Locating Compromised Data Sources in IoT-Enabled Smart Cities: A Great-Alternative-Region-Based Approach

Locating Compromised Data Sources in IoT-enabled Smart Cities: A Great-Alternative-Region-based Approach

Ming Tao, Kaoru Ota, *Member, IEEE,* Mianxiong Dong, *Member, IEEE*

***Abstract*—Sensing devices acting as interconnected data sources are becoming increasingly ubiquitous in concepts of IoT-enabled smart cities, but they typically lack physical protection and are susceptible to being compromised. To address this issue, a great-alternative-region (GAR)-based approach for deploying network monitors to locate compromised data sources is proposed. The GAR concept is introduced according to the network topology and connectivity characteristics, and the GARs with the most complete connectivity are identified as the candidate monitor locations, thereby transforming the problem of monitor deployment into a traditional K-center problem. Based on the demonstrated relationship between the monitor locations and the locating accuracy, the optimization objective for reasonably deploying monitors is designed to minimize the maximum number of hops between the data sources and their nearest monitors, and the optimal deployment pattern is achieved using an improved genetic algorithm. Finally, simulation-based results are presented to illustrate the performance of this approach.**

***Index Terms*—Smart city, monitor deployment, compromised data source, great alternative region, genetic algorithm.**

I. Introduction

Smart cities have become an important research field not only because of their important and varied application scenarios but also because of their well-addressed fundamental infrastructures [1]. Recently, Internet of Things (IoT), as a supporting technology, has played an irreplaceable role in smart city design. Typically, via efficient deployment, numerous resource-constrained sensing devices, also called data sources, can be deployed to construct an interconnected network, therein promising drastically enhanced capabilities for automatic data collection and exploring physical phenomena from surrounding environments by dynamically interacting with human activities and/or machine systems [2]. However, such sensing devices are usually deployed in open areas, where these unattended data sources lack physical protection. The open nature of deploying these data sources, with its thus-far-unresolved security issues, provides adversaries or attackers with opportunities to intrude upon such data sources and perform a variety of malicious activities, such as tampering with data and disseminating harmful information. Meanwhile, the dissemination of much harmful information also results in the problem of uneven energy consumption [3]–[6], especially, for battery-powered sensing devices, which has cascading effects with disastrous results. Botnets of Things, with its foreseeable damage potential, has been investigated as a challenging academic and industrial topic [7]. Therefore, the ability to locate compromised data sources in a timely manner is important in effectively ensuring the network performance of an IoT-enabled smart city [8], [9].

Because of the unpredictability of attacks, a commonly used method in related proposals is to deploy monitors in a network formed by interconnected data sources to collect the disseminated information and thus to determine the compromised data sources based on the collected local information and the network topology. The locating efficiency primarily depends on the adopted monitor deployment strategy, i.e., random deployment [10], betweenness-centrality-based deployment [11], degree-centrality-based deployment [10], [12] and rumor-centrality-based deployment [12]. Although these approaches have been found to be effective in some scenarios, they require the complete subgraph of information dissemination in the network of interest to be obtained, which is difficult to achieve. Hence, the problem of how to deploy network monitors such that compromised data sources can be located in an optimal manner remains a challenging issue, and to the best of our knowledge, this problem has been investigated to a much lesser extent. However, deployment problems have generally received considerable attention in recent decades and are widely and well investigated in other fields of research [13]–[17]. These proposals typically adopt similar methodologies. The many constraints and objectives involved are first systematically considered as a basis for the formulation of single-objective or multi-objective models. Then, these models are solved using intelligent algorithms to achieve optimal deployment.

Motivated by these previous proposals in various research fields, this paper proposes a heuristic monitor deployment scheme for locating compromised data sources to address the realistic security challenge facing IoT-enabled smart cities via the following main contributions.

This work was supported in part by National Natural Science Fund, China (Grant No. 61300198), Guangdong University Scientific Innovation Project (Grant No. 2017KTSCX), outstanding young teacher training program of Education department of Guangdong province (Grant No. YQ2015158), Guangdong Provincial Science & Technology Plan Projects (Grant No. 2016A010101035), JSPS KAKENHI (Grant Nos. JP16K00117 & JP15K15976) and KDDI Foundation.

Ming Tao is with School of Computer Science and Network Security, Dongguan University of Technology, Dongguan 523808, China and Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan (Email: ming.tao@mail.scut.edu.cn)

Kaoru Ota is with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan (Email: ota@csse.muroran-it.ac.jp)

Mianxiong Dong (corresponding author) is with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan (Email: mx.dong@csse.muroran-it.ac.jp)

1. Based on an analysis of the network topology and connectivity, the network area formed by interconnected data sources is divided into different alternative regions (ARs), and the new concept of great alternative regions (GARs) is introduced to investigate the problem of monitor deployment. In this context, the GARs are treated as the candidate monitor locations, thereby transforming the addressed problem into a traditional K-center problem.

2. The relationship between the monitor deployment locations and the accuracy of locating compromised data sources is first investigated. Then, an optimization objective is formulated to minimize the maximum number of hops between the data sources and their nearest monitors for reasonably deploying a set of monitors in the GARs.

3. An improved genetic algorithm is proposed to solve the problem to achieve optimal deployment. Analytical and simulation-based results for four different representative synthetic networks with different connectivity characteristics are presented to demonstrate the effectiveness and efficiency of the proposed approach.

The remainder of this paper is organized as follows. Section II introduces the AR generation principle and the GAR concept. Section III presents the design of the optimization objective ensuring the reasonable deployment. In Section IV, an improved genetic algorithm is proposed to achieve optimal deployment. In Section V, the experimental setup and analysis results are addressed. Finally, we summarize and conclude the paper in Section VI.

II. Determining Candidate Locations for Monitor Deployment

A. Definitions

Without loss of generality, in a given network area of R^2, assuming that $V = \{v_1, v_2, ..., v_n\}$ is the set of deployed data sources, (x_i, y_i) represents the plane coordinates of node v_i, and r is the valid node signal coverage radius. If the Euclidean distance $l_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ between any two nodes v_i and v_j satisfies $l_{ij} \leq r$, then the two nodes act as neighbors and can communicate directly, and the line linking these two adjacent nodes is treated as an edge e_i. In accordance with these adjacency relations, the deployed nodes constitute a limited undirected communication network, represented by $G(V, E)$, where V is the set of nodes and E is the set of generated edges.

Based on the above definitions, Fig. 1(d) presents an illustration introducing the AR concept, where v_i and v_j cannot engage in direct communication. To connect them, a node acting as a relay must be deployed in the overlapping region of signal convergence, i.e., the shaded area in Fig. 1(d). This overlapping area can be treated as an AR, where v_i and v_j are the generating nodes and $AR = \langle v_i, v_j \rangle$. Accordingly, an AR is a convergence region in which a deployed monitor can directly communicate with all the generating nodes. Note that the monitor can be deployed in any location in the AR with the same connectivity and that the valid signal coverage area of the deployed monitor is a coverage circle of r, which can cover all nodes generating the AR.

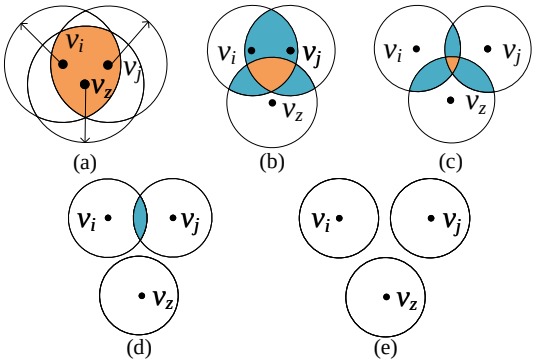


Fig. 1. Illustrations of the generation principle of ARs.

Based on the definition of an AR, we can formulate a general conclusion for the generation principle of the ARs in a scenario with m nodes. For a set of nodes $\{v_1, v_2, ..., v_m\}$, if all Euclidean distances between pairs of nodes satisfy $l \leq r$, then the generated AR is the overlapping area of signal coverage, as shown in Fig. 1(a). If at least one Euclidean distance between a pair of nodes satisfies $r < l \leq 2r$ and no Euclidean distance between any two nodes satisfies $l > 2r$, then although the nodes in $\{v_1, v_2, ..., v_m\}$ generate several ARs, only a monitor deployed in $AR = \langle v_1, v_2, ..., v_m \rangle$ can simultaneously communicate with all nodes in the set, as shown in Fig. 1(b) and (c). If any Euclidean distance between a pair of nodes satisfies $l > 2r$, then there is no AR in which a deployed monitor could connect to all nodes simultaneously, as shown in Fig. 1(d) and (e).

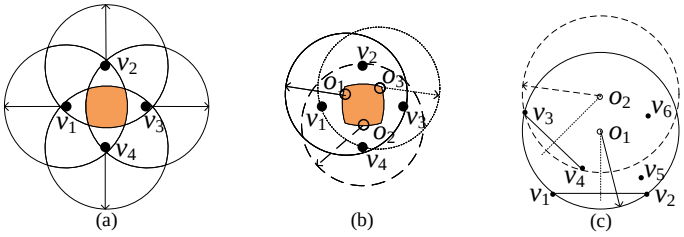


Fig. 2. Illustrations of coverage circles and the process for determining a coverage circle.

Because the valid signal coverage area of a monitor deployed at any location in an AR is a coverage circle of r, multiple coverage circles of this type are possible. As shown in Fig. 2, the shaded area corresponding to $AR = \langle v_1, v_2, v_3, v_4 \rangle$ is generated by the set of nodes $\{v_1, v_2, v_3, v_4\}$. Regardless of where the monitor is deployed in the AR, e.g., at o_1, o_2 or o_3, the generated coverage circle covers all four generating nodes. Interestingly, as shown in Fig. 2(b), for the generated coverage circles, we find that there is at least one coverage circle whose boundary will cross two of the generating nodes. In other words, two generating nodes may define a coverage circle covering all generating nodes. Although the circle defined by any two generating nodes might not necessarily be a coverage circle, there will always be a pair of generating nodes that can define a coverage circle to determine an AR. As shown in Fig. 2(c), for the $AR = \langle v_1, v_2, v_3, v_4, v_5, v_6 \rangle$, because the circle O_2 determined by the two generating nodes v_3 and v_4

does not cover v_1, v_2 and v_5, it is not a coverage circle. However, the circle O_1 determined by the two generating nodes v_1 and v_2 can be regarded as a coverage circle because the circle covers all generating nodes.

Based on the above analysis, the concepts of boundary points and boundary chords are introduced to more explicitly describe the generating principles for ARs. Specifically, any two nodes separated by a Euclidean distance that satisfies $l \leq 2r$ are boundary points, and the line linking these two nodes is a boundary chord. Two boundary points can generate two symmetrical coverage circles with respect to the boundary chord, which can be regarded as potential ARs. However, because the number of boundary chords is large in complex communication networks, a huge number of potential ARs can exist. If we directly use the AR-based approach to solve the addressed problem, it will undoubtedly increase the solving complexity. Considering some of the generated potential ARs have the same connectivity, we introduce the following two rules to eliminate unnecessary ARs.

Rule **1**. For the two potential ARs generated by any two boundary points, if the set of nodes covered by one AR is a subset of that covered by another AR, then the AR with the larger coverage area is retained.

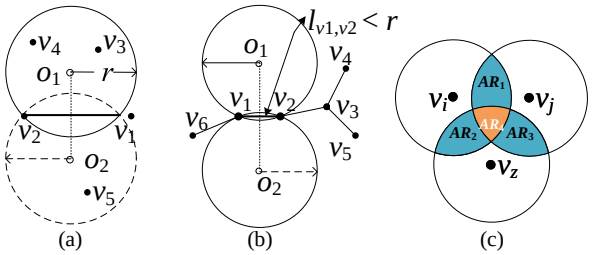


Fig. 3. Illustrations describing the elimination of unnecessary ARs.

As shown in Fig. 3(a), two symmetrical coverage circles, O_1 and O_2, are generated by the two boundary points v_1 and v_2; in turn, these coverage circles determine two potential ARs: $AR_{O_1} = \langle v_1, v_2, v_3, v_4 \rangle$ and $AR_{O_2} = \langle v_1, v_2, v_5 \rangle$. Because the set of nodes covered by one AR is not a subset of that covered by another AR, both AR_{O_1} and AR_{O_2} are retained. If there is no v_5, then the set of nodes covered by AR_{O_2} will be a subset of that covered by AR_{O_1}; thus, only O_1 is a coverage circle, and AR_{O_1} is retained.

Rule **2**. If a boundary chord conforms to $l \leq r$ and neither of the two generated symmetrical coverage circles covers any nodes other than these two boundary points, then both potential ARs are unnecessary and can be eliminated.

As shown in Fig. 3(b), the two symmetrical coverage circles O_1 and O_2 generated by v_1 and v_2 do not cover any nodes other than the two boundary points. Because a monitor can be deployed at either node while maintaining connectivity, the two determined potential ARs are unnecessary.

Using these two rules, unnecessary ARs are eliminated. However, the retained ARs may still have some overlapping connectivity. It is therefore necessary to introduce the GAR concept to identify the ARs with the most complete connectivity, thereby reducing the complexity of solving the addressed problem. In Fig. 3(c), several ARs exist: $AR_1 = \langle v_i, v_j \rangle$, $AR_2 = \langle v_i, v_z \rangle$, $AR_3 = \langle v_j, v_z \rangle$ and $AR_4 = \langle v_i, v_j, v_z \rangle$. Clearly, a monitor deployed in AR_4 will provide all the connection functionality of the monitors deployed in AR_1, AR_2 and AR_3, that is, AR_4 has the most complete connectivity. Thus, AR_4 can be regarded as a GAR. Accordingly, the concept of a GAR is derived as follows. Let $AR = \{AR_1, AR_2, ..., AR_m\}$ be the set of ARs in a given plane area. If AR_i satisfies (1), that is, if the set of the generating nodes of AR_i is not a proper subset of that of any other ARs in the set, then AR_i can be regarded as a GAR.

$$\exists AR_i \subseteq AR, \forall AR_j \subseteq AR(j \neq i), AR_i \not\subset AR_j \qquad (1)$$

B. Identifying Great Alternative Regions (GARs)

Based on the above definitions and analyses, the procedure for identifying GARs is summarized as follows.

1. After traversing the plane network area of interest, if there is a pair of nodes separated by a Euclidean distance that satisfies $l \leq 2r$, then these two nodes are regarded as boundary points, and the line linking them is regarded as a boundary chord.

2. The determined boundary points and boundary chord generate two symmetrical coverage circles, which are regarded as potential ARs. Then, a binary matrix $AR = \{a_{ij}\}_{m \times n}$ is created to record the ultimately retained ARs by applying *Rule* **1** and *Rule* **2**, where m is the number of generated ARs and n is the number of nodes. If AR_i covers the j-th node, then a_{ij}=1; otherwise, a_{ij}=0, that is, for the i-th row in the matrix, the node with a_{ij}=1 is the generating node for AR_i.

3. In accordance with the definition of a GAR, if AR_i satisfies (1), then AR_i is regarded as a GAR.

C. Generating the New Complete Network

After identifying the GARs, because a monitor deployed at any location in a GAR will have the same connectivity, each GAR can be abstracted as a virtual node and inserted into the original network alongside the existing nodes to generate a new complete network. To effectively express the topological relations between the existing nodes and these virtual nodes and to maintain the structural integrity of the newly generated network, the node connectivity must be revised as follows.

The connections between existing nodes are two-way connections, whereas the connection directed from an existing node to a virtual node is one-way connection. Moreover, because any given virtual node might not be ultimately chosen as a monitor location, to avoid the routing of data through such virtual nodes, no connections are established between virtual nodes. After these revisions, a new complete network with the set of nodes $V' = \{v_1, v_2, ...v_n, v_{GAR1}, ...\}$, in which the GARs are treated as the candidate monitor locations, is generated. Thus, the addressed problem of monitor deployment is transformed into a traditional K-center problem.

Note that a monitor deployed in a GAR will manage all the generating nodes of that GAR. In a given network, regardless of the locations of compromised data sources, if

the optimal monitor deployment is achieved using the GAR-based approach, it will facilitate the identification of any compromised data sources.

III. Optimization Objective and Problem Formulation

In the classical data propagation model, the data propagation delay is directly determined by the propagation distance. Based on the assumption that each data source will diffuse data along the shortest path to the destination, Pinto et al. [10] proposed a method for estimating the locations of data sources. In this method, for any candidate data source, if the actual delay in receiving the diffused data at each monitor is very close to the theoretical delay calculated from the data propagation model, then this candidate data source is identified as the true source with a high probability. However, because of the difficulty in acquiring the initial data diffusion moment, the theoretical delay in receiving the diffused data at each monitor cannot be directly obtained. Motivated by this difficulty, Pinto et al. attempted to determine the true data source by comparing the vectors of the actual delay and the theoretical delay of receiving the diffused data at each monitor. In this paper, to reasonably deploy a set of monitors in the GARs to enhance the locating accuracy of compromised data sources, we draw on the method proposed by Pinto et al. to deduce the optimization objective.

Given a network $G(V, E)$, the nodes in V are all likely to be true data sources. At some point in time, if a set of monitors denoted by $\{m_1, m_2, ..., m_k\}$ has received data diffused by any candidate data source and the set of actual delays of first receiving these data at the k monitors is represented by $\{t_1, t_2, ..., t_k\}$, then the vector of the actual delays of the data reception at the k monitors can be written as $\vec{v} = \{v_1, v_2, ..., v_{k-1}\}^T$, where $v_i = t_{i+1} - t_1$ is the delay difference between monitor m_{i+1} and monitor m_1. In the classical data propagation model, according to the central limit theorem, it is generally assumed that the propagation delay on each network edge can be approximated as a normal distribution represented by $d \sim N(\mu, \sigma^2)$. Therefore, for any candidate data source, the set of theoretical delays in data reception at the k monitors is represented by $\{t'_1, t'_2, ..., t'_k\}$, where $t'_i = \mu \cdot |h(s, m_i)|$, with $|h(s, m_i)|$ being the hop count along the shortest path from the candidate data source s to m_i. Accordingly, the vector of the theoretical delays in data reception at the k monitors can be written as $\vec{v'} = \{v'_1, v'_2, ...v'_{k-1}\}^T$, where $v'_i = t'_{i+1} - t'_1 = \mu \cdot (|h(s, m_{i+1})| - |h(s, m_1)|)$. Then, as shown in (2), a multivariate normally distributed probability density, represented by $\hat{s}$, is employed to calculate the similarity between $\vec{v}$ and $\vec{v'}$. In this equation, for the deployed monitors, if $i = j$, then $[\Lambda_s]_{i,j} = \sigma^2 \cdot |h(m_1, m_i)|$; otherwise, $[\Lambda_s]_{i,j} = \sigma^2 \cdot |h(m_1, m_i) \cap h(m_1, m_j)|$. Among all candidate data sources, the source with the maximum $\hat{s}$ is identified as the true data source.

$$\hat{s} = \exp[-\frac{1}{2}(\vec{v} - \vec{v'})^T \Lambda_s^{-1} (\vec{v} - \vec{v'})] / \sqrt{|\Lambda_s|} \tag{2}$$

From the representation of $\vec{v'}$, we can clearly see that the differences in the hop counts along the shortest paths between the candidate data source and the monitors serve as the basis for calculating $\vec{v'}$. For any candidate data source, when these differences are increased, the similarity between $\vec{v}$ and $\vec{v'}$ is higher. Thus, the probability of correctly determining the true data source is increased, resulting in a higher locating accuracy. The relationship between the hop-count difference and the delay-vector similarity is analyzed in the following.

Assuming that $M = \{m | m \in M\}$ is the set of deployed monitors and $S = \{s | s \in V\}$ is the set of candidate data sources. $h(s_i, m)$ is a path from a candidate data source s_i to a monitor m, and $|h(s_i, m)|$ is the hop count. Based on the assumption that the propagation delay on a network edge e_i is independent identically distributed and can be approximated as a normal distribution $d(e_i) \sim N(\mu, \sigma^2)$ [10], for any two monitors m_i and m_j, the value of $\overline{d}$ is defined in (3) as the arithmetic mean of $d(e_i)$, $e_i \in (h(s_i, m_i) \cup h(s_i, m_j))$. Then, the expectation $E(\overline{d})$ and the variance $D(\overline{d})$ can be deduced as shown in (4) and (5), respectively.

$$\overline{d} = \frac{|\sum_{e_i \in h(s_i, m_j)} d(e_i) - \sum_{e_i \in h(s_i, m_i)} d(e_i)|}{|h(s_i, m_j)| - |h(s_i, m_i)|} \tag{3}$$

$$\begin{aligned} E(\overline{d}) &= \frac{\sum_{e_i \in h(s_i, m_j)} d(e_i) - \sum_{e_i \in h(s_i, m_i)} d(e_i)}{|h(s_i, m_j)| - |h(s_i, m_i)|} \\ &= \frac{|h(s_i, m_j)| \cdot \mu - |h(s_i, m_i)| \cdot \mu}{|h(s_i, m_j)| - |h(s_i, m_i)|} = \mu \end{aligned} \tag{4}$$

$$\begin{aligned} D(\overline{d}) &= \frac{\sum_{e_i \in h(s_i, m_j)} D(d(e_i)) + \sum_{e_i \in h(s_i, m_i)} D(d(e_i))}{\left(|h(s_i, m_j)| - |h(s_i, m_i)|\right)^2} \\ &= \frac{|h(s_i, m_j)| \cdot \sigma^2 + |h(s_i, m_i)| \cdot \sigma^2}{\left(|h(s_i, m_j)| - |h(s_i, m_i)|\right)^2} \\ &= \frac{|h(s_i, m_j)| + |h(s_i, m_i)|}{\left(|h(s_i, m_j)| - |h(s_i, m_i)|\right)^2} \cdot \sigma^2 \end{aligned} \tag{5}$$

Based on the principle of Chebyshev's inequality, the formula (6) is deduced, where ε is an arbitrary positive integer.

$$\begin{aligned} p(|\overline{d} - \mu| < \varepsilon) &\geq 1 - \frac{D(\overline{d})}{\varepsilon^2} \\ &= 1 - \frac{|h(s_i, m_j)| + |h(s_i, m_i)| \cdot \sigma^2}{\left(|h(s_i, m_j)| - |h(s_i, m_i)|\right)^2 \cdot \varepsilon^2} \end{aligned} \tag{6}$$

If $|h(s_i, m_j)| - |h(s_i, m_i)| \to \infty$, then $p(|\overline{d} - \mu| < \varepsilon) \to 1$, that is, as the hop-count difference between the paths to the two monitors approaches infinity, $\overline{d}$ approaches μ, and for s_i, $v \approx v'$ such that s_i would be a true data source. Based on the above analysis, we conclude that the locations of the deployed monitors affect the locating accuracy for the data sources. Accordingly, we now deduce the optimization objective for the problem of deploying a set of monitors in GARs to achieve enhanced accuracy in locating compromised data sources.

Given a monitor deployment strategy for the problem discussed in this paper, following the above assumptions, $\min_{m \in M}(|h(s_i, m)|)$ is the minimum hop count along the paths from s_i to any monitor in M, and H_M =

$\max\limits_{s\in S}\{\min\limits_{m\in M}(|h(s,m)|)\}$ is the maximum of these minimum hop counts for all candidate data sources. Suppose that m_i and m_j are the nearest monitors to any two candidate compromised data sources s_i and s_j, respectively. The hop counts between the candidate compromised data sources and their respective nearest monitors satisfy the following: $|h(s_i,m_i)| \le H_M$ and $|h(s_j,m_j)| \le H_M$. A triangle can be constructed using s_i, s_j and m_j, where $|h(s_i,m_j)| \ge |h(s_i,s_j)| - |h(s_j,m_j)|$ according to the geometrical relationship among the edges of a triangle. Accordingly, the sum of the differences between the hop counts from s_i to each other monitor (except the nearest monitor m_i) and the hop count from s_i to m_i, represented by $\mathrm{SUM}(|h(s_i,M)|)$, is calculated as shown in (7), where $|M|$ is the number of deployed monitors.

$$\begin{aligned} SUM(|h(s_i,M)|) &= \sum_{j=1,j\neq i}^{|M|}(|h(s_i,m_j)| - |h(s_i,m_i)|) \\ &\ge \sum_{j=1,j\neq i}^{|M|}(|h(s_i,s_j)| - |h(s_j,m_j)| - |h(s_i,m_i)|) \\ &= \sum_{j=1,j\neq i}^{|M|}|h(s_i,s_j)| - \sum_{j=1,j\neq i}^{|M|}|h(s_j,m_j)| \\ &\quad -(|M|-1)\cdot|h(s_i,m_i)| \end{aligned} \tag{7}$$

If AH is the average hop count between nodes, then because $|h(s_i,m_i)| \le H_M$ and $|h(s_j,m_j)| \le H_M$, we can write $\mathrm{SUM}(|h(s_i,M)|) \ge (|M|-1)(AH - 2H_M)$. Accordingly, given two different monitor deployment strategies with different sets of monitors M_1 and M_2, if $H_{M_1} < H_{M_2}$, then $\mathrm{SUM}(|h(s_i,M_1)|) > \mathrm{SUM}(|h(s_i,M_2)|)$. We now investigate the relationship between $\mathrm{SUM}(|h(s_i,M)|)$ and the locating accuracy, and then we deduce the optimization objective for monitor deployment.

If $\mathrm{SUM}(h(s_i,M_1)) > \mathrm{SUM}(h(s_i,M_2))$, then according to the above discussion on the relationship between the hop-count difference and the delay-vector similarity, we can write $p(|\overline{d_{M_1}} - \mu| < \varepsilon) > p(|\overline{d_{M_2}} - \mu| < \varepsilon)$, that is, the arithmetic mean value of the propagation delay in M_1 (denoted by $\overline{d_{M_1}}$) is much closer to μ than that in M_2. We can thus conclude that for any candidate compromised data source s_i, the actual delay in the reception of data diffused from s_i is closer to the theoretical delay for M_1. Thus, the locating accuracy of M_1 (denoted by P_{M_1}) is higher than that of M_2 for the location estimation method proposed by Pinto et al. [10].

$$\begin{aligned} min\ H_M &= \max_{s\in S}\{\min_{m\in M}(|h(s,m)|)\} \\ \text{s.t.}\quad &\sum_{i=1}^{N} x_{ij} = 1 \\ &\max\sum_{i=1}^{N}(\max_{j=1,\ldots,|V|}\{x_{ij}\}) = N \end{aligned} \tag{8}$$

According to the above discussion, if $H_{M_1} < H_{M_2}$, then $\mathrm{SUM}(|h(s_i,M_1)|) > \mathrm{SUM}(|h(s_i,M_2)|)$; consequently, $P_{M_1} > P_{M_2}$. Hence, as shown in (8), to achieve a higher locating accuracy for any compromised data sources, the optimization objective for monitor deployment should be designed to minimize the maximum of the minimum hop counts for all candidate data sources (H_M). $|V|$ is the number of deployed data sources, and N is the number of GARs (candidate monitor locations) in the newly generated complete network. Let x_{ij} and $\max\limits_{j=1,\ldots,|V|}\{x_{ij}\}$ be the decision variables. If v_j is served by monitor m_i, then x_{ij}=1; otherwise, x_{ij}=0. If GAR_i is actually set as a monitor, then $\max\limits_{j=1,\ldots,|V|}\{x_{ij}\} = 1$; otherwise, $\max\limits_{j=1,\ldots,|V|}\{x_{ij}\} = 0$. Accordingly, the first constraint on the optimization objective function ensures that each data source is served by only one monitor, and the second constraint ensures that the monitors are deployed in GARs and that the total number of deployed monitors is not greater than the number of generated GARs.

IV. Design of the Solution Algorithm

In terms of the introduced formulation, the problem of optimizing monitor deployment for locating compromised data sources using the proposed optimization objective has been proven to be NP-hard [18]. Here, an improved genetic algorithm (GA), the pseudo-code of which is shown in Algorithm 1, is proposed to solve this problem. The specific improvements are described below.

Algorithm 1 Improved GA for deployment optimization

Input: the newly generated complete network with GARs; the maximum number of monitors, N; the maximum number of iterations, max_g; the population size, N'; the crossover probability, $p_{crossover}$; the mutation probability, $p_{mutation}$; and the generation gap, GAP.
Output: the locations of the deployed monitors

```
Use the binary coding method to generate the initial population X(0) for the first generation g = 0; /* Initiation */
while (g ≤ max_g) do
    Use dynamic calibration to map the optimization objective to a fitness function, F(X) = G(X)_max − G(X) + ϖ^g;
    Perform crossover and mutation operations unconstrained by the two constraints on the optimization objective function;
    /* Crossover: with the crossover probability p_crossover.
    Execute the crossover operation for each selected parent pair (X_i, X_j) from X(g) using the double tangent point crossover operator;
    Generate a population of hybrid offspring, X_crossover;
    /* Mutation: with the mutation probability p_mutation.
    if (p_i > p_mutation)
    /* p_i is a randomly generated probability for the i-th individual in X_crossover */
        Execute the mutation operation for the i-th individual in X_crossover;
        Generate a population of mutated offspring, X_mutation;
    endif
    /* Selection: random-traversal-sampling-based selection.
    Calculate the fitness value for each individual in X_mutation;
    Select individuals with a selection probability of p(X_i) = F(X_i)/Σ_{i=1}^{n} F(X_i);
    /* reinsertion strategy.
    In accordance with the GAP, select individuals from X_mutation ⋃ X(g);
    /* Generate the next generation of the population.
    g = g + 1;
end while
```

(1) Using dynamic calibration, the optimization objective function is first mapped to a fitness function, $F(X) = G(X)_{max} - G(X) + \varpi^g$, where $G(X)$ is the objective function, g is the number of iterations, $G(X)_{max}$ is the maximum value of the objective function in each iteration, and ϖ ($\varpi < 1$) is a

positive number. Subsequently, during the process of solving the optimization objective, operations are performed without explicit concern for whether the two specified constraints on the objective are satisfied. Specifically, the first constraint is naturally satisfied in all operations on the fitness function. To ensure the satisfaction of the second constraint, a chromosome repair strategy is applied to ensure that the maximum number of monitors to be deployed is N and that the chromosomes generated in each generation of the population are optimal, thereby positively influencing the convergence.

(2) For the chromosome selection operation for each generation of the population, a strategy of random traversal sampling is employed. Specifically, based on the cumulative sum of the fitness vector, a random traversal sampling table is created, and the corresponding fitness value is selected according to a generated index. The index of the selected individual is determined by comparing the generated random value with the cumulative sum of the fitness vector. The selection probability of an individual X_i is represented by $p(X_i) = F(X_i)/\sum_{i=1}^{n} F(X_i)$, where $F(X_i)$ is the fitness value of individual X_i. Additionally, to avoid eliminating excellent individuals in each generation, a generation gap between each pair of adjacent generations is set as a fixed ratio, and a reinsertion strategy is employed to retain the best individuals of the parent generation in the offspring generation. Accordingly, via this combination of a random traversal sampling strategy and a reinsertion strategy, the inherent problems of premature convergence and low search efficiency in genetic algorithms are effectively eliminated.

V. Simulations and Analysis

A. Simulation Setup

The simulations were conducted on the MATLAB 2012a platform running on the executing host: 64-bit Windows 7 operating system, Intel(R) Core(TM) i5-3450 CPU @ 3.10 GHz, and 4 GB of RAM. Although the proposed GAR-based approach would have better performance in a network with a concentrated node distribution, not all real-world networks have such feature. To better verify the efficiency of the proposed approach for various applications, the nodes deployed in the test networks should be distributed in a fragmented but uniform manner, therein eliminating areas of both isolated and concentrated nodes. Following these guidelines, four different typical synthetic networks formed by interconnected data sources were generated using $NetworkX$: a random regular network, an Erdős-Rényi (ER) network, a WS network and a BA (scale-free) network. To reflect the large-scale node features in the discussed practical background of a smart city, 1×10^4 nodes were deployed in each test network.

The detailed characteristics of the four generated test networks are presented in Table I. $average\_degree$ is the average degree of the nodes; max_degree is the maximum degree, which reflects the edge density in the network; and $diameter$ is the network diameter, which is the maximum path length among all pairs of accessible nodes.

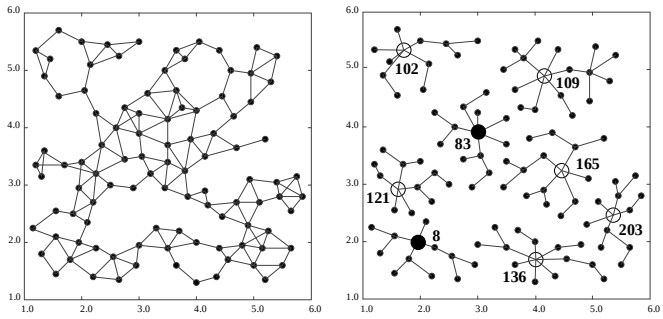
The simulation-based analyses presented below were conducted to illustrate the performance of the improved GA in solving the addressed problem and to demonstrate the locating accuracy of the proposed GAR-based approach.

TABLE I
Connectivity characteristics of the four test networks.

Network type	$average\_degree$	max_degree	$diameter$
random regular network	4	6	16
ER network	5	8	13
WS network	5	7	9
BA network	6	9	11

B. Algorithm Evaluation

To clearly observe the results of monitor deployment, as shown in Fig. 4(a), a 5×5 random regular network consisting of 100 randomly distributed nodes is discussed here as an illustration. With the stated guidelines for node distribution and connectivity, the radius of the valid signal coverage area for each node is 0.6. The parameters for the improved GA were set as follows: N'=500 (initial population size); max_g=300 (maximum number of iterations); $p_{crossover}$=0.9 and $p_{mutation}$=0.05 (crossover and mutation probabilities); and GAP=0.8 (generation gap). To clearly illustrate the effect of monitor deployment after one run of the algorithm, the result obtained when the number of deployed monitors was $|M| = 8$ is shown in Fig. 4(b), where the deployed monitors, indicated by solid dots, are existing nodes in the original network, whereas the monitors indicated by open circles are virtual nodes in the newly generated network with GARs.



(a) Network topology (b) Deployment result for $|M|$=8

Fig. 4. Illustration of the application of the GAR-based approach.

The sets of generating nodes for the GARs corresponding to each deployed monitor are shown in Table II. Although an existing node might appear in the sets of generating nodes of several GARs, it is covered by only a single monitor according to the corresponding constraint on the optimization objective. Additionally, the distribution of the nodes covered by the deployed monitors yield an acceptable performance in terms of load balancing. Fig. 5 is an illustration of the performance of load balance among the deployed monitors, which could be indicated by the distributions of the nodes covered by the deployed monitors. Here, the absolute value of deviation of the number of nodes covered by each monitor calculated by $|n_i - \overline{n}|/\overline{n}$ is adopted to represent the distributions of the covered nodes, where, n_i is the number of nodes covered by

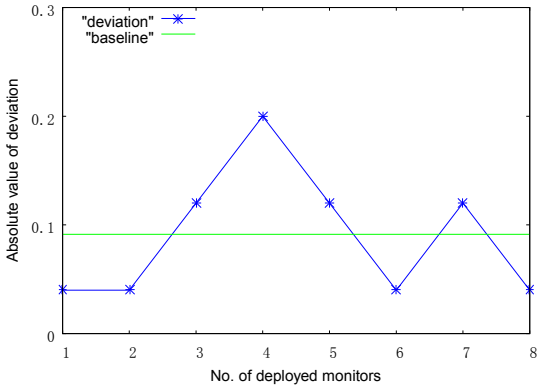


Fig. 5. The performance of load balance among the deployed monitors.

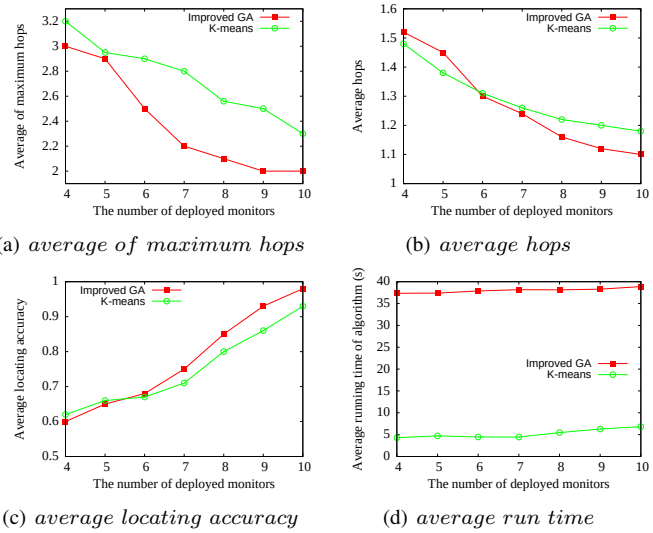
i-th monitor and $\overline{n}$ is the arithmetic mean value of n_i. From Fig. 5, we can clearly see that the initial nodes are relatively balanced covered.

TABLE II
SETS OF GENERATING NODES FOR THE GARS CORRESPONDING TO THE DEPLOYED MONITORS.

Location	Set of generating nodes for each GAR
8	4,5,7,8,9,10,14,15,16,17,18,19,34,39,51,52
83	12,13,42,43,44,67,68,71,78,80,81,82,83,84,86,88,90
102	1,2,3,6,11,12,13,22,24,30,31
109	42,43,60,62,63,64,65,71,78,81,82,83,91,93,94,95,96,97,98,99,100
121	16,52,64,68,70,73,75,76,77,81,85,87,89,90,92
136	17,18,19,40,41,45,49,51,54,56,61,66,69,72,74,79
165	21,23,25,27,28,34,37,38,44,45,48,50,52,53,57,58,59,60,62,63
203	20,23,26,28,29,33,35,36,41,45,46,47,55,56

To further convincingly demonstrate the efficiency of the improved GA, as shown in Fig. 6, a K-means clustering algorithm was also used to solve the problem of interest for comparison, with all reported results for the two algorithms averaged over 100 independent runs. Because the deployed nodes were all likely candidates for attack, 10 nodes were randomly selected in each run as the true compromised nodes diffusing malicious data.

For both compared algorithms, as the number of deployed monitors increases, the nodes can all be served by a monitor within an increasingly small range, leading to a gradual reduction in $average\ of\ maximum\ hops$ and $average\ hops$; however, these downward trends flatten out once a certain number of deployed monitors is reached. The improved GA solves the problem of interest based on the proposed GAR-based approach, and the set of monitors finally deployed in the GARs is determined by solving the proposed optimization objective. Therefore, the improved GA clearly outperforms the K-means algorithm in terms of $average\ of\ maximum\ hops$, and it is also superior in terms of $average\ hops$ for an increasing number of deployed monitors. According to the previously discussed relationship between the accuracy with which compromised data sources can be located and the monitor deployment locations, a reduction in $average\ hops$ corresponds to an increase in the average locating accuracy for both algorithms. However, because of its evident superiority in terms of $average\ hops$, the improved GA is also superior



(a) $average\ of\ maximum\ hops$ (b) $average\ hops$

(c) $average\ locating\ accuracy$ (d) $average\ run\ time$

Fig. 6. Efficiency comparisons of the two tested algorithms.

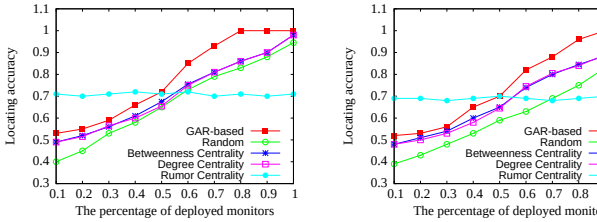
to the K-means algorithm in terms of locating accuracy.

From Fig. 6(d), the convergence time of the improved GA is longer than that of the K-means algorithm. Theoretical analysis revealed that the time complexity of the K-means algorithm is $O(|V| \cdot |M| \cdot max\_g)$, whereas the upper bound on the time complexity of the improved GA is $O((|V|-|M|)^2 \cdot |M| \cdot max\_g \cdot N')$. Therefore, the computational efficiency of the K-means algorithm is undoubtedly higher than that of the improved GA. However, from the presented comparison in terms of $average\ of\ maximum\ hops$ and $average\ hops$, it is clear that the K-means algorithm is prone to premature convergence when solving the addressed problem, whereas the improved GA effectively eliminates the inherent problems of premature convergence and low search efficiency, allowing the improved GA to more reliably converge to a global optimal solution. In short, although the improved GA has a longer convergence time, it produces superior results when solving the monitor deployment problem.

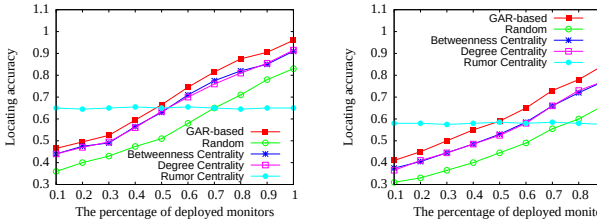
C. Comparison and Analysis of Locating Accuracy

To comprehensively demonstrate the locating accuracy of the proposed GAR-based approach, four different typical monitor deployment schemes, i.e., random deployment [10], betweenness-centrality-based deployment [11], degree-centrality-based deployment [10], [12] and rumor-centrality-based deployment [12], were selected for comparative studies. Note that the former three schemes and the GAR-based approach are all based on the method of estimating the locations of data sources proposed by Pinto et al. [10]. By contrast, as a baseline for comparison, the rumor-centrality-based scheme is based on the direct estimation of the locations of data sources, also known as rumor centers. Specifically, the rumor centrality is a 'graph score' function, as defined in [12]; in this approach, a positive number or score is assigned to each vertex, and the estimated rumor center is the vertex with the maximal score.

For a fair comparison, regardless of the deployment scheme used, the maximum number of deployed monitors must be less

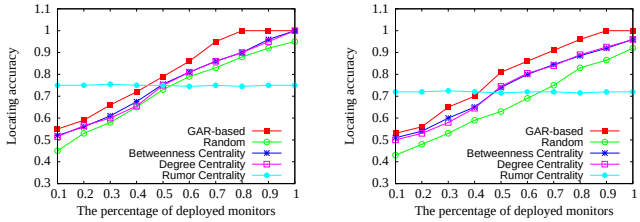


(a) $\gamma = 0.05$ (b) $\gamma = 0.1$

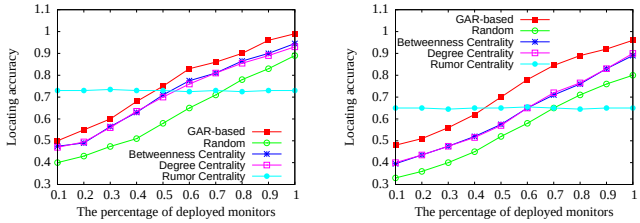


(c) $\gamma = 0.15$ (d) $\gamma = 0.2$

Fig. 7. Comparison of the locating accuracies in the random regular network.

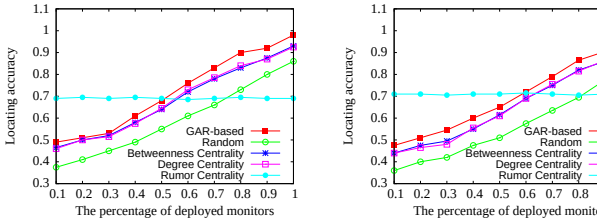


(a) $\gamma = 0.05$ (b) $\gamma = 0.1$

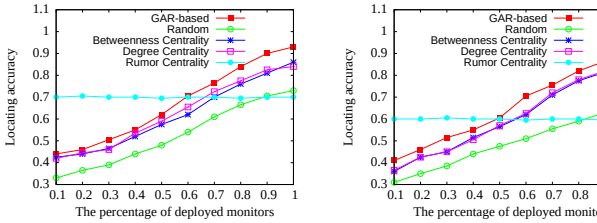


(c) $\gamma = 0.15$ (d) $\gamma = 0.2$

Fig. 9. Comparison of the locating accuracies in the WS network.

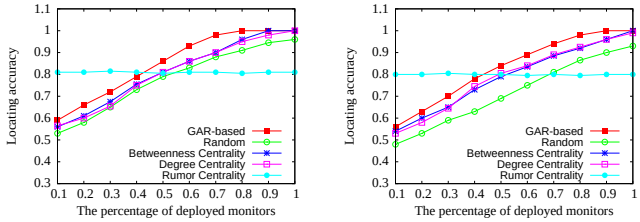


(a) $\gamma = 0.05$ (b) $\gamma = 0.1$

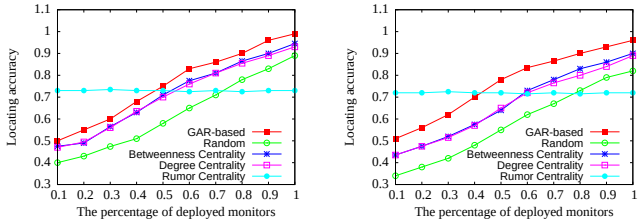


(c) $\gamma = 0.15$ (d) $\gamma = 0.2$

Fig. 8. Comparison of the locating accuracies in the ER network.



(a) $\gamma = 0.05$ (b) $\gamma = 0.1$



(c) $\gamma = 0.15$ (d) $\gamma = 0.2$

Fig. 10. Comparison of the locating accuracies in the BA network.

than or equal to the number of generated GARs. Therefore, in the simulations of the four test networks, we first deployed a group of monitors corresponding to a certain percentage of the number of generated GARs following each monitor deployment scheme. Because all nodes except those acting as monitors are likely candidates for attack, a percentage γ of these nodes were randomly selected as the compromised data sources, and analysis was performed to locate them and thus to determine the locating accuracy. The locating accuracy results for several values of γ on each test network are shown in Fig. 7-Fig. 10; all reported results for the five schemes were averaged over 100 independent runs.

In each test network with different percentages of compromised data sources, the coverage of the monitors becomes finer as the number of deployed monitors increases. The locating accuracies achieved by the four schemes using the method proposed by Pinto et al. [10] all increase with enhanced coverage. Simultaneously, because of the differences in network connectivity, the differences in locating accuracy are evident among the four test networks. From the comparisons, regardless of the deployment scheme used, the locating accuracy is much higher for the BA network, whose connectivity is superior, compared with the other three test networks.

Additionally, in each test network, if the percentage of deployed monitors is sufficient, all nodes will be covered in the GAR-based approach because of the manner in which the GARs are defined, i.e., based on the network topology and connectivity. By contrast, because of specific characteristics of the network connectivity, some nodes may remain uncovered by the deployed monitors in the random, betweenness-centrality-based and degree-centrality-based schemes. It is therefore possible for uncovered nodes to be chosen as attack targets, which would then be impossible to identify. Therefore, from the comparison of the results, the locating accuracy achieved by the GAR-based approach totally outperforms that achieved by the three other schemes.

VI. Conclusions

In this paper, the problem of locating compromised data sources in the sensing network of an IoT-enabled smart city was studied, and a GAR-based heuristic monitor deployment scheme was proposed to achieve effective identification. Simulation results for four typical synthetic networks with different connectivity characteristics show that the GAR-based approach is more effective and efficient than other representative schemes. In our future work, we would like to generalize the GAR-based approach to a richer variety of real network topologies. Additionally, the GAR-based approach was developed based on full network knowledge; another interesting issue would be to investigate the addressed problem with incomplete network knowledge.

References

[1] L. Filipponi, A. Vitaletti, G. Landi, V. Memeo, G. Laura, and P. Pucci, "Smart city: An event driven architecture for monitoring public spaces with heterogeneous sensors," in *Proceedings of IEEE 4th International Conference on Sensor Technologies and Applications (SENSORCOMM)*, 2010, pp. 281–286.

[2] D. S. Deif and Y. Gadallah, "Classification of wireless sensor networks deployment techniques," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 834–855, 2014.

[3] M. Dong, K. Ota, and A. Liu, "Rmer: Reliable and energy efficient data collection for large-scale wireless sensor networks," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 511–519, 2016.

[4] Y. Liu, M. Dong, K. Ota, and A. Liu, "Activetrust: Secure and trustable routing in wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2013–2027, 2016.

[5] M. Dong, K. Ota, A. Liu, and M. Guo, "Joint optimization of lifetime and transport delay under reliability constraint wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 1, pp. 225–236, 2016.

[6] J. Long, M. Dong, K. Ota, and A. Liu, "Reliability guaranteed efficient data gathering in wireless sensor networks," *IEEE Access*, vol. 3, pp. 430–444, 2015.

[7] B. Elisa and I. Nayeem, "Botnets and internet of things security," *Computer*, vol. 50, no. 2, pp. 76–79, 2017.

[8] J. Duan, W. Zeng, and M.-Y. Chow, "Attack detection and mitigation for resilient distributed dc optimal power flow in the iot environment," in *Proceedings of IEEE 25th International Symposium on Industrial Electronics (ISIE)*, 2016, pp. 606–611.

[9] A. Antonopoulos and C. Verikoukis, "Misbehavior detection in the internet of things: A network-coding-aware statistical approach," in *Proceedings of IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 1024–1027.

[10] P. C. Pinto, P. Thiran, and M. Vetterli, "Locating the source of diffusion in large-scale networks," *Physical review letters*, vol. 109, no. 6, 2012.

[11] W. L. Luo, W. P. Tay, and M. Leng, "How to identify an infection source with limited observations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 4, pp. 586–597, 2014.

[12] D. Shah and T. Zaman, "Rumors in a network: Who's the culprit?" *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5163–5181, 2011.

[13] M. Tao, S. Huang, Y. Li, M. Yan, and Y. Zhou, "Sa-pso based optimizing reader deployment in large-scale rfid systems," *Journal of Network and Computer Applications*, vol. 52, pp. 90–100, 2015.

[14] Y. Gong, M. Shen, J. Zhang, O. Kaynak, W. Chen, and Z. Zhan, "Optimizing rfid network planning by using a particle swarm optimization algorithm with redundant reader elimination," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 900–912, 2012.

[15] Y. Wu, V. K. Lau, D. H. Tsang, and L. Qian, "Energy-efficient transmission strategy for cognitive radio systems," in *Proceedings of IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2012, pp. 41–46.

[16] S. Huang, H. Chen, Y. Zhang, and H.-H. Chen, "Sensing-energy tradeoff in cognitive radio networks with relays," *IEEE Systems Journal*, vol. 7, no. 1, pp. 68–76, 2013.

[17] R. C. Luo and O. Chen, "Wireless and pyroelectric sensory fusion system for indoor human/robot localization and monitoring," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 3, pp. 845–853, 2013.

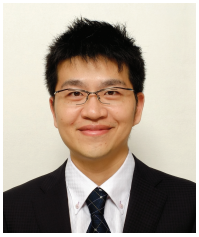
[18] S. Kyoungwon, G. Yang, J. Kurose, and D. Towsley, "Locating network monitors: complexity, heuristics, and coverage," *Computer Communications*, vol. 29, no. 10, pp. 1564–1577, 2006.

Ming Tao received B.S. degree in computer science and technology from Anhui University, Hefei, China, in 2007 and M.S. and Ph.D. degrees in computer application technology from South China University of Technology, Guangzhou, China, in 2009 and 2012, respectively.

His primary research interests include protocol design and performance analysis in next-generation wireless/mobile networks, high performance computing and grid technology.

Kaoru Ota received M.S. degree in Computer Science from Oklahoma State University, USA in 2008, B.S. and Ph.D. degrees in Computer Science and Engineering from The University of Aizu, Japan in 2006, 2012, respectively.

Currently, she is an Assistant Professor with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. Her research interests include Wireless Networks, Cloud Computing, and Cyber-physical Systems.



Mianxiong Dong received B.S., M.S. and Ph.D. in Computer Science and Engineering from The University of Aizu, Japan.

Currently, he is an Associate Professor with the Department of Information and Electronic Engineering at the Muroran Institute of Technology, Japan. His research interests include Wireless Networks, Cloud Computing, and Cyber-physical Systems.