense. In these scopes, using key management mechanisms to settle security issues under the wireless sensor network environment is the most crucial and challenging problem [2].

Although key management mechanisms in the cable network have been deeply studied, the research is still immature in WSNs [3] because of limited communication bandwidth, computing and storage capacity of sensor nodes, and unfixed infrastructures. There is also a contradiction between the maximum security performance and minimum resource consumption.

It is worth noting that, due to the resource limitations, asymmetric encryption algorithms are seldom applied to the sensor network and most of the related works are based on symmetric key systems.

Although a number of classic protocols and schemes have been proposed for WSNs, many protocols concentrated on communication and processing technologies without paying enough attention to security issues, such as TEEN [4] and LEACH [5].

In recent years, scholars have proposed more sophisticated protocols which are mainly divided into two categories: predistribution scheme based on symmetric key and key management scheme based on public key.

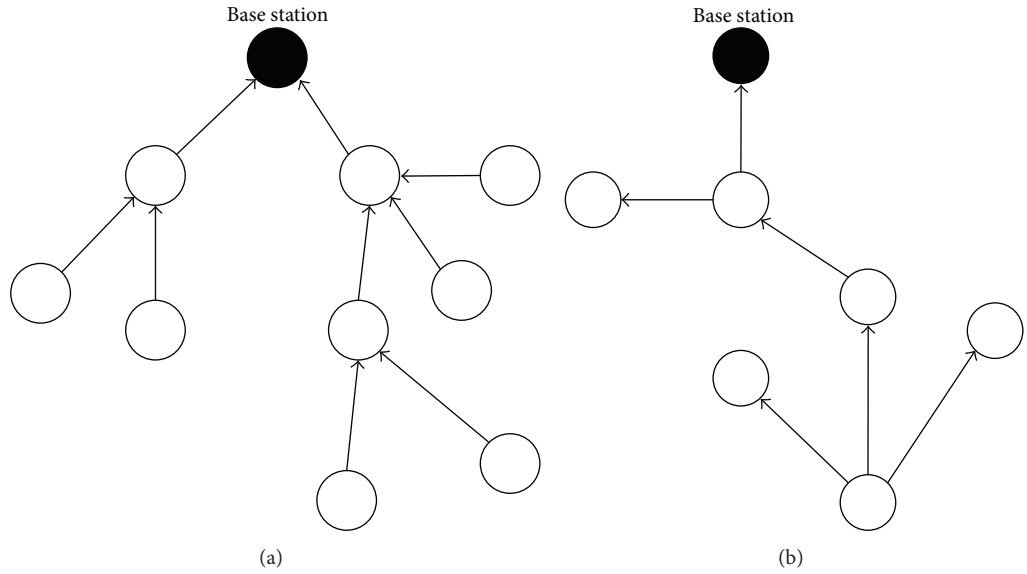

Figure 1: Examples of in-network processing.

Among the predistribution schemes, SPINS [6] is recognized as a classical secure protocol for WSNs. It consists of two modules: SNEP for data confidentiality, two-party data authentication and data freshness, and μTESLA for authenticated broadcast. It provides security for the entire network based on a single key and is easy to implement but the expansibility is limited.

To balance the security performance and resource consumption, random key predistribution schemes, polynomial key predistribution schemes, and key predistribution scheme based on deployment knowledge are subsequently proposed.

E&G [7] scheme is one of the earliest random key predistribution schemes. It achieves the establishment of pairwise key in WSNs for the first time based on the idea of preallocated key generation, solves the problem of unpredictable network topology, and provides a probability-based security. After that, the proposed Q-composite scheme [8] improves E&G schemes based on multicommon keys to generate pairwise keys.

Though quite a lot of superior security protocols have been proposed recently, most of them have their own deficiencies. Park proposed a lightweight security protocol (LISP); it can tolerate packet loss but the protocol cannot handle node revocation problem. After that, SRDA [9] proposed a secure data aggregation protocol, which takes the integrity into consideration but ignores the confidentiality of the information. LDP [10] proposes a local key management protocol based on dynamic cluster. It effectively supports the WSN security data fusion but does not give an effective solution of revoking captured nodes and updating keys.

To avoid above deficiencies, LEAP [11] establishes four kinds of keys and provides a strong application and scalability but requires huge amount of communication for key establishment and update. Furthermore, its security is heavily dependent on the initial secure time. ChengY's predistribution scheme [12] is based on clusters with advantages of the good connectivity, network survivability, and low communications costs. However, the cost for rekeying is significant.

Based on previous studies, this paper proposes improved strategies to overcome some defects. In addition, how to apply the established keys to form security mechanisms to confront kinds of attacks is described in detail.

2. Requirements of Sensor Networks

Many security requirements of WSNs are similar to those of traditional networks, such as data confidentiality, authentication, and integrity. What is more, it should guarantee low energy consumption and high efficiency [13].

It is proved in recent researches that in-network data processing (shown in Figure 1), which mainly includes passive participation and data aggregation, is quite energy-efficient and should be widely employed.

The typical application of in-network processing is to divide the network into multiple clusters where the cluster head node collects and aggregates information from its neighbors and delivers the summary directly to the base station to avoid redundant transmissions and save communication bandwidth.

Generally, the pairwise key performs better over achieving data confidentiality, authentication, and integrity of WSNs, whereas, the cluster key or network-wide key is needed to achieve in-network data processing (shown in Figure 1) [14].

The particularity of the WSNs requires the ability of resistance to physical attacks and trapping. For example, once a node is compromised, the loss of secret information does not threaten remaining security links. Moreover, well-designed security mechanism should have capabilities of key revocation and update.

Therefore, it is fundamental to design a security mechanism, which satisfies above requirements, in order to achieve the security of WSNs.

3. Prerequisite Knowledge

3.1. Notations. The notations used in this paper are given in Notations section.

Note that, in order to simplify the representation in the following discussion, notations A and B are used to represent their node identifiers instead of ID_A and ID_B.

In addition, since keys for various security uses can be derived from the same key k, such as $K_0 = f(K, 0)$ for authentication and $K_1 = f(K, 1)$ for encryption, we just say a message M is authenticated or encrypted with K instead of saying in detail.

3.2. Function and Algorithm Description

One-Way Hash Function. One-way hash function H meets the following properties [15].

(i) Given x, it is easy to compute y using function $y = H(x)$.

(ii) Given y, it is difficult to compute x from function $y = H(x)$.

(iii) Given x, it is difficult to find a y meeting the condition that $y \neq x$ and $H(y) = H(x)$.

One-way hash chain is a sequence of the following hash value $\{x_m, x(m-1), \ldots, x_j, \ldots, x_1\}$, fulfilling the restriction $\{x_j \mid 0 < j \leq m, x_{j-1} = H(x_j)\}$, where x_m is a random selection of key seed. Due to the unidirectional feature, one-way hash key chain is widely used in secure authentication. For example, when x_1 is given, it can be verified that whether x_i is an element of the one-way hash key chain sequence using the equation $x_1 = H^{i-1}(x_i)$.

Key Generation Function. Pseudorandom function f is employed as the key generation function here for its high computational efficiency. When it is used in key establishment process, the computational cost is negligible. Note that, this function is stored in all the network nodes as well as the base station.

Diffie-Hellman Algorithm. Diffie-Hellman provides a method to ensure safety of shared key through insecure networks and it is an integral part of OAKLEY algorithm.

The ingenious point is that two sides of communication can use this method to determine the symmetric key, which can be used for encryption and decryption. Note that the key exchange protocol can only be used for key exchange, without being able to encrypt and decrypt the messages [16].

Since the key exchange algorithm itself is usually limited to be used as key exchange technology for many commercial products, it is usually called Diffie-Hellman key exchange (abbreviated as DH algorithm, key exchange based on DH algorithm is also commonly referred to as *DH exchange*).

The purpose of this key exchange technique is to enable two users to achieve secure key exchange in order to ensure the encryption of subsequent packets. The effectiveness of Diffie-Hellman key exchange algorithm relies on the difficulty of computing discrete logarithms [17]. In short, the discrete logarithm can be defined as follows.

First define primitive root of prime number p, which is integer roots generated from each of its powers from 1 to $p-1$; that is, if a is a primitive root of prime number p, the values of $a \bmod p$, $a^2 \bmod p, \ldots, a^{p-1} \bmod p$ are all different integers from 1 to $p-1$ in a certain arrangement.

For an integer b and a primitive root a of prime number p, we can find the unique index i, making $b = a^i \bmod p$, where $0 \leq i \leq (p-1)$, index i is called discrete logarithm or exponent of modulus p which is based to cardinal number a of integer b.

Based on the definition and nature of the primitive root, Diffie-Hellman key exchange algorithm is described as follows [18].

(1) There are two global parameters: prime number p and integer a, where a is a primitive root of p.

(2) Suppose users A and B wish to exchange a key, user A selects a random number X_A ($X_A < p$) as private key and calculates the public key $Y_A = a^{X_A} \bmod p$. The confidentiality store of X_A by user A makes Y_A publicly available to user B. Similarly, user B also selects a random number X_B ($X_B < p$) as private key and calculates the public key $Y_B = a^{X_B} \bmod p$. The confidentiality store of X_B by user B makes Y_B publicly available to user A.

(3) User A calculates shared secret key by $K = (Y_B)^{X_A} \bmod p$, and user B similarly calculates shared secret key K by $K = (Y_A)^{X_B} \bmod p$.

Since

$$\begin{aligned} K &= (Y_B)^{X_A} \bmod p = \left(a^{X_B} \bmod p\right)^{X_A} \bmod p \\ &= a^{X_B X_A} \bmod p = \left(a^{X_A}\right)^{X_B} \bmod p \\ &= \left(a^{X_A} \bmod p\right)^{X_B} \bmod p = (Y_A)^{X_B} \bmod p. \end{aligned} \tag{1}$$

Thus, it corresponds that two sides have exchanged the same secret key K. Because X_A and X_B are confidential, an adversary can only use parameters q, a, Y_A and Y_B. Thus, adversary is forced to use discrete logarithm to determine the shared key K. The security of Diffie-Hellman key exchange algorithm relies on the fact that although computing exponent, which takes prime number as module, is relatively easy, computing discrete logarithm is very difficult. For large prime numbers, calculating the discrete logarithm is almost impossible.

3.3. Assumptions. Basic assumptions are as follows.

(i) Topology is unknown before the deployment of the nodes.

(ii) The sensor network is static (sensor nodes are not mobile) after deployment.

(iii) Sensor nodes have similar computational and communication capabilities.

(iv) Transmission power of nodes can be adjusted to control the propagation distance.

(v) The base station has enough energy supply and computing power.

(vi) The attacker has the ability to eavesdrop on all the channels as well as to replay former messages and inject malicious packets.

(vii) Once a node is captured, all the stored information will be obtained by the adversary.

(viii) Every node has enough space to store hundreds of bytes for key establishment materials.

(ix) Each node has some degree of ability to resist attack and it will not be captured with in a limited period of time.

4. Protocol Description

This section introduces the basic protocol in detail, including four kinds of secure key establishment mechanisms to satisfy various secure communication requirements and mechanisms for key erasure and update.

4.1. Overview. As discussed above, the single key mechanism cannot provide appropriate protection to all the required communication in the WSNs. Moreover, the security performance and resource consumption have to be balanced when making use of different kinds of keys.

The degree of sharing keys in the security mechanism has to be taken into consideration. For example, if unique pairwise keys are used for each two nodes in the WSNs to guarantee secure communication, the node captured by an attacker will not reveal any security information of other normal nodes, which is ideal to prevent threat to the entire network. However, it requires significant communication bandwidth and energy resources, which is quite inefficient.

On the contrary, if only a network-wide key is used for authentication and encryption, no communication between nodes is required for establishment of additional keys, and the storage costs and energy consumption can also be minimized. However, the security will be extremely poor. Once any node in the system is captured by an attacker, the whole network suffers an enormous risk.

4.2. Key Establishment. In this section, the establishment of four kinds of keys is discussed in detail as well as their characteristics and abilities to resist attacks.

4.2.1. Individual Key Establishment. Individual key is a unique key of each sensor node that shared with the controller (the base station) which is used for individual authentication and secure communication assurance [19].

For example, individual key can be used to encrypt sensitive information, such as special instructions and rekeying commands, exchanged between a sensor node and the base station. It can also be used for message authentication to get verification of the base station or other nodes.

Since every node in the network shares a unique individual key with the base station, it is neither practical nor efficient to store all these keys for the base station especially when the network scalability is very huge. Thus, it is important to adopt a strategy to reduce the storage overhead, which can be achieved by the key generation function f.

First of all, it is argued that each node holds the key establishment function f and an initial key K_I which is derived from the master key K that is only possessed by the controller; all of them are preloaded in the nodes before the key establishment phase. The generation of individual key for node A (here A indicates the unique ID of node A) is as follows:

$$K_A = f(K_I, A). \tag{2}$$

In the above, the function f for key establishment is a pseudorandom function and it is efficient enough to be used on sensor nodes.

Once the individual key is generated, the related node stores it within its life cycle. Since the base station has full knowledge of the initial key K_I and efficient establishment function f, the storage overhead for individual keys of each sensor node can be reduced.

4.2.2. Pairwise Key Establishment. Pairwise keys of a node indicate the keys shared with each of its direct neighbors, so the storage overhead of such keys for each node depends on the number of its neighbors [20, 21].

In this protocol, pairwise keys have a lot of uses. For example, it can be used for a cluster head to encrypt the cluster key, which has to be transmitted to all of its neighbors, to achieve the distribution security. It is also a component to improve system security.

However, it will impede passive participation, which is important in saving communication energy, if such key mechanism is employed individually. The initial pairwise key establishing progress is shown in the Figure 2.

The generation of pairwise keys for nodes A and B (here A is assumed to be the node that call for key establishment) is as follows:

$$\begin{aligned} A &\longrightarrow * : \mathrm{Nonce}_A \\ B &\longrightarrow A : B, \mathrm{MAC}_{K_B}(\mathrm{Nonce}_A \mid B). \end{aligned} \tag{3}$$

Here, node A broadcasts a nonce to all of its direct neighbors to request establishing pairwise without authenticating its identity, because if it cannot provide its own identity (namely, it does not own the individual key), it will fail to generate the pairwise in the following steps:

$$K_{AB} = f(K_B, A). \tag{4}$$

Since node A possesses both the key establishment function f and the initial key K_I, it can compute K_B independently and then obtains the pairwise key K_{AB} as well.

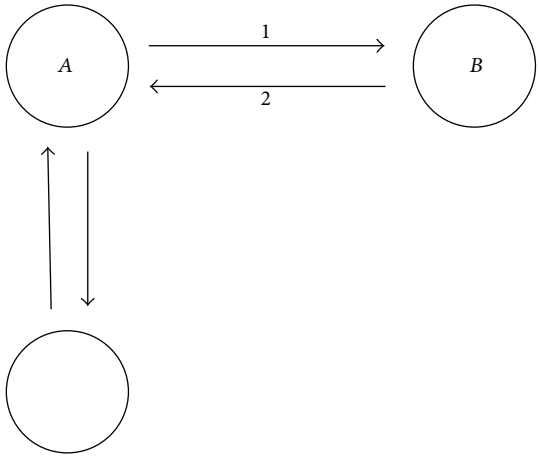

Figure 2: Pairwise key establishing phase.

Note that, each node has a timer which conducts it to achieve key erasure when it makes sure that the pairwise keys establishment is finished. This process is significant because all the nodes keep the network-wide initial key K_I to help complete the establishments in the initial period, and once the relatively safe period passes by, it will face great risk that some nodes may be compromised.

So it is suggested that, after a reasonable length of time, the initial key K_I and the neighbors individual master keys stored in the node be all erased (but its own individual master key will always be held).

In this way, when almost the pairwise keys are established successfully, no nodes will possess the necessary generating key materials until there is a new group of nodes to be joined. The key erasure mechanism is so necessary that how to control the key erasing time is worth exploring, but it is not an emphasis in this paper.

In addition, it can also be seen from the above equation that after the establishing time, namely, related key materials are erased, once the node A is compromised by an attacker and a A' broadcasts a nonce for establishing pairwise keys, it cannot success due to such establishment mechanism.

But once the attacker uses A' to take passive joining strategy, the responding node A' will generate the pairwise key with B (here B is one of a new batch of joining nodes that is asking to establish pairwise key with its neighbors including A') as follows: $K_{BA'} = f(K_{(A')}, B)$ and then the attacker will be able to inject erroneous packets into the network at will.

For the new added nodes, an alternative is proposed to establish secure pairwise key:

$$K_{AB} = f\left(K_B, A\right) \oplus f\left(K_A, B\right). \tag{5}$$

Since the pseudorandom function f is efficient, such improvement could be accepted.

The advantage of above key establishing scheme is that there is no message exchanging between nodes A and B during the computing step which extremely saves communication overhead.

Note that there will be a situation that two nodes want to establish the pairwise key while one of them does not possess the master key K_I, such as one new added node and an older node which has finished all its pairwise key establishments and erased the master key K_I.

To deal with such situation, a scheme that asks for help from controller is simply presented as follows:

$$\begin{gathered} A \longrightarrow B : \text{Nonce}_A, A \\ B \longrightarrow \text{Base station} : R_{K_{AB}}, A, B, \text{MAC}_{K_B}\left(R_{AB}, A, B\right) \\ \text{Base station} \longrightarrow A : E_{K_A}\left(K_{AB}\right), \text{MAC}_{K_A}\left(B, E_{K_A}\left(K_{AB}\right)\right) \\ \text{Base station} \longrightarrow B : E_{K_B}\left(K_{AB}\right), \text{MAC}_{K_B}\left(A, E_{K_B}\left(K_{AB}\right)\right). \end{gathered} \tag{6}$$

Here A is a new node who calls for establishing pairwise key with its neighbor B. Here B is an older node that has generated all its own pairwise keys and erased the initial key K_I, which makes it unable to generate new pairwise key.

If B wants to verify the identity of node A, the most credible way is asking for help of base station.

However, reducing the use of base station is an important goal here and the improvement is worth further exploring.

4.2.3. Cluster Key Establishment. Cluster key is a key generated by an elected cluster head and shared with its neighbors and it is mainly used for encrypting local broadcast packets. Its most significant advantage is that it enables the in-network processing such as passive participation and data aggregation, which cannot be supported by the pairwise key but could save energy consumption efficiently.

This key establishing process is obvious as follows:

$$A \longrightarrow B_i : E_{K_{AB_i}}\left(K_A^C\right). \tag{7}$$

Here node A is the elected cluster head and B_i represents one of its immediate neighbors: $B_1, B_2, \ldots, B_n$ $(1 \leq i \leq n)$. Cluster head A first generates a key K_A^C randomly and encrypts it with its pairwise keys and then sends it to each neighbor B_i. Moreover, node B_i decrypts the cluster key and then stores K_A^C in a table.

When any neighbor of A is revoked which means there will be a risk to continue using the old cluster key, cluster head A regenerates and transmits the $K_A^{C'}$ in the same way.

Cluster division and cluster head selection approaches are also worthy of discussion. But it is not an emphasis in this paper. A simple mesh division method is shown in Figure 3 based on virtual cluster idea.

4.2.4. Group Key Establishment. The group key K_g is used for encrypting messages that need to be broadcasted to the whole group. Note that, different from above situations, the key point here is no longer about key establishment or encrypting schemes because there is only one group key shared among the entire network; meanwhile it does not make sense to encrypt a broadcast message using master key of each sensor node separately.

It is also because there is only one group key shared among sensor nodes; once a compromised node is revoked,

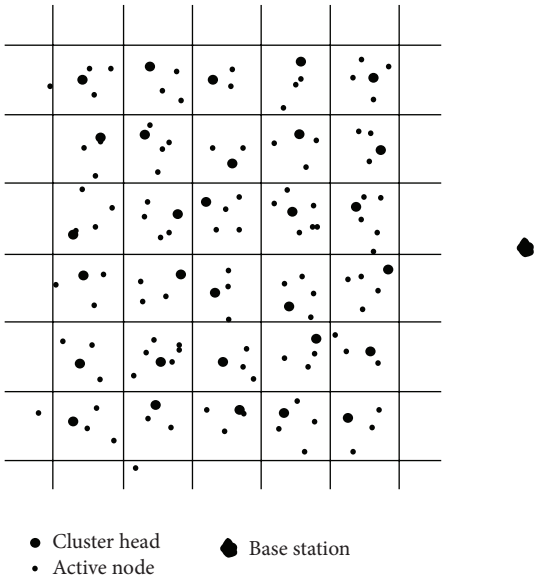

Figure 3: Mesh division method.

the rekeying and updating mechanism comes to be important.

μTESLA [22] is a widely employed protocol due to the high efficiency and perfect tolerance for packet loss. A one-way hash function H is used here to help achieve the process. Firstly, the controller generates a random seed k_m and uses the function H to get a sequence of the following hash values: $\{k_m, k_{m-1}, \ldots, k_j, \ldots, k_1\}$ that meets the restriction $\{k_j \mid 0 < j \le m, k_{j-1} = H(k_j)\}$.

Then preload this key chain $\{k_m, k_{m-1}, \ldots, k_j, \ldots, k_1\}$ in the base station and use delayed key disclosure to achieve message authentication. Let A be the revoked node and K'_g the new group key; the process is as follows:

$$\text{Base station} \longrightarrow * : A, f\left(K'_g, 0\right), \text{MAC}_{k_j}\left(A \mid f\left(K'_g, 0\right)\right). \tag{8}$$

When the verification is done, all the nodes will remove related information of node A and restore the group key K'_g in the table.

Note that the initial Group key K_g is preloaded in all the sensor nodes before their deployment like the initial key K_I, but we cannot take K_I also as the group key because it will be erased in a very short time after the pairwise key establishment. The key used for deriving related keys must be protected separately from normal ones.

Figure 4 simply illustrates the authentication mechanism:

$$k_{j-1} = H\left(k_j\right). \tag{9}$$

5. Enhanced Protocol

5.1. Requirements Analysis. The design of the basic scheme presented in the previous section is motivated by the observation that single keying mechanism is not suitable for meeting

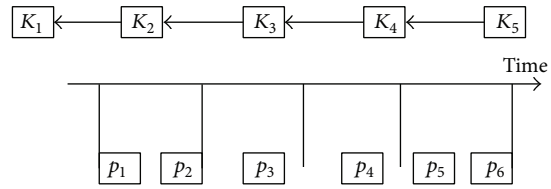

Figure 4: Using the one-way hash function for source authentication.

all the security requirements of different types of exchanged messages.

The advantage of this scheme is that the captured node does not threat the safety of the other nodes in case the master key K is absolutely safe in time interval $T_{\min}$.

During the time interval $T_{\min}$, all the nodes of the WSN will hold the general master key K and we note that this scheme cannot provide confidentiality when a node is compromised in $T_{\min}$. Because, by using the stolen information like the master key K, an attacker can easily derive the master keys of all the rest normal nodes that are deployed in the same time interval as well as negotiating new pairwise key with normal nodes in any region, which means once a node is compromised in time interval $T_{\min}$, the security of the entire network is extremely dangerous.

5.2. Enhanced Scheme. Based on the Diffie-Hellman algorithm above, presenting the improved scheme, prior to deployment of the network, each node prestores the large prime number p and its primitive root a instead of the initial key K_I which is derived from the master key K.

Note that the generation of individual key for node A is still same:

$$K_A = f\left(K_I, A\right). \tag{10}$$

Different from the basic scheme, this process is completed once the node is deployed, after that the information of the initial key K_I is deleted. Thus, the attacker cannot get any information about the initial key K_I or the master key K even if it is compromised during the working period.

Since the node no longer keeps initial key K_I, which is required to participate in relevant calculations (function) in the pairwise key generating process, the basic scheme cannot be achieved. For this situation, make the following improvements.

Gain a key evolution function to each node. Takes node A and B for examples:

$$\begin{aligned} X_A &= h\left(A \mid K_A\right) \bmod p \\ X_B &= h\left(B \mid K_B\right) \bmod p. \end{aligned} \tag{11}$$

Then calculate the public message:

$$\begin{aligned} Y_A &= a^{X_A} \bmod p \\ Y_B &= a^{X_B} \bmod p. \end{aligned} \tag{12}$$

The pairwise key generation process is as follows:

$$A \longrightarrow * : \text{Nonce}_A, Y_A$$
$$B \longrightarrow A : \text{MAC}_{K_{AB}}(B \mid Y_B), B, Y_B. \quad (13)$$

Here, node A broadcasts a nonce to all its direct neighbors and asks to establish pairwise key and broadcasts the public message Y_A at the same time. When its neighbor (take node B, for example) receives the message, it first verifies the legitimacy of Y_A and then calculates the pairwise key using the following function:

$$K_{AB} = (Y_A)^{X_B} \bmod p. \quad (14)$$

After that, node B sends messages B and Y_B back to the asking node A and sends a message $\text{MAC}_{K_{AB}}(B \mid Y_B)$ to authenticate its identity. If node B cannot respond to node A in this way, it means node B cannot get K_{AB} only taking use of Y_A; then consider node B as untrusted. In addition, node A does not need to send authenticating message back to node B anymore because if it cannot prove its own identity (namely, it cannot get K_{AB} only taking use of Y_B, and it will fail to generate the pairwise key K_{AB}).

Compared with the basic protocol, the most obvious improvement of enhanced protocol is that it takes use of Diffie-Hellman algorithm to generate pairwise keys instead of storing the initial key K_I in a certain period of time. Thus, even if a node is compromised in $T_{\min}$, the attacker can merely get the information of key related to the compromised node, which means only limited security threats can be caused, avoiding the disruption of the entire network caused by losing initial key K_I. Despite the slight increment in the computational overhead, the security of the WSN is greatly improved.

6. Performance Evaluation

The ability of the protocol to fight against kinds of attacks is discussed in detail in above sections. This section analyzes the storage requirement and energy efficiency.

6.1. Storage Requirement. In the basic protocol, a node needs to store four types of keys. Considering a node with m neighbors in the WSN, it needs to store one individual key, m cluster keys, m pairwise keys, and one group key. In the enhanced protocol, each node stores the same number of keys as the basic protocol.

When the key establishment is complete in a network having a scale of N, there is an upper limit of the number of keys to be stored in the nodes including N individual keys, $C(N, 2)$ pairwise keys, $N/2$ cluster keys, and N group keys (though there is only one group key in a certain period), which add up to $((5/2)N+(N!/2(N-2)!) = (N^2+3N)/2)$ and average to each node is $(5/2 + (N-1)!/2(N-2)! = N/2+2)$.

Note that communication distance of sensor node is limited so that it will not reach a high complexity that each two nodes are connected.

In addition, using an efficient clustering method can reduce the number of required cluster keys and the real storage complexity is much smaller.

Although memory is a quite scarce resource for the current generation of nodes in WSNs, for a reasonable degree, storage is not an issue in our protocol. For example, 100 keys totally take 800 bytes when the key size is 8 bytes.

6.2. Communication Cost. In this paper, the average communication cost increases with the connection degree of a sensor network and decreases with the network size N. Efficient preloaded functions are widely used, which greatly reduces the message exchanges in key establishing phase so that to save communication cost. Whats more, the use of located cluster key enables in-network data processing which also helps achieve communication and energy efficiency.

It is worth noting that the communication cost of the enhanced protocol remains at the same level as that of the basic protocol.

6.3. Computational Cost. Functions used in the proposed protocols are all of high computational efficiency. For example, pseudorandom function f is employed to be the key generation function, and the computational cost will be negligible when it is used in key establishment process. In the enhanced protocol, although computational cost is slightly increased by using Diffie-Hellman algorithm, for a network of reasonable density, we believe that the computational overhead is applicable for a network of reasonable density in our protocols. For example, for a WSN of size $N = 1000$ and connection degree of 20, the average computational cost is 2.7 symmetric key operations per node per revocation and a larger N will reduce the cost further.

Overall, we conclude that the protocols proposed in this study are scalable and efficient enough in storage, communication, and computation.

7. Security Analysis

This section analyzes the security of the key management protocols. The survivability of the network is discussed when undetected compromised nodes occur and the robustness of proposed schemes is studied in defending against various attacks.

7.1. Survivability. Once a sensor node A is compromised, the adversary can launch attacks by utilizing keying materials of node A. If the threat is detected somehow, the protocols can revoke node A efficiently and update the information of nodes quickly throughout the whole network. Basically, each neighbor of compromised node A could delete its pairwise key shared with node A as well as updating the cluster key. The group key could also be updated efficiently by taking use of μTESLA mechanism. When the revocation is completed, the adversary cannot launch further attacks anymore.

However, security detection in WSNs is more difficult than in other systems since sensor systems are often deployed in unattended environments. Thus, the survivability of

the network is one of most important security requirements when compromised nodes is not detected.

Firstly, because individual key is only shared between the base station and each sensor node, it usually does not help the attacker launch attacks.

Secondly, obtaining the cluster keys and pairwise keys of a compromised node enables the attacker to establish trust with the neighbor nodes, which can be used by the attacker to inject malicious sensor readings and routing control information into the network. However, in the proposed protocols in this study, the attacker usually has to achieve such attacks by taking use of the identity of the captured node.

Note that a salient feature of the proposed protocols is the ability in localizing possible threats. Because after the deployment of the network and the pairwise key establishing phase, every node will keep a list of trusted neighbor nodes. As compromised node and its copy nodes cannot establish trust relationship with other nodes except its neighbors, the attacker can only damage secure links within limited range.

Finally, obtaining the group key enables the attacker to decrypt messages broadcast by the base station. The broadcast messages, by their nature, are intended to be received by all the nodes in the network. Thus, compromising any single node is enough to possess this message, whatever security mechanism is used. However, obtaining the group key does not allow the attacker to damage the entire network with malicious packets by impersonating the base station because all messages sent from the base station are authenticated by μTESLA mechanism.

7.2. Dealing with the Attacks on Secure Routing. Ciou et al. have described various possible attacks of routing protocols for WSNs [18]. How the proposed schemes can defend against such attacks is shown in this section.

An inside attacker may attempt to alter and replay routing information to make routing loops, attract or repel network traffic, and generate false messages. Moreover, the attacker can launch the selective forwarding attack, in which the captured node suppresses routing packets sent from a few selected nodes while forwarding the other packets reliably.

In this paper, the schemes cannot protect the WSNs from such attacks; however, the schemes can hinder or minimize the consequences caused by such attacks.

First, based on the key establishment and authentication phases of the proposed protocols, it is apparent that such attacks are only possible within a small area of two-hops from the captured node.

Second, since such attacks are localized in a certain zone, the attacker faces a high risk of being detected when launching such attacks. For example, the probabilistic challenge mechanism can help detect the spoofing attack and the detection of altering attack is also possible since the related sending node may overhear the forwarded messages altered by the captured node.

Last but not least, once a compromised node is detected, the group rekeying process of the protocols can efficiently revoke the compromised node from the network.

The proposed protocol can protect WSNs from the following attacks.

Sybil Attacks. In Sybil attacks, the attacker may replicate the captured node and deploy multiple replicas into the original network. With help of the base station, such replica nodes will then try to establish pairwise and cluster keys with normal nodes that are not neighbors of the captured node [23]. If the base station does not know the precise topology of the wireless network, this attack may work in pairwise key establishment. However, it cannot happen for proposed protocols because each normal node keeps a list of its approved neighbors and the base station is not involved for pairwise or cluster key establishments in this study.

HELLO Flood Attack. The attacker may send a HELLO message to all nodes in the network by increasing the transmission power to be high enough to make all the nodes convinced that it is their neighbor. Once this attack succeeds, nodes of the entire network may send their readings and some other packets in vain. However, it cannot succeed in proposed protocols because the attacked does not have a network-wide key for authentication.

It is worth noting that the group key in the protocols is not for authentication purpose but for the distribution of secure messages to the entire network from the base station.

7.3. Defending against Sinkhole and Wormhole Attacks. The combination of the sinkhole and the wormhole attacks is one of the most difficult attacks to be prevented.

In the sinkhole attack, a malicious node tries to attract packets from the neighbor nodes and then drops them. It can launch such attack by advertising information of high reliability or high remaining energy, which is very hard to detect in the WSNs.

In the wormhole attack, two distant malicious nodes conceal their distance information to the network. After placing one such node near the target zone and another one near the base station, the attacker will convince the nodes within the target area, which are usually multiple hops away from the base station, as only one or two hops to create a sinkhole. Moreover, nodes which are multiple hops away may believe that they are neighbors of each other. Since to launch wormhole attack the attacker does not need to compromise any sensor nodes, such attack is very powerful in practice [24].

In the proposed protocols, an outside attacker cannot succeed in launching wormhole attack except in the neighbor discovery process, since a node will know all its neighbor nodes after the pairwise key is established, which means the attacker cannot convince two distant nodes to believe that they are neighbors of each other.

Because the time of neighbor discovery process is very short (usually for seconds), the probability that the attacker achieves such attacks is also quite small. If an inside attacker compromises two or more nodes, it can launch such attacks. However, it cannot convince two distant nodes as neighbors when the neighbor discovery phase is finished. The authenticated neighborhood information is critical to deal with the wormhole attacks.

In the sinkhole attack, if the attacker compromises a node A that is close to the base station and another node B in

the target area, the attacker will succeed in making node A as a sinkhole. Since the number of hops between node B and the base station turns smaller, node B will be especially attractive to surrounding nodes. In practice, the location of base station is usually static. When the network is constructed, topology will be known to the entire network, and then sensor nodes will know the approximate number of hops from the base station. Thus it is difficult for an attacker to make a very attractive sinkhole in the WSN without being detected.

7.4. Conclusion. This paper proposes a basic key management protocol based on initial secure time, which assumes that the attacker cannot compromise a node in a short time. It satisfies various security requirements of WSNs using the combination of four kinds of secure keys. Meanwhile, the erasure and update mechanism of keys is important to support network security.

To further improve the security of the basic scheme, an enhanced protocol based on Diffie-Hellman algorithm is proposed, which avoids storing the master key in sensor nodes so as to restrict the security impact of a captured node to the rest network.

The proposed protocol achieves high communication and energy efficiency by supporting in-network data processing and enhances the network security through strict authentication and encryption mechanisms. Compared to original ideas, the proposed scheme improves not only the network security but also the extensibility of WSNs.

This paper presents a proposal for key establishment and achieves security mainly based on the combining application of four kinds of keys. This is a critical step and how to use such keys to found a protection mechanism is a focus in our future research.

Notations

N :	The number of nodes in the network
A, B:	Two communicating nodes in the network (also represents the node identifier)
$f(K, A)$:	Calculate with parameter A using the key K in pseudorandom function f
$H(K)$:	One-way hash function to generate a chain of keys using the seed K
$\mathrm{MAC}_K(m)$:	Message authentication code (MAC) of message m using MAC key K
K:	The master key only possessed by base station
K_A:	Individual key of node A
$E_K(m)$:	Encryption of message m with a symmetric key K
$M_1 \mid M_2$:	Concatenation of the sequences M_1 and M_2
$A \rightarrow B : M$:	Node A sends a message M to node B
$A \rightarrow * : M$:	Node A sends a local broadcast message M to all its neighbors
$h(m)$:	Calculate hash value of message m.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by National ratural Science Foundation of China (nos. 61170268, 61100047, and 61272493), International S&T Cooperation Special Projects of China (no. 2013DFG72850), and The National Basic Research Program of China (973 Program) (no. 2012CB724400).

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] X. He, M. Niedermeier, and H. de Meer, "Dynamic key management in wireless sensor networks: a survey," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 611–622, 2013.

[3] R. Riaz, A. Naureen, A. Akram, A. H. Akbar, K. H. Kim, and H. Farooq Ahmed, "A unified security framework with three key management schemes for wireless sensor networks," *Computer Communications*, vol. 31, no. 18, pp. 4269–4280, 2008.

[4] C. Intanaonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '00)*, pp. 56–67, ACM/IEEE, Boston, Mass, USA, August 2000.

[5] A. Manjeshwar and D. P. Agrawal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks," in *Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS '01)*, pp. 2009–2015, IEEE Computer Society, San Francisco, Calif, USA, April 2001.

[6] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: security protocols for sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (Mobicom '01)*, pp. 189–199, Rome, Italy, July 2001.

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

[8] 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, Oakland, Calif, USA, May 2003.

[9] H. O. Sanli, S. Ozdemir, and H. Cam, "SRDA: secure reference-based data aggregation protocol for wireless sensor networks," in *Proceedings of the IEEE 60th Vehicular Technology Conference (VTC '04)*, pp. 406–410, IEEE, Los Angeles, Calif, USA, 2004.

[10] T. Dimitriou and I. Krontiris, "A localized, distributed protocol for secure information exchange in sensor networks," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05)*, pp. 37–45, IEEE, April 2005.

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

[12] J. Shen and L. Xu, "Cluster-based key pre-distribution seheme for wireless sensor networks," *Journal of Wuhan University: Natural Science Edition*, vol. 55, no. 1, pp. 117–120, 2009 (Chinese).

[13] X. Huang, M. Yang, and S.-S. Lv, "Secure and efficient key management protocol for wireless sensor network and simulation," *Journal of System Simulation*, vol. 20, no. 7, pp. 1898–1903, 2008.

[14] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New algorithms for secure outsourcing of modular exponentiations," in *Computer Security—ESORICS 2012: 17th European Symposium on Research in Computer Security (ESORICS '12), Pisa, Italy, September 10–12, 2012*, vol. 7459 of *Lecture Notes in Computer Science*, pp. 541–556, Springer, Berlin, Germany, 2012.

[15] L.-C. Li, J.-H. Li, and J. Pan, "Self-healing group key management scheme with revocation capability for wireless sensor networks," *Journal on Communications*, vol. 30, no. 12, pp. 12–17, 2009.

[16] Z. Ming, W. Suo-ping, and X. He, "Dynamic key management scheme for wireless sensor networks based on cluster," *Journal of Nanjing University of Posts and Telecommunications (Natural Science)*, vol. 32, no. 1, 2012.

[17] G.-J. Wang, T.-T. Lv, and M.-Y. Guo, "Transitory initial key-based key management protocol in wireless sensor networks," *Chinese Journal of Sensors and Actuators*, vol. 20, no. 7, pp. 1581–1586, 2007.

[18] Y.-F. Ciou, F.-Y. Leu, Y.-L. Huang, and K. Yim, "A handover security mechanism employing the Diffie-Hellman key exchange approach for the IEEE802.16e wireless networks," *Mobile Information Systems*, vol. 7, no. 3, pp. 241–269, 2011.

[19] J. Li, X. Chen, J. Li, C. Jia, J. Ma, and W. Lou, "Fine-grained access control system based on outsourced attribute-based encryption," in *Computer Security—ESORICS 2013: 18th European Symposium on Research in Computer Security, Egham, UK, September 9–13, 2013. Proceedings*, vol. 8134 of *Lecture Notes in Computer Science*, pp. 592–609, Springer, Berlin, Germany, 2013.

[20] A. Zhu, S. Xu, S. Setia, and S. Jajodia, "Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach," in *Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP '03)*, pp. 326–335, Atlanta, Ga, USA, November 2003.

[21] 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 10th ACM Conference on Computer and Communications Security (CCS '03)*, pp. 42–51, Washington, DC, USA, October 2003.

[22] D. Liu and P. Ning, "Multi-level μTESLA: broadcast authentication for distributed sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 4, pp. 800–836, 2004.

[23] J. Li, Q. Wang, C. Wang, and K. Ren, "Enhancing attribute-based encryption with attribute hierarchy," *Mobile Networks and Applications*, vol. 16, no. 5, pp. 553–561, 2011.

[24] Y. S. Lee, J. W. Park, and L. Barolli, "A localization algorithm based on AOA for ad-hoc sensor networks," *Mobile Information Systems*, vol. 8, no. 1, pp. 61–72, 2012.