A SECURE ROUTING PROCESS TO SIMULTANEOUSLY DEFEND AGAINST FALSE REPORT AND WORMHOLE ATTACKS IN SENSOR NETWORKS

SOO YOUNG MOON, TAE HO CHO*

College of Information and Communication Engineering, Sungkyunkwan University,
Suwon 440-746, Republic of Korea

ABSTRACT

Most research related to secure routing in sensor networks has focused on how to detect and defend against a single attack. However, it is not feasible to predict which attack will occur in sensor networks. It is possible for multiple attacks to occur simultaneously, degrading the performance of the existing security schemes. For example, an attacker may try simultaneous false report and wormhole attacks to effectively damage a sensor network. Hence, a multiple simultaneous attack environment is much more complex than a single attack environment. Thus, a new security scheme that can detect multiple simultaneous attacks with a high probability and low energy consumption is needed. In this paper, we propose a secure routing scheme to defend against wormhole and false report attacks in sensor networks. The proposed method achieves a higher attack detection ratio and consumes less energy in a multi-attack scenario compared to existing schemes. It can also be extended to other types of attacks and security schemes to detect and defend against possible combinations of multiple attacks.

KEYWORDS

Wireless Sensor Networks, wormhole attacks, false report attacks

1. INTRODUCTION

Sensor networks are systems that collect environmental information from sensors that are attached to objects, provide that information to a user, and react to certain events on demand [1-5]. A sensor network is composed of a large number of tiny sensor nodes that monitor the surrounding environment and one or more sink node(s) that collect information from the sensor nodes. There are very limited available resources such as the energy, bandwidth, and computation capability in a sensor network. Applications of sensor networks vary to include disaster prevention, battlefield monitoring, and U-health [1, 2].

Many security attacks can occur to violate the security objectives, such as the data integrity, confidentiality, and authentication of a sensor network [6, 7]. In false report attacks, an attacker injects forged event information into the network for the purpose of depleting the energy resources of sensor nodes and causing unnecessary user responses. False reports are injected via compromised nodes whose keys are used to generate the false reports. Figure 1 shows an illustration of false report attacks.

DOI : 10.5121/ieij.2014.2403

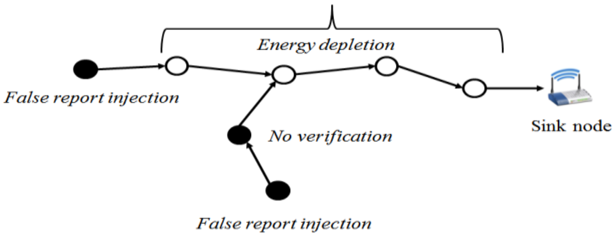


Figure 1 False report attacks

The black nodes within Fig. 1 represent compromised nodes that generate false event reports and inject them into the network. The false reports cause unnecessary energy consumption of intermediate nodes as they are forwarded to the sink node. In addition, a compromised node does not comply with any security protocol; instead, it just forwards a false event report without verifying it.

A wormhole is a physical or virtual link that connects two points in the network and enables two remote nodes to communicate with each other as if they are neighbors. The goal of an attacker in wormhole attacks is to disrupt the communication between the source nodes and the sink node. The attacker may try a message drop, fabrication or modification of messages, or selective forwarding attacks via a wormhole included in the routing paths in the network.

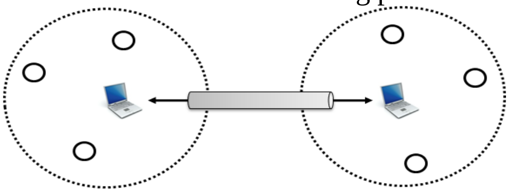


Figure 2 Wormhole attacks

The dotted circle represents the transmission range of the sensor nodes. We assume that the wormhole nodes are powerful devices such as laptops. A wormhole node eavesdrops on messages sent by the nodes within the transmission range and forwards them to the other wormhole node through the wormhole. One of three actions may be taken in a wormhole: 1) message modification, 2) message drop, and 3) message forwarding (no attack).

Security attacks in a sensor network system do not always occur one by one. For example, the above false report attacks and wormhole attacks are two representative attacks in the application layer and network layer, respectively. Attackers may combine the two attacks to damage the network.

In false report filtering schemes, each node detects a false report with a certain probability. As a false report performs more hops, there is a higher probability that the false report will be detected and removed en-route. A wormhole in routing paths decreases the number of hops performed by a false report and hence degrades the en-route detection capability of a filtering scheme.

The sender and receiver nodes of an event report in wormhole countermeasures detect a wormhole between them based on mechanisms such as the message authentication code (MAC) and acknowledgement (ACK) messages. False report attacks involve a node compromise attack, and compromised nodes do not participate in wormhole detection mechanisms. If either a sender or a receiver is a compromised node, it degrades the wormhole detection ratio. Figure 3 illustrates the multi-attack scenario.

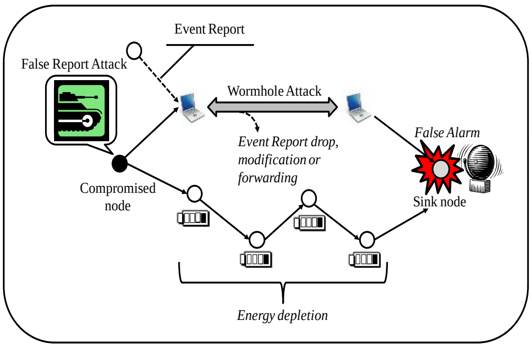


Figure 3 Multi-attack scenario

In Fig. 3, a compromised node generates and injects false reports into the network, and the injected false reports deplete the energy of the nodes on the forwarding path. Wormhole nodes are powerful devices such as laptops, and they forward normal event reports or false reports through the wormhole.

Hence, we need a new security protocol to solve the performance degradation problems that may occur in existing security schemes [8-21] in the multi-attack scenario. We propose a security scheme to effectively detect the false report and wormhole attacks when the two attacks occur simultaneously. The proposed scheme makes use of a key partition-based routing protocol to mitigate the detection probability degradation problem caused by a wormhole. In addition, we define a new format for event report messages that contain the number of verifications by the forwarding nodes and the corresponding wormhole detection mechanisms.

The proposed method is able to detect and remove false event reports within a few hops and can detect a wormhole even when a sender or receiver node that is connected through the wormhole is compromised. Also, the proposed method detects the two attacks with low computation overhead.

The remaining sections of the paper are as follows. In Section 2, we review the false report and wormhole attacks and their countermeasure schemes and confirm the complexity of the multiple-attack environment. In Section 3, we describe the motivation and explain the assumptions and operational phases of the proposed scheme. We show the experimental results about the detection ratio and the energy consumption in Section 4. We conclude the paper and speculate about future work in Section 5

2. BACKGROUND

False report attacks and wormhole attacks are representative attacks at the application and the network layer, respectively. In this section, we describe the characteristics of the two attacks and existing countermeasures to be used against them. In addition, we explain the problem of the performance degradation of existing schemes under a multi-attack scenario.

2.1. False report attacks & countermeasures

The objective of false report attacks is to deplete the energy resources of nodes that forward the false event reports, leading to unnecessary responses on the part of the user. In false report attacks, an attacker gains control of compromised nodes. Hence, false report attacks are insider attacks in which the attacker makes use of information and the resources of authorized sensor nodes. The attacker injects false reports about non-existent events via the compromised nodes into the network. The injected false reports are forwarded to the sink node via intermediate nodes. Thus, they deplete the energy of the forwarding nodes and waste the user's time in useless responses [8].

Many security schemes have been proposed [8, 10-14, 20, 21] to defend against false report attacks. The Statistical En-route Filtering (SEF) scheme [8] proposed by Ye et al. aims to detect and remove false reports early in their phases. The sink node in SEF manages a set of authentication keys in a global key pool. The global key pool is divided into several key partitions, and each sensor node is assigned a portion of a key partition before deployment. Sensor nodes collaboratively authenticate event information using their keys. When an interesting event occurs, the sensing nodes elect a center of stimulus (CoS) node. The CoS broadcasts an initial report and collects message authentication codes (MACs) from the other sensing nodes. Then, it generates the final event report that contains the event information, the MACs, and the corresponding key indices. Next, the CoS sends the event report to its parent node. When a sensor node receives an event report, it checks to see if it has the same key as one of the keys that were used to generate the MACs in the report. If so, it verifies the corresponding MAC and determines whether or not to forward it based on the verification result. When the sink node receives the event report, it verifies all of the MACs in the report, since it knows all of the keys in the global key pool. If any of the verification results is false, it detects a false report and drops it. Figure 4 shows the filtering operation in SEF.

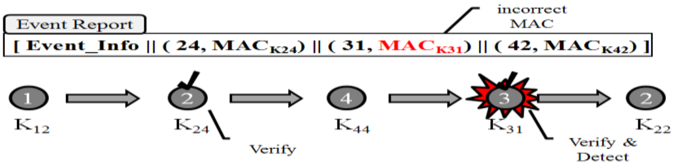


Figure 4. SEF filtering operation

In Fig. 4, each circle represents a sensor node that is assigned an authentication key (the number below each circle) from a key partition (the number in each circle). The event report contains three MACs and the corresponding key indices in addition to the event information. A key index represents both the key partition and the order of the key in the partition. For example, K_{24} in the figure denotes the fourth key in the second key partition. MAC_{K24} denotes the MAC generated using K_{24} from the event information. MAC_{K31} is the incorrect one among the three MACs, since K_{31} was not disclosed to the attacker. When the second node on the path receives the event report, it verifies MAC_{K24}. It forwards the event report to the next node, since the MAC is correct. When the fourth node on the path receives the event report, the node attempts to verify MAC_{K31}. The verification fails at the node, since the MAC is an incorrect MAC. Thus, the node drops the event report.

2.2. Wormhole attacks & countermeasures

The objective of an attacker in wormhole attacks [9] is to disconnect a network and to interrupt the communication between the source nodes and the sink node. For this, the attacker tries to include a wormhole, which is a physical or logical link, in the routing paths in the network. We assume that, in the proposed method, the wormhole nodes are powerful devices such as laptops. Hence, wormhole attacks are outsider attacks in which the attacker has no access to the network resources and a laptop-class attack in which a long transmission range and sufficient energy resources are available [7]. Wormhole attacks may occur in a neighbor discovery phase or a data dissemination phase. A wormhole node eavesdrops control or data messages and then forwards them to the other wormhole node through the wormhole. In a wormhole, one of three actions may occur: 1) message drop, 2) message modification, and 3) message forwarding. Wormhole attacks disrupt reliable communication between the source nodes and the sink node and hence degrade the availability of the network.

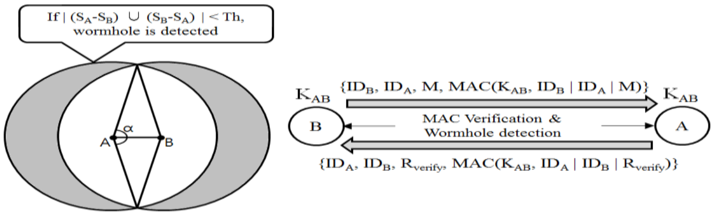
Many security schemes have been proposed [9, 15-19] to defend against wormhole attacks, and among them, we explain two security schemes: 1) the secure routing protocol against wormhole attacks (SeRWA) [17] and 2) the localized encryption and lightweight protocol (LEAP) [18].

A secure routing protocol against wormhole attacks (SeRWA) [17] was proposed by Madria and Yin. SeRWA is able to detect a wormhole and update routing paths without requiring special hardware.

There are four phases in SeRWA: 1) one-hop neighbor discovery, 2) initial route discovery, 3) data dissemination and wormhole detection, and 4) secure route discovery against a wormhole attack. In the one-hop neighbor discovery phase, sensor nodes that are within radio range discover and add each other to their neighbor node lists by exchanging hello and reply messages. Each node also exchanges its neighbor node list with its neighbor nodes. Then, the initial wormhole detection is performed based on the neighbors' neighbor node lists. If a wormhole is detected, the neighbor list is reconstructed. At the end of the phase, the neighbor nodes share the message authentication keys.

In the initial route discovery phase, the sink node broadcasts a route request message. Each node sets the first node from which it receives a route request message as its parent node and then re-broadcasts the message. At the end of the phase, a routing tree is constructed.

In the data dissemination and wormhole detection phase, the source nodes forward event reports to the sink node through the routing paths. The integrity of the event reports is verified in a hop-by-hop manner by using the shared key between the immediate sender and immediate receiver. If a message drop or message modification is detected, a wormhole is detected. Figure 5 represents the wormhole detection in SeRWA at the one-hop neighbor discovery phase and the data dissemination phase.


If | $(S_A\text{-}S_B)$ ∪ $(S_B\text{-}S_A)$ | < Th, wormhole is detected
α
A
B
K_{AB}
{ID_B, ID_A, M, MAC(K_{AB}, ID_B | ID_A | M)}
K_{AB}
B
MAC Verification & Wormhole detection
A
{ID_A, ID_B, R_{verify}, MAC(K_{AB}, ID_A | ID_B | R_{verify})}

(a) Initial wormhole detection [16]

(b) Wormhole detection in data dissemination phase

Figure 5. Wormhole detection in SeRWA

Figure 5(a) illustrates wormhole detection in the one-hop neighbor discovery phase in SeRWA. Nodes A and B are neighbor nodes, and the two circles represent their transmission ranges. S_A is the set of node A and its neighbor list, while S_B is the set of node B and its neighbor list. $|(S_A - S_B) \cup (S_B - S_A)|$ is the number of nodes within either S_A or S_B but not both. If the value is less than a threshold value *Th*, a wormhole is detected. The threshold value *Th* is determined by the maximum distance between two 'close nodes,' the transmission range of the sensor nodes, and the node density in the network.

Figure 5(b) shows the wormhole detection in the data dissemination phase. We assume that nodes A and B are neighbor nodes of each other, and that node A is the parent node of node B. Each data message transmitted from node B to node A includes a MAC that is generated using a pairwise key that is shared between the two nodes. Node A verifies the MAC and notifies node B of the verification result by replying back with an ACK message. The ACK message also includes a MAC generated using the pairwise key, and hence, node B is able to verify the ACK message. If a data message is illegally modified, a wormhole is detected at node A and node B.

Otherwise, if a data message is dropped, a wormhole is detected at node B, which is the sending node. If a wormhole is detected, the neighbor list reconstruction is performed.

LEAP [18] is a key management scheme that defines the establishment, deployment, and update of keys. It also provides an inter-node authentication mechanism. Each node in the scheme organizes its own One-way Hash Chain (OHC), the elements of which are authentication keys. When a node sends a packet to its neighbor node, it selects the next authentication key in its own OHC and attaches the key to the packet. The receiving node is able to verify the authentication key in the packet based on the last authentication key received from the sending node. In this way, LEAP can detect wormhole or sinkhole attacks.

2.3. Environment of multiple simultaneous attacks

As we mentioned in the previous section, the existing countermeasures for false report and wormhole attacks suffer from performance degradation if the two attacks occur simultaneously. Figure 6 shows the performance degradation of SEF in a multiple attack environment.

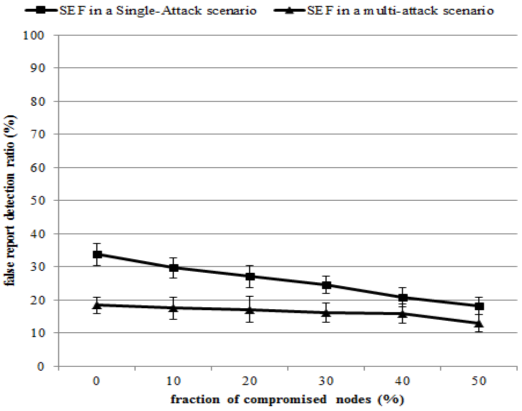


Figure 6. Performance degradation of SEF under a multiple attack environment

(false report attack perspective)

In the figure, the false report detection ratios decrease as the fraction of compromised nodes increases, under both the single-attack and multi-attack scenarios. In addition, the false report detection ratio of SEF in the multi-attack scenario is lower than that in the single-attack scenario. The reason is that a wormhole reduces the number of hops through which the false reports pass. Figure 7 shows the performance degradation of SeRWA in the multiple attack scenario.

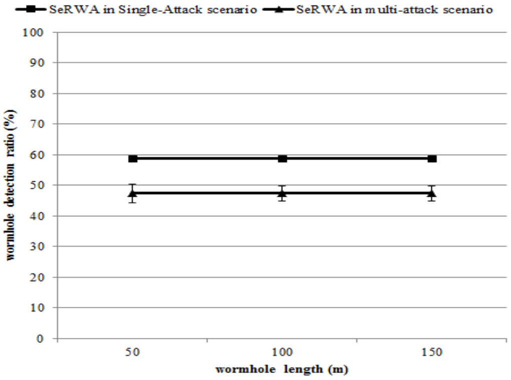


Figure 7. Performance degradation of SeRWA under a multiple attack environment

(wormhole attack perspective)

Figure 7 compares the wormhole detection ratios of SeRWA under single-attack and multi-attack scenarios. As shown in the figure, the wormhole detection ratio of SeRWA in the multi-attack scenario is lower than that in the single-attack scenario. That is because the compromised nodes in the multi-attack scenario do not comply with the wormhole detection processes such as MAC verification or the generation and exchange of ACK messages.

As shown in the above results, the performance of the existing security schemes decreases when there are multiple attacks. Thus, a new security protocol that can handle a combination of multiple attacks is needed.

3. Proposed scheme

The proposed scheme aims to detect false report and wormhole attacks under a multi-attack scenario. The two main goals are 1) to detect false reports early even when there is a wormhole in the network and 2) to detect a wormhole even when one of the two nodes connected via the wormhole is compromised. We explain the advantages of the proposed scheme compared to existing schemes in this section.

3.1 Motivation

As mentioned in Section 2, the existing security schemes suffer performance degradation in a multi-attack scenario. We can improve the detection ratio of false event reports by exploiting a key partition-based routing and can mitigate the performance degradation of the existing filtering schemes due to a wormhole. In addition, we can enhance the detection ratio of a wormhole based on the number of verifications for an event report by the forwarding nodes, even when one of the two nodes that are connected through a wormhole is compromised.

3.2 Assumptions

We assume a static sensor network in which the sensor nodes do not change their locations. Each node can obtain its location information. The sensor nodes have the same fixed transmission range.

We assume that there is no attack until the data dissemination phase, and we focus on false report injection and wormhole attacks in the data dissemination phase. There are two types of attack nodes in the network: 1) compromised nodes and 2) wormhole nodes. Compromised nodes are authorized nodes that are controlled by an attacker to perform false report injection attacks. 2) Wormhole nodes are laptop-level devices that are owned by the attacker and are equipped with a long transmission range and sufficient energy resources.

False reports are injected through compromised nodes to the sink node along the intermediate nodes. A wormhole node eavesdrops on neighbor nodes and forwards event reports to the other wormhole node through the wormhole. In a wormhole, one of three actions may occur: 1) message drop, 2) message modification, and 3) message forwarding.

We assume that the compromised nodes do not comply with security protocols and hence, do not detect false reports or a wormhole. However, we assume that there is no additional cooperation between the compromised nodes and the wormhole nodes.

3.2 Operation

The specific description of the operation of the proposed scheme is as follows. In the pre-deployment phase, the assignment of unique IDs and authentication keys is performed, as in

SEF. Additionally, two empty lists are created and initialized at each node. One is a neighbor list, which will contain a list of the other nodes within its transmission range, and the other is a candidate parent list, which will contain a list of the neighbor nodes that are closer to the sink node than the current node.

In the neighbor discovery phase, every sensor node broadcasts a hello message to the nodes within its transmission range and waits to receive reply messages from them. Each hello message contains the sender's ID, whereas each response message includes the sender's and the receiver's IDs. When a sensor node receives a hello message, it creates and sends back a response message to the sender of the hello message. When the sender receives the response message, it adds the receiver's ID to its neighbor list if the sender's ID is the same as that in the response message. Figure 8 shows the neighbor discovery process.

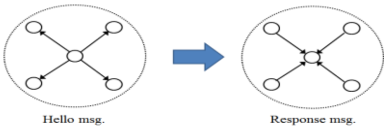


Figure 8. Neighbor discovery of sensor nodes

In the route setup phase, each node organizes its own candidate parent list, which contains the information of its neighbor nodes that are closer to the sink node. That is, a node can choose one of the nodes in its candidate parent list as the next forwarding node on the routing paths for event reports. The sink node initializes the route setup phase by broadcasting a control message. A control message contains 1) the ID of the sender, 2) the distance of the sender to the sink node, and 3) the key partition associated with the sender. A receiving node of the control message inserts the record (sender's ID, distance to the sink, key partition) to its candidate parent list if the sender is closer to the sink node than itself. Every pair of neighbor nodes establishes a pairwise key at the end of the phase [18]. Figure 9 shows the route setup process.

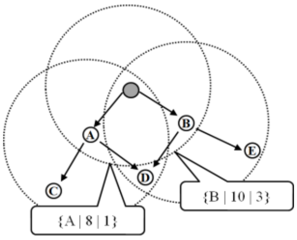


Figure 9 Flooding of control messages in the route setup phase

In the figure, node D receives a control message from nodes A and B, respectively. Node A is 8m from the sink node and is associated with the first key partition, whereas node B is 10m from the sink node and is associated with the third key partition. Both of the two nodes are closer to the sink node than node D, so node D organizes its candidate parent list as {(A, 8, 1), (B, 10, 3)}.

In the data dissemination phase, the sensor nodes detect events and report them to the sink node. In addition, they detect false reports and (or) a wormhole based on the content of the event reports and the ACK messages that have been forwarded. Figure 10 shows the format of an event report and an ACK message in the proposed scheme.

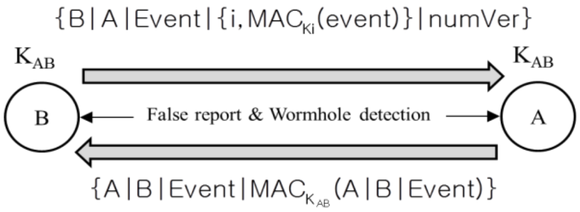

{B|A|Event|{i, MAC_Ki(event)}|numVer}
K_AB
K_AB
B
False report & Wormhole detection
A
{A|B|Event|MAC_KAB(A|B|Event)}

Figure 10 Event report and ACK msg. format

In Figure 10, nodes A and B are neighbors of each other, and node A is the next node after node B on the routing path. An event report contains the sender's ID (i.e., B), the receiver's ID (i.e., A), the event information, a list of (key index, MAC) pairs that have been generated by using the global authentication keys, and the number of verification operations performed by the forwarding nodes, which includes node B.

Node A receives the event report r and replies with an ACK message if no wormhole is detected. An ACK message includes the receiver's ID, the sender's ID, event information, and a MAC that is generated by using the pairwise key shared between nodes A and B. Figure 11 shows the operation of a receiving node and the detection process of the two attacks.

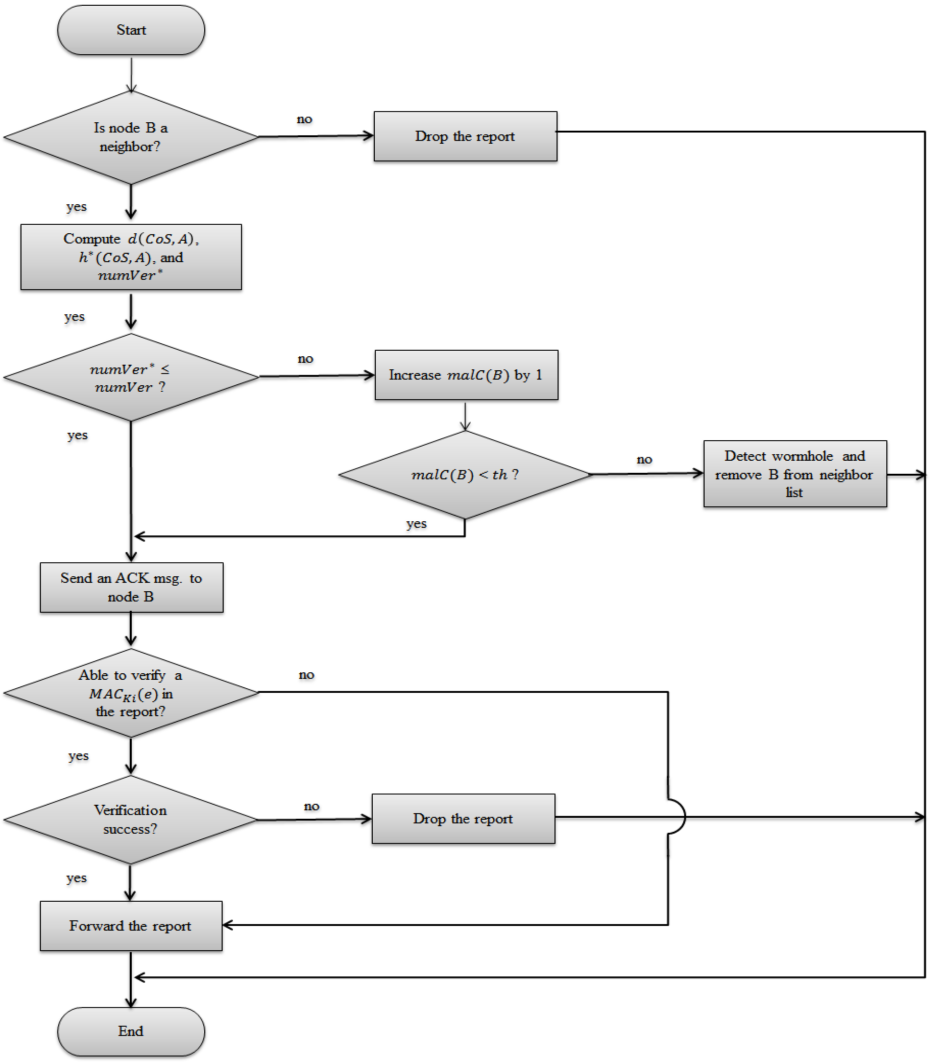


Figure 11. Wormhole and false report detection at the receiving node A

The process of attack detection at a forwarding node is as follows:

1) The receiving node A checks to see if the ID of the sending node B is in its neighbor list, and if not, it drops the report.
2) Node A computes the distance d(CoS, A) between the CoS and itself. The computation is based on node A's location information and the event information in the report.
3) Node A estimates the hop counts from the CoS to itself by dividing the distance by the transmission range of the nodes. The estimated hop count is:

   $$h^*(CoS, A) = d(CoS, A) / TRANS\_RANGE$$

   where TRANS_RANGE is the transmission range of the sensor nodes.
4) The receiving node A estimates the number of verification operations for the received event report performed by the former nodes of the current node on the forwarding path. The estimated number of verifications is computed as follows:

   $$numVer^* = h^*(CoS, A) * OVP$$

   where OVP is the probability of being able to verify a MAC in the event report at a randomly selected node.
5) If the actual number of verifications (*numVer*) for the received event report is less than *numVer**, the malicious count of node B (*malC(B)*) is increased by one.
6) If *malC(B)* becomes greater than or equal to a threshold value *th*, a wormhole is detected.
7) If a wormhole is not detected, node B replies to node A with an ACK message.
8) If node A has one of the MAC-generating keys for the received report, it verifies the corresponding MAC using its own authentication key. It forwards the report to the next node only when the verification succeeds.

After that attack detection, the current node selects one of its candidate parent nodes as the next node to forward the event report.

It chooses one of the candidate parent nodes based on an evaluation function f(n) as follows:

$$f(n) = \alpha \cdot \frac{1}{rank(n)} + \beta \cdot v(n) \qquad (1)$$

$$(0 \le \alpha, \beta \le 1, \alpha + \beta = 1)$$

In equation (1), n is a candidate parent node, and rank(n) is the order of n when we sort the candidate parent nodes by the distance to the sink in increasing order. The rank(n) has a positive integer value (1, 2, 3, …), and v(n) is the probability of verifying a MAC in the forwarded report at node n when it is selected as the next node.

The current node can compute the v(n) of each candidate parent node based on the candidate parent list stored in its memory.

First, it computes the key partitions of MAC-generating keys from their indexes in the event report. Then, it compares each of them with the key partition of node n. If the key partition of node n is the same as one of the key partitions of the MAC-generating keys, v(n) can be computed as the number of keys stored at node n divided by the total number of keys in the key partition. Otherwise, if the key partition of node n does not match any of the key partitions of the MAC-generating keys, v(n) becomes 0. Equation (2) shows the values of v(n) for the two cases.

$$v(n) = \begin{cases} \dfrac{k}{m} & : \text{key partition of } n \text{ matches with one of MAC generating keys} \\ 0 & : else \end{cases} \qquad (2)$$

In the above equation, *k* is the number of keys in the key partition stored at node n, and *m* is the total number of keys in the matching key partition. The parameters α and β are weights for the two factors. As α increases, the priority of the nodes close to the sink node increases. On the other hand, if β increases, the priority of nodes with a high probability of verifying a MAC increases. Figure 12 illustrates the selection of the next node under different values of α and β.

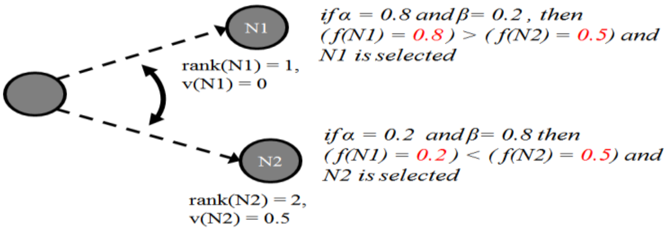

N1
rank(N1) = 1,
v(N1) = 0
N2
rank(N2) = 2,
v(N2) = 0.5
if α = 0.8 and β= 0.2 , then
(f(N1) = 0.8) > (f(N2) = 0.5) and
N1 is selected
if α = 0.2 and β= 0.8 then
(f(N1) = 0.2) < (f(N2) = 0.5) and
N2 is selected

Figure 12. Selection of a parent node at a forwarding node

The two nodes N1 and N2 are candidate parent nodes of the sending node (on the left in the figure). If we sort the candidate parent list of the sending node based on the distances to the sink node in increasing order, the orders of N1 and N2 are 1 and 2, respectively. Hence, rank(N1) = 1 and rank(N2)=2. The key partition associated with N1 is not the same as any of the key partitions of the MAC-generating keys for the report, so v(N1) = 0. On the other hand, the key partition associated with N2 is the same as one of the MAC-generating keys for the report, so, if we assume that k=5 and m=10, v(N2)=0.5.

If we use 0.8 for α and 0.2 for β, f(N1) = 0.8*1 + 0.2*0 = 0.8, and f(N2) = 0.8*1/2+0.2*0.5=0.5. Hence, f(N1) > f(N2), and N1 is selected by the current node as the next forwarding node. In another case, if we use 0.2 for α and 0.8 for β, f(N1) = 0.2*1 + 0.8*0 = 0.2 and f(N2) = 0.2*(1/2) + 0.8*0.5=0.5. Hence, f(N1) < f(N2), and N2 is selected by the current node as the next forwarding node.

4. PERFORMANCE EVALUATION

4.1 SECURITY ANALYSIS

False report injection attacks involve a node compromise attack. The attackers can take information from and control of the compromised nodes. The assumptions about compromised nodes are as follows. A compromised node performs no activities related to the attack detection; instead, it just performs essential operations such as message transmission and reception. On the other hand, we assume no cooperation between the compromised nodes and the wormhole attack nodes.

The following figure shows the process of message forwarding through a routing path that contains a wormhole.

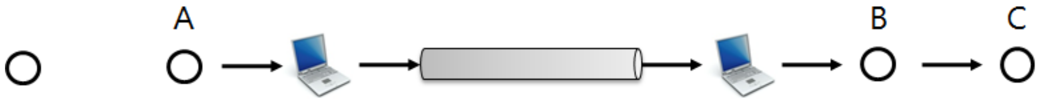


Figure 13. A routing path that contains a wormhole

In the figure, nodes A and B are two nodes that are connected by a wormhole. Node A sends a message to node B, and the message is forwarded to a wormhole, node B, and node C in that order. In this scenario, four cases may exist: 1) neither node A nor node B is compromised, 2) node A is compromised, 3) node B is compromised, and 4) both nodes A and B are compromised. For each of the four cases, three sub-cases exist: A) a message modification occurs in a wormhole, B) a message drop occurs in a wormhole, and C) message forwarding occurs in a wormhole.

Case 1: Neither A nor B is compromised

In this case, neither node A nor node B is compromised by an attacker. In sub-cases A) and C), node B computes the distance between the CoS and itself. It then estimates the number of hop counts traveled and the number of verifications of the received event report. Since the actual number of verifications in the report is less than the estimated value, node B increases the malicious count of node A by one. If the malicious count of node A reaches a threshold value, node B detects a wormhole, removes node A from its neighbor list, and does not send an ACK message to node A. Then, node A does not receive a corresponding ACK message within a timeout period, so it also detects a wormhole and removes node B from its neighbor list. Note that the proposed method can detect a wormhole even when an event report is forwarded without modification through the wormhole, because each node detects a wormhole by checking the number of verifications of the received event report, not the MACs in the report.

If a message drop occurs in the wormhole, node B cannot receive the event report. In this case, only node A can detect a wormhole because it does not receive a corresponding ACK message within a timeout period.

Case 2: Node A is compromised

If node A is compromised, it does not comply with the proposed method. Similar to case 1), if message modification or message forwarding occurs in a wormhole, node B compares the number of verifications in the event report and the estimated value and increases the malicious count of node A. If the malicious count of node A reaches a threshold value, node B detects a wormhole and removes node A from its neighbor list.

However, node A does not detect the wormhole since it is compromised. The proposed method does not include compromised node detection and eviction mechanisms. If a message drop occurs in a wormhole, the wormhole is not detected in this case.

Case 3: Node B is compromised

Node B is compromised and does not follow the proposed method. If message modification or message forwarding occurs in a wormhole, node B does not detect a wormhole and forwards an event report to node C. Node C computes the distance between itself and the CoS and estimates the number of hop counts and the number of verifications of the event report. Since the actual number of verifications in the report is less than the estimated number, node C increases the malicious count of node B. If the malicious count of node B reaches a threshold value, node C detects a wormhole and removes node B from its neighbor list. If a message drop occurs in a wormhole, node A does not receive an ACK message within the timeout period, detects a wormhole and removes node B from its neighbor list.

Case 4: Both A and B are compromised

In the case, both nodes A and B are compromised nodes. If message modification or message forwarding occurs in a wormhole, an event message is forwarded to node C. Node C compares the actual number of verifications of the received report and the estimated number and increases the malicious count of node B by one. If the malicious count of node B reaches a threshold value, node C detects a wormhole and removes node B from its neighbor list. If a message drop occurs in a wormhole, the wormhole is not detected. The following table compares the wormhole detection capabilities of SeRWA, LEAP, and the proposed method.

Table 1 Wormhole detection capabilities of the existing schemes and the proposed method

Case	SeRWA	LEAP	P.M
Case 1: No compromise	1) Message modification ⇨ Node A and B detect a wormhole 2) Message drop ⇨ Node A detects a wormhole 3) Message forwarding (no attack) ⇨ no detection	1) Message modification ⇨ Node B detects a wormhole 2) Message drop ⇨ no detection 3) Message forwarding (no attack) ⇨ no detection	1) Message modification ⇨ Node A and B detect a wormhole 2) Message drop ⇨ Node A detects a wormhole **3) Message forwarding (no attack)** ⇨ Node A and B detects a wormhole
Case 2: Node A is compromised	1) Message modification ⇨ Node B detects a wormhole 2) Message drop ⇨ No detection 3) Message forwarding (no attack) ⇨ No detection	1) Message modification ⇨ Node B detects a wormhole 2) Message drop ⇨ No detection 3) Message forwarding (no attack) ⇨ No detection	1) Message modification ⇨ Node B detects a wormhole 2) Message drop ⇨ No detection **3) Message forwarding (no attack)** ⇨ Node B detects a wormhole
Case 3: Node B is compromised	1) Message modification ⇨ No detection 2) Message drop	1) Message modification ⇨ No detection 2) Message drop	**1) Message modification** ⇨ Node C detects a wormhole

	⇨ Node A detects a wormhole 3) Message forwarding (no attack) ⇨ No detection	⇨ No detection 3) Message forwarding (no attack) ⇨ No detection	2) Message drop ⇨ Node A detects a wormhole **3) Message forwarding (no attack)** ⇨ Node C detects a wormhole
Case 4: Nodes A and B are compromised	1) Message modification ⇨ No detection 2) Message drop ⇨ No detection 3) Message forwarding (no attack) ⇨ No detection	1) Message modification ⇨ No detection 2) Message drop ⇨ No detection 3) Message forwarding (no attack) ⇨ No detection	**1) Message modification** ⇨ Node C detects a wormhole 2) Message drop ⇨ No detection **3) Message forwarding (no attack)** ⇨ Node C detects a wormhole

There are 12 total situations, considering the four cases with three sub-cases each. SeRWA and LEAP can detect a wormhole in four and two of those situations, respectively, and the proposed method can detect a wormhole in ten situations. If we assume that the threshold value is one and that all of the situations occur with the same probability, the probability that the proposed method will detect a wormhole attack is 10/12, whereas the probabilities that SeRWA and LEAP would detect a wormhole attack are 4/12 and 2/12, respectively.

In en-route filtering schemes, the routing paths strongly affect the detection probability of false event reports, since each node has only a small portion of the global key pool, and its detecting capability is limited.

In the proposed method, every node chooses the next node to forward an event report based on the key partitions of its candidate parent nodes. Hence, the filtering probability is higher than that of existing schemes such as SEF.

The following equation represents the filtering probability of false event reports at each node in the proposed method.

$$P_1' = \left(1 - \left(1 - \frac{T}{\mathrm{n}}\right)^{nc}\right) \cdot \frac{k}{m} \cdot \frac{(T-N_c)}{T} \tag{3}$$

In the above equation, n is the number of key partitions in a global key pool, T is the number of MACs in each event report, m is the number of keys in each key partition, k is the number of keys loaded into each node, and N_C is the number of compromised partitions with disclosed keys. *nc* is the average number of candidate parent nodes for each node (i.e., the neighbor nodes that are closer to the sink node than the current node).

Table 2 shows that the probability that a receiving node in the proposed method detects a false report increases as *nc* increases.

Table 2 filtering probabilities of SEF and the proposed method

nc	n	m	k	T	N_C	P1	P1'
1	10	10	4	5	1	0.16	0.16
2	10	10	4	5	1	0.16	0.24
3	10	10	4	5	1	0.16	0.28
4	10	10	4	5	1	0.16	0.30
5	10	10	4	5	1	0.16	0.31

For example, if nc = 5, n = 10, m = 10, k = 4, T = 5, and N_C = 1, then the filtering probability of the proposed method is 0.31, whereas the filtering probability of SEF is 0.16.

4.2 EXPERIMENTS

We obtained experimental results for the existing schemes and for the proposed method under single attack and multi-attack scenarios. The main performance measures are the attack detection ratios and energy consumption. The parameter values used in our experiments are as follows.

The total number of sensor nodes is 1000, and the sensor nodes are randomly deployed in a 500 × 500 m^2 field based on uniform distribution. The transmission range of the nodes is 50m and is the same for each node. The energy consumption for transmission / reception of a single byte is 16.25/12.5 μJ, and we assume that the length of each event message is 36 bytes [22-24]. The energy consumption for one verification operation of a MAC is 75 μJ. There are also parameters that are related to false report or wormhole attacks. The percentage of compromised nodes is within the range of 0 ~ 50%. Each false report contains four valid MACs and one incorrect MAC. The length of a wormhole is between 50m and 150m.

Figure 13 shows the false report detection ratios of SEF and the proposed method as the fraction of compromised nodes increases under a false report attack. Figure 13 and all of the following figures represent the sample means and the 95% confidence interval of the y-axis values.

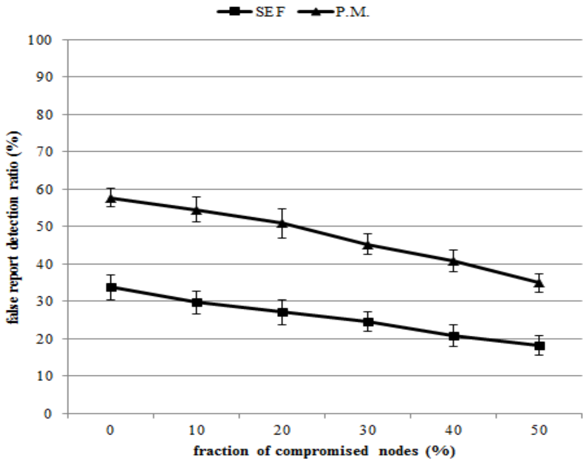


Figure 13 Attack detection ratio under false report attack (single attack)

We can see that the proposed method achieves a much higher false report detection ratio than SEF. The proposed method detects 58% of the false reports when the fraction of compromised nodes is zero, while SEF detects 34% of the false reports. The reason is that each node chooses the next forwarding nodes based on the key partition information and the distance to the sink of its neighbor nodes.

The detection ratios for the two schemes decrease as the fraction of compromised nodes increases. However, the proposed method still shows a higher detection ratio (35%) than SEF (18%), even when the fraction of compromised nodes is 50%. Figure 14 compares the energy consumption of SEF and the proposed method under a false report attack.

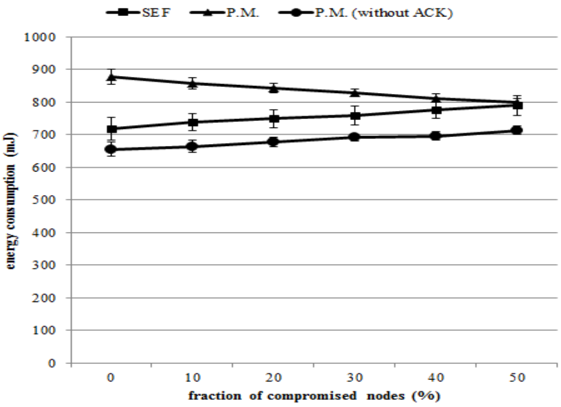


Figure 14 Energy consumption under false report attack (single attack)

The energy consumption of the proposed method is higher than that of SEF under a false report attack because the nodes in the proposed method exchange ACK messages with each other. The energy consumption for ACK messages decreases as the fraction of compromised nodes increases, since compromised nodes do not generate and send ACK messages. If no ACK message is used in the proposed method (‘P.M. (without ACK)’ in the figure), the proposed method consumes less energy than SEF. For example, the proposed method without ACK consumes 89% as much energy as SEF when the fraction of compromised nodes is 50%. We can selectively turn off the functions related to the ACK messages to save energy in the proposed method if the false report attack is the only attack in the field.

Figure 15 shows the wormhole detection ratios of SeRWA, LEAP, and the proposed method under a wormhole attack.

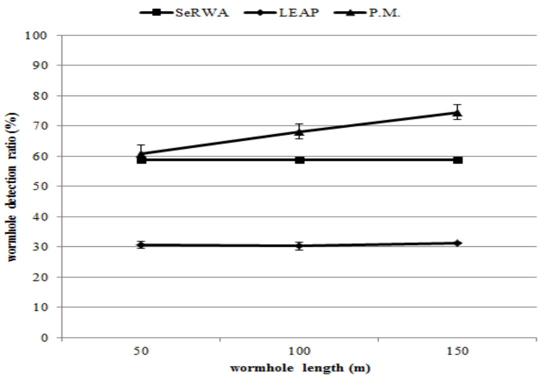


Figure 15 Attack detection ratio under wormhole attack (single attack)

The wormhole detection ratio of the proposed method increases as the length of the wormhole increases, since the proposed method detects a wormhole based on the number of MAC verifications performed by the forwarding nodes of the event messages. The proposed method achieves detection ratios of 61%, 68%, and 75% when the wormhole lengths are 50, 100, and

150m, respectively. On the other hand, SeRWA and LEAP show relatively constant detection ratios (59% and 31%, respectively) for increasing values of wormhole length. Figure 16 illustrates the energy consumptions of SeRWA, LEAP, and the proposed method under a wormhole attack.

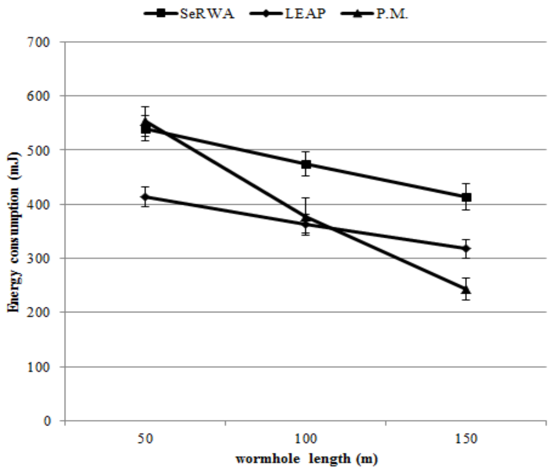


Figure 16 Energy consumption under wormhole attack (single attack)

The energy consumption of all three of the schemes tends to decrease as the wormhole length increases. This is because the forwarding paths for the event messages are shortened due to the wormhole, so the event messages travel fewer hops. In addition to that, the proposed method saves energy since it detects more wormholes and drops the corresponding event messages earlier than the existing schemes. Figure 17 represents the false report detection ratios of the combined scheme of the two existing schemes (SEF+SeRWA) and the proposed method under a multi-attack (false report and wormhole attacks) scenario.

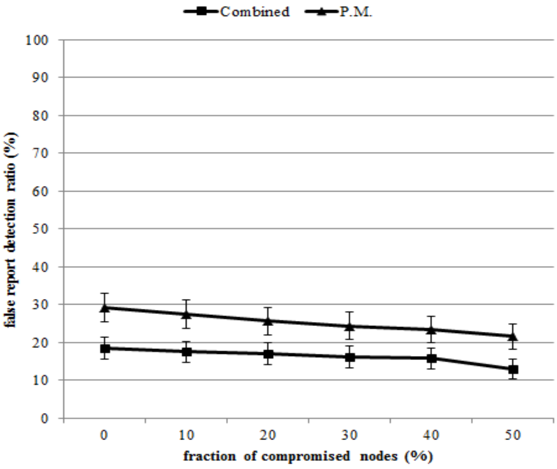


Figure 17 False report detection ratio under multi-attack (false report and wormhole attack)

The false report detection ratios of the combined scheme and the proposed method under multi-attack are less than those for a single attack (i.e., a false report attack), since the number of hops that the event messages are forwarded decreases due to the wormhole. They also decrease as the fraction of compromised nodes increases. Still, the proposed method shows a higher false report detection ratio than the combined scheme since it chooses the forwarding paths of the event messages based on the key partitions of the nodes. For example, the proposed method detects 22% false reports, while the combined scheme detects 13% false reports.

Figure 18 compares the wormhole detection ratios of the combined scheme and the proposed method under the multi-attack scenario.

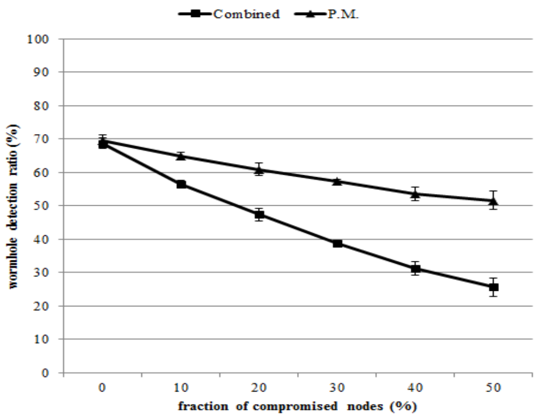


Figure 18 Wormhole detection ratio under multi-attack (false report and wormhole attack)

When there is no compromised node, the combined scheme shows a comparable wormhole detection ratio compared to the proposed method. The wormhole detection ratios of the two schemes degrade as the fraction of compromised nodes increases. However, the detection ratio of the combined scheme is far lower than the proposed method, whereas the proposed method shows graceful degradation. Note that when the fraction of compromised nodes is 30%, the proposed method detects 57% of the wormholes, which is higher than the 39% detected by SEF.

Figure 19 compares the energy consumption of the proposed method and the combined scheme under the multi-attack scenario.

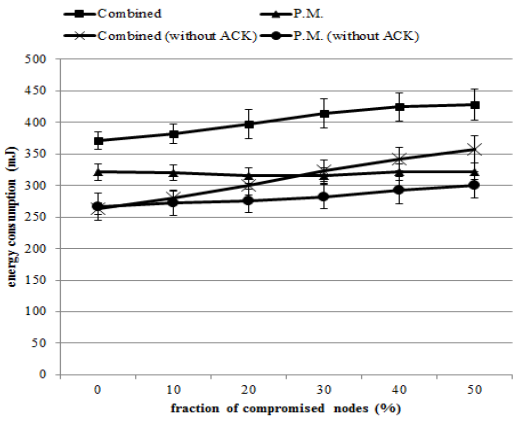


Figure 19 Energy consumption under multi-attack (false report and wormhole attack)

The result shows that the energy consumption (without ACK) of the combined scheme and the proposed method increases as the fraction of compromised nodes increases. In addition, the curve of the combined scheme is higher than that of the proposed method. When the fraction of compromised nodes is 50%, the energy consumption of the proposed method is 75% of that of the combined scheme.

The energy consumption of the combined scheme (with ACK) increases as the fraction of compromised nodes increases. On the other hand, the energy consumption of the proposed method shows relatively constant energy consumption. The reason is that the energy consumption for the ACK messages decreases for both of the two schemes as the fraction of the compromised nodes increases.

In summary, the proposed method achieves a higher attack detection ratio and less energy consumption compared to the combined scheme in a multi-attack (false report + wormhole) scenario.

4.3 DEFENDING AGAINST OTHER ATTACKS

There are many attacks other than the false report and wormhole attacks, and we can use the proposed method to defend against some of them.

The hello flood attack is one in which an attacker broadcasts a Hello message with high transmission power to advertise the attack node to most of the nodes in the network. The nodes that receive the Hello message assume that the attack node is one of their neighbors. However, most of them are too far from the attack node, and the messages sent by the nodes to the attack node are lost due to signal attenuation, and the energy used for transmitting the messages is wasted. The goal of the attacker in a Hello flood attack is to cause message loss due to signal attenuation, thereby wasting energy. In ACK spoofing attacks, an attack node eavesdrops messages and transmits forged ACK messages to cause other nodes to keep using a weak or dead link. The attack disrupts the exchange of messages among the nodes and harms the availability of the network.

The proposed method is able to defend against the above two attacks. In the proposed method, each node sends a hello message and receives the corresponding response messages from its neighbor nodes. Each node adds to its neighbor list only the nodes from which it receives response messages. This guarantees that the two links of the opposite directions between any two neighbor nodes are available (i.e., they are able to share a message). Even if an attack node transmits a hello message with high transmission power, the receiving nodes do not add the attack node to their neighbor lists. Hence, the proposed method can prevent hello flood attacks.

In addition, the proposed method authenticates messages that are transmitted between two neighbor nodes based on the pairwise key. An attacker can forge only the ACK messages from the compromised nodes but not the ACK messages from the other nodes. Hence, the proposed method mitigates the damage because of the ACK spoofing attack.

5. CONCLUSIONS

Most research related to secure routing in sensor networks has focused on how to detect and defend against a single attack. However, if multiple attacks occur simultaneously, the existing security schemes suffer performance degradation. For example, false report and wormhole attacks can both be applied to effectively damage a sensor network. In this paper, we proposed a secure routing scheme to defend against wormhole and false report attacks in sensor networks. The proposed scheme achieves a higher attack detection ratio than the existing schemes and their combined scheme, which is executed by modifying the two security schemes SEF and SeRWA so that they share data. The proposed scheme can be extended to defend against other combinations of multiple attacks.

ACKNOWLEDGEMENTS

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-2013R1A2A2A01013971)

REFERENCES

[1] Xu, N. (2002). A Survey of Sensor Network Applications. Tech. Rep., University of Southern California.

[2] J. A. Stankovic, "When Sensor and Actuator Networks Cover the World," ETRI Journal, vol. 30, no. 5, 2008.

[3] Akyildiz, I. F., and Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). A Survey on Sensor Networks. IEEE Communications Magazine, 40(8), 102-116.

[4] Al-Karaki , J.N. , Kamal , A. E. (2004). Routing techniques in wireless sensor networks: a survey. IEEE Wireless Communication Magazine, 11(6), 6-28.

[5] Baronti, P., Pillai, P., Chook, V. W. C., Chessa, S., Gotta, A., Hu, Y. F., “Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards,” Computer Communications, Vol. 30, No. 7, pp. 1655-1695.

[6] Djenouri, D., Khelladi, L., Badache, N. (2005). A Survey of Security Issues in Mobile Ad-Hoc and Sensor Networks. IEEE Communications Surveys & Tutorials, 7(4), 2-28.

[7] Karlof, C., Wagner, D. (2003). Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. Elsevier's Ad Hoc Networks Journal, Special Issue on Sensor Network Protocols and Applications, 1(2-3), 293-315.

[8] Ye, F., Luo, H., Lu, S., Zhang, L. (2005). Statistical En-route Filtering of Injected False Data in Sensor Networks. IEEE Journals on Selected Areas in Communications, 23(4), 839-850.

[9] Hu, Y. C., Perrig, A. and Johnson, D. B. (2006). “Wormhole Attacks in Wireless Networks”, IEEE Journal on Selected Areas in Communications, Vol. 24, No. 2, 370-380.

[10] Zhu, S., Setia, S., Jajodia, S., Ning, P. (2004). An Interleaved Hop-by-Hop Authentication Scheme for Filtering of Injected False Data in Sensor Networks. Proc. S&P, 259-271.

[11] Yu, Z., Guan, Y. (2005). A Dynamic En-route Scheme for Filtering False Data Injection in Wireless Sensor Networks. Proc. of SenSys, 294-295, ACM.

[12] Lee, H. Y., Cho, T. H. (2006). Key Inheritance-Based False Data Filtering Scheme in Wireless Sensor Networks. Lecture Notes in Computer Science, LNCS 4317, 116-127, Springer Verlag.

[13] Nghiem, P. T. and Cho, T. H. (2009). “A fuzzy-based interleaved multi-hop authentication scheme in wireless sensor networks”, Journal of Parallel and Distributed Computing, Vol. 69, No. 5, 441-450.

[14] Lee, H. Y. and Cho, T. H. (2007). “Fuzzy Adaptive Selection of Filtering Schemes for Energy Saving in Sensor Networks”, IEICE Trans. Comm., Vol. E90-B, No. 12, 3346-3353.

[15] Khalil, I., Bagchi, S., and Shroff, N. B. (2007). "LiteWorp: Detection and isolation of the wormhole attack in static multihop wireless networks," Computer Networks, Vol. 51, No. 13, 3750-3772.

[16] Khalil, I., Bagchi, S., and Shroff, N. B. (2008). "MOBIWORP: Mitigation of the wormhole attack in mobile multihop wireless networks," Ad Hoc Networks, Vol. 6, No. 3, 344-362.

[17] Madria, S. and Yin, J. (2009). "SeRWA: A secure routing protocol against wormhole attacks in sensor networks", Ad Hoc Networks, Vol. 7, No. 6, 1051-1063.

[18] Zhu, S., Setia, S., Jajodia, S. (2003), "LEAP: Efficient Mechanisms for Large-Scale Distributed Sensor Networks," ACM Conference of Computer and Communications Security, pp. 62-72.

[19] Yun, J-H., I-H. Kim, J-H. Lim and S-W. Seo, "WODEM: Wormhole Attack Defense Mechanism in Wireless Sensor Networks", Lecture Notes in Computer Science (LNCS), Vol. 4412(2007), 200-209.

[20] Lee, H. Y. and T. H. Cho (2009), "Fuzzy-Based Path Selection Method for Improving the Detection of False Reports in Sensor Networks", IEICE Trans. Inf. & Syst., Vol. E92-D, No. 8, 1574-1416.

[21] Sun, C. I., Lee, H. Y., and Cho, T. H. (2009). "A Path Selection Method for Improving the Detection Power of the Statistical Filtering in Sensor Networks", Journal of Information Science and Engineering, Vol. 25, No. 4, 1163-1175.

[22] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., and Pister, K. (2000). System Architecture Directions for Networked Sensors. Proc. of ACM ASPLOS IX, 93-104.

[23] Xbow sensor networks, http://www.xbow.com

[24] Btnodes, http://www.btnode.ethz.ch/

[25] Roy, S. D., Singh, A., Choudhury, S., and Debnath, N. C.(2008), "Countering Sinkhole and Black hole Attacks on Sensor Networks using Dynamic Trust Management," IEEE Symposium on Computers and Communications, pp. 537-542.

[26] da Silva, A. P. R., Matins, M. H. T., and Rocha, B. P. S.,(2005), "Decentralized Intrusion Detection in Wireless Sensor Networks," Q2SWinet '05, pp. 16-23.

Authors

Soo Young Moon received his B.S. and M.S. degrees in Electrical and Computer Engineering from Sungkyunkwan University in 2007 and 2009, respectively. He is now a doctoral student in the College of Information and Communication Engineering at Sungkyunkwan University. His research interests include modeling and simulation, wireless sensor networks, network security, and artificial intelligence.

Tae Ho Cho received his Ph.D. in Electrical and Computer Engineering from the University of Arizona, USA, in 1993 and his B.S. and M.S. degrees in Electrical Engineering from Sungkyunkwan University, Korea, and the University of Alabama, USA, respectively. He is currently a Professor in the College of Information and Communication Engineering at Sungkyunkwan University, Korea. His research interests include wireless sensor networks, intelligent systems, modeling and simulation, and enterprise resource planning.