WSN LIFETIME EXTENSION USING GA OPTIMISED FUZZY LOGIC

Sang-Hyeok Lim[1] and Tae-Ho Cho[2]

[1]College of Information and Communication Engineering, Sungkyunkwan University, Suwon 16419, Republic of Korea

[2]College of Software Platform, Sungkyunkwan University, Suwon 16419, Republic of Korea

ABSTRACT

A wireless sensor network (WSN) consists of multiple sensor nodes and base stations that collect information from widely deployed sensors. However, the disadvantage is that WSNs are randomly distributed in an open environment, which makes them difficult to manage individually and more easily found and compromised by an attacker. An attacker can execute a false report insertion or invalid vote insertion attack through a compromised node. The Probabilistic Voting Filtering System (PVFS) is a system that prevents these two types of attacks. Before sending a report, the proposed method probabilistically selects a validation node, determines the validity of the report, and filters the report based on the thresholds that have been set. In this paper, the proposed scheme improves the lifetime, detection rate, and report delivery rate of the entire network by increasing the lifetime of the cluster head (CH) by selecting the numbers of message authentication codes (MACs) and verification nodes of the report. Using this system, the event detection rate and the network lifetime are improved by up to 18% and 6%, respectively, relative to the existing PVFS.

KEYWORDS

Wireless sensor networks, False report injection attack, False vote injection attack, Secure routing, Fuzzy system, Genetic algorithm, Interactive authentication

1. INTRODUCTION

Wireless sensor networks (WSNs) are composed of many sensor nodes and a base station (BS). It is installed to detect events that occur in the field. When an event occurs, a sensor node detects the event and reports it to the BS via multiple hops between the sensor nodes, thus creating a report on the event [1]. These WSNs are used for data collection and event detection in various applications, including home networks, military systems, and forest fire monitoring [2]. However, sensor nodes are vulnerable to attack due to disadvantages such as limited computational ability, limited energy, random distribution in an open environment that operates independently, and individual management difficulties [3, 4]. Attackers exploit these vulnerabilities to attack WSNs by injecting reports containing false information or false votes.

DOI:10.5121/ijcsit.2017.9501

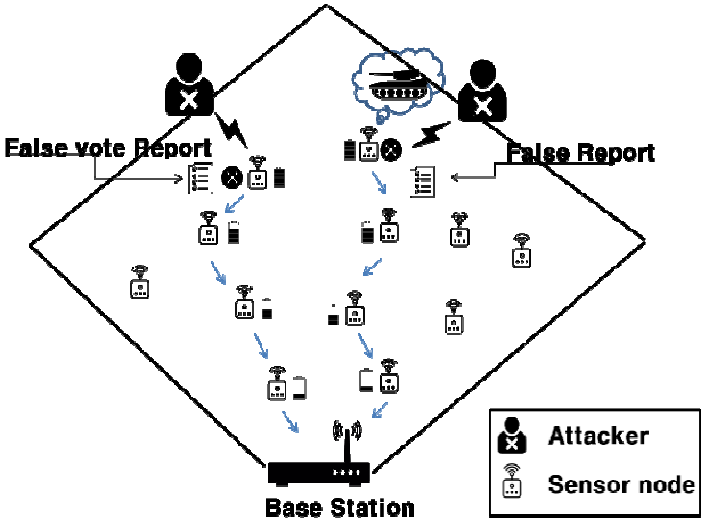

Figure 1: False report and false vote injection attacks.

Figure 1 shows a schematic of these attacks. A false report injection attack is one that injects a report about a non-existent event through the compromised sensor node. The goal of this attack is to exhaust the energy resources of the nodes on the propagation path and generate a false alarm at the BS. The false vote injection attack injects false votes into legitimate reports, thereby preventing the legitimate report from reaching the BS. Li and Wu proposed a probabilistic voting-based filtering scheme (PVFS) [5] to prevent such attacks. In the PVFS, all nodes constitute a network that exploits cluster-based organization. When a cluster head (CH) recognizes an event, it generates a report for that event. It then sends the generated report to the member nodes. Then, the member nodes determine the authenticity of the report and generate their own message authentication codes (MACs), alternatively referred to as votes in the PVFS. The CH randomly selects votes and inserts them into the report. Verification nodes on the path use MAC and threshold values to defend against attacks. An attacker can attempt a false report injection attack and false vote injection attack through a compromised member node.

2. RELATED WORKS

2.1. Fuzzy logic

One of the advantages of fuzzy rule systems is that they can be used for pseudo-reasoning, which is very useful if there is uncertainty in the inference process or if the data are ambiguous [6 ~ 9]. There are uncertainties in attacks that occur at the application layer of the actual WSN because of their type and frequency. It is also difficult to determine accurate attack figures when using a limited range of thresholds. Therefore, similar inferences are needed to deal with this fuzzy information. The inference of the fuzzy logic method uses the min-max synthesis approach of the Mamdani model, and the reverse fuzzy method for the output uses the center of gravity method. The Mamdani-type fuzzy inference process consists of four steps. The first stage is the fuzzing of input variables. Here, we determine how many input values belong to each of the appropriate fuzzy sets. In the second stage, rule evaluation, we take a fuzzy input and obtain a number representing the evaluation result of the previous case. We apply this number to the membership function of the latter case. Step 3 integrates the rules as an output and combines the membership functions of all rules after the rule set in the previous step into one fuzzy set. Finally, in Step 4, we apply reverse fuzzy logic. For the output value to be a number, the input in the deserialization

process must be a combined output fuzzy set in which the output is represented by a single number. At this stage, the center of gravity method is used.

2.2. PVFS

To cope with false report injection and false vote injection attacks in WSNs, the proposed PVFS uses a true threshold value (Tt) and a false threshold value (Tf) to detect and filter false reports and false vote injection reports at certain validation nodes. Figure 2 shows the report generation and verification node selection process. In the initial network configuration, the nodes are divided into cluster units, and the CHs responsible for report generation are selected for each cluster.

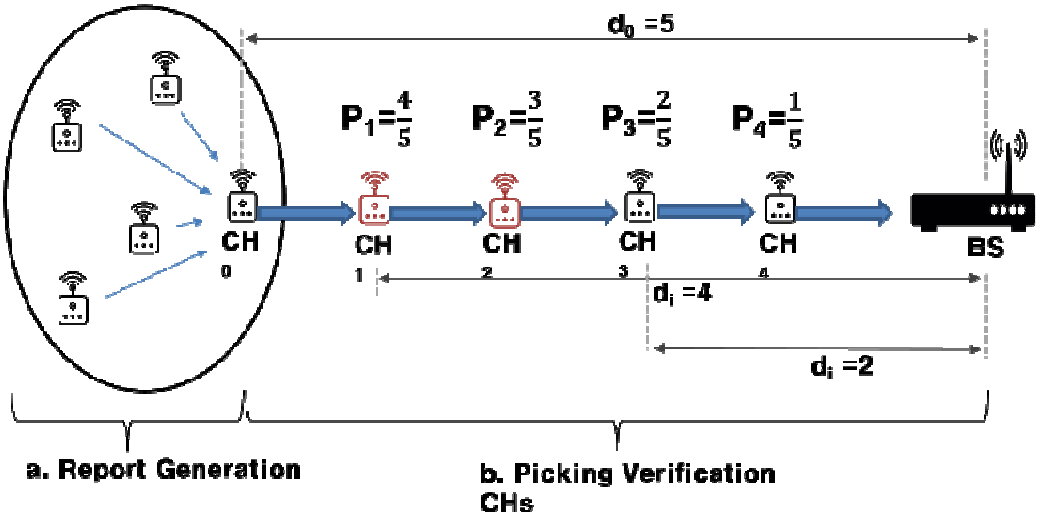

Figure 2: Report generation and verification node selection processes.

To verify the report, verification nodes are probabilistically selected from the CHs. The probability p depends on the distance d_0 between the BS and the event cluster and the distance d_i between the BS and CH_i. The verification node selection process is shown in Figure 2-b. Figure 3 shows the key allocation step, in which the BS divides the key pool into N partitions and delivers them to each CH. Each partition contains L keys that are the size of the cluster. The CH uses one of the keys in the partition as its own key and distributes the remaining L-1 keys to the member nodes. A key is allocated to each of the member nodes according to the partition of the key pool.

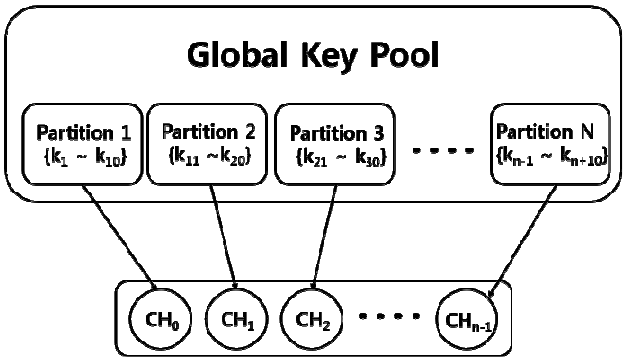

Figure 3: Key distribution process.

The node selected as the verification node stores the keys of the member nodes of the event occurrence cluster one by one. In the report generation step, the CH generates a report on an event and broadcasts it to the member nodes. Each member node confirms this, and if the report is determined to be a normal report, the MAC created by its own key is transmitted to the CH. CH extracts a predetermined number of the MACs received from the member nodes and adds them to the report. In the report verification process, the verifying nodes compare their own keys with the keys in the report. If the keys match, they verify the MAC of the report. If the MAC values generated by the same key are different, the vote is regarded as false, and Tf is increased. In the filtering process, if the false vote count reaches the threshold value, the report is determined to be false and is immediately dropped. If the true vote count reaches the threshold value, the report is considered to be legitimate and sent to the BS without further validation.

2.3. Genetic Algorithm

Genetic algorithm (GA) was proposed by Holland John and takes ideas from evolutionary processes occurring in the natural world. This kind of algorithm represents solutions to a problem as bit strings and finds an optimal solution through evolution. The GA unit probabilistically selects two individuals constituting the current generation. The selection probability for each individual is proportional to the fitness of the individual, and the child generations consist of individuals with greater fitness than the members of the parent generation. Selected individuals generate a new generation through mating, mutation, and elitism as needed. In order to prevent the optimization process from converging to a locally optimal solution, the proposed GA mutates chromosomes with a certain probability after mating.

3. PROPOSED SCHEME

3.1. Problem Statement

Li and Wu proposed a PVFS to prevent false report injection and false vote injection attacks in WSNs. In their PVFS, sensor nodes are deployed on a cluster basis, and CHs perform the generation, transmission, verification, and delivery of reports, resulting in considerable energy consumption at the CH nodes.

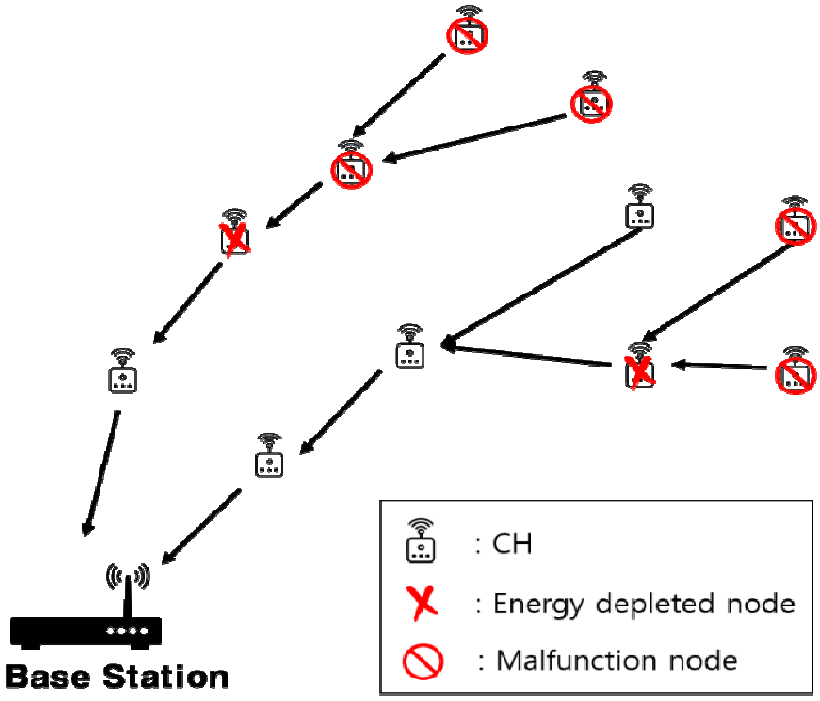

Figure 4: Event detection and reporting failure due to energy exhaustion.

In this system, if the energy of a CH is depleted, event detection and report generation become impossible for the region covered by the cluster, and it cannot serve as a report delivery node. The problem in this case is that the report transmitted from the upstream path cannot reach the BS. These problems are represented schematically in Figure 4. Because of this issue, intensive management of CH nodes with higher energy consumption is a better method for increasing the event detection rate and network lifetime of the whole network than increasing the energy efficiency of the network through member node management. In the proposed scheme, we apply GA and fuzzy logic to control the size of the report generated by the CH node, limit the role of the verification nodes of the CH as needed, adjust the lifetime of the CHs placed in the field, and increase the report transmission/reception ratio.

3.2 Assumptions

In open areas, the sensor field is cluster-based and is not attacked at the node placement stage. Since the data size and energy used to transmit the residual energy of the CH to the BS are insufficient, energy consumption is excluded as the experiment is performed. The sensor nodes used in the WSN are equipped with GPS sensor devices. After a node has been deployed, it informs the BS of its geographical location and the hop count to the BS. Therefore, the BS knows the hop count of all CH nodes.

3.3 System overview

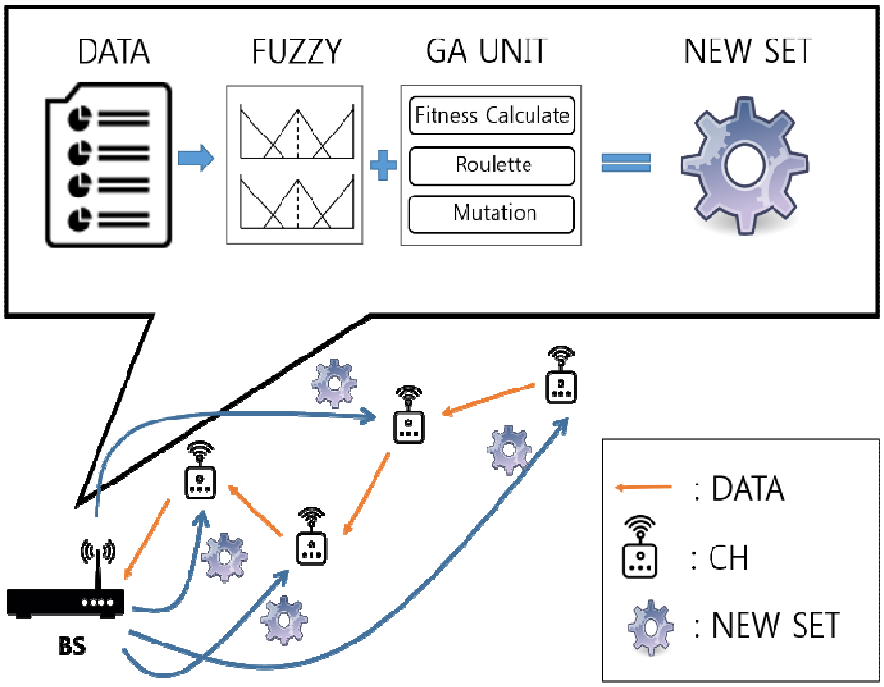

Figure 5: System overview

In the proposed scheme, the CH node transmits its residual energy to the BS every cycle. Based on the data received from each CH, the BS computes the relative energy of each CH node and the attack rate on the network and uses this as input for the optimized fuzzy system and the GA. The scheme uses a double fuzzy system. In a fuzzy system, precomputed input values are used, of which there are four kinds: attack rate, energy, relative energy, and hop count. The number of MACs to be attached to the report generated by each CH node and whether that CH can be used as a verification node are derived as a result. At the beginning of a new cycle, the BS retransmits the new configuration values to each of the CH nodes, and the process continues until the network is depleted of energy. The reason for modifying the fuzzy membership function at the BS using GA is to derive new value settings before the new cycle starts. This method can be used to distribute and conserve the energy consumption of nodes, whereas in existing PVFS, energy exhaustion is concentrated by nodes acting as verification nodes. Reports that pass through dead nodes can no longer be transmitted unless a separate routing technique is applied. It is also impossible to report events that occur in areas where dead nodes should be detected.

3.4 Fuzzy system

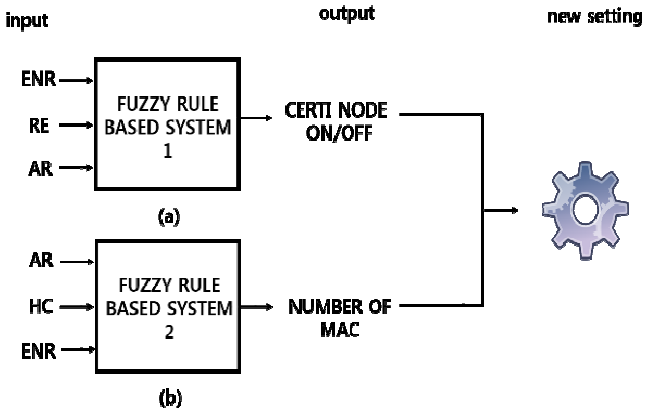

Figure 6: Fuzzy system operation of the proposed scheme.

The fuzzy system used in the proposed scheme consists of two subsystems. As shown in Figure 6, System 1 is a fuzzy system that selects the role of the verification node of the CH, and System 2 is a fuzzy system that determines the number of MACs to be attached to the report generated by the CH. Each of these receives three input values. As shown in the figure, the first fuzzy system takes the energy of the node, the relative energy of the node, and the attack rate as inputs. The second fuzzy system takes the attack rate, the hop count, and the residual energy as inputs. The reasons for using each of the inputs are described below.

- Residual Energy (ENR): The amount of energy directly affects the lifetime of the sensor node. If the amount of residual energy is high, the number of MACs can increase, and the corresponding CH node can act as a verification node.

- Relative Energy (RE): The relative energy is obtained by comparing the energy of the corresponding CH with the average residual energy of the other CHs. A CH with a high value of RE can be used as a verification node. If this value is measured to be low, the report is submitted to another node for verification.

- Hop count (HC): The hop count indicates the number of mobile hops between the corresponding CH node and the BS. This means that the forwarding role of the corresponding CH node is important, and the smaller is the hop count, the larger is the number of reports to pass through the CH node and the more energy saving is required, so the node does not carry out a verification role.

- Attack rate (AR): The attack rate is based on the number of reports inserted into the report sent to the BS. In the case of a report being dropped from a verification node, the drop count is transmitted to the BS in order to determine the total number of reports transmitted. When the attack rate is high, the fuzzy system increases the number of MACs to maintain security.

- (a) ENR = {L (LOW), M (MID), H (HIGH)}
- (b) RE = {L (LOW), M (MID), H (HIGH)}
- (c) HC = {L (LOW), M (MID), H (HIGH)}
- (d) AR = {L (LOW), M (MID), H (HIGH)}

The following is a description of the fuzzy rules. Each fuzzy system has a total of 27 rules, detailed values of which are specified in the table below.

Table 1: Fuzzy1 rules

No	Input			output
	ENR	RE	AR	**OUT**
0	LOW	LOW	LOW	OFF
7	LOW	HIGH	MID	OFF
10	MID	LOW	MID	OFF
18	HIGH	LOW	LOW	ON
26	HIGH	HIGH	HIGH	ON

Table 2: Fuzzy2 rules

No	Input			output
	AR	ENR	HC	**OUT**
1	LOW	LOW	MID	THREE
6	LOW	HIGH	LOW	FOUR
10	MID	LOW	MID	THREE
15	MID	HIGH	LOW	FOUR
22	HIGH	MID	MID	FOUR

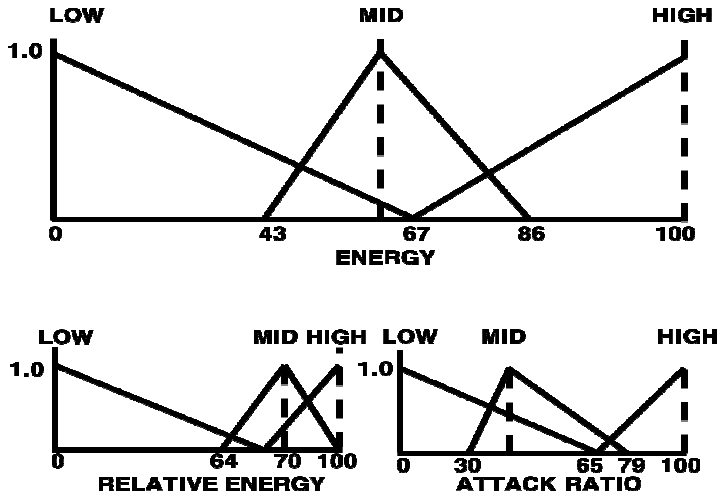

Figure 7: Membership function of Fuzzy System 1, modified by GA.

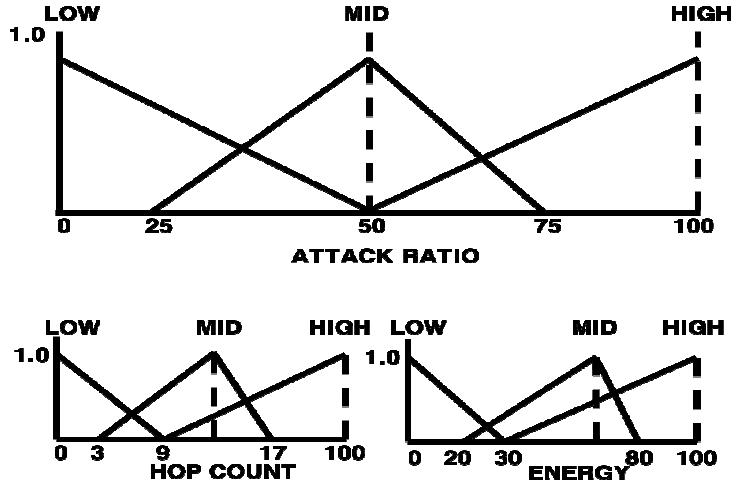

Figure 8: Membership function of Fuzzy System 2, modified by GA.

Figures 7 and 8 show the fuzzy function obtained by applying GA over 100 generations.

3.5 Fuzzy optimization using GA

A GA is used to determine the membership function of the fuzzy logic systems [10 ~ 15]. The performance evaluation function of the GA calculates the fitness of the membership function by measuring the error between the expected result and the actual result according to the fuzzy input. If the GA is used to generate the initial membership function or change the membership function to improve the security or the cycle, it is possible to obtain a membership function with close to the desired error rate in a relatively short time. Given that the number of all cases is about 10,000,000, the GA can be used to obtain a fuzzy membership function with an acceptable error rate in 50–100 generations. Figure 9 shows the process of obtaining the desired fuzzy membership function by applying the GA to a randomly generated fuzzy membership function. This creates a chromosome with a binary value for each range and then generates a descendant chromosome through fitness calculation, roulette, and mutation.

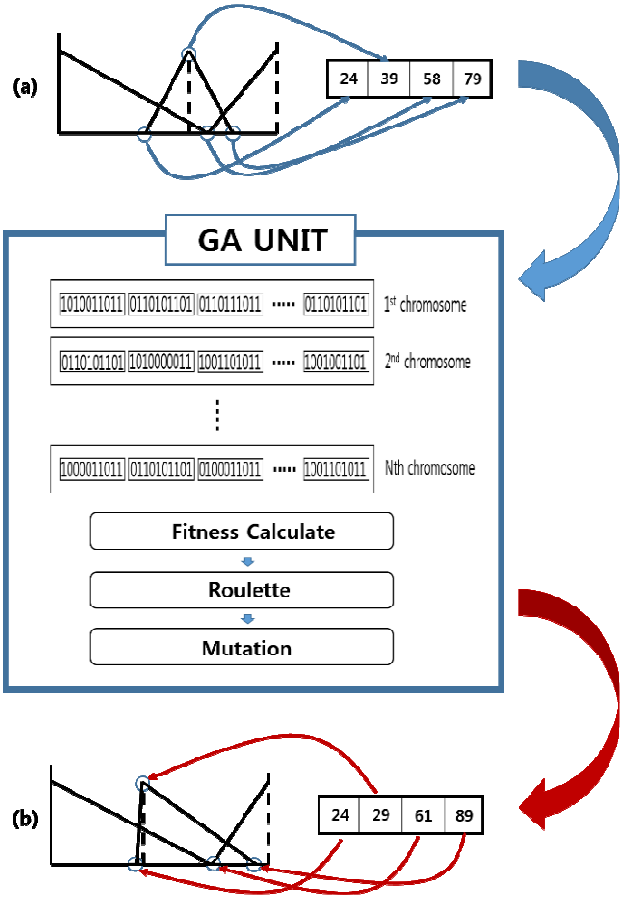

Figure 9: Modification of the membership function by GA.

Figure 9 (a) shows a fuzzy membership function that sets a random value. The random set of 10 makes up the parent generation. In the proposed system, we generate 10 membership functions that randomly select a value without expecting the result. Applying the GA to the membership function modification allows the administrator to get the desired result without having to modify the membership function arbitrarily. The value of this set becomes a chromosome, enters a GA unit, goes through a series of processes, and is output as membership function (b) with the desired error rate. In the proposed method, the elitist technique is applied, in which part of the parent chromosome with the lowest error rate is applied to the descendant chromosomes as is, thereby resulting in descendant chromosomes that produce the desired error rate in fewer generations. (A disadvantage is that it is difficult to escape unwanted values, which can be corrected by mutation rate modification) Also, if a user who uses a GA obtains a function with a performance similar to a membership function modified by an expert who does not use GA, or if the desired error rate is set to be very low, the user obtains a membership function with better performance than that of the expert.

4. Performance analysis

4.1 Experimental environment

The following table shows the experimental parameters. The field size of the sensor network used in the experiment was 800 x 800 meters, and 4000 sensor nodes were used. The number of CH nodes was 400, and the number of cluster member nodes L was 10. The sensor node

consumed 16.25 μJ to transmit a 1 byte message and 12.25 μJ when receiving. The size of the data packet was 24 bytes, and the size of one MAC was 1 byte.

Table 3: Parameters for the experiments

Parameter	Value
Field size	800 x 800 (m^2)
Number of nodes	4000
Number of experiments	400–3000
L	10
S	2–5
Tt	5
Tf	3

In the experimental environment where the number of CH nodes is 400, the average hop count is about 13, and the maximum hop count is 25. This value changes as the number of nodes and the size of the field change. The number of nodes and the size of the field are not changed in these experiments, so GA is applied only once.

4.2 Experimental results

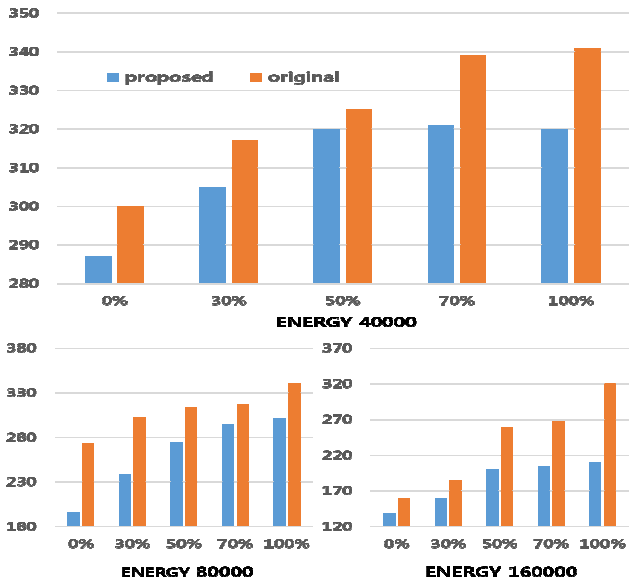

Figure 10: Number of malfunctioning nodes.

Figure 10 shows the number of malfunctioning nodes as a function of the attack rate for both the proposed technique and the pre-existing technique. When the number of CH nodes on the field is 400, the difference between the maximum number of malfunctioning nodes in the two methods is about 20. The number of malfunctioning nodes does not decrease to half or one-fourth even though the energy of the CH node increases by 2 to 4 times because the energy consumption is relatively large for the CH nodes at positions where the reports are moved. Therefore, increasing the energy of the CH node has no great effect on the number of malfunctions.

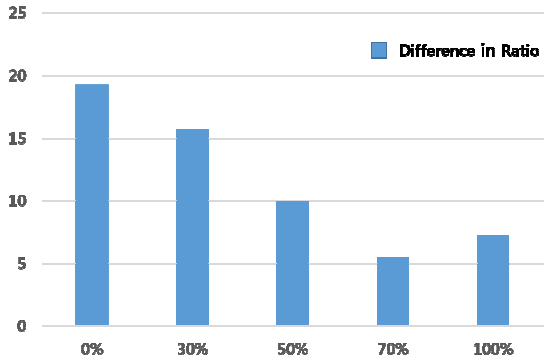

Figure 11: Difference in the ratio of the number of malfunctions as a function of attack rate

Figure 11 shows the difference in the ratio of the number of malfunctions as a function of the attack rate. The malfunction ratio refers to the number of CH nodes that cannot operate among all CH nodes. As shown in the figure, the difference in the malfunction ratio gradually decreases as the attack rate increases. This is because, if an attack occurs, the number of MACs increases, the probability that a CH becomes a verification node increases, and the energy consumed during the MAC verification process increases.

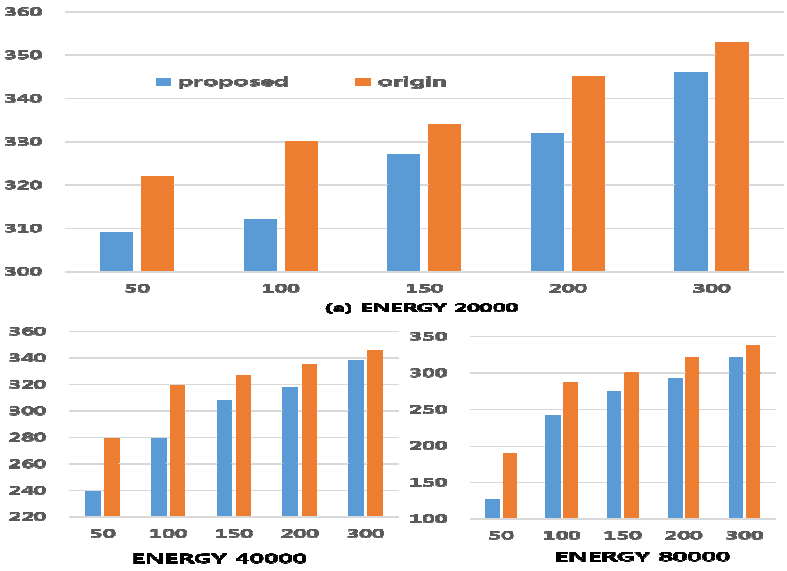

Figure 12: Comparison of number of malfunctions as a function of number of cycles.

Figure 12 shows the difference in the number of malfunctions between the two techniques as a function of cycle. At the beginning of the experiment, the number of malfunctions in both the

proposed method and the existing method surges, but only increases slowly toward the end of the experiment. This shows that there are some CH nodes that consume the majority of the energy among all CH nodes.

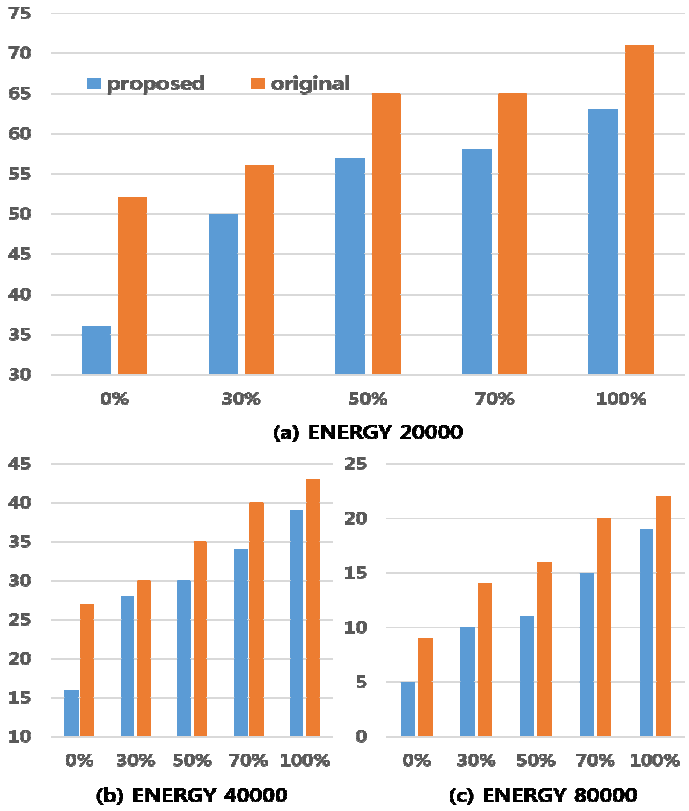

Figure 13: Number of dead CH nodes

Figure 13 shows the number of dead CH nodes as a function of attack rate; unlike the number of malfunctions, the number of dead CH nodes decreases as the energy increases by a factor of 2 to 4.

5. CONCLUSION AND FUTURE WORK

This study shows not only that the number of CH nodes that die due to energy exhaustion is important, but also that the performance and lifetime of the network are significantly affected by the death of a node that plays an important role in the network. In experiments with 400 CH nodes, the proposed method showed a maximum event detection rate improvement of about 18% compared to the existing method, and 120 cases of node malfunction reduction when a node energy of 160000 nodes were used. The larger the energy, the bigger the difference. In the experiment, the largest difference was observed when the unit energy was 160,000. The proposed system uses GA to obtain the desired membership function in a short time. As the field of the field changes with time, the experiment will be reflected. Currently, one fuzzy membership function obtained by using GA is applied to every cycle in the same way. However, future research will apply a membership function that applies changes to inputs whose fuzzy characteristics vary from cycle to cycle.

ACKNOWLEDGEMENTS

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (No. NRF-2015R1D1A1A01059484).

REFERENCES

[1] Perrig, Adrian, John Stankovic, and David Wagner. "Security in wireless sensor networks." Communications of the ACM 47.6 (2004): 53-57
[2] Zhang, Wensheng, and Guohong Cao. "Group rekeying for filtering false data in sensor networks: A predistribution and local collaboration-based approach." INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE. Vol. 1. IEEE, 2005
[3] Al-Karaki, Jamal N., and Ahmed E. Kamal. "Routing techniques in wireless sensor networks: a survey." IEEE wireless communications 11.6 (2004): 6-28.
[4] Perrig, Adrian, John Stankovic, and David Wagner. "Security in wireless sensor networks." Communications of the ACM 47.6
[5] Li, Feng, and Jie Wu. "A probabilistic voting-based filtering scheme in wireless sensor networks." Proceedings of the 2006 international conference on Wireless communications and mobile computing.ACM, 2006
[6] Zadeh, Lotfi A. "Fuzzy sets." Information and control 8.3 (1965): 338-353
[7] J. Yen and R. Langari, Fuzzy Logic: Intelligence, Control, and Information. Prentice-Hall, Inc., 1998
[8] G. Klir and B. Yuan, Fuzzy Sets and Fuzzy Logic. Prentice hall New Jersey, 1995
[9] R. Babuška, "Fuzzy Systems, Modeling and Identification," Delft University of Technology, Department of Electrical Engineering Control Laboratory, Mekelweg, vol. 4, 1996.
[10] C. L. Karr, "Design of an adaptive fuzzy logic controller using a genetic algorithm." in Icga, 1991, pp. 450-457.
[11] C. L. Karr and E. J. Gentry, "Fuzzy control of pH using genetic algorithms," Fuzzy Systems, IEEE Transactions On, vol. 1, pp. 46, 1993.
[12] D. E. Golberg, "Genetic algorithms in search, optimization, and machine learning," Addion Wesley, vol. 1989, 1989.
[13] Y. Yuan and H. Zhuang, "A genetic algorithm for generating fuzzy classification rules," Fuzzy Sets Syst., vol. 84, pp. 1-19, 1996.
[14] J. J. Buckley and Y. Hayashi, "Fuzzy genetic algorithm and applications," Fuzzy Sets Syst., vol. 61, pp. 129-136, 1994.
[15] A. Geyer-Schulz, Fuzzy Rule-Based Expert Systems and Genetic Machine Learning. Physica Verlag, 1997

Authors

Sang Hyeok Lim
Received a B.S. degree in Digital Information Engineering from Hanguk University of Foreign Studies in 2017, and is now working toward an M.S. degree in the Department of Electrical and Computer Engineering at Sungkyunkwan University.

Tea Ho Cho
Received a Ph.D. degree in Electrical and Computer Engineering from the University of Arizona, USA, in 1993, and B.S. and M.S. degrees in Electrical and Computer Engineering from Sungkyunkwan University, Republic of Korea, and the University of Alabama, USA, respectively. He is currently a Professor in the College of Information and Communication Engineering, Sungkyunkwan University, Korea.