Energy Efficient Key Management Protocol in Wireless Sensor Networks

Jinsu Kim[1], Junghyun Lee[1], and Keewook Rim[2]
[1] *Dept. of Computer Science Engineering, Inha University*
[2] *Dept. of Computer and Information Science, Sunmoon University, South Korea*
{kjspace,jhlee}@inha.ac.kr, rim@sunmoon.ac.kr

Abstract

Wireless sensor networks (WSNs) are a technology of various uses from monitoring surrounding environment to collecting data. Because WSNs operate with limited resources of sensor nodes, its life is extended by cluster-based routing methods. However, WSNs which communicate through a wireless transmission medium are quite vulnerable in terms of security compared to wired networks and their sensor nodes have very limited communication and computing capacities and a small memory space. For these reasons, it is difficult to apply the established public key encryption technology. Accordingly, a large number of sensor nodes should be tolerable to errors and failures, and it is essential to achieve effective management and reinforced security functions through autonomous network construction. The cluster-based routing protocol proposed in this study showed energy-efficient key management by establishing a common shared key through the key of key ring or through authentication by a reliable institution.

Keywords: *Cluster-based Routing, Secure Protocol, Key Management Scheme*

1. Introduction

Wireless Sensor Network (WSN) is a base network for implementing ubiquitous computing. It is a wireless network composed of many ultralight and low-power sensors with limited resources such as memory and processor. Particularly because of mutual communication through wireless media, it is quite vulnerable in terms of security compared to wired networks. Because sensor nodes have very limited communication and computing capacities and a small memory space, it is not easy to apply existing public key encryption techniques such as RSA and Diffie-Hellman to sensor nodes. In addition, as they are deployed in physical unsecure environment, a very large number of sensor nodes should allow errors and failures and are interconnected through autonomous networking, and for this reason, effective management and security functions are critical elements. Recently, intensive research is being made in WSNs for developing various encryption methods through secure key distribution [1-6]. However, methods proposed for secure communication have a key distribution solution applicable limitedly to a specific structure [7]. For efficient energy management in general ad hoc networks or WSNs, a cluster-based routing protocol structure was proposed [8-10]. In a cluster-based network, generally member nodes form a cluster and transmit information to cluster head (CH), and CH compresses the information and sends it to Base Station (BS). Protocols such as LEACH [8], LEACH-C [9], 3DE_var [10] are representative methods that enhanced scalability and energy efficiency in cluster-based WSN. Because CH is elected at each round and plays the role of routing for a specific time, it becomes the major target for a malicious attacker that tries to make it difficult to authenticate routing elements or to damage them. Accordingly, in order to enhance the security of communication in cluster-based protocol, links among nodes should be changed by reshuffling distributed keys dynamically or after a specific time, but this process causes a large overhead, so should be avoided as much as possible. In addition, most of existing key management methods have the special structure that nodes in the network are not given mobility but finish their life at a given fixed position. However, if the network has a trouble or the density of sensor nodes is too high or low in a specific area, the whole network should be reconfigured into an efficient structure for smooth information collection, and for this, sensor nodes should be given mobility or new nodes should be inserted.

This study purposed not only efficient energy use in cluster formation, which is an advantage of the cluster-based routing method, but also for securer and more efficient communication within the cluster and higher overall network efficiency through securer participation in the established cluster when new nodes are added or existing nodes are assigned mobility.

2. Related Works

The key management methods proposed in WSN can be divided into Random key predistribution scheme [1,3,4,11], master key-based scheme [5], and BS-based scheme [12].

2.1. Random Key Pre-Distribution

The Random Key Pre-Distribution (RPK) [1] guarantees secure authentication among nodes through the three-step process of random key predistribution, shared key discovery, and path key establishment. Because connection weight is determined probabilistically in RPK, the entire graph representing WSN may not be connected completely and this problem is even more serious if sensor nodes are deployed irregularly or there are physical obstacles to communication in the environment. In particular, the size of key ring to be stored in each node has to be enlarged in order to increase network connection weight, and this enables a malicious attacker to get more keys through node compromise. In order to solve this problem, a method that utilizes information on sensor node deployment was proposed, but it still has the problem that a malicious attacker can use a key obtained from node compromise in other areas of the sensor network [13]. In addition, this method does not consider security analysis, through which compromised nodes can tap or hide compromise efficiently through mutual cooperation [14]. However, the key predistribution scheme is advantageous in that when mobility such as insertion of new nodes or cluster change of existing nodes has been assigned it can form a cluster for secure communication using the shared key owned by each node.

2.2. LEAP

As it was considered difficult to design a secure key mechanism using a key in a sensor network where a large number of sensors are scattered, LEAP [5] was proposed, which has 4 encryption keys and a key setting protocol. The 4 encryption keys are private key shared with BS, broadcasted group key that BS shares with all nodes in the network, pairwise key shared with other sensor nodes, and cluster key shared with a number of neighbor nodes. In LEAP, an attack node cannot know the private key, and the pairwise key and the cluster key are used only for authenticating surrounding neighbor nodes, and the group key is used only to decrypt broadcasted messages. Thus, this scheme can maximize the viability of a sensor network with threatened nodes. Because the private key and the group key are assigned before sensor nodes are deployed, a malicious attacker may compromise a sensor node. In addition, when a sensor node is compromised before initialization as in the master key-based scheme, the malicious attacker may be able to generate all the keys used in WSN by acquiring all information stored in the sensor node within a minute [15].

2.3. HIKES

Hierarchical Key Establishment Scheme (HIKES) [12] is a method in which BS plays the role of the Trust Authentication (TA) and assigns part of its roles to CH. It can generate a key from partial key escrow table in all nodes and can be elected as CH, and after data integration, information is transmitted to BS through message exchange among CHs. However, because sensor node authentication is carried out by BS and partial key escrow table has to be stored in every node, this method requires an additional storage space. Moreover, when a malicious attacker obtains partial key escrow table through node compromise, it can infer from the partial key escrow table a pairwise key between CH and sensor nodes situated in other areas. Furthermore, with increase in the number of nodes belonging to the cluster, CH has to send a larger size of message for node authentication and this may reduce the life of the whole network.

3. Key management schemes proposed in the secure cluster-based routing protocol (3DE_sec)

That a malicious attacker can compromise a node and use it in another cluster may cause many problems in communication among nodes, but if a new node authenticated by reliable BS is added to a cluster through the move or insertion of the node, it results in the occurrence of unnecessary data transmission including the exchange of a large volume of information for generating a shared key with neighbor nodes, so such unnecessary data transmission should be minimized. CH can be elected and sensed information can be transmitted through stable information exchange by applying an existing key management method to the cluster-based routing protocol, but when a new node is added or existing nodes are assigned mobility, the existing key management method has a limitation. Thus, this study proposes a key management method that can minimize CH load even with mobile nodes or the insertion or deletion of nodes. This key management method is called 3DE_sec, and it is composed of steps as follows.

3.1. Key PreDistribution Step

The key predistribution step is divided into the process of distributing k random keys and the process of setting the private key for unique communication with BS. In the key predistribution process, a large pool with P keys and their key identifiers are generated before all the nodes are deployed in WSNs, and k keys out of P keys are randomly selected and stored in the memory of each sensor node. In generating P unique keys, if the size of each key is N bits, we can generate 2^N different keys. Thus if the size of keys to be generated in the pool is over $\log_2 P$ bit, the uniqueness of keys is guaranteed.

In cluster-based routing, all nodes should have a common key with BS for sending information from other nodes in the same cluster or for receiving a query from BS. This key is called private key. The private key is generated and assigned before nodes are deployed. That is, the unique key (K_u^m) of each node (u) is generated by BS using $K_u^m = f_{K^m}(u)$, where f is a pseudo-random function, and K^m is the master key known only to the controller. Because of the efficient computing ability of the pseudo-random function, overhead caused by private key generation is negligible.

3.2. Shared Key Search Step

This study searches for a shared key with member nodes within the range of wireless communication from CH. CH can know whether the member nodes have a shared key by broadcasting its own key ID. Using the shared key, secure links to nodes are established and secure communication is guaranteed. The probability that the key ring assigned to each node is shared with neighbor nodes can be calculated from P and k as in Equation (1).

$$p' = 1 - \frac{((P-k)!)^2}{(P-2k)!P!} \quad ,0 \le p' \le 1 \tag{1}$$

where, since P is very large, we use Stirling's approximation for $n!$ (equation (2)) to simplify the expression of p', and obtained equation (3).

$$n! \approx n^n e^{-n} \sqrt{2\pi n} \tag{2}$$

$$p' = 1 - \frac{(1-\frac{k}{P})^{2(P-k+\frac{1}{2})}}{(1-\frac{2k}{P})^{(P-2k+\frac{1}{2})}} \tag{3}$$

If the number of nodes in WSNs and probability are predefined, suitable key ring size is calculated by equation (3).

3.3. Shared Key Generation through Authentication

If CH and shared key have not been set in the second step, namely, the shared key search step, a shared key like a path key in the key predistribution scheme should be generated. In the cluster-based scheme, CH is changed periodically and the size of the key ring (k) is smaller than the total number of nodes within the cluster, it is highly likely that nodes without a shared key and path key setting are frequent and it is impossible to reset the path key every time using the master key.

This study establishes stabler communication through authentication using an unique shared key among the nodes or through authentication by an authentication center commissioned by BS or BS. Accordingly, each node goes through one or more authentication processes and, as a result, securer and more reliable communication can be established than existing methodologies.

3.3.1. Authentication through shared key authentication: After the key predistribution step, each node has a k -sized key ring and a unique private key for secure communication with BS. This authentication, which is called primary authentication, is made through setting a shared key among nodes using k keys assigned to each node, and through this, a secure communication path is established.

3.3.2. Authentication by an authentication node trusted by BS: If authentication cannot be made using a shared key, it is carried out by an authentication node trusted by BS at the previous cluster formation step. That is, in case there is no shared key between newly elected CH and nodes in the cluster, if authentication is obtained from BS, it may increase the overhead of BS and network traffic. In this case, authentication can be performed quickly and overall delay time can be minimized by reusing node information obtained from CH. In case of 3DE_var, the existing CH can elect the optimal CH using various types of information, so authentication can be performed by a reliable node. In addition, whether a node has been an existing member of the cluster is confirmed using the cluster key from the higher cluster head, and this is called secondary authentication. The cluster key used by the authentication node trusted by BS is used as a shared key.

3.3.3. Authentication through BS: If there is a node not authenticated through the primary and secondary authentication, it has to be authenticated directly by BS as in Figure 1. That is, for a newly inserted node or a node moved from another cluster, which does not have authentication information such as shared key or previous cluster key, it is hardly possible to be authenticated at the primary or secondary authentication step. Such a node chooses values at random from the keys in the key ring assigned to each key, sends Authentication Request Packet (AREQ) to BS, and goes through authentication by BS. That is, in order to be authenticated by BS, it extracts some values from its key ring at random, and send a message encrypted using the private key to the elected CH. CH forwards encrypted messages from nodes without the shared key to BS, and BS decrypts the encrypted messages, and checks if they agree with the set of $< idx, BA, length, val >$ pairs, the authentication request sample code of the sender node ID. The applied idx means the index of the node in the key ring, BA is the base address, $length$ is length, and val is value. And these values are set by random functions, then val indicates the value of the key at a distance of $length$ from BA. If the values agree with each other Authentication Reply Packet (AREP) is sent, and if not and ID of the malicious node intending the disruption of communication is detected and sent to each CH so that it be removed from the key ring of all nodes. Because, due to the characteristic of the cluster-based scheme, CH should be changed at a regular interval of time, AREP contains the shared key assigned by BS after authentication. In Figure 1, the nodes sending the Join_REQ message means nodes that have passed the primary authentication.

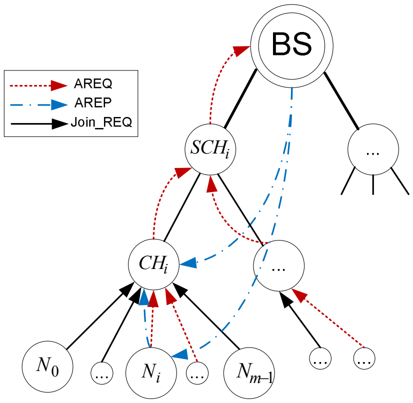
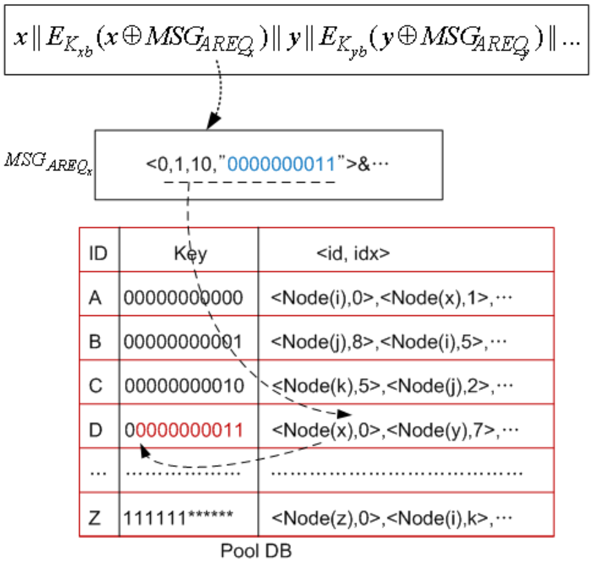


Figure 1. The example of the authentication request and authentication reply process

Vulnerable nodes such as nodes not shared with CH and newly inserted nodes should be authenticated by BS. Accordingly, as in Figure 2, the nodes build $< NodeID, idx >$ pairs, information contained in large pool P before key distribution and in the key ring with k keys distributed to each node, in the pool database of BS simultaneously with key predistribution. Here, $NodeID$ means the ID of each node, and idx is the index of each node ID in the key ring.


$x \| E_{K_{xb}}(x \oplus MSG_{AREQ_x}) \| y \| E_{K_{yb}}(y \oplus MSG_{AREQ_y}) \| \ldots$
MSG_{AREQ_x}
<0,1,10,"0000000011">&⋯
Pool DB

ID	Key	<id, idx>
A	00000000000	<Node(i),0>,<Node(x),1>,⋯
B	00000000001	<Node(j),8>,<Node(i),5>,⋯
C	00000000010	<Node(k),5>,<Node(j),2>,⋯
D	00000000011	<Node(x),0>,<Node(y),7>,⋯
...	....	....
Z	111111******	<Node(z),0>,<Node(i),k>,⋯

Figure 2. Example of the cryptographic message authentication validation process of BS

Figure 2 shows the process of checking an authentication sample code sent by a node without the shared key against the pool database owned by BS. Message (MSG_{AREQ_x}), which is the authentication sample code that node ID x encrypted using its private key, is decrypted by BS, and $< 0,1,10,"0000000011" >$ is its part. This means that the value of 10 bits at a distance of 1 from the key at the position of 0th index in node x. If this value coincides with the value in the BS database, tertiary authentication is performed. That is, node ID x is authenticated and AREP is sent so that CH and x can set the shared key. The message, whose value is the same as MSG_{AREQ_x}, is sent to AREP and reconfirmed by the node and CH, and then the shared key is used. The shared key generated through authentication is used as a cluster key for the secure receiving of data sensed during the present round.

Figure 3 is a protocol showing the overall situation of secure key management at each round of 3DE_sec. Step 2 shows key management through the authentication of the 3DE_sec protocol in the 'Else' statement. Table 1 shows the signs and definitions of terms used in Figure 3.

4. System Evaluations

The simulator used in this study was built in Visual C++. Previous research mentioned only the number of nodes required for secure communication in a network, and conducted experiment with simply expanding the number of nodes without mentioning network size($M \times M$). However, network density is closely related with the overall delay of cluster formation, and the size of the whole network also works as a large overhead to wireless sensors. In our experiment, accordingly, we limited network size ($100m \times 100m$) and formed clusters by determining the optimal number of clusters according to distance between BS and network area using Equation (4).

```
Setup Phase

1. Advertisement Step
$CH \Rightarrow G : id_{CH}, nonce, ADV$
$A_i : choose \quad r \quad such \quad that \quad r \in (R_{CH} \cap R_{A_i})$
2. Join-Req. Step
If $r \notin Ø$ then
  $A_i \to CH : id_{A_i}, id_{CH}, r, Join-Req., MAC_{k[r]}(id_{A_i} | id_{CH} | r | nonce)$
    $A_i \to CH : id_{A_i}, id_{CH}, Join-\mathrm{Re}\,q.,$
    $MAC_{k[A_i]}(id_{A_i} | id_{CH} | nonce | nonce | SC(..., < idx_{A_i}, BA_{A_i}, length_{A_i}, val_{A_i} >, ...))$
    $CH \to BS : id_{CH}, id_{BS}, AREQ_{CH}, MAC_{k[CH]}(id_{CH} | id_{BS} | nonce | F'(..., MAC_{A_i}, ...) | c_{CH})$
Else
 If $val_{BS_{A_i}} = val_{A_i}$ then
  $BS \to A_i : id_{BS}, id_{A_i}, AREP_{A_i}, MAC_{k[A_i]}(id_{BS} | id_{A_i} | nonce | ASC(..., < idx_{A_i}, BA_{A_i}, length_{A_i}, val_{A_i} >, ...))$
  $BS \to CH : id_{BS}, id_{CH}, AREP_{CH}, MAC_{k[CH]}(id_{BS} | id_{CH} | nonce | ASC(..., < idx_{A_i}, BA_{A_i}, length_{A_i}, val_{A_i} >, ...))$
 End If
End If
3. TDMA Scheduling Step
$CH \Rightarrow G : id_{CH}, (..., < id_{A_i}, TDMA_{A_i} >, ...), TDMA\ schedule$

Steady-State Phase

4. Receiving Sensing Data Step
$A_i \to CH : id_{A_i}, id_{CH}, d_{A_i}, MAC_{k[r]}(id_{A_i} | id_{CH} | d_{A_i} | nonce + j)$
5. Sending Aggregated Information Step
$CH \to BS : id_{CH}, id_{BS}, F(..., d_{A_i}, ...), MAC_{k[CH]}(F(..., d_{A_i}, ...) | c_{CH})$
```

Figure 3. 3DE_sec protocol

Table 1. Symbols and definitions

Symbol	Definition
$MAC_k(msg)$	MAC calculated using key k
c_x	Counter shared by node x and BS
r	id of the keys in the key ring
$k[r]$	Symmetric key associated with id r
R_x	Set of key ids in node x 's key ring
j	Reporting cycle within the current round
A_i, CH, BS	Node, Cluster head, Base Station
G	Set of nodes in cluster
$\Rightarrow, \to$	Broadcast, Unicast
id_x	id of node x
$ADV, Join-Req.$	Advertisement Message, Join Requestment Message

F'	Authentication request data fusion function
F	Data fusion function
SC	Authentication Request Sample Code ($A_i \rightarrow BS$)
ASC	Authentication Validation Sample Code ($BS \rightarrow A_i$)

$$k_{opt} = \frac{\sqrt{P}}{\sqrt{2\pi}} \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}} \frac{M}{d_{toBS}^2} \qquad (4)$$

where, P is the number of the nodes in WSN, d_{toBS} is the distance from BS to cluster head node. For our experiments, we set $\varepsilon_{fs} = 10pJ/bit/m^2$, $\varepsilon_{mp} = 0.0013pJ/bit/m^4$, $M = 100m$, and $P = 10{,}000$.

When the value was put in Equation (4), the optimal number of cluster k_{opt} [9] is 62. Accordingly, the size of key ring in 3DE_sec is around 162 ($\approx 10{,}000/62$).

Assuming that, transmission energy to BS for the authentication of unauthenticated nodes at each round is $n \times E_{Tx} \times Length'$ in the proposed 3DE_sec and $N \times E_{Tx} \times Length''$ in HIKES, for 3DE_sec to be more energy-efficient than HIKES, its total energy consumption should be less and this condition is satisfied in Equation (5).

$$\begin{aligned} n \times E_{Tx} \times Length' &\le N \times E_{Tx} \times Length'' \\ Length' &\le \frac{N \times Length''}{n} \end{aligned} \qquad (5)$$

where, $Length''$ is a size of the each key, and $Length'$ is a size of authentication sample code for authenticating with BS, that is, $\{< idx, BA, length, val >\}^*$.

Table 2. Authentication sample code size according to the key ring size for energy efficiency (P=10,000)

N	p'	**# of non-shared key**	**Authentication Sample Code Size(Max)**	**Length'**
50	0.2222	7,778	≤ 20.5708 bit	38 bit
94	0.5901	4,099	≤ 39.0339 bit	39 bit
95	0.5979	4,021	≤ 39.7911 bit	39 bit
100	0.6358	3,642	≤ 43.9319 bit	39 bit
162	0.9306	694	≤ 230.548 bit	40 bit
200	0.9831	169	≤ 946.746 bit	40 bit
220	0.9929	71	≤ 2253.52 bit	40 bit
250	0.9984	16	≤ 10000 bit	40 bit

Table 2 below shows the number of keys not shared according to the size of key ring and the size of authentication sample code for optimizing energy efficiency when P is 10,000. Here, when the minimum size of key ring is 94, 3DE_sec is more energy-efficient and securer in key management than HIKES. Furthermore, uniqueness can be guaranteed when the authentication sample code is at least 39 bit long.

Table 3. Comparison of the storage requirement for cryptographic primitives

Cryptographic Primitive	LEAP	RKP	HIKES	3DE_sec
Initialization key	0	N/A	1	1
Cluster-wide key	1	N/A	1	1
Node-to-cluster head key	1	N/A	50	1
Node-to-node keys	50	250	50	k
Node-to-sink key	1	N/A	1	1
Global key	1	N/A	1	1
Backup key	N/A	N/A	1	1
Commitment keys	50	N/A	N/A	N/A
Length of Key Chain	20	N/A	N/A	N/A
Size of Key Escrow Table	N/A	N/A	16	N/A
Total Primitives	124	250	121	k + 6

Table 3 compares required storage capacity between existing key management methods and 3DE_sec proposed in this study. When the total number of nodes is 10,000 and the size of key ring (k) is 94, each node uses a space of $100 \times |Key|$ and, as a result, 3DE_sec uses a space, respectively, 19.4%, 60%, and 17.4% less than LEAP, RPK, and HIKES. Here, $|Key|$ indicates key size for uniqueness in the entire pool database, and in case stability within WSN is more important than energy efficiency, it can be achieved through increasing k .

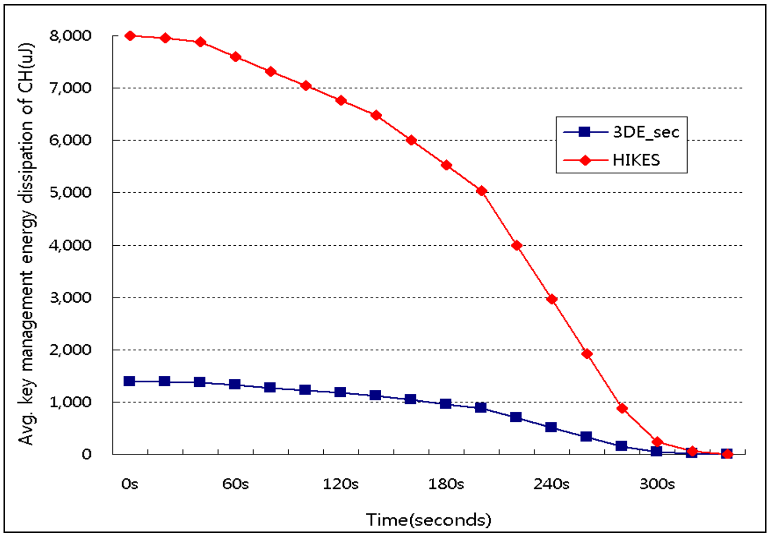


Figure 4. Average Authentication Energy Dissipation of Cluster Head per round

Figure 4 shows the average energy consumption of CH in authentication with BS at each round. When the key size is 16 bits in the key management mechanism for security, 3DE_sec consumes energy around 576.4% less than HIKES, so it enables more efficient energy use throughout the entire network.

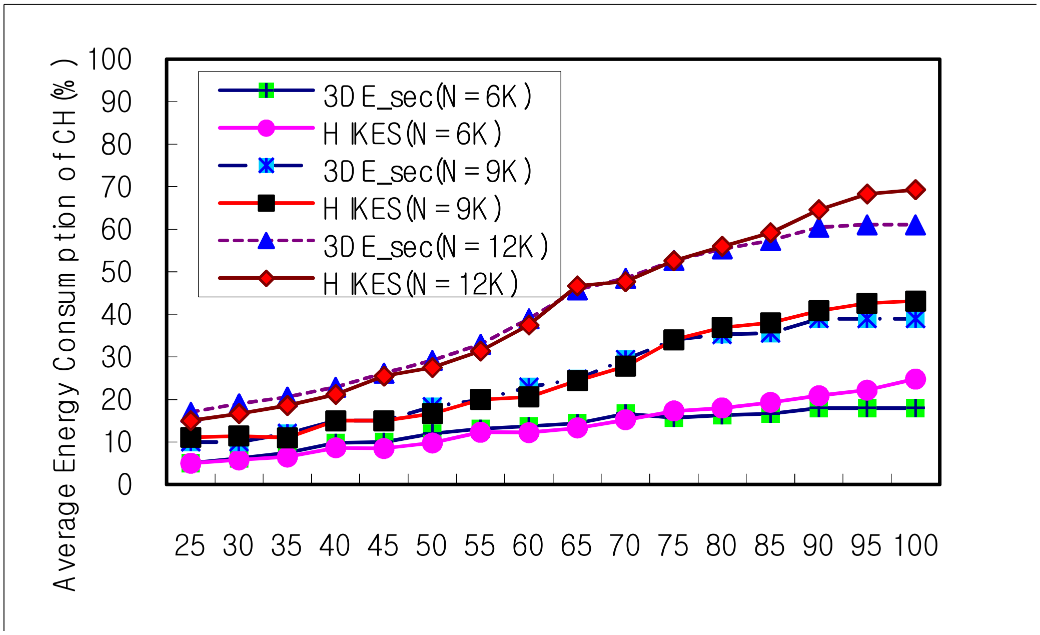


Figure 5. Average energy consumption of cluster head over cluster density.

Figure 5 shows the average energy consumption of cluster head over cluster density. When we simulated the test groups 6,000, 9,000, and 12,000 sensor nodes respectively, 3DE_sec is more performed about 2.1% than HIKES.

5. Conclusions and Future Works

This study proposed a cluster-based key management method that can set a shared key faster and more securely using a multiple-key ring assigned to each node before deployment in cluster formation within WSN. This key management method showed that it can work more energy-efficiently than existing key management methods even when nodes are mobile or new nodes are inserted. In future research on the application of the cluster-based routing protocol, we need to enhance the overall performance of network by minimizing the delay time in cluster formation.

Acknowledgement

"This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) Support program supervised by the NIPA(National IT industry Promotion Agency)" (NIPA-2010-C1090-1031-0004)

References

[1] L. Eschenauer and V. D. Gligor, “A key management scheme for distributed sensor networks”, In 9th ACM conference on Computer and communications security, 2002, pp. 41-47.

[2] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, “A pairwise key pre-distribution scheme for wireless sensor networks”, ACM Transactions on Information and System Security, Vol. 8, 2005, pp. 228-258.

[3] J. Hwang and Y. Kim, "Revisiting random key predistribution schemes for wireless sensor networks," In 2nd ACM workshop on Security of ad hoc and sensor networks, pp. 43-52, 2004.

[4] D. Liu, P. Ning, and R. Li, “Establishing pairwise keys in distributed sensor networks”, ACM Transactions on Information and System Security (TISSEC), Vol. 8, 2005, pp. 41-77.

[5] S. Zhu, S. Setia, and S. Jajodia, “LEAP: efficient security mechanisms for large-scale distributed sensor networks”, In 10th ACM conference on Computer and communication security, 2003, pp. 62-72.

[6] S. Zhu, S. Xu, S. Setia, and S. Jajodia, “Establishing pairwise keys for secure communication in ad hoc networks: A probabilistic approach”, In 11th IEEE International Conference on Network Protocols (ICNP'03), 2003, pp. 326-335.

[7] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks”, IEEE Communications Magazine, Vol. 40,, Aug. 2002, pp. 102-114.

[8] W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks”, Proc. 33rd Hawaii Int'l. Conf. Sys. Sci., Jan. 2000.

[9] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An Application-Specific Protocol Architecture for Wireless Microsensor Networks”, IEEE Trans. Wireless Commun., Vol. 1, no.4, Oct. 2002, pp. 660-670.

[10] J.S. Kim, S.Y. Choi, S.J. Han, J.H. CHoi, J.H Lee, and K.W. Rim, “Alternative Cluster Head Selection Protocol for Energy Efficiency in Wireless Sensor Networks”, In First Software Technologies for Future Dependable Distributed Systems (STFDDS'09), Mar. 2009, pp. 159-163.

[11] M. G. Sadi, D S. Km, and J. S. Park, “GBR: Grid Based Random Key Predistribution for Wireless Sensor Network”, Proceedings of the 11th Annual IEEE International Conference on Parallel and Distributed Systems(ICPADS '05), Vol. 2, Jul. 2005, pp. 310-314.

[12] J. Ibriq and Imad Mahgoub, “A Hierarchical Key Establishment Scheme or Wireless Sensor Networks”, Proceedings of 21st International Conference on Advanced Networking and Applications (AINA'07), 2007, pp. 210-219.

[13] R. M. S. Silva, N. S. A. Pereira, and M. S. Nunes, “Applicability Drawbacks of Probabilistic Key Management Schemes for Real World Applications of Wireless Sensor Networks”, Proceedings of the Third International Conference on Wireless and Mobile Communications (ICWMC'07), 2007.

[14. T. Moore, “A Collusion Attack on Pair-wise Key Predistribution Schemes for Distributed Sensor Networks”, Proceedings of the Fourth Annual IEEE International Conference Pervasive Computing and Communications Workshops (PERCOMW06), Mar. 2006, pp. 13-17.

[15] C. Hartung, J. Balasalle and R. Han, “Node Compromise in Sensor Networks: The Need for Secure Systems”, Technical Report CU-CS-990-05, Jan. 2005.

Authors

Jin-Su Kim was born in Incheon, Korea, on October 9, 1971. He received the B.S. degree from the Incheon University, Korea, in 1998 and M.S. and Ph.D. degrees from the Inha University, Korea, in 2001 and 2010, respectively. Since 2009, he has been a lecture-only professor at the Department of Computer Science Engineering, Yuhan University. His research interest includes cluster based routing protocol, wireless sensor network, data mining, information retrieval, and sensibility engineering.

Jung-Hyun Lee was born in Incheon, Korea, on July 8, 1951. He received the B.S., M.S. and Ph.D. degrees from the Inha University, Korea, in 1977, 1980 and 1988, respectively, all in Electrical Engineering. Since 1989, he is a Professor in the School of Computer Science & Engineering, Inha University. In 1979-1981, he was a researcher at the Korea Institute of Electronics Technology. In 1984-1989, he was an Associate Professor at the Kyonggi University. His research interests are in computer architecture, speech recognition, data mining, HCI, information retrieval, and sensibility engineering.

Kee-Wook Rim was born in Seoul, Korea, on August 22, 1950. B.E. degree in Electronics, Inha University, in 1977. M.E. in Computer Engineering, Hanyang University, in 1987. Ph.D. in Computer Science, Inha University, in 1994. Member of Technical Staff, ETRI, in 1977-1988. Visiting Scholar at University of California, Irvine, in 1988-1989. Director of Computer Systems Division, ETRI in 1989-2000. V.P. of Computer and Software Laboratory, ETRI in 2001-2003. Dr. Rim managed several national projects that are based on the large computer server systems. He is currently a director of ITRC for Embedded Software Development Tools supported by Ministry of Information and Communication, Korea. His research interests are high performance computer systems, operating systems, and database systems.