Security Impacts of Rotating a Cluster Headamong Trustworthy Nodes on Wireless Sensor Networks

GicheolWang and Eulho Chung
Agency for Defense Development, UAV Systems PMO, Daejeon, Republic of Korea
Email: {gcwang, tigerchung}@add.re.kr

***Abstract*—In a clustered sensor network, a CH (Cluster Head) plays a role of local data collector and deliverer of the collected data to the sink. That is the reason why we should protect a CH election process from internal and external attackers. Even though many schemes for the protection of CH election have been proposed, there has been little attention to compromise of CHs during the data forward phase. However, if we rotate a CH role among members during the data forward phase, we can reduce the data volume exposed to attackers even though a CH is compromised. In this paper, we propose a scheme which rotates a CH role among trustworthy members as well as protects a CH election. Our security analyses show that our scheme reduces CH role duration time of compromised CHs and consequently the data volume that the compromised CHs can obtain. Besides, our performance analysis shows that our scheme outperforms a rival scheme in terms of communication and computation overheads.**

***Index Terms*—secure cluster head election, rotation-based cluster head election, secure cluster formation,wirelesssensor network**

I. Introduction

A prominent advantage of clustering in wireless sensor network is saving energy consumption of nodes and consequently extending the lifetime of network. In many cases, a cluster holds a local aggregator which is called a CH (Cluster Head) which collects data from its members and delivers the collected data to the sink. Due to the crucial functions of a CH, attackers will try to become a CH or compromise a CH [1]. Therefore, assigning a CH role to a legal and eligible node has attracted a lot of attention from researchers and some schemes [2]-[5] are part of those research efforts.

Holczer et al. proposed a distinctive scheme [5] which deviates from such a research trend. In Holczer's scheme, a CH election is hidden not only from external attackers but also from eligible members. However, the scheme cannot prevent a compromised node from declaring itself as a CH since it never tries to find suspected nodes among normal nodes and to expel them from the network. Even worse, if a compromised node is elected as a CH at the beginning of a round, it can keep collecting data from members during the rest time of the round.

This paper is an initial effort of resolving the problem. First, our scheme estimates the trust level of nodes and expels some disreputable nodes at the beginning of a round. Next, our scheme periodically rotates the CH role among the trustable nodes to reduce the size of data volume disclosedto a CH during the remaining time of the round.

Our paper is organized as the following. Some CH election schemes are reviewed in Section 2. Section 3 first provides the backgrounds of our scheme and then the details of our scheme are unfolded in Section 4. Section 5 analyzes the security and performance aspects of our scheme and a rival scheme and conclusion is drawn in Section 6.

II. RelatedWork

Random number based election was first introduced in [2]. The first step of the election is to make and share the common random number among members and the second step is to elect a CH using the common random number. That is, we can get a remainder dividing the common random number by the number of members and the remainder indicates the position of the CH in the member list. The authors of [2] divide the election into three sub-methods following how to generate and distribute the common random number; Merkle's puzzle based scheme, commitment based scheme, and seed based scheme.

In the first scheme, the Merkle's puzzle is employed for distributing pairwise keys between the current CH and its members. Then, each member generates a random number and encrypts it with its pairwise key and adds the encrypted random number into the received sum if there is a received sum. Then it delivers the updated sum to one of other members. These actions are repeated at every member and the last member plays a role of broadcaster of the total sum. Next, the current CH broadcasts all pairwise keys to members to help them convert the total sum into the plain sum using the double additive homomorphic encryption [2]. The plain sum plays a role of the common random number.

Commitment scheme first forces each member pair to establish a pairwise key. Then, whenever a new CH is required, each member makes a commitment for its

Manuscript received June 25, 2014; revised September 19, 2014.

doi: 10.12720/lnit.2.3.208-213

random number which is the random number's ciphertext and delivers it to other members in the P2P way. Then, it delivers the original random number to other members to help them verify the corresponding commitment and add the random number into the common random number.

The seed based scheme first makes each member generate its seed which is an initial random number and distribute the seed through a broadcast. Whenever a new CH is needed, each member generates an availability message representing its election participation and broadcasts it. As soon as members get an availability message, they produce the sender's new random number using its seed and the election round number. Finally, members add those new random numbers into the common random number to decide a new CH.

In the Merkle's puzzle based scheme, three main steps (that is, the pairwise key establishment, creation of the common random number, and the pairwise key distribution) are burdensome to sensor nodes in terms of communication and computation overheads. Even though two other schemes are more lightweight than the Merkle's puzzle based scheme, they are vulnerable to transmission avoidance and selective message transmission. They are harmful to a CH election protocol because the transmission avoidance arbitrarily changes a CH election result and the selective message transmission splits one election result into multiple ones.

Dong's scheme [3] focuses on preventing external attackers from taking part in a CH election. Its strength highly depends on the ID assignment scheme that ties a node's ID, its 'Yes' and 'No' key chains and its polynomial shares. Whenever a new CH election is required, each member broadcasts a 'Yes' key to participate the CH election or a 'No' key to give up the CH election participation. Among the active members, a real CH is selected in the round-robin manner. Due to this predictable feature, a compromised node can avoid transmitting a message to change the CH election result or selectively transmit a message to split the CH election result.

Buttyan's scheme [4] was the first scheme that veiled a CH election process from external attackers. Since the veiling mechanism highly depends on an encryption and the corresponding decryption, the scheme is robust against external observers while it is still vulnerable to an internal observer's eavesdropping. Furthermore, the scheme cannot prevent the internal observer from declaring itself as a CH illegally.

Contrarily, Holczer's scheme [5] hides the CH election process from not only external observers but also internal observers. In the first step, each member decides to become a CH depending on the CH winning probability. In the second step, each member investigates whether a member which decided to become a CH exists or not. Since the investigation discloses not which node decided to take over the CH role but there exists such a node, it is robust against internal observers as well as external observers. Although the authors in [6] commented that the Holczer's scheme is one of the most secure schemes, it is also vulnerable to an illegal CH declaration of a compromised node.

III. Network and Threat Model

A. Network Model

We assume that sensors are scattered into a mission field by an aircraft and they securely form clusters after the deployment using some promising schemes [1], [7], [8]. During the cluster formation, each member pair also establishes a pairwise key with each other. Then, our scheme removes some untrustworthy nodes from CH candidates. Next, our scheme elects a CH node among the survivors depending on a predefined probability and the elected CH advertises its role. Then, the elected CH aggregates the data from members and the CH forwards the aggregated data to the sink. These two steps of the CH election and the data aggregation and forward are repeated until the end of the round to mitigate the threat of a compromised CH. When a round ends, those three steps after the cluster formation are starts again. Fig. 1 shows the flowchart of our scheme's network operation.

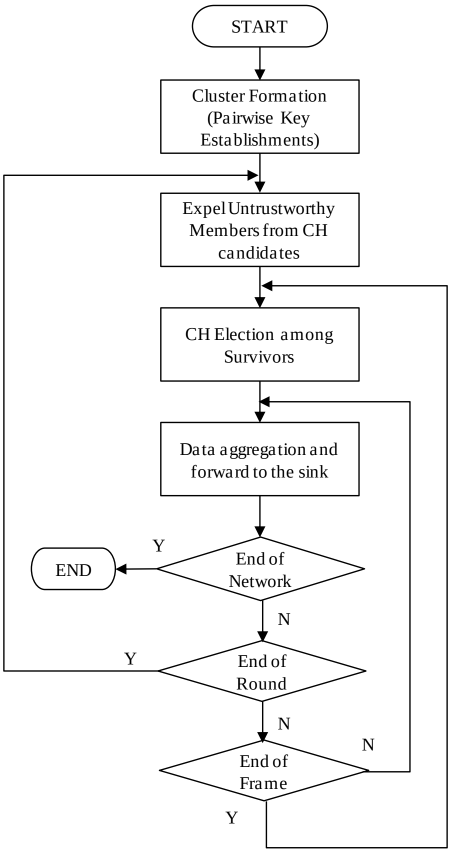
START
Cluster Formation (Pairwise Key Establishments)
Expel Untrustworthy Members from CH candidates
CH Election among Survivors
Data aggregation and forward to the sink
End of Network
Y
END
N
End of Round
Y
N
End of Frame
N
Y

Figure 1. Network operation of the proposed scheme.

B. Threat Model

A compromised node is assumed to advertise a CH role regardless of its eligibility during every CH election period. This malicious action's impact on the network is very clear. The compromised node can keep maintaining a CH role until it is identified as a malicious node and screened out of the network. So, it can obtain a large amount of traffic from members illegally. Especially, as long as the compromised node behaves like a normal node, other nodes can hardly identify such a node and exclude it from the network. To our best knowledge, the only way to prevent this malicious action is to rotate a CH role among behaving members in a round-robin manner.

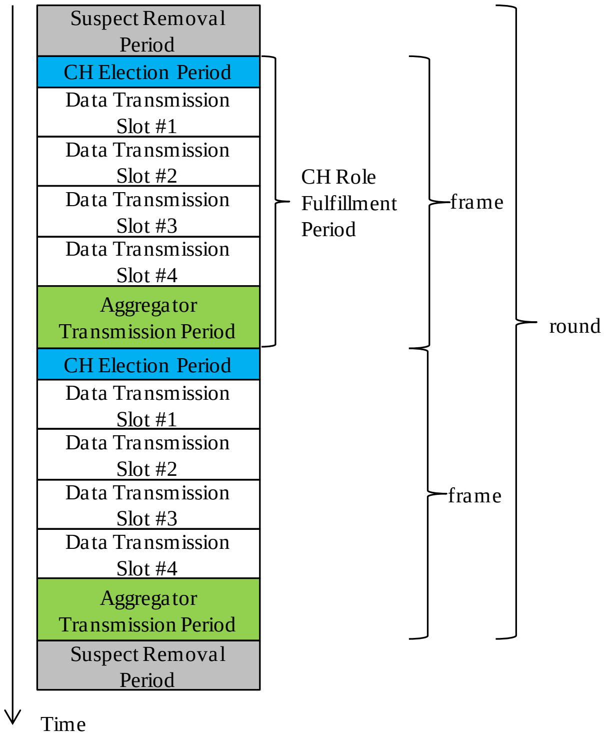
Suspect Removal Period
CH Election Period
Data Transmission Slot #1
Data Transmission Slot #2
Data Transmission Slot #3
Data Transmission Slot #4
Aggregator Transmission Period
CH Election Period
Data Transmission Slot #1
Data Transmission Slot #2
Data Transmission Slot #3
Data Transmission Slot #4
Aggregator Transmission Period
Suspect Removal Period
CH Role Fulfillment Period
frame
frame
round
Time

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

IV. Rotation of Cluster Head amongTrustworthy Nodes

External adversaries are assumed to be removed from the CH candidatesover the course of the cluster formation.Recall that some promising cluster formation protocols [1], [7], [8]can be employed to attain the purpose.Note that each member pair in a cluster has established a pairwise key before entrance of the CH election process.Adversaries referred in the rest of this paper mean the internal nodes which take part in the CH election process.Two steps constitute a round in our scheme. The first step screens out some suspected nodes from CH candidates. To screen out some misbehaving nodes from CH candidates, each member evaluate the trust level of other members according to their behaviorand generates the corresponding trust values. We look into the trust value in the previous round and the frequency of CH declaration in the current round to generate the current trust value. Initial trust value is always one. After getting all trust values of the current round, each member obtains the average and removes some nodes whose trust value is lower than the average from the CH candidates. Details of the first step are depicted in SectionIV.A.A round's second step is split into multiple frames and a frame is again split into multiple data transmission slots and the aggregator transmission period. Note that the number of data transmission slots in a frame is same as the number of members in the cluster. Once a CH is elected in the CH election period,it fulfills its role during the rest of the frame to receive the data from its members. Therefore, the time length of a CH role fulfillment period is almost same as that of a frame.Over the course of cluster formation,each member recognizes its transmission schedule delivered from the sink and it wakes up in its transmission slot and sleeps during other slots. Fig. 2 shows how our scheme is operated with the lapse of time. Note that all members should wake up every CH election period to elect a new CH during the current frame. The election depends on the CH winning probability which is explained in detail in the following subsections.If a node is elected as a CH, it broadcasts a CH declaration message to preventa duplicate CH declaration of other members. At the aggregator transmission period of a frame, the corresponding CH aggregates the received all data and delivers the aggregated data to the sink. Note that a member employs its pairwise key to securely transmit its data to the CH and the pairwise key was previously established between nodes. We explain the second step in Section IV.B in detail.

A. Removal of Suspects

At each start of round, each node estimates trust level of other members in the same cluster. Let R_k^i be node *i*'s trust value in a round *k*while all R_1^i is one. Each node can compute node *i*'s trust value in the previous round through expected CH frequency (F_e^i) and real CH frequency (F_r^i) of the node *i*. (1) shows how the expected CH frequency is generated where P_{win}^i, I_{frame}, and I_{round}mean CH winning probability, time interval of a frame, and the time interval of a round each. Also, to give all nodes a fair opportunity of becoming a CH, we settle the CH winning probability of each node as (2) where n_{cand} , T_c , T_{CH}^i mean the number of members, current time, and node *i*'s CH declaration time respectively.

$$F_e^i = \left\lfloor P_{win}^i \times \frac{I_{round}}{I_{frame}} + 0.5 \right\rfloor \qquad (1)$$

$$P_{win}^i = \left(\frac{1}{n_{cand}} \times \frac{1}{F_r^i + 1} \times \frac{T_c - T_{CH}^i}{T_c} \right) \qquad (2)$$

Now each node's trust value for the previous round (R_{k-1}^i) is generated using F_e^i and F_r^i as shown in (3).

Then, each node's current trust value can be computed using the previous trust value (R_{k-1}^i) and the total frequency of CH role fulfillment (F_{CH}^i) as shown in (4). Namely, a current trust value becomes also high if its previous value is high while it decreases as the total frequency of CH role fulfillment rises up. Note that the F_r^i returns to zero whenever each new round starts.

$$R_{k-1}^{i} = \frac{1}{\max\left[F_r^i - F_e^i, 0\right] + 1} \tag{3}$$

$$R_k^i = \frac{R_{k-1}^i}{F_{CH}^i + 1} \tag{4}$$

In a cluster, the average of all current trust values is computed and employed to screen out some disreputablemembers from the CH candidates. In other words, some members whose current trust value is lower than the average are expelled from the CH candidates. Survived members set the expelled members' current trust value to their previous value.

B. CH Elections and Data Transmissions

1) CH elections

After the removal of suspected nodes, each member elects itself as a CH or becomes a member of a CH declaration node according to the CH winning probability (P_{win}^i). The role determination depends on the 'first-come-first-served' principle. That is, when the CH winning probability of all members is equal, the first declaration node becomes a CH and other members become the members of the CH. The CH role is valid during the current frame and after that a new election is performed to give a CH role to a different member during the next frame. This approach's advantage is quite straightforward. Even if an attacker succeeds in compromising a member, its CH role duration time is too short to gain a sufficient amount of information. Furthermore, since the CH winning probability decreases by $1/(F_{CH}^i + 1)$ wheneverthe node fulfills a CH role and the opportunity is equalized by $(T_c - T_{CH}^i)/T_c$, it can hardly become a CH again during the rest of the round.If no CH is elected in a CH election period, each member doubles the CH winning probability and reelects a CH.

2) Data transmissions

In a CH fulfillment period, the elected CH gathers data from its members and each member wakes up in its assigned time slot to transmit its reading and sleeps during other nodes' time slot. At the end of a CH fulfillment period, the elected CH transmits the aggregated data to the sink. Then, the next CH role fulfillment period starts again.

V. Analyses

This section provides the security and overhead comparisons of our scheme and a rival scheme(that is, Holczer's scheme). We identify the CH role duration time of compromised nodes and the message volume that compromised nodes get from members as two security metrics. Then, we analyze the communication and computation overheads of two schemes. We list the variables and their values used for these analyses in Table I.

First, we focus on thesecurity aspects of Holczer's scheme.The CH winning probability of a compromised node(P_{win}^c) is computed by (5). Next, we can get theexpected number of compromised nodes (E_{win}^c)using (6).Because Holczer's scheme has no mechanism to expel a compromised node from the network during its operation, a compromised node can keep doing its malicious action.So, we can get the CH role duration time of a compromised node (D_{CH}^i)using (7).Last, we can get the sum of CH role duration time of all compromised nodes (D_{CH}^{total}) using (8).

$$P_{win}^c = \frac{n_c}{n} \times \frac{1}{n_{cand}} \tag{5}$$

$$E_{win}^c = P_{win}^c \times n_{cand} \times c = \frac{cn_c}{n} \tag{6}$$

$$D_{CH}^i = \frac{i}{n} \times \left(t_{i+1} - t_i\right) \tag{7}$$

$$D_{CH}^{total} = \sum_{i=1}^{n_c} D_{CH}^i \tag{8}$$

Concerning our scheme, the CH winning probability of a compromised node is computed by (9). Since there are $n_{cand}^{'}$ nodes in a cluster and *c* clusters in the network, we can get the expected number of compromised nodes using (10).Because our scheme expel compromised nodes from network during every suspect removal period, CH role duration time of a compromised node can be reduced to theremaining time until the next removal period as shown in (11).Therefore, the sum of CH role duration time of all compromised nodes can be obtained by (12).

$$P_{win}^c = \left(\frac{n_c}{n} \times \frac{1}{n_{cand}^{'}} \times \frac{1}{F_r^i + 1}\right)\left(\frac{T_c - T_{CH}^i}{T_c}\right) \tag{9}$$

$$E_{win}^c = c\sum_{i=1}^{n_{cand}^{'}}\left(\frac{n_c}{n} \times \frac{1}{n_{cand}^{'}} \times \frac{1}{F_r^i + 1}\right)\left(\frac{T_c - T_{CH}^i}{T_c}\right) \tag{10}$$

$$D_{CH}^i = \sum_{k=1}^{n_c}\left(\frac{k}{n} \times \frac{1}{n_{cand}^{'}} \times \frac{1}{F_r^k + 1}\right)\left(\frac{T_c - T_{CH}^k}{T_c}\right)\left(I_{round} - t_i \bmod I_{round}\right) \tag{11}$$

$$D_{CH}^{total} = \sum_{k=1}^{n_c} D_{CH}^i \tag{12}$$

Because we have the CH role duration time of all compromised nodes, we can estimate how many messages they can get during the network operation time. To this aim, each member is assumed to sense *d* messages in a round and delivers them to its CH. So, the number of messages collected by a CH is $d \times n_{cand}$. Therefore, we can get the number of whole messages that compromised CHs gather during a round using (13).

$$M_{comp} = D_{CH}^{total} \times \frac{d \times n_{cand}}{I_{round}} \tag{13}$$

Now, we illustrate the above analyses by substituting some variables with real values. Network operation time is set to 1800 seconds and a compromise period is set by dividing the network operation time by the number of compromised nodes. For instance, if we have 10 compromised nodes, we have a new compromised node every 180 second. The real values which substitute the variables are shown in Table I. Fig. 3 shows how the increase of compromised nodes affects the CH role duration time of compromised nodes. As depicted in Fig. 3, our scheme significantly reduces the CH role duration time of compromised nodes. This result is caused by two facts. First, since our scheme rotates a CH role during a round, the CH role duration time of a compromised node is very short and transient. Second, since our scheme periodically excludes untrustworthy nodes during the suspect removal period according to trust level of nodes, compromised node hardly survive the exclusion process. On the contrary, Holczer's scheme has no mechanism to exclude some disreputable nodes from the network. This makes a compromised node can keep doing a malicious action since its birth. That is the reason why the CH role duration time of compromised nodes grows up rapidly as the number of compromised nodes gradually increases

TABLE I. VARIABLES AND THEIR MEANING

Variable	Meaning	Value
n	Population of nodes	100
n_c	Population of compromised nodes	10~50
c	The number of clusters	10
n_{cand}	Population of members in a cluster	10
n'_{cand}	Population of CH candidates in a cluster($\leq n_{cand}$)	10
I_{round}	Interval of a CH election round	30 sec.
d	The number of messages each sensor reads during a round	30
t_i	Time when$i$-th compromised node happens	
P^c_{win}	CH winning probability of a compromised node	
E^c_{win}	Expected population of compromised nodes	
D^i_{CH}	CH role duration time of the i-th compromised node	
D^{total}_{CH}	CH role duration time of all compromised nodes	
M_{comp}	The number of messages which compromised CHs gather from members	

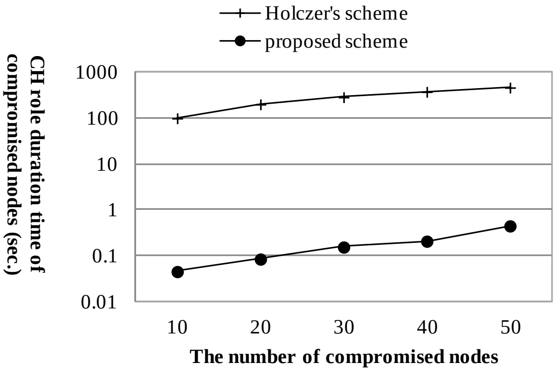

Figure 3. CH role duration time of compromised nodes.

Fig. 4 shows how the increase of compromised nodes affects the volume of messages that compromised CHs gather from their members.As depicted in Fig. 4, our scheme greatly reduces the message volume exposed to compromised CHs. Because our scheme significantly reduces the CH role duration time of compromised CHs, the data volume that the compromised CHs can gather is also reduced accordingly. Holczer's scheme has no mechanism to screen out a compromised CH even if it hides a CH election process from not only external attackers but also internal attackers.Therefore, a compromised node can keep gathering data from members since its first CH declaration. That makes the difference of data volume that compromised CHs gather is quite big as depicted in Fig. 4.

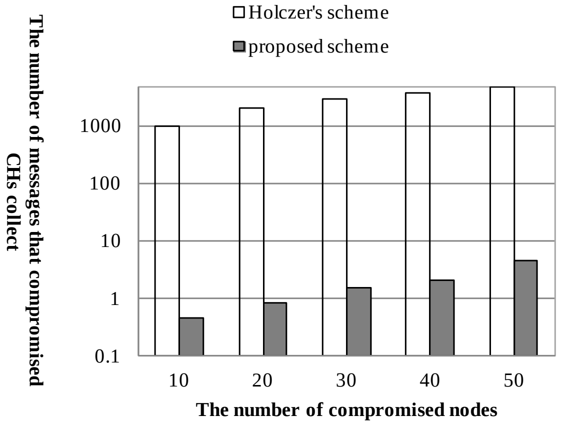

Figure 4. The number of messages that compromised CHs capture.

TABLE II. OVERHEAD COMPARISON

Scheme	Communication overhead	Computation overhead
Holczer's scheme	$O(n^2)$	$O(n)$ exponentiations
Our scheme	$O(n)$	$O(n)$ arithmetic operations

Next, we compare overheads of two schemes. First, we investigate the communication overhead of two schemes. Holczer's scheme makes each node transmit $r(n^2+n+1)$ messages during network operation time. Here, r is the number of rounds and n is the number of members in a cluster and consequently its communication overhead is $O(n^2)$ .On the other hand, our scheme makes each node deliver $r(n+2)$ messages during network operation and its communication overhead is $O(n)$. Next, we investigate computation overhead of two schemes. In Holczer's scheme, each member executes four exponentiations for two messages and knowledge proof and receivers executes $4n-4$ exponentiations to verify the knowledge proof sent from $n-1$ other members. Because, each node executes $4rn$ exponentiations during the network operation time, its computation overhead is $O(n)$ exponentiations. Contrarily, our scheme makes each member execute four arithmetic operations for every other member during a round. Therefore, each node executes $4r(n-1)$ arithmetic operations during the network operation time and the computation overhead is $O(n)$ arithmetic operations. Because the overhead of an exponentiation is much heavier than that of an arithmetic operation, our scheme's computation overhead is much lower than Holczer's scheme. Table II shows the overhead comparison of two schemes.

VI. Conclusion

In this paper, we presented a secure CH election scheme which greatly mitigates the bad impact of compromised CHs on the network. To this aim, our scheme employs two promising techniques.First,at the beginning of each round,our scheme measures the trust value of nodes and screens out some untrustworthy nodes from CH candidates. Second, during the remaining time of each round, our scheme periodically assigns a CH roleto one of the survivors in a round-robin manner.We evaluated the security of our scheme and a rival scheme through analyses. Our scheme exceeds over the rival scheme in terms of CH duration time of compromised CHs and traffic volume that the compromised CHs collect illegally. Furthermore, the following performance analysis showed that our scheme causes much less overhead than the rival scheme. As a future work item, we plan to do some simulations using a well-known simulator for proving the security and performance superiority of our scheme over other schemes.

References

[1] G. Wang, D. Kim, and G. Cho, "A secure cluster formation scheme in wireless sensor networks," *Int'l Journal of Distributed Sensor Networks*, vol. 2012, Oct. 2012.

[2] M. Sirivianos, D. Westhoff, F. Armknecht, and J. Girao, "Non-manipulable aggregator node electionprotocols for wireless sensor networks," in *Proc. Int'l Sympo. onModeling and Optimization in Mobile, Ad Hoc, and WirelessNetworks*, Cyprus, Apr. 2007, pp. 1-10.

[3] Q. Dong and D. Liu, "Resilient cluster leader election for wireless sensor networks," in *Proc. IEEE 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 cluster head election in wireless sensor networks," in *Proc. the Fifth IEEE Int'l Workshop on Wireless and Sensor Network Security*, IEEE, 2009, pp. 1048-1053.

[5] T. Holczer and L. Buttyan, "Anonymous aggregator election and data aggregation in wireless sensor networks," *Int'l Journal of Distributed Sensor Networks*, vol. 2011, pp. 1-19, Jun. 2011.

[6] P. Schaffer, K. Farkas, A. Horvath, T. Holczer, and L. Buttyan, "Secure and reliable clustering in wireless sensor networks: A critical survey," *Computer Networks*, vol. 56, no. 11, pp. 2726-2741, Jul. 2012.

[7] K. Sun, P. Peng, P. Ning, and C. Wang,"Secure distributed cluster formation in wireless sensor networks,"in *Proc. 22nd Annual Computer Security Applications Conference*, 2006, pp. 131-140.

[8] H. Rifà-Pous and J. Herrera-Joancomartí, "A fair and secure cluster formation process for Ad Hoc networks,"*Wireless Personal Communications*, vol. 56, no. 3, pp. 625-636, Feb. 2011.

Gicheol Wangreceived 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, 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 to the ADD(Agency for Defense Development) and he is currently serving as a senior research scientist. His current researchinterests include ad hoc networks, sensor networks, security of wireless networks, and global smart roaming.Dr. Wang is a member of IEEE, KICS, and IEEK.

Eulho Chungreceived the B.S. degree from Kwangwoon University, Seoul, Korea, in 1989, and the M.S. degree from Kwangwoon University, Seoul, Korea, in 1991, in electronicengineering.He worked for ADD(Agency for Defense Development),Daejeon, Korea, as a researcher from Feb. 1991. He is currently serving as a principal researcher. His current researchinterests include data link system for UAS, security of wireless networks and datalink controller.