AICN: An Efficient Algorithm to Identify Compromised Nodes in Wireless Sensor Network

Rongxing Lu, Xiaodong Lin, Chenxi Zhang, Haojin Zhu, Pin-Han Ho and Xuemin (Sherman) Shen
Department of Electrical and Computer Engineering,
University of Waterloo, Waterloo, Ontario, Canada N2L 3G1
Email: {rxlu, xdlin, h9zhu, pinhan, xshen}@bbcr.uwaterloo.ca, c14zhang@engmail.uwaterloo.ca

***Abstract*—Wireless sensor networking is an emerging technology, which potentially supports many emerging applications for both civilian and military purposes, ranging from environmental monitoring to battlefield surveillance. However, since sensor nodes are inexpensive devices, which could be easily compromised and controlled by an adversary, the compromised nodes could report false sensed results and degrade the reliability of the whole network. Therefore, how to identify these compromised nodes in a wireless sensor network is a very important security issue. To solve this problem, we propose an efficient algorithm, called AICN, to logically identify the compromised nodes in an efficient and effective way. Based on the network reliability estimation (NRE), we also present its enhanced version to further improve the efficiency.**

***Keywords*: Wireless sensor network, security, identify compromised nodes**

I. Introduction

Wireless sensor networking has been subject to extensive research efforts in recent years, and has been well recognized as a ubiquitous and general approach for some emerging applications ranging from the civilian domain, such as environmental monitoring, to the military domain, such as battlefield surveillance [1]–[3]. A wireless sensor network (WSN) is usually composed of a large number of sensor nodes which use wireless links to perform distributed sensing tasks. Here, each sensor node is cheap with low battery power and computation capacity, but is equipped with sensing, data processing, and communicating components. Therefore, when a sensor node receives a certain query from the data collection unit, also known as *sink*, the sensor node will report its sensing results.

Since a WSN is usually deployed in a harsh environment and each low-cost sensor node could be easily compromised, the security becomes a major concern [4]–[22]. For example, when some sensor nodes are compromised and controlled by an adversary, they could behave abnormally and randomly report true / false sensing results to the *sink*, by which the reliability of the whole network is seriously degraded. Therefore, it will be a critical issue to develop an effective strategy for the *sink* to detect and identify these compromised nodes.

Assume that a WSN consists of n sensor nodes, where the compromised nodes are less than the normal nodes, and each normal node can know other nodes' trust status by adopting the similar techniques in [23], [24]. Then, a straightforward way for the *sink* to identify the compromised nodes is that the *sink* inquiries each senor node with $n-1$ queries on other nodes. In such a way, after total $n(n-1)$ queries, the *sink* can logically identify all compromised nodes based on these returned results. Although this approach is effective, but it is not efficient because the $n(n-1)$ queries could consume a significant amount of the sensor nodes' energy.

In this paper, in order to reduce the query number, we propose an efficient Algorithm to Identify Compromised Nodes (AICN) in wireless sensor networks. The main contributions of this paper lie in two aspects:

1) The proposed AICN algorithm can efficiently distinguish a compromised node in a wireless sensor network. To the best of our knowledge, this is the first effort in developing an effective algorithm for *sink* to identify compromised nodes;
2) By taking advantage of the network reliability estimation (NRE), we further investigate an enhanced approach in which the compromised node identification can be performed efficiently.

The remainder of this paper is organized as follows. In Section II, we describe the sensor network model, trust assumptions and identify our design goal. Then, we present the AICN algorithm in Section III, followed by its enhanced version in Section IV. We analyze the performance in Section V. Finally, we draw our conclusions in Section VI.

II. Preliminaries

In this section, we describe our sensor network model, provide a brief overview on the trust assumptions, and define our design goal.

A. Network Model

Without loss of generality, we consider a simple abstraction of a WSN consisting of a fixed *sink* $\mathcal{S}$ and a number of sensor nodes $\mathcal{N} = \{N_1, N_2, \cdots, N_n\}$, where $n \geq 3$, deployed at a remote area (around one or more hop neighborhood of the *sink*), as shown in Fig. 1. Note that such an abstraction can be taken as a building block of a large-scale WSN. The *sink* $\mathcal{S}$ is a data collection unit, which could be a powerful workstation with plentiful resources. However, the sensor nodes $\mathcal{N} = \{N_1, N_2, \cdots, N_n\}$ are inexpensive, low-power devices which have limited resources, memory space, computation capability, and communication bandwidth. Once

978-1-4244-2075-9/08/$25.00 ©2008 IEEE

such a WSN is deployed and the corresponding data paths are established, the *sink* $\mathcal{S}$ can inquire each sensor node $N_i \in \mathcal{N}$ with a certain query, and then the sensor node N_i reports its sensing results back to the *sink* $\mathcal{S}$ over the pre-defined path. Additionally, in order to provide confidentiality to the sensing results, a pre-shared key between each $N_i \in \mathcal{N}$ and $\mathcal{S}$ is assumed and used to encrypt/decrypt the sensing results. We assume that the compromised nodes will be excluded from the network if the nodes resist to forward data for the other nodes. While identifying the compromised nodes under a Denial of Service (Dos) attack is not our focus, we concentrate on the case where the compromised nodes still forward data but could falsely report the sensing results.

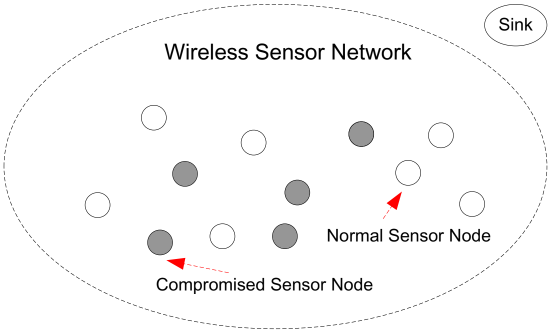

Fig. 1. Sensor network model under consideration

B. Trust Assumptions

Due to the different costs between the *sink* $\mathcal{S}$ and the sensor nodes $\mathcal{N} = \{N_1, N_2, \cdots, N_n\}$, the *trust assumptions* in the WSN are imbalance. Similar to [16]–[22], we present the following four *trust assumptions*.

- *Assumption 1:* The *sink* $\mathcal{S}$ equipped with tamper-proof devices is trustworthy and unassailable. All sensor nodes $\mathcal{N} = \{N_1, N_2, \cdots, N_n\}$ will be able to report their sensing results to the *sink* $\mathcal{S}$ through the pre-defined routes.
- *Assumption 2:* Each sensor node $N_i \in \mathcal{N}$ is inexpensive, and easily compromised by an adversary. Once the sensor node N_i is compromised and controlled by an adversary, N_i will behave abnormally and may randomly report true / false sensing results to *sink* $\mathcal{S}$ when it is inquired to do so. However, the normal sensor nodes, which have not been compromised, always behave normally and report true sensing results to *sink* $\mathcal{S}$ through the pre-configured path.
- *Assumption 3:* With the passing of time, n_c sensor nodes are compromised by an adversary, and form compromised-set $\mathcal{N}_c = \{N_1^*, N_2^*, \cdots, N_{n_c}^*\}$; and the rest n_d sensor nodes keep normal and form normal-set $\mathcal{N}_d = \{N_1^\dagger, N_2^\dagger, \cdots, N_{n_d}^\dagger\}$. Then, we will have $\mathcal{N} = \mathcal{N}_c \cup \mathcal{N}_d$. As in the most application scenarios, it is reasonable for us to assume the normal-set $\mathcal{N}_d$ is larger than the compromised-set $\mathcal{N}_c$. Then, we will have the following relations:
$$\begin{cases} n_c < n_d \\ n_c + n_d = n \end{cases} \tag{1}$$
- *Assumption 4:* In the WSN abstraction, the *sink* $\mathcal{S}$ can clearly know the network topology and properly reconfigure the data path connecting each sensor node, and each normal sensor node $N_i \in \mathcal{N}_d$ can exchange their local information and get to know all nodes' trust status within its communication range by adopting similar techniques in [23], [24].

C. Design Goal

Because the compromised nodes report the true / false sensing results randomly, the reliability of the whole network will be seriously impaired. Therefore, undoubtedly, it is of ultimate importance for the *sink* $\mathcal{S}$ to exactly know each sensor node's trust status, which will be addressed in this paper. Specifically, based on the above trust assumptions, we design a novel AICN algorithm along with its enhanced version to help the *sink* $\mathcal{S}$ to identify the compromised nodes. Compared with the straightforward approach where $n(n-1)$ queries are required, the proposed approaches are much more efficient.

III. Proposed AICN Algorithm

Based on the network model and trust assumptions described in the previous section, the proposed AICN aims to distinguish the compromised nodes from the normal ones with the lowest number of queries. Since AICN is firmly rooted on a useful lemma, we first present the lemma.

A. Lemma 1

Let $\mathcal{F}(x)$ be a function that denotes whether the sensor node x is compromised, which is 1 if x is a compromised node, and 0 otherwise. Formally, $\mathcal{F}(x)$ can be expressed as
$$\mathcal{F}(x) = \begin{cases} 1, & x \in \mathcal{N}_c; \\ 0, & x \in \mathcal{N}_d. \end{cases} \tag{2}$$

Let $\mathcal{J}(x, y)$ denote the event that the sensor node $x \in \mathcal{N}$ provides the recommendation on the status of another sensor node $y \in \mathcal{N}$. $\mathcal{J}(x, y) = 1$ indicates x considers y is a compromised node, and $\mathcal{J}(x, y) = 0$ shows x considers y is a normal node. Formally,
$$\mathcal{J}(x, y) = \begin{cases} 1, & x \text{ says } y \text{ is compromised;} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Lemma 1: $\mathcal{J}(x, y) = 1 \Longrightarrow (\mathcal{F}(x) \vee \mathcal{F}(y)) = 1$.

Proof: Based upon the above trust assumptions, a normal sensor node always tells the truth, and a compromised sensor node could tell the truth only with $\frac{1}{2}$ probability. Therefore, we can construct the following truth table for $\mathcal{J}(x, y)$:

$\mathcal{F}(x)$	$\mathcal{F}(y)$	$\mathcal{J}(x, y)$
0	0	0
0	1	1
1	0	1, 0
1	1	1, 0

From the truth table, it is clear to conduct that

$$\Pr\left[(\mathcal{F}(x) \vee \mathcal{F}(y)) = 1 | \mathcal{J}(x,y) = 1\right] = 1. \tag{4}$$

Thus, the proof is completed. ■

This lemma seems straightforward but is very useful in the development of the proposed AICN which shows that if the event $\mathcal{J}(x,y) = 1$ occurs, then either x or y or both are compromised nodes.

B. Description of AICN Algorithm

As stated in *trust assumptions*, the *reliability* of a wireless sensor network $\mathcal{N} = \{N_1, N_2, \cdots, N_n\}$ will decrease as time increases. In this subsection, we will present the proposed AICN in detail, which can help the *sink* $\mathcal{S}$ to distinguish the compromised nodes $\mathcal{N}_c = \{N_1^*, N_2^*, \cdots, N_{n_c}^*\}$ from the normal nodes $\mathcal{N}_d = \{N_1^\dagger, N_2^\dagger, \cdots, N_{n_d}^\dagger\}$ in an efficient and effective way.

Based on the *trust assumptions* in section II-B, the proposed AICN can be described in the following steps.

Step 1: The *sink* $\mathcal{S}$ places all wireless sensor nodes $\mathcal{N} = \{N_1, N_2, \cdots, N_n\}$ in Set A, renumbers them, and forms them into an array, $N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow \cdots \rightarrow N_{n-1} \rightarrow N_n$, as shown in Fig. 2. The right neighbor and left neighbor of each node $N_i \in \mathcal{N}$ are as follows:

$$\textit{RightNeighbor}(N_i) = \begin{cases} N_{i+1}, & 1 \le i \le n-1; \\ N_1, & i = n. \end{cases} \tag{5}$$

$$\textit{LeftNeighbor}(N_i) = \begin{cases} N_n, & i = 1; \\ N_{i-1}, & 2 \le i \le n. \end{cases} \tag{6}$$

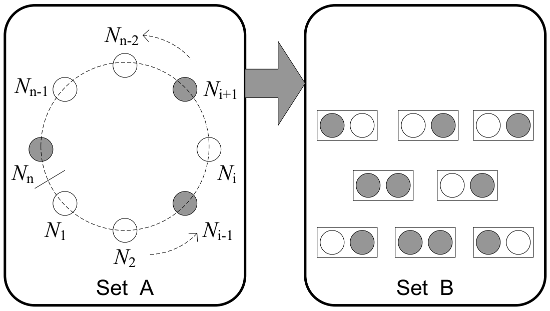

Fig. 2. Possible compromised nodes are moved from set A to set B

Step 2: Starting from the sensor node N_1, the *sink* $\mathcal{S}$ inquiries each node's opinion on its right neighbor *one by one*, and the corresponding inquired node will report its opinion back to the *sink* $\mathcal{S}$. The report of node N_i is subject to the following format:

$$\begin{aligned}&\mathcal{J}(N_i, \textit{RightNeighbor}(N_i)) \\ &= \begin{cases} 1, & N_i \text{ believes its right-neighbor is compromised;} \\ 0, & \text{otherwise.} \end{cases}\end{aligned} \tag{7}$$

Suppose the sensor node N_j is the first one who reports

$$\mathcal{J}(N_j, \textit{RightNeighbor}(N_j)) = \mathcal{J}(N_j, N_{j+1}) = 1 \tag{8}$$

Then, according to Lemma 1, we know

$$(\mathcal{F}(N_j) \vee \mathcal{F}(N_{j+1})) = 1 \tag{9}$$

In this case, the *sink* $\mathcal{S}$ runs the following operations:

1) Move the pair-sensor (N_j, N_{j+1}) from Set A to Set B, as shown in Fig. 2.
2) Choose N_j's old left-neighbor

$$\textit{LeftNeighbor}(N_j) = N_{j-1} \tag{10}$$

N_{j+1}'s old right-neighbor

$$\textit{RightNeighbor}(N_{j+1}) = N_{j+2} \tag{11}$$

and set N_{j-1}'s new right-neighbor as N_{j+2}, that is

$$\textit{RightNeighbor}(N_{j-1}) = N_{j+2} \tag{12}$$

3) Inquiry N_{j-1}'s opinion on its new right-neighbor N_{j+1}.

Following the operations in Step 2, the *sink* $\mathcal{S}$ continues to inquire the sensor nodes until reaching the rear of array.

Step 3: In the end, after $(n-1)$ queries, there are β pair-sensors in Set B, and there are α $(\alpha \ge 1)$ sensor nodes in Set A such that $\alpha + 2\beta = n$. According to the operations in Step 2, each node in Set A believes its right-neighbor is a normal node, and each left node in Set B trusts that its right-neighbor is a compromised node.

In order to illustrate the rest of the algorithm, we first introduce the following four facts. Let n_{ac}, n_{ad} be the number of the compromised nodes and the normal nodes in Set A, n_{bc}, n_{bd} be the number of the compromised nodes and the normal nodes in Set B, respectively.

Fact 1: $n_{bc} \ge n_{bd}$, the number of compromised nodes is larger than or equal to the number of normal nodes in Set B.

Proof: According to Lemma 1, when $\mathcal{F}(x,y) = 1$, we will have $(\mathcal{F}(x) \vee \mathcal{F}(y)) = 1$, which means at least one of each sensor pair in Set B is compromised. Therefore, it is clear that $n_{bc} \ge n_{bd}$. ■

Fact 2: $n_{ac} < n_{ad}$, the number of compromised nodes is less than the number of normal nodes in Set A.

Proof: According to Assumption 3, the following relations holds:

$$\begin{cases} n_c < n_d \\ n_c + n_d = n \\ n_c = n_{bc} + n_{ac} \\ n_d = n_{bd} + n_{ad} \end{cases} \tag{13}$$

Thus, we have

$$n_{bc} + n_{ac} < n_{bd} + n_{ad} \tag{14}$$

According to Fact 1:

$$n_{bc} \ge n_{bd} \Rightarrow n_{bc} - n_{bd} \ge 0 \tag{15}$$

Based on Eqs. (14)-(15), we can conclude that

$$n_{ad} - n_{ac} > n_{bc} - n_{bd} \ge 0 \Rightarrow n_{ac} < n_{ad} \tag{16}$$

Therefore, the number of compromised nodes is less than the number of normal nodes in Set A. ■

Fact 3: If $\alpha = 1$, the only sensor node in Set A must be a normal node.

Proof: According to Fact 2, the relation in Set A is described as follows:

$$\begin{cases} n_{ac} < n_{ad} \\ n_{ac} + n_{ad} = \alpha \end{cases} \tag{17}$$

When $\alpha = 1$, we must have

$$\begin{cases} n_{ad} = 1 \\ n_{ac} = 0 \end{cases} \tag{18}$$

Therefore, if $\alpha = 1$, then the only sensor node in Set A must be normal. ■

Fact 4: If $\alpha \geq 2$, then in Set A, the last two sensor nodes in the array must be normal.

Proof: Based on the operations in Step 2, each node in Set A believes its right-neighbor is a normal node. We can further conclude that the $\left[\frac{\alpha}{2}\right] + 1$ sensor nodes in the rear of array must be normal nodes, as shown in Fig. 3 (a). If not, when the case in Fig. 3 (b) occurs, the left normal node must report its right compromised node. Then the length of array is less than α, which will contradict with assumption that the array length is α. Therefore, the case (b) will not occur.

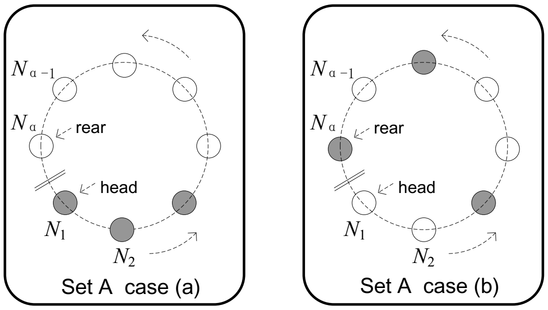

Fig. 3. After $n-1$ queries, case (a) is the correct array in Set A and case (b) is the impossible array in Set A

As a result, all compromised nodes are located at the head of the array, the number of which is less than $\left[\frac{\alpha}{2}\right]$, and the $\left[\frac{\alpha}{2}\right] + 1$ nodes in the rear of the array must be the normal nodes. Specifically, when $\alpha \geq 2$, the last two sensor nodes in the array are normal nodes. ■

On the basis of the above four facts, the *sink* $\mathcal{S}$ performs the following operations:

1) If $\alpha = 1$, then the only sensor node in Set A is a normal node. Then, the *sink* $\mathcal{S}$ inquires the node by launching a number of $\beta = \frac{n-1}{2}$ queries on β pair-sensors in Set B. Note that in this case, only one node in each pair-sensor is compromised in Set B; otherwise, it will contradict with $n_c < n_d$. Therefore, after totally $n - 1 + \frac{n-1}{2} = \frac{3n-3}{2}$ queries, the *sink* $\mathcal{S}$ can distinguish all the compromised nodes $\mathcal{N}_c$ from the normal ones $\mathcal{N}_d$.
2) If $\alpha \geq 2$, the last two nodes in the array are normal ones. The *sink* $\mathcal{S}$ can inquire the last node with at most $n-2$ queries on other sensor nodes' status. Therefore, in this case, after launching at most $n-1+n-2 = 2n-3$ queries, the *sink* $\mathcal{S}$ can distinguish all the compromised nodes $\mathcal{N}_c$ from the normal nodes $\mathcal{N}_d$.

By observing the above two cases, it can be sufficiently concluded that the *sink* $\mathcal{S}$ can judge the status of each sensor node within at most $2n-3$ queries.

C. Illustration Example

We use an example to exhibit the effectiveness of AICN where $n = 3$. Based on the Eq. (1) in *Assumption* 3, we can conclude that the number of the compromised nodes n_c equals to 0 or 1. Thus, there are $\binom{3}{0} + \binom{3}{1} = 4$ cases, as shown in Fig. 4. In the following, we discuss all cases and show that the *sink* $\mathcal{S}$ can judge the status of each sensor node by at most $2n-3 = 3$ queries.

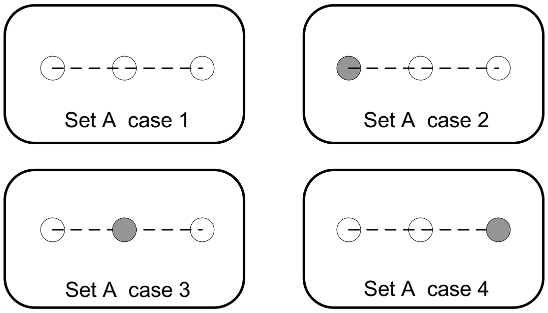

Fig. 4. Four possible cases of the wireless sensor network when $n = 3$

Case 1: $N_1 \in \mathcal{N}_d$, $N_2 \in \mathcal{N}_d$, $N_3 \in \mathcal{N}_d$.

1) **Query:** $sink \to N_1$ **on** N_2

$$N_1 \in \mathcal{N}_d \longrightarrow \mathcal{J}(N_1, N_2) = 0$$

2) **Query:** $sink \to N_2$ **on** N_3

$$N_2 \in \mathcal{N}_d \longrightarrow \mathcal{J}(N_2, N_3) = 0$$

Based on the Fact 4, the *sink* knows N_2, N_3 are normal nodes, and inquiries N_3 for the status of N_1.

3) **Query:** $sink \to N_3$ **on** N_1

$$N_3 \in \mathcal{N}_d \longrightarrow \mathcal{J}(N_3, N_1) = 0$$

Because N_3 is known as a normal node, the *sink* believes that N_1 is also normal.

Case 2: $N_1 \in \mathcal{N}_c$, $N_2 \in \mathcal{N}_d$, $N_3 \in \mathcal{N}_d$.

1) **Query:** $sink \to N_1$ **on** N_2

$$N_1 \in \mathcal{N}_c \longrightarrow \mathcal{J}(N_1, N_2) = 1 \quad \text{with } 1/2 \text{ probability}$$

(N_1, N_2) will be moved to Set B, then N_3 is the only sensor node in Set A. Based on Fact 3, the *sink* knows N_3 is normal, and inquires N_3 on the status of N_1, N_2.

2) **Query:** $sink \to N_3$ **on** N_1

$$N_3 \in \mathcal{N}_d \longrightarrow \mathcal{J}(N_3, N_1) = 1$$

Because N_3 is known as a normal node, the *sink* believes that N_1 is a compromised node. Furthermore, the *sink* knows N_2 is a normal node. Otherwise, if N_2 is also compromised, then it will contradict with the assumption $n_c < n_d$.

1) **Query:** $sink \rightarrow N_1$ **on** N_2

$$N_1 \in \mathcal{N}_d \longrightarrow \mathcal{J}(N_1, N_2) = 0 \quad \text{with } 1/2 \text{ probability}$$

2) **Query:** $sink \rightarrow N_2$ **on** N_3

$$N_2 \in \mathcal{N}_d \longrightarrow \mathcal{J}(N_2, N_3) = 0$$

Based on the Fact 4, the *sink* knows N_2, N_3 are normal nodes, and inquiries N_3 for the status of N_1.

3) **Query:** $sink \rightarrow N_3$ **on** N_1

$$N_3 \in \mathcal{N}_d \longrightarrow \mathcal{J}(N_3, N_1) = 1$$

Because N_3 is known as a normal node, the *sink* believes that N_1 is a compromised node.

Case 3: $N_1 \in \mathcal{N}_d$, $N_2 \in \mathcal{N}_c$, $N_3 \in \mathcal{N}_d$.

1) **Query:** $sink \rightarrow N_1$ **on** N_2

$$N_1 \in \mathcal{N}_d \longrightarrow \mathcal{J}(N_1, N_2) = 1$$

(N_1, N_2) will be moved to Set B, then N_3 is the only sensor node in Set A. Based on Fact 3, the *sink* knows N_3 is normal, and inquires N_3 on the status of N_1, N_2.

2) **Query:** $sink \rightarrow N_3$ **on** N_1

$$N_3 \in \mathcal{N}_d \longrightarrow \mathcal{J}(N_3, N_1) = 0$$

Because N_3 is known as a normal node, the *sink* believes that N_1 is also normal. Furthermore, based on Lemma 1, the *sink* knows N_2 must be a compromised node.

Case 4: $N_1 \in \mathcal{N}_d$, $N_2 \in \mathcal{N}_d$, $N_3 \in \mathcal{N}_c$.

1) **Query:** $sink \rightarrow N_1$ **on** N_2

$$N_1 \in \mathcal{N}_d \longrightarrow \mathcal{J}(N_1, N_2) = 0$$

2) **Query:** $sink \rightarrow N_2$ **on** N_3

$$N_2 \in \mathcal{N}_c \longrightarrow \mathcal{J}(N_2, N_3) = 1$$

(N_2, N_3) will be moved to Set B, then N_1 is the only sensor node in Set A. Based on the Fact 3, the *sink* knows N_1 is normal, and inquires N_1 on the status of N_2, N_3.

3) **Query:** $sink \rightarrow N_1$ **on** N_2

$$N_1 \in \mathcal{N}_d \longrightarrow \mathcal{J}(N_1, N_2) = 0$$

Because N_1 is known as a normal node, the *sink* believes that N_2 is also a normal node. Furthermore, based on the Lemma 1, the *sink* knows N_3 must be a compromised node.

IV. Enhanced AICN Algorithm

The AICN provides an efficient and effective way for the *sink* $\mathcal{S}$ to identify compromised nodes. In this section, with an additional assumption, we introduce a novel approach for enhancing AICN, which can further reduce the number of queries. Assume that after a WSN $\mathcal{N} = \{N_1, N_2, \cdots, N_n\}$ is deployed, each sensor node $N_i \in \mathcal{N}$ could be compromised by an adversary with probability p_c, where $0 < p_c < 0.5$[1]. Then, we will have the following optimal strategy on AICN.

Let X be a random variable denoting the number of sensor nodes that could be compromised among $\mathcal{N}$. Then, we have

$$\Pr(X = x) = \binom{n}{x} p_c^x (1 - p_c)^{n-x} \qquad (19)$$

and

$$\mathsf{E}[X] = np_c \quad \text{and} \quad \mathsf{Var}[X] = np_c(1 - p_c) \qquad (20)$$

$\mathsf{E}[X] = np_c$ is called network reliability estimation (NRE), which shows that the average np_c nodes could be compromised among total n sensor nodes. When the *sink* $\mathcal{S}$ randomly picks $2 \cdot np_c + 1$ nodes from $\mathcal{N}$, the normal nodes among $2 \cdot np_c + 1$ nodes will be larger than the compromised ones.

Therefore, the *sink* $\mathcal{S}$ only picks $2 \cdot np_c + 1$ nodes from $\mathcal{N}$ and runs AICN in Section III, it could distinguish the compromised nodes first from $2 \cdot np_c + 1$ nodes with at most

$$2(2 \cdot np_c + 1) - 3 = 4 \cdot np_c - 1 \qquad (21)$$

queries, and plus

$$n - (2 \cdot np_c + 1) \qquad (22)$$

queries for the rest sensor nodes. Therefore, the total queries for the *sink* $\mathcal{S}$ to identify the compromised nodes is at most

$$4 \cdot np_c - 1 + n - (2 \cdot np_c + 1) = (1 + 2p_c) \cdot n - 2 \qquad (23)$$

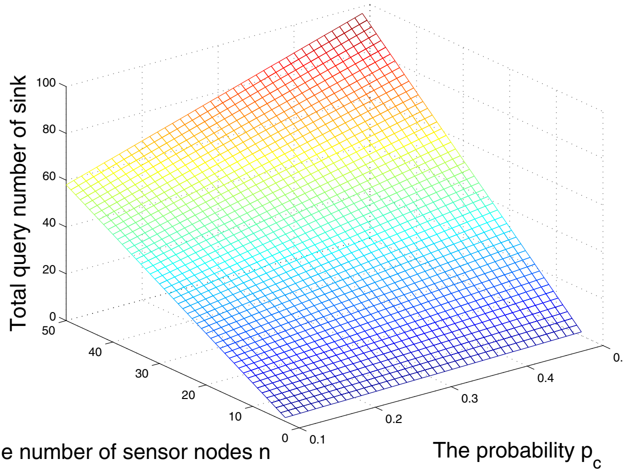

Fig. 5. Total query number of sink vs. total number of sensor nodes n with respect to different compromised probability p_c

Fig. 5 illustrates how the total number of queries by the *sink* $\mathcal{S}$ varies with the number of total sensor nodes n and the probability p_c, where $3 \leq n \leq 50$ and $0.1 \leq p_c < 0.5$. It is clearly shown that the network reliability estimation np_c will efficiently reduce the total number of queries. For example, when $n = 50$, $p_c = 0.2$, the *sink* $\mathcal{S}$ only requires 68 queries to know all sensor nodes' status.

[1] The compromised probability p_c should be less than 0.5. Otherwise, it will contradict with the assumption $n_c < n_d$ in the whole sensor network.

V. Performance Analysis

To demonstrate the performance improvement and efficiency, we have conducted extensive analysis to compare the proposed AICN and its enhanced version with the straightforward approach. Without loss of generality, we assume that the considered WSN has n sensor nodes. To identify the compromised nodes out of the totally n sensor nodes, the straightforward approach requires $n(n-1)$ queries, while the AICN algorithm requires $2n-3$ queries, and the enhanced version requires $(1+2p_c)\cdot n-2$ queries. We define the efficiency ratio (ER) as

$$\mathsf{ER}_{\mathrm{AICN}} = \frac{2n-3}{n(n-1)} \tag{24}$$

for comparing the AICN algorithm with the straightforward approach, and define

$$\mathsf{ER}_{\mathrm{EAICN}} = \frac{(1+2p_c)\cdot n-2}{n(n-1)} \tag{25}$$

for comparing the enhanced AICN with the straightforward approach. Fig. 6 shows the $\mathsf{ER}_{\mathrm{AICN}}$, $\mathsf{ER}_{\mathrm{EAICN}}$ vary with the number of sensor nodes n, where $3 \le n \le 50$. We can see from the figure that the efficiency of AICN and its enhanced version will improve with the increase of n. Furthermore, the less the NRE np_c, the more efficient the enhanced AICN can achieve.

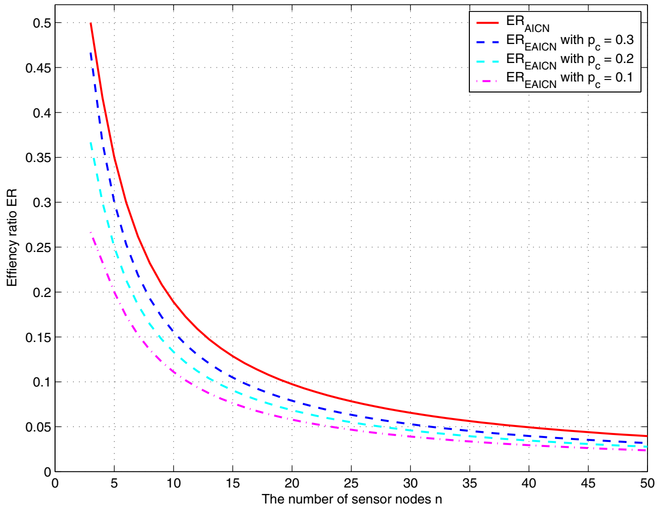

Fig. 6. The efficiency ratio $\mathsf{ER}_{\mathrm{AICN}}$, $\mathsf{ER}_{\mathrm{EAICN}}$ vary with the number of sensor nodes n, where $3 \le n \le 50$

VI. Conclusions

In this paper, we introduced AICN along with its enhanced version, aiming to identify compromised nodes in WSNs. From our extensive analysis, both AICN and its enhanced version are effective and more efficient than the straightforward approach. In our future work, we will extend our study in the actual WSNs, and integrate the AICN algorithm with robust private-preserving data aggregation schemes.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survery on sensor networks”, *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102-116, 2002.

[2] I. Akyildiz and I. Kasimoglu, “Wireless sensor and actor networks: research challenges”, *Ad Hoc Networks*, Vol. 2, No. 4, pp. 351-367, 2004.

[3] I. Akyildiz and E. Stuntebeck, “Wireless underground sensor networks: research challenges”, *Ad Hoc Networks*, Vol. 4, No. 6, pp. 669-686, 2006.

[4] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. Culler “SPINS: security protocols for sensor networks”, *Wireless Networks*, Vol. 8, No. 5, pp. 521-534, 2002.

[5] W. Du, J. Deng, Y. Han, and P. Varshney, “A key predistribution scheme for sensor networks using deployment knowledge”, *IEEE Transactions on Dependable and Secure Computing*, Vol. 3, No. 1, pp. 62-77, 2006.

[6] A. Perrig, J. Stankovic, and D. Wagner, “Security in wireless sensor networks”, *Communications of the ACM*, Vol. 47, No. 6, pp. 53-57, 2004.

[7] H. Chan, A. Perrig, and D. Song, “ Random key predistribution schemes for sensor networks”, in *Proc. Symposium on Security and Privacy 2003*, May 2003, pp. 197-213.

[8] S. Zhu, S. Setia, and S. Jajodia, “LEAP: efficient security mechanisms for large-scale distributed sensor networks”, in *Proc. ACM CCS 2003*, October 2003, pp. 62-72.

[9] B. Przydatek, D. Song, and A. Perrig, “SIA: secure information aggregation in sensor networks”, in *Proc. Conference on Embedded Networked Sensor System 2003*, November 2003, pp. 255-265.

[10] X. Lin, H. Zhu, B. Lin, P.-H. Ho, and X. Shen, “ A novel voting mechanism for compromised node revocation in wireless ad hoc networks”, in *IEEE Global Communications Conference (GLOBECOM’06)*, San Francisco, Nov. 27-Dec. 1, 2006.

[11] X. Lin, P.-H. Ho, and X. Shen, “Towards compromise-resilient localized authentication architecture for wireless mesh networks”, in *Proc. of QShine 2007*, Vancouver, British Columbia, August 14 - 17, 2007.

[12] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh, “Toward resilient security in wireless sensor networks”, in *Proc. ACM Mobihoc 2005*, May 2005, pp. 34-45.

[13] F. Ye, H. Luo, S. Lu, and L. Zhang, “Statistical en-route filtering of injected false data in sensor networks”, in *Proc. IEEE Infocom 2004*, March 2004, pp. 2446-2457.

[14] W. Du, R. Wang, and P. Ning, “An efficient scheme for authenticating public keys in sensor networks”, in *Proc. ACM Mobihoc 2005*, May 2005, pp. 58-67.

[15] W. Zhang, H. Song, S. Zhu, and G. Cao, “Least privilege and privilege deprivation: towards tolerating mobile sink compromises in wireless sensor networks”, in *Proc. ACM Mobihoc 2005*, May 2005, pp. 378-389.

[16] W. Yu and K. Liu, “Secure cooperative mobile ad hoc networks against injecting traffic attacks”, in *Proc. IEEE SECON 2005*, September 2005, pp. 55-64.

[17] W. Zhang, S. Das, and Y. Liu, “ A trust based framework for secure data aggregation in wireless sensor networks”, in *Proc. IEEE SECON 2006*, September 2006.

[18] F. Delgosha and F. Fekri, “ Threshold key-establishment in distributed sensor networks using a multivariate scheme”, in *Proc. IEEE Infocom 2006*, April 2006.

[19] M. Miller and N. Vaidya, “ Leveraging channel diversity for key establishment in wireless sensor networks ”, in *Proc. IEEE Infocom 2006*, April 2006.

[20] X. Lin, R. Lu, H. Zhu, P.-H. Ho, X. Shen and Z. Cao, “SRPAKE: An Anonymous Secure Routing Protocol with Authenticated Key Exchange for Wireless Ad Hoc Networks”, in *Proc. IEEE ICC 2007*, June 2007.

[21] K. Ren, W. Lou, and Y. Zhang, “LEDS: providing location-aware end-to-end data security in wireless sensor networks ”, in *Proc. IEEE Infocom 2006*, April 2006, pp. 1-12.

[22] K. Ren, W. Lou, and Y. Zhang, “Multi-user broadcast authentication in wireless sensor networks”, in *Proc. IEEE SECON 2007*, June 2007.

[23] A. P. da Silva, M. Martins, B. Roacha, A. Loureiro, L. Ruiz, and H. Wong, “Decentralized intrusion detection in wireless sensor networks”, in *ACM Q2SWinet’05*, 2005.

[24] Y. Huang and W. Lee, “A cooperative intrusion detection system for ad hoc networks”, in *ACM SASN’03*, October 2003.