Discrete event simulation–based energy efficient path determination scheme for probabilistic voting–based filtering scheme in sensor networks

Su Man Nam¹ and Tae Ho Cho²

Abstract
In wireless sensor networks, sensors are extremely vulnerable to false positive and false negative attacks due to their stringent energy and computational constraints. Several en-route filtering schemes mainly focus on saving energy through early detection of false data within a short distance against these attacks; however, they cannot immediately block the false data injected by compromised nodes. A security scheme uses context-aware architecture for a probabilistic voting–based filtering scheme to detect the compromised nodes and block the injection of false data, unlike security protocols. Although these schemes effectively obstruct the false data forwarding, they cannot make any detour around the compromised node to avoid it during data forwarding. In this article, we propose a discrete event simulation–based energy efficient path determination scheme that takes a detour around the compromised node against the attacks. Our proposed scheme extracts candidate paths considering the network status and selects a path with the highest energy efficiency from among the candidates using discrete event simulation. Simulation results indicate that the proposed scheme provides energy savings of up to 12% while maintaining the security strength against the two attacks compared to the existing schemes.

Keywords
Wireless sensor networks, probabilistic voting–based filtering, discrete event simulation, network security, energy efficiency

Date received: 11 March 2020; accepted: 4 July 2020

Handling Editor: Kim-Kwang R Choo

Introduction
Wireless sensor networks (WSNs) are an economically feasible solution for regular monitoring of events in the environment and habitats, in microclimate surveillance, and in tracking for military applications¹–⁶. A WSN consists of a large number of sensors and a base station (BS) in a sensor field.²⁻⁷,⁸ The sensor nodes sense the events, generate the event data, and wirelessly forward the data between the nodes toward the BS. The BS collects the data from the sensor, analyzes it, and provides analyzed results to the users. Because sensors are randomly distributed in an open, collaborative, and large-scale environment and are left unattended for long periods of time, adversaries can easily compromise nodes and waste network resources or block data delivery.⁴,⁸,⁹⁻¹¹ When a node is compromised physically or

¹DUDU Information Technologies, Seoul, Republic of Korea
²Department of Computer Science and Engineering, Sungkyunkwan University, Suwon, Republic of Korea

Corresponding author:
Tae Ho Cho, Department of Computer Science and Engineering, Sungkyunkwan University, Suwon, Republic of Korea.
Email: thcho@skku.edu
through fabricated messages, adversaries can achieve full control over it by reading its memory and influencing the operation of its software (SW).\textsuperscript{12,13} Therefore, sensor nodes are considered vulnerable to being captured or damaged because of their relatively limited batteries, restricted computation, and unattended operation.\textsuperscript{6,14}

In WSNs, there are several attacks that can happen through compromised nodes. Two types of attacks, false positive attacks (FPAs)\textsuperscript{5,6,10,12,14–18} and false negative attacks (FNAs)\textsuperscript{4,6,10,12,19} are a major concern. In an FPA, the compromised node can inject false reports to cause unnecessary energy consumption and false alarms for normal nodes. In an FNA, the other compromised node can insert a false message authentication code (MAC) into a legitimate report to lose information.

Even though researchers have proposed several schemes\textsuperscript{5,6,10,12,14,19} to detect FPAs and/or FNAs in the sensor network, these schemes counter the attacks through probabilistic verification, in particular through intermediate nodes while forwarding reports. Nam et al., however, proposed a context-aware architecture (CAA)\textsuperscript{4} for effectively identifying the compromised nodes against the two attacks. The architecture is integrated with a probabilistic voting–based filtering scheme (PVFS)\textsuperscript{5} to overcome the weaknesses of the security protocol. However, in CAA for PVFS (CAA-PVFS), after a compromised node, which is located in the middle of a path, is identified, a secure transmission is not guaranteed when the compromised node receives a legitimate report from the previous node. Thus, it is important to find a new path for an effective detour around the compromised node, unlike several path selection schemes\textsuperscript{8,20} that only consider the condition of the next forwarding nodes.

In this article, we propose a scheme that determines energy efficient paths based on a discrete event simulation in our CAA for the sensor network. After identifying compromised nodes as in the previous CAA,\textsuperscript{4} our proposed scheme extracts paths, analyzes the paths through the simulation, and selects an effective path from among them. In the proposed scheme, the simulation model of our CAA performs state transitions to identify normal or abnormal behaviors. To decide the behaviors, when the CAA receives context data from the network, it performs state transitions through its own information. To determine which of the paths are suitable for selection, the proposed scheme creates actual normal and attack data and analyzes the performance of the path using the actual data. The scheme determines the path with the highest energy efficiency according to the analysis results of energy consumption and MAC verification using the injected actual data within the models of the CAA. Simulation results illustrate that the proposed scheme provides average energy savings of approximately 10% with the energy efficient path determination, which results in a network lifetime extension.

The main contributions of this article are as follows:

- Determination of an effective path by simulation, and
- Improvement in energy efficiency.

The rest of this article is organized as follows: section “Background and motivation” presents the background and motivation for the proposed scheme. A detailed description of the proposed scheme follows in section “Proposed scheme.” In section “Simulation results,” we present a performance evaluation of the proposed scheme using analysis and simulation. Section “Related work” discusses security protocols of FPAs and FNAs and path configuration protocols with security related to our proposal. The conclusions are given at the end of the article.

**Background and motivation**

FPAs and FNAs pose serious security threats to all sensor networks. FPAs cause energy drain through the transmission of false reports, and FNAs cause information loss through the filtering of legitimate reports. In this section, we briefly describe the operation of PVFS and CAA-PVFS to deal with these two attacks. We also discuss the motivation of the proposed scheme.

**PVFS**

PVFS\textsuperscript{5,6,21} simultaneously detects both FPAs and FNAs through verification records of reports using a voting method, a cluster-based organization, and a probabilistic key assignment. PVFS contains three phases: initialization and key assignment, report generation, and en-route filtering.

In the initialization and key assignment phase, the BS pre-generates a global key pool ($n$ partitions $\times m$ keys), and randomly distributes keys from the key pool to each node. After the nodes are deployed, they are organized into clusters, each of which consists of a CH and $L$ member nodes (MBs) within one hop. Intermediate CHs are probabilistically selected as verification nodes for report authentication based on their distance from a source CH.

In the report generation phase, when an event occurs, the CH broadcasts the event data to its MBs. After accepting the event data, the MBs generate each MAC and forward it to its CH. The CH collects all the MACs and attaches the randomly selected MACs to the report. It then forwards the report to its next forwarding node toward the BS.
In the en-route filtering phase, while forwarding the report, each verification node verifies the MACs in the report. According to the result of the verification records, the verification node checks that the reports reach a special threshold by the recorded result. When the recorded result at the node reaches the threshold, the verification node immediately filters out the report against the FPA; while the result has not yet reached the threshold, the node continually forwards it against the FNA. When the report arrives at the BS, the BS verifies all the MACs through the keys of the key pool.

CAA-PVFS

CAA-PVFS\cite{4,15} is proposed for effectively identifying compromised nodes using its CAA, which coordinates with the PVFS in a sensor network. The CAA is implemented through simulation models and the algorithm of PVFS. The architecture analyzes the data collected from the sensor network based on its security knowledge base (KB) and identifies the compromised nodes using the analyzed data against FPAs and FNAs. The sensor network excludes the identified compromised nodes to block false data.

CAA-PVFS comprised three architectures: PVFS-based WSN, Communication Architecture (Comm-Arch),\cite{4,15} and a CAA. PVFS works as described in section “PVFS.” Comm-Arch consists of a number of data collection nodes (DCNs) to collect sensing data from the nodes of the sensor network and forward it to the CAA. The DCNs are organized into each cluster with a CH and MBs. The CAA of this scheme is implemented through simulation models and the PVFS algorithm to find the compromised nodes of the sensor network.

When a sensor node broadcasts sensing data (e.g. event data, MACs, and reports) as shown in Figure 1(a), a DCN collects the data using a frequency of the sensor. In Figure 1(b), the DCN exploits the other frequency to protect the collision frequency and transmits the sensing data through data encryption and compression to the CAA of the scheme. In Figure 1(c), the CAA detects intrusions using data collected from the sensor network based on its security knowledge (e.g. knowledge of normal behavior within the network and knowledge of the attacks). In the CAA, the number of atomic models (ACH, AMB, and CAA_CNTR) equals the number of sensor nodes (CH, MB, and BS). The atomic models are as follows:

- **ACH** (context-aware-based CH): This model corresponds to a CH of the sensor network. As this model receives a report collected through the Comm-Arch, it analyzes the report based on its KB. If the model’s behavior is abnormal, it detects the false report against an FPA and provides its controller model for a detection result.
- **AMB** (context-aware-based MB): This model corresponds to an MB. The model analyzes a MAC based on its KB. If its behavior is abnormal, the model detects the fabricated MAC against an FNA and notifies its controller of the result.
- **CAA_CNTR** (CAA controller): This model corresponds to the BS. When the model receives the detection result of the compromised node’s location information from the ACH and AMB, it synthetically analyzes the result and authentically identifies the compromised node. The model then sends a final decision to a BS.

![Figure 1. An overview of CAA-PVFS: (a) PVFS-based WSN; (b) Comm-Arch; and (c) CAA.](image)
The CAA transmits the context-aware decision including the compromised node’s location to the BS and excludes the compromised node to block false data injection. Therefore, CAA-PVFS integrates the CAA and PVFS to effectively identify a compromised node and improves security strength and energy efficiency.

**Discrete event system**

The discrete event system (DEVS) specification formalism developed by Zeigler specifies a model in a hierarchical and modular form for DEVS modeling based on set theory. In addition, DEVS provides the framework for information modeling which has the advantage of model reusability, verifiability, expansibility, and availability to analyze and design complex systems. DEVS has two kinds of models to represent the systems: atomic models and coupled models. In the two models, the input and output event sets are associated with the input and output ports. The atomic model executes an input event on its basis state; it generates an output event and transfers its state. The model has states, which are related to the behavior to be executed during a specific time. The atomic model is defined by the following structure

\[ M = \langle X, S, Y, d_{int}, d_{ext}, \lambda, t_a \rangle \]  

where \( X \) is an external input set, \( S \) is a sequential state set, \( Y \) is an external output set, \( d_{int} \) is an internal transition function, \( d_{ext} \) is an external transition function, \( \lambda \) is an output function, and \( t_a \) is a time advance function.

The coupled model is constructed by coupling other models and has the hierarchical model structure to build a complex model. In addition, the coupled model can itself be used as a component in a larger coupled model. As this model receives input or output events, it is converted into input or output events of the other models through the coupling. A coupled model is defined by the following structure

\[ DN = \langle D, \{M_i\}, \{I_i\}, \{Z_{i,j}\}, \text{select} \rangle \]

where \( D \) is a set of component names, \( M_i \) is a component of the basic model, \( I_i \) is a set of influences of \( I \), \( Z_{i,j} \) is an output translation, and \( \text{select} \) is a tie-breaking function.

**Motivation**

In WSNs, sensor nodes are vulnerable to FPAs and FNAs due to stringent energy and computational constraints. These attacks cause unnecessary energy consumption and normal data removal at intermediate nodes. The FPA in particular causes energy drain at multiple intermediate nodes and rapidly shortens the network life. To detect these attacks, PVFS uses an energy-efficient route filtering technique through a voting method; CAA-PVFS identifies the compromised nodes using the context-awareness collected from the sensor network. Although compromised nodes are identified, the compromised nodes try to inject other attacks such as data capture and manipulation as they receive reports. Thus, it is important that intermediate nodes transmit the report while bypassing the compromised node along an efficient route.

We propose the determination of a new path considering the network situation by means of the DEVS simulation after our CAA detects a compromised node. To select a new path effectively, our proposed scheme extracts candidate paths and chooses a path from among them using the analysis result of the DEVS simulation. Therefore, the proposed scheme securely transmits reports using the detour around the compromised node along the selected path.

**Proposed scheme**

In this section, the proposed scheme is discussed in detail. The proposed scheme extracts candidate paths after our CAA identifies compromised nodes and selects an effective path from among them. The system model and assumptions are outlined in section “System model and assumptions,” an overview is presented in section “Overview,” the detailed procedure is presented in section “Detailed procedure,” and an example of the proposed scheme is given in section “Example.”

**System model and assumptions**

A sensor network comprised a BS and sensor nodes, which are powerful nodes (e.g. high-end sensors) to serve as CHs and normal nodes (e.g. low-end sensors) to serve as MBs. The BS is a reliable and powerful sensor with high hardware (HW) performance (e.g. processing speed and great storage capabilities). Each CH organizes its cluster with the MBs using the cluster-based model after deployment. The CH’s sensing range is much larger than its transmission range, and it uses a larger transmission range than does an MB. The CH discovers a multi-hop path to the BS. To forward data, all nodes transmit the data along the path toward the BS. The CH has more powerful resources (e.g. memory size, battery power, computation ability, transmission range) than an MB. We assume that the sensor nodes do not move after initial deployment in the sensor field. In addition, a compromised node probabilistically injects bogus data into the network. Since our work focuses on FPAs and FNAs, other attacks launched by compromised nodes in the sensor network, such as selective forwarding, sinkhole, and wormhole, are beyond the scope of this article.
In a Comm-Arch,4,15 DCNs gather all the data of the sensor network and transmit it to the proposed CAA. A DCN exploits two types of frequencies for collecting and forwarding the data and uses HW models instead of SW models to protect it from various types of intrusions. In addition, this node applies an advanced encryption standard (AES) HW module29–31 for data confidentiality and integrity, and a Huffman coding HW model32–35 for data compression. The transmission range of each DCN is the same as that of a CH. Thus, the DCNs are designed using HW modules with SW, and they are inexpensive because their modules are simple.4

Overview

Our proposed scheme determines an energy efficient path through a DEVS simulation to bypass identified compromised nodes in our CAA. In the proposed scheme, the CAA is implemented using the same hierarchical structure of the DEVS models as proposed in Nam and Cho4; it redesigns the atomic DEVS models’ behaviors for our proposal. An ACH, an AMB, and a CAA_CNTR correspond to the atomic DEVS models of a CH, an MB, and a BS, respectively. These atomic models of the proposed CAA are as follows:

- ACH: This model detects false reports injected by compromised CHs based on its security KB against FPAs and identifies the node’s location. After identifying the compromised node, the model analyzes and forwards normal and attack simulation data along the extracted candidate paths. The model then provides analysis results in the CAA_CNTR.
- AMB: This model detects false MACs through its KB against FNAs and finds compromised MBs.
- CAA_CNTR: This model comprehensively evaluates the analytical results of the ACH and AMB and accurately identifies the compromised node. The model then extracts candidate paths, selects an effective path from among them, and notifies the BS of the path.

Figure 2 shows an overview of our proposed scheme. The PVFS-based WSN works as described in section “PVFS.” As sensor nodes forward sensing data, the sensing result of the data is transmitted via a Comm-Arch to the CAA. In the CAA, ACH and AMB models are implemented using the DEVS formalism. An ACH model detects a compromised node based on its KB using the collected data of the sensor, and CAA_CNTR extracts candidate paths for a detour around the compromised node based on a minimum cost forwarding path algorithm. The ACH injects actual normal and attack data into the path and analyzes the paths through the simulation. The CAA_CNTR selects the most effective path with respect to energy efficiency and notifies the BS of the selected path. The BS compares the selected path and the existing path and changes the specific path. Therefore, the proposed scheme determines the energy efficient path using the DEVS simulation to bypass the identified compromised node.

Detailed procedure

Our proposed scheme adds the following phases based on the phases (initialization and key assignment, report

![Figure 2. An overview of the proposed scheme.](image-url)
generation, en-route filtering, data verification) proposed in Nam and Cho4:

- Extraction and analysis of paths: In the CAA, the CAA_CNTR extracts candidate paths after finding a compromised node and analyzes paths using simulation.
- Selection and configuration of a path: The CAA_CNTR selects the most effective path with respect to energy efficiency using analyzed results. This model notifies the BS of the selected path, and the BS forwards the path configuration message to specific sensor nodes.

**Extraction and analysis of paths.** The proposed scheme extracts three types of candidate paths based on the shortest path algorithm to make a detour around the compromised node after identifying the node.

**Algorithm 1. CreateRoutingPaths (ACH\text{comp}).**

1. $ACH\text{prev}$ \text{←} previous node of $ACH\text{comp}$;
2. for $i = 1$ to $3$ do
3. $routingPath_i$ \text{←} ExtractCandidatePaths($type_i$, $ACH\text{prev}$, $ACH\text{comp}$);
4. $routingPath$ \text{←} getExistingPaths($ACH\text{prev}$, $ACH\text{comp}$);

**Algorithm 2. ExtractCandidatePaths(type, src, comp).**

1. $M\text{src}$ \text{←} src;
2. while true do
3. $M\text{neigs}$ \text{←} FindNeighborNodes($M\text{src}$);
4. $M\text{next}$ \text{←} GetShortestDistance($type$, $M\text{neigs}$, $M\text{comp}$);
5. if $M\text{next}$ \neq 0 then
6. path.insert($M\text{next}$);
7. else
8. break;
9. end if
10. $M\text{src}$ \text{←} $M\text{next}$;
11. end while
12. return path;

$ACH\text{prev}$ finds one of its neighbors without the identified compromised model, and $M\text{next}$ chooses a next forwarding model according to the type. If $M\text{next}$ is the CAA_CNTR, $path$ is returned as a candidate.

Figure 3 shows the performance measurements of the extracted candidate paths based on the DEVS simulation. A source $ACH$ uniformly injects reports including MACs with normal and attack data into two candidate paths and the existing path. The intermediate models of each path receive two types of normal and attack data. Each intermediate model calculates and records the energy consumed for transmission, reception, and verification while forwarding the injected reports toward the CAA_CNTR. After receiving all the reports, the CAA_CNTR selects an effective path based on the amount of energy consumed.

**Selection and configuration of a path.** In order to determine an energy efficient path, the proposed scheme measures the performances of candidate paths through simulation. In the DEVS simulation, it is possible to execute the behavior of models using defined rules with elapsed time according to procedures similar to those of a target system. The proposed scheme obtains comprehensive and accurate performances with actual normal and attack data using an iterative simulation such as a real system. After finishing the simulation, the proposed
scheme selects the most effective path from among the candidates using the whole sensor network status.

Algorithm 3 demonstrates the selection of one new path using the performance measurement of the candidates after the CAA_CNTR completely receives two types of data. The CAA_CNTR model obtains the total average consumed energy from records in the data and selects a path for the highest energy efficiency from among the candidate paths. The controller model notifies the BS of the new path, compares the new path and the existing path, and forwards path configuration messages to specific nodes, which should then modify their paths.

**Algorithm 3. SelectNewRoutingPath(routingPath).**

1: min ← maxValue;
2: for each path in routingPath do
3: \[Eng_{sum}\] ← path. GetTotalAverageEnergy();
4: if \[Eng_{sum}\] < min then
5: min ← \[Eng_{sum}\];
6: selectedPath ← path;
7: end if
8: end for
9: return selectedPath;

Figure 4 shows the path determination procedure in the CAA after identifying a compromised node in model ACH₅. The CAA sets up an initial path between the models and the verification model as verification nodes such as in the sensor network. We assume that the existing paths of ACH₅ are ACH₆, ACH₉, ACH₁₀, and ACH₁₁ and its verification models are ACH₆, ACH₉, and ACH₁₀. In addition, ACH₆ blocks the injection of data by ACH₅ since ACH₅ is identified as a compromised node. As shown in Figure 4, ACH₂, which is ACH₅’s previous model, extracts three types of candidate paths to bypass the compromised node. To extract the shortest path, ACH₂ selects ACH₄ and excludes ACH₅. ACH₄ chooses ACH₈, which is shorter than ACH₆, since ACH₈ is identified as a compromised node. The controller model notifies the BS of the new path, compares the new path and the existing path, and forwards path configuration messages to specific nodes, which should then modify their paths.

**Example**

Figure 4 shows the path determination procedure in the CAA after identifying a compromised node in model ACH₅. The CAA sets up an initial path between the models and the verification model as verification nodes such as in the sensor network. We assume that the existing paths of ACH₅ are ACH₆, ACH₉, ACH₁₀, and ACH₁₁ and its verification models are ACH₆, ACH₉, and ACH₁₀. In addition, ACH₆ blocks the injection of data by ACH₅ since ACH₅ is identified as a compromised node. As shown in Figure 4, ACH₂, which is ACH₅’s previous model, extracts three types of candidate paths to bypass the compromised node. To extract the shortest path, ACH₂ selects ACH₄ and excludes ACH₅. ACH₄ chooses ACH₈, which is shorter than ACH₆, since ACH₈ is identified as a compromised node. The controller model notifies the BS of the new path, compares the new path and the existing path, and forwards path configuration messages to specific nodes, which should then modify their paths.
Candidate path₁ (the shortest path): The source model, ACH₂, forwards normal and attack reports to ACH₄ in a regular sequence. ACH₄ writes the amount of energy consumption for transmission and reception in the data analysis as it receives the report. When ACH₁₀ receives the report, this model records the calculated result of the path analysis data if it verifies a MAC. As the four types of reports arrive at the CAA_CNTR, the controller model calculates and saves the sum of the results of the analysis data. The model waits for all the reports to arrive, after which they are forwarded along the other paths.

Candidate path₂ (the second shortest path): ACH₂ forwards the generated reports to ACH₃. Reports are transmitted via the ACH₃, ACH₇, ACH₉, ACH₁₀, and ACH₁₁ models. Their models calculate and record the results of the path analysis data. After the CAA_CNTR receives four types of the reports, it saves the sum of the results and waits for the next state transition.

Candidate path₃ (the existing path): ACH₂ forwards four reports to ACH₁ in a regular sequence. The reports are sent via ACH₇, ACH₆, ACH₁₀, and ACH₁₁ for calculating the analysis data. The CAA_CNTR computes the sum of the results after receiving the report.

If the calculation results of the candidate paths are 15,060, 16,500, and 23,640 μJ, the CAA_CNTR selects candidate path₁ from among the candidates using Algorithm 3 in section “Selection and configuration of a path” with the results of the simulation analysis. The CAA_CNTR forwards the selected path. Such a choice is the best among all results, but it is rare that this path is chosen.

Simulation results

In this section, we analyze the consumed energy of the candidate paths and evaluate the performance of our proposed scheme.

Paths analysis

To determine an effective path, the proposed scheme extracts candidate paths in the CAA, analyzes the performances of the paths, and selects a path through path analysis. In the proposed scheme, as models are operated through the simulation, it is important to execute them as if they were a target system WSN. We use the simulation methodology instead of using simple calculations, then create actual data such as real reports, and finally inject data into the path. In the CAA_CNTR model, the sum of the energy consumption of a path is calculated as follows

\[ TE = R_{size}((E_t \cdot HC) + (E_r \cdot (HC - 1))) + (E_v \cdot VM_c) \]  

(3)

In equation (3), \( R_{size} \) is the size of a report, \( E_t \) is the energy consumed to transmit, \( E_r \) is the energy consumed to receive, \( HC \) is the distance from the source to the controller model, \( E_v \) is the energy consumed to verify a report, and \( VM_c \) is the count of verified MACs.

For example, as shown in Figure 4, we assume that the hop counts of candidate paths 1, 2, and 3 are 4, 5, and 6, respectively, and that the \( VM_c \) in the paths is 1, 2, and 3, respectively. As intermediate models forward reports along the paths, the energy consumption of the three paths is 3765, 4875, and 5910 μJ, respectively. After completing the four types of actual data forwarding, their energy consumption amounts to, respectively, 15,060, 16,500, and 23,640 μJ and those values are recorded. Our proposed scheme selects candidate path₁ because it has the lowest energy consumption and notifies the BS of the path. Actually, the proposed scheme leads to energy savings of up to 12% instead of causing a large amount of energy consumption as described in section “Simulation results.” The proposed scheme focuses on energy savings to effectively forward more reports rather than creating a high verification count after the compromised node is identified.

Simulation results

We performed simulation experiments to evaluate the proposed scheme and compared it to PVFS and CAA-PVFS. In the initialization phase, a sensor field consists of 100 CHs and 900 MBs for a WSN and 100 DCNs for the Comm-Arch. All nodes are evenly distributed in each cluster, which contains one CH, nine MBs, and one DCN. Their transmission ranges are 150, 60, and 150 m, respectively. They are implemented using atomic DEVS models and are coupled for the sensor network and Comm-Arch. When they receive input data, they execute their state transitions as in real sensor nodes. The BS is located in the lower middle section of the sensor field.

The size of each report is 36 bytes, and the key size is 8 bytes. In Ye et al.,¹⁴ each node uses 16.25 μJ per byte to transmit, 12.5 μJ per byte to receive, and 15 μJ per byte to generate. To verify a MAC at the verification nodes, each node consumes 75 μJ. In Nam and Cho,⁴ each DCN consumes 19 μJ and 0.006 μJ per byte are consumed to compress and encrypt. We randomly selected 10 specific clusters from the 100 clusters to generate five FPAs and five FNAs. Compromised nodes are located within 5–10 hops, and they generate attacks according to the false traffic ratio (FTR). To verify MACs of the FPAs and FNAs, a report threshold is set to 2 or 3. In the simulation experiments, we randomly generated 500 events. There was no packet loss in the experiment.

Figure 5 shows the attack success count of the FPAs and FNAs versus the FTR for PVFS, CAA-PVFS, and
the proposed scheme. In cases (a) and (b), the proposed scheme and CAA-PVFS are almost similar to the attack success count since they identify compromised nodes to block false data as compared to PVFS. In addition, the two schemes are consistent with the number of detected FPAs and FNAs, even though the FTR increases, unlike PVFS that probabilistically detects attacks.

Figure 6 shows the energy consumption versus distance with respect to threshold values 2 and 3 to present the performances of the three schemes with FTR values of 0%, 10%, 30%, 50%, 70%, and 90%. In cases (a) and (b), nearly the same amount of energy is consumed between 9 and 11 due to the forwarding of a legitimate report. PVFS has a delicate difference using probabilistic en-route filtering within small distances. In cases (c) and (d), the difference in energy consumption between one and eight hops is as a result of the identification of compromised nodes. On the other hand, the proposed scheme shows an energy savings of roughly 8% which results from configuring an effective path on the same hop as compared to CAA-PVFS. Thus, the proposed scheme saves more energy with a shorter distance traveled than the other schemes.

Figure 7 shows the total energy consumption of the sensor network versus the FTR for three schemes. Overall, in cases (a) and (b), the proposed scheme generally improves energy efficiency due to the establishment of an effective routing path as compared to the other schemes. For a threshold of two (Figure 7(a)), the proposed scheme improves by 13% and 11% for an FTR of 10% and by 32% and 39% for an FTR of 90% as compared to PVFS and CAA-PVFS, respectively. For a threshold of three, the proposed scheme improves by 7% and 5% for an FTR of 10% and by 9% and 39% for an FTR of 90% as compared to the aforementioned schemes. Therefore, the proposed scheme improves the energy effectiveness by an average of 9% as compared to CAA-PVFS and results in an extension in the network lifetime.

Related work

Recently, WSNs have exploited computational intelligence (CI) techniques to deal with various issues such as security, data aggregation, data routing, and deployment and location optimization. Some techniques require excessive computation, high energy, and lots of storage to execute complex algorithms. In WSNs, it is still important to solve such issues using limited resources. In this section, we discuss several existing filtering schemes and path configuration schemes for increased resource efficiency and CI.

Attack detection schemes

In WSNs, several en-route filtering schemes focus on the detection of only FPAs. As a result, Li et al. proposed PVFS to simultaneously detect both FPAs and FNAs by introducing a voting method. According to this method, intermediate nodes either immediately drop or continually forward reports through a predefined threshold until the verification process is complete. PVFS achieves strong protection against the two attacks while maintaining a sufficiently high filtering power. Nghiem and Cho suggested a fuzzy-based interleaved multi-hop authentication (FIMA) scheme based on an interleaved hop-by-hop authentication (IHA) scheme. In FIMA, the fuzzy logic system is executed in the BS based on three input parameters, which are the remaining energy of the intermediate nodes, the distance from the source cluster to the BS, and the cumulative number of false MACs in reports generated by that source cluster. The fuzzy system selects verification nodes for the best energy savings.
and network security. FIMA effectively detects the two attacks and improves the energy efficiency by 13% as compared to IHA. Lee and Cho\textsuperscript{19} proposed a scheme for adaptively countering FPAs and FNAs. The scheme detects the security attacks and selects efficient countermeasures against the attacks through the fuzzy system in the BS. It provides high-energy savings and reliable data delivery by adaptively activating and selecting...
countermeasures. Akram et al.\textsuperscript{6} suggested fuzzy-based adaptive selection of intermediate verification nodes to achieve higher energy savings compared to PVFS. All of the source nodes choose effective verification nodes using a fuzzy rule–based decision-making engine, unlike probabilistic selection in PVFS. The scheme enhances energy conservation and guarantees sufficient protection against FPAs and FNAs. Unlike the en-route filtering schemes in Nam and Cho,\textsuperscript{4} a CAA is integrated with PVFS to greatly complement the weaknesses of the security protocol, unlike conventional techniques. CAA-PVFS identifies the compromised nodes using its context-awareness implemented by simulation models against the FPAs and the FNAs. The scheme reduces the number of attack successes by excluding the compromised nodes in the sensor network and improves energy efficiency by reducing the attack data.

The en-route filtering schemes focus on early detection of false data and improving energy efficiency within a short distance against FPAs and FNAs. CAA-PVFS identifies compromised nodes to immediately obstruct false data injected by the nodes. Our proposed scheme determines an energy efficient path using new CAA to avoid the compromised node based on discrete event simulation.

\textbf{Path configuration schemes}

Several path configuration schemes have been suggested that use various techniques to hedge against attacks of the network. Savvides et al. proposed a technique called AHLos (ad hoc localization system)\textsuperscript{38} to discover the locations of sensors using a set of distributed iterative algorithms. In the technique, each node dynamically discovers its own location and sets effective paths. The node estimates its distance from its neighbors using a range of techniques and uses range and location information to estimate their positions. Kim et al.\textsuperscript{20} proposed a path renewal method (PRM) to enable balanced energy consumption, which extends the network lifetime. In PRM, as the remaining energy of the next node decreases below a threshold value, the source node chooses a new next forwarding node based on an evaluation function. PRM enhances the filtering capability and the network lifetime as compared to statistical en-route filtering (SEF)\textsuperscript{14} scheme. Shahzad et al. proposed pre-deterministic key-distribution-based commutative cipher en-route filtering (CCEF),\textsuperscript{9} which extends the network life with the selection of a next forwarding node. In the scheme, to select the next forwarding node, the distance, the remaining energy, and key information are considered among the neighbors. The scheme improves energy efficiency and extends the network lifetime as compared to CCEF. Nam et al. proposed a fuzzy rule–based path configuration method\textsuperscript{8} to select the next forwarding node based on a fuzzy logic system to hedge against sinkhole attacks.\textsuperscript{8,39–41} A source considers multiple characteristics of the next forwarding node based on the fuzzy rule–based system to determine a suitable selection. This method achieves improved energy efficiency, distributed traffic, and secure transmission.

Most schemes establish a path using the information of only the next forwarding node in a source node. Our proposed scheme, on the other hand, analyzes candidate paths through the simulation and selects a path using the analysis results.

![Figure 7](image-url)
Conclusion and future work

In WSNs, sensors are exposed to FPAs and FNAs by compromised nodes due to their stringent energy and computational limitations. Several path configuration schemes suggest various methodologies with consideration for the next nodes to create a detour around the compromised node. Our proposed scheme extracts candidate paths in its CAA, analyzes paths through a discrete event simulation, and determines an energy efficient path from among the candidates. Thus, our scheme results in an extension of the network lifetime through the selection of an effective path. The proposed scheme was able to achieve the following contributions:

- Determination of an effective path by simulation: The proposed scheme extracts candidate paths considering the status of the sensor nodes after identifying a compromised node in the CAA. This scheme analyzes paths through simulation using actual data of the sensor network and selects an effective path among them. The CAA provides the BS with the selected path.
- Improvement in energy efficiency: The proposed scheme determines the effective path through path analysis using simulation. In the sensor network, specific nodes make a detour around the compromised node using the determined path. The proposed scheme results in an extension of the network lifetime using a reduced number of data transmissions between the sensor nodes.

For future work, we will improve the CAA technique for detecting various attacks in WSNs and apply suggested security protocols in the KB of the CAA to further enhance the energy efficiency and network security.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: 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. NRF-2018R1D1A1B07048961).

ORCID iD

Su Man Nam https://orcid.org/0000-0001-6222-2096

References

1. Akkaya K, Su W, Sankarasubramaniam Y, et al. A survey on sensor networks. *IEEE Commun Mag* 2002; 40(8): 102–114.
2. Akkaya K and Younis M. A survey on routing protocols for wireless sensor networks. *Ad Hoc Netw* 2005; 3(3): 325–349.
3. Liu X. A survey on clustering routing protocols in wireless sensor networks. *Sensors* 2012; 12(8): 11113–11153.
4. Nam SM and Cho TH. Context-aware architecture for probabilistic voting-based filtering scheme in sensor networks. *IEEE Trans Mobile Comput* 2017; 16(10): 2751–2763.
5. Li F, Srinivasan A and Wu J. PVFS: a probabilistic voting-based filtering scheme in wireless sensor networks. *IJUSN* 2008; 3(3): 173–182.
6. Akram M and Cho TH. Energy efficient fuzzy adaptive selection of verification nodes in wireless sensor networks. *Ad Hoc Netw* 2016; 47: 16–25.
7. Al-Karaki JN and Kamal AE. Routing techniques in wireless sensor networks: a survey. *IEEE WIREC Commun* 2004; 11(6): 6–28.
8. Nam SM and Cho TH. A fuzzy rule-based path configuration method for LEAP in sensor networks. *Ad Hoc Netw* 2015; 31: 63–79.
9. Toor KS and Cho TH. Extending the network lifetime by pre-deterministic key distribution in CCEF in wireless sensor networks. *Wirel Netw* 2015; 21(8): 2799–2809.
10. Yu Z and Guan Y. A dynamic en-route filtering scheme for data reporting in wireless sensor networks. *IEEE/ACM Trans Netw* 2010; 18: 150–163.
11. Karlof C and Wagner D. Secure routing in wireless sensor networks: attacks and countermeasures. *Ad Hoc Netw* 2003; 1(2-3): 293–315.
12. Nghiem TP and Cho TH. A fuzzy-based interleaved multi-hop authentication scheme in wireless sensor networks. *J Parallel Distrib Comput* 2009; 69(5): 441–450.
13. Karl H and Willig A. Protocols and architectures for wireless sensor networks. Hoboken, NJ: John Wiley & Sons, 2007.
14. Ye F, Luo H, Lu S, et al. Statistical en-route filtering of injected false data in sensor networks. *IEEE J Select Area Commun* 2005; 23(4): 839–850.
15. Nam SM and Cho TH. Context aware architecture based WSN security for detecting compromised nodes in probabilistic voting-based filtering. In: 2016 6th international workshop on computer science and engineering, Tokyo, Japan, 17–19 June 2016, pp.161–166.
16. Zhu S, Setia S, Jajodia S, et al. An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks. In: *IEEE symposium on security and privacy*, Berkeley, CA, 12 May 2004, pp.259–271. Berkeley, CA: IEEE.
17. Lu R, Lin X, Zhu H, et al. BECAN: a bandwidth-efficient cooperative authentication scheme for filtering injected false data in wireless sensor networks. *IEEE Trans Parallel Distrib Syst* 2011; 23(1): 32–43.
18. Yang X, Lin J, Yu W, et al. A novel en-route filtering scheme against false data injection attacks in cyber-physical networked systems. *IEEE Trans Comput* 2013; 64(1): 4–18.
19. Lee HY and Cho TH. A scheme for adaptively counteracting application layer security attacks in wireless sensor networks. *IEICE Trans Commun* 2010; 93(7): 1881–1889.
20. Kim JM, Han YS, Lee HY, et al. Path renewal method in filtering based wireless sensor networks. *Sensors* 2011; 11(2): 1396–1404.
21. Nam SM and Cho TH. Modeling and simulation of threshold analysis for PVFS in wireless sensor networks. *Int J Res* 2016; 4(8): 1–10.
22. Choi K, Bae DH and Kim T. An approach to a hybrid software process simulation using the DEV5 formalism. *Softw Proces* 2006; 11(4): 373–383.
23. Hong KJ and Kim TG. DEVSpecL: DEV5 specification language for modeling, simulation and analysis of discrete event systems. *Inform Softw Technol* 2006; 48(4): 221–234.
24. Zeigler BP. Object-oriented simulation with hierarchical, modular models: intelligent agents and endomorphic systems. Cambridge, MA: Academic Press, 2014.
25. Zeigler BP, Praehofer H and Kim TG. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Cambridge, MA: Academic Press, 2000.
26. Kofman E. Discrete event simulation of hybrid systems. *SIAM J Sci Comput* 2004; 25(5): 1771–1797.
27. Du X, Xiao Y, Guizani M, et al. An effective key management scheme for heterogeneous sensor networks. *Ad Hoc Networks* 2007; 5(1): 24–34.
28. Du X, Guizani M, Xiao Y, et al. Two tier secure routing protocol for heterogeneous sensor networks. *IEEE T Wirel Commun* 2007; 6(9): 3395–3401.
29. Ottoo G, Hamelinck T, Preneel B, et al. AES data encryption in a ZigBee network: software or hardware? (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering) In: *International conference on security and privacy in mobile information and communication systems*, Sicily, 27–28 May 2010, pp.163–173. New York: Springer.
30. Didla S, Ault A and Bagchi S. Optimizing AES for embedded devices and wireless sensor networks. In: *Proceedings of the 4th international conference on testbeds and research infrastructures for the development of networks & communities*, Brussels, 18–20 March 2008, pp.1–10. New York: ACM.
31. Mangard S, Pramstaller N and Oswald E. Successfully attacking masked AES hardware implementations. In: *International workshop on cryptographic hardware and embedded systems*, Worcester, MA, 12–13 August 2005, pp.157–171. Berlin: Springer.
32. Karp R. Minimum-redundancy coding for the discrete noiseless channel. *IRE Tran Inform Theory* 1961; 7(4): 27–38.
33. Medeiros HP, Maciel MC, Demo Souza R, et al. Lightweight data compression in wireless sensor networks using Huffman coding. *Int J Distrib Sensor Netw* 2014; 10(1): 1–11.
34. Kolo JG, Shannumag SA, Lim DWG, et al. An adaptive lossless data compression scheme for wireless sensor networks. *J Sensor* 2012; 2012: 1–20.
35. Sheng J, Yang C and Herbordt MC. Hardware-efficient compressed sensing encoder designs for WBSNs. In: *2015 IEEE high performance extreme computing conference (HPEC)*, Waltham, MA, 15–17 September 2015, pp.1–7. New York: IEEE.
36. Johnson DB. A note on Dijkstra’s shortest path algorithm. *J ACM* 1973; 20(3): 385–388.
37. Lee HY and Cho TH. Key inheritance-based false data filtering scheme in wireless sensor networks (Lecture Notes in Computer Science). In: *International conference on distributed computing and internet technology*, Bhubaneswar, India, 20–23 December 2006, pp.116–127. New York: Springer.
38. Savvides A, Han CC and Strivastava MB. Dynamic fine-grained localization in ad-hoc networks of sensors. In: *Proceedings of the 7th annual international conference on mobile computing and networking*, Rome, 16–21 July 2001, pp.166–179. New York: ACM.
39. Zhu S, Setia S and Jajodia S. LEAP: efficient security mechanisms for large-scale distributed sensor networks. In: *Proceedings of the 10th ACM conference on computer and communications security*, Washington, DC, 27–31 October 2003, pp.62–72. New York: ACM.
40. Zhu S, Setia S and Jajodia S. LEAP +: Efficient security mechanisms for large-scale distributed sensor networks. *ACM T Sensor Netw* 2006; 2(4): 500–528.
41. Lim CH. LEAP + +: a robust key establishment scheme for wireless sensor networks. In: *28th international conference on distributed computing systems workshops*, Beijing, China, 17–20 June 2008, pp.376–381. New York: IEEE.