DOI: https://doi.org/10.29121/granthaalayah.v3.i12.2015.2894

INTERNATIONAL JOURNAL OF RESEARCH – GRANTHAALAYAH
A knowledge Repository

Science

GAFS: GENETIC ALGORITHM-BASED FILTERING SCHEME FOR IMPROVING DETECTION POWER IN SENSOR NETWORKS

Tae Ho Cho [*1], **Su Man Nam** [2], **Muhammad K. Shahzad** [3]
[*1, 2, 3] College of Information and Communication Engineering, Sungkyunkwan University, KOREA

ABSTRACT

Wireless sensor networks (WSNs) have stringent energy and computational requirements. Security has become very crucial issue with the widespread acceptance of the WSNs in numerous decision-critical and hostile environments. Since sensor nodes are left unattended, they can be compromised by adversaries to launch various application layer attacks. Effective countermeasures against these attacks can lead to improved security. A probabilistic voting-based filtering scheme (PVFS) uses probabilistic filtering based on the distance to counter attacks of fabricated reports with false votes and real reports with false votes. Genetic algorithm-based filtering scheme (GAFS) uses a genetic algorithm with a fuzzy rule-based system that considers remaining energy and number of filtered votes in addition to the distance. The analysis results of the current study demonstrate the effectiveness of our scheme against these attacks in comparison with PVFS. The results show increased detection power achieved through effective verification while maintaining energy consumption.

Keywords:
Wireless sensor networks; Wireless network security; Probabilistic voting-based scheme; Genetic algorithm; False negative attacks, False positive attacks.

Cite This Article: Tae Ho Cho, Su Man Nam, and Muhammad K. Shahzad, "GAFS: GENETIC ALGORITHM-BASED FILTERING SCHEME FOR IMPROVING DETECTION POWER IN SENSOR NETWORKS" International Journal of Research – Granthaalayah, Vol. 3, No. 12(2015): 100-116.

1. INTRODUCTION

Wireless sensor networks (WSNs) have emerged as potential technologies to facilitate wireless communication for a variety of applications [1, 2]. These sensor network technologies enable the development of low-cost, low-power, and multi-functional sensors in hostile environments [3, 4]. A WSN comprises a large number of sensor nodes and a base station (BS) in a sensor field. The sensor nodes detect a real event in the densely deployed sensor field without an infrastructure [5, 6]. The BS collects the events information from the sensor nodes for critical decision making. However, these nodes can be captured and compromised because they are left unattended.

Moreover, they have limited computation, memory, and energy supply capacities [7-10]. Malicious attackers can generate attacks with various patterns, resulting in reduced network lifetime. Potential attacks include (1) fabricated reports with false votes (FRFV) [6, 11-13] and (2) real reports with false votes (RRFV) [13-15], as shown in Fig. 1. A vote is defined as a message authentication code (MAC) [12, 13, 16]. A compromised node (Fig. 1(a)) can generate a FRFV attack (Fig. 1(b)) in the absence of a real event and send this report to the BS (Fig. 1(c)). The fabricated reports drain the nodes' energy in a routing path (Fig. 1(d)). When the BS receives the fabricated report, a false alarm is generated. Another compromised node (Fig. 1(e)) can generate a RRFV attack (Fig. 1(f)) after injecting a false vote. The legitimate report is filtered out by an intermediate node due to an injected false vote, even though the report includes information about a real event. These attacks waste scarce energy and block the flow of the event reporting to the BS.

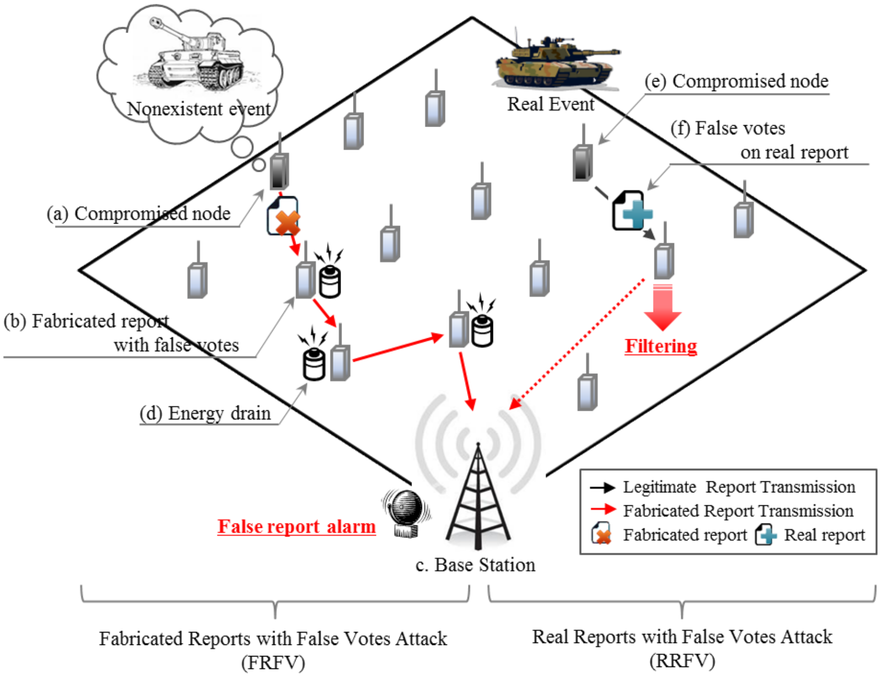

Nonexistent event
Real Event
(e) Compromised node
(f) False votes on real report
(a) Compromised node
(b) Fabricated report with false votes
Filtering
(d) Energy drain
False report alarm
c. Base Station
Legitimate Report Transmission
Fabricated Report Transmission
Fabricated report
Real report
Fabricated Reports with False Votes Attack (FRFV)
Real Reports with False Votes Attack (RRFV)

Figure 1: Multiple attacks: FRFV and RRFV attacks

Li et al. [13] proposed a probabilistic voting-based filtering scheme (PVFS) to detect these two attack types with en-route filtering. Verification nodes are selected using probabilistic decisions based on their distances (i.e., hop count). Although, it is possible to perform probabilistic optimization to select the verification nodes, the cost of verification node selection can be impractical for most sensor networks [6]. There are many optimization algorithms available; however, these algorithms require large amounts of energy and computation to determine the optimal solutions in a multi-parameter design space [17]. A genetic algorithm (GA) is a parallel and global search technique that emulates natural genetic operations [18] for effective problem solving. This algorithm is highly likely to converge on a global solution since it can simultaneously evaluate many points in the parameter space. It does not need to determine whether the search space is differentiable or continuous and can also iterate several times on each datum received [17].

The genetic algorithm-based filtering scheme (GAFS) uses a GA with a fuzzy rule-based system to effectively select verification nodes in order to improve detection power against these attacks. In order to choose the verification node, every intermediate cluster head (CH) is evaluated by the fuzzy rule-based system [14, 15, 19], which considers remaining energy, number of filtered false votes, and hop counts. The GAFS then performs a ranking selection, a one-point crossover, and a bit flip mutation operation. Our experimental results validate the effectiveness of the proposed scheme to improve security of the WSN by increasing detection power.

The rest of this paper is organized as follows. The background and motivation of this study are described in Section 2. We introduce the details of the GAFS in Section 3. Section 4 provides the analysis and results. We present the related works in Section 5. Conclusions and future work are discussed at the end of this paper.

2. BACKGROUND AND MOTIVATION

This section presents overview of the PVFS countermeasure against FRFV and RRFV attacks, as well as the motivation behind GAFS.

2.1.PROBABILISTIC VOTING-BASED FILTERING SCHEME (PVFS)

The PVFS was proposed to detect FRFV and RRFV attacks in a sensor network. This scheme is suitable for filtering in a cluster-based model. This scheme deploys a *CH* with L nodes within each cluster. The *CH* receives votes from sensor nodes in the cluster, then randomly selects votes and attaches them to a report. The scheme counters fabricated votes to detect the attacks when forwarding a report. The PVFS has three phases: (1) key initialization and assignment, (2) report generation, and (3) en-route filtering. In the key assignment phase, every node receives a key from a partition of a global key pool at the BS. Each of the source *CHs* selects verification nodes using a probability calculated based on hop counts from the intermediate *CHs* to the BS. Each of the source *CHs* randomly distributes a key from the cluster's nodes to all verification nodes in the path. Report generation and en-route filtering phases are illustrated in Fig. 2 and 3 against the FRFV and RRFV attacks, respectively.

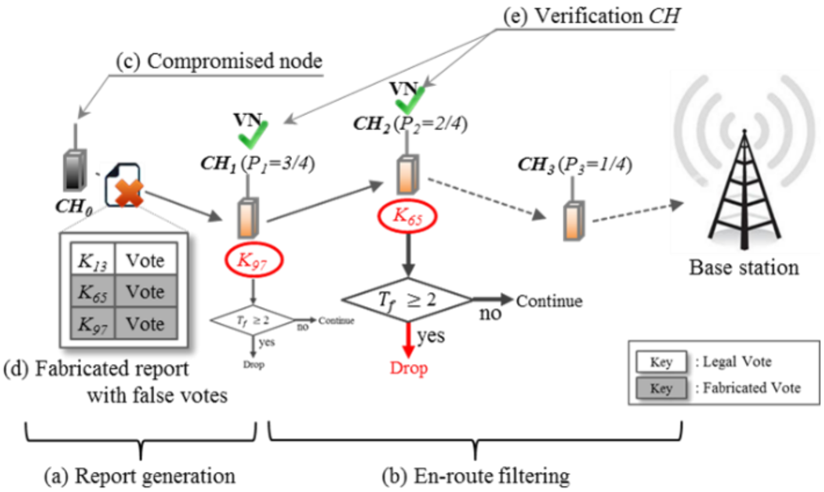


Figure 2: Detection of a FRFV attack

Fig. 2 shows the detection of a FRFV attack in the phases of report generation (Fig. 2 (a)) and en-route filtering (Fig. 2 (b)). A source CH_0 (Fig. 2 (c)) is compromised to inject false reports. We consider $s = 3$ (s is the required number of votes for a legitimate report) and $T_f = 2$ (T_f is the threshold of false votes required to drop a report). In addition, the verification nodes CH_1 and CH_2 are chosen among the intermediate *CHs* using a probability at each of them based on their hop counts. The probability is $P = d_i/d_0$ (d_i is the distance from CH_i to the BS; d_0 is the distance from CH_0 to the BS). That is, the probabilities at the intermediate *CHs* are $P_1 = 3/4$, , $P_1 = 2/4$, and $P_1 = 1/4$, respectively. In Fig. 2 (a), if CH_0 fabricates a report about a non-existing event, it injects two false votes using the captured keys K_{65} and K_{97} and attaches the votes to the fabricated report (Fig. 2 (d)). When the fabricated report arrives at CH_1, it detects a false vote using K_{97}. The fabricated report is transmitted to CH_2 since $T_f = 2$ has not yet been reached. CH_2 also detects the second false vote using K_{65}, and the fabricated report is dropped against the FRFV attack because $T_f = 2$ has been reached.

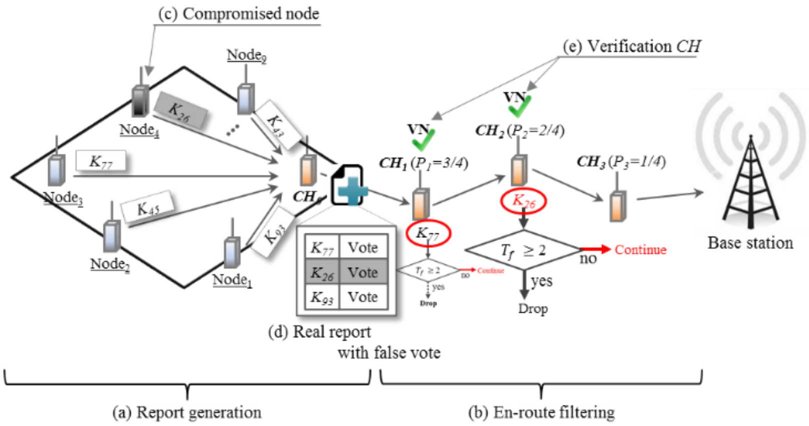


Figure 3: Detection of a FRFV attack

Fig. 3 illustrates the RRFV attack detection. A cluster of a source CH_0 has nine normal nodes ($L = 9$), including a compromised node (Fig. 3 (c)). After collecting all of the votes from the normal nodes, CH_0 randomly selects three votes ($s = 3$) including a false vote to attach to a real report (Fig. 3 (d)). After verifying the vote using K_{77} in CH_1, the real report is forwarded to CH_2. CH_2 then detects the false vote through K_{26} and continually transmits the report to CH_3 in response to the RRFV attack because $T_f = 2$ has not been reached. The legitimate report then arrives safely at the BS.

The PVFS drops the forged reports that the FRFV attack generated in the verification nodes during the forwarding process as the threshold is reached. The PVFS also detects the RRFV attack prior to reaching the threshold, and legitimate reports are transmitted to the intermediate *CHs*. Therefore, the PVFS simultaneously prevents the FRFV and RRFV attacks after verifying votes in the selected verification nodes.

2.2.MOTIVATION

FRFV attacks limit the WSN lifetime, and RRFV attacks results in blocking the flow of the event reporting to the BS. In the FRFV attacks, a fabricated report is injected with false votes, consuming unnecessary energy at each of the sensor nodes as it is forwarded via intermediate *CHs* to the BS. In the RRFV attacks, a legitimate report is generated with false votes, filtering out the real report at the verification nodes as it is forwarded to the next node. The PVFS can detect these attacks at the verification nodes before they reach the BS. The verification nodes are probabilistically selected according to hop count from the BS.

In the GAFS, we consider a vast search space for effective selection of verification nodes. When a source *CH* determines the selection of the verification nodes using the probability at each of n hops, it has 2^{n-1} cases for selection. For example, if there are 11 hops in a path, a source *CH* generates 1,024 cases. In order to find the most effective case, we propose the use of global optimization, rather than local optimization, to select the most effective verification nodes.

For evaluation of intermediate *CHs* using fuzzy rule-based systems, we consider three factors: (1) remaining energy level, (2) number of filtered false votes, and (3) hop counts. This enables us to effectively identify the verification nodes using a GA. Even though the PVFS selects the verification nodes using a probabilistic decision-making process that is based solely on hop counts, the GAFS determines the effective verification nodes using a GA with the fuzzy rule-based system. These factors are subsequently evaluated on every hop in order to determine the most effective solution, according to the fuzzy system, using the GA. After determining the global optimization for selecting the verification nodes using the GA with the fuzzy logic system, the GAFS demonstrated increased detection power with early detection of the FRFV and RRFV attacks, compared to the PVFS.

3. PROPOSED METHOD

This section describes the network model and the assumptions in Section 3.1, the scheme details in Section 3.2, and analytical examples of the GAFS in Section 3.3.

3.1. ASSUMPTIONS

The sensor nodes hold fixed positions after their deployment in the sensor field. The sensor network is comprised of a BS and a number of sensor nodes, e.g., Berkeley MICAz motes [20]. The initial paths are established through directed diffusion [21]. We use a cluster-based model [13, 22] to organize the sensor nodes for message communication. In this model, one node in each cluster is elected to be a *CH*. A *CH* is assumed to be more powerful than the normal nodes in terms of transmission range, memory size, computation, and battery power [6, 13, 14]. Since a *CH* has sufficient resources, we do not consider memory size for the GA based computation. Moreover, each *CH* selects a routing path based on hop counts from the BS to the *CH*. We further assume that every *CH* forwards reports to the BS using this path. The FRFV and RRFA attacks are generated to drain energy of sensor nodes and block the flow of the event reporting to the BS. We consider bidirectional communication paths (i.e., if node A can communicate to node B, then node B can also communicate to node A). The key assignment is achieved by using keys from global key pool of the BS. In this paper, we only consider the FRFV and RRFA attacks from multiple compromised nodes.

3.2.SCHEME DETAILS

In this section, a detailed description of the GAFS is presented. Section 3.3.1 gives an overview of the proposed scheme. An initial population is formulated to extract the best individual using the GA in Section 3.3.2. A detailed description of the fitness function using the fuzzy logic system is illustrated with this population in Section 3.3.3. The design of the GA with the fuzzy rule-based fitness function is presented in Section 3.2.4.

3.2.1. SCHEME DETAILS

The GAFS effectively selects verification nodes using a GA based on a fuzzy rule-based system to detect the FRFV and RRFA attacks. The GA maintains a population of individuals (i.e., chromosomes) in which each individual represents a potential solution to the problem at hand. The population is implemented as parts of the data structure [17]. Each solution is evaluated to measure its fitness, which indicates how close it is to the optimal solution [14]. The *CH* evaluates every intermediate *CH* according to the fuzzy system using three factors: (1) remaining energy level, (2) number of filtered false votes, and (3) hop counts in order to select the verification nodes through the GA.

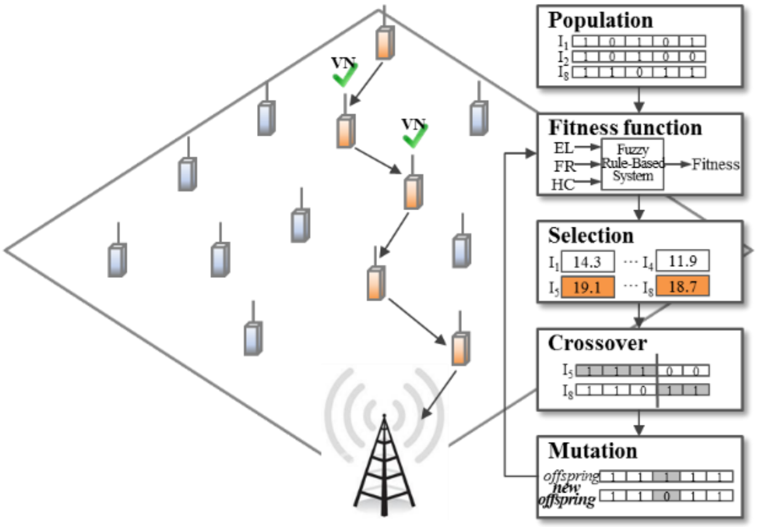


Figure 4: Overview of the GAFS

Error! Reference source not found. illustrates the selection of the verification nodes using the GA. A source *CH* generates eight individuals (i.e., I_1–I_8) in a path without duplicating the binary codes (i.e., 0, 1). The GAFS selects the two individuals (i.e., I_5 and I_8) with the highest fitness values using a ranking selection. Then, the two selected individuals produce a *new offspring* through a one-point crossover. The *offspring* may be mutated at a defined probability. Next, the fitness of the *offspring* is compared to the minimum fitness among the eight individuals in order to replace the minimum fitness after mutation. These phases are repeated ten times to determine the most effective individual. The source *CH* selects verification nodes according to the best individual after finishing the GA execution. Therefore, the GAFS selects the most effective verification nodes after evaluating the intermediate *CHs* using the GA according to the fuzzy rule-based system.

3.2.2. SCHEME DETAILS

The GAFS can dynamically adapt to the network conditions based on the three factors considered for evaluation of intermediate *CHs*. After executing the GA, the proposed scheme effectively selects verification nodes based on the highest fitness value of an individual.

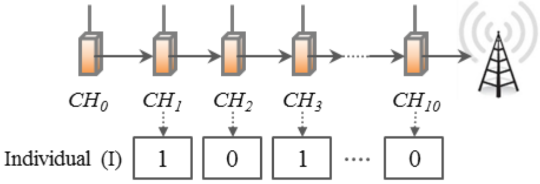
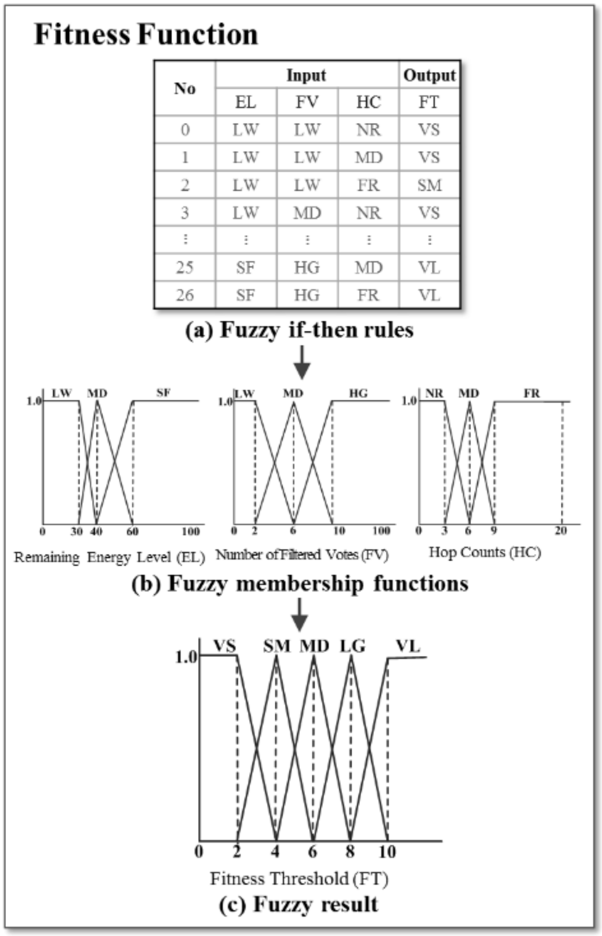


Figure 5: Example of an individual routing path

Fig. 5 illustrates an individual in a routing path from CH_0 to the BS. All genes are always reserved for intermediate *CHs* (CH_1-CH_{10}). A gene is composed of a binary value generated by the probability $P = d_i / d_0$. The genes of the intermediate *CHs* represent either 1 or 0 in a binary code based on their hop counts. The binary code of each individual is generated using probability without duplication, as in the selection of a verification node in the PVFS. The binary length of the individual is dynamically based on the hop counts.

3.2.3. FITNESS FUNCTION USING FUZZY LOGIC

In the GA, the evaluation phase is implemented in our scheme by using the fuzzy logic system. We use the factors of remaining energy level (EL), and number of filtered false votes (FV), and hop count (HC) to effectively select verification nodes. The EL is a key parameter because each sensor has limited energy. If the intermediate *CH* has a higher level of energy, it is more likely to be selected as a verification node for authentication than are the *CHs* with low energy levels. However, if the remaining energy level is low, an intermediate *CH* transmits the report to the next *CH* without authentication. The FV is most important in terms of security. If the frequency of the FRFV and RRFV attacks is high, more verification *CHs* can be selected to increase the detection power. If the frequency of the attacks is low, fewer verification *CHs* can be selected in order to reduce energy consumption. The HC is another important input. If this value is large, the number of verification nodes can be increased close to a source *CH* to improve early detection. If the HC value is small, intermediate *CHs* can forward the reports instead of acting as verification nodes in order to decrease energy consumption at the BS.



No	Input			Output
	EL	FV	HC	FT
0	LW	LW	NR	VS
1	LW	LW	MD	VS
2	LW	LW	FR	SM
3	LW	MD	NR	VS
⋮	⋮	⋮	⋮	⋮
25	SF	HG	MD	VL
26	SF	HG	FR	VL

(a) Fuzzy if-then rules

(b) Fuzzy membership functions

(c) Fuzzy result

Figure 6: Fitness function using a fuzzy rule-based system

Fig. 6 shows the fitness function containing: (a) the fuzzy if-then rules, (b) the fuzzy membership functions, and (c) the fuzzy result. We define the fitness function using the fuzzy rule-based system in order to more accurately evaluate each individual. When a gene is assigned to a *CH* with a binary value of 1, the verification node of the gene is evaluated. The input factors for fuzzy inference are EL, FV, and HC, and the output factor is the fitness threshold (FT). The labels of the fuzzy variables are as follows:

- EL = {LW (Low), MD (Medium), SF (Sufficient)}
- FV = {LW (Low), MD (Medium), HG (High)}
- HC = {NR (Near), MD (Middle), FR (Far)}

These labels represent the fitness threshold as follows:

- FT = {VS (Very Small), SM (Small), MD (Medium), LG (Large), VL (Very Large)}

In the fuzzy if-then rules, three input factors with three labels each were used to obtain a total of 27 ($= 3 \times 3 \times 3$) rules based on the analysis and experimental results.

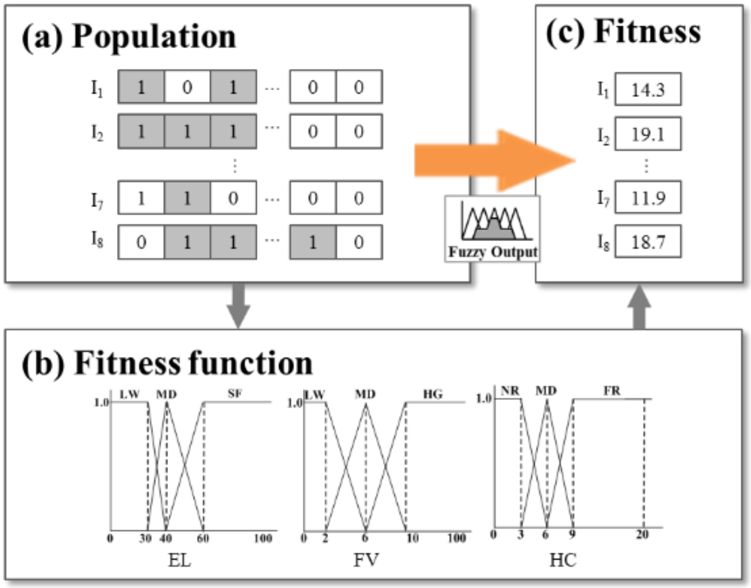


Figure 7: Fitness Function Results

Fig. 7 shows the fitness results for the eight individuals evaluated using the fitness function. For (a), the population, the source *CH* selects the sizes of the individuals according to hop count, and binary codes are generated for each gene in the individual. For (b), the fitness function, each gene is evaluated according to the fuzzy membership function. For (c), fitness, each individual has a total fitness of its genes evaluated as 1 for its ranking selection.

3.2.4. DESIGN OF THE GENETIC ALGORITHM

Fig. 8 represents the five phases of the GA: (a) ranking selection, (b) one-point crossover, (c) flip bit mutation, (d) fitness function, and (e) the replacement and ranking selection. As shown in phase (a), the ranking selection is used to improve the quality of the population. In the ranking selection phase, the two individuals with the highest fitness are chosen for crossover. As shown in phase (b), the selected individuals are examined during crossover to identify the one with the greatest fitness. A one-point crossover then generates an *offspring* in order to improve the fitness value. The one-point crossover is a random number between 0 and *n* (i.e., hop count). Individuals I_2 and I_8 interact to generate the *offspring* based on a random number. In phase (c), mutation is required to avoid a local optimum. In order to solve this problem, we set the flip bit mutation rate to 1%, which is an extremely low probability. The binary code of a gene was inverted in the *offspring* generated between 0 and *n* to produce the *new offspring*. The fitness function accurately measures *offspring* quality in phase (d). In phase (e), the fitness of the *new offspring* is then compared to the minimum fitness result in order to update the minimum fitness. The five phases are repeated ten times in order to produce the most effective individual based on the three input factors. Subsequently, CH_0 selects the individual with the highest fitness value among the eight individuals to determine the effective verification nodes.

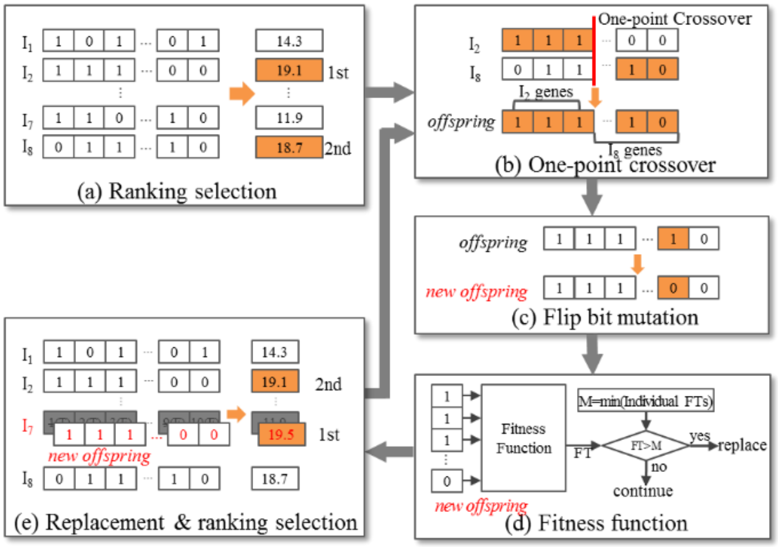


Figure 8: Design of the GA

3.3.ANALYTICAL EXAMPLE

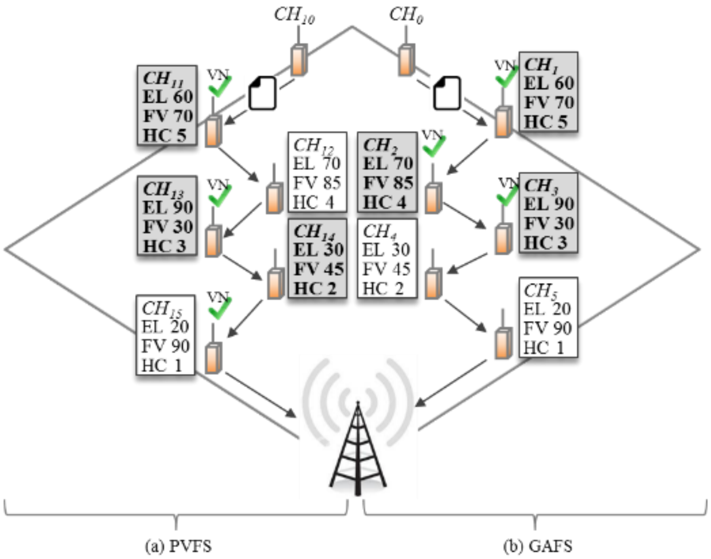


Figure 9: Verification nodes of the PVFS and GAFS

Figure 9 depicts how the PVFS (Fig. 9 (a)) and the GAFS (Fig. 9 (b)) select verification nodes. As shown in Fig 9 (a), CH_{10} selects the verification nodes using the probability values of intermediate *CHs* in the key initialization and assignment phase. *CHs* 11, 12, 13, 14, and 15 have probability values of $P_{11} = 5/6$, $P_{12} = 4/6$, $P_{13} = 3/6$, $P_{14} = 2/6$, and $P_{15} = 1/6$, respectively. *CHs* 11, 13, and 14 are thus fixed as the verification nodes based on only hop counts. In the GAFS (Fig. 9 (b)), CH_0 selects the verification nodes using a GA, according to the network conditions. The fitness of each *CH* in a path is computed according to the fuzzy rule-based system based on the three factors. CH_0 identifies the effective verification nodes after executing five phases, as shown in Figure 8, and chooses CH_1, CH_2, and CH_3 to improve the detection power.

4. SIMULATION ANALYSIS

In this section, cost analysis and experimental results are described in detail in Sections 4.1 and 4.2, respectively.

4.1.COST ANALYSIS

This section describes the energy consumption cost analysis using the GA with the fuzzy rule-based system in a source *CH*. As discussed in Section 3.2.3, a total of 27 rules in the fuzzy rule-based system are evaluated for each individual in the GAFS. Furthermore, for the GA, three algorithms (ranking selection, one-point crossover, and flip bit mutation) are executed. A cost analysis comparison for the GA with the fuzzy rule-based system and vote verification is summarized in Table 1.

Table 1: Cost analysis comparison

	GA with fuzzy system	Vote verification
Addition Operations	8,640 operations/10 times	3,520 operations/a vote
Energy Consumption	183.75 µJ	75 µJ

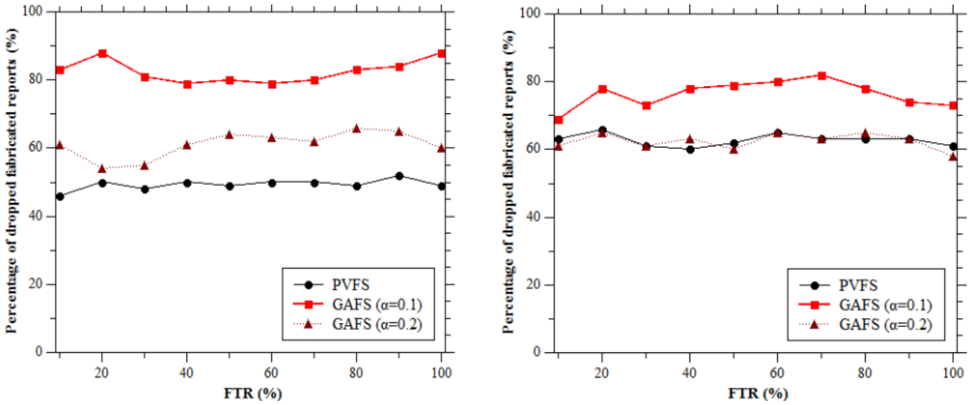
In the above table, an evaluation is performed for the fuzzy rule-based system according to the standard additive model [19]. For the expression of the standard additive model in the fuzzy rule-based system, the number of multiplication operations, which is equivalent to twice the calculations using addition operations, is 378 (=(27×4+27×3)×2), and the number of addition operations is 54 (27×2). Thus, the total number of operations is 432. For the GA, the ranking selection for the eight individuals executes 96 operations based on [23], a one-point crossover for two individuals executes 70 operations, and a bit flip mutation for one individual executes 50 operations. Because two-byte words are standard, the GA executes 864 operations. Therefore, the total number of addition operations for the GA with the fuzzy rule-based system is 8,640 over10 times, which is equal to 183.75 µJ.

In contrast, a one-vote verification computes messages that are split into 44 bytes in a keyed-hash message authentication code (HMAC) [16] as a hash function is operated. The total number of addition operations is 80. The total number of operations for two-byte words and the hash function is 3,520 (=80×2×44/2) which is equal to 75 µJ.

The GA computation is approximately 245% greater than the vote verification. The GA with the fuzzy rule-based system computation consumes more energy than the vote verification. However, the frequency of the vote verifications (for s=5, 5×75=375µJ) is higher than the execution of the GA with the fuzzy rule-based system (183.75 µJ) that is used to select verification nodes. Therefore, the use of the GA with the fuzzy rule-based system is feasible on a *CH*.

4.2.EXPERIMENTAL RESULTS

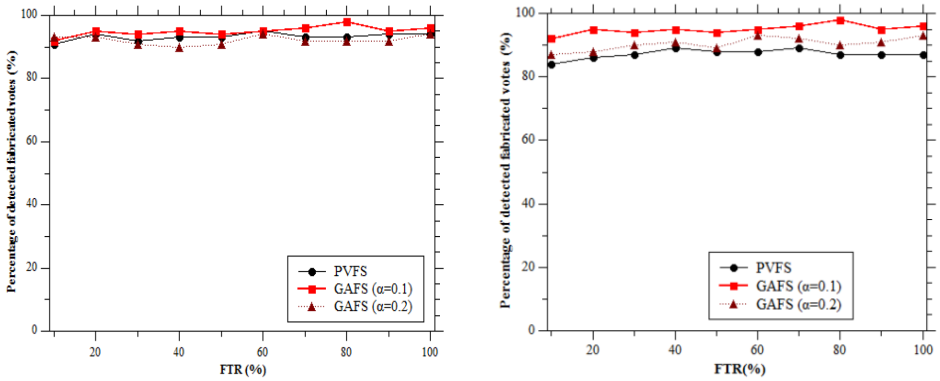
The simulation experiment is performed to evaluate the GAFS and compare it with the PVFS. The experimental environment has a field size of 2000×2000 m^2. Each cluster in this field has a size of 100×100 m^2 consisting of one *CH* and nine randomly deployed nodes. The compromised nodes generate FRFV and RRFV attacks using false votes corresponding to the false traffic ratio (FTR) relative to the total number of events. For this simulation, the energy consumption model in Ye *et al.* [1] is used. Each node uses 16.25 µJ/byte to transmit data and 12.5 µJ/byte to receive data, each vote generation consumes 15 µJ/byte, and one vote verification consumes 75 µJ/byte. The size of each report is 24 bytes, and the size of a single vote is 1 byte. In the comparative analysis, the required number of votes for a report is $s = 5$, and the threshold of false votes is $T_f = 2$. For example, when an event occurs in a cluster, a *CH* selects five votes to be attached to a report. Then, if the verification node detects two false votes out of these five votes, then it drops the report. The design parameters *α* is the ratio of the attacks per legitimate reports, which is predefined in the initialization phase.



(a) FRFV with compromised nodes=6 (b) FRFV with compromised nodes=10

Figure 10: Defense ratio against an FRFV attack with compromised nodes=6 and 10

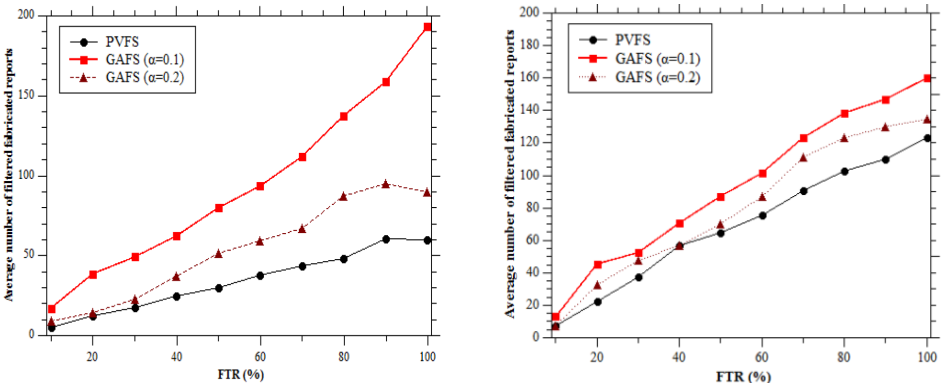
Fig. 10 compares the GAFS with the PVFS for their percentages of dropped fabricated reports against FRFV attacks for two different values of design parameter α for the GAFS. In case (a), the GAFS (α=0.1) successfully dropped an average of 80% of the fabricated reports while the PVFS averaged 45%. In case (b), the number of compromised nodes increased from 6 to 10. It is observed that, with a larger number of compromised nodes, the performance difference between the two schemes decreases. However the performance of the GAFS (77% for α=0.1) is still better than that of the PVFS (62%). We also note that, for α=0.1, the detection power is more than for α=0.2. This is because, for lower α, the GA is executed more times to effectively select the verification nodes.



(a) FRFV with compromised nodes=6   (b) FRFV with compromised nodes=10

Figure 11: Detection of FRIA and FVIA

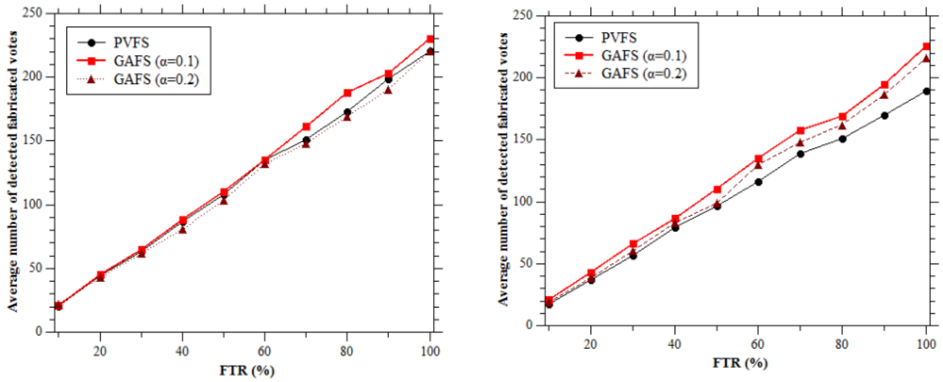
Fig. 11 compares the GAFS with the PVFS for their percentages of detected fabricated votes in reports against RRFV attacks for two different values of α for GAFS. In case (a), the GAFS (α=0.1 and 0.2) and the PVFS show similar performances for the detection power; however, the GAFS (95% for α=0.1) performs better than the PVFS (93%). In case (b), the GAFS (93% for α=0.1) is still better than the PVFS (87%).



(a) FRFV with compromised nodes=6   (b) FRFV with compromised nodes=10

Figure 12: Average number of dropped fabricated reports that traveled within four hops

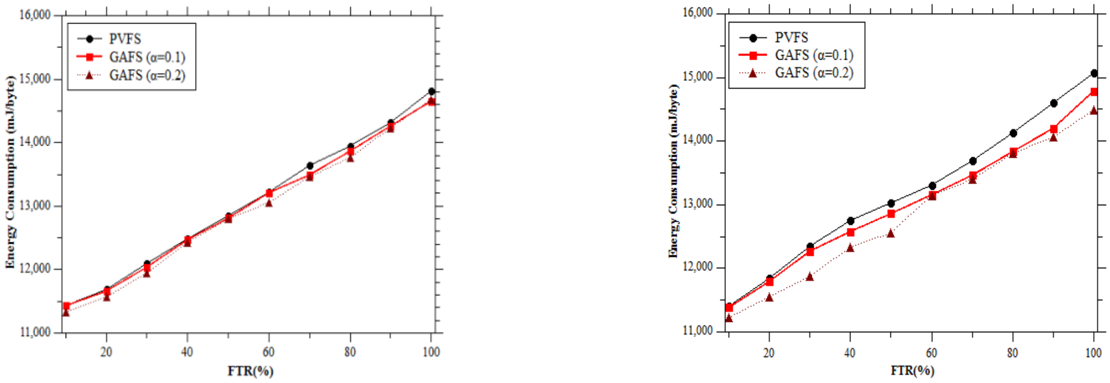
Fig. 12 shows the average number of dropped fabricated reports that traveled within four hops for different values of FTR. We measured how early the filtered reports could be dropped after FRFV attacks are generated. In case (a), the GAFS (α=0.1) performs significantly better with increased FTR by dropping more fabricated reports. In case (b), as we increase the number of compromised nodes to 10, the GAFS performs better, although the performance difference is small. Therefore, the GAFS improves the early detection power for the detection of fabricated reports against FRFV attacks, compared to the PVFS.



(a) FRFV with compromised nodes=6 (b) FRFV with compromised nodes=10

Figure 13: Average number of filtered votes that traveled within four hops

Fig. 13 shows the average number of detected fabricated votes in a report within four hops of compromised nodes. In this figure, we focus on early detection against RRFV attacks. In case (a), the GAFS (α=0.1 and 0.2) and the PVFS performances are very similar. In case (b), after increasing the number of compromised nodes, the GAFS (α=0.1 and 0.2) performs better than the PVFS for early detection of fabricated votes in RRFV attacks.



(a) Energy consumption with compromised nodes=6 (b) Energy consumption with compromised nodes=10

Figure 13 Average number of filtered votes that traveled within four hops

Fig. 14 shows the network-level energy consumption for different numbers of compromised nodes against FRFV and RRFV attacks. In case (a), the GAFS (α=0.1 and 0.2) and the PVFS performances are similar for energy consumption when there are 6 compromised nodes. In case (b), the GAFS saves more energy than the PVFS when subjected to attacks in the presence of 10 compromised nodes. Therefore, the GAFS demonstrates improved detection power while it does not incur more energy consumption than the PVFS.

5. RELATED WORKS

In the past decade, a number of research papers have addressed sensor network security by using en-route filtering to detect and drop attacks. These proposals (such as SEF [12], IHA [24], and KIF [25]) differ in terms of energy efficiency, en-route detection power, and early detection power [11]. *Yu et al.* [12] presented statistical en-route filtering (SEF) to probabilistically detect false reports after receiving MACs from neighbors during the events generation. *Zhu et al.* [24], proposed interleaved hop-by-hop authentication (IHA) to guarantee that the BS will detect false reports when t nodes (security threshold) are compromised. *Lee et al.* [25] proposed key inheritance-based filtering (KIF), which prevents forwarding of false reports to subsequent nodes.

In [11], the adaptive selection of filtering (ASF) scheme was proposed, in which three schemes, SEF, IHA, and KIF, are loaded on each of the nodes. The BS periodically computes fitness values to select one of the schemes depending on network conditions by using a fuzzy rule-based system. The algorithms then forward and confirm legitimate reports based on the selected scheme. In [14], a GA-based membership function optimizer for fuzzy adaptive filtering (GAOFF) was presented. The efficiency of the membership functions was measured based on the simulation results and was optimized by the GA. GAOFF improved the energy efficiency, en-route detection power, and early detection power using GA and a fuzzy rule-based system. All of these proposed methods (SEF, IHA, KIF, ASF, and GAOFF) are guaranteed to effectively detect only FRFV attacks in a sensor network.

6. CONCLUSION AND FUTURE WORK

An adversary can seriously harm sensor networks by launching complex attacks such as FRFV and RRFV attacks. Such attacks, which are generated at the application layer, increase unnecessary energy consumption and inhibit the flow of event reporting. In the GAFS, the GA with the fuzzy rule-based system demonstrated its ability to effectively select verification nodes based on the remaining energy, the number of false votes, and the hop counts. A source *CH* used the GA to extract the best individual among eight individuals. Experimental results demonstrate the effectiveness of our proposed method with increased detection power as compared with the PVFS while maintaining the energy consumption. Therefore, our proposed method offers an effective solution capable of providing global optimization [14, 26].

In this paper, the GAFS resulted in the following contributions:

- Increased detection power
- Early detection
- Global optimization

In future work, we will evaluate the performance of our proposed method against other types of attacks.

7. ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2013R1A2A2A01013971)

8. *REFERENCES*

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," Communications Magazine, IEEE, vol.40, no.8, pp.102-114, Aug.

[2] *A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor network," Communication of the ACM, vol.47, pp.53-57, 2004.*

[3] *B. Przydatek, D. Song, and A. Perrig, "SIA: Secure information aggregation in sensor networks,"SenSys '03 Proceedings of the 1st international conference on Embedded networked sensor systems, pp.255-265, 2003.*

[4] *J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," Wireless Communications, IEEE, vol.11, no.6, pp.6-28, 2004.*

[5] *K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," Ad Hoc Networks, vol.3, pp.325-349.*

[6] *H. Y. Lee and T. H. Cho, "Fuzzy adaptive selection of filtering schemes for energy saving in sensor networks," IEICE Transactions on Communications, vol.E90-B, no.12, pp.3346-3353.*

[7] *S. Zhu, S. Setia, and S. Jajodia, "LEAP+: Efficient security mechanisms for large-scale distributed sensor networks," ACM Trans. Sen. Netw., vol.2, no.4, pp.500-528, nov.*

[8] *S. Slijepcevic, M. Potkonjak, V. Tsiatsis, S. Zimbeck, and M. B. Srivastava, "On communication security in wireless ad-hoc sensor networks," Enabling Technologies: Infrastructure for Collaborative Enterprises, 2002. WET ICE 2002. Proceedings. Eleventh IEEE International Workshops on, pp.139-144.*

[9] *S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanisms for large-scale distributed sensor networks," Proc. of the 10th ACM conf. on Computer and communications security, ACM, pp.62-72.*

[10] *C. H. Lim, "LEAP++: A robust key establishment scheme for wireless sensor networks," Distributed Computing Systems Workshops, 2008. ICDCS '08. 28th International Conference on, pp.376-381.*

[11] *H. Y. Lee and T. H. Cho, "A scheme for adaptively countering application layer security attacks in wireless sensor networks," IEICE Transactions on Communications, vol.E93.B, no.7, pp.1881-1889, 2010.*

[12] *F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," Selected Areas in Communications, IEEE Journal on, vol.23, no.4, pp.839-850.*

[13] *F. Li, A. Srinivasan, and J. Wu, "PVFS: A probabilistic voting-based filtering scheme in wireless sensor networks," International Journal of Security and Network, vol.3, no.3, pp.173-182.*

[14] *H. Y. Lee and T. H. Cho, "Optimized fuzzy adaptive filtering for ubiquitous sensor networks," IEICE Transactions on Communications, vol.E94.B, pp.1648-1656, Jun.*

[15] *T. P. Nghiem and T. H. Cho, "A fuzzy-based interleaved multi-hop authentication scheme in wireless sensor networks," Journal of Parallel and Distributed Computing, vol.69, no.5, pp.441-450, May.*

[16] *H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-hashing for message authentication," Internet Engineering Task Force, vol. Request for Comments (RFC) 2104, pp.1-11.*

[17] *J. W. Kim, Y. K. Moon, and B. P. Zeigler, "Designing fuzzy net controllers using GA optimization," Computer-Aided Control System Design, 1994. Proceedings, IEEE/IFAC Joint Symposium on, pp.83-88.*

[18] *Z. Michalewicz, Genetic algorithms + data structures = evolution programs (3rd ed.), Springer-Verlag, London, UK, UK, 1996.*

[19] J. Yen and R. Langari, Fuzzy logic: Intelligence, control, and information, Prentice Hall, 1991.

[20] Crossbow technology Inc., "MICAz," http://www.xbow.com.

[21] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," Proceedings of the 6th annual international conference on Mobile computing and networking, pp.56-67.

[22] Zhen Yu, "A dynamic en-route filtering scheme for data reporting in wireless sensor networks," Networking, IEEE/ACM Transactions on, vol.18, no.1, pp.150-163.

[23] T. Blickle and L. Thiele, "A comparison of selection schemes used in evolutionary algorithms," Evol. Comput., vol.4, no.4, pp.361-394, dec.

[24] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks," Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on, pp.259-271, May 2004.

[25] H. Y. Lee and T. H. Cho, "Key inheritance-based false data filtering scheme in wireless sensor networks," Lecture Notes in Computer Science, vol.4317, pp. 116-127.

[26] P. Manjunatha, A. K. Verma, and A. Srividya, "Fuzzy based optimized routing protocol for wireless sensor networks," Advances in Wireless Sensors and Sensor Networks, vol.64, pp.273-282.