A multi-path interleaved hop-by-hop en-route filtering scheme in wireless sensor networks

Thao P. Nghiem, Tae Ho Cho*

School of Information and Communication Engineering, Sungkyunkwan University, 300 Cheoncheon-dong, Jangan-gu, Suwon 440-746, Republic of Korea

ABSTRACT

A compromised node can generate a fabricated report, which results in false alarms, information loss, and a waste of precious network energy. An interleaved hop-by-hop authentication (IHA) scheme has been proposed to minimize such serious damage by detecting and filtering false reports at the very early en-route nodes. Unfortunately, IHA, with a single path from the source to the BS, cannot keep its security goal if more than \( t \) intermediate nodes are compromised. In this paper, an enhanced multi-path interleaved hop-by-hop authentication (MIHA) scheme is proposed. MIHA sets up disjoint and braided paths and switches to alternate paths when there is more than \( t \) compromised nodes on the current path to continue dealing with en-route insider attacks. A new key assignment mechanism was also applied to enhance network security and to reduce key storage overhead. Through analysis and simulations, MIHA exhibits improved resilience to en-route insider attacks and filters more bogus reports at early hops than IHA. Specially, with an attack frequency of 1/15 and a false traffic rate of more than 60\%, MIHA with three disjoint paths can filter over 27\% of false reports and is more energy efficient when compared to IHA.

1. Introduction

With the rapid development of hardware and wireless network technologies, wireless sensor networks (WSNs) have been widely applied in ubiquitous computing systems [2,3]. Because these WSNs are often unattended when deployed in security-sensitive areas such as hostile fields or homeland security operations, they are prone to capture and may have their security compromised [22]. In addition, the power of sensor nodes is very limited and irreplaceable. Therefore, security and energy efficiency are the most challenging and crucial tasks for the design of WSNs [1,18].

After compromising a number of sensor nodes, adversaries can expose all of the information in this node and subsequently launch insider attacks. Such attacks include the injection of bogus reports to raise false alarms and the consumption of limited network energy [8]. As one solution to minimize serious damage, false reports should be filtered en-route as soon as possible and the few missed reports should then be rejected at the base station (BS). Many en-route filtering schemes have recently been researched and developed in order to address this type of attack [12,21,19,11,9,23]. In the interleaved hop-by-hop authentication (IHA), reports are verified and filtered by every en-route node and finally by the BS. The security goal of IHA is to guarantee that a false report cannot travel more than \( t + 1 \) non-compromised nodes as long as no more than \( t \) nodes are compromised, where \( t \) is a security threshold based on each type of WSN [23]. However, there are several disadvantages to IHA. First, it relies on a single path routing protocol. Whenever more than \( t \) en-route nodes are compromised, IHA may not maintain its security goal. To continue the en-route filtering of fabricated reports, IHA has to set up new paths during the sensing operation time, the highest sensitive time when attacks are very difficult to detect and localize. The second drawback of IHA is that its key sharing mechanism requires that every node must share its authentication key with their lower and upper associated nodes. This decreases the network’s security strength. When a node is compromised, adversaries can obtain all of its information. In this paper, an enhanced multi-path interleaved hop-by-hop authentication (MIHA) scheme is proposed with consideration of the multi-path routing technique and a new secured key assignment. The contributions of this paper are as follows:

- We propose the multi-path technique by pre-defining a number of disjoint and braided paths from the source clusters to the BS in the Initialization and key assignment phase. At a specific time, only one path from a source cluster to the BS is activated and used. However, whenever a path is attacked, the unattacked alternative will be used. This not only maintains the security goal of IHA even when more than \( t \) nodes on the path to the BS are compromised, but also avoids attacks that are very hard to detect in sensing operation time.

* Corresponding author. Tel.: +82 31 290 7221; fax: +82 31 290 7230.
E-mail addresses: thaonp@ece.skku.ac.kr (T.P. Nghiem), taecho@ece.skku.ac.kr (T.H. Cho).

0140-3664/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2010.03.006
2. Background

2.1. En-route filtering schemes and related work

Many en-route filtering schemes have recently been proposed to cope with false data injection in WSNs [9–12,15,19–21,23]. The statistical en-route filtering (SEF) scheme, proposed by Ye et al., is the first research to address the attack and introduce an en-route filtering scheme framework [21]. In SEF, a report is generated by the collaboration of multiple nodes which create and attach MACs into the report. Based on those enclosed MACs, the report is verified and forwarded by en-route nodes on the path to the BS. Kim et al. applies a multi-path approach into SEF as an extend [9]. SEF was also improved by its authors in [20] where the secret keys are bound to geographic locations and thus damages of compromised nodes are limited. However, those schemes are probabilistic since it cannot guarantee that every false message will be en-route filtered. In addition, a symmetric key sharing is approach used in those schemes, where keys for generating and verifying MACs are the same. As a result, verification keys may be misused to generate reports. To avoid this problem, a secure ticket-based en-route filtering (STEF) Scheme was introduced with ticket concept where a MAC on the report uses a shared key between the en-route node and the BS [10]. However, this scheme is presented to authenticate reports only in query-based sensor networks rather than event-driven sensor networks.

The IHA defines a new concept of association among sensor nodes [23]. In the initialization phase, every node discovers its upper-stream and lower-stream associated nodes which share the same keys. Generated reports are verified using the pairwise-keys which shared with associated nodes. As an enhancement of IHA, Nghiem et al. proposed FIMA where fuzzy logic is applied to choose appropriated en-route nodes to forward reports [15]. However, in both scheme an adversary can forge arbitrary reports if there are more than a certain number of compromised nodes. In addition, those schemes are also based on symmetric key assignment, with the problems mentioned above.

2.2. An interleaved hop-by-hop authentication (IHA) scheme

In this section, the main characteristics of the IHA scheme are briefly described. Similar to a general en-route filtering scheme, IHA involves five primary phases: node initialization and deployment, association discovery, report endorsement, en-route filtering, and BS filtering [21].

An intermediate node has two upper and lower associated nodes that are \( t + 1 \) hops away (where \( t \) is a design parameter which is based on the application type). Additionally, a node which is less than \( t + 1 \) hops away from the cluster-head has one of the cluster-nodes as a lower associated node. For example, in Fig. 1, \( CH_1 \) considers \( CN_3 \) as its lower associated node and \( CH_7 \) as its upper associated node. After discovering an association among nodes, IHA establishes symmetric key sharing between a node and its lower and upper associated nodes.

When an event occurs, the surrounding \( t + 1 \) cluster-nodes collaboratively generate an event report and forward the report towards the BS. After receiving the report, the intermediate node verifies the authenticity of the report by comparing the message authentication code (MAC) generated by its lower associated node using the pairwise key shared with this node. If the verification succeeds, the en-route node will compute another MAC using the key shared with its upper associated node and append this MAC onto the report. The report continues to be forwarded to the BS if there is no failure in en-route authentication. Otherwise, the report is discarded. The BS serves as the final gate to perform the last verification of the transmitted report.

IHA guarantees that the BS will detect any injected false data packages when no more than \( t \) nodes are compromised. Furthermore, if every node knows the IDs of the nodes that are \( t + 1 \) hops away, a false report can travel only \( t \) hops at the most. Otherwise, the report can be forwarded up to \( t(1/t - 2) \) hops away.

However, there are several existing problems in IHA:

- There is only one path from the source cluster to the BS. Therefore, if there are more than \( t \) compromised en-route nodes, IHA cannot keep its security goal. False reports from the source may travel more than \( t + 1 \) non-compromised nodes or even reach the BS.
- Every node must exchange its AK with its lower and upper associated nodes. This method is inefficient for saving sensor node memory and can cause information loss when a specific node is compromised.

3. Proposed scheme

To achieve improved resilience to en-route insider attacks with acceptable trade-off energy, the MIHA scheme, inspired by IHA, is proposed. MIHA, based on cluster organization, uses multi-path routing to rapidly find alternate paths between the source and the BS and applies a new key assignment method.

3.1. System model and assumption

A large-scale static sensor network, which breaks down into local clusters with one node (the cluster-head) acting as the local BS, is considered. In a cluster, the cluster-head position rotates among the various sensors to prevent draining the battery of a single sensor. Once the cluster-head has sufficient data from its cluster-nodes, it aggregates the data and transmits the compressed data to the BS [5].
Sensor nodes are assumed to be of ample density so as to guarantee that a real report could collect \( t + 1 \) votes. Furthermore, each node has enough memory to store several hundred bytes of key material. The radio unit can be turned off in order to avoid receiving unintended reports and to save sensor energy [23].

When a node is compromised or physically attacked by an attacker, the assumption is made that the attacker can achieve full control over the captured node (i.e. the attacker can read the memory and influence the operation of the node software). However, the Master BS should have sufficient protection so that it is not compromised [7].

In this work, the focus is on handling en-route insider attacks where all compromised nodes in the source cluster are compromised and no nodes on the paths are attacked.

3.2. Notations and definitions

The notations and definitions below will be used in the latter sections of this paper:

3.2.1. Notations

- \( CN_i \) is a cluster-node, ID \( i \), at the current round.
- \( CH_i \) is a cluster-head of cluster \( i \) at the current round.
- \( t + 1 \) is the minimum number of cluster-nodes, where \( t \) is a design parameter based on the properties of applications [23].

3.2.2. Definitions

**Definition 1.** The definition of associated nodes in IHA is now extended. In IHA, two nodes are considered associated if there are \( t + 1 \) hops between them; association is only considered between two nodes each time. However, in MIHA, two arbitrary nodes are associated if and only if there are \( n(t + 1) \) hop counts between them, where \( n = 1, 2, 3, \ldots \). In addition, each node in the source cluster has an associated node within \( t + 1 \) en-route nodes near the cluster. For example in Fig. 2, \( CN_1 \), \( CH_1 \), and \( CH_2 \) are mutually associated nodes. Similarly, \( CN_2 \), \( CH_2 \) and \( CH_6 \); \( CN_3 \), \( CH_3 \), and \( CH_7 \); \( CH_4 \) and \( CH_8 \) are also associated nodes, respectively.

**Definition 2.** A path attack occurs when there is a path which has more than \( t \) compromised en-route nodes.

3.3. Scheme details

An overview of the MIHA scheme is given in Section 3.3.1. This is followed by a detailed description of the four phases of MIHA in Sections 3.3.2, 3.3.3, 3.3.4, 3.3.5. These four phases are as follows: **Initialization and key assignment**, **Report generation**, **En-route filtering**, and **BS verification**. The **Route switching** phase is described in Section 3.3.6.

3.3.1. Overview

We combine the **Initialization and deployment** and **Association discovery** phases in IHA into the **Initialization and key assignment** phase in MIHA. This is because association discovery is not our focus and it was described in detail in IHA. The proposed scheme is divided into four phases as a general en-route filtering framework:

- **Initialization and key assignment**: Since association discovery is not emphasized in MIHA, it is merged into this phase. This phase forms clusters, discovers routes from each node to the BS, associates discovery, and distributes keys for each sensor node. In this phase, instead of only finding and setting-up a single path from the source to the BS, MIHA sets up and maintains many alternate paths and distributes their keys in advance. Moreover, a new key assignment method is applied in which the AK of a node results from a hash function of its downstream node’s AK.
- **Report generation**: When an event occurs, surrounding clusters compete with each other to prepare a report. The cluster-head of the winner collects MACs from collaborated cluster-nodes, places them into the report, and forwards the report towards the BS [5].
- **En-route filtering**: After receiving the report and using the hash function to extract the authentication key, en-route nodes check the MACs’ information in the report to decide if the forwarding procedure should continue.
- **BS verification**: The final verification is the last shield of the network. When the BS receives the report, it verifies if all of the nodes in the source cluster have correctly endorsed the report. If the verification succeeds, the report is accepted; otherwise, the report is discarded.

Additionally, the scheme has a route-switching procedure, which is called by the BS after receiving a fixed number of reports. If this procedure detects certain path attacks, it will decide to change to an alternate path. As a result, MIHA prevents en-route insider attacks.

3.3.2. Initialization and key assignment

After the sensor nodes are scattered on the field of interest, the BS generates the secret keys and shares the keys with each node. This type of key is called an individual key, \( IK_i \). Each cluster elects a cluster-head and the BS broadcasts a route request to the cluster-head nodes for route discovery. Instead of setting-up only one path from the source to the BS, the scheme establishes multi disjoint and braided paths [6,4,16]. It should be noted that, although we set up multi-paths from a source cluster to the BS, only one path is used at a specific time. The other paths are alternatives which are considered when the current path is attacked.

Disjoint and braided paths are constructed according to an idealized algorithm shown below:

- For disjoint paths (see Fig. 3):
  - Construct the primary path \( P \) between the source and the BS.
  - The first alternate disjoint path \( P_1 \) is the shortest path node-disjoint with \( P \).
  - The second alternate path \( P_2 \) is the shortest path that is node-disjoint with \( P \) and \( P_1 \), and so forth.
The key mechanism illustrated in Fig. 5. Accordingly, nodes in the group nearest to the BS.

For each node on the primary path, find the shortest path from the source to the BS that does not contain that node. In Fig. 4, the first braided path links from CH5 and the BS without including CH1. Similarly, the second braided path links from CH0 and the BS without including CH2.

When all paths from the cluster-heads to the BS are established, the scheme begins an associate discovery procedure consisting of two steps: BS Hello and Cluster Acknowledgement. The purpose of this procedure is for a node to discover the ID of its associated nodes [23]. This procedure is repeated for all paths from the source to the BS.

For each path, the BS distributes AKs as shown in the mechanism illustrated in Fig. 5. Accordingly, nodes in the group nearest the BS generate an AK. All of their associated nodes and one associated node in the source cluster subsequently get a key based on this key. Specifically, associated nodes get a key from a result of a hash function of their upper associated node’s key. The associated nodes considered in this example are: CN1, CH1, CH9, and CH13. CH13, the node nearest the BS, gets key AK13 from the BS. The key CH9 is the result of a hash function of the key of its upper associated node CH13. Similarly, CH1 has the key AK5 = hash(AK9), CH9 has the key AK9 = hash(AK5), and CN1 has the key AK31 = hash(AK5). In this method, every en-route node does not require knowledge of its lower and upper keys, as in IHA. Therefore, key storage memories are saved and key information loss decreases when an en-route node is compromised. More importantly, in the case that a node is compromised, the AKs of upper and lower associated nodes are not easily exposed. For example, we assume CH5 is compromised and the adversary can obtain its AK: AK5. In IHA, all upper and lower AKs of CH5, including AK31, AK1, AK9 and AK13, are exposed and security is broken. However, in MIHA, it is impossible for the attacker to get AK9 and AK13 because the hash function is irreversible. In addition, AK31 and AK1 are not apparent as in IHA, but are a result of the hash function. AK31 and AK1 may be exposed when the adversary knows the hash function.

3.3.3. Report generation

When a source cluster-head is triggered by an event E, it generates a report and asks if the cluster-nodes agree on that report. If a cluster-node CNi accepts the report, two MACs for E are generated using IK and AKi. If the cluster-head collects enough MACs from t + 1 cluster-nodes (including itself), a final report is generated. The generated report is composed of compressed t + 1 individual MACs named XMAC(E) by the XOR operation and uncompressed t + 1 association MACs. Specifically, the report R from cluster C, has the following form:

\[ R = E, C, \text{XMAC}(E), \text{MAC}(AK_{51}, E), \ldots, \text{MAC}(AK_{50}, E), \text{MAC}(AK_{5}, E) \]

For example, if the report R on the event E is generated by the cluster C0 as in Fig. 5, the format is:

\[ R = E, C_0, \text{XMAC}(E), \{\text{MAC}(AK_{51}, E), \text{MAC}(AK_{52}, E), \text{MAC}(AK_{53}, E), \text{MAC}(AK_{50}, E)\} \]

3.3.4. En-route filtering

When an en-route node receives report R, it first examines whether there are s different association MACs if the node is s (s < t + 1) hop(s) away from the BS; otherwise, an en-route node should evaluate if there are enough t + 1 different association MACs. The en-route node then extracts the last MAC in the association MAC list for comparison with the computed MACs. The key used to generate the MAC is the result of a hash function of its AK. For example, in Fig. 5, when CH9 receives the report, a hash function of AK9 is called to obtain the authentication key. This key is then used to generate a MAC to compare with the first MAC, generated by CH3 using AK9, in the uncompressed MAC list carried by the report. If the authentication procedure succeeds, the first MAC in the association MAC list is removed and the new generated MAC is appended to the MAC list. CH9 continues to forward to CH13, the next upstream nodes. Otherwise, if there is any failure in authentication, the report is immediately discarded. This phase is illustrated in Fig. 6.

3.3.5. BS verification

When the BS receives the report, the XMAC is checked. Based on the cluster ID, the BS determines the corresponding nodes and computes the individual MACs using the IK of the cluster-nodes in the source cluster. The BS then uses the XOR operation with these MACs and compares the result with the XMAC. If verification fails, the BS will discard the report.

3.3.6. Route switching

Whenever the number of intermediate compromised nodes is more than t, the path is considered attacked. There are many algorithms to detect compromised nodes [22,13,14]. However, methodologies to detect attacked nodes are not the focus of this study. After receiving a fixed number of reports, the BS examines whether any paths have been attacked. If any path attack exists, the BS switches to an unattacked path that is resilient to en-route insider path attacks.

It is noted that our multi-path techniques are different from the local repair mechanism in IHA. Our scheme aims to proactive mechanism to path attacks where alternative paths are available for path switching request. As a result, MIHA can avoid significant
time consumption, and deters excessive control traffic for route determination. More importantly, we can limit attacks that are very hard to detect during the sensitive route establishment time.

4. Performance evaluation

4.1. Security analysis

The first assumption in IHA is that false data injection can only be launched by up to \( t \) compromised nodes or there are no more than \( t \) compromised nodes on the path to the BS. Therefore, when the number of compromised nodes in the path to the BS exceeds \( t \), this security goal is broken. As such, there is no guarantee that false reports traveling on this path are filtered after they are forwarded. This security goal is broken. As such, there is no guarantee that false reports. Due to the scenario of setting up alternate paths at an earlier stage at en-route nodes. Consequently, MIHA can save energy and minimize damage from these false reports. Due to the scenario of setting up alternate paths at the initialization and key assignment phase, re-finding and setting-up new paths are also avoided during the operation time, when attacks are hard to detect.

MIHA applies a new key assignment which not only saves key storage memory, but also enhances network security. Based on our hash-based key assignment, on a path to the BS each node only needs to store one \( AK \) which is either generated by the BS or is the result of a hash-based function of the key of its upper associated nodes. Compared with the key assignment methods in IHA, where each en-route node must store two \( AKs \) of all its associate nodes, our method can save key storage memory for each node. For example, in Fig. 5, node CH5 must store two \( AKs \) shared with CH1 and CH6 in IHA. However, in MIHA, there is only one \( AK \) of its own CH5 that should be stored.

If an en-route node in MIHA is compromised, attackers cannot determine the key of its lower associated nodes and have to solve the hash algorithm in order to obtain the key of its upper associated nodes. Therefore, by decreasing the probability of key information loss, MIHA effectively enhances network security.

4.2. Computational cost

In this section, we evaluate computational cost of MIHA and compare it with that of IHA in two operations: Report authentication and route switching.

4.2.1. Report authentication

In both IHA and MIHA, each cluster-node has to compute three MACs for a report. One uses its \( IK \), one uses its \( AK \) shared with its upper associated node, and one uses key sharing with its cluster-head. Every en-route node in either IHA or MIHA normally computes four MACs. One is used for comparison with its lower associated MAC, one is used to attach to the received report, and the others are employed to verify with its one-hop upstream and downstream neighbor nodes. The energy required for computing a MAC is approximately equal to the energy used for transmitting to one byte. In fact, with the exception of the energy used to compute the same number of MACs for a report in IHA, MIHA has to spend more energy to perform a hash function to get the key. According to [17], energy cost to perform a SHA-1 function for a two-byte key are 1.52 \( \mu \)J, nearly 9% of the energy required for computing a MAC. Therefore, the overhead of MIHA compared to IHA in this operation is insignificant.

4.2.2. Route switching

When a path attack is recognized, MIHA examines an alternate path and picks one unattacked path. Similar to a binary search algorithm, this operation costs \( O(\log n) \) where \( n \) is the number of alternate paths. Since \( n \) is usually a small number and this operation is done by the BS, this cost is also insignificant.

4.3. Communication cost

In MIHA, the BS needs to periodically flood low-rate data over the alternate paths in order to keep these paths alive. The total volume of low-rate data sent over alternate paths was equalized. Therefore, the energy for multi-paths maintenance was calculated based on the length of alternate paths and the primary path as:

\[
\frac{(L_a - L_p)}{L_p}
\]

where \( L_a \) is the average length of alternate path and \( L_p \) is the length of the primary path [4].

4.4. Minimum number of nodes for efficient paths

Every cluster consists of \( t + 1 \) nodes. To exploit the multi-path technique, there should be at least \( t + 1 \) on the path. Otherwise, the scheme cannot detect any path attack to switch to another alternative available path. Assuming there are \( d \) disjoint paths from the source cluster to the BS, the minimum number of required nodes to build efficient disjoint paths, \( N \), from a certain source cluster to the BS is calculated as below: \( N = (t + 1) + d(t + 1) = (d + 1)(t + 1) \)

For braided paths, it is impossible to know how many nodes need to go round a node because it depends on the network topology. Nevertheless, more than \( 2(t + 1) \) nodes would be involved to build efficient paths from a specific source cluster to the BS.
4.5. Simulation model and results

A network field with a size of $1500 \times 500$ unit$^2$ was constructed in which the four-node cluster size was $50 \times 50$ unit$^2$ and the initial energy of each node was 1 $\mu$J. All sensor nodes had the same hardware and transmission power. Each node requires 16.25 and 12.5 $\mu$J to transmit and receive a byte over one-hop, respectively, and each MAC generation consumes 15 $\mu$J [21]. The size of reports was 12 bytes and the MAC size was 1 byte [11].

Both disjoint and braided multi-path routing were deployed and evaluated for comparison with the original IHA. The BS was assumed to know the location of all nodes by beacon messages or GPS systems and performs the path setting-up task at the initialization and key assignment phase. Two hundred reports, including legitimate and fabricated reports, were sent by randomly chosen cluster-head nodes.

Each node was assumed to have a probability of being compromised, $p$, during a small interval $T$. The resilience to path attack was then defined as the probability of at least one alternate path being available within the interval $T$. Specifically, the resilience of a multi-path to path attack was the average value assigned to sets in which at least $t$ nodes on the primary path were attacked [4]. The attack frequency was also defined as the frequency that an attack occurs. For example, an attack frequency of 1/30 means that, for every 30 generated reports, there is one attack. The number of attacked nodes at each attack was called the attack rate and was set as a parameter of the network. After each attack, the BS decides which path is attacked based on the number of attacked nodes on the path.

In Fig. 7, the first evaluation of a trade-off between the energy consumption and the filter rate (the rate of fabricated reports which are en-route filtered) is shown. In this evaluation, the false traffic rate (rate of false reports to total reports) was set at 60% and IHA and MIHA were evaluated with an attack frequency of 1/30 and 1/15, respectively. Overall, the average resilience rate and filter rate of the MIHA schemes were always better than those of IHA. This is because there are two or three alternate paths in MIHA that are resilient to path attacks, a scenario that keeps the network safe from en-route insider attacks. Furthermore, when the attack frequency is low, MIHA spends more energy than IHA, since MIHA needs to send low-rate data to set up and maintain these paths. However, when the attack frequency increases, there are more path attacks. In IHA, when single paths are attacked, IHA cannot guarantee en-route filtered false reports. Conversely, there are several backup paths to handle a path attack in the MIHA scheme. Therefore, fabricated reports are filtered at very early intermediate nodes; this saves precious network energy.

In Fig. 8, the trade-off energy was evaluated while the false traffic rate was varied from 0% to 100% with an attack frequency of 1/30 and 1/15. In general, IHA exhibited better energy efficiency when the attack frequency and false traffic rate were low. However, when the false traffic rate was over 40% and the attack frequency was 1/15, energy efficiency was better in the MIHA scheme, since MIHA is more resilient to path attacks and handles en-route insider attacks better than IHA. For a network with a low false traffic rate and attack frequency, MIHA with three disjoint paths is not very economical.
Fig. 9 compares the efficiency in filtering en-route false reports between the original IHA and the MIHA with two or three disjoint or braided paths. Accordingly, if more alternate paths exist, the filter rate is improved. Moreover, there is a trade-off between the disjoint and braided paths. Although the cost of maintaining disjoint paths is higher than that of braided paths, MIHA with a disjoint path filters false reports much better than MIHA with braided paths. Specifically, with an attack frequency of 1/15 and a false traffic rate greater than 40%, MIHA with three disjoint paths can filter over 27% of false reports when compared to IHA.

5. Conclusions and future work

In this paper, a new en-route filtering scheme, MIHA, inspired by the original IHA scheme, was developed. MIHA constructs multi-paths from a source to the BS and applies a new key assignment method for enhanced network security with acceptable trade-off energy. Through simulations it was demonstrated that, for an attack frequency of 1/15 and a false traffic rate greater than 60%, MIHA with three disjoint paths can filter 27% more en-route false reports than IHA and is more energy efficient. In addition, the key assignment mechanism that was implemented can reduce key storage overhead and make it more difficult for adversaries to determine the secret sharing keys between associated nodes.

Our work can be extended into three directions as following:

- Taking into consideration of query-based sensor networks. The current scheme is effective in filtering false report for event-driven sensor network where reports are gathered, aggregated and sent only when there is an event. We intend to adjust and implement our scheme to fit the on-demand fashion where the BS will send queries to specific clusters for information acquisition.
- Taking advantage of multi-path technique to solve bottleneck problem. Nodes which are nearer the BS are always in charge of forwarding reports from distant nodes. As a result, these nodes tend to run out of energy very soon and create bottleneck problem. Exploiting multi-path and increasing density number of nodes near the BS to enhance network security as well as deal with bottleneck is one of our research directions in future.
- Extending our scheme to fit mobile sensor networks.

Acknowledgement

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the government (MEST) (No. 2009-0076504).
References