A Key Management Scheme for Clustered Wireless Senor Network in Remote Field

http://dx.doi.org/10.3991/ijoe.v9i3.2623

Chonghuan Xu
Zhejiang Gongshang University, Hangzhou, China

***Abstract*—Nowadays, wireless senor networks are widely applied in many remote fields such as disaster management and habitat monitoring. Clustered WSN have been proven to outperform their non-clustered counterparts in many aspects. Security is very crucial because these networks usually operate in the hazardous environments and the sensor nodes in these networks are resource-limited. Key management is one of the fundamental protocols to build a secure WSN. In this paper, we propose a light-weight key management scheme for clustered WSN. Our proposed scheme is based on LU decomposition which is less computation, memory, communication cost than the usual schemes. We also present performance analysis.**

***Index Terms*—WSN, Remote Field, Key Management, LU Decomposition.**

I. INTRODUCTION

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. Wireless senor networks(WSN) is one of the most promising network technologies recently, which has been widely applied in various remote fields (e.g., health, military) [1].Clustered WSN consists of many clusters. Each cluster has a cluster head (CH) and many member nodes (Cluster Member, CM) in its control. In the data transfer phase, the cluster members generally communicate only with its own cluster head, while the cluster head is responsible for data forwarding. Good Clustering algorithm can not only ensure the data communication within the original coverage, but also largely save energy of sensor nodes. The thought of cluster has many advantages. For example, the cluster head is responsible for data fusion tasks, reducing the data traffic. The topology structure of cluster is conducive to distributed algorithm, and suitable for large-scale deployment of network. Most of sensor nodes in the cluster can close the communication module for quite a long time (extending the nodes' sleep time), and does not participate in the data forwarding process(reducing the volume of inter-node communication), so clustering can minimize the energy usage of individual sensor nodes, and therefore significantly extend the survival time of the entire network.

However, there still exist a lot of unresolved issues in wireless sensor networks, of which security is one of the hottest topics [2]. Security is very crucial as these networks usually operate in the hazardous environments in which there are more attacks. Authentication between nodes and key management are the fundamental protocols to build a secure WSN [3]. In wireless sensor network, the large amount of sensor nodes and the high frequency of key operation, such as key generation, key distribution and key update, make key management more difficult. Without a proper key management scheme, it is difficult to guarantee the safe communication between sensor nodes and the security of the whole networks. The sensor nodes in WSN are all resource-limited devices, so key management schemes like Diffie-Hellman or public-key based schemes widely applied in other networks are not very suitable for using directly in wireless sensor networks [4]. Therefore, key management scheme based on symmetric encryption technology is still a hot research topic.

In this paper, we propose a low-overhead but efficient key management scheme for clustered wireless sensor networks. The purpose of this paper is to ensure less computation, storage, communication cost without losing the security of WSN. Our scheme utilizes LU decomposition adopted from literatures [5][6] and effectively improves the security and scalability of the network.

The rest of this paper is organized as follows. Section II describes the related work. Section III describes the proposed scheme. Section IV presents performance analysis. Section V concludes the paper.

II. RELATED WORK

Perrig *et al* . [7] presented a key management scheme called SPINS. It is a base station (BS) participated scheme in which each sensor node has a share key with the BS, and the BS sends the pairwise key for two sensors encrypted with the share key. This scheme has perfect resilience but does not have good scalability. The schemes based on LU decomposition in literatures [5] and [6], significantly improves the network-wide memory usage and security, guarantees that any pair of neighboring nodes can directly derive a common secret key between themselves, and are substantially more storage efficient and flexible.

The topological structure of cluster, can improve the scalability of the network, also be suitable for large scale networks, so it becomes one of the key technology in WSN research [8]. Reza *et al.* [9] proposed a key management scheme for clustered WSN. The key management scheme for clustered WSN is based on elliptic curve cryptography and symmetric key system. In this scheme, however, the capture of the gateway node will affect the security of the entire cluster. Rahman *et al.* [10] presented a hybrid, robust, and non-interactive protocol for key management for WSNs. It scales very well when network

size increases without increasing memory overhead. It is the first protocol in WSNs which combines the pair-wise symmetric key based solution with IBC based solution and brings the good of both.

Although ECC (Elliptic Curve Cryptography) was proven to be able to apply to sensor in the literature [11], symmetric encryption technology is still a research focus for its simplicity, high efficiency and low energy consumption. Ma *et al.* [12] proposed an efficient key management mechanism which divides key management into two groups: in-group and cross-group by the adoption of the node-deployment information. The authors develop an optimization method for the nodes communication between groups. The theoretical analysis and simulation show that the proposed key management mechanism has a great improvement in network connectivity, security and memory utilization.

III. PROPOSED SCHEME

In this section, we state our scheme. At first, we briefly introduce some terms in our scheme and then show the detail of our scheme, including the formation and decomposition of the matrix *A*, the distribution of the key materials, the generation of share key for CH-CMs and session key for CM-CM, and key update.

A. LU Decomposition

LU decomposition [9] is procedure which can decompose a square matrix *A* (*N* x *N*) into two matrices, a lower triangular matrix *L* and an upper triangular matrix *U*, such that,

$$A=LU \tag{1}$$

Where *L* and *U* meet the following requirements:

$$L_{ij}=\begin{cases} l_{ij} & for\ i \geq j \\ 0 & for\ i < j \end{cases} \qquad U_{ij}=\begin{cases} u_{ij} & for\ i \leq j \\ 0 & for\ i > j \end{cases} \tag{2}$$

A 3 x 3 equation (1) is as follow:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix} * \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix} \tag{3}$$

Here, we use elementary row operations which include elimination, scaling, and interchange, to change *A* to *U*. We define the elementary matrices corresponding to the row operations E_1, E_2, E_3,…, E_k, as row operation, *E* is on the left hand of *A*. so we can see:

$$E_1*E_2*E_3*...*E_k*A=U \tag{4}$$

Hence,

$$L=(E_1*E_2*E_3*...*E_k)^{-1}. \tag{5}$$

So, we can get both *U* and *L*.

B. The Content Stored in Nodes

There is a 3-column list in every cluster head CH which stores some information about cluster members(CMs) in its control. Table 1 shows the detail of the list (the parameters derive from the following example):

TABLE I.
THE CONTENT STORED IN CH

Name	$L_r(i)$	***Flag***
S_x	[1]	1
S_y	[1 2]	1
…	…	…

The first column is the name of CMs. As we can see from table 1, the name of the first two CMs are S_x and S_y. The second column is the $L_r(i)$ of each CM which can be used to calculate the next share key for the communication between CH and CM. Cluster head also stores the *Flag* of every CM which shows whether a CM has been revoked or not. Here, 1 stands for not being revoked.

Each CM obtains a row $L_r(i)$ from *L* and a column $U_c(i)$ from *U*. Moreover, each one also stores the $U_c(i)$ of revoked nodes. And after the materials update, these information will be emptied.

C. Detail of The Proposed Scheme

Our scheme consists of the formation and distribution of key material, key generation and key update.

The formation of key material: For a cluster which has *N* member node, it needs to generate (*n**(*n*-1))/2 random positive integer as the session key (k_1,k_2,…,$k_{(n*(n-1))/2}$). Together with the next share key for CH and each CM (sk_1, sk_2,…, sk_n), (sk_n is also a random positive integer), we can construct the matrix *A*. The form of matrix *A* looks like:

$$\begin{pmatrix} sk_1 & k_1 & \text{L} & k_{n-2} & k_{n-1} \\ k_1 & sk_2 & \text{L} & k_{2n-2} & k_{2n-3} \\ \text{M} & \text{M} & \text{O} & \text{M} & \text{M} \\ k_{n-2} & k_{2n-2} & \text{L} & sk_{n-1} & k_{\frac{n(n-1)}{2}} \\ k_{n-1} & k_{2n-3} & \text{L} & k_{\frac{n(n-1)}{2}} & sk_n \end{pmatrix}$$

Here, we take a cluster of four CMs as an example. At first, we generate $(n(n-1))/2+n=(4\times3)/2+4=10$ key, we assume that (k_1,k_2,…,k_6) are 2,3,4,6,7,9, respectively; and the share key for CH and each CM are 1,5,8,10, respectively. So we can get the 4 x 4 matrix *A* as follow:

$$A=\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 5 & 6 & 7 \\ 3 & 6 & 8 & 9 \\ 4 & 7 & 9 & 10 \end{pmatrix}$$

Then we decompose the matrix *A* by (4), (5) and get *L* and *U*. *L* and *U* matrices are as follows:

$$L=\begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 \\ 4 & -1 & 3 & 1 \end{pmatrix}$$

and

$$U = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & -1 & -3 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

The distribution of key material: The order i of CMs in the list of the CH determines what CMs get, $L_r(i)$ and $U_c(i)$. For example, in the table 1, S_x ranks first, so S_x gets the first row in L and first column in U; S_y ranks second, and gets the second row in L and second column in U. Here what S_x and S_y get as follows:

S_x: $L_r(1)$=[1 0 0 0] and $U_c(1)$=[1 0 0 0]
S_y: $L_r(2)$=[2 1 0 0] and $U_c(2)$=[2 1 0 0]

As we can see that the last three number of $L_r(1)$, $U_c(1)$ and the last two number of $L_r(2)$, $U_c(2)$ are 0, and can be deleted when distributed and stored. Hence what stores in S_x and S_y are:

S_x:$L_r(1)$=[1] and $U_c(1)$=[1]
S_y:$L_r(2)$=[2 1] and $U_c(2)$=[2 1]

So it has less memory cost which also leads to less communication cost.

And what the cluster head gets from the LU decomposition can be seen at the second column of the table 1.

Key generation: During network initialization, each CM will get a share key with the CH, as well as the CH. CH uses this share key to send the key materials, and the next share key can be calculated by these key materials. As we can see from the matrix A, sk_i is the result of $L_r(i)$* $U_c(i)$, so CM can directly get the next share key sk_i by calculating its $L_r(i)$*$U_c(i)$. For CH, it only has the $L_r(i)$ of each CM, so it needs the $U_c(i)$ sent by each CM, and then it can calculate the next share key as CM does.

Before two CMs calculate their key materials to produce the session key, they need to check whether the other one has been revoked by looking up whether the received $U_c(i)$ has already been in its memory. If $U_c(i)$ is in store, it indicates that the node has been revoked by CH or is not a legal node, so there is no need doing the key negotiation. If not, then do the following steps.

Let's start with the general. Assume that CM with id S_x contains [$L_r(i)$, $U_c(i)$] and with id S_y contains [$L_r(j)$, $U_c(j)$]. When S_x and S_y need to find a common secret key between them for communication, they first exchange their columns, and then compute as follows:

$$S_x:L_r(i)*U_c(j)=A_{ij} \quad (6)$$
$$S_y:L_r(j)*U_c(i)=A_{ji} \quad (7)$$

As A is a symmetric matrix, $A_{ij}=A_{ji}$. A_{ij} (or A_{ji}) is then used as a common key between S_x and S_y.

The following three steps show the detail of our scheme for CM to CM.

Step 1. S_x -> S_y: S_x, S_y, $U_c(i)$

In this step, S_x sends [S_x, S_y, $U_c(i)$] to S_y. S_y knows the sender is S_x and itself is the receiver. Then S_y checks whether $U_c(i)$ has stored in its memory. If not existing, it means that S_y is a legal CM and is worthy of communication. So S_y generates the zeros of the received $U_c(i)$ and its own $L_r(j)$. Then S_y can calculate the session key A_{ji} by (7).

Step 2. S_x <- S_y: S_y, S_x, $U_c(j)$, N_y, $A_{ji}(N_y)$

In this step, S_y send [S_y, S_x, $U_c(i)$, N_y, $A_{ji}(N_y)$](N_y is a random number) to S_x. S_x gets this message, checks whether $U_c(j)$ has stored in its memory. If not existing, it means that S_y is a legal CM and is worthy of communication. Then, S_x generates the zeros of the received $U_c(j)$ and its own $L_r(i)$. So S_x can calculate the session key A_{ij} by (6). Then S_x uses A_{ij} to decrypt $A_{ji}(N_y)$, and if the result is equal to N_y, it means that S_y is the one who has the same session key with it. So S_x knows that S_y is the one it wants to communicate.

Step 3. S_x -> S_y: S_x, S_y, N_x, $A_{ij}(N_x)$

In this step, S_x sends [S_x, S_y, N_x, $A_{ij}(N_x)$] (N_x is a random number, too)to S_y. S_y uses its A_{ji} to decrypt $A_{ij}(N_x)$, and if the result is equal to N_x, it means that S_x is the one who has the same session key with it. So S_y knows that S_x is the one it needs to communicate.

So after these three steps, both sides get their session key for communication.

Key update: If we always use the same materials for key management between CMs in the same cluster, or between CH and CM, it will be unsafe when the time goes. So we define an expiration time T. After the CH sends materials to the CMs in its control, it sets an expiration time T. When T expires, the CH regenerates (n*(n-1))/2+n random positive integer, it constructs a new matrix A, then decomposes the new matrix A, and sends materials to the corresponding CMs.

If there are some CMs being about to use up its energy, these CMs will report this information to CH. This communication is secured by the share key. Then CH modifies the third column of these CMs to 0 and then send the $U_c(i)$ of these CMs to all its other CMs. And its other CMs store these $U_c(i)$ which can be used to verify whether a CM is worthy of communication or not. Moreover, CH will count the number of the revoked CMs, if the number is larger than a threshold which is set during network initialization, CH allows new nodes to add into this cluster and regenerates key materials. The set of threshold in our scheme reduces the frequency of key update.

After CH uses the first share key, which is set when the sensor nodes first add into the network, to send the key materials, both the CH and CM need to update their share key. The process of share key(between CH and CM) update has been described as CM -> CH: $U_c(i)$, N_{cm}, $sk_i(N_{cm})$.

CM can directly calculate the next share key sk_i using its own $L_r(i)$, $U_c(i)$ with the next equation:

$$Sk_i=L_r(i)*U_c(i). \quad (8)$$

Then CM sends [*Name*, $U_c(i)$, N_{cm}, $sk_i(N_{cm})$] to CH(N_{cm} is a random number). CH gets the message and knows the name of that CM. Then CH uses the received $U_c(i)$ and the corresponding $L_r(i)$ stored in its list, and calculates sk_i as (8). Therefore, CH can decrypt $sk_i(N_{cm})$, and if the result is equal to N_{cm}, it means that the share key update completes correctly. So at the next time, CH will use this share key to send key materials.

IV. PERFORMANCE ANALYSIS

In this section, we discuss the performance of our scheme from the following four perspectives: storage analysis, computation analysis, security analysis, connectivity analysis. Sometimes we have to make a trade-off as these factors are dependent on each other.

A. Storage Analysis

The CH has to store three items for each CM, namely, *Name*, $L_r(i)$, *Flag*. So the total storage of CH is n**Length*(*Name*|$L_r(i)$|*Flag*). Here, n is the number of CMs in its control and function *Length*() is used to get the length of a string. The storage of CM will change when the time goes. At first, CM only needs to store $L_r(i)$ and $U_c(i)$, but when the number revoked or authentication-failed CMs increase, each CM has to store M*$U_c(i)$ more content. Here, M is the number of revoked or authentication-failed CMs. After once key update, CH reverses the table 1. So the nth CM will become the 1st CM and this reversal is conducive to the average energy consumption of all the CMs.

B. Computation Analysis

The cluster head is responsible for the main computation cost, including the generation and decomposition of the symmetric matrix A, the 0 deletion and regeneration operation of $L_r(i)$ and $U_c(i)$, as well as the calculation of the share key. CMs need to do the 0 deletion and regeneration operation of $L_r(i)$, $U_c(i)$, $L_r(j)$, $U_c(j)$, as well as the calculation of the share key. The computational overhead of our proposed scheme is very small compared with the schemes which use public key technology.

C. Security Analysis

If the CH is captured, the adversary can't know the next share key which is obtained from $L_r(i)$*$U_c(i)$, because CH only stores the $L_r(i)$ of each CM. As long as the base station is able to detect the captured CH, it can generate new cluster head by the cluster algorithm(the cluster algorithm is beyond the scope of this article) for that cluster.

If a CM is captured, it also needs communication with other CM to get the session key. So once CH find a CM has been captured, it can send the $U_c(i)$ of the captured CM to other CMs.

D. Connectivity Analysis

We generate $(n(n\text{-}1))/2$ session key for n CMs, so any two CMs in the same cluster can derive a session key and can communicate with each other without the help of a third one. And we generate n share key for CH and CMs, so each pair of CH and CM has a share key. Hence, the connectivity of our proposed scheme can reach 100%.

V. CONCLUSION

In this paper, we propose a key management scheme for clustered WSN which is less computation, memory and communication cost, and also improves the security without losing scalability. Our scheme utilizes LU decomposition to produce the key material, so the computational overhead of our proposed scheme is very small compared with the schemes which use public key technology. As the CH has more resource, it is responsible for the main computation cost. So CMs can work for longer time, and significantly extend the survival time of the entire network. We generate $(n(n\text{-}1))/2$ keys for n CMs, so it guarantees that any pair of CMs can directly find a common secret key, and we generate n share key for CH and CMs, so each pair of CH and CM has a share key. Besides, the procedure of deleting and regenerating zeros in $L_r(i)$ and $U_c(i)$ is also more storage efficient which also saves the network-wide memory and traffic. The set of threshold which determines whether new sensor nodes can be added into the cluster, reduces the frequency of key update which also saves the energy of sensor nodes. The $U_c(i)$ stored in every CM makes our scheme more secure. Take advantage of cluster, our scheme is also more scalable. A suitable cluster algorithm for the proposed scheme will be considered in our future research.

ACKNOWLEDGMENT

This work was supported in part by NSFC of China under Grant No. 71071140, 60905026 and 71071141, Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20103326110001, Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20093326120004), Zhejiang Science and Technology Plan Project (No. 2010C33016, 2012R10041-09), and the Key Technology Innovation Team Building Program of Zhejiang Province (No. 2010R50041) as well as Zhejiang Provincial Natural Science Foundation of China under Grant No. Z1091224, LQ12G01007 and Y6110628.

REFERENCES

[1] Akyildiz I F, Weilian S, Yogesh S, and Erdal C., "A Survey on Sensor Networks," IEEE Communications Magazine., Vol. 40, pp. 102-114, 2002. http://dx.doi.org/10.1109/MCOM.2002.1024422

[2] G. Padmavathi, Dr. and M. D. Shanmugapriya, "A Survey of Attacks, Security Mechanisms and Challenges in Wireless Sensor Networks," International Journal of Computer Science and Information Security. Vol. 4, August. 2009.

[3] PANG Liao-jun, JIAO Li-cheng, WANG Yu-mi., "Authentication and Key Negotiation Protocol Between Two Nodes of Wireless Sensor Networks," CHINESE JOURNAL OF SENSORS AND ACTUATORS. Vol. 21, p. 1422-1426, 2008.4.

[4] Liu, D., Ning, P., and Li, R., "Establishing Pairwise Keys in Distributed Sensor Networks," ACM Transactions on Information and System Security. Vol. 8, pp. 41-77, Feb. 2005. http://dx.doi.org/10.1145/1053283.1053287

[5] Choi, S. and Youn, H., "An Efficient Key Pre-distribution Scheme for Secure Distributed Sensor Networks," EUC Workshops 2005, IFIP International Federation for Information Processing, LNCS. pp. 1088-1097, 2005.

[6] Al-Sakib Khan Pathan, Tran Thanh Dai, Choong Seon Hong, "A Key Management Scheme with Encoding and Improved Security for Wireless Sensor Networks," Distributed Computing and Internet Technology Lecture Notes in Computer Science. V. 4317, pp. 102-115, 2006.

[7] Perrig A, Szewczyk R, Wen V, Cullar D, Tygar JD, "SPINS: security protocols for sensor networks," Wireless Networks. Vol. 8, pp. 521 – 534, Sep. 2002. http://dx.doi.org/10.1023/A:1016598314198

[8] Younis O., Kmnz M., Ramasubramanian S, "Node clustering in wireless sensor Networks: Recent Developments and deployment challenges," IEEE Network. Vol. 20, pp. 20-25, May~June. 2006.

[9] Reza A, Arash R-M, Zine-Eddine A, "A Key Management Scheme for Cluster Based Wireless Sensor Networks," Processings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, DC:IEEE Computer Society. pp. 222-227, 2008.

[10] Musfiq Rahman, Srinivas Sampalli, "A Hybrid Key Management Protocol for Wireless Sensor Networks," 2012 IEEE 11th Interna-

tional Conference on Trust, Security and Privacy in Computing and Communications. pp. 769-776, 2012.

[11] P. Szczechowiak, L. Oliveira, M. Scott, M. Collier, and R. Dahab, "Nanoecc: Testing the limits of elliptic curve cryptography in sensor networks," Wireless Sensor Networks Lecture Notes in Computer Science. Vol. 4913, pp 305-320, 2008,

[12] Xiaofei Ma, Zhuoya Dong, Jie Li, "A Novel Key Management Scheme for Wireless Sensor Networks," Internet Computing for Science and Engineering (ICICSE), 2012 Sixth International Conference on. pp. 198-203, 2012.

AUTHOR

C. H. Xu Author is with the Business Administration College and the Center of Studies of Modern Business of Zhejiang Gongshang University, Hangzhou, China, (e-mail: talentxch@ mail.zjgsu.edu.cn).

Received 24 March 2013. Published as resubmitted by the author 12 June 2013.