A Loop-Based Key Management Scheme for Wireless Sensor Networks

YingZhi Zeng[1], BaoKang Zhao[1,2], JinShu Su[1], Xia Yan[1,3], and Zili Shao[2]

[1] School of computer, National University of Defense Technology, ChangSha Hunan, China
[2] Department of Computing, The Hong Kong polytechnic University, Hong Kong
[3] School of Computer and Communication, Hu'nan University, ChangSha Hunan, China
zyz1234@gmail.com, sjs@nudt.edu.cn,
sunofxy@hotmail.com,{csbzhao,cszlshao}@comp.polyu.edu.hk

Abstract. Wireless sensor networks are emerging as a promising solution for various types of futuristic applications for both military and the public. The design of key management schemes is one of the most important aspects and basic research field of secure wireless sensor networks. Efficient key management could guarantee authenticity and confidentiality of the data exchanged among the nodes in the network. In this paper, we propose a new key management scheme based on loop topology. Comparing with cluster-based key management schemes, loop-based scheme is proved to be more efficient, cost-saving and safe.

1 Introduction

Recent advancements in wireless communications and micro electromechanical technologies have promoted the development and applications of wireless sensor networks (WSN). WSN increasingly become viable solutions to many challenging problems for both military and the public applications, including battlefield surveillance, border control, target tracking and infrastructure protection.

In a WSN, sensor nodes are typically deployed in adversarial environments such as military applications where a large number of sensors may be dropped from airplanes. Sensor nodes need to communicate with each other for data processing and routing. Secure communication between a pair of sensor nodes requires authentication, privacy and integrity. However, the wireless connectivity, the absence of physical protection, the close interaction between sensor nodes and their physical environment, and the unattended deployment of sensor nodes make them highly vulnerable to node capture as well as a wide range of network-level attacks. Moreover, the constrained energy, memory, and computational capabilities of the employed sensor nodes limit the adoption of security solutions designed for traditional networks.

As a successful security mechanism of wired networks, key management is crucial to the secure operation of sensor networks. A large number of keys need to be managed in order to encrypt and authenticate all sensitive data exchanged. The characteristics of sensor nodes and WSNs render most existing key management solutions developed for other networks infeasible. To provide security in such a distribution environment, the

M. Denko et al. (Eds.): EUC Workshops 2007, LNCS 4809, pp. 103–114, 2007.

well-developed public key cryptographic methods have been considered at first, but these demand excessive computation and storage from the resource extra-limited sensor nodes [1]. The symmetric key cryptography is considered as the only feasible way for wireless sensor networks. Therefore, there must be a secret key shared between a pair of communicating sensor nodes. Sensor nodes can use pre-distributed keys directly, or use keying materials to dynamically generate pair-wise keys.

Since the network topology is unknown prior to deployment, a key pre-distribution scheme is required where keys are stored in ROMs of sensor nodes before the deployment. The stored keys must be carefully selected so to increase probability that two neighboring sensor nodes, which are within each other's wireless communication range, have at least one key in common. Those nodes which have no shared keys may setup secure communicate through the help of neighboring nodes. After the deployment, each sensor node should connect with its neighboring nodes and generate their security keys in a self-organized method. After Key generation, next important step is distributing the keys to relative nodes.

The main contribution of this work is to shed some light on the basic framework of the key management scheme of WSN. Loop-based scheme includes key material pre-distribution, key generation, key distribution and rekeying. In particular, we bring in a novel loop-based topology for key management. To the best of our knowledge, this paper is the first one to apply loop topology to key management scheme in distributed wireless sensor networks. Our analysis and comparison indicate that this approach has substantial advantages over the traditional cluster-topology scheme.

The remainder of the paper is organized as follows. Section 2 provides an overview of the related works. The loop-based key management scheme is introduced in section 3. Section 4 deals with the detailed performance analysis and comparisons. We conclude in Section 5 and point out some future research directions.

2 Related Works

A number of key management schemes have been developed for sensor networks in the recent years. In this section, we review the major existing key management schemes in wireless sensor networks.

Eschenauer and Gligor [2] proposed a random key pre-distribution scheme. Each sensor node is assigned k keys out of a large pool P of keys in the pre-deployment phase. Neighboring nodes may establish a secure link only if they share at least one key, which is provided with a certain probability based on the selection of k and P. A major advantage of this scheme is the exclusion of the base station in key management. However, successive node captures enable the attacker to reveal network keys and use them to attack other nodes. Based on the EG scheme, q-composite keys scheme was proposed by Chan in [3]. The difference between this scheme and the EG scheme is that q common keys (q >1), instead of just a single one, are needed to establish secure communication between a pair of nodes. Using the framework of pre-distributing a random set of keys to each node, Chan presented two other mechanisms for key management. The first mechanism is a multi-path key reinforcement scheme, applied in conjunction with the basic scheme to yield improved resilience against node capture attacks. The main attractive feature of this scheme is that it can enhance the security of an established link key by establishing the link key through multiple paths. The second mechanism is a random pair-wise keys scheme. The purpose of this scheme is to allow node-to-node authentication between communicating nodes.

Liu and Ning [4] provided further enhancement by using t-degree bivariate key polynomials. Since an attacker needs to capture at least t+1 nodes to obtain any t-degree polynomial, this solution was shown to significantly enhance network resilience to node capture as long as the number of captured nodes is below a certain threshold. However, if the number of captured nodes exceeds this threshold, the network is almost entirely captured by the attacker.

Du et al. [5] proposed a method to improve the basic scheme by exploiting a priori deployment knowledge. They also proposed a pair-wise key pre-distribution scheme for wireless sensor networks [6], which uses Blom's key generation scheme [7] and basic scheme as the building blocks.

Choi and Youn [8] proposed a key pre-distribution scheme guaranteeing that any pair of nodes can find a common secret key between themselves by using the keys assigned by LU decomposition of a symmetric matrix of a pool of keys.

3 Loop-Based Key Management Scheme

Existing approaches in key management scheme mainly inefficiently utilize the cluster topology information. In fact, the loop-based topology has many special benefits in WSN. We present a new key management scheme based on the loop topology. To our knowledge, this is the first paper in this area that combines the node topology with key management.

3.1 Basic Definitions

In Graph Theory, a loop is a non-directional path, which begins and ends with the same node. Since there is at most one connection between every two nodes in an undirected graph G=(V, E) [9], a path from v_i to v_j representing a wireless sensor network link can be defined as a sequence of vertices $\{v_i, v_{i+1}, \ldots, v_j\}$, where V representing the set of nodes and E is the set of connections.

Loop length: The length of a loop also can be called path length, is the number of hops from v_i to v_j. Let L be a loop. It is obviously that if length (L)<3, either the node on L is isolated or L is a round trip between two nodes.

Loop type: In a large scale WSN, there may be some isolated nodes. A loop with only two nodes is also a special loop. For example, in Fig.1, L2 and L3 are typical loops and L1 is a two-nodes special loop. In the following parts, nodes on the loops with greater length than 2 are called on-loop nodes. Let L be the set of the loops that node v is on. If max (len (l) $\leq$ 2) (for every l in L), we say v is non-on-loop node.

3.2 The Loop-Based Topology

Unlike traditional wired networks, WSN is a data-center network. Its core function is to aggregate data and to forward data through the route nodes to the sink. In our key management scheme, we consider the key management topology and the data process topology should not be separated.

Old key management schemes are mainly based on cluster topology. Under the assumption that a sensor node either acts as a data producer or is just a router, every node should take part in a voting to choose some nodes acting as cluster headers. After the deployment of nodes and the CH's voting, the cluster headers play an important role in the next steps which include initializing keys, distributing group keys and rekeying. There are two kinds of working flows in cluster-based key management schemes. Key management flow is under the control of those cluster headers. Data aggregating flows are processed between nodes doing sensor works.

In this paper we take loop as the basic unit and the entire network is grouped into inter-connected loops in self-organized mode. Within a loop, nodes can exchange information with each other by forwarding messages along the loop in either of the two directions. For inter-loop communications, messages are first routed to the gateways nodes (router nodes joining multiple loops) and transferred from gateway to gateway till reach the destination. As for inner Loop transmission, messages are finally forwarded to the destination.

Loop topology has many special benefits in WSN:

(1) The loop topology is relative to the physical positions of those nodes directly. When a node within the loop receives an order to sense some special information, the node becomes an information aggregator immediately. Every neighboring node gets some sensor data and sends it to the aggregator. The aggregator will compare and integrate it with its own report. The result would be shortened before it is sent to the next hop. Hop by hop, the sensor data will be shortened and be aggregated many times until it arrives at the sink node. (2) There are no critical header nodes defined in a loop, so the network topology never suffers from chain change caused by the re-election of headers. The scenario of a group without leader will never happen in a loop-based WSN. (3) Local loop information can be reserved in every node on the loop. The topology information redundancy enhances the network robustness. (4) One of the features of a loop that there are two paths between every two nodes on the same loop provides a backup route for link failure during message transmission.

3.3 Creation of a Loop Topology for Key Management

1、(Key material pre-distribution phase) Before the deployment, every node should be assigned some key materials, including a unique ID, a private key (only known by the key server and node itself), a Hash function and a global key. After deployment, every node will start broadcasting its ID message encrypted by the global key. This action can prevent malice listening during the initialization phase of key management.

2、Every node which receives a message can build up its neighbor table.

3、**Condition 1** for Loop formation: After checking their neighbors' information, those nodes with only one neighbor will start the second round broadcasting, such as node A in Figure-1. The information of their neighbor table (NT) is broadcasted. Neighboring nodes received NT messages will add the neighbor information into their link table (LT) and broadcast the latest LT messages to neighboring nodes.

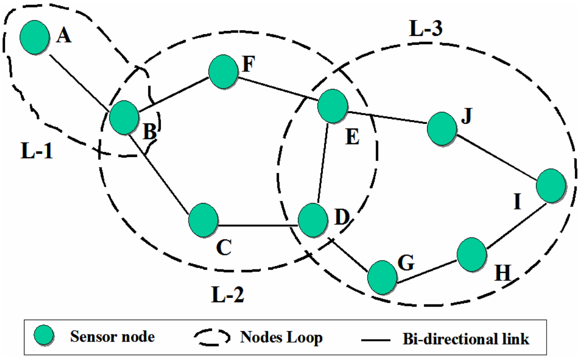
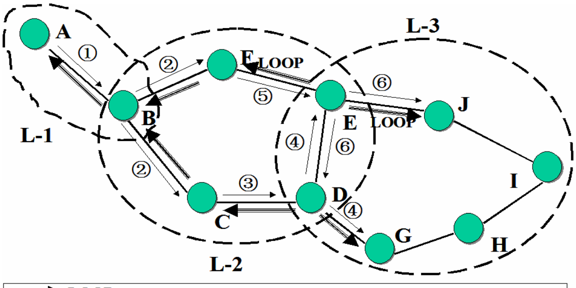


Fig. 1. An example for loop-based wireless sensor networks

If the sensor nodes are deployed close enough then none of them has only one neighbor. C**ondition 2** for Loop formation should be taken into consideration. Timing is the first key point. At time T1 after the deployment, one-neighbor node can start sending message**.** If none of the nodes has only one neighbor, those nodes with at least M neighbors(M>=3) can start broadcasting their NT at time T1+nT (Unit time T equals to the time a node broadcast would need). If n=5 in Figure-1, then node I will start sending its NT message. Table-2 lists those messages (including messages sender, receiver and contents) passed among some nodes in Figure-1. The message processing details and sequence are shown in Table-1 and Figure-2.

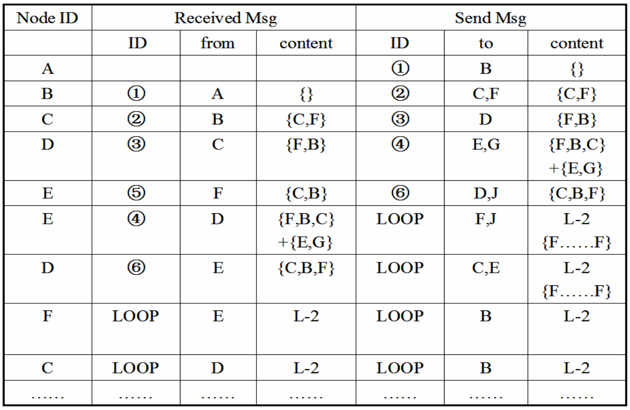


* T1 time after deployment, A starts sending message ①;
* if none of the node has only one neighbor, after T1+nT time,nodes with the largest neigbors can broadcast his Ntable. If n=5 then node I will start sending its NT ;

Fig. 2. An example of a loop's creation

4、Forming loop: After several units of times nT, some nodes, such as B in Figure-1, may receive two loop messages from neighbors. Within the node sequence that a node can find a multiple-hop path to connect itself, a loop of those nodes can be formed by the conjunction of loop messages. Thus the whole sensor networks can be divided into many loops, among them are some special loops. Two loops may share two and even more common nodes, such as L-2 and L-3 in Figure-1.

Table 1. Loop creation Messages



Node ID	Received Msg			Send Msg		
	ID	from	content	ID	to	content
A				①	B	{}
B	①	A	{}	②	C,F	{C,F}
C	②	B	{C,F}	③	D	{F,B}

D	③	C	{F,B}	④	E,G	{F,B,C}
						+{E,G}

E	⑤	F	{C,B}	⑥	D,J	{C,B,F}

E	④	D	{F,B,C}
			+{E,G}
				LOOP	F,J	L-2
						{F……F}

D	⑥	E	{C,B,F}	LOOP	C,E	L-2 {F……F}
F	LOOP	E	L-2	LOOP	B	L-2
C	LOOP	D	L-2	LOOP	B	L-2
……	……	……	……	……	……	……

5、Special loop format: A single-link node, such as node A in Figure-1, has only one link with a neighbor node. Those two nodes (A and B) form a special loop L-1. Only when a node receive a message {} come from his neighbor node can this kind of special loop be created. Through step 1 to 4, another loop L-3 can be formed by node E, D, G, H, I and J. It is obvious that two nodes (D and E) are shared in loop L-2 and L-3. This type of loop format is determined by the loop size and the node position.

3.4 The Loop-Based Key Management Scheme (LBKMS)

As described in section 3.3, the first stage of LBKMS is to form loops through step 1 to 5. All the nodes of a WSN are divided into different loops or shared between neighbor loops.

Based on the loop topology, this paper develops a new key concept: loop-key. Upon loop information (every node get its neighbor table and link table and loop sequence), the loop-creator node can set up a new loop-key for those nodes in the loop. The computing formula of loop-key is:

Loop-key=Hash (time stamp || private key || loop-creator node ID || some loop members' ID). (1)

Time stamp is introduced into above formula to prevent replay attack that comes from neighboring nodes. The private key is a proprietary key of loop-creator. It is also the creator's privilege that how many loop members' IDs are used in the hash function. For example of Figure-1, the loop-key may be equal to hash (T_s|| K_B|| B|| C|| D|| E).

This formula is based on the preloaded material on each sensor node, using time stamp and other loop nodes' ID can guarantee the production-loop key be safe.

In the third stage, loop-creator will send the loop-key encrypted with the global key to its loop members through the loop routing. If the loop format is not special, the key messages will be sent to its two loop-neighbors at first. Every node on the loop will send the key message to next node on the neighbor table until some node receives the same message twice.

After the above three stages, every node in WSN should belong to a loop group and should keep a loop-key shared with other loop members. Sensor data aggregation and communication within the loop should be encrypted using the loop key.

The loop-based rekey: Well known as a resource-limited network, a WSN cannot afford changing loop-keys continually. But there are still two scenarios in which rekeying is sometimes needed. In the **first scenario**: If a loop member is recognized as a defection node, or the sink sends a command to clean some node, the urgent affair is to kick it out of the loop member list. First of all, such an abnormal message arrives at the closest loop member. The node will send a cleaning message to its two loop neighboring nodes (if the defection node has just one direct neighbor, then just one cleaning message is enough.). As is shown in Figure-3, cleaning message should be sent to every node on the loop except the defection node. After that, the first leader node will start sending rekeying message to replace the old loop-key.

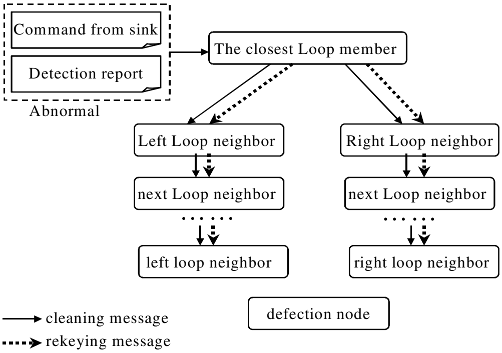


Fig. 3. Loop-based rekeying in WSN (1)

Compared with first scenario, the **second scenario** deals with normal rekeying. If a loop member is out of battery and can-not work properly any more, it should be deleted from the loop list, and the loop-key that it shared with other members should also be abandoned. So the working flow in Figure-4 is to clean old loop-key stored on every loop member. The second step is to set up new loop-key. For the sake of saving rekeying time, the new key's creator is the loop node that has received the same cleaning messages twice.

In one word, the rekeying process is very important in long-time WSN. Loop-key should be changed as quickly as possible if some defection nodes are found. At the same time, normal key updating is also a good step to keep WSN safety.

Security enhancement in rekeying: Because defection nodes can overhear neighbors' messages during the rekey process, so some measures should be taken to keep the communication between remain nodes of loop in the overhearing area to be safe. Here we assume that a defection node can only overhear its one-hop neighbors' messages. It is obviously that we cannot prevent a defection node from hearing the first cleaning message, but we can stop him from getting new keys and other damages may cause by him. For example in Figure-1, if the node I is defected, link E-J and G-H should use new keys which node I cannot compute base on the pre-shared material and overhearing contents.

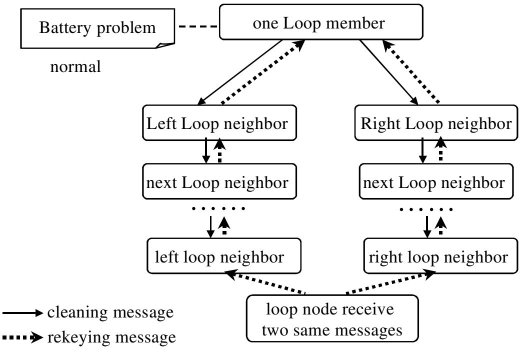


Fig. 4. Loop-based rekeying in WSN (2)

We use the polynomial-based key pre-distribution protocol proposed by Blundo et al. [10] to establish a new key shared between the last cleaning message's sender and receiver. The new key is only created and used between the sender and receiver, so it is a pair-wise key. Firstly before sensor nodes' deployment, one key sever randomly generates a bivariate t-degree polynomial $f(x, y) = \sum_{i,j=0}^{t} a_{ij}x^i y^i$ over a finite field F_q. where q is a prime number that is large enough to accommodate a cryptographic key, and has the property of f(x, y) = f(y, x). For each sensor node i with a unique ID, the key server computes a polynomial share of f(x, y), that is, f(i, y). For any two sensor nodes i and j, node i can compute the common key f(i, j) by evaluating f(i, y) at point j, and node j can compute the common key with i by evaluating f(j, y) at i. So to establish a pair-wise key both nodes just need to evaluate the polynomial with the ID of the other node without any key negotiation and the defection nodes know nothing of the new key. The scheme is proved secure and t-collusion resistant in mathematics.

At the same time, we also can use the time stamp to prevent fake cleaning messages made by the defection nodes.

4 Analysis, Simulation and Comparison

Nodes organization is the basic for research of WSN. WSNs of clustered organization are viewed as the most energy-efficient and most long-lived class of sensor networks [11]. There exist some key management schemes for WSN that are based on the cluster topology [12~14].

Creating a cluster for key management in a wireless sensor network at least includes 5 steps. Here we use the max connection degrees method as an example:

1. Similar to our loop-based scheme, every node broadcasts its ID to its neighbor nodes;
2. After received neighbor's ID message, every node calculates its neighbor numbers and send it with the neighbors' IDs to the neighbor nodes;
3. A node whose connections is bigger than its neighbors can send a cluster-head-request message to its neighbors;

4. Every node with lower connections sends a reply message to those cluster-head-request messages: join or reject. Nodes that received different request messages have to choose one of those cluster-head campaigners as their cluster header. Which node to be chosen is determined by ID or other parameters.
5. After received enough join messages from neighbor nodes, the cluster-head candidate can set up a cluster key with its cluster members.

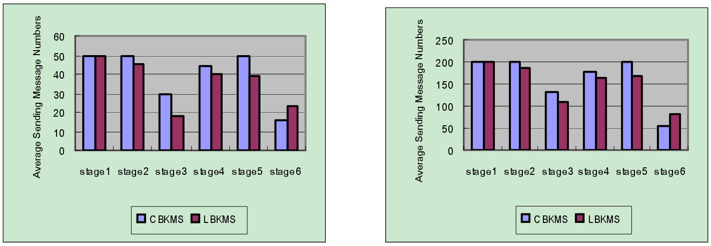
It is obvious that the key management based on cluster topology is more complicated than our scheme described in section 3. According to the comparison in table-2 and 3, the results can be showed as follows:

Communication cost: As a resource-poor network, WSN cannot afford too much communication among its nodes. The cluster-to-cluster relationship is more complex than that of loop-to-loop. It is common that some neighboring nodes are shared between two loops. But it would be redundant that more than one node are shared between two clusters. Two close clusters will cost more energy on the communication than two loops.

Storage cost: The cluster-based topology has to save neighbor clusters' information as route in the header and some members' storage. On the contrary, in the loop-based topology, the neighbor route information is already broadcasted during the second stage of the loop's forming.

Table 2. Cluster-based VS loop-based in communication

CBKMS				LBKMS			
stage	action	content	cost	stage	action	content	cost
1	All nodes broadcast	Self ID		1	All nodes broadcast	Self ID	
2	All nodes broadcast	Neighbor IDs and numbers	large	2	A few nodes broadcast	Neighbor table	small
					other nodes broadcast	Link table	small
3	Some nodes broadcast	Cluster head request		3	those nodes find match Loop link broadcast	LOOP	small
4	Neighbor nodes of some nodes	Reply message	large	4	other nodes broadcast	LOOP	
5	all nodes broadcast	Cluster inform		5	Some Nodes receive same LOOP messages from two neighbors broadcast	Loop inform	
6	All headers broadcast	Cluster key	small	6	Loop-creator	Loop key	large



Network size=50 nodes

Network size=200 nodes

Fig. 5. Sending message numbers contrast

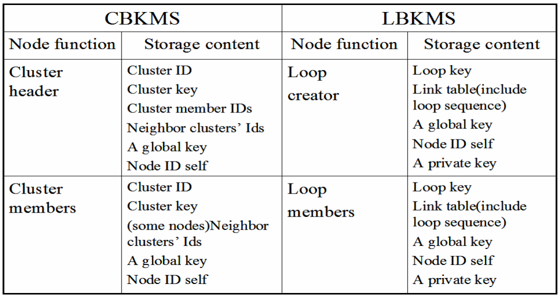
Communication is the biggest energy consumer. Especially the cost of sending message is much larger than receiving message. We use ns2 to simulate WSN with different network size and apply CBKMS and LBKMS at same conditions. After calculating average sending messages numbers, the contrast result is list in Figure-5. We can find that CBKMS send more messages than LBKMS from stage1 to 5, only in stage 6 that loop key have to be transmitted more hops than cluster key.

From perspective of security, the loop-based Key management scheme is safer and more stable than the cluster-based one.

Firstly, those two schemes have different role assignment among sensor nodes. The difference is listed in Table-4. From the comparison table we can find that CBKMS assigns many important tasks on cluster headers. A header node will play as a header all the time till it is replaced by another node. A loop creator's identifier initializes a loop's forming and has right to generate a loop key. After the loop is formed, there is no difference between normal nodes and the loop creator.

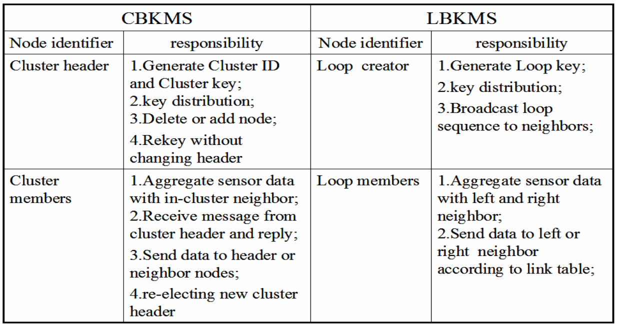
According the probability theory, every member in a loop topology has equal probability to be caught. Once a loop member is lost, its loop-neighbors can set up new loop quickly. What they need to do is to deleting the lost node ID from the loop sequence and generating a new loop key. If a cluster header is caught, then its member nodes have to take part in a new cluster header's election. At the same time, the probability of a cluster header being caught is determined by the result that cluster

Table 3. Cluster-based VS loop-based in node storage



CBKMS		LBKMS	
Node function	Storage content	Node function	Storage content
Cluster header	Cluster ID Cluster key Cluster member IDs Neighbor clusters' Ids A global key Node ID self	Loop creator	Loop key Link table(include loop sequence) A global key Node ID self A private key
Cluster members	Cluster ID Cluster key (some nodes)Neighbor clusters' Ids A global key Node ID self	Loop members	Loop key Link table(include loop sequence) A global key Node ID self A private key

Table 4. Node responsibility comparison between CBKMS and LBKMS

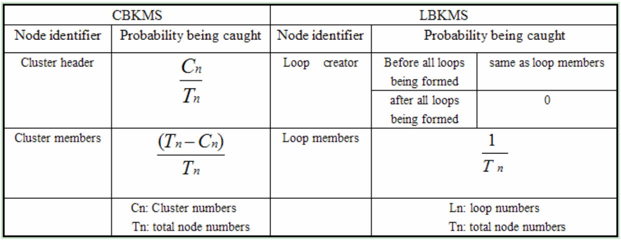


CBKMS		LBKMS	
Node identifier	responsibility	Node identifier	responsibility

Cluster header	1.Generate Cluster ID and Cluster key;
	2.key distribution;
	3.Delete or add node;
	4.Rekey without changing header
		Loop creator	1.Generate Loop key;
			2.key distribution;
			3.Broadcast loop sequence to neighbors;

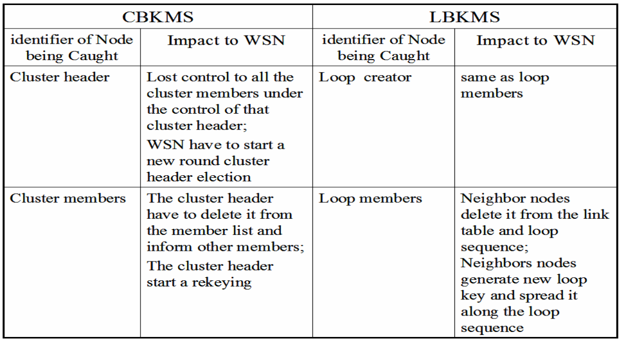
Cluster members	1.Aggregate sensor data with in-cluster neighbor; 2.Receive message from cluster header and reply; 3.Send data to header or neighbor nodes; 4.re-electing new cluster header	Loop members	1.Aggregate sensor data with left and right neighbor; 2.Send data to left or right neighbor according to link table;

Table 5. Comparison of probability of node being caught



CBKMS		LBKMS		
Node identifier	Probability being caught	Node identifier	Probability being caught	
Cluster header	$\frac{C_n}{T_n}$	Loop creator	Before all loops being formed	same as loop members
			after all loops being formed	0
Cluster members	$\frac{(T_n - C_n)}{T_n}$	Loop members	$\frac{1}{T_n}$	
	Cn: Cluster numbers Tn: total node numbers		Ln: loop numbers Tn: total node numbers	

Table 6. Comparison of impact of node being caught



CBKMS		LBKMS	
identifier of Node being Caught	Impact to WSN	identifier of Node being Caught	Impact to WSN
Cluster header	Lost control to all the cluster members under the control of that cluster header; WSN have to start a new round cluster header election	Loop creator	same as loop members
Cluster members	The cluster header have to delete it from the member list and inform other members; The cluster header start a rekeying	Loop members	Neighbor nodes delete it from the link table and loop sequence; Neighbors nodes generate new loop key and spread it along the loop sequence

numbers compare to the total node numbers. This probability is greater than that of a loop creator being caught. The probability comparison and impact comparison is listed in Table-5 and Table-6.

5 Conclusion

Key management is one of the most important technologies in the security mechanism of WSN. In this paper, we present a new key management scheme called LBKMS which integrates key pre-distribution mechanism in a loop-based infrastructure. LBKMS is also a dynamic scheme that can accommodate changing scenarios. The rekeying scheme based on loop topology and its security enhancement is also described in detail. Comparing with cluster-based key management schemes, LBKMS key management is proved to be more efficient, cost-saving and safe. Future research should focus on further reduction of communication cost in key establishment.

Acknowledgments

This work was supported by the National Research Foundation for the Doctoral Program of Higher Education of China under grant No. 20049998027, and the National Science Foundation of China under grant No. 90604006 and No. 90104001.

References

[1] Carman, D.W., Kruus, P.S., Matt, B.J.: Constraints and approaches for distributed sensor network security. Technical Report #00-010, NAI Labs (2000)

[2] Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: The 9th ACM conference on Computer and Communications, Washington, DC, USA, November 18-22, pp. 41–47 (2002)

[3] Chan, H., Perrig, A., Song, D.: Random key pre-distribution schemes for sensor networks. In: Proc. 2003 IEEE Symposium on Security and Privacy, May 11-14, pp. 197–213 (2003)

[4] Liu, D., Ning, P.: Establishing pairwise keys in distributed sensor networks. In: ACM Conference on Computer and Communications Security, pp. 52–61 (2003)

[5] Du, W., Deng, J., Han, Y.S., Chen, S., Varshney, P.K.: A key management scheme for wireless sensor networks using deployment knowledge. In: INFOCOM 2004, vol. 1, pp. 586–597 (March 7-11, 2004)

[6] Du, W., Deng, J., Han, Y.S., Varshney, P.K., Katz, J., Khalili, A.: A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. ACM Transactions on Information and System Security 8(2), 228–258 (2005)

[7] Blom, R.: An optimal class of symmetric key generation systems. In: Beth, T., Cot, N., Ingemarsson, I. (eds.) EUROCRYPT 1984. LNCS, vol. 209, pp. 335–338. Springer, Heidelberg (1985)

[8] Choi, S., Youn, H.: An Efficient Key Pre-distribution Scheme for Secure Distributed Sensor Networks. In: EUC 2005. LNCS, vol. 3823, pp. 1088–1097. Springer, Heidelberg (2005)

[9] Li, Y., Wang, X., Baueregger, F., Xue, X., Toh, C.K.: Loop-Based Topology Maintenance in Wireless Sensor Networks. In: Lu, X., Zhao, W. (eds.) ICCNMC 2005. LNCS, vol. 3619, Springer, Heidelberg (2005)

[10] Blundo, C., Santix, A D, Herzberg, A., Kutten, S., Vaccaro, U., Yung, M.: Perfectly-secure key distribution for dynamic conferences. In: The 12th Annual International Cryptology Conference on Advances in Cryptology, pp. 471–486. Springer, Berlin (1992)

[11] Vlajic, N., Xia, D.: Wireless Sensor Networks: To Cluster or Not To Cluster? In: IEEE International Symposium on WoWMoM 2006, Niagara-Falls, Buffalo-NY, USA (June 2006)

[12] Chorzempa, M., Park, J.-M., Eltoweissy, M.: SECK: survivable and efficient clustered keying for wireless sensor networks. In: IPCCC 2005 (2005)

[13] Younis, M.F., Ghumman, K., Eltoweissy, M.: Location-Aware Combinatorial Key Management Scheme for Clustered Sensor Networks. Parallel and Distributed Systems, IEEE Transactions 17(8), 865–882 (2006)

[14] Lin, L., Ru-chuan, W., Bo, J., Hai-ping, H.: Research of Layer-Cluster Key Management Scheme on Wireless Sensor Networks. Journal of Electronics & Information Technology 28(12) (December 2006)