A Dynamic Key Management Scheme Based on Secret Sharing for Hierarchical Wireless Sensor Networks

Enjian Bai*, Xueqin Jiang
College of Information Science & Technology, Donghua University
Shanghai 201620, P. R. China
Engineering Research Center of Digitized Textile & Fashion Technology,
Ministry of Education Shanghai 201620, P. R. China
*Corresponding author, e-mail: baiej@dhu.edu.cn, xqjiang@dhu.edu.cn

Abstract

Since wireless sensor networks (WSN for short) are often deployed in hostile environments in many applications, security becomes one of the critical issues in WSN. Moreover, due to the limitation of the sensor nodes, traditional key management schemes are not suitable for it. Thereby, a feasible and efficient key management scheme is an important guarantee for WSN to communicate securely. For the moment, many protocols have been proposed and each has its own advantages. However, these protocols cannot provide sufficient security in many cases, such as node capture attack, which makes WSN more vulnerable than traditional wireless networks. Key protection and revocation issues must be considered with special attention in WSN. To address these two issues, we propose a dynamically clustering key management scheme based on secret sharing for WSN. The scheme combined the hierarchical structure of wireless sensor networks with dynamic key management scheme. The analysis results show that the scheme has strong security and resistance of captured attack, as well as low communicational overhead, and it well meets the requirement of scalability.

Keywords*: wireless sensor networks, key management, threshold secret sharing, hierarchical.*

1. Introduction

Compared to the traditional wireless networks, wireless sensor networks (WSN for short) have such characteristics as the large scale of deployment, power limitation, limited computing ability and memory capability, the limitation of communication bandwidth and range and the dynamic characteristic. Due to these limitations, WSN is faced with the challenge of security in such applications as military fields, environment monitoring, medical treatment and hazardous area [1-5]. Therefore, security is the critical issue in the WSN and the key management is the crucial point of the security issues. Because of the characteristics, following security problem for the key management in a WSN also should be taken into consideration:

(1) Because of the wireless communication, it is easy to eavesdrop, intercept or interrupt the messages in a network.

(2) The key management scheme must be scalable because that the size of network would change even after deployment.

(3) The limitation of the sensor nodes such as power and computing ability, need to be taken into consideration in the design of a key management scheme.

(4) Due to the limitation of the sensor nodes, asymmetric key which is widely used in traditional networks is not suitable for WSN.

In the past few years, several famous key management schemes in WSN have been proposed. Eschenauer and Gligor proposed the first key pre-distribution scheme-E-G scheme [6]. It includes three steps: key pre-distribution step, key sensor and sharing step, and path of key establishment step. This method is secure and easy to implement. However, it can not provide sufficient security when the number of compromised nodes increases. The adversary can easily extract the cryptographic information stored in a captured sensor node. On the basis of E-G scheme, Chan proposed a *q*-composite scheme [7]. In his scheme, two sensor nodes have to share at least *q* (*q*≥2) common keys to establish a secure connection. It improves the network resilience against node capture attacks, but it limits the expanding of the network. Other

Received October 25, 2012; Revised January 20, 2013; Accepted February 1, 2013

key pre-distribution schemes based on dynamic calculation were proposed, such as polynomial-based key pre-distribution scheme [8] and matrix-based key pre-distribution scheme [9]. They can improve the security of the network. The problem is that it increases the calculation and memory consumption of the sensor nodes. In order to meet the requirements of practical applications, researchers have proposed some key management schemes based on clustering. For example, the key management scheme based on KDC (Key Distribution Center) makes each sensor node share a key with KDC [10]. Although it reduces the consumption of the sensor nodes, it depends on the base station excessively. LEAP [11] establish four kinds of separate keys for different communication patterns in WSN, and the compromise of any node won't affect the security of other nodes. But the exposure of the global key used to facilitate the key establishment between neighboring nodes will threaten the security of whole network. Later, many key management schemes with different points of emphasis have been proposed [12-14]. Cheng and Agrawal made use of bivariate polynomials to propose an improved key distribution mechanism which is suitable for a large-scale WSN [12]. Chen and Li proposed a dynamic key management protocol which can lower the probability of the key to being guessed correctly by the dynamic update of the key. But they increase the communication overhead and waste the energy [13]. In [14], Wen et al. presented a new key management protocol with robust continuity for WSN to minimize the key management redesign effort as well as to make the node flexible and adaptable to many different applications. But the key update mechanism of this scheme is relatively complicated without taking overheads into consideration.

Each of those schemes has its advantages and shortcomings, and they cannot provide sufficient security when the compromised sensor nodes increased. To meet this problem, our protocol presents a dynamic key management scheme based on threshold secret sharing. It divides the master key into several sub keys and each cluster head only stores its sub key. The adopting of secret sharing and dynamically clustering key management improves the security of the network, and the attackers cannot reconstruct the master key due to the key update mechanism proposed in our scheme. Furthermore, our protocol is balanced in terms of computation and storage overhead.

2. Related Works

2. 1. Threshold Secret Sharing Scheme

A (t, n) threshold secret sharing scheme is a method that a secret K can be shared among n participants. At least t or more participants can reconstruct the secret, but t-1 or fewer participants can get nothing about the secret.

In 1979, Shamir proposed the first (t, n) threshold secret sharing scheme based on Lagrange interpolation polynomial [15]. The scheme can be described as follows:

Step 1: The secret holder randomly chooses $x_1, x_2, \ldots, x_n \in GF(q)$ ($x_i \neq 0$, q is a prime) and distributes them to n participants.

Step 2: The secret holder randomly chooses t-1 integers $a_1, a_2, \ldots, a_{t-1} \in GF(q)$ and constructs (t-1)th degree polynomial $f(x)=s+a_1x+a_2x^2+\ldots+a_{t-1}x^{t-1} \bmod q$, where s is the secret.

Step 3: For $1 \le i \le n$, the secret holder computes $y_i=f(x_i)$ and distributes them to the participants in security as their sub keys.

Step 4: By using the Lagrange interpolation polynomial, with the knowledge of t pairs of (x_i, y_i), the t participants of n can reconstruct the secret s.

$$s = f(0) = \sum_{i=1}^{t} y_i \prod_{\substack{j=1 \\ j \neq i}}^{t} \frac{-x_j}{x_i - x_j} \bmod q \tag{1}$$

In hierarchical WSN, each cluster head takes charge of the sensor nodes in its cluster. They aggregate data transmitted from their sensor nodes, fuse and filter received data and send them to the base station. Therefore, the communication between base station and cluster head needs reliable security assurance to prevent the important data revealing. The divided master key for secure communication in cluster heads can prevent the node capture attack based on the idea of secret sharing. In addition, there is no need for cluster head to communicate with station at frequent intervals. So the introduction of the secrete sharing won't increase the communication and computational overheads for the master key reconstruction markedly. To sum up, by using the threshold secret sharing, the divided storage of the mater key in our scheme has actual significance.

2. 2. Hierarchical Wireless Sensor Networks

Before introducing our protocol, we must make some assumptions:
(1) The timing of the sensor nodes is precisely synchronized with each other.
(2) The base station is located in a well-protected place and takes charge of the whole network's operation.
(3) The base station is provided with an intrusion detection mechanism. It can detect out whether the node is normal or not, thereby deciding whether to trigger the operation of node deletion.

In our scheme, the sensor nodes are divided into three different layers:
BS: Base station (BS) is used to connect the WSN with external network. As a control center, BS has unlimited computational and communication power and memory storage capacity.
CH: Cluster head (CH) is responsible for the coordination, the data retransfer and the management of all the nodes in the cluster.
SN: Sensor node (SN) can collect information of surrounding environment and transmit them to the cluster head.

Illustrated by Figure 1, the sensor nodes of sensor node layer can communicate not only with its cluster head directly, but also with other sensor nodes of the same cluster. Meanwhile, the CHs of cluster head node layer can communicate with each other and relay data between its cluster members and the BS.

In the clustering WSN, network is divided into several clusters according to different clustering protocol. In each of the clusters, a node is chosen as the CH. Because that the CH is responsible for processing of the data in cluster and transmission to BS, it has relatively large energy consumption and must be replaced periodically to balance the energy cost. Low-Energy Adaptive Clustering Hierarchy (LEACH [16]), one of the first clustering algorithms provides a balance of energy consumption through a random rotation of CHs. Our protocol uses LEACH to randomly choose CHs, and no longer describes in detail.

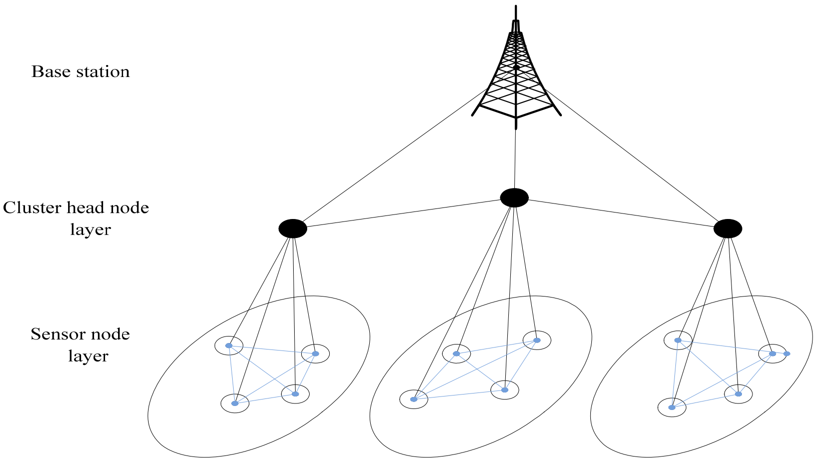

Figure 1. Hierarchical wireless sensor network model

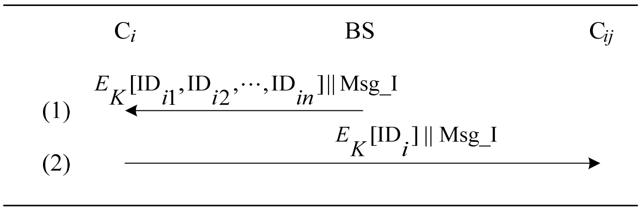
C_i
BS
C_{ij}
(1)
$E_K[\mathrm{ID}_{i1}, \mathrm{ID}_{i2}, \cdots, \mathrm{ID}_{in}] \| \mathrm{Msg\_I}$
(2)
$E_K[\mathrm{ID}_i] \| \mathrm{Msg\_I}$

Figure 2. Network initialization

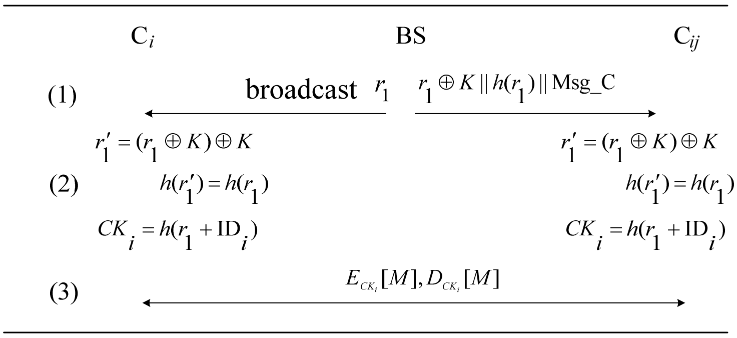
C_i
BS
C_{ij}
(1)
broadcast r_1
$r_1 \oplus K \| h(r_1) \| \text{Msg\_C}$
(2)
$r_1' = (r_1 \oplus K) \oplus K$
$h(r_1') = h(r_1)$
$CK_i = h(r_1 + \text{ID}_i)$
$r_1' = (r_1 \oplus K) \oplus K$
$h(r_1') = h(r_1)$
$CK_i = h(r_1 + \text{ID}_i)$
(3)
$E_{CK_i}[M], D_{CK_i}[M]$

Figure 3. Session key establishment in cluster

3. Proposed Key Management Scheme

Our protocol adopts (*t*, *n*) threshold secret sharing scheme in hierarchical WSN, and makes use of the key update mechanism to meet the dynamic requirements of WSN. There are two keys used to encrypt information in this scheme. The encryption key between sensor node and cluster head is generated by the preset information to reduce the required storage overhead. And The master key shared between cluster heads and base station is divided into *n* sub keys and distributed to each cluster head dynamically so that to increase the security of network. Descriptions of the notations involved in this paper are listed as Table 1.

3.1. Network Initialization

Before network deployment, each sensor node is initially preloaded with a random number as its identity, a same one-way hash function and an initial key *K* shared with BS. Figure 2 is the illustration of network initialization.

Step 1: Because that the timing of the nodes is precisely synchronized with each other, BS can judge the distance between each sensor node and itself by the timestamp and can determine how to divide the network. Then BS will make an identity list for each cluster and send them to CHs.

$$\text{BS} \rightarrow C_i\text{: } \text{E}_K[\text{ID}_{i1}, \text{ID}_{i2, \ldots,} \text{ID}_{in}] \| \text{Msg\_I} \qquad (2)$$

Step 2: When CHs received the message, it broadcasts to the sensor nodes, informing them to join the cluster. Thus clustering is finished.

$$C_i \rightarrow C_{ij}\text{: } \text{E}_K[\text{ID}_i] \| \text{Msg\_I} \qquad (3)$$

Table 1. Notation

C_i	Cluster head of the *i*th cluster
ID_i	The identity of C_i
C_{ij}	The *j*th sensor node of the *i*th cluster
ID_{ij}	The identity of C_{ij}
CK_i	The key shared between C_{ij} and C_i
BK	The master key shared between CHs and BS
y_i	The sub key of C_i
x_i	The session of C_i
$E_K [M]$	The symmetric encryption for *M* using key *K*
$D_K [M]$	The symmetric decryption for *M* using key *K*
$h()$	One-way hash function
r	Random number
Msg_I	Message of network initialization
Msg_C	Message of session key establishment in cluster
Msg_M	Message of sub key y_i establishment
Msg_R	Message of mater key *BK* reconstruction
Msg_UC	Message of session key update in cluster
Msg_UM	Message of sub key y_i update

3.2. Session Key Establishment

Though each CH has preloaded an initial key shared with BS, once the attacker captures the CH, it will reveal lots of messages if the initial key doesn't change. It is necessary to change the shared keys between CHs and BS. Meanwhile, after joining the cluster, each sensor node should establish a shared key with its CH to be safely transmitting data. The scheme is illustrated in Figure 3.

Step 1: At the beginning, BS chooses a random number and transmits it to all nodes, encrypted by initial key.

$$\text{BS}\rightarrow C_{ij}\text{: } r_1 \oplus K||h(r_1)||\text{Msg\_C} \tag{4}$$

$$\text{BS}\rightarrow C_i\text{: } r_1 \tag{5}$$

Step2:When receiving the random number, CH and nodes C_{ij} computes and validates

$$\begin{aligned} r_1' &= (r_1 \oplus K) \oplus K \\ h(r_1') &= h(r_1) \end{aligned} \tag{6}$$

the key shared between CH and C_{ij} can be generated as:

$$CK_i = h(r_1 + \text{ID}_i) \tag{7}$$

Step3: If the sensor node want to exchange the information with CH, it only needs to use to encrypt the information and transmit it to CH:

$$C_i \leftrightarrow C_{ij}\text{: } E_{CK_i}[M], D_{CK_i}[M] \tag{8}$$

3.3. Master Key Distribution and Reconstruction

Though each CH has preloaded an initial key shared with BS, once the attacker captures the CH, it will reveal lots of messages if the initial key doesn't change. It is necessary to change the shared keys between CHs and BS. Meanwhile, after joining the cluster, each sensor node should establish a shared key with its CH to be safely transmitting data. The scheme is illustrated as follows.

Step 1: BS randomly chooses t-1 integers a_1, a_2, …, a_{t-1} and constructs a (t-1)th degree polynomial $f(x)=BK+a_1x+a_2x^2+\ldots+a_{t-1}x^{t-1}\text{mod}q$.

Step 2: BS utilizes r_2 and the identities of CHs to compute y_i as the sub key of C_i, where $y_i=f(\text{h}(r_2+\text{ID}_i))$, 1≤ i≤m. m is the number of CHs. Then BS encrypts y_i by initial key K and sends it to C_i:

$$\text{BS}\rightarrow C_i\text{: } r_2 \oplus K||h(r_2)||E_K[y_i]||\text{Msg\_M} \tag{9}$$

Step 3: When received the message, C_i deletes the initial key and make use of r_2, hash function and ID$_i$ to generate x_i as its session key with other CHs,

$$\begin{aligned} r_2' &= (r_2 \oplus K) \oplus K \\ h(r_2') &= h(r_2) \\ x_i &= h(r_2 + \text{ID}_i) \\ y_i &= D_K[E_K[y_i]] \end{aligned} \tag{10}$$

Step 4, 5: Once the CH C$_i$ has received a certain amount of information from sensor nodes, it will fuse the received data and then transmits to BS. According to the (t, n) threshold secret sharing scheme, C$_i$ must get the sub keys of any other t-1 CHs so that to reconstruct the

master key. First, C_i sends BS request to reconstruct the master key. Then BS chooses t-1 CHs randomly and broadcasts to them the message of reconstruction and the identity of C_i .
Step 6: When CH C_j received the message, it uses K to encrypt the sub key y_j and transmits it to C_i , together with its identity ID_j:

$$C_j \rightarrow C_i : \mathrm{ID}_j \| E_K[x_j \| y_j] \| \mathrm{Msg\_R} \tag{11}$$

Step 7: After receiving t-1 sub keys, C_i will reconstruct the master key BK shared with BS, and can establish secure communication with BS.

$$\begin{aligned} x_j \| y_j &= D_K[E_K[x_j \| y_j]] \\ BK &= \sum_{i=1}^{t} y_i \prod_{\substack{j=1 \\ j\neq i}}^{t} \frac{x_j}{x_j - x_i} \bmod q \end{aligned} \tag{12}$$

Figure 4 is the illustration of above steps.

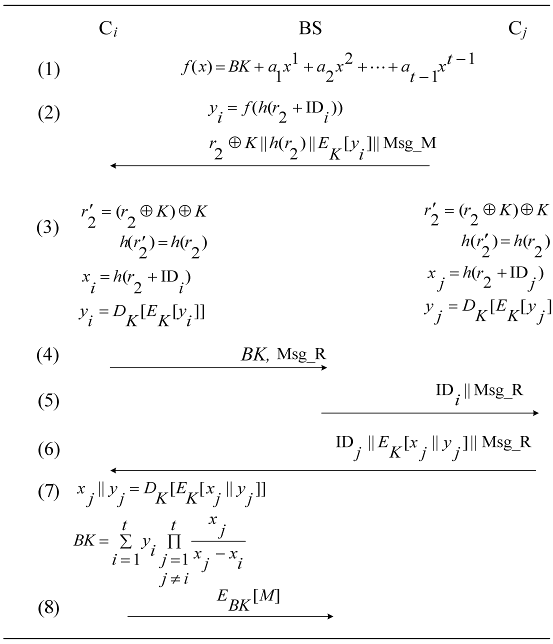
C_i
BS
C_j
(1) $f(x)=BK+a_1x^1+a_2x^2+\cdots+a_{t-1}x^{t-1}$
(2) $y_i=f(h(r_2+\mathrm{ID}_i))$
$r_2\oplus K\|h(r_2)\|E_K[y_i]\|\mathrm{Msg\_M}$
(3) $r_2'=(r_2\oplus K)\oplus K$
$h(r_2')=h(r_2)$
$x_i=h(r_2+\mathrm{ID}_i)$
$y_i=D_K[E_K[y_i]]$
$r_2'=(r_2\oplus K)\oplus K$
$h(r_2')=h(r_2)$
$x_j=h(r_2+\mathrm{ID}_j)$
$y_j=D_K[E_K[y_j]]$
(4) BK, Msg_R
(5) $\mathrm{ID}_i\|\mathrm{Msg\_R}$
(6) $\mathrm{ID}_j\|E_K[x_j\|y_j]\|\mathrm{Msg\_R}$
(7) $x_j\|y_j=D_K[E_K[x_j\|y_j]]$
$BK=\sum_{i=1}^{t} y_i \prod_{j=1, j\neq i}^{t} \frac{x_j}{x_j-x_i}$
(8) $E_{BK}[M]$

Figure 4. Master key distribution and reconstruction process

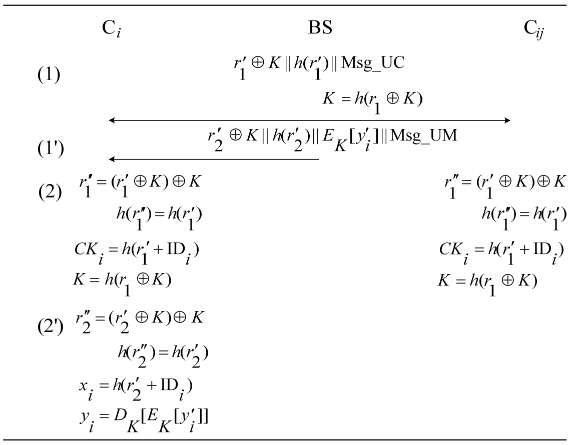
C_i
BS
C_{ij}
(1) $r_1'\oplus K\|h(r_1')\|\mathrm{Msg\_UC}$
$K=h(r_1\oplus K)$
(1') $r_2'\oplus K\|h(r_2')\|E_K[y_i']\|\mathrm{Msg\_UM}$
(2) $r_1''=(r_1'\oplus K)\oplus K$
$h(r_1'')=h(r_1')$
$CK_i=h(r_1'+\mathrm{ID}_i)$
$K=h(r_1\oplus K)$
$r_1''=(r_1'\oplus K)\oplus K$
$h(r_1'')=h(r_1')$
$CK_i=h(r_1'+\mathrm{ID}_i)$
$K=h(r_1\oplus K)$
(2') $r_2''=(r_2'\oplus K)\oplus K$
$h(r_2'')=h(r_2')$
$x_i=h(r_2'+\mathrm{ID}_i)$
$y_i=D_K[E_K[y_i']]$

Figure 5. Key update

3.4. Key Updating

Periodic *key updating*: In the dynamic key management scheme, when the network exists for a certain period, BS will update the keys so that the adversary cannot gain the current information. In our protocol, BS only needs to change the random number r_1 and r_2, and then it can update all keys stored in nodes. The key update mechanism is illustrated in Figure 5.

Node addition and deletion: When a sensor node is added in the network field, BS will transmit the current random number to it. And the sensor node can establish secure communication with its CH, only after calculating the key shared with it. One the other hand, if BS detects out that a sensor node is captured by adversaries or exhausts energy, it will inform the CH of failure sensor node. Then the CH will broadcast that the sensor node is invalid and delete its identity. Since that each sensor node has a unique key shared with the CH, and sensor node's compromise won't reveal the communication between other nodes. So there's no need to update the keys.

CH replacement: According to LEACH, a cluster head selection algorithm quoted in our protocol, nodes are randomly selected as CH and rotated so as to balance the energy dissipation in WSN. Once a new CH is selected, in order to ensure the security of network, BS must renew the random number and recalculate the sub keys for CH. It only needs to broadcast the message to the sensor nodes in its cluster and notice BS to trigger the key update mechanism to refresh the keys.

4. Security and Performance Analysis

4.1. Security Analysis

Our protocol adopts the conception of threshold secret sharing scheme, dividing the master key into several sub keys and assigning them to each CH. Each CH only stores the sub key instead of the master key used to encrypt data, so the attacker only can get the sub key. As long as the attacker captures fewer than t CHs, it cannot reconstruct the master key *BK*. Besides, the BS triggers the key update mechanism regularly, so that to refresh the key information stored in cluster heads. Even if the attacker captures t CHs of different time quantum, it also cannot reconstruct the master key.

In our scheme, since the key shared between sensor node and CH is generated by preloaded one-way hash function and its identity, each sensor node has a unique shared key with its CH. So, any sensor node's compromise only reveals it shared key and won't affect the secure communication between other sensor node and CH. On the other hand, one-way function can prevent the attacker from analyzing the information because that, for any given value h, it is computationally infeasible to find x such that $H(x)=h$, and it is computationally infeasible to find $y \neq x$ with $H(y)=H(x)$ for any given x. Moreover, the BS is provided with an intrusion detection mechanism. If any sensor node is captured or exhausts energy, BS will inform the CH and other sensor nodes not to communicate with the failure node.

Figure 6-8 shows the relationships between threshold t, key update cycle, the number of cluster heads and master key disclosure, respectively. Shown as Figure 6-8, the probability of revealing the master key has relation not only with the number of CHs but also with the threshold. On the other hand, thought the increase of t can reduce the probability of getting the master key, it also cost more time to generate the keys. So an appropriate threshold can keep a good balance between the security and the overhead. Conclusively, our scheme provides sufficient security and can achieve perfect compromise resilience.

4.2. Dynamic Analysis

The variety of network topology or the replacement of cluster heads may reveal the information stored in the nodes. To ensure the security of WSN, the BS should dynamically update the keys. In our scheme, by changing the random number r_1 and r_2, BS can update the shared keys between sensor nodes and CHs as well as the sub keys of CHs. In this way, BS has no use for reselection of all keys and polynomial.

When a new node is added to the cluster, the CH of this cluster only needs to store its identity and broadcast to other sensor nodes. Since the sensor node can generate the shared key by itself, BS has no need to update the old keys and it won't affect secure communication between CH and other sensor nodes. When a node is deleted out of the cluster, there is no need to update the key information since each sensor node has an unique key shared with the CH. BS will inform CH to delete its identity and broadcast to other sensor nodes. Therefore, our protocol can well meet the dynamic requirements of WSN.

4.3. Overhead Analysis

To recover the master key shared with BS, the CH should use the sub keys of any other t-1 CHs to reconstruct the master key. According to the Lagrange interpolation polynomial, it is easy to get the master key *BK.* So the computational complexity of the master key reconstruction is $O(t^3)$. Applied to the practice, we can chose the value of t according to the size of WSN. It can determine the computational overhead, and appropriate value of t can keep the balance between security and overheads. As to sensor nodes, the shared keys are generated by preloaded one-way function and identity and the random number r. Each sensor node only needs to compute a hash function to produce a shared key, and its computational overhead is very low.

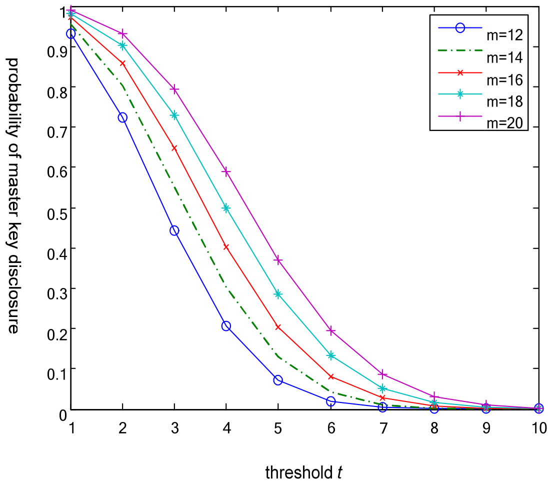

Figure 6. Relationship between threshold *t* and master key disclosure

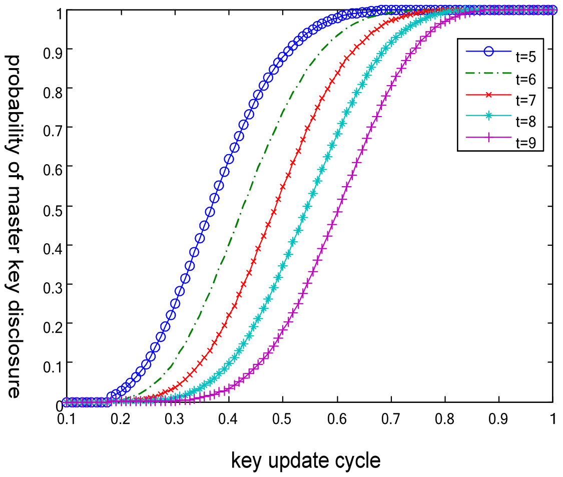

Figure 7. Relationship between key update cycle and master key disclosure

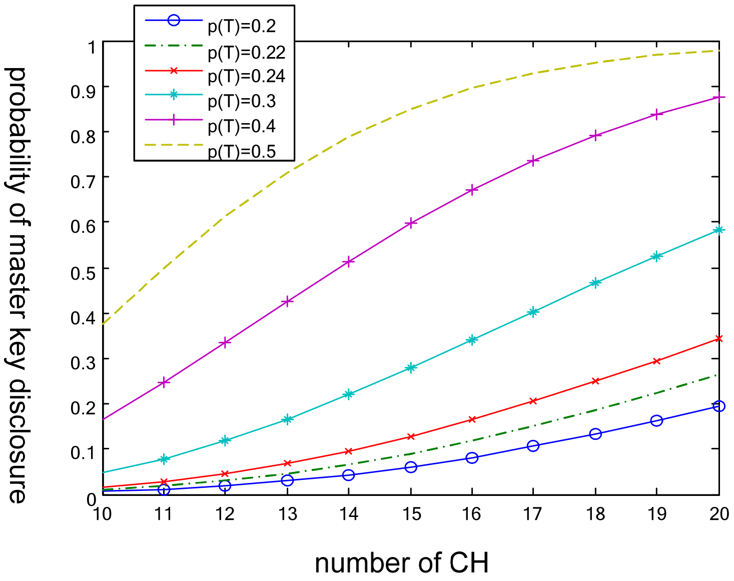

Figure 8. Relationship between the number of cluster heads and master key disclosure

In many protocols, each node must memory many keys to achieve the required connectivity and security. While in our method, each sensor node only stores its identity, preloaded one-way hash function and the initial key shared with BS. And to each CH, it is no need to store all shared keys with its sensor nodes, because that it can get the keys from their identities. In addition, due to the limitation of the node's power and memory, CHs use their session keys to encrypt the messages when they want to communicate with other CHs. They don't need to store other CHs' session keys and can figure them out by their identities and the one-way function. It not only reduces the required storage overhead, but also meets the demands of security.

Based on these reasons, our protocol provides low computational overhead and storage. Figure 9 and 10 make a comparison (Node energy consumption) with the dynamic key management protocol [11] mentioned before in the key establishment phase and key update phase. It is shown that the overhead of proposed scheme is precede that of LEAP protocol.

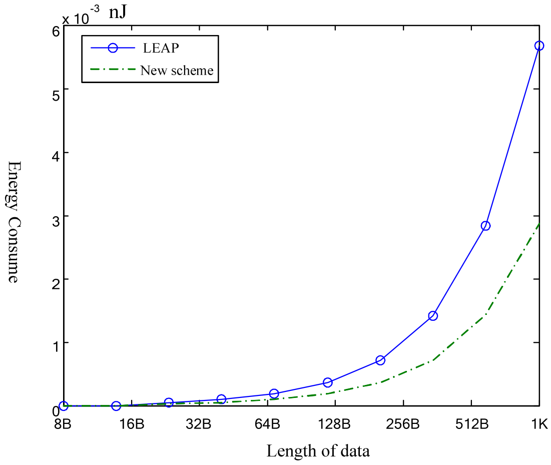

Figure 9. Node energy consumption in the key establishment phase

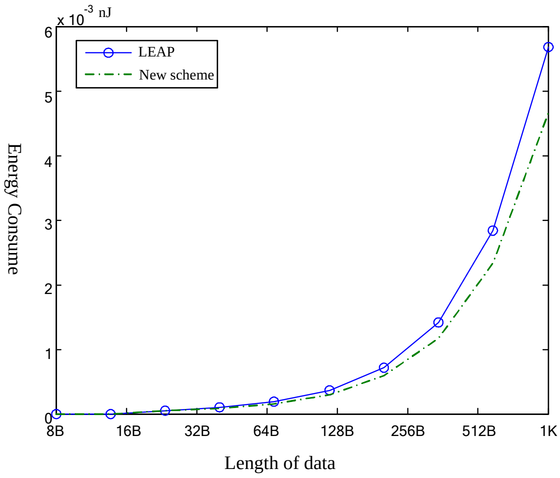

Figure 10. Node energy consumption in the key update phase

5. Conclusion

In this paper, we have proposed a secure and effective key management scheme based on dynamically clustering of WSN. Our protocol adopts the main idea of threshold secret sharing scheme, dividing the master key into several sub keys and transmitting them to each CH. By this way, the protocol provides strong security and resistance of captured attack. Meanwhile, our method combines the strengths of dynamic key management scheme, updating key information periodically. It not only ensures the security but also meets the demands of the scalability. Because that each sensor node can generate its shared key by preloaded information and it is different from each other, the compromise of any sensor node won't reveal other node's key information. It also can reduce the storage and energy cost of each node. In our further research, we are going to model the possible attacks in our scheme and simulate it.

Acknowledgments

This work was financially supported by National Natural Science Foundation of China under Grant 61201249 and Shanghai Municipal Natural Science Foundation under Grant 10ZR1401400.

References

[1] Chen XQ, Makki K, Yen K. Sensor Network Security: A Survey. *IEEE Communications Surveys & Tutorials*. 2009; 11(2): 52-73.

[2] Marcos AS, Simpllcio J, Paulo SLM. A Survey on Key Management Mechanisms for Distributed Wireless Sensor Networks. *Computer Networks.* 2010; 54: 2591-2612.

[3] Zhang JQ, Varadharajan V. Wireless Sensor Network Key Management Survey and Taxonomy. *Journal of Network and Computer Applications.* 2010; 33: 63-75.

[4] Zhou Y, Fang YG, Zhang YC. Securing Wireless Sensor Networks: A Survey. *IEEE Communications Surveys &Tutorials*. 2008; 10(3): 6-28.

[5] Kimand HS, Lee SW. Enhanced Novel Access Control Protocol over Wireless Sensor Networks. *IEEE Transactions on Consumer Electronics*. 2009; 55(2): 492-498.

[6] Eschenauer L, Gligor VD. *A Key Management Scheme for Distributed Sensor Networks*. Proceedings of the 9th ACM Conference on Computer and Communications Security. Washington. 2002; 41-47.

[7] Chan H, Perrig A, Song D. *Random Key Pre-distribution Schemes for Sensor Networks*. Proceedings of IEEE Symposium On Security and Privacy. Berkeley. 2003; 197-213.

[8] Blundo R, Suntis AD, Herzbeg A, Kutten S, Vaccaro U, Yung M. *Perfectly-secure Key Distribution for Dynamic conferences*. Proceedings of the 12th Annual International Conference on Advances in Cryptology. Berlin. 1993; LNCS 740: 471-486.

[9] Blom R. *An Optimal Class of Symmetric Key Generation Systems*. Advances in Cryptology: Proceedings of EUROCRYPT84. Paris. 1985; 335-338.

[10] Perrig A, Song D, Tygar D. *ELK: A New Protocol for Efficient Large-group Key Distribution*. Proceedings of IEEE Symposium on Security and Privacy. Oakland. 2001; 247-262.
[11] Zhu S, Setia S, Jajodia S. *LEAP: Efficient Security Mechanisms for Large-scale Distributed Sensor Networks*. Proceedings of the 10th ACM Conference on Computer and Communication Security. Washington. 2003; 62-72.
[12] Cheng Y, Agrawal DP. An Improved Key Distribution Mechanism for Large-scale Hierarchical Wireless Sensor Networks. *Ad Hoc Networks*. 2007; 5(1): 35-48.
[13] Chen CL, Li CT. Dynamic Session-Key Generation for Wireless Sensor networks. *EURASIP Journal on Wireless Communications and Networking*. 2008; 2008: 1-10.
[14] Wen M, Zhang YF, Ye WJ, Chen KF, Qin WD. A Key Management Protocol with Robust Continuing for Sensor Networks. *Computer Standards & Interfaces*. 2009; 31(4): 642-647.
[15] Shamir A. How to Share a Secret. *Communications of the ACM*. 1979; 22(11): 612-613.
[16] Heinzelman WR, Chandrakasan A, Balakrishnan H. *Energy-efficient Communication Protocol for Wireless Microsensor Networks*. Proceedings of the 33rd Annual Hawaii International Conference on System Sciences. Hawaii. 2000; 2: 33-43.