Centralized Key Management Scheme in Wireless Sensor Networks

Saber Banihashemian, Abbas Ghaemi Bafghi & Mohammad Hossien Yaghmaee Moghaddam

Wireless Personal Communications
An International Journal

ISSN 0929-6212
Volume 60
Number 3

Wireless Pers Commun (2011)
60:463-474
DOI 10.1007/s11277-011-0302-y

Wireless Pers Commun (2011) 60:463–474
DOI 10.1007/s11277-011-0302-y

Centralized Key Management Scheme in Wireless Sensor Networks

Saber Banihashemian · Abbas Ghaemi Bafghi ·
Mohammad Hossien Yaghmaee Moghaddam

Published online: 27 April 2011

Abstract Today, key management is widely recognized as an important aspect of security in wireless sensor networks. In these networks, sensor nodes can be either mobile or static. Therefore, supporting the mobility of the nodes can be regarded as a purpose of key management schemes. In our previous work, we presented a key management scheme that was more efficient with respect to security and connectivity compared to the other ones. In that scheme, it is assumed that the nodes are static. In this paper we are going to present a scheme that supports the mobility of the nodes and makes the initial scheme more flexible. The basic criterion for the evaluation of the scheme is the communication overhead. First, the nodes establish a secure link with the cluster heads and then establish a secure link among themselves with the help of the cluster heads. We have analyzed this scheme with regards to the communication overhead and we will compare it with the other schemes.

Keywords Key management · Pre-distribution · Wireless sensor network · Heterogeneous · Mobility

1 Introduction

A wireless sensor network (WSN) be made up of an a lot number of very small sensor nodes. These sensor nodes have limited computing capability and energy power. They are deployed in any place, and act without any help. These characteristic are caused security problems. It is important to be able to do different cryptographic operations, including

S. Banihashemian (✉) · A. Ghaemi Bafghi · M. H. Yaghmaee Moghaddam
Department of Computer Engineering, Ferdowsi University of Mashhad (FUM), Mashhad, Iran
e-mail: saberbanihashemi@yahoo.com

A. Ghaemi Bafghi
e-mail: ghaemib@ferdowsi.um.ac.ir

M. H. Yaghmaee Moghaddam
e-mail: hyaghmae@ferdowsi.um.ac.ir

encryption, authentication and … to achieve security in sensor networks. Neighboring nodes must establish keys for these operations before they can communicate securely. Key management schemes are methods utilized for distribute these cryptographic keys in the network.

Eschenauer and Gligor [1] proposed a probabilistic key pre-distribution technique. It generates a key pool consists of enormous number of symmetric keys. Then some keys randomly picked from the key pool and stored within each node. Neighboring nodes set up a secure link if they had at least one common key. After this phase, neighboring nodes that do not have a shared key with each other can set up their own keys by two or more key path.

Asymmetric Predistribution (AP) key management scheme is proposed by Du et al. [2]. Its main idea is to preload great number of keys in each powerful node (H-sensor), and a little number of keys in each ordinary sensor, L-sensor. L-sensors have restricted storage space and communication ability. AP scheme is more scalable than the basic scheme and it decreases the amount of stored keys contrasted to the basic scheme. Chan et al. [3] offer the q-composite key pre-distribution to enhance resiliency against node capture attacks. In this scheme, two sensor nodes set up a secure link if they have at least q shared keys. A new communication link key K is created as the hash of all q' common keys that $q' > q$, e.g., $k = hash(k_0||k_1|| \cdots ||k_{q'})$. The rising of q in the q-composite scheme causes the lessening of the probability of setting up a secure link among two neighboring nodes. Hence, the size of key pool is reduced to avoid this effect. The q-composite scheme is effective when the number of captured sensor nodes is low. The performance of the q-composite scheme versus node capture will lessen by raising the number of captured nodes. From the other point of view, this scheme doesn't support key revocation. The key revocation in wireless sensor networks is suggested in [1] for the first time and is further investigated in [3]. The shared key discovery should be done again, if we want to revoke the revealed keys from the network. This procedure has many communications overhead on the sensor nodes. A framework for key management schemes in distributed sensor networks with heterogeneous nodes was proposed by Liu et al. [4]. Traynor et al. [5] use a small fraction of more capable nodes reduces the result of node compromise. Traynor et al. [6] studied the impacts of the unbalanced key management scheme, and designed a complementary set of key establishment schemes known as LIGER. Some location-aware schemes are proposed in [7,8] which improve the security of the key pre-distribution schemes. The idea of threshold key predistribution schemes is suggested in [9] and additional studied in [10].

In our previous article we presented a key management scheme, RKPH [11] that uses the heterogeneity ability among the nodes. This scheme was based on [2]. It used separate keys in distinctive clusters and take into consideration the distance of sensor nodes from theirs cluster head. We compared this scheme with AP [2] and EG [1] schemes and showed that the scheme has gained noticeable improvements with respect to security and connectivity compared to the other schemes. Then we evaluated the impact of effective parameters on resiliency in RKPH scheme in [12] and we proposed ARKPH scheme for improving the RKPH scheme in [13]. In these articles it was assumed that the sensor nodes are static. However in some applications mobile nodes might be used. In the present work, for more flexibility, we presented a scheme called CRKPH that supports the mobility of the nodes. In this scheme a certain percentage of the nodes are mobile. Our second purpose of presenting this scheme is to consider the communication overhead for establishing secure links and make it efficient with respect to energy consumption. For this end, we changed the discovery stage in such a way that it supports this propose. We will analyze this scheme with regards to the communication overhead and security. In Sect. 3, the

CRKPH scheme will be described. In Sect. 4, we will come to the evaluation of the scheme with respect to the communication overhead and security. Finally, a conclusion will be presented.

We use the following notations to describe the involved cryptographic operations in this paper.

BS	Base Station
SN_A	Sensor node A
M	Plain message
id_{BK_l}	Identifier of base key l
$\rightarrow$	Sending a message
id_A	Identifier of node A
CH_A	Cluster head number of node A (cluster head that has shared key with Node A)
CH_{now}	Cluster head of Current cluster that node A has moved to it
$Neighbor_A$	Neighbor of node A
$E(K\|M)$	Encrypt message M with key K
$shared_{CH_A,A}$	Shared key between node A and cluster head CH_A for authentication of Mobile node after movement
K_{C_i}	Cluster key of cluster i
K_{BS,CH_i}	Pairwise key between BS and cluster head i
$K_{BS,A}$	Pairwise key between BS and node A
dist	Distance between adjacent levels
$seed_{i,z}$	Seed related to ith cluster and zth level
$DK_{k,j}$	ith key in a sensor node hashed by seed j
hash(K,seed)	Hash key K with *seed*

2 Network Models

In this scheme it is assumed that the BS is reliable and there is no limitation in terms of energy, memory and processing power. In this model, the sensor network consists of a few number of H-sensor and a large number of L-sensors. H-Ss have energy, as well as processing power, however they are limited compared to the BS. These H-Ss have been equipped with resistant hardware and can communicate directly to the BS. H-Ss function as the head cluster and L-Ss are the components of the cluster. Also, it is assumed that the attackers in the initial moments following the node deployment are unable to capture a node. After this time, an attacker might be able to capture any nodes. A similar assumption has been presented in the [14]. Due to cost limitations, L-Ss have not been equipped with resistant hardware. Assuming that an attacker captures an L-S, it may be able to extract all the documents, data and the storage code on that node.

The following assumptions exist:

1. H-Ss have been equipped with resistant hardware. It is appropriate to assume that H-Ss are equipped with this technology.
2. H-S location is fixed.
3. H-Ss have a wide range of message propagation in a way that most L-Ss can receive the Hello message from one or more H-S.
4. Each H-S is equipped with a GPS and can identify its location.

Table 1 Key pool of proposed scheme

Seed_S	DK_{1-S}	DK_{2-S}	DK_{3-S}	....	DK_{n-S}
⋮ ⋮					
Seed_3	DK_{1-3}	DK_{2-3}	DK_{3-3}	....	DK_{n-3}
Seed_2	DK_{1-2}	DK_{2-2}	DK_{3-2}	....	DK_{n-2}
Seed_1	DK_{1-1}	DK_{2-1}	DK_{3-1}	....	DK_{n-1}
Base keys	BK_1	BK_2	BK_3	....	BK_n

3 Centralized Mobile Key Management Scheme

In this section, the centralized mobile key management scheme is presented. In this scheme establishing a secure link between the sensor nodes is done by the cluster heads. This causes the communication overhead in the sensor nodes to reduce considerably. Establishing a secure link for the mobile sensor nodes is divided into two parts: A mobile sensor node which has been moved to another location in the same cluster and a mobile sensor node that has left its previous cluster and joins to a new cluster. If the node moves to a location in the same cluster, then it tries to establish a secure link with the other nodes within the cluster by using the cluster head. If the mobile node moves to another cluster, then by using the previous cluster head and the Base Station, first a secure link is established with the current cluster head and then initiates establishing secure links with its neighboring nodes.

The centralized mobile key management scheme consists of five stages namely, pre-distribution, localization and cluster formation, deriving new keys, establishing secure link with the cluster head, and establishing secure link with the neighboring nodes. The first three stages are similar with RKPH. In the pre-distribution stage a number of keys are assigned to the nodes. In the localization and cluster formation stage, the clusters are formed and are partitioned into different levels in a logical way. In the deriving of new keys stage, new keys in each node are produced with respect to the cluster and the level at which they are located. In the fourth stage the nodes discover their shared keys with the cluster heads. In the final stage the nodes establish secure link among themselves. When a sensor node move to another placed, the final stage is done again.

3.1 Key Pre-Distribution Stage

In the first stage, a key pool consisting of base keys and derived keys as shown in Table 1 are produced. A number of k and c base keys are randomly chosen and are stored in each sensor node and cluster head respectively in which c is much greater than k.

3.2 Localization and Cluster Formation

Following the nodes deployment, the cluster heads obtain their location information through GPS and send it to the base station. The base station can estimate the maximum distance of a point in the cluster. The base station creates a virtual space with regards to the deployment area. Then considering the location of the cluster heads, the scope of each cluster and the number of the levels are determined. Simultaneously with the formation of the virtual space by the base station, each cluster head issues a 'Hello' message and the nodes get their distance from the cluster head based on the RSSI and join the cluster that has the least distance with them. The sensor nodes maintain this distance to use in the next stage.

3.3 Deriving New Keys

The base station sends the related seeds to each cluster head:

$$BS \rightarrow CH_i : E(K_{BS,CH_i} | [seed_{i,1}, dist_1], [seed_{i,2}, dist_2], \ldots, [seed_{i,z}, dist_z])$$

The head cluster stores the received seeds and sends them to the nodes inside the cluster and encrypt this message with the cluster keys:

$$CH_i \rightarrow Nodes\ in\ cluster : E(K_{C_i} | [seed_{i,1}, dist_1], [seed_{i,2}, dist_2], \ldots, [seed_{i,z}, dist_z])$$

Then each node calculates its distance to the cluster head and uses the related seeds to produce derived keys. In addition to producing its own level keys, each node produces the next level keys to link among the neighboring levels.

$$\begin{array}{ll} DK_{1,j} = hash(BK_1, seed_i, j) & DK_{1,j+1} = hash(BK_1, seed_i, j+1) \\ DK_{1,j} = hash(BK_2, seed_i, j) & DK_{2,j+1} = hash(BK_2, seed_i, j+1) \\ \ldots & \ldots \\ DK_{k,j} = hash(BK_k, seed_i, j) & DK_{k,j+1} = hash(BK_k, seed_i, j+1) \end{array}$$

Following the production of the new keys, the sensor nodes, delete the base and cluster keys in order to prevent revealing the seeds and the base keys as a result of node capture attack. Since the head clusters are tamper-resistance, capturing the head clusters does not give keys information to the attacker. Therefore, the base keys and seeds will remain in the cluster heads and in the following stages, if necessary, derived keys will be created by the derivation of base keys and the seeds.

3.4 Link Establishment with the Cluster Head

In the first moment of deployment each of the sensor nodes send a list of the identifiers of k base keys and the identifiers of the seeds they use to the cluster head.

$$SN_i \rightarrow its\ cluster\ head : M(\{id_{BK_u} | u = 1, 2, \ldots, k\} seed_M, seed_{M+1})$$

After receiving the IDs of the sensor keys, the cluster head sends the list of the shared keys with each node to the same node. Each sensor node has a cluster head number which relates to the cluster head that has shared key with node A. Accordingly cluster head number of node A in the first moment of deployment is the cluster in which the sensor node is placed. The cluster head number of node A has been show with CH_A in the next stages. The cluster head stores the list of the shared keys with each node to be used in the next stage.

3.5 Secure Link Establishment Among the Sensor Nodes

This stage can be done at each time whether before movement or after movement. In this stage each sensor node A sends its identifier, id_A together with the identifier of the neighboring node B, id_B, in addition to the cluster number of each A and B nodes to the current cluster head, CH_{now}.

$$SN_A \rightarrow CH_{now} : M(id_A, CH_A, id_B, CH_B)$$

Having received the key establishment request, the current head cluster, CH_{now}, establishes a secure link among the two nodes. Here, four alternatives might occur:

1. The current cluster head belongs to the same cluster which has identified the cluster number of A and B nodes. In other words, $CH_{now} = CH_A = CH_B$, in which case CH_{now} sends a shared key for each one of the A and B nodes.
2. The current cluster head belongs to the cluster that has identified the cluster number of A. in other words, $CH_{now} = CH_A \# CH_B$, in which case CH_{now} creates a shared key, sends it to A and sends the shared key to B with the help of the cluster head CH_B.
3. The current cluster head belongs to the cluster which has identified the cluster number of node B. in other words, $CH_{now} = CH_B \# CH_A$, in which case CH_{now} creates a shared key, sends it to B and sends the shared key to A with the help of cluster head CH_A.
4. The current cluster head does not belong to the cluster which has identified the cluster number of A and B. in other words, $CH_{now} \# CH_B \# CH_A$, in which case CH_{now} creates a shared key, sends it to B with the help of CH_B and sends it to A with the help of CH_A.

When a mobile node is joined to a new cluster, considerations must be made as to the prevention of node replication. For this reason we are going to describe a mechanism that provides node authentication.

3.6 Mobile Node Authentication

In this section, prior to leaving the cluster, the mobile node does the *cluster leaving stage* in which the secure links of the mobile node with the other neighboring nodes will be expired. Following the movement of the sensor node to another location the node authentication stage will be done.

(1) Cluster leaving stage

When leaving the cluster, the mobile node A sends the LEAVE_REQ message to the cluster head. Having received this message, the cluster head first sends a message to the mobile node's neighbors so that they can expire their link to the mobile node.

$$CH_A \rightarrow Neighbor_A : E(K_{CH_A, Neighbor_A} | \exp ire\_\sec ure\_link, id_A)$$

In which *expire_secure_link* is the request to expire secure links with the mobile node with the ID of ID_A. Having received this message, the mobile node's neighboring nodes expire their links to the mobile node. Then the cluster head sends a message to the mobile node and gives a key for the identification of the node in another time.

$$CH_A \rightarrow A : E(K_{CH_A, A} | shared_{CH_A, A})$$

In which $K_{CH_A, A}$ is an encryption key between the cluster head and the mobile node. $shared_{CH_A, A}$ is a random key created by the cluster head which is used for identifying the mobile node A. Procedure of cluster leaving is shown in Fig. 1.

(2) Cluster joining stage

Following the arrival of the node to the destination, mobile node authentication is first done by the present cluster head. Node authentication will be done by the key which was given to the mobile node at the *leaving cluster stage*, $shared_{CH_A, A}$. In node authentication two alternatives might occur.

(a) The node is in its previous cluster:

In this way the mobile node A sends a nonce together with the ID number and its cluster number to the CH_{now}.

$$A \rightarrow CH_{now} : M(id_A, CH_A, nonce_A, E(shared_{CH_A, A} | nonce_A))$$

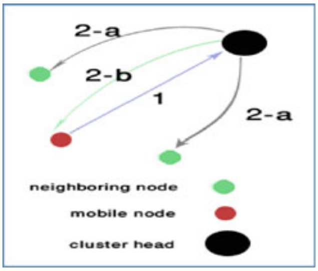


Fig. 1 Procedure of cluster leaving. **1** The mobile node sends the LEAVE_REQ to the cluster head to leave the cluster. **2a** Having received the LEAVE_REQ, the cluster head sends a message to the mobile node's neighbors so that they expire their link to the mobile node. **2b** The cluster head creates a random key and gives it to the mobile node, so that it can identify that node in the other requests

The cluster head CH_{now} which is actually CH_A has the shared key $shared_{CH_A,A}$, therefore it can calculate $E(shared_{CH_A,A}|nonce_A)$ and authenticate node A. Following the authentication of node A, the head cluster sends a new nonce as an acknowledgement to node A.

$$CH_{now} \rightarrow A : M(id_A, CH_A, nonce_A, nonce_{CH_A}, E(shared_{CH_A,A}|nonce_A, nonce_{CH_A}))$$

After the authentication of node A is completed, the *secure link establishment among the sensor nodes* stage will be done.

(b) The node leaves its previous cluster and moves to a new one.

In this way, the mobile node A sends a nonce together with the ID and its cluster head number to the CH_{now}.

$$A \rightarrow CH_{now} : M(id_A, CH_A, nonce_A, E(shared_{CH_A,A}|nonce_A))$$

Then CH_{now} forwards this message to the base station.

$$CH_{now} \rightarrow BS : E(K_{BS,CH_{now}}|id_A, CH_A, nonce_A, E(shared_{CH_A,A}|nonce_A))$$

Having received this message, the base station requests $shared_{CH_A,A}$ from CH_A.

$$BS \rightarrow CH_A : E(K_{BS,CH_A}|shared\_request, id_A, CH_A)$$

Then the cluster head CH_A sends $shared_{CH_A,A}$ to the head cluster.

$$CH_A \rightarrow BS : E(k_{BS,CH_A}|shared_{CH_A,A})$$

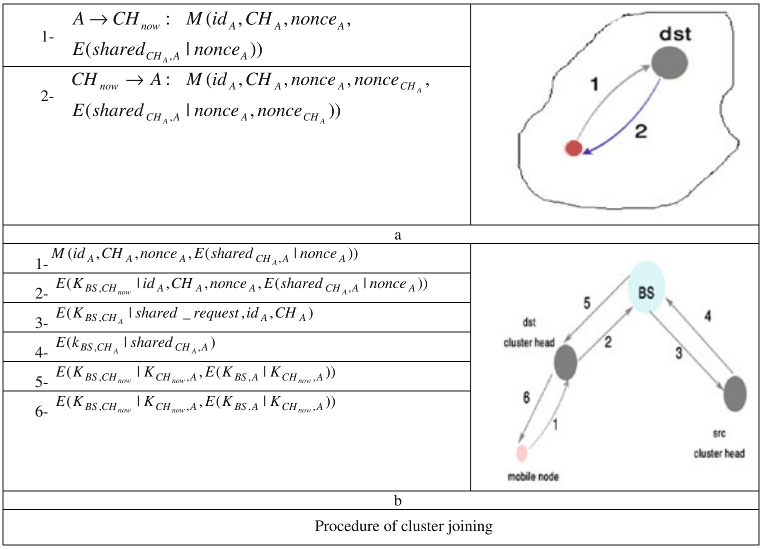
The base station authenticates the node. If the node is authenticated, the base station creates a pairwise key for CH_{now} and the mobile node A and sends it to the CH_{now}.

$$BS \rightarrow CH_{now} : E(K_{BS,CH_{now}}|K_{CH_{now},A}, E(K_{BS,A}|K_{CH_{now},A}))$$

Having received this key, CH_{now} sends a message including the pairwise key to node A.

$$CH_{now} \rightarrow BS : E(k_{BS,A}|K_{CH_{now},A})$$

Procedure of cluster joining is shown in Fig. 2.



1- $A \rightarrow CH_{now}: \; M(id_A, CH_A, nonce_A, E(shared_{CH_A,A} \mid nonce_A))$

2- $CH_{now} \rightarrow A: \; M(id_A, CH_A, nonce_A, nonce_{CH_A}, E(shared_{CH_A,A} \mid nonce_A, nonce_{CH_A}))$

a

1- $M(id_A, CH_A, nonce_A, E(shared_{CH_A,A} \mid nonce_A))$

2- $E(K_{BS,CH_{now}} \mid id_A, CH_A, nonce_A, E(shared_{CH_A,A} \mid nonce_A))$

3- $E(K_{BS,CH_A} \mid shared\_request, id_A, CH_A)$

4- $E(k_{BS,CH_A} \mid shared_{CH_A,A})$

5- $E(K_{BS,CH_{now}} \mid K_{CH_{now},A}, E(K_{BS,A} \mid K_{CH_{now},A}))$

6- $E(K_{BS,CH_{now}} \mid K_{CH_{now},A}, E(K_{BS,A} \mid K_{CH_{now},A}))$

b

Procedure of cluster joining

Fig. 2 Procedure of cluster joining. **a** Node stay in the same cluster. **b** Node leaves its previous cluster and moves to a new one

4 Evaluation

In this section we will evaluate the scheme with the parameter of the number of the keys in each sensor node. The criterion for evaluation in this section is the number of bits sent and received in the sensor nodes and the cluster heads. For the evaluation of the proposed scheme, we simulated the scheme in Matlab. For the simulation of the proposed scheme, we selected a number of default values and based our simulation on these parameters. These values are obtained through trial and error.

Parameter	Default value
Deployment area	400 m × 400 m
Distance of each level	40 m
radio transmission range of sensor node	40 m
The number of base key pool	1,000
The number of base keys in each sensor node	20
The number of base keys in each cluster head	100
The number of cluster heads	10
The number of ordinary sensor nodes	1,000
Key id length	15 bits
Node id length	15 bits
CH id length	8 bits
Key length	64 bits

In order to show the effect of the parameter on the protocol, we change the aforementioned parameter and keep the other parameters fixed. For evaluation of impact of mobility on communication overhead, we move respectively 10, 20 and 30 percent of sensor nodes. After

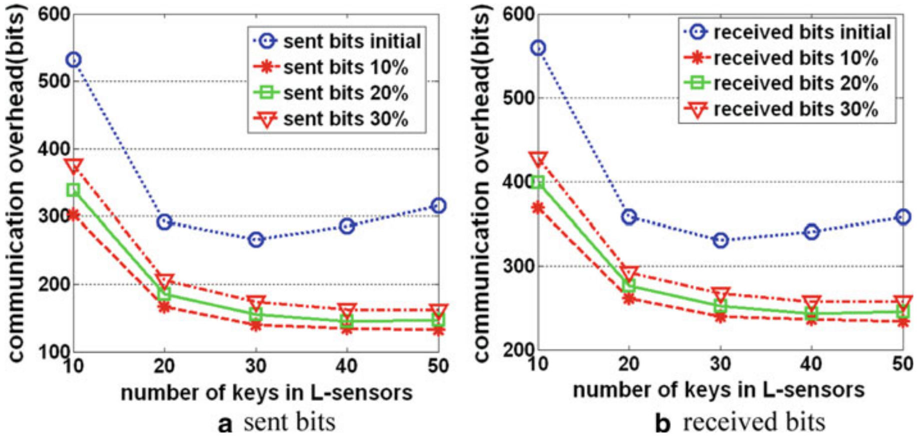


Fig. 3 Average of **a** sent and **b** received bits in each sensor node for establishing a secure link

node movement, only final stage is done again. In order to evaluate the effect of the number of base keys on the amount of communications, we selected the number of the base keys in each sensor nodes respectively 10, 20, 30, 40 and 50 and analyzed its effect on the communication overhead. In this experiment header of messages have been overlooked and just we confined ourselves to the analysis of the number of the sent and received bits by the nodes and the cluster heads.

4.1 Communication Overhead on the Nodes

In this section we will try to evaluate the centralized mobile key management protocol in terms of the communication overhead on the sensor nodes. The communication overhead on the sensor nodes has been analyzed on the basis of the received and sent bits.

We analyzed the sent and received bits in the nodes for establishing a secure link. This amount is obtained by dividing the total number of the sent and received bits by the number of the secure links. Figure 3a and b respectively show the average of the sent and received bits in each node for establishing a secure link in *CRKPH* scheme.

As it can be seen, average of sent and received bits are decreased with increasing the number of base keys from 10 to 30 in sensor nodes, before node movement and after that. Cause of this behavior is that connectivity will be increased by increasing the number of base keys from 10 to 30 and then connectivity approaches one. With increasing the number of base keys in sensor nodes from 30 to 50, connectivity is almost one and the number of secure links is fixed but communication overhead in initially after deployment and before movement is increased in *Link establishment with the cluster head* stage due to increasing the number of base keys. Accordingly average of send and received bits in each sensor node for establishing a secure link is increased.

4.2 The Communication Overhead on the Cluster Heads

In this section we try to evaluate the centralized mobile key management protocol in terms of the communication overhead on the cluster heads. In Fig. 4 the average sent and received bits in the cluster heads for establishing a secure link has been shown. As it can be seen in Fig. 4, after node movement in the *CRKPH* scheme, with the increase in the number of base

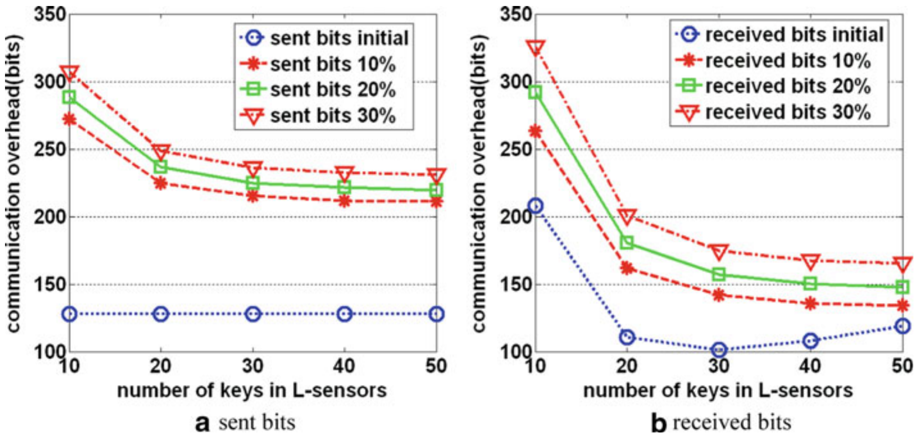
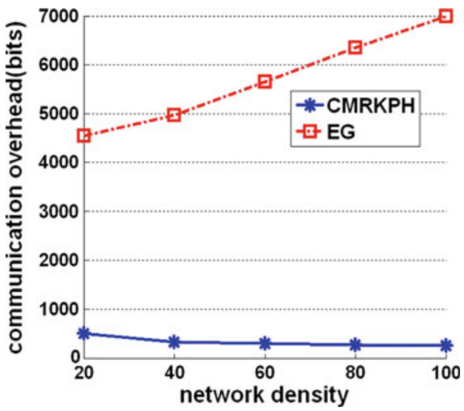


Fig. 4 Average of **a** sent and **b** received bits in each cluster head for establishing a secure link

Fig. 5 Comparison of CRKPH with EG scheme



keys in each node, the average received bits in the cluster heads for establishing a secure link decreases and then reaches a fixed rate.

4.3 Comparison

In this section, we compare our scheme with another key management scheme EG [1] based on communication overhead of establishing a secure link for various density of nodes. Density of nodes refers to the average number of nodes per unit disc communication area (i.e., the expected number of neighbors per node). In EG scheme, increasing the network density increases the number of broadcast messages needed to perform path-discovery to the distance of three hops [15]. Hence, the communication overhead of EG rises steadily with increasing density of nodes. But in our scheme, as it can be seen in Fig. 5, communication overhead is fixed with increasing density of nodes. Also communication overhead of our scheme is less than EG scheme.

5 Conclusion

In this paper a protocol called *CRKPH* has been proposed for the key management of the mobile wireless sensor networks. In this scheme, we added two further stage, *link establishment with the cluster head* and *secure link establishment* to the *RKPH* scheme. In this scheme before establishing secure link among the nodes, each node establishes a link with the cluster heads. Before establishing secure link with its neighbors, the mobile node is first authenticated and then the two neighboring nodes establish secure link between themselves through the cluster head. Then the communication overhead was chosen as the criterion for evaluation and was compared with the other schemes. The results show that great improvements have been obtained in terms of the linking overhead.

References

1. Eschenauer, L., & Gligor, V. D. (2002). A key management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on computer and communications security* (pp. 41–47).
2. Du, X., Xiao, Y., Guizani, M., & Chen, H.-H. (2007). An effective key management scheme for heterogeneous sensor networks. *Ad Hoc Networks, 5*(1), 24–34.
3. Chan, H., Perrig, A., & Song, D. (2003). Random key pre distribution schemes for sensor networks. In *IEEE symposium on research in security and privacy*.
4. Lu, K., Qian, Y., & Hu, J. (2006). A framework for distributed key management schemes in heterogeneous wireless sensor networks. In *IEEE international performance computing and communications conference* (pp. 513–519).
5. Traynor, P., Kumar, P., Bin Saad, H., Cao, G. & La Porta, T. (2006). Establishing pair-wise keys in heterogeneous sensor networks. In *Proceedings of the 25th IEEE international conference on computer communications*. doi:10.1109/INFOCOM.2006.260.
6. Traynor, P., Kumar, R., Bin Saad, H., Cao, G., & La Porta, T. (2007). Efficient hybrid security mechanisms for heterogeneous sensor networks. *IEEE Transactions Mobile Computing, 6*(6), 663–677.
7. Price, A., Kosaka, K., & Chatterjee, S. (2004). A secure key management scheme for sensor networks. In *Proceedings of the 10th Americas conference on information systems*.
8. Liu, D.D., & Ning, P. (2003). Location-based pairwise key establishments for static sensor networks. In *Proceedings of the 1st ACM workshop on security of ad hoc and sensor networks* (pp. 72–82).
9. Liu, D., Ning, P., & Li, R. (2005). Establishing pairwise keys in distributed sensor networks. *ACM Transactions on Information and System Security, 8*(1), 41–77.
10. Du, W., Deng, J., Han, Y. S., Varshney, P. K., Katz, J., & Khalili, A. (2005). A pairwise key pre-distribution scheme for wireless sensor networks. *ACM Transactions on Information and System Security, 8*(2), 228–258.
11. Banihashemian, S., & Ghaemi Bafghi, A. (2010). A new key management scheme in heterogeneous wireless sensor networks. In *12th international conference on advanced communication technology* (pp. 141–146).
12. Banihashemian, S., & Ghaemi Bafghi, A. (2010). Performance study of RKPH key management protocol and analysis of effective parameters on it. In *Proceeding national csicc2010 (in pesinan)*.
13. Banihashemian, S., & Ghaemi Bafghi, A. (2010). Alternative shared key replacement in heterogeneous wireless sensor networks. In *8th annual communication networks and services research conference* (pp. 174–178).
14. Zhu, S., Setia, S., & Jajodia, S. (2003). LEAP: Efficient security mechanisms for large-scale distributed sensor networks. In *Proceedings of the 10th ACM conference on computer and communications security* (pp. 62–72).
15. Chan, H., & Perrig, A., (2005). PIKE: Peer intermediaries for key establishment in sensor networks. In *Proceedings of INFOCOM 2005* (pp. 524–535).

Author Biographies

Saber Banihashemian was born on March 1983 in Ramsar, Iran. He received his B.S. degree in Software Engineering from Islamic Azad University of Lahijan, Lahijan, Iran in 2006, and M.S. degree in software engineering from Ferdowsi University of Mashhad in 2010. His research interests are in security, Wireless Sensor Networks (WSNs) and Distributed Systems.

Abbas Ghaemi Bafghi was born on April 1973 in Bojnord, Iran. He received his B.S. degree in Applied Mathematics in Computer from Ferdowsi University of Mashhad, Iran in 1995, and M.S. degree in Computer engineering from Amirkabir (Tehran Polytechnique) University of Technology, Iran in 1997. He received his Ph.D. degree in Computer engineering from Amirkabir (Tehran Polytechnique) University of Technology, Iran in 2004. He is member of Computer Society of Iran (CSI) and Iranian Society of Cryptology (ISC). He is an assistant professor in Department of Computer Engineering, Ferdowsi University of Mashhad, Iran. His research interests are in cryptology and security and he has published more than 50 conference and journal papers.

Mohammad Hossien Yaghmaee Moghaddam was born on July 1971 in Mashad, Iran. He received his B.S. degree in Communication Engineering from Sharif University of Technology, Tehran, Iran in 1993, and M.S. degree in communication engineering from Tehran Polytechnic (Amirkabir) University of Technology in 1995. He received his Ph.D. degree in communication engineering from Tehran Polytechnic (Amirkabir) University of Technology in 2000. He has been a computer network engineer with several networking projects in Iran Telecommunication Research Center (ITRC) since 1992. November 1998 to July1999, he was with Network Technology Group (NTG), C&C Media research labs., NEC corporation, Tokyo, Japan, as visiting research scholar. September 2007 to August 2008, he was with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, USA as the visiting associate professor. He is author of 3 books all in Farsi language. He has published more than 90 international conference and journal papers. His research interests are in Wireless Sensor Networks (WSNs), traffic and congestion control, high speed networks including ATM and MPLS, Quality of Services (QoS) and fuzzy logic control.