Securing Cluster Head Elections in Wireless Sensor Networks

Gicheol Wang and Gihwan Cho

***Abstract*—In wireless sensor networks, since a CH gathers data from members and delivers the gathered data to the sink, preventing a compromised node from being a CH is very important. Even though unveiling the CH election process enhances the security of network, it cannot prevent compromised nodes from declaring themselves as CHs without qualification. In this paper, we propose a scheme which identifies the compromised nodes by evaluating the trust level of members and excludes untrustworthy nodes every CH election round. Our analyses show that our scheme outperforms the scheme which only unveils the CH election process without filtering.**

***Index Terms*—Secure cluster head election, trust-based election, clustering, wireless sensor network.**

I. Introduction

A cluster structure is generated by transforming a physical network into logical groups of nodes which are called clusters [1]. Wireless sensor networks frequently adopt the cluster structure to save energy consumption of nodes and to extend the network longevity. If a cluster needs a local coordinator which is called CH (Cluster Head), a node is selected as a CH among the members. Because a CH node not only gathers the data from the normal nodes but also sends the gathered data to the sink, it becomes a compromise target for attackers [1]. Even worse, if all CHs are compromised by attackers, the attackers get the whole control of the network. For such a reason, CHs should be changed as frequently as possible and it is desirable to elect a new CH in a periodic manner. That is why many secure CH election schemes [2]-[5] have been proposed up to now.

Recently, Holczer *et al*. proposed an anonymous CH election scheme [5] where a member never knows which node is going to be a CH except the existence of a CH declaration node. However, this scheme unfortunately allows a compromised node to arbitrarily declare itself as a CH. This is because this scheme does not evaluate the trust level of members and does not evict the untrustworthy nodes from CH candidates. Even worse, if a node is compromised by an attacker, the compromised node can continuously declare itself as a CH in later election rounds.

Manuscript received November 20, 2013; revised April 2, 2014. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (KRF) funded by the Ministry of Education, Science and Technology (2012R1A1A2042035).

Gicheol Wang is with the Agency for Defense Development, Daejeon 305-152, Korea (e-mail: gcwang@ add.re.kr).

Gihwan Cho is with the Div. of Computer Science and Engineering Cloud Open R&D Center, Chonbuk Nat'l University, Jeonbuk, Jeonju 561-756, Korea (e-mail: ghcho@chonbuk.ac.kr).

In this paper, we try to resolve the problem as follows. First, our scheme evaluates the trust value of members by tallying their CH fulfillment frequency. Since the compromised nodes undoubtedly try to become a CH, their CH fulfillment frequency and trust value are going to become smaller with the lapse of time. For every CH election round, our scheme expels some untrustworthy members from CH candidates to mitigate the threat of compromised nodes.

We organize our paper as follows. Section II briefly describes previous work dealing with secure CH election. In Section III, the network and threat model is presented and the detailed explanation of our scheme is provided in Section IV. We provide the analyses of our scheme and Holczer's scheme in Section V and finally we draw conclusion in Section VI.

II. Related Work

Sirivianos *et al.* proposed a scheme where each member generates a random number and delivers the random number to other members in order to share a common sum [2]. Each member divides the common sum by the number of members and settles the remainder as the index of the CH node. According to the generation and distribution method of the common sum, the scheme is divided into Merkle's puzzle based scheme, commitment based scheme, and seed based scheme.

In Merkle's puzzle based scheme, the current CH first establishes pairwise keys with its members using the Merkle's puzzle. Then, the first member creates a random number and encrypts it using the pairwise key before delivering it to another member. Upon receiving the encrypted random number, the receiver adds its own value to the received value and passes the sum to another member. This procedure repeats until all members add their own value to the received value. The last member broadcasts the total sum and the current CH distributes members' pairwise keys to all members. Then, all members convert the total sum into a plain sum using the pairwise keys and the double additively homomorphic encryption. Note that the plain sum is employed as a common sum.

In the commitment based scheme, each member pair shares a unique pairwise key. For every CH election round, each member transmits its commitment to other members in the P2P manner. Here, the commitment is a random number which is encrypted using a shared pariwse key. Then, each member sends the proof for its commitment ownership (that is, original random number) to other members. Members verify the random numbers using the shared pariwse keys and sum them to make a common sum.

In the seed based scheme, each member generates an initial random number and broadcasts it. For every CH election

DOI: 10.7763/JACN.2014.V2.120

round, each member broadcasts an availability message which expresses the willingness of its join in the current CH election. Members receiving the message maintain the list of the availability message senders and generate a new random number of the senders with their initial random number and the number of election rounds. Then, members obtain the sum of the new random numbers and it is employed as a common sum.

Merkle's puzzle based scheme causes a lot of overhead due to the pairwise key establishments, generation of the common random value, and the distribution of pairwise keys. The commitment based scheme and the seed based scheme are vulnerable to the intentional transmission avoidance and selective transmission of random number or availability message. Intentional transmission avoidance of random number or availability message changes the CH election result. Besides, selective transmission of random number or availability message makes multiple common sums in a cluster and splits a cluster into multiple ones.

Dong *et al*. proposed a CH election scheme which strongly prevents external attackers from joining in CH elections. The scheme's ability highly relies on their ID assignment scheme which tightly couples a node's ID, its commitments, and its polynomial shares [3]. In this scheme, members which do not distribute a participation message for CH election or explicitly distribute a non-participation message are removed from the CH candidates. The real CH is selected in a round robin manner from the rest of the candidates. In this scheme, a compromised node can readily change a CH election result by avoiding the transmission of its participation message. The compromised node can also produce multiple election results by transmitting its participation message to only a part of CH candidates. Even if this scheme has a recovery algorithm combining more than one result into single result, it premises the spontaneous cooperation of the CH candidates. However, compromised nodes will not cooperate with it.

Buttyan *et al*. proposed a scheme which veils the election process from external nodes using the message encryption and decryption [4]. However, a compromised node can easily know the election result. Moreover, the compromised node can declare itself as a CH regardless of its suitability.

Holczer *et al.* proposed a completely hidden election scheme in [5]. This scheme consists of two steps. At the first step, each member elects itself as a CH according to the probability that a member becomes a CH. At the second step, each member checks if there is a member which elected itself as a CH. The check reveals only the existence or inexistence of CH node and none of the members knows which node is the CH in the cluster due to the difficulty of discrete logarithm. Even though the authors in [6] claim that the Holczer's scheme is one of the best CH election schemes in terms of security, it also has the same vulnerability as [5]. That is, a compromised node can declare itself as a CH without any constraint.

III. Network and Threat Model

A. Network Model

We assume that nodes securely form clusters to perform the energy-efficient TDMA communication after they are deployed in the mission field. After nodes formed the clusters, their network operation is split into multiple rounds and each round is again split into three steps. Those steps are candidate reselection, secure CH election, and data aggregation and forward as depicted in Fig. 1. In this paper, we only focus on those steps after the secure cluster formation. Please refer to [1], [7], [8] concerning the secure cluster formation. In the candidate reselection step, nodes pick up behaving nodes among all members in their cluster. In the secure CH election phase, nodes elect a node as the CH among the CH candidates. In the data aggregation and forward phase, nodes deliver their reading to the CH and the CH combines the readings and sends the combined data to the sink.

B. Threat Model

We assume that a compromised node arbitrarily declares itself as a CH during the CH election process. The effect of this malicious behavior is definite. The compromised node can play as a CH until it is recognized as an unqualified CH by other normal nodes and expelled from the network. Since the compromised node can get a great benefit from playing as a CH, it must try to sustain the CH role as long as possible. To our best knowledge, there is no way to prevent the malicious action. The only way to prevent the misbehavior is to exclude the compromised node prior to the CH election.

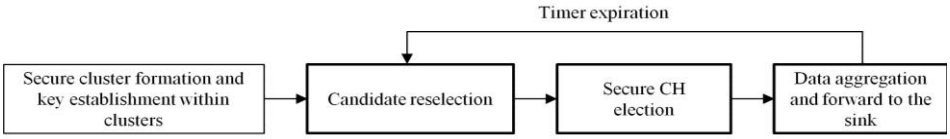

Fig. 1. Network operation of the proposed scheme.

IV. Securing Cluster Head Elections Using Reselection of Candidates

We assume that external attackers are excluded from the network during the initial cluster formation process. Some cluster formation protocols [1], [7], [8] can be employed for that purpose. We also assume that all pairs of members in a cluster have established a pairwise key with each other. Therefore, the attackers in the remaining of this paper mean the compromised nodes. They are going to intentionally declare themselves as a CH in every CH election opportunity. In a round, our scheme consists of two steps. The first step picks out some candidates among all members in the cluster. To pick out some behaving nodes among all members, each member computes the trust value of other members in the cluster. The trust value is calculated by considering the previous trust value and the CH role fulfillment frequency. Note that the initial trust values of all members are one. Then each member averages the current trust values of members and excludes the nodes whose trust value is lower than the

average from the candidates. We detail the first step in Section IV Part A. In the second step, the remaining time of the current round is divided into multiple frames and each frame is divided into CH election and fulfillment periods and the transmission period from CH to the sink. The number of CH election and fulfillment periods is equal to the number of members in the cluster. Besides, time length of the CH fulfillment period is exactly equal to the assigned time slot of each member and each member comes to know the assigned time slot when they have to wake up for the transmission during the initial cluster formation process. Fig. 2 shows the timeline of our scheme's network operation. Note that all members wake up every CH election period and determine whether it is going to play as a CH during the following CH fulfillment period and the determination relies on the CH winning probability which is explained in the following subsection. If a node decides to play as a CH during the following CH fulfillment period, it announces the result through a broadcast message to prevent the CH declaration of other members. Then, it plays the CH role during the CH fulfillment period. Besides, a node that is scheduled to transmit in the assigned time slot (that is, a CH fulfillment period) transmits its data to the declared CH. At the end of a frame, all CH role nodes in the current frame transmit the received data to the last CH role node in the frame and the last CH role node aggregates them and sends it to the sink. To provide the confidentiality of the transmissions between a member and the CH and the transmissions between CHs in a cluster, the pre-established pairwise keys can be used. We detail the third step in Section IV Part B.

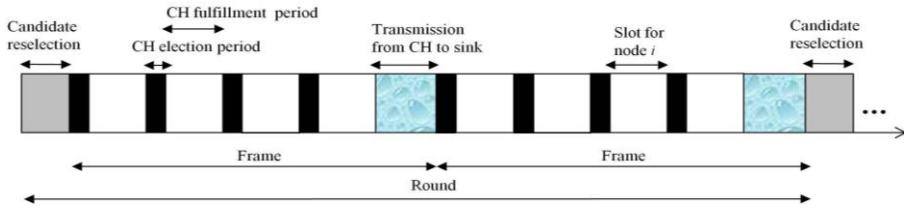

Fig. 2. Timeline of the proposed scheme's network operation.

A. Reselection of CH Candidates

At the beginning of each round, each node evaluates the trust level of other members in the cluster. To quantify the trust level of a node, we introduce the variable T_k^i which means trust value of node *i* in the round *k*. All values of T_1^i is one. A node's trust value in the previous round is computed using the expected CH role frequency (E_{CH}^i) in the previous round and real CH role frequency (F_{CH}^i) in the previous round. The expected CH role frequency is computed by (1) and P_{CH}^i and n_{CH} mean the node *i*'s CH winning probability and the number of CH candidates respectively. The node *i*'s CH winning probability (P_{CH}^i) is computed using the node *i*'s CH winning frequency and the number of members as shown in (2).

$$E_{CH}^i = \lfloor P_{CH}^i \times n_{CH} + 0.5 \rfloor \quad (1)$$

$$P_{CH}^i = \left(\frac{1}{n_{CH}} \times \frac{1}{F_{CH}^i + 1} \right) \quad (2)$$

Then, each member can compute the previous trust value of other members (T_{k-1}^i) using E_{CH}^i and F_{CH}^i as shown in (3). Using T_{k-1}^i and N_{CH}^i which means the total frequency of node *i*'s CH role fulfillment, each node can compute the trust value of other members (T_k^i) as shown in (4). So, a node's current trust value is proportional to its previous trust value and inversely proportional to the total frequency of the node's CH role fulfillment. Note that F_{CH}^i is reset at the beginning of every round while N_{CH}^i is never reset.

$$T_{k-1}^i = \frac{1}{\max[F_{CH}^i - E_{CH}^i, 0] + 1} \quad (3)$$

$$T_k^i = \frac{T_{k-1}^i}{N_{CH}^i + 1} \quad (4)$$

Last, each node averages the values of T_k^i and compares the average trust value with the trust value of other members. The members whose trust value is lower than the average trust value are excluded from the CH candidates. If a node is excluded from the CH candidates, other members in the cluster set the node's current trust value to the node's previous trust value.

B. CH Elections and Data Aggregation

1) CH elections

After filtering out some suspected nodes from CH candidates, each node first goes to sleep and periodically awakes to elect itself as a CH according to its CH winning probability (P_{CH}^i) and fulfills the CH role during the following CH fulfillment period in case of the winning. The combination of CH election and CH fulfillment process repeats as the number of members in the cluster. The advantage of this approach is rotating the CH role node among the candidates so that a compromised CH can get a little amount of data from members during a short time. So, the CH winning probability (P_{CH}^i) is inversely proportional to real CH role frequency (F_{CH}^i) to equalize the opportunity of being a CH as shown in (2). Namely, whenever a member elects itself as a CH, its CH winning probability decreases as $1/F_{CH}^i + 1$. If no CH is elected in a CH election period, each member recalculates the current trust value of all members and the most trustable node becomes the CH. If multiple nodes have the same trust value, ID is a tie breaker. That is, a lower ID is preferred as the CH.

2) Data Aggregation

In a CH fulfillment period, an elected CH and a node which

is scheduled to transmit its reading remain in active state and other members go to sleep state again. Then, the node which is scheduled to transmit sends its reading to the elected CH. At the end of a frame, each CH role node sends the collected data to the last CH role node in the frame. The reason why the last CH role node gathers data from other CH role nodes is to avoid the pre-compromise of the data gatherer. That is, an attacker cannot predict which member is going to be the last CH role and therefore cannot aim at such a node as a compromise target. The last CH role node in the frame aggregates the received data and sends the aggregated data to the sink.

V. ANALYSES

We compare the proposed scheme with Holczer's scheme described in Section II. Under the circumstance where some compromised nodes exist, we seek the time during which compromised nodes play as a CH and the number of messages that the compromised CHs acquire from their members. Table I shows the variables used in our analyses.

TABLE I: VARIABLES USED IN OUR ANALYSES

Variable	Meaning	Value
N	The number of nodes	100
N_c	The number of compromised nodes	10~50
c	The number of clusters in the network	10
n_{CH}	The number of members in a cluster	10
$n_{CH}^{'}$	The number of CH candidates in a cluster($n_{CH}^{'} \le n_{CH}$)	10
T_{CH}	CH election period(i.e. round)	30 sec.
T_i	Time when then *i*-th compromised node is generated in a round	
R_i	Round when the i-th compromised node is generated	
P_{CH}^{ic}	Probability that a compromised node is elected as a CH	
E_{CH}^{c}	Expected number of compromised nodes in the network	
T_i^{one}	Time during which the *i*-th compromised node acts as a CH in a round	
T_i^{multi}	Time during which the *i*-th compromised node acts as a CH over multiple rounds	
$T_{all}^{compromise}$	Time during which all compromised nodes act as a CH	
D_{comp}	Total number of messages that compromised CHs collect	

In the Holczer's scheme, the probability that a compromised node is elected as a CH (P_{CH}^{ic}) is shown in (5).Then, the expected number of compromised nodes in the network (E_{CH}^{c}) is computed as shown in (6). Now, if any two compromised nodes are generated in a round, the time during which the *i*-th compromised node acts as a CH (T_i^{one}) can be computed using (7). Contrarily, if any two compromised nodes are generated over multiple rounds, the time during which the *i*-th compromised node acts as a CH (T_i^{multi}) can be computed using (8). Therefore, the time during which all compromised nodes act as a CH ($T_{all}^{compromise}$) is computed by (9).

$$P_{CH}^{ic} = \frac{N_c}{N} \times \frac{1}{n_{CH}} \tag{5}$$

$$E_{CH}^{c} = P_{CH}^{ic} \times n_{CH} \times c = \frac{cN_c}{N} \tag{6}$$

$$T_i^{one} = \frac{ci}{N} \times (T_{i+1} - T_i) \tag{7}$$

$$\begin{aligned} T_i^{multi} &= \frac{ci(T_{CH} - T_i)}{N} + \frac{ciT_{CH}}{N}(R_{i+1} - R_i - 1) + \frac{ciT_{i+1}}{N} \\ &= \frac{ciT_{CH}}{N}(R_{i+1} - R_i + \frac{T_{i+1} - T_i}{T_{CH}}) \end{aligned} \tag{8}$$

$$T_{all}^{compromise} = \sum_{i=1}^{N_c - 1} T_i^{one} \text{ or } T_i^{multi} \tag{9}$$

In our scheme, the probability that a compromised node is elected as a CH is shown in (10). Next, the expected number of compromised nodes in the network is calculated by (11). If any two compromised nodes are generated in a round, the time during which the *i*-th compromised node acts as a CH can be obtained by (12). On the other hand, if any two compromised nodes are generated over multiple rounds, the time during which the *i*-th compromised node acts as a CH can be obtained by (13). Consequently, the time during which all compromised nodes act as a CH is obtained by (14).

$$P_{CH}^{ic} = \frac{N_c}{N} \times \frac{1}{n_{CH}^{'}} \times \frac{1}{F_{CH}^{i} + 1} \tag{10}$$

$$E_{CH}^{c} = \sum_{i=1}^{n_{CH}^{'}} \frac{N_c}{N} \times \frac{1}{n_{CH}^{'}} \times \frac{1}{F_{CH}^{i} + 1} \tag{11}$$

$$T_i^{one} = \sum_{k=1}^{n_{CH}^{'}} \left(\frac{ci}{N} \times \frac{1}{n_{CH}^{'}} \times \frac{1}{F_{CH}^{k} + 1} \right) \times (T_{i+1} - T_i) \tag{12}$$

$$T_i^{multi} = \sum_{k=1}^{n_{CH}^{'}} \left(\frac{ci}{N} \times \frac{1}{n_{CH}^{'}} \times \frac{1}{F_{CH}^{k} + 1} \right) \times (T_{CH} - T_i) \tag{13}$$

$$T_{all}^{compromise} = \sum_{i=1}^{N_c - 1} T_i^{one} \text{ or } T_i^{multi} \tag{14}$$

Now, we estimate the number of messages that compromised CHs acquire from their members. First, we assume that each member senses d messages and delivers them to the CH in a round. If there is no message loss, the total number of collected messages in a cluster is $d \times n_{CH}$. Therefore, the total number of messages that compromised CHs acquire is obtained by multiplying $T_{all}^{compromise}$ by the total number of messages collected per round as shown in (15).

$$D_{comp} = T_{all}^{compromise} \times \frac{d \times n_{CH}}{\frac{T_{CH} - T_i}{T_{CH}} + (R_{i+1} - R_i) T_{CH} + \frac{T_{i+1}}{T_{CH}}} \tag{15}$$

Using the above equations, we evaluate the security of our scheme and Hloczer's scheme. We assume that the network operation time is 1800 seconds and compromise period is computed by dividing the network operation time by the number of compromised nodes. For example, if 10 nodes are compromised during the network operation time (i.e. 1800

seconds), a new compromised node is generated every 180 seconds. First, we show how the increase of compromised nodes affects their CH role duration time in Fig. 3. As shown in Fig. 3, our scheme dramatically reduces the CH role duration time of compromised nodes. This is because our scheme manages all members' trust value and excludes some untrustworthy nodes from CH candidates periodically. Contrarily, Holczer's scheme never evaluates the trust level of CH candidates and never reselects the trustworthy nodes. Therefore, if a node is compromised, it can keep declaring itself as a CH regardless of its qualification since then. That is why the performance difference between two schemes becomes larger as the number of compromised nodes increases.

Now, we show how the increase of compromised nodes affects the number of messages that compromised CHs collect. For the sake of simplicity, we assume that each member gets a sensor reading every second. So, each member sends 30 messages to its CH during a round. As shown in Fig. 4, our scheme significantly decreases the number of messages that compromised CHs gather from their members. This is greatly credited to a combination of our scheme's trust evaluation and exclusion of untrustworthy nodes. That is, because our scheme periodically evaluates the trust value of members and prevents some untrustworthy nodes from remaining as CH candidates, compromised nodes are likely to lose their chances to declare themselves as a CH. On the contrary, Holczer's scheme has no mechanism to evaluate the trust level of members in a cluster and reselects the CH candidates. So, if a node is compromised, it can keep declaring itself as a CH every CH election round. It makes the volume of messages that compromised CHs gather from their members greatly swollen as the number of compromised nodes increases as shown in Fig. 4.

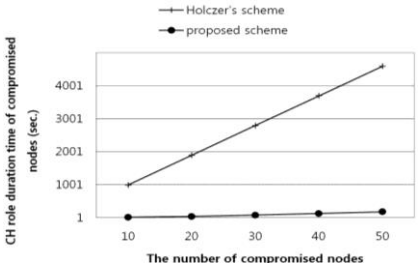

Fig. 3. CH role duration time of compromised nodes.

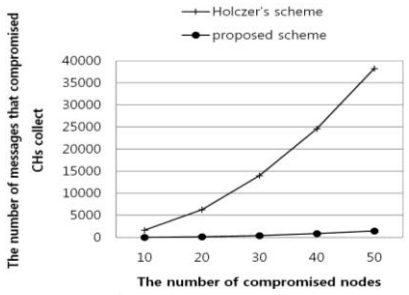

Fig. 4. The number of messages that compromised CHs collect.

VI. Conclusion

In this paper, we proposed a secure election scheme that minimizes the effect of generation of compromised CHs. Our scheme periodically evaluates the trust level of members in a cluster considering their CH role fulfillment frequency. Then, our scheme picks out some untrustworthy nodes and expels them from the CH candidates. Our analysis shows that our scheme outperforms Holczer's scheme in terms of CH role duration time of compromised nodes. Besides, another analysis shows that our scheme allows compromised CHs to collect much smaller messages from members than Holczer's scheme.

References

[1] G. Wang *et al*., "A secure cluster sormation scheme in sireless sensor setworks," *Int. J. of Distributed Sensor Networks*, pp. 1-14. Oct. 2012.
[2] M. Sirivianos *et al*., "Non-manipulable sggregator sode slection srotocols for sireless sensor setworks," in *Proc. Int Sympo. on MOMAWN*, 2007, pp. 1-10.
[3] Q. Dong and D. Liu, "Resilient sluster seader slection for sireless sensor setworks," in *Proc. 6th Annual Comm. Society Conf. on Sensor, Mesh and Ad Hoc Communications and Networks*, 2009, pp. 108-116.
[4] L. Buttyan and T. Holczer, "Private sluster sead slection in sireless sensor setworks," in *Proc. the Fifth IEEE Int'l Workshop on Wireless and Sensor Network Security*, 2009, pp. 1048-1053.
[5] T. Holczer and L. Buttyan, "Anonymous aggregator election and data aggregation in wireless sensor networks," *International Journal of Distributed Sensor Networks*, pp. 1-19, 2011.
[6] P. Schaffer *et al.*, "Secure and seliableslustering in sireless sensor setworks: A sritical survey," *Computer Networks*, vol. 56, no. 11, pp. 2726-2741, Jul. 2012.
[7] K. Sun *et al*., "Secure sistributed sluster sormation in sireless sensor setworks," in *Proc. 22nd Ann. Comp. Secu. Appli. Conf.*, pp. 131-140.
[8] H. Rif à-Pous and J. Herrera-Joancomart í, "A sair and secure sluster sormation srocess for Ad Hoc setworks," *Wireless Personal Communications*, vol. 56, no. 3, pp. 625-636, Feb. 2011.

Gicheol Wang received the B.S. degree from Gwangju University, Gwangju, Korea, in 1997, and the M.S. degree from Mokpo National University, Mokpo, Korea, in 2000, in computer science and statistics. He received Ph. D degree in computer science and statistics from Chonbuk National University, Jeonju, Korea, in 2005.

He worked for CAIIT (Center for Advanced Image and Information Technology) at Chonbuk Nat'l University, Jeonju, Korea, as a postdoctoral research fellow from Jan. 2006 to Dec. 2007. He worked for the Research Center for Ubiquitous Information Appliances at Chonnam Nat'l University, Gwangju, Korea, as a postdoctoral research fellow from Jan. 2008 to Dec. 2008. From Jan. 2009 to Nov. 2013, he worked for the Advanced KREONET Center at KISTI (Korea Institute of Science and Technology Information), Daejeon, Korea. From Dec. 2013, he joined the ADD (Agency for Defense Development) and he is currently serving as a senior research scientist. His current research interests include ad hoc networks, sensor networks, vehicular networks, security of wireless networks, and mobile computing.

Dr. Wang is a member of IEEE, KICS, and IEEK.

Gihwan Cho received the B.S. degree from Chonnam University, Gwangju, Korea, in 1985, and the M.S. degree from Seoul National University, Seoul, Korea, in 1987, both in computer science and statistics. He received Ph.D degree in computer science from University of Newcastle, Newcastle Upon Tyne, England, in 1996.

He worked for ETRI (Electronics and Telecommunications Research Institute), Daejeon, Korea, as a senior member of technical staff from Sep. 1987 to Aug. 1997, for the Dept. of computer science at Mokpo National University, Mokpo, Korea, as a full time lecture from Sep. 1997 to Feb. 1999. From Mar. 1999, he joined the Div. of computer science and engineering at Chonbuk National University, Chonju, Korea, and he is currently serving as a professor. His current research interests include mobile computing, computer communication, security of wireless networks, sensor networks, and distributed computing system.

Prof. Cho is a member of IEEE, KIISE, KIPS, KMMS, KSII.