International Journal of Applied Engineering Research, ISSN 0973-4562 Volume 13, Number 8 (2018) pp. 5589–5599

Energy-Efficient Detection of Compromised Nodes in Wireless Sensor Networks

Minki Kim and Haengrae Cho
Department of Computer Engineering, Yeungnam University, Republic of Korea.

Abstract: A wireless sensor network (WSN) can provide a low cost and flexible solution to sensing and monitoring for large distributed applications. To save energy and prolong the network lifetime, the WSN is often partitioned into a set of spatial clusters. Each cluster includes sensor nodes gathering similar data, and just a few sensor nodes (samplers) report their sensed data to a base node. The base node may predict the missed data of non-samplers using the spatial correlation between sensor nodes. The problem is that the WSN is vulnerable to internal security threats such as node compromise. If the samplers are compromised and report incorrect data intentionally, then the WSN will become contaminated rapidly due to the process of data prediction at the base node. In this paper, we propose three algorithms to detect compromised samplers for secure data collection in the WSN. The proposed algorithms leverage the unique property of spatial clustering to alleviate the overhead of compromised node detection. Experiments indicate that the proposed algorithms can identify compromised samplers with a high accuracy and low energy consumption when as many as 50% of the sensor nodes misbehave.

Keywords: wireless sensor network, security, energy, cluster, data collection, node compromise.

INTRODUCTION

A wireless sensor network (WSN) can provide a low cost and flexible solution to collect data for sensing and monitoring. There are two types of data collection in the WSN, *event-based* and *periodic* approach [1]. In event-based data collection, sensor nodes are responsible for detecting and reporting events such as spotting moving targets. They perform local filtering and sometimes collaborate with each other to detect events. On the other hand, in periodic data collection, every sensor node reports periodically its sensing data to the base node. Many studies adopt the periodic approach because it enables arbitrary data analysis at the base node [2], [3], [4]. However, extracting the vast amount of data generated by large-scale WSN can cause sensor nodes to drain their batteries soon. This means that we need an energy-efficient way of data collection to prolong the lifetime of WSN.

Spatial clustering is a representative way of saving energy in the periodic data collection [5], [6]. It partitions the network into a set of clusters. Each cluster includes sensor nodes with similar sensing data, and just a few sensor nodes (*samplers*) report their data to the base node. The remaining sensor nodes can save their energy by keeping in sleep mode. The base node may use the spatial correlation between sensor nodes to predict the data of non-sampler nodes. To balance the energy consumption, the role of samplers can be distributed evenly for all sensor nodes of the cluster [1].

The WSN is vulnerable to both external and internal security threats due to unreliable wireless channels, unattended operation of sensor nodes, and resource constraint [7]. *Node compromise* is a major type of internal attacks. Compromised sensor nodes release all the security information to the adversary. Then, the adversary can easily launch internal attacks with data alteration, message negligence, selective forwarding, and jamming. Note that the node compromise is especially problematic for periodic

data collection, where only the samplers may report sensing data to the base node. If the samplers are compromised and report incorrect data intentionally, then the WSN will become contaminated rapidly due to the process of data prediction at the base node. This means that detecting and defending against node compromise is inevitable task to guarantee the correctness of data collection at the WSN.

In this paper, we propose three algorithms to detect compromised samplers in the spatially clustered WSN. They are *monitoring by neighbors* (MBN), *cooperation of multiple samplers* (CMS), and a *hybrid algorithm of MBN and CMS* (HYB). MBN follows the traditional watchdog approach [,–7,9] to exploit the spatial correlation between a sampler and its neighbors. As a watchdog of the sampler, each neighbor node listens promiscuously to the sampler's broadcasting messages and monitor if the sampler is compromised. CMS does not impose any security roles to the neighbor nodes. Instead, it requires that each cluster has multiple samplers. Then the cluster head determines the compromised sampler with the majority consensus among samplers. HYB integrates MBN and CMS to increase the probability of detecting compromised samplers.

Most of previous algorithms detect compromised nodes by monitoring the communication behaviors such as packet dropping rate, packet sending rate, and forwarding delay time [7], [8], [10]. Note that those algorithms cannot work for detecting compromised samplers. This is because samplers may not deliver packets for other nodes. They just generate new packets including the sensed data and send them to the base node. Hence, to detect compromised samplers, we have to determine if the reported data is correct. In the proposed algorithms, either neighbor nodes or other samplers have a role to monitor the correctness of the reported data.

The rest of this paper is organized as follows. In the next section, we review the related work. Section 3 presents our model of WSN and describes the proposed algorithms in detail. Section 4 compares the performance of proposed algorithms qualitatively. Section 5 presents the experiment model. Section 6 discusses the experiment results. Finally, Section 7 concludes the paper.

RELATED WORK

For the past decade, many algorithms have been proposed to detect compromised nodes in the WSN [7], [11]. However, only a few algorithms considered the spatial clustering. Authors of [12] partition the WSN into several clusters by extending the distributed spatial clustering algorithm [5]. To detect compromised nodes, they divide the cluster into equal-sized sub-groups. Each sub-group monitors the entire cluster in turn to reduce the power consumption. The problem of this algorithm is not to consider the location of monitoring nodes. If a monitoring node is not located in the routing path between a sampler and the base node, it cannot detect if the sampler is compromised. Furthermore, there is no central decision point and thus every node in the cluster has to decide the node compromise by itself.

Authors of [13] proposed an algorithm to detect and revoke compromised nodes in a cluster. The cluster head monitors its member nodes and is responsible to determine if they are compromised. Note that this is contrary to [12] where every node has to decide the node compromise. The main problem of [13] is that they did not present how to detect the misbehavior of each sensor node. This should be performed for several monitoring attributes on sensing data and communication behaviors [8], [12], [14].

Recently, a few anomaly detection algorithms have been proposed that consider the notion of cluster. For example, authors of [9] proposed an intrusion detection system for a cluster-based WSN. Their system consists of three independent intrusion detection algorithms designed for a base node, cluster heads, and sensor nodes, respectively. Authors of [15] and [16] used the notion of cluster to minimize the communication overhead of anomaly detection. Note that these algorithms did not consider the selective sampling of spatial clustering, and thus every sensor node should perform sampling operations. On the other hand, in this paper, we focus on the unique property of spatial clustering such as selective sampling and strong correlation between sensor nodes in the cluster. Then we leverage it to improve the detection ratio of compromised samplers with less energy consumption.

It is worthy to compare our security problem with the secure data aggregation problem [17], [18]. In the typical data aggregation process, sensor nodes are organized into an aggregation tree rooted at the base node. Non-leaf nodes fuse data collected from their child nodes and forward the aggregated results toward the base node. If a non-leaf node is compromised and forwards incorrect results, then the WSN should be contaminated rapidly since the result includes every data collected by its descendant nodes. This means that the problem is very similar to the case of compromised samplers. However, the solution domain of the

secure data aggregation is relatively limited. The reason is that only child nodes can verify whether their parent has done right aggregate operation. On the other hand, in spatially clustered WSNs, any node can monitor samplers in the same cluster because they are strongly correlated. As a result, there are more options to detect compromised samplers. The goal of this paper is to describe each option in detail and compare their performance.

SECURE DATA COLLECTION

Model of WSN

Let $S = \{s_1, \ldots, s_n\}$ be a set of all sensor nodes in the WSN. Each sensor node can communicate only with its neighbors. The set of neighbor nodes of a sensor node s_i is denoted by $nbr(s_i)$. Since s_i is forced to sample periodically, the time-ordered sequence of its sensing data forms a time series $X_i = \{x_i^1, \ldots, x_i^q\}$ where x_i^t is a sensing data of s_i at a specific time t. To measure the similarity between two sensor nodes s_i and s_j in S, we employ the *Manhattan distance*, $\mathrm{MD}(s_i, s_j)$ which is defined as $\Sigma_{t=1}^{q}|x_i^t - x_j^t|/q$. Assuming that a dissimilarity threshold δ is given, we can define the degree of correlation and a cluster as follows.

Definition 1. Two sensor nodes, s_i and s_j in S, are *strongly correlated* if $\mathrm{MD}(s_i, s_j) \leq \delta$.

Definition 2. A set of sensor nodes C is called a *cluster* if the following two conditions hold for every pair of $s_i \in C$ and $s_j \in C$: (1) s_i can communicate with s_j directly or via any nodes in C, and (2) s_i and s_j are strongly correlated.

The construction of clusters based on the spatial correlation is an interesting research issue and has been studied by many researchers [2], [3], [4], [5]. In this paper, we assume that the WSN is already partitioned into disjoint clusters. For each cluster, a sensor node is selected as a *cluster head* (CH). Every node in the cluster periodically samples its sensor data and sends it to the CH. We call the period of sampling as a *forced-sampling period* (τ_f). At each τ_f, the CH evaluates the degree of correlation and starts a new cluster construction phase if sensor nodes in the cluster are not strongly correlated anymore.

Each node in a cluster becomes a sampler with a probability λ. Combining this randomized scheduling with the round robin scheduling, we can guarantee that at least one node becomes a sampler for each cluster. A sampler sends its sensing data to the base node for every τ_d. Then the base node can predict the sensing data of non-sampler nodes using some statistical inference methods [1]. We call τ_d as a *data-sampling period*. Note that we can save energy significantly by setting τ_f to be much longer than τ_d.

Sampler Monitoring

In this section, we present three sampler monitoring algorithms, MBN, CMS, and HYB. Every algorithm focuses on the data-sampling period only, because detecting compromised nodes at the forced-sampling period is rather easy and thus compromised nodes may behave normally at the forced-sampling period.

Monitoring by Neighbors (MBN)

MBN follows the traditional watchdog approach that relies on the broadcast nature of the wireless communications [8], [14]. At MBN, every neighbor node of a sampler has a role to the *watchdog* that overhears the message sent by the sampler. Suppose that a sensor node s_p is selected as a sampler. Then MBN performs the following steps to monitor s_p.

1. s_p notifies itself to the CH, and the CH wakes up the sensor nodes in $nbr(s_p)$. To make the compromised sampler perform this procedure, the CH may assign some unique id to the sampler in response to the notification. The base node should reject any message of the sampler if it does not contain the id.
2. For each data-sampling period, s_p reads its sensing data and sends it to the base node. If s_p is compromised, it may send incorrect data intentionally.
3. Every node $s_i \in nbr(s_p)$ also reads its sensing data for the data-sampling period. Furthermore, it overhears the message sent by s_p at the promiscuous mode. If s_p is not strongly correlated to s_i, s_i reports to the CH that s_p would be compromised.
4. The CH decides that s_p is compromised if majority in $nbr(s_p)$ reported. In this case, the CH informs the base node that s_p is compromised and selects a new sampler. s_p should be segregated from the network to enforce the security.

The underlying assumption of MBN is that majority in $nbr(s_p)$ are not compromised. However, if the assumption

International Journal of Applied Engineering Research, ISSN 0973-4562 Volume 13, Number 8 (2018) pp. 5589–5599

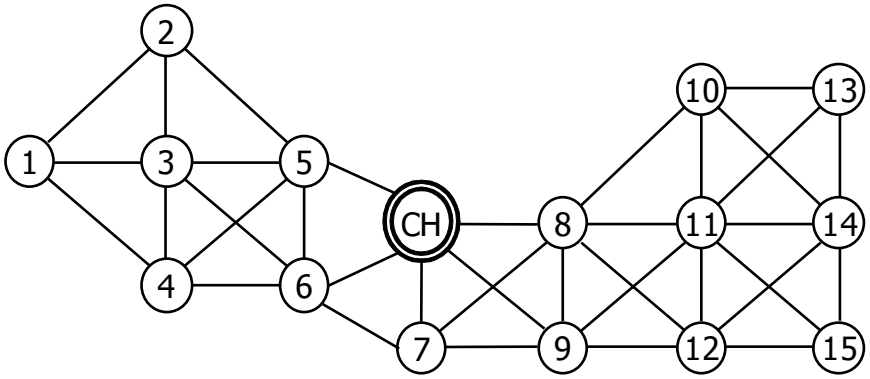

FIGURE 1. An example cluster

is not hold, MBN would fail to detect the compromised samplers. For example, suppose an example cluster of Figure 1. It consists of 15 sensor nodes and a CH. A line between sensor nodes represent that they are within a 1-hop distance. Since neighbors of a node include every other node within its communication range, we can deduce neighbor relationship from the Figure. For example, $nbr(3) = \{1, 2, 4, 5, 6\}$ and $nbr(9) = \{7, 8, 11, 12\}$. If node 3 is selected as a sampler, then nodes in $nbr(3)$ work in promiscuous mode and overhear the message sent by node 3. If some of the neighbor nodes find that node 3 is not strongly correlated, they report to the CH. The CH decides that node 3 is compromised if at least three neighbor nodes report the correlation mismatch, since $nbr(3)$ consists of five nodes.

Cooperation of Multiple Samplers (CMS)

Unlike MBN, CMS does not impose any security roles to neighbor nodes. Instead, it requires that there are at least three samplers for each cluster. CMS can detect any compromised sampler by testing the degree of correlation between samplers. If there are k samplers in a cluster, CMS performs the following steps to monitor the samplers.

1. For each data-sampling period, a sampler reads the sensing data and sends it to the CH. If the sampler is compromised, it may send incorrect data intentionally.
2. After the CH receives the sensing data from the k samplers, it performs a majority vote to detect compromised samplers. If a sampler is strongly correlated to at least $\frac{k}{2}$ samplers, it is a normal node. Otherwise, the CH decides that the sampler is compromised.
3. Among the normal samplers, the CH selects one in a random manner and sends its sensing data to the base node. Then the base node can predict the missed data of other nodes in the cluster.
4. If any sampler is determined as compromised, the CH informs the base node and replaces it with a new one.

The CH receives the sensing data of every sampler and performs the majority vote. This causes the CH to drain its battery more rapidly. As a result, CMS requires a traditional CH rotation algorithm to prolong the lifetime of WSN [1]. The positive effect is that the CH may not send every sensing data of the normal samplers (step 3). Instead, it forwards only one sensing data to the base node where other data can be predicted statistically. This way CMS can reduce the energy to deliver the sensing data.

The complexity of the majority vote at step 2 is $O(k^2)$, since it needs to test the degree of correlation for every pair of samplers. Note that CMS can be more accurate as the number of samplers increases. If there are only a few samplers, small number of compromised samplers would lead to a wrong decision. This shows an interesting tradeoff between energy consumption and security enforcement. We will experiment on the tradeoff at Section 6.

Hybrid Algorithm of MBN and CMS (HYB)

If many nodes are compromised at the same time, both MBN and CMS could not detect compromised samplers. For example, in Figure 1, suppose that MBN selects node 3 as a sampler. If node 3 and at least three nodes of $nbr(3)$ are compromised at the same time, the CH cannot detect that node 3 is compromised. Similar problems may also happen at CMS. For example, in Figure 1, suppose that node 3 and 9 are compromised. If the current samplers are node 3, 9, and

14, then the CH may determine node 14 as a compromised node. As a result, node 14 will be forced to disable (*false positive error*) and the incorrect data of node 3 or 9 is delivered to the base node (*false negative error*). False negative errors are particularly problematic in periodic data collection since the WSN wiil become contaminated rapidly due to the process of data prediction at the base node. HYB tries to reduce the frequency of false negative errors by combining MBN and CMS. Suppose that a cluster has k samplers ($k \geq 3$). Then HYB performs the following steps.

1. For each data-sampling period, every sampler reads the sensing data and sends it to the CH. If the sampler is compromised, it may send incorrect data intentionally.
2. Every neighbor node of a sampler, say s_p, also reads the sensing data and overhears the message of s_p. If any neighbor node finds that s_p is not strongly correlated to itself, then it reports to the CH that s_p would be compromised.
3. The CH performs *two-phase checking* to detect the compromised samplers. At the first phase, the CH checks if majority in $nbr(s_p)$ reported. If s_p gets the majority of votes, the CH decides it as compromised. At the second phase, the CH performs another majority vote among samplers passing the first phase. Some sampler may also be decided as compromised at this phase.
4. The CH selects one normal sampler that passes both phases and sends its sensing data to the base node. If any sampler is determined as compromised, the CH informs the base node and replaces it with a new one. The compromised sampler is segregated from the network.

PERFORMANCE ANALYSIS

The performance of the proposed algorithms depends on several factors of WSN, such as network density, number of samplers, and sampling cost. In this section, we compare qualitatively the performance of the sampler monitoring algorithms with respect to the security and the energy consumption.

Security Performance

The security of MBN relates to the number of neighbor nodes, $|nbr(s)|$. MBN can detect a compromised sampler s if more than $\frac{|nbr(s)|}{2}$ nodes are normal. Note that MBN is effective only if $|nbr(s)|$ is sufficiently large. If $|nbr(s)|$ is small, a few compromised neighbor nodes can get the majority of votes easily and thus can violate the correctness of data collection.

CMS is also dependent on the network density because it is more secure as there are more samplers. However, unlike MBN, CMS does not concern the number of neighbor nodes of a sampler. This means that CMS can be applied to a cluster where sensor nodes are distributed in a non-uniform manner. HYB is also resilient to the non-uniformity of node distribution due to the two-phase checking.

Energy Performance

We assume that each sensor node consumes E_{tx} energy unit for transmitting a packet, E_{rv} energy unit for receiving a packet, and E_{sp} energy unit for sampling a data.

MBN

At MBN, a sampler reads the sensing data and sends it to the base node for each data-sampling period. Suppose that the routing path between the sampler and the base node consists of N_{base} nodes. Then the energy spent by the sampler (E_{smp}) and the energy to deliver the sensing data to the base node (E_{data}) are as follows:

$$E_{smp}(\text{MBN}) = E_{sp} + E_{tx} \quad (1)$$

$$E_{data}(\text{MBN}) = (E_{rv} + E_{tx}) * N_{base} \quad (2)$$

If each cluster has N_{smp} samplers, both E_{smp} and E_{data} have to be multiplied by N_{smp}.

MBN requires that neighbor nodes of a sampler s have to (1) sample at each data-sampling period, (2) overhear the message sent by the sampler, and (3) report to the CH in case of correlation mismatch. As a result, the energy spent by neighbor nodes (E_{nbr}) is as follows:

$$E_{nbr}(\text{MBN}) = (E_{sp} + E_{rv}) * |nbr(s)| + E_{tx} * |nbr(s)| * P_{mis} \quad (3)$$

where P_{mis} is a probability that the sampler is not strongly correlated to its neighbor. Note that even in a normal case when P_{mis} is very low, every neighbor node has to always spend E_{sp} and E_{rv}. This is particularly problematic when E_{sp} is large because each node is equipped with many expensive sensors [3].

The report message of a neighbor node should be delivered to the CH. Suppose that N_{CH} is the number of intermediate nodes in the routing path between a neighbor node and the CH. Then the energy spent by intermediate nodes in the routing path (E_{report}) is as follows:

$$E_{report}(\text{MBN}) = (E_{rv} + E_{tx}) * |nbr(s)| * P_{mis} * N_{CH} \quad (4)$$

Finally, the CH receives the report messages of neighbor nodes and notifies the base node when the sampler is compromised. The energy spent by the CH (E_{CH}) is determined as follows:

$$E_{CH}(\text{MBN}) = E_{rv} * |nbr(s)| * P_{mis} + (E_{vote} + E_{tx} * P_{detect}) * P_x \quad (5)$$

where E_{vote} is the energy spent by the majority vote and P_{detect} is a probability that the sampler is determined as compromised. P_x is set to 1 if $|nbr(s)| * P_{mis} \geq 1$, and set to 0 otherwise. This means that both E_{vote} and E_{tx} should not be spent if none of neighbor nodes report to the CH.

CMS

Suppose that N_{smp} is the number of samplers in a cluster. Note that CMS requires N_{smp} to be at least 3. Then the total energy spent by samplers in a cluster (E_{smp}) is as follows:

$$E_{smp}(\text{CMS}) = (E_{sp} + E_{tx}) * N_{smp} \quad (6)$$

Since the sensing data of each sampler is sent to the CH, the energy spent by intermediate nodes in the routing path (E_{report}) is as follows:

$$E_{report}(\text{CMS}) = (E_{rv} + E_{tx}) * N_{CH} * N_{smp} \quad (7)$$

The CH receives the sensing data of every sampler and detects the compromised samplers by performing a majority vote. After that, the CH sends the sensing data of a normal sampler to the base node. The energy spent by the CH (E_{CH}) is determined as follows:

$$E_{CH}(\text{CMS}) = E_{rv} * N_{smp} + E_{vote} + E_{tx} \quad (8)$$

Suppose that there are N_{base} nodes in the routing path between the CH and the base node. Since the CH sends only one sensing data of any normal sampler, the energy to send sensing data to the base node (E_{data}) is as follows:

$$E_{data}(\text{CMS}) = (E_{rv} + E_{tx}) * N_{base} \quad (9)$$

HYB

Since HYB combines MBN and CMS, it has to spend both E_{nbr} and E_{smp}. The following equations summarize the energy consumption of HYB.

$$E_{smp}(\text{HYB}) = E_{smp}(\text{CMS}) \quad (10)$$

$$E_{nbr}(\text{HYB}) = E_{nbr}(\text{MBN}) * N_{smp} \quad (11)$$

$$E_{report}(\text{HYB}) = E_{report}(\text{MBN}) * N_{smp} + E_{report}(\text{CMS}) \quad (12)$$

$$E_{data}(\text{HYB}) = E_{data}(\text{CMS}) \quad (13)$$

E_{smp}(HYB) is equal to E_{smp}(CMS) if N_{smp} is set to the same value. However, compared to CMS, HYB can achieve similar security level with much smaller N_{smp}. At the next section, we will show the experiment results for various settings of N_{smp} and $|nbr(s)|$.

Like MBN, HYB makes neighbor nodes perform sampling at each data-sampling period. If there are N_{smp} samplers, E_{nbr}(HYB) is N_{smp} times as high as E_{nbr}(MBN). For each sampler, its neighbor nodes may send report messages to the CH. Furthermore, the sampler forwards its sensing data to the CH. This means that E_{report}(HYB) is the sum of E_{report}(MBN) for every sampler and E_{report}(CMS). Finally, since the CH may send only one sensing data to the base node, E_{data}(HYB) is equal to E_{data}(CMS.)

EXPERIMENT MODEL

We use a simulation approach to investigate the performance of the sampler monitoring algorithms in a large-scale WSN. Figure 2 illustrates a WSN configuration used

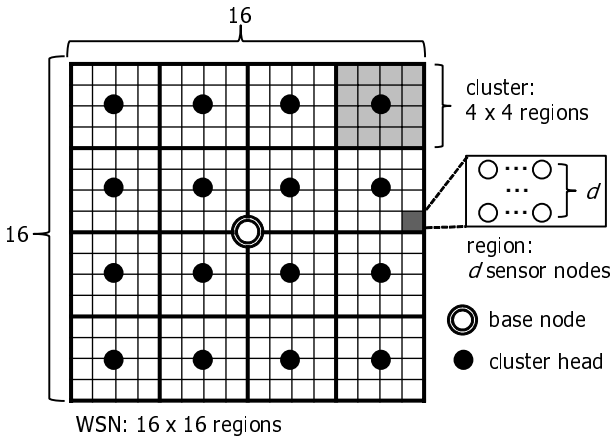

FIGURE 2. A WSN configuration

in the experiments. The simulation parameters are summarized at Table 1. The entire network consists of 16 × 16 square regions. For each region, d sensor nodes are uniformly distributed and can communicate directly with each other. We change the value of d from 4 to 16. Then the total number of sensor nodes is up to 4096. Every cluster consists of 4 × 4 square regions. Hence, there are 16 clusters in the WSN. A CH is located at the center of the cluster, which means that every sensor node in the cluster is within 2-hop distance to the CH. For each cluster, a CH selects N_{smp} samplers such that there is at most one sampler in a region. For a sampler s, the size of $nbr(s)$ is $d-1$. We vary N_{smp} from 3 to 15. The base node is located at the center of the network.

We model the security behavior of each sensor node by two parameters, P_c and P_m. A sensor node is compromised with a percentage of P_c. The compromised node may misbehave, i.e. modify its sensing data, with a percentage of P_m. We vary the setting of P_c from 5% to 50%. P_m is fixed to 50%. To model the energy consumption of each node, we adopt the characteristics of MPR420CB type of MICA2 Mote [19]. Specifically, we set the current draw of transmitting to 25 mA, receiving to 8 mA, and majority vote to 1 mA. The current draw for sampling (E_{sp}) is the cost to read every sensor of the node. We vary it from 1 mA to 16 mA to model various sensor combinations.

We implement four sampler monitoring algorithms: MBN, CMS, HYB, and Group-based Intrusion Detection (GID) algorithm [12]. As noted in Section 2, GID is a representative algorithm to detect compromised nodes in the spatially clustered WSN. It partitions the cluster into equally-sized sub-groups, and each sub-group monitors the entire cluster in turn. In our implementation, the region has a role to the sub-group. Since GID does not consider the location of the sub-group, any region in the cluster can be selected as a monitoring group. However, the security performance must degrade significantly if the monitoring group is not located in the routing path between a sampler and the CH. To observe this drawback, we implement two versions of GID: GID-R and GID-A. GID-R implements the original version of GID, and it selects the monitoring group in a random manner. On the other hand, GID-A assumes that only four regions surrounding the CH are selected as a monitoring region because they can overhear any messages to the CH. Furthermore, for the fair comparison, we assume that each node in the monitoring group reports the misbehavior of samplers to the CH. Then the CH performs the majority vote similar to the MBN.

Table 1. Simulation parameters

Parameter	Meaning	Value
d	Number of sensor nodes in a region	$4 \sim 16$
N_{node}	Number of sensor nodes in WSN	$1024 \sim 4096$
N_{smp}	Number of samplers in a cluster	$3 \sim 15$
P_c	Percentage of compromised nodes	$5\% \sim 50\%$
P_m	Percentage of misbehavior	50%
E_{tx}	Current draw of transmitting	25 mA
E_{rv}	Current draw of receiving	8 mA
E_{vote}	Current draw of majority vote	1 mA
E_{sp}	Current draw of sampling	$1 \sim 16$ mA

The primary performance metrics are detection ratio, false alarms, and energy consumption. The *detection ratio* is the percentage of compromised samplers that can be successfully detected. The *false alarms* represent the total number of false positive errors where a normal sampler is claimed as a compromised one. The *energy consumption* is the average amount of energy that each sensor node spends during the simulation duration. It aggregates current draw for transmitting, receiving, and sampling. The simulation duration consists of 5000 data-sampling periods. The forced-sampling period happens for each of 32 data-sampling periods. We use a form of batch mean method for the statistical analysis of experiment results. Specifically, each value of the performance metric is given by averaging the results of 20 batches with different seeds.

EXPERIMENT RESULTS

Security Performance

Figure 3 shows the detection ratios of the sampler monitoring algorithms when we change the percentage of compromised nodes (P_c). For each region, the number of nodes (d) is set to 8 and the number of samplers (N_{smp}) is set to 7. Throughout the experiments of this section, we set the energy for sampling (E_{sp}) to 8 mA.

First of all, GID-R suffers from the lowest detection ratio. It can detect at most half of compromised samplers even when P_c is 5%. The detection ratio drops gradually as P_c increases. The reason is that GID-R does not consider the location of the monitoring group. If the monitoring group is not in the routing path between a sampler and the CH, it cannot check the behavior of the sampler. GID-A performs

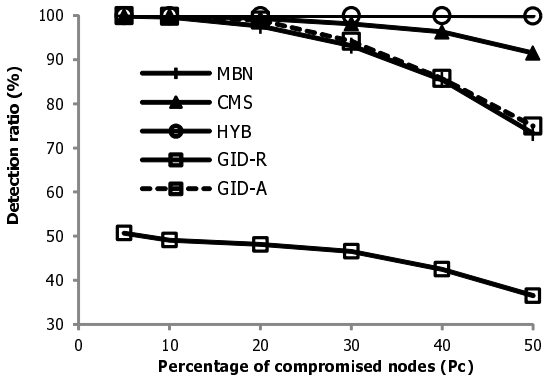

FIGURE 3. Detection ratio for varying P_c ($d = 8$, $N_{smp} = 7$, $E_{sp} = 8$mA)

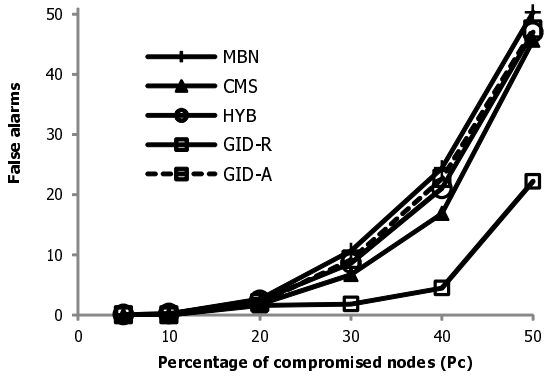

FIGURE 4. False alarms for varying P_c ($d = 8$, $N_{smp} = 7$, $E_{sp} = 8$mA)

much better than GID-R, since the monitoring group can overhear every message of the sampler.

When P_c is 5%, every proposed algorithm can detect the compromised samplers completely. However, both MBN and CMS perform worse as P_c increases. MBN can detect only 73% of the compromised samplers when P_c is 50%. Note that if majority of nodes in a region are compromised, MBN cannot detect compromised samplers of the region. We call such region as a *compromised region*. It is obvious that the number of compromised regions increases as P_c increases. This is why GID-A performs very similar to MBN. Since a monitoring group is set to a region, the number of compromised monitoring group must also increase in proportion to P_c.

CMS performs better than MBN at high P_c. It can detect compromised samplers in the compromised region if other samplers are normal. However, if majority of the samplers are compromised at the same time, CMS cannot detect them either. This is why the detection ratio of CMS drops to 90% when P_c is 50%. HYB performs best in this experiment. The

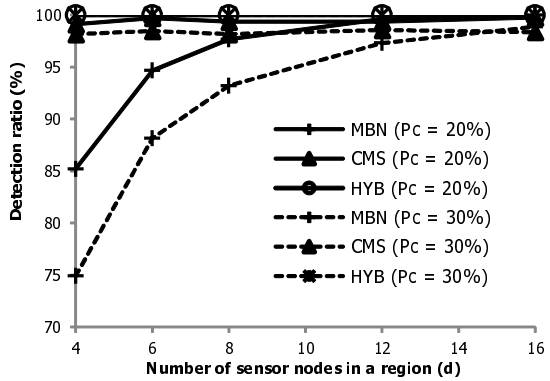

FIGURE 5. Detection ratio for varying d ($N_{smp} = 7$, $E_{sp} = 8$mA)

two-phase checking enables HYB to detect compromised samplers completely at every setting of P_c.

Figure 4 shows the false alarms of every algorithm at the same parameter setting. Once again, MBN performs worst at high P_c. This is due to the effect of compromised region. If any normal node is selected as a sampler in the compromised region, then majority of compromised neighbors may send fake report messages to the CH. Then the CH has to decide the sampler as compromised. As a result, normal nodes are hard to survive at the compromised region. This is why the false alarms of HYB are also higher than that of CMS. CMS performs best in this case since it does not suffer from the compromised region.

GID-R has much smaller false alarms. This result is not surprising, however. It is due to the fact that the monitoring group misses any messages sent by the unreachable samplers. A compromised node in the monitoring group cannot send any fake messages to the CH because it does not know which samplers report their sensing data. Once again, GID-A performs very similar to MBN. The monitoring group may behave as a compromised region at high P_c. Hereafter, we will not consider GID anymore since GID-R suffers from the very low detection ratio and the performance of GID-A is basically same to MBN.

Figure 5 shows the detection ratios of the proposed algorithms for various setting of d. When d is small, a few compromised nodes could be a majority in a region. As a result, the number of compromised regions must increase for small d. This is why MBN performs worst when d is between 4 and 8. The performance difference is significant when P_c is 30%, because there are more compromised regions. MBN performs better as d increases. Its performance is comparable to other algorithms when d is 16. On the other hand, the detection ratios of CMS and HYB are

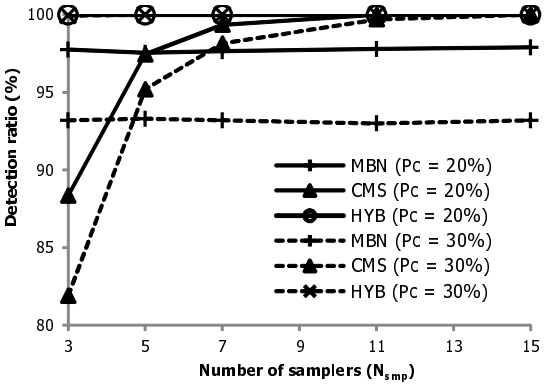

FIGURE 6. Detection ratio for varying N_{smp} ($d = 8$, $E_{sp} = 8$mA)

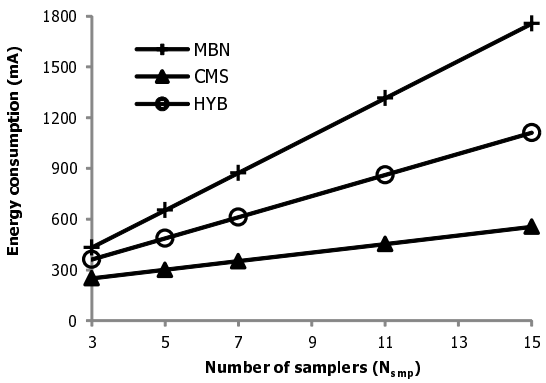

FIGURE 7. Energy consumption for varying N_{smp} ($d = 8$, $P_c = 20\%$, $E_{sp} = 8$mA)

almost constant throughout the experiment. Since they can detect compromised samplers with the help of other samplers, the number of neighbor nodes does not influence the security performance of both algorithms.

We also changed N_{smp} and compared the detection ratios of three algorithms. The experiment result appears at Figure 6. As expected, the security performance of CMS depends on N_{smp}. If N_{smp} is small, a few compromised samplers may win the majority vote. In this case, the detection ratio of CMS should be very low as a result. When N_{smp} is over 5, CMS can detect more than 95% compromised samplers. The detection ratio of MBN is nearly constant for every setting of N_{smp}.

It is interesting to note that the detection ratio of HYB is nearly 100% for every experiment. When N_{smp} is small, HYB can detect compromised samplers with the help of neighbor nodes. On the other hand, when d is small, it makes up for the security weakness with other samplers. Even though we do not show in the graph, HYB can detect over 93% compromised samplers when both parameters are set to small and half of sensor nodes are compromised, i.e. $d = 4$, $N_{smp} = 3$, and $P_c = 50\%$. It means that the two-phase checking of HYB is a very strong mechanism to detect compromised nodes.

Energy Performance

In this section, we evaluate the energy performance of every algorithm. Figure 7 shows the energy consumption of three algorithms when N_{smp} is changed from 3 to 15. MBN consumes energy the most due to the *sampling cost* of neighbor nodes and the *propagation cost* of sensing data from samplers to the base node. When N_{smp} is large, both

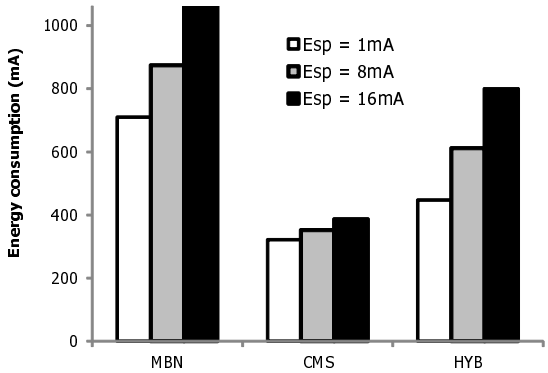

FIGURE 8. Energy consumption for varying E_{sp} ($d = 8$, $P_c = 20\%$, $N_{smp} = 7$)

the sampling cost and the propagation cost must increase significantly at MBN. CMS has the best energy performance. Its energy consumption is only 31% of MBN and 50% of HYB when N_{smp} is 15. Since neighbor nodes do not sample at each data-sampling period, CMS can reduce the sampling cost considerably. Furthermore, CMS can also reduce the propagation cost since the CH forwards only one sensing data of normal samplers to the base node. The sampling cost of HYB is equal to that of MBN and it has the same propagation cost to CMS. As a result, the energy consumption of HYB lies between MBN and CMS.

Figure 8 shows the energy consumption when E_{sp} is changed. Both MBN and HYB consume much energy when E_{sp} is large, but the energy consumption drops significantly for small E_{sp}. This means that the sampling cost is the major component of their energy consumption. On the other hand, E_{sp} has little influence on CMS. CMS consumes most of energy to propagate the sensing data.

We also observed the energy consumption of the CH. Figure 9 shows the experiment result. At CMS and HYB,

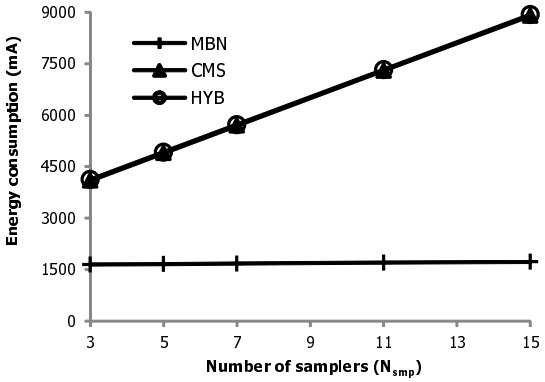

FIGURE 9. Energy consumption of CH for varying N_{smp} ($d = 8$, $P_c = 20\%$, $E_{sp} = 8$mA)

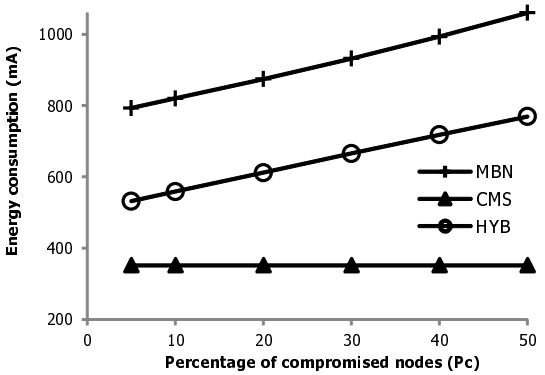

FIGURE 10. Energy consumption for varying P_c ($d = 8$, $N_{smp} = 7$, $E_{sp} = 8$mA)

the CH consumes much energy compared to MBN. The reason is that both algorithms require the CH to receive the sensing data of every sampler. This is why the energy consumption is in proportion to N_{smp}. On the other hand, the CH of MBN receives only the report messages of neighbor nodes for security reason. As a result, its energy consumption is rather constant throughout the experiment. It consumes most of energy to receive the sensing data of every node at the forced-sampling period.

As the last experiment, we changed the setting of P_c and compared the energy consumption. Figure 10 shows the experiment result. Both MBN and HYB spend energy in proportion to P_c. If P_c is high, more samplers and their neighbor nodes would be compromised. As a result, more neighbor nodes will send report messages to the CH just because their samplers are compromised or they try to send fake reports to attack the normal samplers. The energy consumption of CMS remains constant since it does not rely on the monitoring by neighbor nodes.

Concluding Remarks

The WSN is vulnerable to security threats due to unreliable wireless channels, unattended operation of sensor nodes, and resource constraint. In this paper, we proposed three algorithms to detect compromised nodes in the WSN. They are monitoring by neighbors (MBN), cooperation of multiple samplers (CMS), and a hybrid algorithm of MBN and CMS (HYB). Every algorithm is based on spatial clustering and tries to detect compromised nodes in an energy-efficient manner. MBN follows the traditional watchdog approach, while CMS leverages the strong correlation property of the spatial clustering. HYB integrates MBN and CMS.

We compared the performance of the proposed algorithms under a wide variety of WSN configurations and different levels of security attack. The primary results obtained from the experiments can be summarized as follows.

1. HYB can detect the compromised samplers completely for most of the experiments. CMS can also detect over 95% compromised samplers when each cluster has more than three samplers. On the other hand, MBN performs worst and the detection ratio drops under 80% for sparse networks where the number of neighbor nodes is small.
2. CMS outperforms other algorithms significantly with regard to the energy consumption. At the best case, its energy consumption is only 31% of MBN and 50% of HYB. MBN consumes energy the most due to the high cost of sampling and propagation. The energy consumption of HYB lies between MBN and CMS. This means that CMS or HYB can be an energy-efficient solution to defend against node compromised attacks in a spatially clustered WSN.
3. At CMS and HYB, the CH consumes more energy. To prolong the lifetime of WSN, the role of CH should be distributed evenly among sensor nodes using the traditional CH rotation algorithms.

REFERENCES

[1] Gedik, B., Liu, L., and Yu, P. 2007, “ASAP: An adaptive sampling approach to data collection in sensor networks," IEEE Trans. Parallel and Distributed Syst., 18(2), pp. 1766–1783.
[2] Cheng, H., Su, Z., Xiong, N., and Xiao, Y., 2016,

"Energy-efficient node-scheduling algorithms for wireless sensor networks using Markov Random Field model", information sciences, 329, pp. 461–4777.

[3] Cho, H., 2011, "Distributed multidimensional clustering based on spatial correlation in wireless sensor networks," Computer Syst. Science and Eng., 26(4), pp. 275–283.

[4] Liu, Z., Xing, W., Zeng, B., Wang, Y., and Lu, D., 2013, "Distributed spatial correlation-based clustering for approximate data collection in WSNs," Proc. IEEE 27th Int. Conf. Advanced Info. Networking and Applications, pp. 56–63.

[5] Meka, A., and Singh, A., 2006, "Distributed spatial clustering in sensor networks," Proc. Int. Conf. Extending Database Tech.

[6] Villas, L., Boukerche, A., Oliveira, H., Araujo, R., and Loureiro, A., 2014, "A spatial correlation aware algorithm to perform efficient data collection in wireless sensor networks," Ad Hoc Networks, 12, pp. 69–85

[7] Abduvaliyev, A., Pathan, A-S., Zhou, J., Roman, R., and Wong, W-C., 2013, "On the vital areas of intrusion detection systems in wireless sensor networks," IEEE Commun. Surv. & Tutorials, 15(3), pp. 1223–1237.

[8] Liu, F., Cheng, X., and Chen, D., 2007, "Insider attacker detection in wireless sensor networks," Proc. IEEE INFOCOM 2007, pp. 1937–1945.

[9] Wang, S-S., Yan, K-Q., Wang, S-C., and Liu, C-W., 2011, "An integrated intrusion detection system for cluster-based wireless sensor networks," Expert Syst. with Applications, 38(12), pp. 15234–15243.

[10] Farooqi, A., Khan, F., Wang, J., and Lee, S., 2013, "A novel intrusion detection framework for wireless sensor networks," Personal and Ubiquitous Comput., 17(5), pp. 907–919.

[11] Riecker, M., Biedermann, S., and El Bansarkhani, R., 2015, "Lightweight energy consumption-based intrusion detection system for wireless sensor networks", Int. J. Inf. Secur., 14(2), pp. 155–167.

[12] Li, G., He, J., and Fu, Y., 2008, "Group-based intrusion detection system in wireless sensor networks," Computer Commun., 31(18), pp. 4324–4332.

[13] Su, W-T., Chang, K-M., and Kuo, Y-H., 2007, "eHIP: An energy-efficient hybrid intrusion prohibition system for cluster-based wireless sensor networks," Computer Networks, 51(4), pp. 1151–1168.

[14] Stetsko, A., Folkman, L., and Matya, V., 2010, "Neighbor-based intrusion detection for wireless sensor networks," Proc. 6th Int. Conf. Wireless and Mobile Commun., pp. 420–425.

[15] Kumarage, H., Khalil, I., Tari, Z., and Zomaya, A., 2013, "Distributed anomaly detection for industrial wireless sensor networks based on fuzzy data modelling," J. Parallel Distrib. Comput., 73(6), pp. 790–806.

[16] Rajasegarar, S., Leckie, C., and Palaniswami, M., 2014, "Hyperspherical cluster based distributed anomaly detection in wireless sensor networks," J. Parallel Distrib. Comput., 74(1), pp. 1833–1847.

[17] Li, H., Li, K., Qu, W., and Stojmenovic, I., 2014, "Secure and energy-efficient data aggregation with malicious aggregator identification in wireless sensor networks," Future Generation Computer Syst., 37, pp. 108–116.

[18] Ozdemir, S., and Xiao, Y., 2009, "Secure data aggregation in wireless sensor networks: a comprehensive overview," Computer Networks, 53(12), pp. 2022–2037.

[19] http://www.datasheetarchive.com/MPR420CB-datasheet.html, 2015.