Fuzzy-Based Adaptive Countering Method against False Endorsement Insertion Attacks in Wireless Sensor Networks

Hae Young Lee

Department of Information Security, Seoul Women’s University, 621 Hwarang-ro, Nowon-gu, Seoul 139-774, Republic of Korea

Correspondence should be addressed to Hae Young Lee; whichmeans@gmail.com

Received 2 June 2014; Revised 19 August 2014; Accepted 7 September 2014

Academic Editor: Muhammad Khurram Khan

Copyright © 2015 Hae Young Lee. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless sensor networks (WSNs) are vulnerable to false endorsement insertion attacks (FEIAs), where a malicious adversary intentionally inserts incorrect endorsements into legitimate sensing reports in order to block notifications of real events. A centralized solution can detect and adaptively counter FEIAs while conserving the energy of the forwarding nodes because it does not make the nodes verify reports using cryptographic operations. However, to apply this solution to a WSN, the users must carefully select 10 or more security parameters, which are used to determine the occurrences of FEIAs. Thus, an inappropriate choice of a single parameter might result in the misinterpretation of or misdetection of FEIAs. Therefore, the present study proposes a fuzzy-based centralized method for detecting and adaptively countering FEIAs in dense WSNs, where two fuzzy rule-based systems are used to detect an FEIA and to select the most effective countermeasure against the FEIA. A major benefit of the proposed method is that the fuzzy systems can be optimized automatically by combining a genetic algorithm and a simulation. Thus, users only need to write a model of the WSN to apply the proposed method to a WSN. The improved performance with this method is demonstrated by simulation results.

1. Introduction

A wireless sensor network (WSN) comprises a large number of tiny, battery-powered sensor nodes, which monitor the surrounding area and report events of interest (e.g., the appearance of a vehicle) that occur in the area to base stations (BSs) [1]. These nodes use wireless communications and are often deployed in hostile environments, such as battlefields [2], which would result in various security concerns. Moreover, they are not attended by the users, so that a malicious adversary could physically capture some of the nodes without being detected [3]. The adversary might compromise all of the secret information held by the nodes [4], which could then allow the adversary to launch various insider attacks. For example, the adversary could inject “fabricated” reports with “correct” message authentication codes (MACs) (or endorsements) into the WSN through the nodes (Figure 1(a)), thereby generating false alarms and/or shortening the lifespan of the WSN [5]. In contrast to computer networks, it should be noted that a false alarm in a WSN might lead to a real-world response (e.g., dispatching a patrol unit to the area).

Alternatively, the adversary might intentionally insert “incorrect” MACs into “legitimate” reports (Figure 1(b)) in order to intercept the real events being reported to users [6]. This would make it very difficult to distinguish real events from fake ones at the BSs, where the former fabricated reports could be accepted by BSs because of the correct MACs, whereas the incorrect MACs in the latter legitimate reports would result in the reports being discarded during forwarding processes.

Fortunately, major research efforts [2, 4, 5, 7–11] have been made to alleviate the severe damage caused by the former type of security attack, which is called a false data injection attack (FDIA). However, relatively less research [6, 12–15] has focused on investigations of countermeasures to combat the latter type of security attack, which is called a false endorsement insertion attack (FEIA). Note that the detection of FEIAs might be much more important than that of FDIA because of the important role of WSNs in reporting critical events. FEIA countermeasures can be classified into two types: distributed [6, 12–14] and centralized [15]. The centralized solution proposed by Lee and Cho [15], which
is called the adaptive counting scheme (ACS), is more energy-efficient during "peacetime" (i.e., when an FEIA does not occur) because it does not involve en-route verification processes for reports. By contrast, distributed solutions generally require that forwarding nodes verify the correctness of reports using cryptographic operations, which might incur high computation overheads and extra energy consumption. However, a major drawback of ACS is that users need to determine many security parameters, which are used for the detection of security attacks. Therefore, the inappropriate selection of these parameters could result in the false positive detection of real events (i.e., events may be misinterpreted as FDIA or FEIA) or the false negative detection of security attacks (i.e., FDIA or FEIA might not be detected).

In this study, I propose a fuzzy-based version of ACS to combat FEIAs in WSNs used for object-tracking applications. The proposed method employs two fuzzy rule-based systems to detect and counter FEIAs: one to detect FDIA and another to select the countermeasures. After an FEIA has been detected by the former system, the latter selects the most effective countermeasure against the FEIA. These fuzzy systems can be loaded onto a BS where they can use the sensing reports collected by the BS. Thus, the proposed method provides a centralized solution, which can obtain extra energy savings, especially during peacetime. The proposed method has the following merits compared with ACS, which is another centralized solution. (1) The fuzzy systems can be optimized automatically by combining a genetic algorithm with a simulation [16], which virtually eliminates the manual parameter settings used in ACS. (2) Fewer input factors are required to detect FEIAs and to select the countermeasures, which reduces the space requirements and computation complexity. (3) These input factors can be obtained without any special mechanisms or equipment, such as the Global Positioning System (GPS). (4) Errors during the detection of FEIAs can be reduced by the approximate reasoning capacity of fuzzy logic. The effectiveness of the proposed method is demonstrated by simulation results.

The remainder of the paper is organized as follows. Section 2 provides brief descriptions of ACS and the two FEIA countermeasures used by the proposed method, as background. Section 3 describes the method in detail. The simulation results are presented in Section 4. Finally, the conclusions of this study and future work are discussed in Section 5.

2. Background

This section briefly describes ACS, which is a crisp solution for the detection of FDIAs and FEIAs, and the two FEIA countermeasures employed by the proposed method.

2.1. FEIA Countermeasures. In the proposed method, two countermeasures are used to counter FEIAs: the probabilistic voting-based filtering scheme (PVFS) [6] and multipath-based en-route filtering scheme (MEF) [12].

In [6], Li et al. first showed that the collaborative generation of sensing reports causes vulnerability to FEIAs, where a few compromised nodes can be used to intentionally insert incorrect MACs into legitimate reports, thereby suppressing the notification of real events. They then proposed PVFS as a countermeasure against FEIAs. In PVFS, reports are verified while they are being forwarded toward BSs, which is the case with most FDIA countermeasures. However, in contrast to FDIA countermeasures, a single verification failure for a report does not mean that the report is discarded, because a report is only discarded when the number of failures reaches a predefined threshold value. Thus, PVFS could confer some resilience against FEIAs. In addition, a report with a certain number of successful verifications can be delivered to a BS without further verification in order to obtain extra energy savings.

Kim and Cho [12] proposed MEF, which can also be used to counteract FEIAs. In MEF, a single verification failure for a report means that the report is dropped, which is the case with most FDIA countermeasures. However, multiple copies of a report with different MACs are forwarded to BSs via different routing paths. Thus, although a few of the incorrect MACs inserted by a malicious adversary are collected during collaborative generation, some copies of the report can be delivered to the BSs. Thus, MEF could confer some resilience against FEIAs. Other FEIA countermeasures that provide distributed solutions can be found in [13, 14].

In [15], the characteristics of major FEIA countermeasures were analyzed to assess their adaptive use. These countermeasures can provide some resilience against FEIAs up to a certain number of compromised nodes, that is, a security threshold value. However, none of them can guarantee that every event is reported to users, even if the number of compromised nodes does not exceed the threshold value. In addition, they differ from each other in terms of their reliability and energy efficiency during the delivery of reports. For example, the energy efficiency of PVFS is usually better than that of other methods because PVFS uses single-path routing and it minimizes en-route verification. However, PVFS might lack reliability during the long-haul delivery of reports because of the use of single-path routing. Therefore, in terms of reliability and energy efficiency, we should select the most “effective” countermeasure against each FEIA by
considering the status of the WSN and the objective of the FEIA.

2.2. Adaptive Countering Scheme (ACS). ACS [15] was proposed to alleviate the common problem of distributed solutions; that is, the en-route verification of sensing reports consumes extra energy resources during peacetime. The main objective of ACS is to detect FDIAs and FEIAs without en-route verification or the deployment of additional monitoring nodes. To achieve this, ACS exploits a crisp rule-based system that can detect FDIAs and FEIAs at a BS via two inference steps. Five factors are computed based on the reports collected by the BS and they are fed into the system for detection. The centralized detection of security attacks allows ACS to conserve the limited energy resources of WSNs because the reports are not verified by forwarding nodes during peacetime. Furthermore, ACS can provide an adaptive countering capacity against FDIAs and FEIAs. After an attack has been detected, the most effective countermeasure against the attack in terms of reliability (in the case of an FEIA) and energy efficiency during the delivery of reports is selected by a fuzzy rule-based system, which considers the status of the WSN and the attack. The adaptive selection of countermeasures can ensure reliability and obtain extra energy savings. However, to apply ACS to a WSN, the 10 or more security parameters used for detection need to be selected manually by users based on careful considerations of the status of the WSN and potential attacks. Unfortunately, it is not possible to guarantee that these parameters will be determined by experienced users in every WSN application. Thus, the inappropriate selection of parameters could severely increase the numbers of false positive errors and false negative errors during detection, thereby rendering ACS useless.

3. Fuzzy-Based Adaptive Method for Countering FEIAs

This section describes the proposed method in detail.

3.1. Assumptions. The proposed method can be treated as a fuzzy version of ACS and it inherits the basic assumptions of ACS [15]. The method considers a large-scale WSN for tracking moving objects, such as vehicles, in the sensor field. Each object moves continuously in the field. The network is sufficiently dense to be resilient against FDIAs [2]. The dense deployment of nodes means that each moving object can be detected by multiple nodes simultaneously or within a small gap of time. The nodes are similar to the current generation of sensor nodes (e.g., [17]) in terms of their computational and communication capacities and energy resources. Every node has a unique endorsement key, which is used to endorse reports, and it is shared only with BSs. In addition, the node has additional keys, which are used for endorsement and verification in PVFS and MEF. After detecting an object, the node uses its endorsement key to produce a sensing report for the object, which includes a MAC that is generated based on the contents of the report. The MAC is used to check the integrity of the report at a BS. The report is delivered to a BS after passing through multiple forwarding nodes. When PVFS or MEF is activated, the report can be verified by some of these nodes and it might be discarded. In contrast to ACS, it is assumed that a node may not know its geographical location.

Due to cost constraints, the nodes are not equipped with tamper-resistant hardware. Thus, a malicious adversary can physically compromise a few nodes. However, it would be difficult for the adversary to compromise large numbers of nodes and move them without being detected. The adversary cannot compromise BSs because they are attended closely by the users. Thus, the primary objective of the adversary is to interrupt the notifications of real objects to users. Therefore, the adversary will attempt to intentionally insert as many incorrect MACs as possible into legitimate reports using the compromised nodes. The adversary might also launch FDIAs with the aims of deceiving BSs and/or depleting the energy resources of the WSN.

3.2. Basic Detection and Countering Procedure. As shown in Figure 2, the proposed method is usually loaded on a BS. Additional storage (Figure 2(a)) is used to temporarily store reports collected by the BS. Each report is stored for a predefined period of time $T_R$. A fuzzy rule-based system (Figure 2(b)) uses the stored reports to detect FEIAs. Fuzzy inference to detect FEIAs is performed periodically for each of the objects mentioned in the reports. Thus, for each object, the inference process uses the average legitimacy of the reports (ARL) and the variance in ARL (VRL) to determine whether an FEIA has been launched on the object. Note that the determination of $T_R$ and the length of the period are application-dependent, and they are beyond the scope of this study.

After the fuzzy system has concluded that an FEIA has been launched on an object, the FEIA is reported to the users immediately. At the same time, another fuzzy system (Figure 2(c)) selects the most effective countermeasure against the FEIA in terms of reliability and energy efficiency
during delivery by using the stored reports. The fuzzy inference process for selecting countermeasures also uses two factors: the number of nodes that have reported the object (NNN) and the average distance to the object in hop count (ADO). In some cases, other FEIA countermeasures (e.g., [13, 14]) could be considered during fuzzy inference after some modifications of the fuzzy system. Finally, the countermeasure selected by the fuzzy system is activated by propagating a control message (Figure 2(d)), as found in ACS. The countermeasures can be deactivated by propagating another message if the threat has been eliminated.

### 3.3. Fuzzy-Based FEIA Detection and Countermeasure Selection

The proposed method exploits two fuzzy rule-based systems: one for detecting FEIAs and another for selecting countermeasures. The main reasons for using fuzzy systems are as follows.

(i) The membership function of fuzzy systems can be optimized automatically by combining a genetic algorithm and a simulation [16]. Thus, the need for experts (e.g., in ACS) to determine security parameters can be virtually eliminated. We cannot expect that appropriate parameters will always be selected by experienced users and inappropriate choices may severely degrade the performance (see Section 4.1).

(ii) The knowledge used for fuzzy inference can be stated in the form of if-then rules [18]. Thus, the basic rules required for detection and selection can be acquired intuitively from expert knowledge. In addition, these “rough” representations of knowledge might reduce the optimization time. If it is very difficult to acquire this knowledge from experts, the fuzzy systems could be replaced with artificial neural networks because of the equivalence between them [19].

(iii) Fuzzy systems can be used for approximate reasoning, which is particularly useful when the data are imprecise [20]. Note that the factors used during fuzzy inference will include some imprecision due to the malfunction of some nodes. For example, the ARL for a real object might not be 100% (all of the reports are legitimate) because of sensor or communication errors. Thus, we need to use approximate reasoning to handle this fuzzy information.

### 3.4. Factors for FEIA Detection

In the proposed method, FEIAs can be detected by a fuzzy rule-based system. To determine whether an FEIA has been launched on an object, the fuzzy system uses two factors: ARL and VRL.

ARL is the most important factor that determines whether an FEIA occurs or not. When a BS has received a report generated by a node, it first checks the legitimacy of the report. This check is performed by comparing the MAC attached to the report with another that is produced by it using the key shared with the node based on the contents of the report. The report is considered to be legitimate (in terms of integrity) if they match exactly. If not, the report is considered to be fabricated. If an object in the field is not affected by any FEIAs, each report related to the object will be legitimate (i.e., it has a correct MAC). There is a possibility that a few reports will have incorrect MACs due to node or wireless communication malfunctions, but most other reports will still be legitimate. However, if an FEIA has been launched on an object, a number of reports will carry incorrect MACs that have been inserted intentionally by the adversary. Thus, a low ARL for an object could suggest that the object is the target of an FEIA. When \( n \) reports for an object \( r_1, \ldots, r_n \) have been collected at a BS for \( T_R \), the ARL for the object is given by the average distance to the object in hop count, so that it should be equipped with GPS.

\[
ARL = \frac{\sum \text{distance}}{n}
\]

Another important factor during the detection of FEIAs is the VRL. An FEIA on a real object may result in a low ARL for an object, but a low ARL for an object could also be the signature of an FDIA. If the adversary uses the compromised nodes to launch an FDIA, most of the reports fabricated using the nodes will be legitimate. However, if an FDIA has been launched using external nodes (e.g., those of the adversary), most of the reports injected through the nodes will be illegitimate. The primary goal of the adversary might be to drain the limited energy resources of forwarding nodes or to disturb the communications between them, rather than to cause false alarms. In these situations, the ARL for the object will be low and it will change very slightly as time progresses; that is, VRL for the object will be close to 0. By contrast, an FEIA on an object could lead to a considerable change in the ARL due to the movement of the object. For example, when the object is affected by an FEIA, the ARL for the object will be close to 0. By contrast, if it is escaping from an FEIA, the ARL will increase constantly to almost 1.0, that is, a high VRL. Therefore, we need to consider the VRL during the detection of FEIAs in order to distinguish between FDIAs and FEIAs. The VRL for an object is computed simply by \( VRL = ARL_c - ARL_p \), where \( ARL_c \) is the current ARL for the object and \( ARL_p \) is the ARL for the object at the previous step. VRL is 0 if the object was not reported previously.

Other factors may also be related to the signatures of FEIAs. For example, to detect FDIAs and FEIAs, ACS uses the number of nodes that have reported the object, the variance in the number of the nodes, and the average velocity of the object. However, considering these factors during detection requires extra space for storing reports. This could be a nontrivial problem if a large number of objects need to be tracked. Also, it incurs further computation overheads during inference because there is an explosive increase in fuzzy if-then rules. For example, to make a fuzzy system having 25 rules consider two additional factors, each of which has 5 fuzzy variables; at most \( 25 \times 5 \times 5 = 625 \) rules should be given. Moreover, based on the former experiment results [15], these factors are much more useful for detecting FDIAs. Furthermore, to obtain some factors, nodes might require some special equipment. For example, to accurately measure the velocity of each object, every node should know its exact location, so that it should be equipped with GPS.
3.5. Factors for Countermeasure Selection. After the fuzzy system used for detection (described in the previous section) has concluded that an FEIA has been launched on a real object, another fuzzy rule-based system selects whether PVFS or MEF is launched against the FEIA based on an evaluation of their effectiveness, which considers the reliability and energy efficiency during the delivery of sensing reports. Fuzzy inference uses two factors, that is, NNN and ADO, because they have major effects on the reliability and energy efficiency.

Basically, PVFS and MEF were designed to counter FDIA and FEIAs. Thus, there is a probability of discarding reports (RDP) when they are activated, which is also the case with FDIA countermeasures. While a report is being forwarded by passing through intermediate nodes, each of the nodes has a chance to verify the report. The node can discard the report if verification fails once (in MEF) or several times (in PVFS). If an adversary has launched FDIA with the goal of depleting the energy resources of the network, en-route verification mechanisms could save considerable resources that are directed against FDIA. However, in the case of FEIAs, some reports that mention real objects may be filtered out by the forwarding nodes and the objects might not be reported correctly to the users. This problem is highly critical for most WSN applications (e.g., security and military applications). In MEF, multiple copies of a report for an object are forwarded via multiple routing paths, whereas a single report is generated and delivered to a BS in PVFS. Thus, the RDP of MEF is generally lower (better) than that of MEF. However, multipath routing means that MEF consumes more energy resources during forwarding. Therefore, PVFS is generally better than MEF in terms of energy savings.

NNN is one of the factors that affect the RDP. If an object has been reported from many nodes (the object may be large or moving quickly in the field), PVFS can provide a sufficient RDP since multiple reports of the object will be produced and forwarded. Thus, it would be better to choose PVFS in order to conserve energy resources. By contrast, MEF is a better solution if an object has been reported by a few nodes, where multiple copies are generated and forwarded via different paths for each report. For each object, the NNN for the object is the number of the nodes for $T_R$.

Another factor that affects the RDP is ADO. Due to the use of single-path routing, PVFS is more energy efficient than MEF with a long ADO. However, a long ADO also leads to a high RDP, where a report has a greater chance of being verified and discarded because it undergoes more hops. By contrast, some reports could be delivered to BSs for a long ADO with extra energy consumption in MEF. For the short-haul delivery of reports, it has been reported that MEF is sufficiently energy efficient to match the performance of PVFS despite the use of multipath routing [15]. For each object, the ADO for the object is the average number of hops that the reports complete.

The RDP can also be affected by other factors. For example, in MEF, the RDP can be enhanced by using more routing paths because more copies of a report will be produced for an object. However, this factor is usually static and the number of routing paths might not change during runtime. This "static" factor can be applied to the fuzzy system when the system is optimized (see Section 3.7): that is, the membership functions of the system will have been determined already by considering various static factors, including the number of routing paths. Other "dynamic" factors may be used in the selection of countermeasures. However, as stated in Section 3.4, the consideration of additional factors would demand extra space requirements and heavy computation overheads. Also, to obtain some factors, a node should be equipped with some special hardware, such as GPS.

3.6. Fuzzy Logic Design. The labels of the fuzzy variables that are used in the fuzzy system for the detection of FEIAs are as follows:

(i) $\text{ARL} = \{ \text{VL (Very Low)}, \text{L (Low)}, \text{M (Medium)}, \text{H (High)}, \text{VH (Very High)} \}$,

(ii) $\text{VRL} = \{ \text{DR (Decreasing Rapidly)}, \text{D (Decreasing)}, \text{S (Steady)}, \text{I (Increasing)}, \text{IR (Increasing Rapidly)} \}$,

(iii) $\text{ADR (Attack Detection Result)} = \{ \text{FDIA (Perhaps an FDIA)}, \text{RO (Real Object)}, \text{FEIA? (Perhaps an FEIA)}, \text{EFIA (Probably an FEIA)} \}$.

The fuzzy if-then rules for FEIA detection are shown in Table 1. Numbers on the left column of the table are used as identifiers for rules. The first rule (Rule number 1) can be read as, for an object, "if ARL is VL and VRL is DR, then ADR is FEIA." Each of the input variables (i.e., ARL and VRL) has 5

| Rule number | ARL | VRL | ADR  |
|-------------|-----|-----|------|
| 1           | VL  | DR  | FEIA |
| 2           | VL  | D   | FDIA |
| 3           | VL  | S   | FDIA |
| 4           | VL  | I   | FDIA |
| 5           |      |     |      |
| 6           | L   | DR  | FEIA |
| 7           | L   | D   | FDIA |
| 8           | L   | S   | FDIA |
| 9           | L   | I   | FDIA |
| 10          |     |     |      |
| 11          | M   | DR  | FEIA |
| 12          | M   | D   | FEIA |
| 13          | M   | S   | RO   |
| 14          | M   | I   | FEIA |
| 15          | M   | IR  | FEIA |
| 16          |     |     |      |
| 17          | H   | D   | RO   |
| 18          | H   | S   | RO   |
| 19          | H   | I   | FEIA |
| 20          | H   | IR  | FEIA |
| 21          |     |     |      |
| 22          | VH  | D   | RO   |
| 23          | VH  | S   | RO   |
| 24          | VH  | I   | RO   |
| 25          | VH  | IR  | FEIA |
labels, so that the system can have at most $5 \times 5 = 25$ rules. However, some rules are not defined in the system since no cases meet these conditions; for example, a rapid increase in ARL (i.e., VRL is IR) would never result in very low ARL, because of the result of the arithmetic mean (see Section 3.4).

For each object, the fuzzy system used for detection periodically determines whether the object is under an FEIA using ARL and VRL. Let us suppose that an adversary uses a few compromised nodes to launch an FEIA on a moving object. It would be very difficult for the adversary to move the compromised nodes without being detected. Thus, there will be a "stationary" region that is under the influence of the FEIA. Within this region, the adversary will attempt to insert as many incorrect MACs as possible into legitimate reports (during their generation or forwarding) in order to interrupt the object detection notifications. Thus, while the object is entering the region, the ARL and VRL for the object will be low and they will decrease rapidly, thereby allowing the fuzzy system to detect the FEIA. "VRL is DR" in the if-clauses of Rules numbers 1, 6, and 11 describes such a situation, so that the then-clauses of the rules conclude that the object is probably an FEIA (i.e., "ADR is FEIA"). By contrast, while it is exiting the region, the ARL and VRL will be high and they will increase rapidly, thereby allowing the FEIA to be detected. This situation (i.e., "VRL is IR") is described in Rules numbers 15, 20, and 25, which also conclude that the object is probably an FEIA. Suppose that the adversary has launched an FDIA. If the adversary uses external nodes for the FDIA, the ARL for a nonexistent object will be very low (as is the case for an FEIA). However, the VRL will change very slightly, thereby allowing the fuzzy system to warn the users of a possible FDIA occurrence. For example, the if-clause of Rule number 3 states such a situation; thus, the then-clause concludes that the object is perhaps nonexistent. Note that the proposed method is not designed for detecting FDIAs. Furthermore, the method cannot distinguish real objects and nonexistent ones fabricated using compromised nodes (i.e., FDIs using compromised nodes). Therefore, in order to correctly diagnose the occurrences of FDIAs, the users should employ other solutions for FDIA detection (e.g., [15]).

The system for the selection of countermeasures has the 25 fuzzy if-then rules shown in Table 2. The labels of the fuzzy variables are as follows:

(i) $\text{NNN} = \{\text{VS (Very Small)}, \text{S (Small)}, \text{M (Medium), L (Large), VL (Very Large)}\}$,
(ii) $\text{ADO} = \{\text{VN (Very Near)}, \text{N (Near), M (Medium), F (Far), VF (Very Far)}\}$,
(iii) $\text{CSR (Countermeasure Selection Result)} = \{\text{PVFS, PVFS? (PVFS might be better), MEF? (MEF might be better), MEF}\}$.

Table 2: Fuzzy if-then rules for countermeasure selection.

| Rule number | NNN | ADO | CSR |
|-------------|-----|-----|-----|
| 1           | VS  | VN  | PVFS|
| 2           | VS  | N   | MEF |
| 3           | VS  | M   | MEF |
| 4           | VS  | F   | MEF |
| 5           | VS  | VF  | MEF |
| 6           | S   | VN  | PVFS|
| 7           | S   | N   | PVFS|
| 8           | S   | M   | MEF |
| 9           | S   | F   | MEF |
| 10          | S   | VF  | MEF |
| 11          | M   | VN  | PVFS|
| 12          | M   | N   | PVFS|
| 13          | M   | M   | PVFS|
| 14          | M   | F   | MEF |
| 15          | M   | VF  | MEF |
| 16          | L   | VN  | PVFS|
| 17          | L   | N   | PVFS|
| 18          | L   | M   | PVFS|
| 19          | L   | F   | PVFS|
| 20          | L   | VF  | MEF |
| 21          | VL  | VN  | PVFS|
| 22          | VL  | N   | PVFS|
| 23          | VL  | M   | PVFS|
| 24          | VL  | F   | PVFS|
| 25          | VL  | VF  | PVFS|

After the fuzzy system used for detection has confirmed that an object is under an FEIA, PVFS or MEF is employed by the fuzzy system for countermeasure selection based on an evaluation of their effectiveness. The fuzzy inference process uses the NNN and ADO for the object. In the case where NNN is VS as stated in the if-clauses of Rules numbers 2, 3, 4, and 5, a small number of reports would be produced for the object (the object might be very small and moving very slowly in the field). In PVFS, many of these could be discarded en route due to the use of single-path routing. Thus, as stated in the then-clauses of the rules, MEF would be a better solution because reporting real events is the main role of WSNs. By contrast, if NNN is VL (a very large or rapidly moving object), several reports could be delivered to BSs, even in PVFS, and the object will be reported to the users. Thus, PVFS will be selected by the fuzzy system to obtain extra energy savings during delivery (see Rules numbers 21, 22, 23, 24, and 25). Note that PVFS could also provide some reliability enough to make a few reports delivered to BSs within a few hops. Thus, PVFS is recommended in Rule number 1 for energy savings. If an object is located very far from a BS (i.e., ADO is VF), every report for the object will have a greater chance of being dropped. This would be more critical for PVFS, where a single report is forwarded to a BS via a single routing path. Thus, MEF would perform better than PVFS in this case in terms of its reliability (see Rules numbers 5, 10, 15, and 20). By contrast, as stated in Rules numbers 1, 6, 11, 16, and 21, if the ADO is VN (objects located very close to BSs), PVFS would be a better solution because it would conserve the limited energy resources and provide sufficient reliability to report every real object to the users.
3.7. Optimization of Fuzzy Membership Functions. A major drawback of ACS is that users need to carefully select 10 or more security threshold parameters, which are used to determine the features of FDIAs and FEIAs based on considerations of the network settings (e.g., size, density, and routing paths) and the characteristics of objects (e.g., size and velocity). An inappropriate choice of a single parameter might lead to a severe decline in the detection capacity, as well as causing the misinterpretation of legitimate reports as fabricated reports (and vice versa). In the worst case, it might not be possible to detect security attacks at all. However, it is not possible to guarantee that these parameters will always be selected by some “omniscient” experts for every WSN application.

By contrast, a major benefit of the proposed method is that the two fuzzy systems, which are the essential components used for FEIA detection and countermeasure selection, can be optimized automatically by combining a simulation and a genetic algorithm [16]. To optimize the fuzzy systems for a WSN, users only have to build a simulation model of the WSN, which can be obtained using a GUI-based tool. In fact, only two factors need to be determined by the users of this method: the maximum values of NNN and ADO. It is considered that even inexperienced users could easily select appropriate values for these two factors.

In the optimization process, the parameter that determines the fuzzy membership functions is represented as a chromosome (Figure 3(a)). Initially, a set of chromosomes (a population) is generated randomly. The fitness of each chromosome is evaluated by a simulation (Figure 3(b)). Next, based on the simulation results, the population is evolved by three genetic operations: selection, crossover, and mutation (Figure 3(c)). The evaluation-evolution procedure is repeated until a condition is met (e.g., when the number of generation reaches 400). The two fuzzy systems can be optimized separately because they do not affect each other. During the optimization of the fuzzy system used for detection, the false positive error rates and false negative error rates are used to evaluate the fitness of a chromosome. During the optimization of the system used for selection, the reliability and energy efficiency during the delivery of sensing reports are used in the evaluation.

3.8. Countermeasure Activation and Deactivation. When a countermeasure has been selected for an FEIA, it can be activated by propagating a control message to initiate the countermeasure, as found in ACS. This propagation could be limited to small regions, for example, by grouping nodes or by limiting the propagation distance [21], or applied to the entire network. BSs possess a mechanism (e.g., based on μTESLA [22]) to authenticate control messages, which allows the nodes to verify the message. The elimination of the threat may involve some physical actions (e.g., dispatching a patrol unit) because the nodes used to launch the FEIA may be physically captured by the adversary. After eliminating the threat, the countermeasure can be deactivated by propagating another control message.

4. Simulation Results

To demonstrate the performance of the proposed method, it was compared with ACS based on simulations. The size of the sensor field was 1,000 \( \times \) 100 m\(^2\) and a single BS was located in a corner of the field. In total, 4,500 nodes, which could detect objects and communicate with neighboring nodes located within 10 m, were deployed uniformly in the field. A malicious adversary could physically capture 1-15 nodes and then use them to launch FEIAs. The adversary could also launch FDIAs using external nodes. Each object appeared/disappeared on the border of the field and passed random waypoints on the field. The objects moved in the field at a speed of \( U(0.5,1.0) \) m/s. On average, an object could be detected by 15 nodes simultaneously, or within a small period of time. During the detection of FEIAs, the false positive error rate (FPER) and false negative error rate (FNER) were measured for the proposed method and ACS with two different settings. During countermeasure selection, the proportion of legitimate reports delivered to the BS and the average energy consumption during delivery were measured using the proposed method, ACS, PVFS, and MEF.

Figure 4 shows the membership functions of the fuzzy systems for the WSN. Each of the fuzzy input variables has 5 triangular membership functions; for example, (e) ADO has VN—triangle (x; 0, 0, 42), N—triangle (x; 0, 42, 60), M—triangle (x; 42, 60, 84), F—triangle (x; 60, 84, 120), and VF—triangle (x; 84, 120, 120). These functions were optimized for the WSN by combining a genetic algorithm and a simulation, with the consideration of the characteristics of the given network; the genetic algorithm might choose them since they were, based on the results of simulations under the fuzzy if-then rules (see Section 3.6), fittest for the network, among a number of functions in the population. For example, the membership functions for ADO might be chosen since the following hold.

(i) For the given network, PVFS was usually effective in the delivery within 42 hops; as stated in Section 3.5, PVFS would be effective for the short-haul delivery and thus is chosen in the fuzzy if-then rules in most short-haul cases.

(ii) Usually, MEF was a better solution if a report needs to be delivered through 84 or more number of hops.
(iii) In case of the 60-hop delivery, the effectiveness of PFVS was similar to that of MEF.

The membership functions for the fuzzy output variables ((c) ADR and (f) CSR) are fixed (i.e., target-independent) and thus not determined by the genetic algorithm. The final decision for an object is made based on the quantifiable value obtained through the defuzzification process and vertical dotted lines (g), (h), and (i). For example, in the detection of FEIAs, if the quantifiable value is greater than (h) (i.e., >2.0), the fuzzy system finally concludes that the object is under an FEIA.

4.1. Detection Performance. Figure 5 shows the FPERs using ACS (filled diamonds and circles) and the proposed method (empty rectangles) when $T_R$ was 40, 80, 120, 160, and 200 s. Filled diamonds show the FPER of ACS optimized by the inventors of ACS, while filled circles show that of ACS configured by a common user. A false positive error was an error where a real object was misinterpreted as an FEIA. As shown in the figure, the FPER decreased as $T_R$ increased because the data collected at the BS were used to determine the FEIAs. Thus, the detection accuracy increased as more reports were collected. However, a large $T_R$ increased the time required to detect FEIAs and it required extra space. The approximate reasoning employed by the fuzzy system used for detection reduced the FPER with the proposed method compared with that with ACS. In contrast to computer networks, it is important to reduce the FPER because security attacks in WSNs may involve real-world responses. As shown in the figure, the performance of the proposed method (empty rectangles) was similar to that of ACS optimized by the inventors (filled diamonds), while the proposed method was not configured by a highly experienced user. Also, the proposed method was particularly useful with a small $T_R$ due to approximate reasoning. For example, for $T_R = 40$, about 63/1,000 real events were misinterpreted as FEIAs with ACS optimized by the inventors, whereas about 57 were misinterpreted using the proposed method. Note that about 169 were misinterpreted with ACS configured by a common user (filled circles). Therefore, the proposed method is superior to ACS, especially when ACS cannot be configured by experienced users, or if the storage space is restricted (e.g., low-end BSs), or if applications require the rapid detection of FEIAs.

Figure 6 shows the FNERs with ACS optimized by the inventors (filled diamonds), ACS configured by a user (filled circles), and the proposed method (empty rectangles) when $T_R$ was 40, 80, 120, 160, and 200 s. A false negative error was an error where an FEIA was not detected (i.e., misinterpreted as a real object). As shown in the figure, a large $T_R$ could reduce the FNER because the two input factors are computed based on the reports collected by the BS. However, a large $T_R$ incurred a penalty in terms of the detection time and space complexity. As shown in the figure, the proposed method could reduce the FNER as low as ACS optimized by the inventors could do, while it can also eliminate the need for experts to determine security parameters. Also, the FNER...
obtained using the proposed method could be enhanced slightly due to approximate reasoning, especially with a small $T_R$. For example, for $T_R = 40$, about 87/1,000 FEIAs were misinterpreted as real objects with ACS optimized by the inventors. In case of ACS configured by a user, about 256 were undetected. By contrast, about 83 were missed using the proposed method. In addition, the proposed method uses two factors for detection, whereas ACS requires five. Thus, the proposed method can reduce the space required for report storage, which would be particularly useful for (relatively) low-end BSs.

4.2. Reliability and Energy Efficiency. Figures 7 and 8 show the proportions of legitimate reports delivered to the BS (i.e., reliability during the delivery of sensing reports) and the average energy consumption per report delivery, respectively. Case 1 represents the results with a high-density WSN and case 2 represents the results with a medium-density WSN, where an object could be detected by nine nodes on average. The energy consumption levels were computed based on the model in [2]. As shown in the figure, MEF was the best solution in terms of reliability due to the use of tripartite path routing. However, it consumed a large amount of the limited energy resources of the WSN, which would shorten the lifespan of the WSN. By contrast, PVFS performed the best in terms of energy efficiency because it uses single path routing and it minimizes en-route verification. However, many legitimate reports could not be delivered to the BS, which meant that some real objects were not reported to the users. This would be critical in most WSN applications. Both the proposed method and ACS performed reasonably well in terms of reliability, where every real object was reported to the users. The energy efficiency of ACS was slightly better than that of the proposed method because ACS considers an additional FEIA countermeasure [23]. However, this countermeasure involves a more complex process during the generation of a report that requires disjoint dual path routing, which is not easy to implement. In addition, ACS uses three factors during selection, whereas two factors are used in the proposed method. Thus, it is expected that the proposed method could be applied to a wider range of WSNs. Therefore, the proposed method may be less dependent on node devices and BS hardware compared with ACS.

5. Conclusions and Future Work

This study proposed a fuzzy-based method to adaptively counter FEIAs in dense WSNs for object-tracking applications. The proposed method employs two fuzzy rule-based systems: one for detecting FEIAs and another for selecting the FEIA countermeasures. For each object, the former system determines whether an FEIA has been launched based on the data collected at a BS. After detecting an FEIA, the latter system evaluates the effectiveness of PVFS and MEF against the FEIA based on the data. A major benefit of the
proposed method is that the fuzzy systems can be optimized automatically for WSNs. Furthermore, compared with ACS, the detection accuracy is improved by the approximate reasoning based on fuzzy logic. The proposed method also uses fewer factors than ACS, which can reduce the memory requirements and computational complexity. The superior performance of the proposed method was demonstrated based on simulation results.

The following topics will be investigated in future research.

(i) Detection of FEIAs based on stationary events: events can be stationary in some WSN applications, for example, in WSNs used for fire detection. The proposed method would not perform well in the detection of FEIAs based on stationary events because it uses VRL, which is based mainly on the movement of an object.

(ii) Improving the detection accuracy, especially false negative errors, by considering additional or other input factors, or entirely different approaches: although the consideration of extra or other factors may demand additional equipment on nodes (e.g., GPS) and/or space requirements on BSs, the accuracy itself could be enhanced.

(iii) Enhancing the reliability and energy efficiency during delivery by considering additional or other input factors or countermeasures: extra or other factors can be considered with additional assumptions (e.g., for hardware requirements), which could improve the performance.

(iv) Providing some security proofs of the proposed method and/or FEIA countermeasures using formal verification techniques, such as model checking [24].

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by a research grant from Seoul Women's University (2013).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” IEEE Communications Magazine, vol. 40, no. 8, pp. 102–114, 2002.
[2] F. Ye, H. Luo, S. Lu, and L. Zhang, “Statistical en-route filtering of injected false data in sensor networks,” IEEE Journal on Selected Areas in Communications, vol. 23, no. 4, pp. 839–850, 2005.
[3] H. Yu, J. He, R. Liu, and D. Ji, “On the security of data collection and transmission from wireless sensor networks in the context of Internet of things,” International Journal of Distributed Sensor Networks, vol. 2013, Article ID 806505, 13 pages, 2013.
[4] H. Yang and S. Lu, “Commutative cipher based en-route filtering in wireless sensor networks,” in Proceedings of the 60th IEEE Vehicular Technology Conference (VTC ’04), vol. 2, pp. 1223–1227, 2004.
[5] S. Zhu, S. Setia, S. Jajodia, and P. Ning, “Interleaved hop-by-hop authentication against false data injection attacks in sensor networks,” ACM Transactions on Sensor Networks, vol. 3, no. 3, Article ID 1267062, 2007.
[6] F. Li, A. Srinivasan, and J. Wu, “PVFS: a probabilistic voting-based propagation limiting method for wireless sensor networks,” International Journal of Security and Networks, vol. 3, no. 3, pp. 173–182, 2008.
[7] T. P. Nghiem and T. H. Cho, “A fuzzy-based interleaved multihop authentication scheme in wireless sensor networks,” Journal of Parallel and Distributed Computing, vol. 69, no. 5, pp. 441–459, 2009.
[8] Z. Yu and Y. Guan, “A dynamic en-route filtering scheme for data reporting in wireless sensor networks,” IEEE/ACM Transactions on Networking, vol. 18, no. 1, pp. 150–163, 2010.
[9] H. Y. Lee, T. H. Cho, and H.-J. Kim, “Fuzzy-based detection of injected false data in wireless sensor networks,” Communication in Computer and Information Science, vol. 76, pp. 128–137, 2010.
[10] Z. Liu, J. Wang, and X. Zhang, “A false data filtering scheme using cluster-based organization in sensor networks,” in Proceedings of the IEEE International Conference on Communications (ICC ’11), pp. 1–5, 2011.
[11] J. Wang, Z. Liu, S. Zhang, and X. Zhang, “Defending collaborative false data injection attacks in wireless sensor networks,” Information Sciences, vol. 254, pp. 39–53, 2014.
[12] M. S. Kim and T. H. Cho, “A multipath en-route filtering method for dropping reports in sensor networks,” IEICE Transactions on Information and Systems, vol. E90-D, no. 12, pp. 2108–2109, 2007.
[13] C. Krauß, M. Schneider, and C. Eckert, “Defending against false-endorsement-based DoS attacks in wireless sensor networks,” in Proceedings of the 1st ACM Conference on Wireless Network Security (WiSec ’08), pp. 13–23, April 2008.
[14] C.-M. Yu, Y.-T. Tsou, C.-S. Lu, and S.-Y. Kuo, “Constrained function-based message authentication for sensor networks,” IEEE Transactions on Information Forensics and Security, vol. 6, no. 2, pp. 407–425, 2011.
[15] H. Y. Lee and T. H. Cho, “A scheme for adaptively counteracting application layer security attacks in wireless sensor networks,” IEICE Transactions on Communications, vol. 93, no. 7, pp. 1881–1889, 2010.
[16] H. Y. Lee and T. H. Cho, “Optimized fuzzy adaptive filtering for ubiquitous sensor networks,” IEICE Transactions on Communication, vol. E94-B, no. 6, pp. 1648–1656, 2011.
[17] Advanticsys, http://www.advanticsys.com/.
[18] M. Takács, "Approximate reasoning in fuzzy systems based on pseudo-analysis and uninorm residuum," Acta Polytechnica Hungarica, vol. 1, no. 2, pp. 49–62, 2004.
[19] H. X. Li and C. L. P. Chen, “The equivalence between fuzzy logic systems and feedback neural networks,” IEEE Transactions on Neural Networks, vol. 11, no. 2, pp. 356–365, 2000.
[20] N. Serrano and H. Seraji, “Landing site selection using fuzzy rule-based reasoning,” in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA ’07), pp. 4899–4904, Roma, Italy, April 2007.
[21] S. H. Chi and T. H. Cho, “Fuzzy logic based propagation limiting method for message routing in wireless sensor networks,” in
[22] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," Wireless Networks, vol. 8, no. 5, pp. 521–534, 2002.

[23] H. Y. Lee and T. H. Cho, "False negative-resilient report generation for the statistical filtering in sensor networks," in Proceedings of the International Conference on Network and Mobile Computing (NMC '06), p. 27, 2006.

[24] C. Baier and J. P. Katoen, Principles of Model Checking, MIT Press, 2008.
