An Adaptive Key Redistribution Method for Filtering-based Wireless Sensor Networks

Jin Myoung Kim[1] and Hae Young Lee[2*]
[1]College of Information and Communication Engineering, Sungkyunkwan University
Suwon 16419 Republic of Korea
[e-mail: jm.kim@skku.edu]
[2]Major in Digital Security, Cheongju University
Cheongju 28503 Republic of Korea
[e-mail: haelee@cju.ac.kr]
*Corresponding author : Hae Young Lee

Received August 18, 2019; revised December 8, 2019; revised March 2, 2020; accepted April 6, 2020; published June 30, 2020

Abstract

In wireless sensor networks, adversaries may physically capture sensor nodes on the fields, and use them to launch false positive attacks (FPAs). FPAs could be conducted by injecting forged or old sensing reports, which would represent non-existent events on the fields, with the goal of disorientating the base stations and/or reducing the limited energy resources of sensor nodes on the fields. Researchers have proposed various mitigation methods against FPAs, including the statistical en-route filtering scheme (SEF). Most of these methods are based on key pre-distribution schemes and can efficiently filter injected false reports out at relay nodes through the verification of in-transit reports using the pre-distributed keys. However, their filtering power may decrease as time goes by since adversaries would attempt to capture additional nodes as many as possible. In this paper, we propose an adaptive key distribution method that could maintain the security power of SEF in WSNs under such circumstances. The proposed method makes, if necessary, BS update or re-distribute keys, which are used to endorse and verify reports, with the consideration of the filtering power and energy efficiency. Our experimental results show that the proposed method is more effective, compared to SEF, against FPAs in terms of security level and energy saving.

Keywords: False positive attacks, mitigation, key re-distribution, security, wireless sensor networks

http://doi.org/10.3837/tiis.2020.06.011 **ISSN : 1976-7277**

1. Introduction

Recently, the advancement of wireless communication technologies and micro electro-mechanical systems have enabled the development of low-cost, high-performance, tiny sensor nodes that provide various functionalities and communicate with other devices via wireless links [1, 2, 3, 4]. By networking many sensor nodes on a field that needs to be monitored, a wireless sensor network (WSN) can be formed [5, 6]. Then, events of interest on the field are reported to the users, in which sensing reports are generated by the detecting nodes and then delivered to the base stations (BSs) through multiple number of hops.

Sensor nodes are often deployed in hostile environments, such as battlefields, which results in leaving the nodes unattended. Thus, adversaries may physically capture sensor nodes on the fields without being detected. Due to cost-constraints, these nodes do not usually equip with tamper-resistant hardware, so that the adversaries could obtain cryptographic keys loaded on the captured nodes [7]. With the captured nodes, the adversaries may then launch various security (probably insider) attacks since they may take control of the nodes (e.g., by re-programming). One type of such attacks is false positive attacks (FPAs) in which forged or old sensing reports are injected through the captured nodes. In FPAs, the goals of the adversaries may include to make BSs confused by reporting non-existent events, which may involve real-world responses, and/or to consume the limited energy resources of battery-powered sensor nodes on the field [3, 4, 7, 8, 9].

To mitigate the damage from FPAs, forged sensing reports (including replayed ones), generated using compromised information, injected through captured nodes, should be detected and discarded as early as possible. To this end, relay nodes that deliver sensing reports toward BSs should be able to verify the legitimacy of in-transit reports and drop false ones. Many FPA countermeasures [2, 10, 11, 12, 16] use key sharing, through pre-distribution, between source nodes and relay nodes; source nodes use keys for endorsement of reports and relay nodes use them for verification of the reports. Thus, they are often called key pre-distribution-scheme-based filtering solutions.

The statistical en-route filtering scheme (SEF), which is the first type of such solutions, was proposed by Ye *et al*. [11]. In SEF, every sensing report must carry a certain number of message authentication codes (MACs) that are used to verify the authenticity and integrity of the report. When a node generates a report, it must collect different MACs, generated by its neighboring nodes using different keys from different partitions in the global key pool, and then attach the MACs into the report. The interleaved hop-by-hop authentication scheme (IHA) [10] also uses a pre-distribution scheme to filter forged reports out during the forwarding process. In IHA, every key used to endorse and verify reports is shared between two nodes whose distance is a fixed number of hops on a routing path. Thus, in IHA, a forged report could be detected within the fixed number of hops. The key-inheritance-based filtering scheme [12], a modified version of IHA, was proposed to maximize the early detection capability.

Especially in SEF and its variations, such as the dynamic en-route filtering scheme (DEF) [2], it could be very difficult or even impossible to estimate the actual detection power of each individual path since keys used to endorse and verify reports are randomly pre-distributed to nodes. Also, the detection power could decrease as time goes by since adversaries would attempt to capture additional nodes, as many as possible; in the worst case, SEF may be useless.

Furthermore, the detection power may change due to deployment of additional nodes or changes of routing paths.

In this paper, we propose an adaptive key distribution method for maintaining the security power of SEF in WSNs. In the proposed method, BS records the number of routing paths that were used to carry out successful FPAs (i.e., to deliver forged reports to BS). If the number of such routing paths has reached a pre-defined security threshold value, BS updates or re-distributes keys, with the consideration of the security power (i.e., the early detection power) and energy efficiency. For the re-distribution of keys, a modified version of the tree-based key management scheme is used in the method. Our experimental results show that the proposed method, compared to SEF, is more effective against FPAs in terms of security level and energy saving.

2. Background

2.1 False Positive Attacks and Countermeasures

In a WSN, an adversary could physically capture, without being detected, some sensor nodes on the field, which may result in the compromise of the nodes. He/she may then achieve full control over the nodes, by reading their memories and influencing operations of the programs on the nodes [22]. With the full control, he/she can launch so-called *false positive attacks* (FPAs) [3, 4, 7, 8, 9, 10, 11, 23], injecting forged reports into the network, as shown in **Fig. 1**.

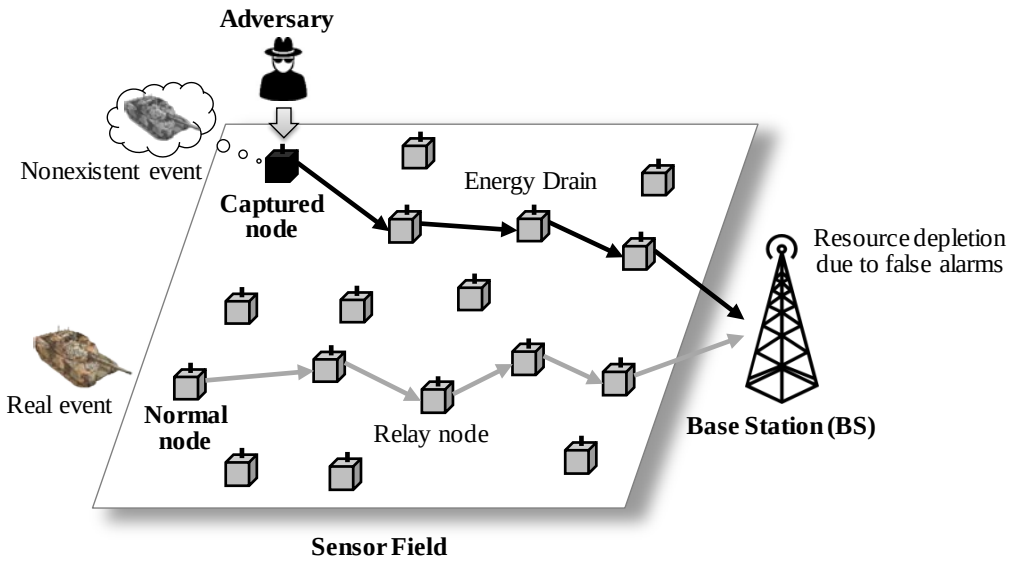

Fig. 1. Overview of false positive attacks (FPAs).

His/her goals may include to confuse BS by reporting non-existent events on the field. To this end, forged or old sensing reports that represent non-existent events would be injected into the network through the compromised nodes. Once delivered to BS, these forged reports may cause false positive alarms, which may lead to physical responses, such as dispatch of responding units. He/she may have another goal; to consume the limited energy resources of the network. To that end, a huge number of forged reports would be injected into the network

through the nodes. While they traverse the network, the energy resources of the relay nodes may be quickly depleted.

To mitigate the damage from FPAs, forged reports should be detected and dropped as early as possible before they consume a large amount of the energy resources of battery-powered nodes. Also, a few ones delivered to BS without being filtered out should be rejected at BS [11] in order to prevent false alarms. Digital signature-based techniques may be used to detect such forged reports [13], but involve relatively complex computations, e.g., with big numbers [14]. Thus, these techniques are unsuitable for most WSNs [15].

Researchers have proposed relatively lightweight security solutions [2, 3, 4, 10, 11] that use symmetric ciphers for en-route detection of forged reports. Ye *et al*. proposed the first en-route detection solution, SEF [11], in which a report for an event should carry multiple MACs generated using symmetric keys from the mulitple detecting nodes. To generate such a report, each of the detecting nodes generates a MAC over the report contents using one of its keys and then sends the MAC to the elected node that has been chosen among the detecting nodes. Then, the elected node compiles a report with these MACs and forwards the report to the direct upstream node. Upon receiving a report, a relay node may verify the legitimacy of the report by comparing one of the MACs in the report with a MAC generated using one of its keys. In SEF, most forged report could be filtered out within 10 hops. IHA proposed by Zhu *et al*. [10] can detect forged reports deterministically; a forged report can be detected within a certain number of hops from the source unless the adversary has compromised the whole nodes in the source. Every node shares symmetric keys with two other nodes – one of the upstream nodes and one of the downstream nodes – on the path from a source to BS. Upon receiving a report, a relay node verifies a MAC generated by one of the downstream nodes in the report and replaces it with a MAC generated using the key shared with one of the upstream nodes. Yu and Guan proposed DEF [2, 16] that could increase the detection power, compared to SEF. In DEF, keys in every node are disseminated to its surrounding nodes during the initial phase, and then used by the surrounding nodes to verify reports generated from the node.

Most of the solutions provide the same (or similar) security level against FPAs, in which the security level (i.e., the detection power) is basically determined based on keys assigned or disseminated to nodes on the field. In this paper, we call this phase the *initial phase*. The security level of SEF is determined by the number of keys loaded on each node and the number of MACs required for each report, while that of IHA is determined only by the number of MACs required for each report. In DEF, the security level is determined by the average number of 1-hop neighboring nodes and TTL (time-to-live), which is used to disseminate keys duringn the initial phase.

2.2 Tree-Based Key Management Scheme

Wallner *et al*. proposed a hierarchical key management structure, called the logical key hierarchy (LKH) [17, 18, 19]. In LKH, keys are managed in the form of a tree structure, by a key distribution center (KDC). KDC uses the tree structure to distribute or update group keys when group members have changed. Each node, including the root node, of the tree represents a symmetric key. Each terminal node, also representing a symmetric key, of the tree is assigned to a group member; thus, the node in the graph would be the member's key. Every group member stores all the keys on the path, from the terminal node assigned to itself, to the root node.

LKH can be applied to WSNs with the support of other keying protocols, such as pre-deployed keying, for maintaining the freshness of shared keys [17, 20]. In the tree structure shown in **Fig. 2**, KDC logically builds a series of symmetric keys, i.e., GK, SGK_0, SGK_1, SGK_2, SGK_3, MK_0, MK_1, MK_2, and MK_3. In the figure, filled squares M_0, M_1, M_2, and M_3 represent sensor nodes (i.e., group members). Each sensor node stores all the keys on the path between the node assigned to itself and the root node. For example, node M_0 stores keys MK_0, SGK_2, SGK_0, and GK. MK_0 is a unique key of sensor node M_0, which is shared with only BS. SGK_2 and SGK_0 are subgroup keys (SGKs). Each SGK is shared with all members in the associated subgroup and BS. For example, SGK_2 is shared with sensor nodes M_0, M_1, and BS. GK is the group key. It is shared with all the sensor nodes and BS in the network. These keys are used to encrypt/decrypt (sub)group communications.

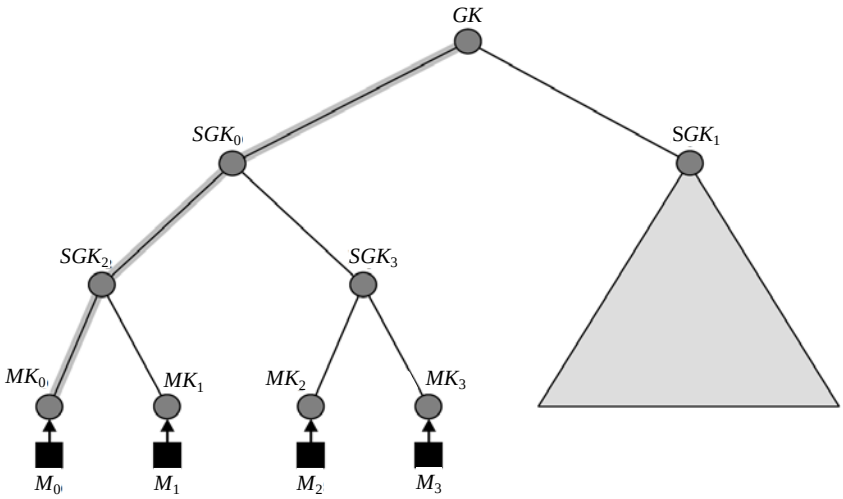

Fig. 2. Tree structure example for key management.

3. Proposed Key Distribution Scheme

3.1 Problem Statement

Since the security level provided by FPA countermeasures is basically determined based on nodes' keys, the level (the detection power) would decrease as time goes by; an adversary would attempt to capture additional nodes as many as possible. Once the adversary has compromised a certain number of keys from different partition in the global key pool, forged reports generated by him/her would be never detected by relay nodes.

In SEF, the network administrator could estimate the detection power based on the following four factors:

- The total number of symmetric keys in the global key pool,
- The number of partitions in the pool,
- The number of keys preloaded in each sensor node, and
- Security threshold value T that defines the number of MACs attached in each sensing node.

After the initial phase, each relay node will go into checking MACs of every in-transit report if the node has one of the keys used to generate the MACs. A forged report would not have T legitimate MACs (unless the adversary has captured many nodes), generated using 'different keys' from 'different partitions' of the global key pool, so that the report could be detected and dropped by a relay node on the routing path to BS, as shown in **Fig. 3**. We call this phase the *runtime phase*.

Although the detection power could be estimated based on the above-mentioned four factors, the actual power of an individual path in the initial phase may differ from the estimated one due to a random fashion in key distribution and node deployment (e.g., by aircrafts). For example, the detection power may vary with the distance from BS; a long-haul delivery of reports would involve a greater number of en-route verifications. In the runtime phase, the detection power of the path may decrease as the adversary have captured more and more nodes.

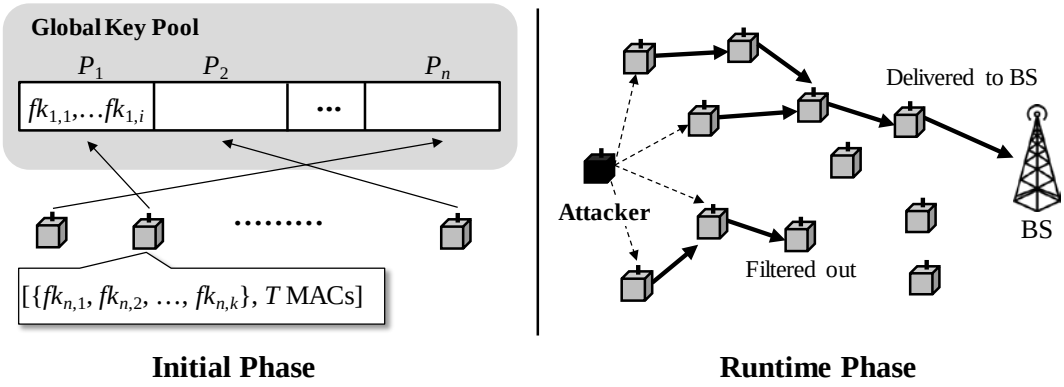

Fig. 3. Initial and runtime phases in SEF.

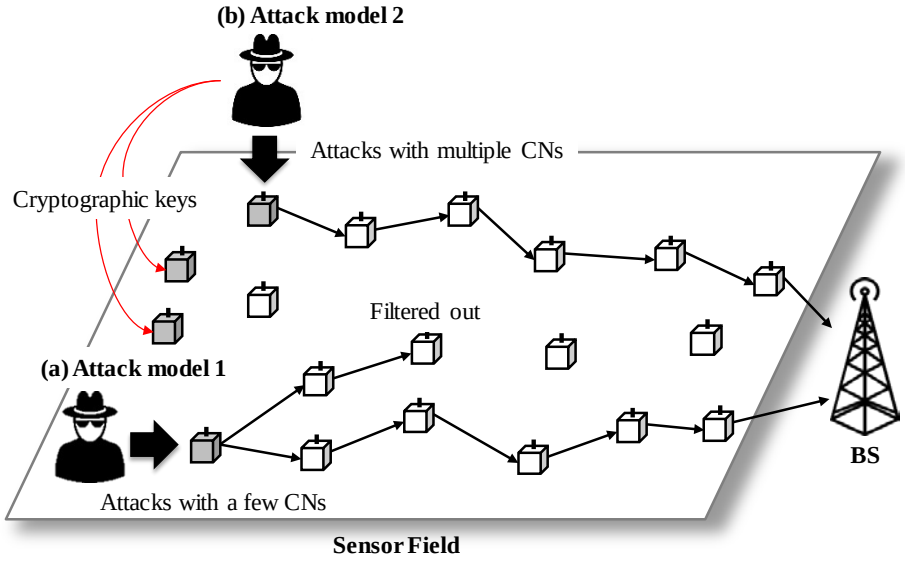

Fig. 4. Network and attack models in the proposed method.

The propose method considers the network and attack models shown in **Fig. 4**. Sensor nodes are deployed on an open outdoor field without infrastructure. As in [23, 24, 25, 26], the method assumes that BS knows the geographical locations of all nodes and can send a message directly to any node through a wireless link. However, a message generated by a source node would be delivered to BS through multiple hops unless the node is a 1-hop neighbor of BS.

The attack models that the proposed method considers are as follows: An adversary can physically capture some nodes on the field. All nodes do not equip with tamper-resistant hardware due to cost-constraints, so that the adversary can acquire all the information on the capture nodes. Then, (a) the adversary may use 'a few' compromised nodes to inject forged reports that represent non-existent events into the network. At this point, the adversary may be able to generate a few ($<< T$) legitimate MACs, generated using keys from the compromised nodes, for each forged report. The adversary may not duplicate legitimate MACs since a report carrying MACs generated using the same key will be dropped by a relay node without the verification of legitimacy of the MACs. Thus, the adversary would attach arbitrary (i.e., forged) MACs to forged reports. To conserve the limited energy resources of the network, these forged reports should be detected and dropped by relay nodes as early as possible. As time goes by, the adversary may attempt to compromise additional nodes in order to acquire (compromise) an enough number of keys. At some point, the adversary may have compromised T keys from different partitions of the pool. (b) In such a case, forged reports would be always delivered to BS; relay nodes would consider forged reports legitimate ones since the reports would have T legitimate MACs, generated using keys from different partitions of the pool. BS would also consider that the reports are legitimate. But BS would be soon able to recognize the attacks, e.g., by dispatch of responding units.

3.2 Proposed Method

When the adversary has launched FPAs, some forged reports may be delivered to BS. Each of them could be classified into one of the two cases shown in **Table 1**, which are associated with the two models of the attacks described in Subsect. 3.1.

Table 1. Cases, reasons, and mitigations of forged report delivery.

Case	Reason	Mitigation
(a) A forged report with many false MACs has been delivered.	Relay nodes do not have keys to verify the forged MACs.	Update keys of the relay nodes on the path.
(b) A forged report with legitimate MACs has been delivered.	Many nodes have been compromised.	Replace the compromised keys with new keys.

(a) The first case is that a forged report with many false MACs has been delivered to BS. In this case, the adversary may have a few ($<< T$) captured nodes. Thus, he/she had to attach some false (arbitrarily generated) MACs to the report since every report must carry T MACs in SEF. Such a forged report should be filtered out during the relaying process. The delivery of such a report to BS would indicate that the false MACs in the report were never verified by the relay nodes, i.e., the nodes do not have keys to verify the forged MACs. Thus, the damage from FPAs may be resource exhaustion and could be mitigated by updating keys of the relay nodes on the path.

(b) The second case is that a report with T legitimate MACs has delivered to BS but is unmasked as a forged report, probably after involving a real-world response (e.g., dispatching

a unit to the location). In the case, the adversary may have T or more captured nodes, so that he/she could generate T MACs with keys from different partitions of the pool. This would indicate that SEF is currently useless against FPAs; every forged report will be delivered to BS and consume the energy resource of relay nodes. To mitigate the damage from FPAs, all compromised keys should be replaced with new keys.

Fig. 5 shows an overview of the proposed method. Forged reports are injected through captured nodes (CNs). Due to the probabilistic verification manner of SEF, even a forged report with many false MACs may be delivered to BS; the relay nodes may not have keys to verify the legitimacy of the false MACs in the report. When a forged report has been delivered to BS due to one of the reasons described in **Table 1**, the report is then sent to the adaptive key distribution system (AKDS). AKDS is composed of the corrupted path management (CPM) and the decision of key distribution (DKD). CPM records paths that delivered forged reports, called *corrupted paths* (CPs) hereafter. Once the number of CPs has reached a security threshold value, T_{path}, DKD determines a mitigation method: updates or replacement of keys.

The choice of T_{path} is important since T_{path} is related to a period of key redistribution. Optimal T_{path} would vary with the network configuration, operations, environments, and so on. Thus, the choice of T_{path} is a separate issue, beyond the scope of this paper. However, our experiment results (see Sect. 6) may be used as reference.

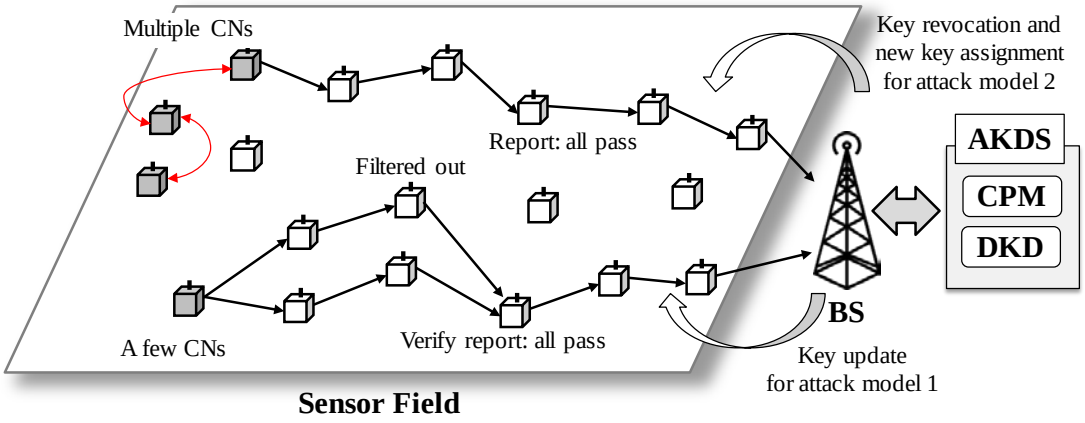

Fig. 5. Overview of the proposed method.

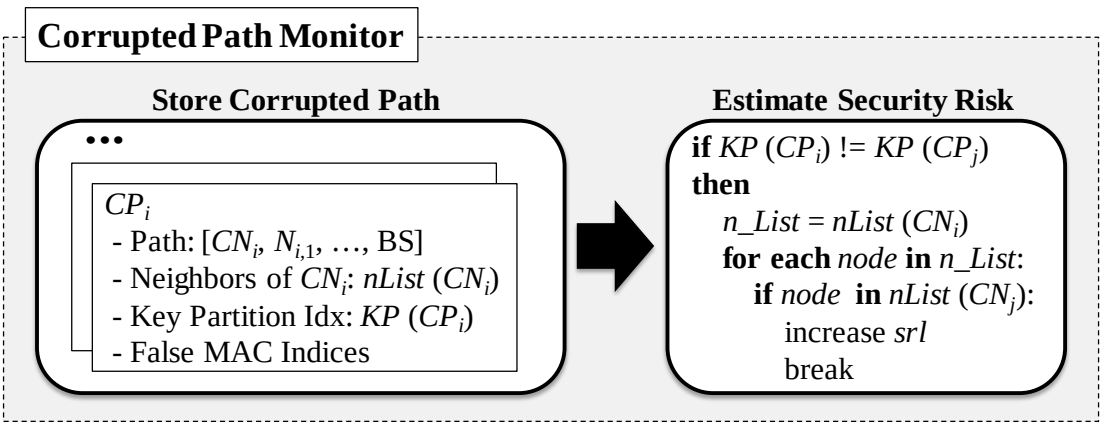

Fig. 6. Two modules for security risk estimation in CPM.

4. Adaptive Key Distribution

4.1 Corrupted Path Monitor

Fig. 6 shows the structure of CPM, which is a subsystem of AKDS. Through analyzing forged reports delivered to BS, CPM could extract some information related to FPAs, such as captured nodes, compromised keys, false MACs' indices, and CPs (i.e., the paths the reports traveled on). Suppose that a forged report has been delivered to BS. BS would verify the legitimacy of all the MACs in the report. Some of the MACs in the report would be correct, whereas the others would not. The correct MACs may be generated using 'compromised' keys, while the false ones would imply that the associated keys have not been compromised. The information extracted by CPM is then stored in the store corrupted path (SCP), a module of CPM. For corruped path CP_i from captured node CN_i to BS, the set of nodes including relay nodes such as $N_{i,1}$ on the paths, *nList* (CN_i), *KP* (CP_i), the indices of the false (i.e., arbitrarily generated by the adversary) MACs are stored, where *nList* (CN_i) is the neighboring nodes of CN_i and *KP* (CP_i) is the partition indices of the keys of the CP_i.

The network's security risk level *srl* is then estimated by the estimate security risk (ESR), another module of CPM, using the information stored in SCP. The risk level is basically influenced by the number of CNs. The estimation is performed for all the partition indices of the compromised keys. *srl* increases if arbitrary two CPs have different KPs and share a node on the paths.

4.2 Decision of Key Distribution

Fig. 7 shows the structure of DKD, which is another subsystem of AKDS. DKD determines a mitigation method, based on CPs and *srl* received from ESR. If *srl* reaches or exceeds T_{path} (i.e., attack model 2), keys exposed to the adversary (i.e., compromised keys) should be revoked and replaced with new keys. Some nodes may have the same compromised keys. These keys should be also replaced with new keys generated by DKD. If forged reports with some false MACs were not filtered on path *P* (i.e., attack model 1), keys of nodes on *P* should be updated by other keys in the pool.

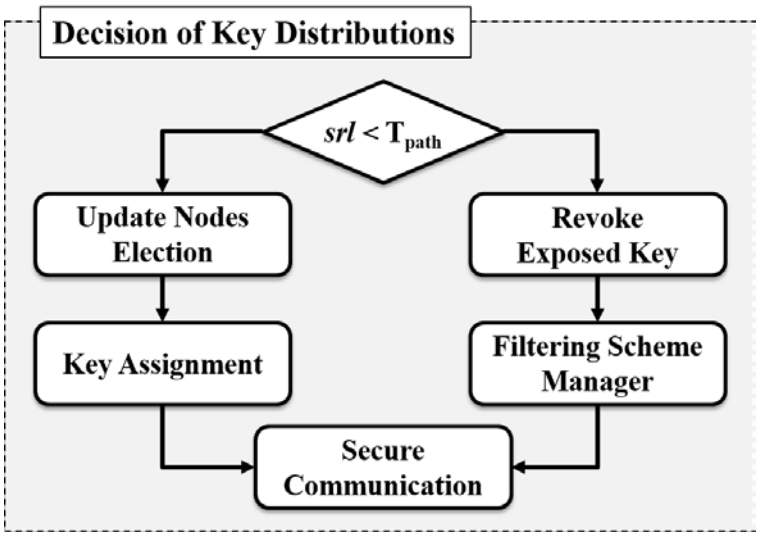

Fig. 7. Determination of a mitigation in DKD.

For arbitrary *CP*, let paths between each of *nList* (CP) and BS be P_0, P_1, …, P_n. Since *CN* may send a forged report to its neighboring nodes, [P_0, P_1, …, P_n] are potential CPs, *PCP*. Thus, the list of target nodes for key updates within *CP* and *PCP* is determined by Algorithm 1.

Algorihtm 1 Target Node Selection

Input: *CP*, *PCP*
Output: *update target nodes*

```
1:  initialize update_target_nodes
2:  for each P in PCP:
3:      inters = intersection (CP, P)
4:      if inters is Null then
5:          insert {P, P [0]} to update_target_nodes
6:      else
7:          if the distance of inters [0] ≤ len (CP) / 2 then
8:              insert {P, inters [0]}to update_target_nodes
9:          else
10:             dPaths: find downstream paths from inter [0]
11:             inter_path = MAX (intersection (dPaths))
12:             insert {P, inter_path [0]} to update_target_nodes
13: return update_target_nodes
```

In Algorithm 1, *PCP* are classified based on the intersection with *CP*. For each path *P* in *PCP*, if there is no intersection between *CP* and *P* (Line 4), *P* [0], which is the first element of *P*, is inserted into *update_target_nodes*. If there is one or more intersections, there can be two cases: 1) If the distance of *inters* [0] from *CN* is shorter than or equal to the half of the length of *CP* (i.e., the intersection nearest from *CN* is not so far from *CN*), then *inters* [0] is inserted to *update_target_nodes*, in order to achieve the early detection capability and energy saving. 2) If the distance of *inters* [0] from *CN* is longer than the half of the length of *CP*, the algorithm finds a list of downstream paths, *dPaths*, that have *inters* [0]. Within *dPaths*, the longest one *inter_path* is elected and then the first element of *inter_path*, *inter_path* [0] is inserted to *udpate_target_nodes*.

Algorihtm 2 Key Assignment

Input: *update_target_nodes*, *verifiable_key_set*
Output: *update_nodes*

```
1:  for each PN in update_target_nodes:
2:      if PN.node exists in CP then
3:          pop vk in verifiable_key_set without overlapping in KP (P)
4:      else
5:          pop vk in verifiable_key_set without overlapping in KP (DP
            (PN.node))
6:      insert {PN.node, vk} to update_nodes
7:  return update_nodes
```

After the selection of target nodes for key updates, verifiable keys (*vk*) to be assigned are selected by Algorithm 2. Here, *vk* can verify forged reports from *CP*. To maintain the diversity of cryptographic keys on *P*, *vk* is elected without overlapping *KP* (*PN.P*), which returns a list of key partition indices on *P*. If *PN.node* is on *CP*, *vk* is then assigned to *PN.node*. If not, the

algorithm finds downstream paths from *PN.node*, *DP* (*PN.node*). Then, *vk* without overlapping *KP* (*DP* (*PN.node*)) is elected. Here, *DP* (*N*) returns a list of downstream paths from node *N*.

5. Case Study: Secure Key Distribution

In group key-based secure communications, when a joining or leaving event of members in a group occurs, the group key and subgroup keys related to the members should be updated. However, it is difficult to directly apply a group key-based communication technique to WSN due to communication cost that is relatively expensive for sensor nodes having the limited energy resources. Also, every message in wireless communications is broadcasted, so that an adversary may be able to: eavesdrop messages, inject arbitrary messages, and replay old messages.

For applying a group key-based secure communication technique to WSN, we classify two kinds of cryptographic keys – filtering keys and message encryption keys – as shown in **Fig. 8**. These keys are distributed in the initial phase. If a forged report with correct MACs was delivered to BS, BS can know the compromised keys that were used to generate the correct MACs in the forged report. From the compromised keys, BS may then guess the captured nodes who need to be kicked out of the group. After the determination of leaving (kicked out) sensor nodes, BS updates the group key and subgroup keys to establish secure communications. Finally, filtering keys are re-distributed by considering the proposed scheme.

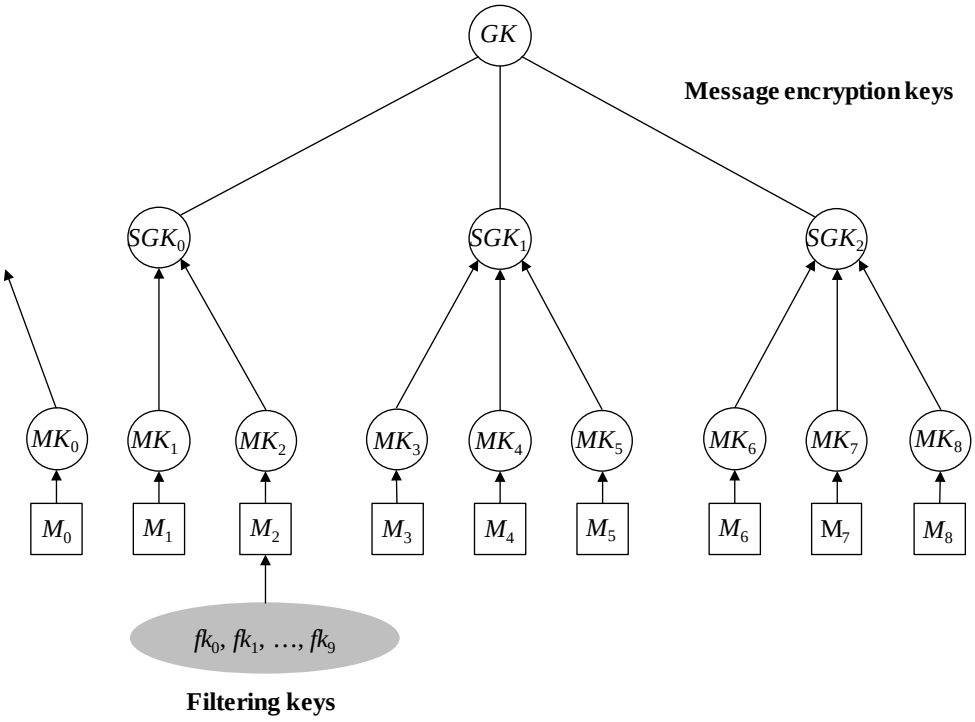

Fig. 8. Tree-based key management structure example.

Fig. 8 shows an example of a tree-based key management structure that manages 9 members. M_0, M_1, ..., M_8 denote sensor nodes. Let M_0 be a node that has been captured by an adversary. If some filtering keys on nodes need to be replaced with new ones, BS sends re-key messages with new filtering keys. If some filtering keys of M_2 need to be replaced (since M_2 has some compromised keys) with new filtering keys $\{fk_0, fk_1, ..., fk_9\}$, BS sends the following re-key messages:

- BS → (M_3, M_4, M_5): Enc $\{GK', SGK_1\}$
- BS → (M_6, M_7, M_8): Enc $\{GK', SGK_2\}$
- BS → (M_1): Enc $\{GK', SGK_0', MK_1\}$
- BS → (M_2): Enc $\{GK', SGK_0', (pi, ni, fk), ..., MK_2\}$

In the above equation, we apply the key-oriented, re-keying scheme proposed in [21]. *pi* denotes the previous key index. *ni* and *fk* denote the new key index and the filtering key for *ni*, respectively. Therefore, M_2 revokes the filtering key for *pi* and replaces the new filtering key for *ni*.

6. Simulation Results

To show the effectiveness of the proposed method (PM), we have conducted simulation-based experiments in which the original SEF (SEF-PRD) [11] was compared with PM. We use term SEF-PRD since keys are pre-randomly distributed before deployment in the original SEF. **Table 2** shows the environmental parameters for our simulation.

Table 2. Environmental parameters for simulation.

Class	Parameters	Values
Sensor Field	Size	100 × 50 m
	Number of nodes	200
Communications	Range	10m
	Size of a message	36 bytes
	Energy consumption for transmission	16.25μJ
	Energy consumption for reception	12. 5μJ
Encryption keys	Size of the global key pool	50
	Number of partitions	5
	Index numbers per partition	10
	Number of filtering keys per node	7

In the table, the energy consumption model for transmission and reception is based on [10]. Energy consumption due to computation within nodes is not considered. Our evaluation criteria for the comparison are as follows:

- Filtering succession ratio (FSR): the portion of ‘filtered’ reports among injected forged reports
- Average hop count (AHC): the average number of hops that injected forged reports traveled over
- Energy consumption (EC)

To evaluate FSR, we generated 1,000 forged reports and injected them into the network. We measured the number of forged reports that were successfully delivered to BS. Then, FSR is the portion of the remaining (i.e., dropped) ones among the injected forged reports. A higher FSR means that more forged reports were filtered out en-route. We could measure the early detection capability based on AHC. A smaller AHC means that forged reports were detected earlier. Since sensor nodes have the limited energy resources, simulation should be performed for measuring EC. A lower EC means that more energy resources were conserved. Note that the detection power of PM is basically equivalent to that of SEF-PRD since PM employs SEF to counter FPAs. However, PM involves key redistribution, which would be similar to resetting of SEF. Thus, FSR of PM could differ from that of SEF-PRD.

Fig. 9 shows FSR measured through our experiments when T_{path} is 30 (PM(30)), 100 (PM(100)), and 200 (PM(200)). As shown in the figure, PM is more secure than SEF-PRD; i.e., more forged reports were filtered out en-route in PM. One of the reasons would be that, in PM, keys are updated or replaced with the consideration of the detection power. Also, smaller T_{path} could achieve more security power in terms of the detection of forged reports since keys were updated and/or re-distributed more frequently under smaller T_{path}.

Fig. 10 shows AHC measured through our experiments. As shown in the figure, PM has more effective, compared to SEF-PRD, in terms of the early detection capability. One of the reasons would be that PM chooese target nodes for key updates by considering the distances from the captured nodes. Thus, PM could detect forged reports at an earlier stage of the forwarding, before they traveled over 3 hops. Although PM with larger T_{path} slightly degraded the early detection capability, PM was still superior than SEF-PRD in term of the capability.

Sensor nodes are energy-constraints, so that every novel solution for WSNs should consider energy first. **Fig. 11** shows the energy consumptions of PM and SEF-PRD when false traffic ratio is between 0% (all traffic is legitimate) and 100% (all traffic is forged and many CNs are involved). The energy consumptions of PM include the consumptions due to key updates and replacements. As shown in the figure, PM could save more energy resources than SEF-PRD, especially when false traffic ratio is high (i.e., FPA launched through many CNs). SEF-PRD could not detect forged reports generated with many CNs (i.e., when false traffic ratio is high). Thus, relay nodes in SEF-PRD just verify and then forward forged reports, which consumes energy due to computations and communications. In contrast to SEF-PRD, PM could achieve energy saving through enhancing the early detection capability with key updates and replacement.

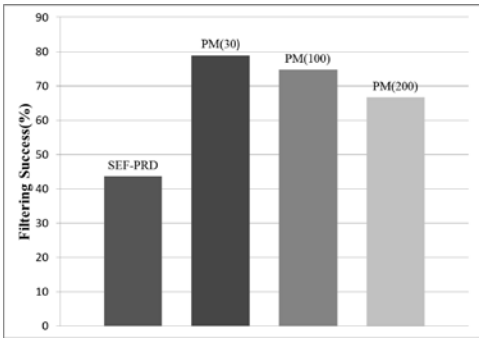

Fig. 9. FSR for 1,000 injections.

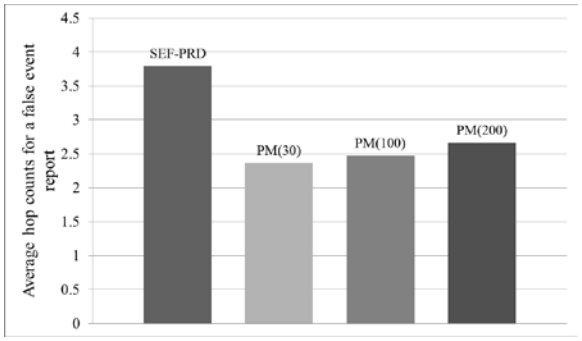

Fig. 10. AHC for 1,000 injections.

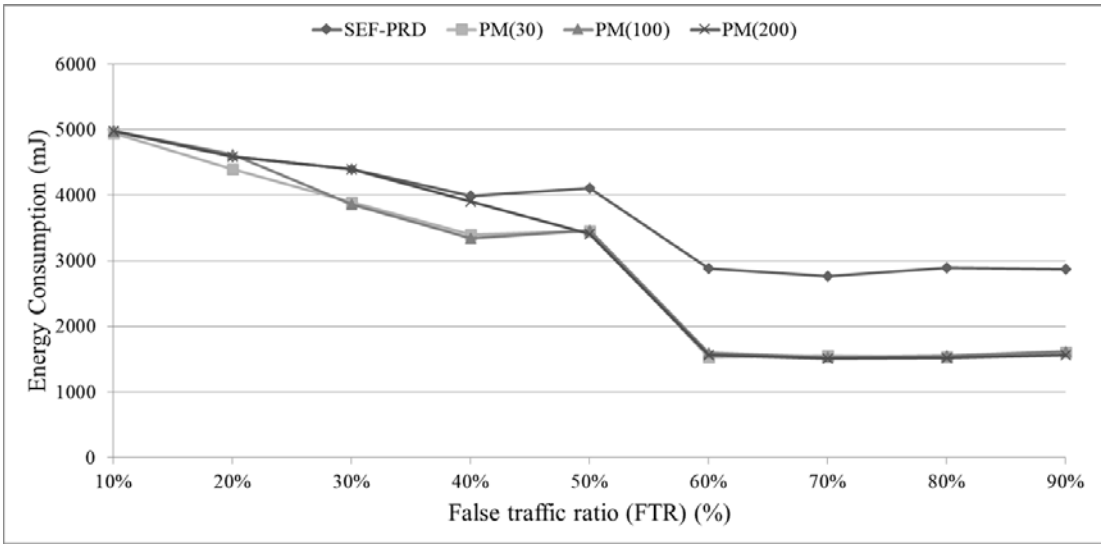

Fig. 11. EC for false traffic ratio (FTR).

7. Conclusions and Future works

In this paper, we proposed a method for maintaining SEF's filtering power against FPAs. While most of the existing FPA countermeasures do not consider the detection power reduction due to additional node compromise, the proposed method does; it monitors the security risk level of the network and flexibly reacts using re-distribution of message encryption keys and filtering keys with the consideration of the filtering power of routing paths. The experimental results showed that the proposed method could achieve some enhancement, compared to the original SEF, in terms of security and energy efficiency.

We will enhance the proposed method, by considering other factors (e.g., the ratio of successful attacks) and investigate optimal search methods for the target node selection in key update. Also, we will investigate methods for determining T_{path} with the consideration of various factors, such as network environments, configuration, operations, and so on. Additionally, intrusion detection schemes for detecting compromised nodes will be studied.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirici, "A Survey on Sensor Networks," *IEEE Communications Magazine*, 40, 102-114, 2002. Article (CrossRef Link)

[2] 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, pp. 150-163, Feb. 2010. Article (CrossRef Link)

[3] L. Rongxing, L. Xiaodong, Z. Haojin, L. Xiaohui and S. Xuemin, "BECAN: A Bandwidth Efficient Cooperative Authentication Scheme for Filtering Injected False Data in Wireless Sensor Networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 23, pp. 32-43, 2012. Article (CrossRef Link)

[4] Xinyu Yang, Jie Lin, Wei Yu, P. Moulema, Xinwen Fu and Wei Zhao, "A Novel En-Route Filtering Scheme Against False Data Injection Attacks in Cyber-Physical Networked Systems," *IEEE Transactions on Computers*, vol. 64, pp. 4-18, 2015. Article (CrossRef Link)

[5] J.N. Al-Karaki and A.E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Communications*, 11, 6-28, 2004. Article (CrossRef Link)
[6] S. M. Nam and T. H. Cho, "A fuzzy rule-based path configuration method for LEAP in sensor networks," *Ad Hoc Networks*, vol. 31, pp. 63-79, 2015. Article (CrossRef Link)
[7] G. Mao, B. Fidan, B.D.O. Anderson, "Wireless sensor network localization techniques," *Elsevier/ACM Computer Networks*, vol.51, pp.2529-2553, 2007. Article (CrossRef Link)
[8] A. Al-Riyami, N. Zhang and J. Keane, "An Adaptive Early Node Compromise Detection Scheme for Hierarchical WSNs," *IEEE Access*, vol. 4, pp. 4183-4206, 2016. Article (CrossRef Link)
[9] W. Dong, J. Yu, J. Wang, X. Zhang, Y. Gao, C. Chen and J. Bu, "Accurate and Robust Time Reconstruction for Deployed Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 24, pp. 2372-2385, 2016. Article (CrossRef Link)
[10] S. Zhu, S. Setia, S. Jajodia, P. Ning, "An Interleaved Hop-by-Hop Authentication Scheme for Filtering of Injected False Data in Sensor Networks," in *Proc. of S&P*, 2004. Article (CrossRef Link)
[11] Fan Ye, Haiyun Luo, Songwu Lu and Lixia Zhang, "Statistical en-route filtering of injected false data in sensor networks," *IEEE J. Selected Areas in Communications*, vol. 23, No. 4, 2005. Article (CrossRef Link)
[12] H.Y. Lee and T.H. Cho, "Fuzzy Adaptive Threshold Determining in the Key Inheritance Based Sensor Networks," *Lecture Notes in Artificial Intelligence, Springer Verlag*, LNAI 4570, pp. 64-73, Jun. 2007. Article (CrossRef Link)
[13] W. Zhang, G. Cao, "Group Rekeying for Filtering False Data in Sensor Networks: A Predistribution and Local Collaboration-Based Approach," in *Proc. of INFOCOM*, 2005. Article (CrossRef Link)
[14] Y. Hu, A. Perrig, D. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks," in *Proc. of INFOCOM*, 2003. Article (CrossRef Link)
[15] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, D,E, Culler, "SPINS: Security Protocols for Sensor Networks," *Wireless Networks*, vol. 8, pp. 521-534, 2002. Article (CrossRef Link)
[16] Z. Yu, Y. Guan, "A Dynamic En-route Scheme for Filtering False Data Injection in Wireless Sensor Networks," in *Proc. of SenSys*, pp. 294-295, 2005. Article (CrossRef Link)
[17] Wallner D, Harder E, Agee R., "Key management for multicast: issues and architectures," *RFC 2627*, 1999. Article (CrossRef Link)
[18] Junqi Zhang, Vijay varadharajan, "Wireless Sensor network key management survey and taxonomy," *Journal of Network and Computer Applications*, vol. 33, pp.63-75, 2010. Article (CrossRef Link)
[19] R.D. Pietro, L.V. Mancini, S. Jajodia, "Providing secrecy in key management protocols for large wireless sensors networks," *Ad Hoc Network*, vol. 1, pp. 455–468, 2003. Article (CrossRef Link)
[20] Carman, D., Kruus, P., Matt. B., "Constraints and Approaches for Distributed Sensor Network Security," *NAI Labs: Technical Report #00-010*, 2000. Article (CrossRef Link)
[21] M. K. Shahzad and T. H. Cho, "Extending the network lifetime by pre-deterministic key distribution in CCEF in wireless sensor networks," *Wireless Networks*, vol. 21, pp. 2799-2809, 2015. Article (CrossRef Link)
[22] F. Li, A. Srinivasan and J. Wu, "PVFS: A Probabilistic Voting based Filtering Scheme in Wireless Sensor Networks," *International Journal of Security and Network*, vol. 3, pp. 173-182, 2008. Article (CrossRef Link)
[23] S. Tanachaiwiwat, P. Dave, R. Bhindwale, and A. Helmy, "Location-centric isolation of misbehavior and trust routing in energy-constrained sensor networks," in *Proc. of IEEE Int'l Conf. on Performance, Computing, and Communications*, pp. 463-469, April, 2004. Article (CrossRef Link)
[24] M. Hollick, C. Nita-Rotaru, P. Papadimitratos, A. Perrig and S. Schmid, "Toward a taxonomy and attacker model for secure routing protocols," *SIGCOMM Comput. Commun. Rev.*, vol. 47, pp. 43-48, 2017. Article (CrossRef Link)

[25] Kumar, D., Aseri, T.C. and Patel, R.B., "EEHC: Energy efficient heterogeneous clustered scheme for wireless sensor networks," *Computer Communications*, vol. 32, pp. 662-667, 2009. Article (CrossRef Link)
[26] Elbhiri, B., Saadane, R. and Aboutajdine, D., "Stochastic Distributed Energy-Efficient Clustering (SDEEC) for heterogeneous wireless sensor networks," *ICGST-CNIR Journal*, vol. 9, pp. 11-17, 2009. Article (CrossRef Link)
[27] C.K. Wong, M. Gouda, S.S. Lam, "Secure group communications using key graphs," *IEEE/ACM Transaction on Networking*, vol.8, pp.16-30, 2000. Article (CrossRef Link)

Jin Myoung Kim received his Ph.D. degree in Computer Engineering from Sungkyunkwan University, Suwon, Korea. He was a research engineer in Seocho R&D Campus, LG electronics. Also he was a Research member of Engineering Staff, Embedded SW Research Department, ETRI. He currently works for the Ministry of National Defense, Korea. His research interests include modeling & simulation, reverse engineering, and sensor network security.

Hae Young Lee received his B.S. degree in Electrical and Computer Engineering and Ph.D. degree in Computer Engineering from Sungkyunkwan University, Suwon, Korea, in 2003 and 2009, respectively. He is currently an Assistant Professor of Digital Security at Cheongju University, Cheongju, Korea. His research interests include secure modeling & simulation, cybersecurity education & training, and sensor network security.