Application Driven Cluster Based Group Key Management with Identifier in Mobile Wireless Sensor Networks

Eui-Nam Huh, ***Member***, ***KSII*** **and Nahar Sultana**, ***Member***, ***KSII***
Department of Computer Engineering, Kyung Hee University,
1 Seocheon-Dong, Giheung-Gu, Yongin, Gyeonggi-Do 446-701, South Korea
[e-mail: johnhuh, nahar@icns.khu.ac.kr]
*Corresponding author: Nahar Sultana

Received October14, 2007; revised November 28, 2007; accepted November 30, 2007; published December 25, 2007

Abstract

This paper proposes and analyzes a scalable and an efficient cluster based group key management protocol by introducing identity based infrastructure for secure communication in mobile wireless sensor networks. To ensure scalability and dynamic re-configurability, the system employs a cluster based approach by which group members are separated into clusters and the leaders of clusters securely communicate with each other to agree on a group key in response to changes in membership and member movements. Through analysis we have demonstrated that our protocol has a high probability of being resilient for secure communication among mobile nodes. Finally, it is established that the proposed scheme is efficient for secure positioning in wireless sensor networks.

Keywords**:** group key, mobility, sensor network, identity, security

Digital Object Identifier 10.3837/tiis.2007.01.001

1. Introduction

A number of group key management approaches to wired networks have been already studied; however the inefficiency of performance is a bottle neck when they are applied to wireless sensor networks (WSN). A new group key management approach, cluster based group key management, has been proposed, which is quite suitable for WSN. In the case of group communication, when a member joins a group, the group key is re-keyed to ensure that the new member cannot decrypt previous messages. This is a requirement known as backward secrecy. When a member leaves the group, the group key is re-keyed to ensure that future communications cannot be decrypted by the leaving member. This requirement is known as forward secrecy. Conceivably, as the number of group members becomes large, group key management can incur significant overheads and cause a potential system performance bottleneck. Several solutions to the group key distribution problem have been suggested in the literature, which are analyzed as tree based keying mechanisms. The main drawbacks of the previous solutions are: i) network partitioning attack ii) compromising the security of the entire group iii) nodes cannot form groups dynamically, iv) key state inconsistencies for high mobility.

In this paper, we propose a reliable and secure cluster based authenticated group key management protocol which contains an identifier. In our scheme we use a contributory key agreement protocol for key generation which does not require a centralized key server. To introduce the identifier for our scheme we consider bilinear pairings. The contribution of this paper is a comprehensive cluster based group key management protocol which considers node mobility.

The rest of this paper is structured as follows. Section 2 surveys related work, Section 3 introduces group key management and describes the proposed cluster based group key management protocol. Section 4 develops the scheme, the group key management protocol with an identifier. Group ID needs to declare to communicate with gateway. It is demonstrated that a group can declare its ID in the presence of hacker. It is demonstrated that it carries the identifier in the presence of hackers. Section 5 analyzes the security and performance evaluation of the proposed scheme. Section 6 applies this proposed scheme for an existing application. Section 7 concludes the paper.

2. Related Works

An important goal is to provide the so called key independence property [1] which states that knowledge of all (but one) group keys cannot be used to efficiently derive the one missing group key. Many key tree schemes such as [2][3][4] have been proposed for the purpose of minimizing the communication and computation complexity of group re-keying. Most key tree schemes are used in the context of centralized key management and the schemes reduce the cost of re-keying from O(n) to O(log n) (where n is the group size). Zhang et al. [5] examined the effect of mobility on the secure re-keying of group communication by using a hierarchical key distribution framework. Amir et al. [6] demonstrated that group communication systems can be enhanced with security services without sacrificing robustness and performance. Amir et al. [7] presented a performance evaluation of distributed key management techniques integrated with a reliable group communication system. However, the work is mainly targeted at wired networks. Kim et al. [8] proposed a new group key agreement protocol for secure group communications to provide a tradeoff of

computation for communication efficiency. Their work extends a CKA protocol [9] to handle dynamic groups and network failures. Again, such a CKA protocol development can be considered as a special case in which there is only a single region in the group. Rodeh et al. [10] described an efficient algorithm for the management of group keys for a group communication system. Their algorithm is based on the use of a key graph maintained in a distributed and collaborative manner by group members.

2.1 Group key management

Secure group communication requires scalable and efficient group membership management with appropriate access control measures to protect data and cope with potential compromises. Every time a membership change occurs, the group key must be changed to ensure backward and forward secrecy. Group key management must be resistant to a wide range of attacks by both outsiders and rouge members. In addition, group key management must be scalable, i.e., their protocols should be efficient in resource usage and able to minimize the effects of a membership change. There has been a lot of research on group key management in the last decade. Prior work can be roughly partitioned into: Centralized approaches where a key centre is responsible for creating and distributing the keys, and collaborative key agreement approaches for which all members contribute to a group key agreement, and there is no key center. We apply a contributory key agreement protocol for key generation which does not require a centralized key server.

2.2 Contributory group key agreement protocol

This protocol is resilient to changes in group membership. It is based on the GDH contributory key agreement to extend the services of a group communication to provide virtual semantics. This protocol generates group keys based on the contributions of all group members. Particularly appropriate for relatively small collaborative peer groups, these protocols are resilient to many types of attacks. This protocol offers strong security such as key independence and perfect forward secrecy.

When a merge event occurs, the current controller refreshes its own contribution to the group key (to prevent any incoming members from discovering the old group key), generates a new key and passes it to the other members (including all new members and old members). Upon receiving the broadcasted key, each group member (old and new) factors out its contributions and unicasts the result to the controller. The controller collects all of them and adds its own contributions to each of them, builds the set of partial keys, and broadcasts it to the group. Every member can then obtain the group key by factoring in its contributions. If the current controller leaves the group, the most recent remaining member becomes the group controller. In this way the protocol works in the case of a merge event.

3. Proposed Scheme

3.1 Models and notations

The system environment is an asymmetric wireless network, which consists of some mobile sensor nodes with strict computational capability restrictions and a wireless gateway with less restriction. In our proposed scheme low power nodes will form a group and in a group they are separated into clusters to minimize the re-keying cost. This is because it is a group key management. For this scheme, we establish a group identifier. The gateway S has its

own ephemeral private key SK and an ephemeral public key SP. For group initialization, S broadcasts the random key K to all low power nodes which can be denoted by key P. **Table 1** specifies all notations used for our algorithm.

Table 1. Notations

Symbol	Meaning
S	Powerful node and gateway
N	Low power node or mobile node
p	A large prime
Sign	A signing algorithm based on the Elgamal or DSA scheme
Verify	The verification algorithm corresponding to the signing algorithm
H( )	A one way hash function
Z_q^*	Multiplicative groups (q is a prime)
Symbol	**Cluster based key management**
CH	Cluster head
Ci	Cluster head for ith group
K_c	The key by which a Cluster head (CH) communicates with other sensor nodes
K_{ch}	Communicating key between all CH's
K_g	The group key
Symbol	**For Probabilistic analysis of proposed scheme**
k	The number of keys which are used to form a group key
Q	Total number of initialization keys
j	The collusion threshold of the network

The proposed scheme adopted signature technique, providing mutual authentication. In summary the proposed model is as follows:

1. Generate random bits x ЄR Z^*_q
2. Compute public key,$y_i = g^x \bmod p$ [for node i]
3. Publish signature,m_i= Sign(SK_i, y_i)
4. For each (y_i, m_i) send to S
5. S checks m_i by Verifying (SP_i, y_i, m_i)
6. Go to Group initialization
7. Nodes take place in cluster
8. CH computes group ID and send to S
9. S communicates by group ID and contact with outside.

Group initialization:

1. for each node N_i
   a) generate random key K_{Ni}ЄR Z^*_q [for node i]
   b) P_{Ni}= H(ID_{Ni}||K_{Ni})[apply hash function and ID_{Ni} is the identity for each node]
   c) Provide N_i with :{P_{Ni}}

Most of the group key distribution solutions focus more on group join and group leave. This invariably leads to the concept of a key distribution centre or a similar central authority

to control a group. In dynamic mobile environments like a battlefield etc., small groups need to form for brief periods on a dynamic basis.

In this paper, we argue for mobile environments with high mobility and high probabilities of link failures. It is more prudent to have node based localized solutions to the problem instead of a tree based structure. **Fig. 1** depicts the structure of a cluster based group key management. The communication between members in a group and the situation in the case of joining, leaving etc. is explained here. In the proposed scheme, the group key can be derived as follows; Kg=MAC (Kch, c). Here, c is a counter when a group member changes. For this protocol, we consider a contributory group key protocol as a hierarchical group key management which can efficiently and securely distribute the keys.

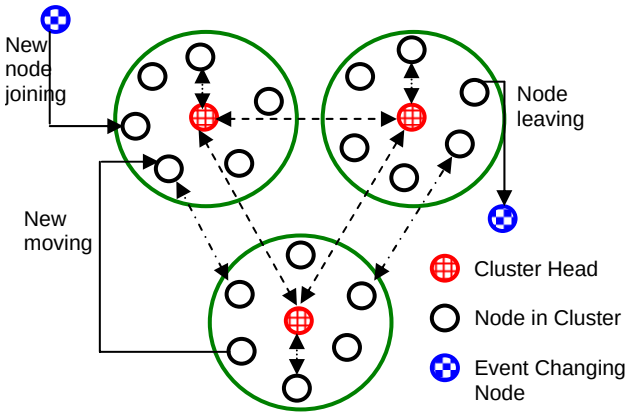


Fig. 1. Cluster based group key management

3.2 Node mobility

Joining case: When a new member proposes to join, it beacons the message with its id. The beacon message is received by neighboring nodes, which forward it to the CH. It may also be received by the CH directly. The CH broadcasts this message both to other members and to other cluster heads. When a new member joins, a new cluster key Kc is generated and distributed by the CKA protocol. All CH's also regenerate Kg. The steps for joining are shown in **Fig. 2.**

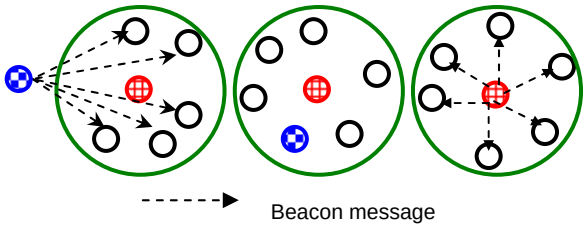


Fig. 2. Node joining

Leaving case: When a node aims to leave, then it sends a beacon message to the corresponding CH or all members. It is intending to leave as shown in **Fig.3**. Here, the CH updates the intention information to its members. Since a group leave event instigates group membership will be changed, which results in a new cluster key *Kc* being generated and

distributed by the CKA protocol to other members. After all CHs receive the information on the current leave event, they also broadcast the changed view to all of their members. Finally, all CHs autonomously regenerate the group key *Kg* and distribute it to their corresponding members.

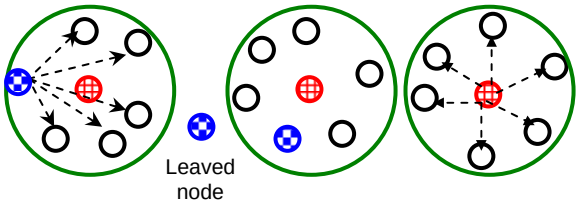


Fig. 3. Node leaving

When a CH leaves: When a CH leaves the group, it sends a beacon message to all of the members like **Fig. 4**. Thus, in addition, all operations are required in the above case for the member leave; a new CH is elected to replace the leaving CH. Since this involves a leadership change, the new *Kc* and *Kg* will be generated by CKA protocol and distributes the new key to the members.

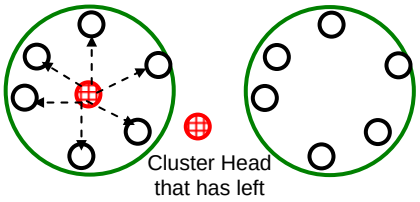


Fig. 4. Cluster head leaving

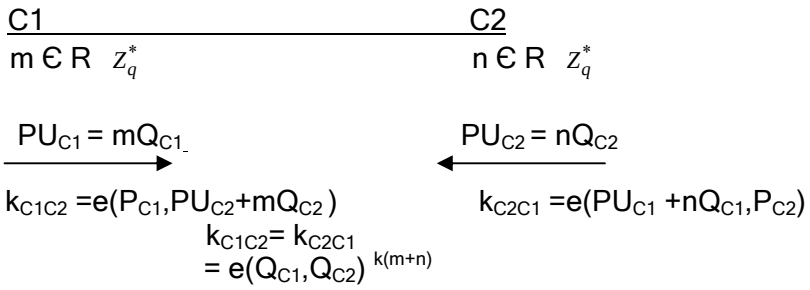
How a new cluster head will be selected: In the CKA protocol, it is possible to store the documents as an output view. There are two views. One is the cluster view and the other is the group view. When a CH leaves, then the most recent added node can be identified using the cluster view. That will be the new cluster head. In addition, the new cluster head beacons the message to other members to inform them that it is the new leader.

Cluster member connection and disconnection: There will be a system whereby each member sends a message to its CH after some period of time. If CH does not receive it, within a defined time period (suppose it is 10 seconds), CH updates that node as a leave node. If that node aims to reconnect then it uses the join operation. We discuss the group identifier in the next section.

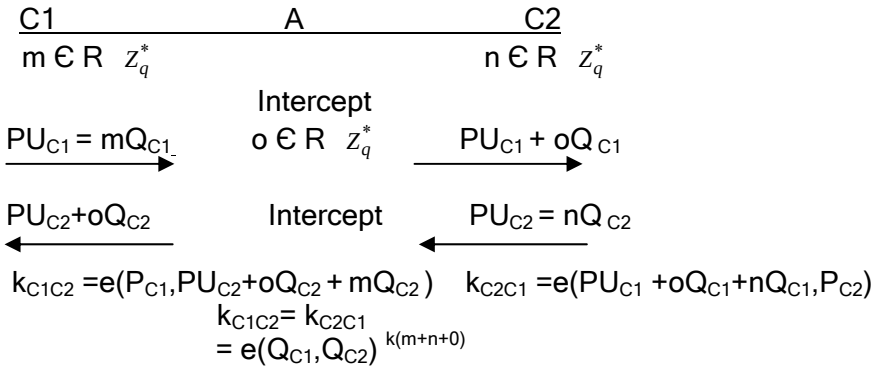
4. Group Identifier

In this section we explain the situation in which a group can declare it's ID. Here, we assume that the sender cluster head and receiver cluster head have a private key and a different ID. After running the following protocol when they have the same session key, both the participating clusters and the other cluster head in that group (because all cluster heads share a common key) get the same session key. For one session, each cluster in a group has the same session key and, the group declares its ID. This identifier becomes known to all its

cluster members. Here, we follow the Chen Kudla protocol which is mainly an ID based protocol. However we consider this protocol for our scheme, in order to create the same session key for all members, and we demonstrate that it also works in the presence of hackers. Now, we assume that C1 and C2 is the sender and receiver cluster head, respectively. The private keys of C1 and C2 are $P_{C1} = kQ_{C1}$ and $P_{C2} = kQ_{C2}$ (k is a generator of private key, $k \in Z^*_q$). Here, $Q_{C1} = H(ID_{C1})$ and $Q_{C2} = H(ID_{C2})$. The Ephemeral public key of the sender cluster head C1 is $PU_{C1} = mQ_{C1}$ and in the case of C2 is $PU_{C2} = nQ_{C2}$. (m and n are the ephemeral private keys). Here *e* denotes bilinear pairings. The following steps show how the protocol works:



Here it is the end of the execution of the protocol. Both C1 and C2 have the same session key. They have $Sk_{C1C2} = H(K_{C1C2}) = Sk_{C2C1} = H(K_{C2C1})$. We are going to demonstrate the execution of the protocol in the presence of hacker A. This time both C1 and C2 have the same session key. They have $Sk_{C1C2} = H(K_{C1C2}) = Sk_{C2C1} = H(K_{C2C1})$. It can be demonstrated that in the presence of a hacker, they can achieve the same session key. The group can declare its own ID. The ID of each group is stored in the gateway for communication. Here is the protocol in the presence of hackers;



5. Security Analysis and Performance Evaluation

5.1 Security analysis

For this group key management we have to ensure forward secrecy, backward secrecy, and key independence.

Implicit key authentication: If a hacker wants to obtain a group key, he must be first compute the key K_{ch} which is shared by all cluster heads. However, without knowing one of

private keys of both parties, a passive hacker cannot compute the shared key by eavesdropping on the system, assuming the communicating key between clusters is not revealed. It is also computationally infeasible for a passive hacker to discover any group key since we generate a group key using a secure MAC.

Forward secrecy: If a hacker can compromise any node and obtain its key, it is possible that the hacker can start new key agreement protocol by impersonating the compromised node. For our scheme we can conclude that a passive hacker who knows a contiguous subset of old group keys cannot discover any subsequent group key. In this way, forward secrecy can be achieved.

Backward secrecy: A passive hacker who knows a contiguous subset of group keys cannot discover how a previous group key is changed upon a group join or leave. A passive hacker who knows a proper subset of group keys cannot discover how any other group key is guaranteed, since our group key is generated using a secure MAC with two different inputs (Kch, c).

5.2 Performance analysis

In the following steps, we analyze the computational complexity and the communication cost of the proposed protocol. For convenience, the following notations are used to analyze the computational complexity and the communication cost. T_{SIG} is the time for computing one signature; T_{VER} is the time for verifying one signature, T_{EXP} is the time for modular exponentiation; T_H is the time for computing one hash function.

Step1: When mobile nodes are in an anonymous situation in a network and they form a group then the computational cost for the whole step is $nT_{SIG} + T_{EXP} + nT_H$, where n is the number of nodes.

Step2: Here, we have to find the cost of the gateway node S. Initially it generates its ephemeral secret key for the sign function then it uses the verify function. For the last stage it uses modular multiplication to receive the group identifier. So the total cost of the gateway node is $nT_{VER} + (2n+1)\ T_{EXP} + n\ T_{MUL.}$

As shown in the **Table 2** comparison model, it is clear that the computational complexity of each low power node in the proposed protocol is larger than that in Bresson et al.'s protocol. However, Bresson et al.'s required extra computation costs to ensure that each low power node was authenticated by the powerful node. The important point is that the proposed protocol provides better security than Bresson et al.'s protocol.

Table 2. Comparison between two authenticated group key agreement protocols

Items	Bresson et al's protocol	Proposed protocol
Forward secrecy	No	Full
Contributory group agreement	No	Yes
Implicit key authentication	No	Yes
Number of rounds	2	2
Computational Cost required by mobile node	$T_H + T_{MUL}$	$nT_{SIG}+T_{EXP}+nT_H$
Computational complexity required by powerful node	$nT_{VER}+ T_{EXP} +(n+1)\ T_H$	$nT_{VER}+(2n+1)$ $T_{EXP}+nT_{MUL}$

To provide perfect forward secrecy and other security requirements we can demonstrate some modifications of Bresson et al.'s protocol. i) During the initialization phase when the powerful node signs the message c || $\{K_i\}_{i \in Ic}$ || y_s to obtain the signature and broadcasts (c,$\{K_i\}_{i \in Ic}$ || y_s, sign) to the low power node, ii) Upon receiving (c,$\{K_i\}_{i \in Ic}$ || y_s, sign) each low power node verifies the signature and computes $\alpha_i = y_s^{xi}$. iii) Then it recovers the shared secret value and computes the same session key. For this modification each α_i is derived as a function of the ephemeral random values x_s and x_i contributed by both the powerful node and low power node. If we measure the computational complexity which is required by the mobile node (the low power node) it needs one more T_{EXP} and T_{SIG}.

From the modified version, it is observed that each run of the improved protocol computes a unique α_i and $H(c||\alpha_i)$ even if y_i is replayed by the hacker. This halts the hacker and provides the main security. Thus, the conclusion is that to provide full security for our scheme the computational complexity is low, as is also the case in Bresson et al.'s protocol.

To provide full security in Bresson et al.'s protocol, the computational complexity of the low power nodes is $T_H + T_{MUL} + T_{EXP} + nT_{SIG}$ and the computational complexity of the powerful nodes is $nT_{VER} + nT_{EXP} + (n+1)T_H + T_{SIG}$. It can be concluded that by comparison with Bresson et al's protocol, our proposed scheme is better.

5.3 Efficiency of our scheme

In this section we will only discuss the efficiency of our cluster based group key management. Here we describe the communication efficiency by demonstrating the communication cost of our proposed cluster based scheme.

Communication cost during join: In our structure, when a new node aims to join the group, it has to send the beacon message directly to the cluster head or cluster members. Here the new member has to send 2 messages. As for the member cluster join, the communication cost is $\log_2(nm)+2$, and *nm* is the number of members in the cluster. Consider the join cost of a new member, for example, the 1024th member. In our approach when a new member joins the cluster, the overhead is $\log_2(64)+2=8$. Thus, the conclusion is that our approach achieves a significant improvement in communication cost, of 20% for a new node join.

Communication cost during leave: There are two types of leaving cost in our approach. 1) Cluster head departure 2) Cluster member departure. In the case of the cluster member leaving the communication cost is $\log_2(nm)+2$, the summation of the rekeying cost. The number of messages sent to the cluster head is twice what it was previously. When the current cluster head leaves the group then the communication cost is $\log_2 nm$. We consider the same example as described above to explain the situation further. In our proposal, when the member of a cluster leaves, the communication cost is $\log_2(64)+2=8$. When the cluster head leaves, then the communication cost is $\log_2 64=6$. From this example, it can be established that our proposal improves the communication efficiency by 20% and 40%, respectively.

6. Application

In case of localization, distance reduction and distance enlargement attacks may make the false location of sensor from its true location. In this case, proposed scheme can be efficiently applicable for secure positioning. The detection of the attack can be established,

which is based on the proposed approach. To apply our solution in case of secure positioning, we have demonstrated some analysis in this section.

Wormhole attack: The wormhole attack is a relay type of attack where an hacker relays information transmitted at one part of the network to some distant part of the network, thus violating the geometry of the network and the communication range constraint. To mount a wormhole attack, the hacker initially establishes a direct link, referred to as a wormhole link, between two points in the network. The wormhole attack is very difficult to detect, since it is launched without compromising any host, or the integrity and authenticity of the communication.

Defense against wormhole attack: In the case of the wormhole attack, cryptography is used to secure the beacon transmissions and the source of the information is authenticated to defend against it. In our group key management, there are shared keys between cluster heads and there is also a key which is shared between the cluster head and other sensor nodes. With this shared key the locator can broadcast the localization information .If the shared key is compromised, at that time sensors are able to detect attacks using a consistency check [2]. Our scheme is authenticated because we have applied a strong cryptographic solution and we have made it an ID based scheme, which provides the authentication. Here, we do not provide any mechanism to prevent the attack – the system just analyzes the position of a sensor in the presence of attack.

The impersonation attack: With respect to a local process, the hacker impersonates reference points and injects bogus local information in the network. In this type of attack the hacker must compromise the cryptography to the degree necessary to prove its impersonated IDs to the nodes under attack. Thus, the nodes properly authenticate an hacker as a trusted source.

Defense against the impersonation attack: Assume that sensor authenticates the set of locators but also detects that they are under attack. In our scheme, the following steps can be taken:

- For finding the local sensors always broadcast the nonce and its ID
- Every locator receiving the broadcast of the sensors replies with a beacon that includes local information.
- The nonce is encrypted with the shared key instead of the broadcast key and sensors identify the locator which first replies with an authenticated message.

In our scheme we use a secure MAC to generate the group key and this group key cannot be compromised by any hackers. This group key controls the implementation of the whole scheme and it also works for mobile sensors. This group key works for event change operation which is one of the important parts of the scheme. In addition, if the locators are mobile, our scheme will able to find the location of the sensors.

6.1 Analysis of our scheme for this application

Here we analyze our scheme with attack probabilities by analyzing the consequences of the collusion between *t* compromised members. In the following discussion we demonstrate that even after *t* maliciously colludes; a group is only probabilistically compromised. Here let the number of ways that α unique identifier can be produced be V (k, Q, α) where k and Q were initialized previously.

$$
\begin{aligned}
V(k, Q, \alpha) &= \text{Total number of ways of choosing exactly } \alpha \text{ identifier} \\
&=^{Q} C_{\alpha}(\alpha^{k} - \sum_{i=1}^{\alpha-1} \text{cases where } i \text{ identifiers appear}) \\
&=^{Q} C_{\alpha}(\alpha^{k} - V(k,\alpha,\alpha-1) - V(k,\alpha,\alpha-2) - \ldots\ldots - V(k,\alpha,1)) \\
&=^{Q} C_{\alpha}(\alpha^{k} - \sum_{i=1}^{\alpha-1} V(k,\alpha,i)) \qquad (1)
\end{aligned}
$$

In section 4 it was discussed for which situation (when the clusters of a group have the same session key) the group can declare its identifier. The probability of producing exactly α unique key identifier is,

$$Pv(k,Q,\alpha) = V(k,Q,\alpha)/Q^{k} \qquad (2)$$

The average number of unique identifier A (k, Q) produced is,

$$A(k,Q) = \sum_{j=1}^{k} Pv(k,Q,i).i \qquad (3)$$

Let us assume that t nodes have been compromised. In order to compromise a group with j unique keys, all the t nodes should have the shares for j keys.

This will happen with probability P_J. Thus the probability $Pc\ (t)$ of a compromising group is,

$$Pc(t) = \sum_{j=1}^{k} Pv(k,Q,j)(p^{j})^{t} \qquad (4)$$

More generally, if T nodes are compromised, the probability $Pc(T)$ of compromising a specific group is,

$$Pc(T) = \sum_{j=1}^{k} Pv(k,Q,j) \sum_{r=t}^{k} {}^{T}C_{r}(p^{j})^{r}(1-p^{j})^{T-r} \qquad (5)$$

This equation (5) is plotted in **Fig. 5(a)** and **5(b)**. It depicts the probability Pc (T) with respect to P_J for varying T and t, respectively. Considering a network as a sample space, here r defines the equality of the collusion threshold of the network t. Even after more than t members have been compromised, a group is only probabilistically susceptible.

In addition, these probabilities are low for significant portions of the entire (p-t) or (p-T) sample space. The probability of compromising a specific group is low. This is a substantial result, since our scheme has a high probability of being resilient even to the compromise of t nodes. As shown in **Fig. 5(a)**, the scheme is resilient as the number of nodes increases. The probability of compromising a specific group is very low. **Fig. 5(b)** shows that the increasing number of nodes leads to an increase in the probability of collusion.

Here, we put the threshold t values for varying. As can be seen, for lower values of t, the probability of compromised node detection is very early in the process. If the values of t are too small, it can be easily recognized as a compromised node and it may increases the false detection. If the values of t are too large, it is difficult to detect the compromised node and it also increases the probability of collusion. Here, it is observed for the values of t are 2, 4, and 6,8,10. The proposed scheme achieves high probability to detect compromised node for

lower values of *t*. So from our analysis, it is proved that, the proposed scheme is really efficient which is only probabilistically compromised and also detect the compromised node. Also, it can be said that, the proposed scheme is applicable for secure positioning in a group communication.

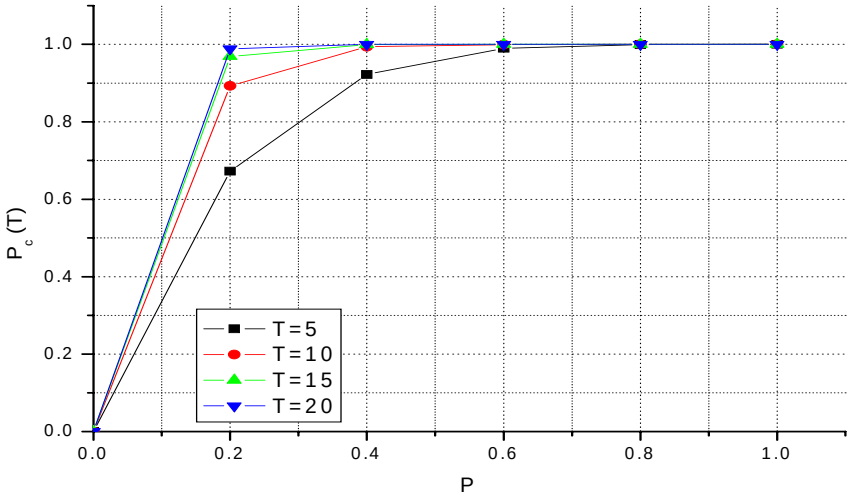


Fig. 5(a). Pc (T) vs. P_j varying T

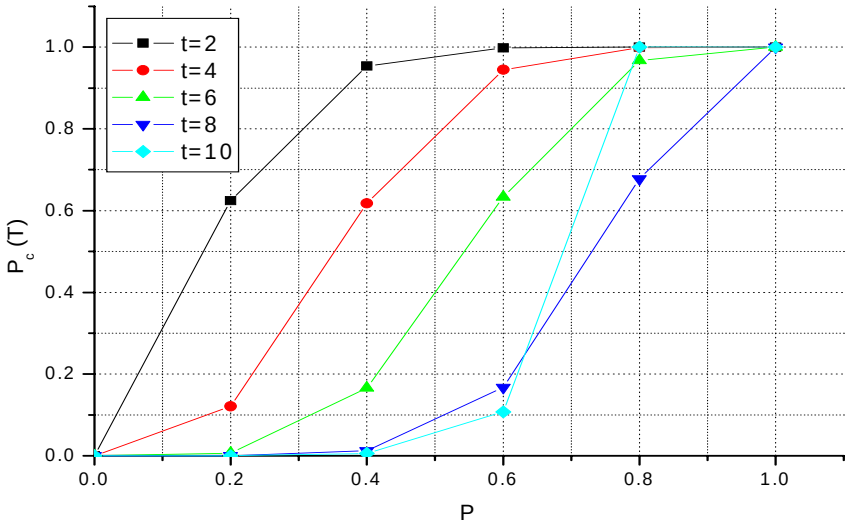


Fig. 5(b). Pc (T) vs. P_j varying t

7. Conclusion

In this paper, we have proposed and analyzed a scalable and efficient cluster based secure group key management protocol to support secure communication in mobile WSNs. By

using a cluster based hierarchical key management technique; the proposed group key management protocol both reduces network communication cost and provides robust security. In addition, through simulation it is established that the proposed scheme has a high probability of being resilient in group communications in mobile WSNs. To provide mutual authentication the scheme is ID based. In the future, we plan to explore our protocol for group wise mobility, which will make it more efficient for real world applications.

Acknowledgement

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2006-C1090-0603-0040).

References

[1] M. Steiner, G. Tsudik and M. Wadner, "Cliques: A new approach to group key agreement," *Proc. IEEE ICDCS 1998*, pp. 380-387, 1998.
[2] D. Balenson, D. Mcgew, and A. Sherman, "Key management for large dynamic groups: One way function trees and amortized initialization," IETF, Feb 1999.
[3] C. Wong, M. Gouda and S. Lam, "Secure group communications using key graphs," *Proc. ACM SIGCOM 1998*, pp. 68-79, Sep. 1998.
[4] B. Aruna, M. Sumita, and R. Sridhar, "Analysis of a hybrid key management solution for adhoc networks," *IEEE Communication Society*, *WCNC*, 2005.
[5] C. Zhang, B. Decleene, J. Kurose, and D. Towlesy, "Comparison of inter area rekeying algorithms for secure wireless group communication," *Performance Evaluation,* vol.49, no. 1-4, pp. 1-20, 2002.
[6] Y. Amir, C. Nita Rotaru, and G. Tsudik, "Secure spread: An integrated architecture for secure group communication," *IEEE Transactions on Dependable and Secure Computing,* vol. 2, no. 3, pp. 248-261, 2005.
[7] Y. Amir, Y.Kim, C. Nita –Rotary, and G. Tsudik, "On the performance of group key agreement protocols," *ACM Transactions on Information and System Securit*, vol. 7, no.3, pp. 457-488, 2004.
[8] Y. Kim, A. Perrig and G. Tsudik, "A common efficient group key agreement," *Proc. IFTP-SEC 2001*, pp. 229-244, 2001.
[9] M. Steiner, G. Tsudik and M. Waidner, "Diffie hellman key distribution extended to group communication," *3rd ACM Conference on Computer and Communications Security*, Jan 1996.
[10] W. Ashraf, S. Olariu, and M. Etoweissy, "Scalable cryptographic key management in wireless sensor network," *Proc. 24th International Conference on Distributed Computing System Workshops*, 2004.

Eui-Nam Huh has earned BS degree from Busan National University in Korea, Master's degree in Computer Science from University of Texas, USA in 1995 and Ph. D degree from the Ohio University, USA in 2002. He was a director of Computer Information Center and Assistant Professor in Sahmyook University, South Korea during the academic year 2001 and 2002. He has also served for the WPDRTS/IPDPS community as program chair in 2003. He has been an editor of Journal of Korean Society for Internet Information and Korea Grid Standard group chair since 2002. He was also an Assistant Professor in Seoul Women's University, South Korea. Now he is with Kyung Hee University, South Korea as Professor in Dept. of Computer Engineering. His interesting research areas are: High Performance Network, Sensor Network, Distributed Real Time System, Grid Middleware, Monitoring, and Network Security.

Nahar Sultana received the B.S. degree in Computer Science from American International University Bangladesh (AIUB), in 2006. She is now MS candidate in the Department of Computer Engineering, Kyung Hee University, South Korea. Her interesting study areas are: Authentication, Key Distribution, and Localization in Wireless Sensor Networks.