Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2008, Article ID 691571, 10 pages
doi:10.1155/2008/691571

Research Article

Dynamic Session-Key Generation for Wireless Sensor Networks

Chin-Ling Chen and Cheng-Ta Li

Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung County 41349, Taiwan

Correspondence should be addressed to Chin-Ling Chen, clc@mail.cyut.edu.tw

Received 28 November 2007; Revised 19 June 2008; Accepted 15 August 2008

Recommended by Jong Hyuk Park

Recently, wireless sensor networks have been used extensively in different domains. For example, if the wireless sensor node of a wireless sensor network is distributed in an insecure area, a secret key must be used to protect the transmission between the sensor nodes. Most of the existing methods consist of preselecting m keys from a key pool and forming a key chain. Then, the sensor nodes make use of the key chain to encrypt the data. However, while the secret key is being transmitted, it can easily be exposed during transmission. We propose a dynamic key management protocol, which can improve the security of the key juxtaposed to existing methods. Additionally, the dynamic update of the key can lower the probability of the key to being guessed correctly. In addition, with the new protocol, attacks on the wireless sensor network can be avoided.

Copyright © 2008 C.-L. Chen and C.-T. Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

1.1. The composition and application of the wireless sensor network

There are four main modules of the wireless sensor network, including the sensor module, the processor module, the communication module, and the power module. The major function of each module is introduced below.

(1) Sensor module: the sensor module is responsible for sensing an analog signal. The signal transformation component transforms the analog signal detected by the sensor module into a digital signal. The data will then be sent to the processing module for additional work to be done.

(2) Processor module: the processor module includes a storage component and a processing component. The function of the storage component is similar to the storage device in computer. The detected information is kept in the storage component. The processing module is similar to the CPU of a PC. It executes the stored programming code to coordinate and control the different components of the detector. The stored programming command or the command from the back-end terminal can, through the processing component, instruct the sensor component to collect information. After the arrangement of the collected information, it will be transmitted through the transmission module.

(3) Communication module: the communication module is mainly responsible for communication with other detectors, or transmission of the collected information to the base station. The media of the communication module include infrared rays, radio waves, and optic fibers. There are different options in accordance with various environments and applications.

(4) Power module: the power supply module is responsible for providing power to all of the components in the detector. As all operations consume electricity, this is a very important component. In general, the power of the detector is provided by a battery. Therefore, conserving electricity is the main consideration of the software and the hardware designs.

The general necessary characteristics of wireless sensor networks include ability for multiple deployments, low cost, small size, and an adequate battery power supply. The route transmissions of the wireless sensor network include the following types.

(1) Cluster: the cluster structure is the most representative routing protocol. The general practice is to group a large number of sensors into several clusters. In each of the clusters, a node is chosen as the cluster head, which collects

and converges on information from other sensor nodes and transmits the information to the base station.

(2) Chaining: the chaining structure differs from the cluster structure. Each detector node in the network is linked as a chain. In each round, a node in the chaining structure is chosen as the chaining head. Both ends of the chain then start transmitting data to adjacent nodes in the direction of the chaining head. In addition, each receiving node gathers the information. Finally, the chaining head transmits the information to the base station.

In recent years, wireless sensor networks have been used extensively in environmental monitoring, such as the collection of meteorological information, monitoring of health information, information gathering, and tracking on the battlefield. Using a sensor network in an environment such as a battlefield leaves information insecure. The enemy can eavesdrop by intercepting information meant to be transmitted from the sending node to the receiving node. Therefore, it is necessary to make use of secure transmission on wireless sensor networks. However, there are obvious restrictions on the resources of wireless sensor networks. The restrictions on the CPU are obvious, including memory, bandwidth, and the consumption of electricity. Therefore, it is very important to choose a proper encryption system. Furthermore, there are obvious pitfalls to the hardware of a wireless sensor node, including high cost and impractical implementation. Public key algorithms, such as Diffie-Hellman key management [1] or RSA signature [2], are not in fact feasible.

In this section, we will review the existing key protocols of wireless sensor networks. We have classified these protocols into three types: random key predistribution protocols, group-based key predistribution protocols, and hierarchical structure protocols.

1.2. Related work

In the past researches, several famous key managements in wireless sensor network have been proposed. Due to the previous method, the m sets of keys are selected from the key pool to form a key chain [3–7], which suffer from many attacks. In this paper, we have proposed a novel scheme for the generation of a dynamic key management to improve the previous methods. In this section, we will briefly review and analyze them.

1.2.1. Random key predistribution protocols

In 2002, Eschenauer and Gligor [4] proposed a random key predistribution infrastructure. This infrastructure includes three steps: a key predistribution step, a key sensor and sharing step, and a path of key establishment step. Before the deployment of any sensor node, m sets of keys are selected from a large key pool. The m keys form a key chain which will be sent to each sensor node. One key is selected between the nodes, which will later be used to transmit data among the group. This method is secure. However, each sensor node must store m keys. This is a problem for the memory and power consumption of the sensor node. Blom's method [8] uses a global matrix pool to replace the global key pool. In the key predistribution phase, each node randomly selects several matrices from the global matrices pool, and then loads a row of elements from each determined matrix into the node. In this case, any two adjacent nodes have a row of elements from the same matrix that can establish a pairwise key. Di Pietro et al. [7] proposed a random key transmission protocol. The random keys are transmitted between the sensor nodes so that any two nodes can establish a communication channel. The shortcoming of this method is that each sensor node must store more than three sets of keys. In order to increase the security, the number of keys must be increased. However, the augmentation of the number of keys also increases the loading of the sensor nodes. Furthermore, power consumption is also increased.

1.2.2. Group-based key predistribution protocols

The so-called group key predistribution protocol is used to divide the area of the nodes into several groups. The helicopter airdrops the nodes into a predefined area so that the sensor nodes have a higher probability of communicating properly.

Liu and Ning [6] proposed a paired key protocol. With a polynomial key pool and predistribution of a grid key, this protocol has higher elasticity on catch and attack, and superior sensor node communication of sensor node. However, a key algorithm is relatively complicated. More time is required to generate a key. Though the security can be improved, it cannot reach the responsiveness and convenience needed by the sensor network.

1.2.3. Hierarchical structure protocols

The hierarchy predistribution protocols include several cluster nodes in the base station and sensor nodes. Cluster nodes have stronger operational ability. Before deployment, each cluster node stores the keys. After deployment, the nodes will exchange the codes. At the same time, the cluster nodes will be informed of the code of the sensor nodes. Through this method, the whole network can communicate. However, if one of the nodes is caught, the information transmitted between the cluster nodes and the sensor nodes could be easily observed by an enemy. Therefore, the cluster nodes must increase the number of keys to improve security. However, the resources of sensor nodes are limited, making this impracticable. Therefore, Cheng and Agrawal [3] have proposed a bivalent polynomial. Cheng and Agrawal presented an improved key distribution mechanism (IKDM) by which the use of bivariate polynomials developed. Each gateway does not directly store nodes' gateway keys, but each gateway stores bivariate polynomial functions. After deployment, a node sends its ID code and the gateway numbers to the nearest gateway. Then, the gateway asks other gateways to obtain subkeys. The gateway can then compute the gateway keys of neighboring nodes from these subkeys. The other related scheme likes Jolly et al. [5] which also based on the identity-based symmetric keying scheme. This paper further discusses the addition of sensors issue.

1.2.4. Other protocols

Chan et al. [9] have proposed two secure protocols. Chan and Perrig presented peer intermediaries for a key establishment protocol (PIKE). Each node has an identity of the form (x, y). A node solely shares a pairwise key with each other node having the same x-coordinate or y-coordinate. After deployment, two adjacent nodes possess the pairwise key if their identities are half matched, or they can route a key with an intermediary node. For the base station, to achieve data security and authentication, an efficient key sharing algorithm must be used. For example, RC5 makes use of this secure algorithm to ensure authentication and security. Secondly, in order to ensure the safety of the source of information, a one-dimension hash chain, such as time efficient streamed loss-tolerant authentication (TESLA), is adopted to conduct the authentication of information.

1.3. Environmental requirements

(1) Confidence of data: in general, the wireless sensor network is deployed a region that people cannot reach, or in a dangerous area to conduct monitoring and information collection. An example of such a location would be a battlefield, where enemy positions are tracked. Therefore, the information collected by the sensor node must be accurate and confidential. Additionally, data transmission in the wireless sensor network is conducted by wireless radio frequency. When the sensor node transmits confidential information to the backend server, if there is no security mechanism to handle the information, the transmitted data could be exposed easily. Especially, when the information is transmitted from enemy positions, the process should be protected by the encryption system. The encryption system can be classified into two types: symmetric encryption systems and asymmetrical encryption systems. In symmetric encryption, the sensor nodes share one conference key for transmission. In asymmetrical encryption, the public key is adopted for transmission. However, due to the resource limitations of the sensor network and high cost, the use of an asymmetrical has proven impractical.

(2) Data authentication: in the sensor network, each region may include hundreds or even thousands of sensor nodes. Data transmission between the nodes is very common. If a hostile node exists, which broadcasts data constantly, and there is no data authentication between the sensor nodes, the network will be paralyzed. In addition, the resource consumption of the nodes will be increased, which will reduce the lifespan of the sensor node. Therefore, minimizing rounds of communication and minimizing rounds of a confirmable dynamic key management are important topics in sensor networking. The sensor nodes on the transmission end can share the key to encrypt the data to be sent. The sensor nodes on the receiving end can also share the same key to decrypt.

(3) Man-in-the-middle attack [10]: the so-called man-in-the-middle attack occurs when data is intercepted by a hostile node. During data transmission between the sensor nodes and cluster nodes, or cluster nodes and base station, the transmission is intercepted by the hostile node. The data transmitted by the sensor nodes is falsified and is resent again. The data received by the receiving nodes is thus not the original data to be transmitted. Therefore, the data received by the base station is not correct, and it must be solved by encryption mechanism.

(4) Replay attack [11]: the replay attack occurs when there is a hostile node among the sensor nodes of the region that wants to get the key. Packets are constantly resent in an attempt to obtain the key between the sensor nodes. Once the key is obtained, further attacks can be conducted. In order to solve this type of attack, we synchronize transmission times between the receiving end and the sending end. The time difference between transmission and reception can be used to determine whether the packet is acceptable; otherwise it can be abandoned.

(5) Memory limitation: with the limitation of the size of the sensor node, the memory capacity is also limited. The memory capacity of each sensor node is usually around dozens of MB. When the security of the wireless sensor network is enhanced, the memory capacity of the sensor node should also be considered.

(6) Computation limitation: the CPU is fixed in the sensor node to handle and calculate the data. However, limiting size and power consumption only allows for a low-end CPU model. For example, the StrongARM [12] from Intel and ATmega [13] from ATmel are the CPU commonly used.

On the basis of the one-way hash function, exclusive or operation and symmetric encryption, we have proposed a method to generate a dynamic key. Each time the sensor node transmits data, a new key will be generated through the previous two old keys. The new key will be used for encryption. When this sensor node transmits data the following time, the operation will be based on the new generated key and one of the old keys. These two keys become the key for this transmission. Other sensor nodes make use of the same method. When the sensor node transmits data to a cluster node, the cluster node will request the key of that sensor node from the base station. Since the base station has the two primary keys from all sensor nodes, it will transmit the required key of that sensor node to the cluster node. After receiving the key, the cluster node can begin decryption. When the number of sets of the received data is larger than a threshold value t, the data will be encrypted and transmitted to the base station. The method of generating the key is the same as with sensor nodes in order to ensure the accuracy of the information. In addition, one of the keys between the base station and cluster nodes, and one between the base station and the sensor nodes will be updated dynamically in order to improve the security of the network.

2. DETAILS OF THIS PROTOCOL

2.1. Notation

In this infrastructure, some abbreviations are used. These symbols and their corresponding meanings are listed as Table 1.

Table 1: Notation.

$h()$	Use for the one-way hash function of key generation
a_j, a_{j-1}	Two parameters for generation of key pre-deployed in the jth sensor node
$\text{msg}_{\text{finish}}$	Message for the cluster node informing sensor node to update the key
K_{si}	The ith of the key of the sensor node
K_{ci}	The ith of the key of the cluster node
K_{msg}	The key used for encryption or decryption of the $\text{msg}_{\text{finish}}$
Seed	The seed for updating the key pre-deployed in each of the sensor nodes
ID_{si}	The identity of the ith sensor node
ID_{list}	The identity set list of the t sensor nodes received from the cluster nodes, such as $\text{ID}_{\text{list}} = (\text{ID}_{s1}, \text{ID}_{s2}, \ldots, \text{ID}_{st})$
K_{list}	The key of the sensor nodes needed by the cluster node, such as $K_{\text{list}} = (K_{s1}, K_{s2}, \ldots, K_{st})$
M_i	The plain text information generated by the ith sensor node
M_f	The latest information received by the base station
SRND_i	The ith nonce is generated by sensor node
CRND_i	The ith nonce is generated by node
BRND_i	The ith nonce is generated by base station
$E(M, K)$	The symmetric encryption of the infrastructure makes use of key K to encrypt M
$D(M, K)$	The symmetric decryption of the infrastructure makes use of key K to decrypt M
$A \cong B$	Compare whether A is equal to B or not

2.2. Environmental conditions

(1) In the wireless sensor network, we will make use of cluster management for transmission of data. In general, we will deploy hundreds or even thousands of sensor nodes in a wireless sensor network. Additionally, we will divide the deployed sensor nodes into different regions so that each sensor node can transmit data in the effective range.

(2) In each of the regions, a sensor node will be chosen automatically as the cluster node. We will use an algorithm to choose the cluster node, for example, Park and Corson [14], Perkins and Royer [15], Johnson and Maltz [16]. When the sensor node transmits the collected data to the backend base station, the encrypted data will be sent to cluster node. Once the cluster node has received a certain amount of packets, the data will be arranged, encrypted, and then transmitted to the backend base station. Figure 1 is the diagram of transmission paths of sensor nodes.

(3) After the first deployment of the sensor network, the cluster nodes will be chosen. The sensor nodes will broadcast to the cluster nodes so that each cluster node knows the number of sensor nodes in the specific region. The cluster nodes also will record the identity of the sensor nodes for future transmission.

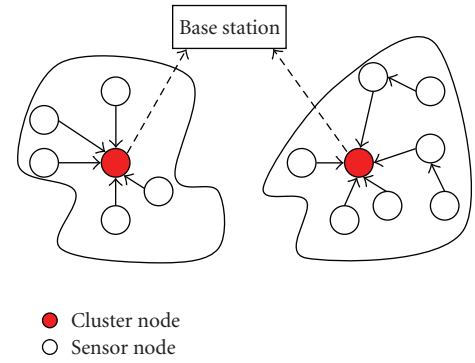

Figure 1: Transmission paths of the sensor network.

(4) Once each of the sensor nodes is dispatched from the factory, we will preset two parameters, such as a_i and a_{i-1}. Also a new key will be generated by a one-way hash function, for which the key will be used to communicate with the cluster node. If the sensor node is chosen as a cluster node, the parameters a_i and a_{i-1} will also be used to generate the session key for communicating with the base station.

(5) Each sensor node will preset a message key K_{msg} and a seed for updating the key in order to encrypt/decrypt the message informing the sensor nodes for the update of sensor nodes. The hash function will be used to update the key of a message in each round so that the sensor nodes can receive the secure message for the update of a key.

(6) For data transmission between the nodes, we make use of jumping transmission. When the first level sensor nodes have collected data, the encrypted data, together with the code of the nodes, will be transmitted to the second-level sensor nodes. The second-level sensor nodes will also encrypt the collected data. Together with the data received from the first-level sensor node and the codes of the nodes, the data will be transmitted to the next level of sensor nodes and so on. Once the cluster node receives a series of data from the codes of the sensor nodes, it knows which sensor nodes have transmitted data to it. According to the codes of the sensor nodes, the cluster node can request the key list from the sensor nodes of the base station.

(7) When a sensor node cannot transmit data to cluster nodes in period time. The base station determines the sensor node lost. It is possible that the sensor node lost power or was captured. The user can use the added new node protocol to join the wireless sensor networks, the whole network can work normally, see Figure 3.

2.3. Key generation protocol

In our secure protocol, dynamic key management mechanism has been proposed. Two keys are preset in each sensor node. The new key for the next round is generated by these two keys. Two keys will also be preset in the cluster node. The generation of the session key will be the same as those in the sensor node.

We have divided the mentioned protocol into the following five steps, as shown in Figure 2.

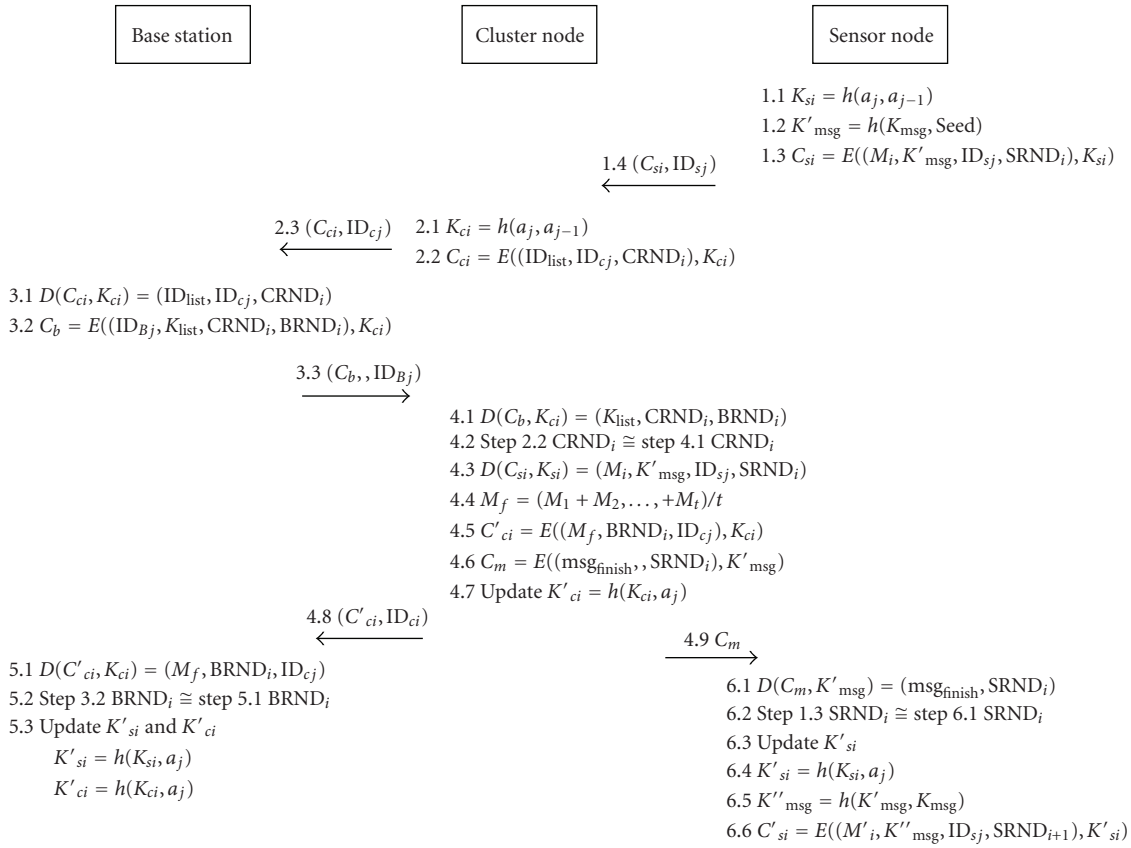

Figure 2: Key generation communication protocol. Note: in Figure 2 scenarios, we present in ith round; jth node identification.

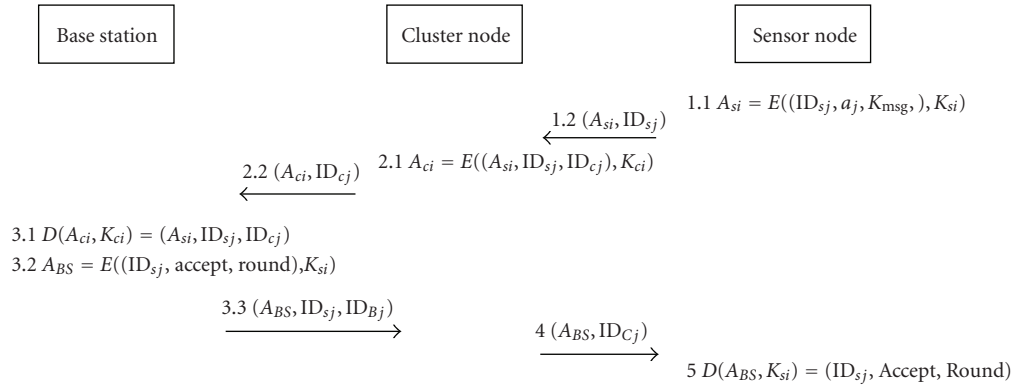

Figure 3: Add new node protocol.

Step 1. When the deployed sensor node i returns the collected information, the sensor node will make use of the preset parameters a_j and a_{j-1} to generate a key, K_{si}, where

$$K_{si} = h(a_j, a_{j-1}). \tag{1}$$

Further, the two parameters K_{msg} and the Seed preset in each of the nodes will use the hash function to generate a new message key, K'_{msg}, where

$$K'_{\text{msg}} = h(K_{\text{msg}}, \text{Seed}). \tag{2}$$

At that moment, the sensor node will make use of K_{si} to encrypt the detected data M_i and the preset K'_{msg}, ID_{sj}, and SRND_i. A complete packet C_{si} will be generated as follows:

$$C_{si} = E((M_i, K'_{\text{msg}}, \text{ID}_{sj}, \text{SRND}_i), K_{si}). \tag{3}$$

The (C_{si}, ID_{sj}) is then transmitted to the cluster node.

Step 2. When the cluster node receives more than t packets, or when the period is longer than a specific time, the cluster node will record and transmit the identity, ID_{sj}, of the sensor

node. It will also arrange a list, $\mathrm{ID}_{\mathrm{list}}$, according to the codes of the received sensor nodes so that

$$\mathrm{ID}_{\mathrm{list}} = (\mathrm{ID}_{s1}, \mathrm{ID}_{s2}, \ldots, \mathrm{ID}_{st}). \quad (4)$$

The cluster node will also make use of the two preset parameters, a_j and a_{j-1} to generate a key, K_{ci}, where

$$K_{ci} = h(a_j, a_{j-1}). \quad (5)$$

At that moment, the cluster node will make use of K_{ci} to encrypt $\mathrm{ID}_{\mathrm{list}}$, ID_{cj}, and nonce CRND_i as a complete packet, C_{ci}, where

$$C_{ci} = E((\mathrm{ID}_{\mathrm{list}}, \mathrm{ID}_{cj}, \mathrm{CRND}_i), K_{ci}). \quad (6)$$

Together, with the code ID_{cj} of the cluster node, it will be transmitted to the base station.

Step 3. When the base station receives the packet from the cluster node, it will confirm the code, ID_{cj}, of the cluster node and seek the key, K_{ci}, of that cluster node in the code database according to the code of the cluster node. The K_{ci} is used for decryption

$$D(C_{ci}, K_{ci}) = (\mathrm{ID}_{\mathrm{list}}, \mathrm{CRND}_i). \quad (7)$$

The base station will receive the $\mathrm{ID}_{\mathrm{list}}$ sent from the cluster node. If this accords with the list, it will search for the key of the corresponding sensor node from the database and arrange them into the key list, K_{list}, where

$$K_{\mathrm{list}} = (K_{s1}, K_{s2}, \ldots, K_{st}). \quad (8)$$

At that moment, the base station will make use of K_{ci} to encrypt $(\mathrm{ID}_{Bj}, K_{\mathrm{list}}, \mathrm{CRND}_i, \mathrm{BRND}_i)$. The encrypted data, C_b, will be returned to the cluster node, where

$$C_b = (E((\mathrm{ID}_{Bj}, K_{\mathrm{list}}, \mathrm{CRND}_i, \mathrm{BRND}_i), K_{ci})). \quad (9)$$

Step 4. When the cluster node receives the returned data from the base station, it will make use of the key, K_{ci}, generated by itself to decrypt

$$D(C_b, K_{ci}) = (\mathrm{ID}_{Bj}, K_{\mathrm{list}}, \mathrm{CRND}_i, \mathrm{BRND}_i). \quad (10)$$

The cluster node compares the CRND_i in (6) whether equal to the CRND_i in (10).

If it is true, the cluster node only can use the K_{si} from K_{list} so that it knows the key of the node that transmitted the data. The key, K_{si}, will then be used for decryption, and the data

$$D(C_{si}, K_{si}) = (M_i, K'_{\mathrm{msg}}, \mathrm{ID}_{sj}, \mathrm{SRND}_i) \quad (11)$$

returned from the sensor node can be obtained. The cluster node will calculate the average value of each set of data and obtain M_f, where

$$M_f = \frac{(M_1 + M_2 + \cdots + M_t)}{t}. \quad (12)$$

This ensures the data is accurate when it is transmitted to backend. This cluster node will make use of K_{ci} to encrypt M_f and nonce BRND_i as a complete packet, C'_{ci}, where

$$C'_{ci} = E((M_f, \mathrm{BRND}_i, \mathrm{ID}_{cj}), K_{ci}). \quad (13)$$

Together with the code, ID_{cj}, of the cluster node, it is transmitted to the base station.

At that moment, the cluster node will update the session key

$$K'_{ci} = h(K_{ci}, a_j) \quad (14)$$

for the next round.

Moreover, the cluster node will make use of the key, K'_{msg}, transmitted from the sensor node to encrypt the transmitted update message $\mathrm{msg}_{\mathrm{finish}}$ of key as follows:

$$C_m = E((\mathrm{msg}_{\mathrm{finish}}, \mathrm{SRND}_i), K'_{\mathrm{msg}}). \quad (15)$$

The encrypted packet, C_m, will then be broadcasted to the sensor nodes, and the sensor nodes will be informed of the completion of message transmission.

Step 5. When the base station receives the packet from the cluster node, it will confirm the identity, ID_{cj}, of the cluster node first. Also, it will search for the key, K_{ci}, of the cluster node from the database according to the code of the cluster node. It will make use of K_{ci} to decrypt

$$D(C'_{ci}, K_{ci}) = (M_f, \mathrm{BRND}_i). \quad (16)$$

The base station compares the BRND_i in (9) whether equal to the BRND_i in (16). If it is true, the base station only convince the received information, M_f, transmitted from the cluster node. Simultaneously, the base station will update the key of the cluster node and sensor node, which will be updated to K'_{si} and $K'ci$, where

$$\begin{aligned} K'_{si} &= h(K_{si}, a_j), \\ K'_{ci} &= h(K_{ci}, a_j). \end{aligned} \quad (17)$$

Step 6. After receiving the message C_m, the sensor node will make use of K'_{msg} for decryption, and obtain the message $(\mathrm{msg}_{\mathrm{finish}}, \mathrm{SRND}_i)$ as follows:

$$D(E(C_m, K'_{\mathrm{msg}})) = (\mathrm{msg}_{\mathrm{finish}}, \mathrm{SRND}_i). \quad (18)$$

The sensor node compares the SRND_i in (3) whether equal to the SRND_i in (18). If it is true, the key will then be replaced. The previously generated keys, K_{si} and a_j, are used to generate a new key, K'_{si}, where

$$K'_{si} = h(K_{si}, a_j). \quad (19)$$

The next time the data is returned, the K'_{si} will be adopted to encrypt the transmitted data. When the sensor node transmits the data in the second round, the original message key, K'_{msg}, will be updated to K''_{msg}, where

$$K''_{\mathrm{msg}} = h(K'_{\mathrm{msg}}, K_{\mathrm{msg}}). \quad (20)$$

The message key, K''_{msg}, together with the message M'_i, the sensor node will make use of K'_{si} to encrypt them to C'_{si}, where

$$C'_{si} = (E((M'_i, K''_{\text{msg}}, \text{SRND}_{i+1}, \text{ID}_{sj}), K'_{si}), \text{ID}_{sj}). \quad (21)$$

When the sensor node transmits data for the third time, the message key must be updated to K'''_{msg}, where

$$K'''_{\text{msg}} = h(K''_{\text{msg}}, K'_{\text{msg}}). \quad (22)$$

The updated message key, together with K'''_{msg}, and the message M''_i, the sensor node makes use of K'''_{si} to encrypt them to C''_{si}, where

$$C''_{si} = (E((M''_i, K'''_{\text{msg}}, \text{ID}_{sj}, \text{SRND}_{i+1}), K''_{si}), \text{ID}_{sj}). \quad (23)$$

The session keys K_{si}, K'_{si}, and K''_{si} are for encrypted message between the cluster node and sensor node. In addition, the updated K''_{msg} and K'''_{msg} are the message keys for the cluster node transmitting complete messages $\text{msg}_{\text{finish}}$, to the sensor node during the second and third rounds.

2.4. Add new node protocol

If the base station cannot obtain the messages from the sensor nodes in a specific period (the sensor node could be power down or captured by adversary), the new sensor node should be redeployed, and the protocol will be executed. The scenarios are shown in Figure 3.

Step 1. When a new sensor node is joined to the wireless sensor networks, the sensor node make use of K_{si} to encrypt the preset parameters a_j and K_{msg} with the ID_{sj} of the sensor node; a complete packet, A_{si}, is generated as follows:

$$A_{si} = E((\text{ID}_{sj}, a_j, K_{\text{msg}}), K_{si}). \quad (24)$$

The (A_{si}, ID_{sj}) is then transmitted to the cluster node.

Step 2. The cluster node receives the request packet from the sensor node, which will make use of the key, K_{ci}, to encrypt the packet, A_{si}; the code, ID_{si}, of the sensor node; and the code, ID_{ci}, of the cluster node

$$A_{ci} = E((A_{si}, \text{ID}_{sj}, \text{ID}_{cj}), K_{ci}). \quad (25)$$

Together with the code, ID_{si}, of the node, it will be transmitted to the base station as a complete packet (A_{ci}, ID_{cj}).

Step 3. The base station will receive the packet from the cluster node, and it will make use of the key, K_{ci}, to decrypt and obtain the complete message

$$D(A_{ci}, K_{ci}) = (A_{si}, \text{ID}_{sj}, \text{ID}_{cj}). \quad (26)$$

The base station can confirm the a_j and K_{msg}, if it is not true, the cluster node will abandon this packet. Otherwise, the base station will make use of the key, K_{ci}, to encrypt the message of the ID_{sj}, Accept and the Round of the network communication times

$$A_{BS} = E((\text{ID}_{sj}, \text{Accept}, \text{Round}), K_{si}). \quad (27)$$

Together with the codes ID_{si} and ID_{Bi}, it will be transmitted to the cluster node as a complete packet, $(A_{BS}, \text{ID}_{sj}, \text{ID}_{Bj})$, and send to cluster node.

Step 4. The cluster node receives the data from the base station so that it can confirm the code, ID_{Bj}, of the base station. If it is not true, the cluster node will abandon this packet. Otherwise, the cluster node can broadcast (A_{BS}, ID_{Cj}) to the sensor nodes.

Step 5. After the sensor node receives the packet, it can use of K_{si} to decrypt and attain the complete message

$$D(A_{BS}, K_{si}) = (\text{ID}_{sj}, \text{Accept}, \text{Round}). \quad (28)$$

According to the Round, the sensor node will calculate the communication key of the wireless sensor network.

3. ANALYSIS OF SECURITY AND PERFORMANCE

3.1. Analysis of security

3.1.1. Dynamic key management

Regarding the generation of a key, the previous predeployment has been changed. M sets of keys from the key pool used to generate a key chain will no longer be chosen. The communication between any two nodes will make use of these m sets of keys to negotiate and communicate. In our infrastructure, for each data transmission, a new key will be generated from the previous two keys. For example, if the key is $K_{si} = h(a_i, a_{i-1})$ for the first transmission, $K'_{si} = h(K_{si}, a_i)$ for the second transmission, and $K''_{si} = h(K'_{si}, K_{si})$ for the third transmission, and so on. This reduces the possibility of the attacker correctly guessing the key from the key chain and using it repeatedly. This also improves the security of the network. In addition, the cluster node makes use of similar dynamic key generation when it transmits a complete message. The predeployed K_{msg} and Seed are used for operation, where $K'_{\text{msg}} = h(K_{\text{msg}}, \text{Seed})$ is the message key. The message key in the second round will be updated to $K''_{\text{msg}} = h(K'_{\text{msg}}, K_{\text{msg}})$, and so on. The attacker is not able to imitate the cluster node to transmit a complete message key to update the key.

3.1.2. Prevention of malicious guessing attacks

When the deployed sensor network exists for a certain period, the key and the database of the base station will be updated so that the attacker cannot have current knowledge pertaining to the key. Furthermore, each node includes the records of not more than three keys, two old keys and one newly generated key. When the new key is generated, the oldest key will be updated. This can improve the security of the network and reduce the memory load of the nodes.

Table 2: The performance analysis of key generation communication protocol.

Relationship between the nodes	Rounds	Time complexity
Sensor node and cluster node	2	$2T_E + 1T_M$
Cluster node and base station	3	$3T_E + 3T_M$

T_E: the time complexity of using symmetric encryption algorithm.
T_M: the time complexity needed for plaintext (e.g., ID_{sj}, ID_{cj}, ID_{Bj}) transmission.

Table 3: The performance analysis of add new node protocol.

Relationship between the nodes	Rounds	Time complexity
Sensor node and cluster node	2	$2T_E + 2T_M$
Cluster node and base station	2	$2T_E + 3T_M$

T_E: the time complexity of using symmetric encryption algorithm.
T_M: the time complexity needed for plaintext (e.g., ID_{sj}, ID_{cj}, ID_{Bj}) transmission.

3.1.3. Prevention of replay attacks

In each of the communication sessions, including the sensor node to the cluster node or the cluster node to the base station, the "two-way" authentication has been adopted to prevent the replaying attack. We use the nonce to confirm each communication message. The related descriptions are shown in step 4.2, 5.2, and 6.2 of Figure 2. Therefore, our scheme can prevent the replaying attacks.

3.1.4. Prevention of the falsification attack

For the transmission between the cluster node and sensor node, we adopt key K_{si} for encryption. When the sensor node returns the data to the cluster node, $E((M_i, K'_{\mathrm{msg}}, \mathrm{ID}_{sj}, \mathrm{SRND}_i), K_{si})$ is adopted for encryption. When the communication between the cluster node and the base station is finished, the K_{list} is obtained. The base station returns the K_{si} to the cluster node and the decryption can occur. If the received key cannot decrypt the received encrypted packet, it will be regarded as an illegal packet and will be abandoned. This practice can ensure the integrity of the data transmission, and guarantee the data is sent from the sensor node administrated by the cluster node.

3.1.5. Prevention of man-in-the-middle-attacks and guarantee of data privacy

When the sensor node communicates with the cluster node, the encryption mechanism is adopted to prevent the attack and ensure data privacy. The transmission message is encrypted into

$C_{si} = E((M_i, K'_{\mathrm{msg}}, \mathrm{ID}_{sj}, \mathrm{SRND}_i), K_{si})$. The cluster node and the base station also adopt a similar method to prevent attacks and ensure data privacy. For example,

(1) key generation communication protocol:

$$C_{ci} = E((\mathrm{ID}_{\mathrm{list}}, \mathrm{ID}_{cj}, \mathrm{CRND}_i), K_{ci}),$$
$$C_b = E((\mathrm{ID}_{Bj}, K_{\mathrm{list}}, \mathrm{CRND}_i, \mathrm{BRND}_i), K_{ci}),$$
$$C'_{ci} = E((M_f, \mathrm{BRND}_i, \mathrm{ID}_{cj}), K_{ci}), \text{ and}$$
$$C_m = E((\mathrm{msg}_{\mathrm{finish}}, \mathrm{SRND}_i), K'_{\mathrm{msg}}).$$

(2) Add new node protocol:

$$A_{si} = E((\mathrm{ID}_{sj}, a_j, K_{\mathrm{msg}}), K_{si}),$$
$$A_{ci} = E((A_{si}, \mathrm{ID}_{sj}, \mathrm{ID}_{cj}), K_{ci}),$$
$$A_{BS} = E((\mathrm{ID}_{sj}, \mathrm{Accept}, \mathrm{Round}), K_{si}).$$

Therefore, the attacker cannot obtain the protected data. Furthermore, the cluster node makes use of K_{msg} to encrypt the complete message and the message key will be updated each round. Therefore, the attacker cannot imitate the cluster node to transmit a message. The man-in-the-middle-attack can thus be prevented.

3.1.6. The node captured attack analysis

For transmission between the cluster node and sensor node, we adopt key K_{si} for encryption. We make use of the one way hash function to generate the key. Because the one way hash function can prevent the attacker from inversing the key. (1) $H(x)$ is relatively easy to compute for any given x making both hardware and software implementations practical. (2) For any given value h, it is computationally infeasible to find x such that $H(x) = h$. This is sometimes referred to in the literature as the one-way property. (3) For any given block x, it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$. This is sometimes referred to as weak collision resistance.

3.2. Performance analysis

In Tables 2 and 3, we analyze the performance of key generation communication protocol and add new node protocol, respectively.

3.3. Comparison

We make a comparison with the related sensor network in Table 4.

4. CONCLUSION

Due to the previous method, the m sets of keys are selected from the key pool to form a key chain, which involves many shortcomings. In this paper, we have proposed the infrastructure for generation of a dynamic key capable of supplanting previous methods. Through dynamic key generation management, the infrastructure we have proposed includes the following contributions.

(1) Due to the limitations of wireless sensor network, such as a limited power source and scarce memory, we adopt batch communication method to reduce the power consumption of the sensor node. In addition, our method requires each node to record not more than three keys and it is not necessary to record the complete key chain. This method can conserve the memory of the sensor node significantly.

Table 4: The comparison of the related sensor network.

Protocol		Our scheme	IKDM [3]	LEKM [5]
Captured attack analysis		Yes	Yes	N/A
Add new node algorithm		Yes	N/A	Yes
Detail security analysis		Complete	Partial (only captured attack analysis)	N/A
Stored cost	Sensor node	Two session keys and one cluster node ID	Two session keys and one cluster node ID	Two session keys and one cluster node ID
	Cluster node	Two session keys and one base station ID	One session key and two polynomial functions	$\frac{n}{m} + (m-1) + 1$
The time cost of	Sensor node	Specific: $(2t_h + 2t_u)$	N/A	N/A
key computation	Cluster node	Specific: $(t_h + t_U)$	$\frac{(n \times t_{\text{poly}} \times l)}{m}$	N/A

m: number of the cluster nodes in sensor networks; n: number of the sensor nodes in cluster.
l: times of the cluster division; t_{poly}: time cost of polynomial function.
t_h: time cost of key generation; t_U: time cost of key update.

(2) The key for each transmission will only be used once. In the next transmission, another key will be used. This method can reduce the probability of the attacker guessing the key correctly and can improve security.

(3) For transmission, we make use of the "two-way" authentication in the process of transmission. Through the comparison nonce of the receiving end and the sending end, the replaying attacks can be prevented.

Regarding the application of the wireless sensor, the infrastructure we have proposed can be used in military situations, such as monitoring the enemy on the battlefield. The cluster node will conduct statistical calculations of the received data from the sensor nodes, and the data is then transmitted to the base station. This can ensure that the information received by the base station is accurate. This can also be applied in weather forecasting. Calculations from the cluster node can increase the accuracy of detected temperature and humidity. In the future, we will implement this prototype in the real environment and prove it is realistic.

ACKNOWLEDGMENTS

The referees' insightful comments helped to improve the paper significantly. This research was supported by National Science Council, Taiwan, under Contract no. NSC-97-2221-E-324 -013.

REFERENCES

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

[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] 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.

[4] 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, Washington, DC, USA, November 2002.

[5] G. Jolly, M. C. Kuscu, P. Kokate, and M. Younis, "A low-energy key management protocol for wireless sensor networks," in *Proceedings of the 8th IEEE International Symposium on Computers and Communication (ISCC '03)*, vol. 1, pp. 335–340, Antalya, Turkey, June-July 2003.

[6] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS '05)*, vol. 8, pp. 41–77, Alexandria, Va, USA, November 2005.

[7] R. Di Pietro, L. V. Mancini, and A. Mei, "Random key-assignment for secure wireless sensor networks," in *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03)*, pp. 62–71, Fairfax, Va, USA, October 2003.

[8] R. Blom, "An optimal class of symmetric key generation systems," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT '84)*, vol. 209, pp. 335–338, Paris, France, April 1984.

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

[10] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS '00)*, vol. 2, pp. 33–43, Maui, Hawaii, USA, January 2000.

[11] H. Soroush, M. Salajegheh, and T. Dimitriou, "Providing transparent security services to sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, pp. 3431–3436, Glasgow, Scotland, June 2007.

[12] Intel company, http://www.intel.com/design/network/products/cpp/ixc1100.htm?iid=SEARCH.

[13] Atmel company: AVR 8-Bit RISC processor, http://www.atmel.com/dyn/products/param_table.asp?family_d=607&OrderBy=part_no&Direction=ASC.

[14] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proceedings of the 16th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '97)*, vol. 3, pp. 1405–1413, Kobe, Japan, April 1997.

[15] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pp. 90–100, New Orleans, La, USA, February 1999.

[16] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imielinski and H. F. Korth, Eds., vol. 353, pp. 153–181, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.