On the Vulnerabilities of the Virtual Force Approach to Mobile Sensor Deployment

Novella Bartolini, *Member, IEEE*, Giancarlo Bongiovanni, Thomas F. La Porta, *Fellow, IEEE*, and Simone Silvestri, *Member, IEEE*

Abstract—The virtual force approach is at the basis of many solutions proposed for deploying mobile sensors. In this paper we study the vulnerabilities of this approach. We show that by compromising a few mobile sensors, an attacker can influence the movement of other sensors and prevent the achievement of the network coverage goals. We introduce an attack, called opportunistic movement, and give an analytical study of its efficacy. We show that in a typical scenario this attack can reduce coverage by more than 50 percent, by only compromising a 7 percent of the nodes. We propose two algorithms to counteract the above mentioned attack, DRM and SecureVF. DRM is a light-weight algorithm which randomly repositions sensors from overcrowded areas. SecureVF requires a more complex coordination among sensors but, unlike DRM, it enables detection and identification of malicious sensors. We investigate the performance of DRM and SecureVF through simulations. We show that DRM can significantly reduce the effects of the attack, at the expense of an increase in the energy consumption due to additional movements. By contrast, SecureVF completely neutralizes the attack and allows the achievement of the coverage goals of the network even in the presence of localization inaccuracies.

Index Terms—Mobile sensors, self-deployment, virtual force approach

1 Introduction

Mobile sensors are used for environmental monitoring in critical scenarios to track the dispersion of pollutants, gas plumes or fires. Several solutions have been proposed to solve the problem of deploying mobile sensors to cover an Area of Interest (AoI). Many of them are based on the Virtual Force Approach (VFA) [1], [2], [3], [4], [5], [6], [7], [8], which models the interactions among sensors as a combination of attractive and repulsive forces. As a result of these antagonist forces, sensors spread throughout the environment.

Mobile sensor networks are prone to several types of security issues. The lack of tamper-proof hardware allows an attacker to capture several nodes, extract their cryptographic material and reprogram them so as to make them behave according to its malicious goal. We refer to such compromised nodes as *malicious nodes*, whereas we refer to the rest of the sensors as *legitimate sensors*. Malicious nodes may perform several types of attacks, influencing the behavior of legitimate nodes. Prior work considers the problem of confidentiality and integrity of communications [9], [10], [11], the sybil attack [12], the problem of false position claims [13], [14] and of sensor clones [15]. Other works consider security issues of routing protocols [16], [17] and neighbor discovery [18], [19].

Despite the abundance of research work on the above mentioned problems, the literature proposed so far does not consider the security vulnerabilities that are specific to deployment and relocation algorithms in mobile sensor networks. In this paper, for the first time in the literature, we show that even if the above mentioned security issues were perfectly addressed, it would still be possible to severely compromise the functionality of a mobile sensor networks based on VFA, by adopting attacks which are specifically designed to compromise movement assisted deployment. As an example, an attack can alter the sensor deployment to prevent the network to achieve its monitoring goals or it can cause useless movements that would deplete the sensor batteries. Note that, although in this paper we focus on VFA, most of the studied vulnerabilities are common to other deployment schemes such as Voronoi based [20] and pattern based [21] approaches.

We introduce the *opportunistic movement* (OM) attack, a new attack specifically targeting mobile sensor deployment algorithms based on virtual forces. The OM attack is not based on any of the security vulnerabilities previously described and works even if the network is endowed with state-of-the-art security mechanisms. It only exploits vulnerabilities inherent to the deployment algorithm.

According to the OM attack, malicious nodes honor the communication protocol but move to positions in which they can exert virtual forces that impede the desired positioning of legitimate sensors. As a clarifying example, we show the effect of an OM attack where few malicious sensors form a barrier which impedes the spreading of legitimate sensors over the AoI, thus creating uncovered areas or corridors, precluding the network from fulfilling its coverage requirements. Our analysis shows that in a typical scenario, by compromising a small fraction as low as 7 percent of legitimate nodes, the attacker is able to reduce the portion of the AoI covered by legitimate sensors by more than

- *N. Bartolini and G. Bongiovanni are with the Department of Computer Science, Sapienza University of Rome, Roma, Italy. E-mail: {bartolini, bongio}@di.uniroma1.it.*
- *T. F. La Porta and S. Silvestri are with the Department of Computer Science and Engineering, Pennsylvania State University, State College, Pennsylvania. E-mail: {tlp, simone}@cse.psu.edu.*

Manuscript received 27 Sept. 2013; revised 17 Feb. 2014; accepted 18 Feb. 2014. Date of publication 24 Feb. 2014; date of current version 26 Sept. 2014. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2014.2308209

50 percent. We also show that, in order to guarantee that the covered area is at least 80 percent of the AoI, the number of legitimate sensors required is more than 23 times higher than the number of malicious sensors.

We propose two algorithms, called Density based Random Movement (DRM) and SecureVF, which are based on a general formulation of the virtual forces provided by the VFA. DRM is a simple and light-weight algorithm which exploits an evaluation of the local density to verify the presence of unwanted concentration of devices possibly due to an attack. SecureVF, on the contrary, provides a mechanism to detect and ignore malicious sensors, but it requires a more complex coordination among sensors.

We formally prove that under SecureVF malicious sensors are detected as soon as their movements violate the rules of the deployment algorithm. We perform extensive simulations which validate the analytical model. Furthermore, we experimentally investigate the ability of DRM and SecureVF to counteract the OM attack. We compare them to the Parallel and Distributed Network Dynamics (PDND) algorithm [3], one of the best existing solutions based on VFA. The experiments show that the OM attack severely reduces the coverage provided by PDND. DRM, despite its simplicity, is able to significantly reduce the impact of the OM attack on the network, but incurs a high expense in terms of energy consumption. SecureVF is able to neutralize the attack and maximize the coverage, and it only introduces a small overhead in terms of communication.

The original contributions of this paper are:

- We investigate the vulnerabilities of mobile sensor deployment algorithms based on VFA and propose a very simple and effective attack, called opportunistic movement, to this approach.
- We provide an analytical model to estimate the effects of the OM attack on VFA solutions.
- We propose two algorithms based on VFA, called DRM and SecureVF to counteract the OM attack.
- Through simulations, we confirm the results provided by our analytical model and we study the efficacy of DRM and SecureVF against the OM attack.

2 Adversary Model and Goals

We consider an adversary which compromises some nodes in the network. This is possible by capturing some legitimate nodes and extracting their cryptographic material, reprogramming and taking full control of them. These corrupted nodes cannot be easily recognized by legitimate nodes, as they are able to send valid messages containing a valid ID, and make use of legitimate cryptographic information. The attacker can thus exploit these corrupted nodes to perform malicious attacks to prevent a successful network deployment.

We assume that network security mechanisms are in place to let each node detect sybil attacks, perform location verification and exchange messages in a secure manner. Furthermore, we assume that the attacker cannot create clones of the compromised nodes. Note that, we do not consider other well-known attacks such as wormhole, grayhole and sinkhole in our adversary model. These attacks would not affect VFA based algorithms because of the locality of communications.

We consider malicious nodes which are able to collude with each other by performing coordinated movements and communications in order to influence the movements of legitimate sensors.

We consider an attacker which aims at impeding the fulfillment of the network coverage requirements. As an example, the attacker can be interested in creating an unmonitored area around a zone of interest, or isolating a part of the network.

3 A General Virtual Force Algorithm

In order to estimate the effects of possible attacks on the performance of deployment algorithms based on virtual forces, we consider a Generalized Virtual Force algorithm (GVF) which extends many previous solutions. We make the typical assumptions found in the works proposing VFA based algorithms: a sensor communicates within a distance R_{tx} (*communication radius*), it covers a circular area of radius R_s (*sensing radius*) and it can move in any direction inside the AoI. Likewise other VFA solutions, the GVF algorithm is round based and sensors are loosely synchronized. Each round is composed by two phases. During the first phase sensors exchange information including their position and ID. In the second phase, each sensor computes the virtual force acting on itself on the basis of the gathered information and moves towards the corresponding destination.

According to GVF, the force acting on a sensor is calculated as follows. Given two sensors s and p located at a distance d from each other, p exerts a force $F(d)$ on s. $F(d)$ models both attractive and repulsive forces and depends on the setting of two parameters: r^* and r_f. The force is null at a distance $d = r^*$, it is repulsive if $d < r^*$ and it is attractive if $d > r^*$. The force also vanishes when the distance d exceeds r_f, where $r_f \leq R_{tx}$.

We hereby define the *area of influence* of a sensor s as the area in which s exerts its virtual force on other sensors. Due to homogeneity, the area of influence of a sensor s is also the area from which other sensors exert a force on s. This area includes all the points at a distance lower than r_f from s. Finally, the force acting on s is therefore the vectorial sum of the forces exerted by all the nodes located in its area of influence.

The GVF algorithm captures the models adopted in most of the previous works based on VFA, such as those presented in [1], [2], [3], [4].

4 The OM Attack

The OM attack is defined on the basis of the adversary model described in Section 2. Malicious nodes can be deployed by the attacker, for example they can be sent from a location which is outside the AoI, or they can be dropped randomly. According to the OM attack, from their initial positions these malicious nodes silently move, that is with no message exchanges, to form an *attack configuration*. Since malicious nodes move silently to their position in the attack configuration, they are not detected by legitimate sensors in this initial phase of the attack.

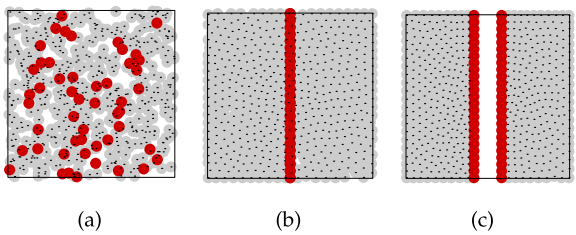

Fig. 1. Initial deployment (a), attack configuration (b) successful creation of an unmonitored corridor (c).

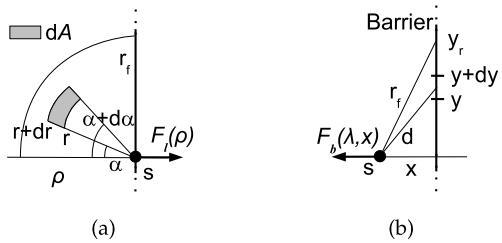

Fig. 2. Force exerted by the uniform distribution (a) and by the barrier (b).

From the attack configuration, malicious nodes give start to the attack by communicating with legitimate sensors according to the communication protocol provided by the deployment algorithm, that is by sending the required information at each round. Nevertheless, the movement of malicious nodes does not comply with the rules provided by the virtual force algorithm. By contrast, malicious nodes move according to the attacker strategy. By gradually adjusting their position, malicious sensors exert forces on legitimate sensors that make them move away from a specific area of interest to the attacker. Since malicious sensors communicate in compliance with the communication protocol, they can influence the movement of legitimate nodes without being recognized as malicious.

Let us consider the following example in which the adversary creates an uncovered corridor over the AoI. Malicious sensors are initially randomly deployed, as depicted in Fig. 1a.[1] Black dots are legitimate sensors and grey circles their sensing ranges, red dots are malicious sensors and red circles their sensing ranges. Malicious sensors perform an initial silent movement so as to form two superimposed barriers, as shown in Fig. 1b. Then they start communicating with legitimate sensors according to the rules of the communication protocol, but move so as to shift the barriers in opposite directions. Legitimate sensors are thus repelled and the attacker successfully creates the unmonitored corridor of Fig. 1c.

The opportunistic movement attack is a general attack which can be performed in many ways, by realizing different attacking configuration and adopting different moving strategy of malicious nodes. In the following we introduce the barrier opportunistic movement (BOM) attack, a specific type of OM attack, which is able to severely reduce the coverage provided by the network while requiring only few sensors to be compromised.

According to the BOM attack, malicious nodes form a linear barrier which touches two sides of the AoI. As provided by the OM attack, malicious sensors periodically communicate their positions in the first phase of each round, while in the second phase they move according to the attacker strategy. In particular, the malicious sensors forming the barrier may move towards legitimate sensors in order to reduce the monitored portion of the AoI, as shown in Figs. 5a, 5b, and 5c. The barrier of malicious sensors may also remain still, in order to prevent legitimate sensors from moving over the uncovered zone isolated by the barrier.

1. These figures are generated by simulating the PDND algorithm under the OM attack. The simulator is described in details in Section 10.

The size of the area in which legitimate sensors can be confined depends on the density of legitimate and malicious sensors. When the density of legitimate sensors becomes too high, they create a pressure on the sensors located in the barrier proximity. As a consequence some of these sensors could cross the barrier and position themselves on the other side.

In the following section we provide an analytical model to estimate the impact of the BOM attack on network coverage.

5 An Analytical Model for the Effect of the BOM Attack

The malicious sensors performing the BOM attack exert a force on the legitimate sensors located in their area of influence. We hereby refer to such legitimate sensors as *frontline sensors*. A frontline sensor is pushed towards the barrier by the other legitimate sensors located in its area of influence. By contrast, it is also pushed in the opposite direction by the malicious sensors of the barrier which reside in the same area. Therefore, a frontline sensor traverses the barrier only if the force exerted by the barrier is lower than the one exerted by legitimate sensors.

The magnitude of such forces depends on the densities of legitimate and malicious nodes. In the following we analytically model this scenario when sensor movements are regulated by the GVF approach.

5.1 Force Exerted by Legitimate Sensors

In the analysis we consider the case in which the legitimate sensors are uniformly deployed with density ρ, on one side of the barrier. Fig. 2a shows the considered scenario. Let s be a frontline sensor and $F_l(\rho)$ be the force acting on s exerted by legitimate sensors. Given the assumption of uniform distribution of legitimate sensors we can assume that the direction of $F_l(\rho)$ is approximately orthogonal to the barrier. We want to calculate the magnitude of the force $F_l(\rho)$.

Let us consider an infinitesimal section of a circular corona $\mathrm{d}A$ with inner radius r and outer radius $r + \mathrm{d}r$, forming an angle α with the horizontal axis centered in s and spanning over an angle $\mathrm{d}\alpha$.

The area $\mathrm{d}A$ can be approximated to $r\mathrm{d}r\mathrm{d}\alpha$, while the number of sensors in $\mathrm{d}A$ are $\rho\mathrm{d}A$. The contribution to $F_l(\rho)$ of the sensors in $\mathrm{d}A$ is $F(r)\cos(\alpha)\rho\mathrm{d}A$. Hence, we can obtain the force acting on s by integrating on α and r:

$$F_l(\rho) = 2\rho \int_0^{\frac{\pi}{2}} \int_0^{r_f} F(r)\cos(\alpha) r \mathrm{d}r\mathrm{d}\alpha. \quad (1)$$

5.2 Force Exerted by the Barrier

Let us consider a barrier of equally spaced malicious sensors with linear density λ. Let us also consider a frontline sensor s located at a distance $x < r_f$ from the barrier. We aim at calculating the force $F_b(\lambda, x)$, orthogonal to the barrier, exerted by the malicious nodes on s.

We refer to Fig. 2b. The infinitesimal segment of the barrier of length $\mathrm{d}y$ placed at distance y from the origin, is at distance $d = \sqrt{x^2 + y^2}$ from s and contains $\lambda \mathrm{d}y$ sensors. The force orthogonal to the barrier, due to malicious sensors, in the above mentioned segment, is therefore $\lambda \mathrm{d}y F(d) \frac{x}{d}$.

The only malicious sensors of the barrier that exert a force on s are the ones located at a distance lower than r_f from s. For this reason we consider the only sensors located at a distance less then $y_r = \sqrt{r_f^2 - x^2}$ from the origin of the considered reference system. By integrating on y we obtain the force exerted on s by the malicious sensors forming the barrier:

$$F_b(\lambda, x) = 2\lambda x \int_0^{y_r} \frac{F(d)}{d} \mathrm{d}y. \tag{2}$$

5.3 Estimate of the Effect of the BOM Attack

The analytical model provided in the previous Sections 5.1 and 5.2 allows us to estimate the forces exerted on frontline sensors by the other legitimate sensors and by the barrier itself. When the former is greater than the latter, some legitimate sensors will eventually cross the barrier; by contrast, none of them will be able to pass through if the force exerted by the barrier is stronger than the one provided by legitimate sensors. The force exerted by the barrier is proportional to its linear density. The case in which the two forces are balanced corresponds to the minimum barrier density value that precludes the flow of legitimate sensors through the barrier.

In order to estimate the effect of the BOM attack on the network we consider the following scenario. N sensors are initially uniformly deployed over a squared AoI. The attack is performed by a barrier of equally spaced malicious sensors, deployed along one side of the AoI. Such a barrier starts moving from outside throughout the AoI, pushing legitimate sensors away. We refer to the area in which legitimate sensors are confined without crossing the barrier as *monitored area* (MA). The MA is gradually reduced by the moving barrier. Thus, as long as no legitimate sensor crosses the barrier, the density ρ of legitimate sensors over the MA increases, as these sensors gradually adjust their positions so as to reach a uniform distribution.

We consider two applications of the analytical model. In the following we denote with x_{max} the minimum distance from the barrier at which the force exerted by the barrier is maximized.[2] We assume that the distance of the barrier from frontline sensors is larger than x_{max}.

2. The existence of x_{max} follows from the fact that the force vanishes at a distance r_f and it is also null on the barrier itself. The uniqueness of such a maximum value depends on the formulation of the virtual force $F(d)$.

First application. We want to calculate the maximum reduction of the monitored area that can be achieved by a barrier of a given density. The maximum reduction is obtained when the barrier has pushed the legitimate sensors at such a high density that a further movement would make some legitimate sensors break through the barrier. When the barrier exerts the maximum reduction, the monitored area has its minimum value mMA. When this occurs, $F_l(\rho) = F_b(\lambda, x_{max})$.

By replacing ρ with N/mMA, we calculate the mMA follows:

$$\mathrm{mMA} = \frac{2N \int_0^{\frac{\pi}{2}} \int_0^{r_f} F(r)\cos(\alpha) r \mathrm{d}r \mathrm{d}\alpha}{F_b(\lambda, x_{max})}. \tag{3}$$

Second application. Here we estimate the minimum number of legitimate sensors needed to ensure that the mMA is at least equal to a threshold value $T_{\mathtt{mMA}}$. More formally, given $T_{\mathtt{mMA}}$ and the density of malicious sensors λ, we want to calculate the number of legitimate sensors N to be deployed such that $\mathrm{mMA} \geq T_{\mathtt{mMA}}$.

When legitimate sensors are deployed on an area of size $T_{\mathtt{mMA}}$, their density is $\hat{\rho} = N/T_{\mathtt{mMA}}$. In order to guarantee that malicious sensors cannot reduce the MA any further, the forces exerted on frontline sensors by malicious sensors with density λ, at distance x_{max}, has to be less than or equal to the force exerted by legitimate sensors with density $\hat{\rho}$, that is:

$$F_b(\lambda, x_{max}) \leq 2\frac{N}{T_{\mathtt{mMA}}} \int_0^{\frac{\pi}{2}} \int_0^{r_f} F(r)\cos(\alpha) r \mathrm{d}r \mathrm{d}\alpha.$$

Therefore, we can calculate N as follows:

$$N \geq \frac{T_{\mathtt{mMA}} F_b(\lambda, x_{max})}{2 \int_0^{\frac{\pi}{2}} \int_0^{r_f} F(r)\cos(\alpha) r \mathrm{d}r \mathrm{d}\alpha}. \tag{4}$$

In Section 10.1 we validate the analytical model by showing that it closely fits the experimental data in the two applications described above.

The proposed analytical model is general and can be applied to several approaches based on virtual forces. In the following section we apply it to the PDND algorithm.

6 Analytical Model of PDND

We now apply the model described in Section 5 to the PDND algorithm [3]. PDND is one of the most complete solutions based on VFA currently available. In particular, unlike several previous proposals, it is formally proved that, under PDND, the sensors stop moving in a finite time without position oscillations which are typical of many VFA based solutions. Furthermore, the algorithm shows very good performance in terms of coverage and uniformity of the final sensor distribution.

PDND is an instance of the GVF model introduced in Section 3, in which the force $F(d)$ is piecewise linear, being composed of two linear pieces joining at $d = r_t$, with $r^* < r_t < r_f$. A detailed definition of the force under PDND is the following:

$$F(d) = \begin{cases} r^* - d, & \text{if } d \leq r_t, \\ (r^* - r_t)(r_f - d)/(r_f - r_t), & \text{if } r_t < d < r_f, \\ 0, & \text{if } d \geq r_f. \end{cases}$$

We now calculate the force exerted on frontline sensors under PDND. As in the previous section we consider a uniform distribution of legitimate sensors with density ρ. By substituting the above formulation of $F(d)$ in Equation (1) we obtain:

$$F_l(\rho) = 2\rho\left\{\frac{r^* r_t^2}{2} - \frac{r_t^3}{3} + \frac{(r^* - r_t)}{r_f - r_t}\frac{r_f^3 - 3r_f r_t^2 + 2r_t^3}{6}\right\}.$$

Similarly, we consider a barrier of malicious sensors with density λ. The force exerted on frontline sensors at a distance x from the barrier can be obtained by substituting the expression of $F(d)$ in Equation (2):

$$F_b(\lambda, x) = 2\lambda x\left\{r^* \ln\left(\frac{r_t + y_t}{x}\right) - y_t. \right. \\ \left. + a\left[r_f \ln\left(\frac{r_f + y_f}{r_t + y_t}\right) + y_t - y_f\right]\right\},$$

where $a = \frac{(r^* - r_t)}{r_f - r_t}$, $y_t = \sqrt{r_t^2 - x^2}$ and $y_f = \sqrt{r_f^2 - x^2}$.

In Section 10 we validate the model through simulations showing that it correctly estimates the impact of the BOM attack on a network running PDND.

7 The DRM Algorithm

In this section we introduce a light-weight algorithm, called Density based Random Movement, designed to counteract the OM attack. DRM extends the GVF algorithm introduced in Section 3 and thus it can be applied to several specific instances of GVF such as PDND.

DRM is based on the observation that the OM attack reduces the area in which legitimate sensors are deployed. A legitimate sensor may perceive such a reduction as an increase in the local density, i.e., the density of sensors in its communication range. If the local density is too high, the sensor acts as if an OM attack is being performed and moves to a random point in the AoI. By means of random movements, some legitimate sensors may move in the area that the attacker wants to keep uncovered.

DRM requires the knowledge of the expected density ρ^*, at which legitimate sensors would be distributed when evenly deployed over the AoI. Similar to GVF, DRM is round based and each round has two phases. During the first phase, a sensor s receives the position information from its neighbors and calculates the local density $\rho(s)$. If $\rho(s)$ is greater than $\rho^* \cdot k_{\texttt{DRM}}$, that is the local density is more than $k_{\texttt{DRM}}$ times higher than the expected density, then s moves to a random point in the AoI. Otherwise, as in GVF, s calculates the virtual forces exerted by its neighbors and moves accordingly.

DRM does not incur in any additional communication overhead with respect to the underlying GVF algorithm, as no additional coordination among sensors is required.

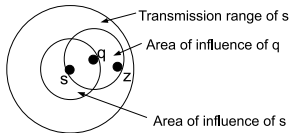

Fig. 3. Communication range and area of influence of a sensor s.

We decided to keep DRM as simple as possible. Nevertheless, several optimizations can be introduced to DRM at the expense of an increased complexity. For example, a major modification requiring additional coordination could provide more informed decisions to allow legitimate sensors move towards the supposed position of the attacker.

8 The SecureVF Algorithm

SecureVF provides a method to enable the detection of malicious sensors performing the OM attack. To this aim, each sensor verifies the correctness of the movements of its neighbors at each round. Sensors deviating from the correct movement are marked as *untrusted* and ignored from the current round on. The virtual force is calculated only on the basis of *trusted* sensors.

SecureVF extends GVF by providing additional phases, namely *movement verification phase* and *trusted neighbors communication phase*.

8.1 Assumptions

SecureVF is designed on the basis of the adversary model introduced in Section 2. We assume the presence of a signature protocol to guarantee authentication of the exchanged messages. We also assume that a node is able to verify the position claimed by other nodes within a range of $2r_f$.[3] If a false position claim is detected, a node is immediately marked as untrusted and ignored by the legitimate sensors located nearby. We assume that $R_{tx} \geq 2r_f$, thus a node is in communication with all the nodes in the area of influence of the neighbors that affect its movement and can verify their positions. To clarify this assumption, let us consider Fig. 3. The sensor q is in the area of influence of s, while z is in the area of influence of q. The above assumption implies that s and z are in radio proximity of each other and can verify their positions. Such an assumption is generally valid: the communication range of sensors is typically 75-100 m in outdoor environments [26], while it is generally assumed that $r_f < 3R_s$ [3], and R_s seldom exceeds a few meters [27]. Notice that, we do not require the communication range of a sensor to be a perfect disk. Indeed, there can be anisotropies provided that a sensor is able to communicate with all sensors located at a distance $2r_f$ from itself. In environments

3. Location verification can be achieved by using dedicated hardware and/or previously deployed anchor nodes. Sensors can autonomously verify position claims if they are equipped with a radar system [22], [23]. These radars conform to our requirements as they are inexpensive, low power and provide object detection up to 20 m distance. Alternatively, Ultra Wide Band systems [24] and anchor nodes can be used for location verification through Verifiable Multilateration (VM) [25]. In this case, anchor nodes are responsible for the location verification and advertise false location claims when detected. Using VM, a sensor incurs in a constant communication overhead for each anchor it communicates with.

with a high level of noise, the distance r_f can be reduced accordingly.

We also assume a maximum moving distance per round of $r_f/2$. This ensures that any two sensors which at a given round are in the area of influence of each other can verify their position at the successive round.[4]

Finally, similar to previous works on mobile sensor deployment [28], [29], we assume that nodes are endowed with low cost GPS[5] and that they are loosely synchronized.

8.2 Nomenclature

We denote by $C^t(s)$ and $N^t(s)$ the set of sensors in the area of influence and in the communication range of the sensor s at round t, respectively. Since we assume $r_f < R_{tx}$, $C^t(s) \subseteq N^t(s)$.

In order to calculate the force acting on itself at round t, a node s takes account of the only sensors in $C^t(s)$ that it considers as trusted. We refer to the set of such trusted sensors with $N^t_{trusted}(s)$ while the set of untrusted nodes discovered until round t is referred to as $N^t_{untrusted}(s)$. Finally, the position of the sensor s at the current round is denoted by $pos^t(s)$.

8.3 The Algorithm

SecureVF extends GVF based solutions with mechanisms for malicious node discovery and isolation. As with DRM, it can be applied to several specific instances of GVF. SecureVF is round based, but each round comprises four phases, namely: position communication, movement verification, trusted neighbors communication and movement. In the following we present such phases in detail. The pseudo-code of the algorithm is shown as Algorithm SecureVF. For the sake of clarity, in the pseudo-code we omit the cryptographic operations that must be performed on the exchanged messages. In the following description, we do not consider localization inaccuracies. We take into account these aspects in Section 8.4.

Position communication (lines 1-3). At the beginning of each round each sensor communicates its position to the neighbors in a secure way. In particular, a sensor s at round t broadcasts the following message: $(s, pos^t(s), t, Sig_s)$ where Sig_s is the signature of the same message signed by s. By receiving the information sent by its neighbors, the sensor s determines the sets $N^t(s)$ and $C^t(s)$. Notice that, if s discovers that a sensor advertises a fake position, s immediately marks it as untrusted.

Movement verification (lines 4-17). In this phase, a sensor s verifies the movements of its neighbor sensors to determine the set of trusted $N^t_{trusted}(s)$ and untrusted $N^t_{untrusted}(s)$ neighbors. At the beginning of the algorithm execution, these sets are initialized so that $N^t_{trusted}(s) = C^t(s)$ and $N^t_{untrusted}(s) = \emptyset$ (lines 4-6).

The set of untrusted neighbors at the current round, $N^t_{untrusted}(s)$, contains all the sensors of $N^{t-1}_{untrusted}(s)$ (line 8) plus possibly other sensors that are detected as malicious in the current round (lines 9-17). The set $N^t_{trusted}(s)$ is used in the successive phase for the calculation of the virtual force acting on s.

A sensor s, in order to verify the trustworthiness of a sensor q, needs to know the position of all the sensors in the area of influence of q. This is possible since $R_{tx} > 2r_f$ by assumption. As a result, a sensor s verifies, for each sensor q in $C^{t-1}(s)$ and not yet in $N^t_{untrusted}(s)$, the correctness of the movement of q at the previous round.[6]

The first check that s performs for a sensor q, in order to verify the correctness of its movement, is on the truthfulness of the set $N^{t-1}_{trusted}(q)$ (lines 12-13). s may determine that q is untrusted if its set $N^{t-1}_{trusted}(q)$ is not consistent.

Algorithm SecureVF: node s at round t.

```
   // Position communication:
1  Broadcast pos^t(s);
2  Receive and verify neighbor positions;
3  Determine the sets N^t(s) and C^t(s);
   // Movement verification:
4  if t = 0 then
5      N^t_untrusted(s) = ∅;
6      N^t_trusted(s) = C^t(s);
7  else
8      N^t_untrusted(s) = N^{t-1}_untrusted(s);
9      for q ∈ C^t(s) s.t. q ∉ N^t_untrusted(s) do
10         if q ∉ C^{t-1}(s) then N^t_trusted(s) ← q;
11         else
12             if (s ∉ N^{t-1}_trusted(q) ∨ N^{t-1}_trusted(q) ⊈ N^{t-1}(s))
               then
13                 N^t_untrusted(s) ← q;
14             else
15                 Calculate pos^t(q) on the basis of
                   N^{t-1}_trusted(q) and pos^{t-1}(q);
16                 if pos^t(q) ≠ pos^t(q) then
                   N^t_untrusted(s) ← q;
17                 else N^t_trusted(s) ← q;
   // Trusted neighbors communication:
18 Broadcast the list of nodes in N^t_trusted(s);
19 Receive N^t_trusted(z) from any z ∈ C^t(s);
   // Moving:
20 Calculate the virtual force on the basis of N^t_trusted(s)
21 and move accordingly;
```

First inconsistency: the sensor q may have maliciously omitted s in the set of its trusted neighbors. Since s knows that it has behaved according to the moving strategy, q must include s in its trusted set. Second inconsistency: the sensor q may have pretended the presence of some trusted sensors in $N^{t-1}_{trusted}(q)$ which are not physically in its area of influence, to try to justify its movement. The sensor s can detect such malicious behaviour because $N^{t-1}(s)$ must include the sensors belonging to $C^{t-1}(q)$ (sensors in the area of influence of q) because we assumed that $R_{tx} \geq 2r_f$.

If an inconsistency is detected, q is marked as untrusted and will be hereafter ignored by s when s calculates the virtual force acting on itself.

4. VFA based algorithms generally introduce a maximum moving distance per round to avoid too long movements which may disconnect the network.

5. Low-cost GPS currently available provide accuracy in the orders of few decimeters [30] and have a cost around 200$ per unit [31].

6. Notice that, the trustworthiness of the sensors belonging to $C^t(s) \setminus C^{t-1}(s)$ will be evaluated at the next round.

If no inconsistency is detected, the sensor s verifies whether q has moved according to the nodes belonging to $N^{t-1}_{trusted}(q)$ (lines 15-17). To this aim, s calculates the expected position of q at the current round t, $\widehat{pos}^t(q)$ on the basis of $pos^{t-1}(q)$ and the set $N^{t-1}_{trusted}(q)$ received at the previous round. The sensor s then compares $\widehat{pos}^t(q)$ with $pos^t(q)$ which q has just broadcast in the previous phase. If the two positions are different, s marks q as untrusted and includes it in $N^t_{untrusted}(s)$. Otherwise, s includes q in the set $N^t_{trusted}(s)$ which will be used to determine the virtual force acting on s at the current round t.

Trusted neighbors communication (lines 18-19). In this phase each sensor s broadcasts the IDs of the nodes belonging to the set $N^t_{trusted}(s)$ calculated in the previous phase. This information enables the neighbors of s to verify its movement at the next round. This broadcast message contains the following information: $(s, q_1, q_2, \ldots, q_k, t, Sig_s)$, where $q_i \in N^t_{trusted}(s)$ and $k = |N^t_{trusted}(s)|$.

Moving (lines 20-21). In the moving phase, each sensor s calculates the virtual force acting on itself on the basis of the trusted set $N^t_{trusted}(s)$ and moves accordingly.

8.4 Dealing with Position Errors

Localization inaccuracies may occur as a consequence of imprecision of the GPS system, which may prevent a sensor to correctly estimate its position. Moreover, any sensor could be unable to position itself precisely due to possible ground asperities. As a result, if a sensor s detects that a sensor q has not moved as expected, it cannot conclude that q is malicious, unless its deviation from the correct movement exceeds a given maximum error threshold.

A malicious sensor could exploit knowledge of the allowed positioning and location errors and perform a series of biased movements which are within the allowed error but which sum up to a movement to a final location which is determined by the attacker. We address these aspects by letting SecureVF deal with positioning and localization errors as follows.

We assume that the GPS system can incur a maximum error $\epsilon_{\texttt{GPS}}$ and that the maximum moving error due to AoI irregularities is bounded by $\epsilon_{\texttt{Mov}}$. Furthermore, we assume that localization and positioning errors are random. We define $\epsilon_{\texttt{MAX}} = \epsilon_{\texttt{GPS}} + \epsilon_{\texttt{Mov}}$. As a result, a legitimate sensor q moving at a round t to its expected position $\widehat{pos}^t(q)$ can deploy in a position $pos^t(q)$ which is at most at a distance $\epsilon_{\texttt{MAX}}$ from $\widehat{pos}^t(q)$. We define a *deviation vector* $\overrightarrow{v}^t(q)$ as the vector that goes from the expected position $\widehat{pos}^t(q)$ to the advertised position $pos^t(q)$. The described situation is depicted in Fig. 4.

Let us consider a legitimate sensor s. For each sensor q that enters in the area of influence of s during the unfolding of the algorithm, s stores the *root-mean-square* $RMS_s(q)$ of the deviation vectors of q, and a counter $n_s(q)$. When q enters in the area of influence of s for the first time, these are both initialized to null. If at a round t the sensor s is able to verify the movement of q, i.e., q was in the area of influence of s at the previous round, s calculates the deviation vector $\overrightarrow{v}^t(q)$. If $|\overrightarrow{v}^t(q)| > \epsilon_{\texttt{MAX}}$ then s marks q as untrusted. Otherwise, s updates $RMS_s(q)$ by including $\overrightarrow{v}^t(q)$ and increases the counter $n_s(q)$.

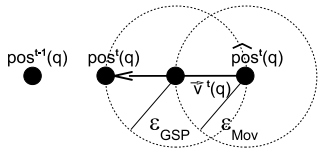

Fig. 4. Effects of localization errors on a sensor q.

For a legitimate sensor q, the sum of deviation vectors can be seen as a random walk starting at the origin where each step has a maximum length $\epsilon_{\texttt{MAX}}$ and is performed in a random direction. The expected value of the root-mean-square distance from the origin after n steps is upperbounded by $\epsilon_{\texttt{MAX}}\sqrt{n}$ [32]. Since, the counter $n_s(q)$ keeps track of the number of vectors that s considers for a sensor q, the expected $RMS_s(q)$ is upperbounded by $\epsilon_{\texttt{MAX}}\sqrt{n_s(q)}$. We define the *normalized RMS* measured by s for a sensor q as $\overline{RMS}_s(q) = \frac{RMS_s(q)}{\sqrt{n_s(q)}}$. If s detects that $\overline{RMS}_s(q) > T_{\texttt{max}}\epsilon_{\texttt{MAX}}$, then it marks q as untrusted. The threshold $T_{\texttt{max}}$, ensures that it is very unlikely that a legitimate sensor is marked as untrusted.

In Section 10 we show that the above described technique allows SecureVF to defeat the OM attack and to achieve the coverage goals of the network even in the presence of localization errors.

9 Algorithm Properties

In this section we study the security properties of SecureVF against the OM attack and we discuss the termination of DRM and SecureVF.

9.1 Security Analysis of SecureVF

In the following we denote by L and M the set of legitimate and malicious sensors, respectively. Given a sensor $s \in L \cup M$, we define the set L^t_s as the set of legitimate sensors whose movement can be influenced by s at round t. In the following analysis we do not consider localization errors.

We first show that under SecureVF legitimate nodes are never marked as untrusted.

Theorem 9.1. *A legitimate sensors never mark other legitimate sensors as untrusted.*

Proof. Let us consider two legitimate sensors q and s at round t. A legitimate sensor q marks s as untrusted only if s advertises an inconsistent trusted set $N^t_{trusted}(s)$ or if it does not move according to the virtual force generated by the nodes in $N^t_{trusted}(s)$. Since s is legitimate, it never includes sensors in $N^t_{trusted}(s)$ which are not physically present in its area of influence, nor does it misrepresents their position. As a result, $N^t_{trusted}(s)$ never contains inconsistencies. Moreover, s always moves in compliance with the virtual force generated by $N^t_{trusted}(s)$, thus its movement is always correct. As a result, q does not mark s as untrusted. □

Next, we consider the detection capability of SecureVF with respect to malicious sensors. Notice that, if a

malicious node m moves in compliance to the rules of SecureVF, that is it advertises a trusted set with no inconsistencies and moves according to the virtual force generated by such a set, it cannot be detected by SecureVF since m is actually behaving as a legitimate sensor. Nevertheless, such movements are unlikely to meet the attacker goals. In the following we define a *malicious movement* as a movement which is not in compliance with the deployment rules.

Theorem 9.2. *Given a malicious sensor $m \in M$ performing a malicious movement at round t, if $L_m^t \neq \emptyset$ then m is marked as untrusted by at least one sensor in L_m^t at round $t+1$.*

Proof. According to the assumptions made in Section 8.1, legitimate sensors are able to detect false location claims, sybil attacks and false identities by using standard techniques. By the assumption that $r_f < R_{tx}/2$, we derive that $d^t(s,m) < R_{tx}/2$, $\forall s \in L_m^t$, where $d^t(\cdot,\cdot)$ is the distance at round t. Thus s is able to verify if $N_{trusted}^t(m)$ is inconsistent. As a result, the only degree of freedom that m has in order to try to justify its malicious movement without being detected is the selection of the nodes to be advertised as trusted.

Notice that, all nodes in L_m^t are legitimate and are at a distance less than r_f from m, thus such sensors should be included in the trusted set of m. If m does not include one or more of them in $N_{trusted}^t(m)$, then such sensors eventually mark m as untrusted at round $t+1$ and then the theorem is proved.

On the contrary, if m includes all sensors in L_m^t in $N_{trusted}^t(m)$, such sensors are at a distance less than or equal to $2r_f$ from m at round $t+1$. Indeed, since we assumed that the maximum moving distance per round is $r_f/2$, we have that $d^{t+1}(s,m) \leq 2r_f$ because:

$$d^{t+1}(s,m) \leq 2(r_f/2) + d^t(s,m) \leq 2r_f.$$

As a result, at the next round all sensors in L_m^t are able to verify the correctness of the current movement of m. Since m is performing a malicious movement, it is not moving in compliance with the virtual force generated by the advertised trusted set, as a consequence it is eventually detected and all sensors in L_m^t mark m as untrusted at round $t+1$. □

Notice that, it is possible to formulate more complex attacks than the OM attack which cannot be detected by SecureVF. As an example, it is possible to formulate attacks in which the set of malicious sensors is split in two teams. The first team performs malicious movements, as in the BOM attack. The second team creates an additional layer of sensors between legitimate sensors and the team performing the BOM, but moves in compliance to the VFA algorithm pretending not to discover the malicious behavior of the first team. If the layer of the second team is sufficiently thick, legitimate sensors do not reach the area of influence of any sensor of the first team, thus they are not able to detect the attack on the basis of a local observation.

Nevertheless, it should be noted that the number of malicious sensors necessary to perform the above described attack is conspicuously larger than in the case of a simple BOM attack. Furthermore, in order to ensure that the second team properly surrounds the first one, malicious sensors must form the attack configuration in the AoI *before* the deployment of the network, which is a not required in the case of the BOM attack. An early deployment is necessary to ensure that no legitimate sensors can reach a sensor of the first team even by chance.

9.2 Termination of DRM and SecureVF

The random movements caused by DRM may not terminate if the setting of the threshold $k_{\texttt{DRM}}$ is too low. Our experiments show that even by setting $k_{\texttt{DRM}}$ as low as 1.1, performance stability is achieved with and without the OM attack. By contrast, the termination of SecureVF is proved by the following Lemma.

Lemma 1. *SecureVF terminates in a finite time, provided that the underlying GVF algorithm has a guaranteed termination.*

Proof. Theorem 9.2 ensures that a malicious sensor performing a malicious movement is either ignored by its neighborhood or detected by at least one legitimate sensor. Hence, the number of malicious movements for each malicious sensors that influence the deployment of legitimate sensors is limited by the number of legitimate sensors. Therefore, let t^* be the time of the last influential malicious movement. After t^*, malicious sensors will no longer influence the movement of legitimate sensors, i.e., $L_m^t = \emptyset$ for each $m \in M$ and $t > t^*$. From round t^* on, the sensor deployment follows the underlying GVF algorithm. If GVF has a guaranteed termination, SecureVF terminates in a finite time. □

In this paper we adopt PDND as an instance of GVF, which has a guaranteed termination [3]. Note that, the termination of PDND is not guaranteed in the presence of localization errors, as the experiments in Section 10.3 show, however, even in this setting, SecureVF is still able to detect malicious sensors and cover the AoI.

10 Experimental Results

In this section we experimentally validate the results provided by the analytical model described in Section 5 and we compare the performance of DRM and SecureVF. Finally, we investigate the robustness of SecureVF to localization and positioning inaccuracies. In order to do so, we developed a simulator on the basis of the Wireless Module of the Opnet simulation environment [33]. We use the following simulation parameters: $R_s = 5\,\text{m}$, $R_{tx} = 25\,\text{m}$, $r^* = \sqrt{3}R_s$, $r_f = 1.2r^*$, moving speed $1\,\text{m/s}$, size of the AoI $150 \times 150\,\text{m}^2$.

10.1 Validation of the Analytical Model

In this section we verify through simulations the capability of the analytical model introduced in Section 5 to predict the effects of the BOM attack on a network running the PDND algorithm [3] without any security mechanism.

The goal of the attacker is to reduce the monitored area, that is the portion of the AoI in which legitimate sensors are confined without crossing the barrier. We use the analytical model to estimate the impact of the BOM attack on

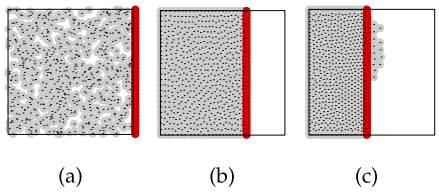

Fig. 5. BOM attack on a network running PDND.

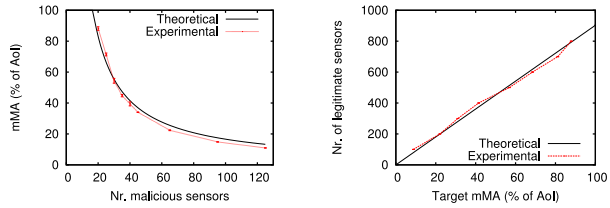

Fig. 6. mMA by increasing the number of malicious sensors (a) and minimum number of legitimate sensors to ensure a target mMA (b).

the network considering the two applications described in Section 5.3. In particular, we aim at estimating: (1) the minimum monitored area (mMA) as a function of the number of malicious sensors deployed, given the number of legitimate sensors, by using Eq. (3); (2) the minimum number of legitimate sensors necessary to ensure that the mMA is equal to a target value, given the number of malicious sensor deployed, by using Eq. (4).

We consider a scenario where legitimate sensors are initially randomly deployed over the AoI while malicious sensors form a barrier parallel to one edge of the area. Fig. 5a shows the considered scenario with 500 legitimate sensors and 35 malicious sensors.

Malicious sensors perform the BOM attack by moving the barrier from right to left. When the barrier starts moving across the AoI legitimate sensors are repelled, resulting in a reduction of the monitored area (Fig. 5b). As the size of the monitored area decreases, the density of legitimate sensors increases, thus the force exerted by the barrier is no longer sufficient to repel legitimate sensors and some break through (Fig. 5c).

We compare the estimates provided by the analytical model for the two applications to the results obtained through simulations. In the experiments the mMA is calculated as the portion of AoI in which legitimate sensors are confined when no more than 3 percent of legitimate sensors cross the barrier.

In the first set of experiments, we deploy 500 legitimate sensors and we increase the number of malicious sensors. Fig. 6a shows the obtained results. The theoretical analysis shows a good fit with the experimental curve. The results highlight that even a small number of malicious nodes can cause serious damage to the network. As a numerical example, the attacker is able to reduce the mMA to less than 50 percent of the AoI by compromising only the 7 percent of legitimate sensors. This shows the detrimental effect of the BOM attack when no security mechanisms are in place.

In the second set of experiments, we deploy 30 malicious sensors and show the minimum number of legitimate sensors necessary to balance the effect of the barrier, as a function of the target mMA. Results are depicted in Fig. 6b. Also in this case the analytical model has a good fit with the simulations in predicting the effect of the BOM attack. In order to achieve an mMA larger than 80 percent of the AoI, the number of legitimate sensors has to be more than 23 times higher than the number of compromised sensors (700 legitimate sensors, against 30 malicious sensors). Similar to the first set of experiments, this set also shows how easy it is for an attacker to compromise the monitoring capability of a VFA based network.

10.2 Performance Comparisons

In this section we experimentally study the efficacy of DRM and SecureVF against the OM attack and we evaluate their performance. We consider two dynamic attack scenarios on an already deployed network. In the first scenario malicious sensors form a barrier parallel to an AoI edge, which gradually shifts towards legitimate sensors, while in the second scenario malicious sensors create a circular coverage hole by radially moving from the center of the area. In order to maximize the effect of the attack, the attacker stops moving malicious sensors as soon as at least 3 percent of legitimate sensors have penetrated the barrier. Under SecureVF, malicious sensors advertise a trusted set that contains every node in their area of influence to avoid easy detection due to a malformed set.

In the experiments, DRM and SecureVF adopt the force formulation of PDND. For this reason, we compare them to the basic version of PDND. In the evaluation we do not consider localization errors, which are studied in Section 10.3.

10.2.1 *Barrier Attack*

In this set of experiments we consider a scenario where legitimate sensors are initially randomly deployed over the AoI while malicious sensors form a barrier parallel to an AoI edge and are initially located outside the area. The barrier gradually moves towards legitimate sensors to reduce the monitored area. An example of the considered scenario with 600 legitimate sensors and 50 malicious sensors is shown in Fig. 7a.

Before showing the experimental results, we give an example of the execution of PDND, DRM and SecureVF in the considered scenario. In this example, we set $k_{\text{DRM}} = 1.5$ for DRM. Figs. 7b, 7c, and 7d show the final deployment achieved under PDND, DRM and SecureVF, respectively. Although the number of legitimate sensor is more than 10 times higher than the number of malicious sensors, the OM attack severely reduces the monitored area of PDND. Under DRM the attack has less impact, nevertheless the attacker is still able to significantly reduce the coverage because it stops the barrier as soon as only a few sensors have crossed it. As a result, the density on the left side of the barrier does not exceed the critical threshold which would ignite more random movements. On the contrary, under SecureVF legitimate sensors detect and ignore malicious sensors. As a result, the barrier is crossed by several legitimate sensors as soon as the it enters the AoI, thus it stops moving and complete coverage is achieved. We also investigated the case in which the attacker pushes the barrier to the left side of the AoI without stopping it. Even in

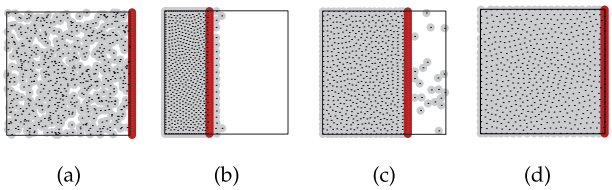

Fig. 7. Barrier attack: Initial deployment (a), final deployment of PDND (b), DRM (c) and SecureVF (d).

such a case malicious sensors are detected and ignored and legitimate sensors achieve full coverage.

We now show the performance comparison between PDND, DRM and SecureVF in the considered scenario. In the experiments we study several performance metrics by increasing the number of legitimate sensors while the number of malicious sensors is 50. We consider two settings for the density threshold of DRM, $k_{\mathrm{DRM}} = 1.2$ and $k_{\mathrm{DRM}} = 1.5$. The performance metrics are related to legitimate sensors only.

Fig. 8a shows the coverage achieved by the considered algorithms. PDND is strongly affected by the OM attack; it achieves only 75 percent of coverage even when legitimate sensors are more than twice the minimum number necessary to achieve full coverage. DRM is able to reduce the impact of the OM attack. A lower setting of the threshold k_{DRM} results in a higher coverage because a lower density is required to give start to the random movements. Nevertheless, DRM is not able to achieve full coverage even when 900 sensors are deployed. SecureVF enables legitimate sensors to detect and ignore malicious sensors, as a result legitimate sensors successfully spread over the AoI, maximizing the coverage.

The average distance traversed by sensors is shown in Fig. 8b. PDND shows a decreasing traversed distance as the number of legitimate sensors increases. This is due to the lower impact of the attack when more legitimate sensors are present. If few legitimate sensors are deployed, the barrier pushes and confines them on a small area, resulting in a long traversed distance. On the contrary, as the number of legitimate sensors increases, the barrier stops earlier, reducing the distance traversed by sensors. Under DRM, the traversed distance depends on the setting of the threshold k_{DRM}. A lower value causes more random movements, thus increasing the overall traversed distance. Under SecureVF the barrier stops moving as soon as it enters the AoI, as a result legitimate sensor quickly reach an equilibrium of the virtual forces and traverse short distances.

Fig. 8c shows the average number of moving actions per sensor. This is an important metric to evaluate mobile sensor deployment algorithms, since a sensor consume a high amount of energy to start and stop a movement [28]. The behavior of the number of movements under PDND is similar to that of the traversed distance: the movements end as soon as the barrier stops. DRM is not particularly affected by the setting of k_{DRM}. The number of movements under DRM depends on the distance traversed by the barrier as well as on the amount of random movements. With a lower setting of k_{DRM} the barrier stops earlier, but the number of random movements is higher. On the contrary, a higher setting of k_{DRM} lets the barrier move further in the AoI, but incurs in a lower number of random movements. These two effects compensate, causing the two settings of k_{DRM} to behave similarly. SecureVF requires a significantly lower number of movements with respect to the other approaches, because the barrier is promptly detected and it does not impact the deployment of legitimate sensors.

In Fig. 8d we show the cumulative energy consumption per sensor. Sensors consume energy for communications (sending and receiving messages), start and stop actions, and movements. We consider the energy cost model expressed in energy units (eu) adopted in [21], [28], [29], [34]: receiving a message costs 1eu, sending a message 1.125eu, 1m movement and starting/stopping a movement cost the same as 300 messages.

The considerations made for the traversed distance and the number of movements apply also to the energy consumption for the considered algorithms. SecureVF shows a local maximum around 400 sensors. This is a common

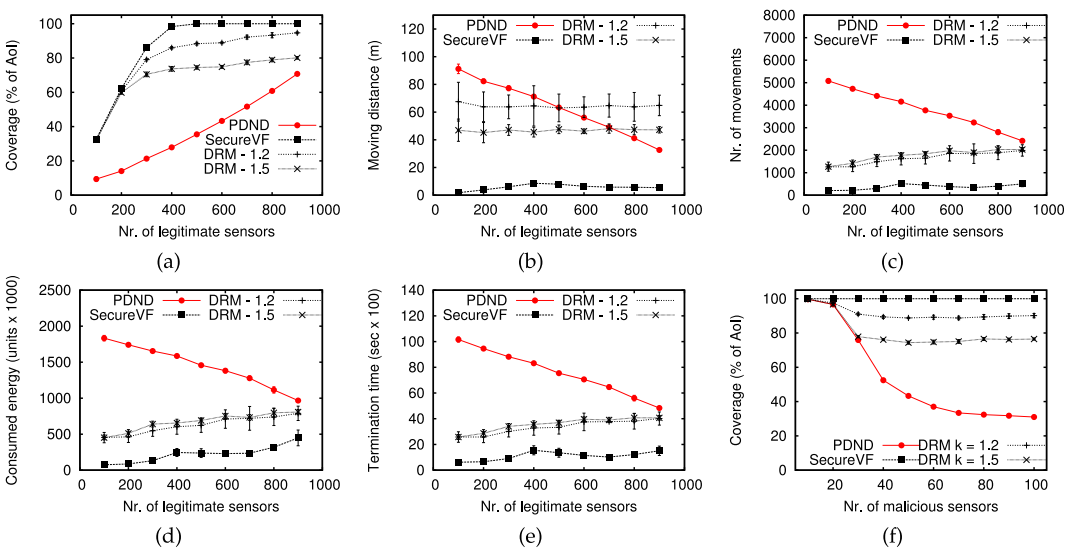

Fig. 8. Barrier attack: coverage (a), traversed distance (b), number of movements (c), consumed energy (d), termination time (e). Coverage achieved with 600 legitimate sensors.

behavior of mobile sensor deployment algorithms when the number of deployed sensors is close to the minimum required for full coverage, because all sensors have to move to contribute to the coverage of the AoI. The energy consumption under SecureVF increases with the number of sensors because of the higher communication overhead required to exchange the trusted set as the sensor density increases.

The termination time of the considered algorithms, i.e., the time at which all sensors stop moving, is depicted in Fig. 8e. PDND, DRM and SecureVF have a similar trend with respect to the above described metrics. SecureVF is able to achieve a higher coverage in a shorter time when compared to DRM and PDND.

In order to further study the performance of the algorithms, we performed a set of experiments by deploying 600 legitimate sensors and increasing the number of malicious sensors. Fig. 8e shows the achieved coverage. The coverage provided by PDND decreases as the number of malicious sensors increases, because the OM has more impact when more malicious sensors are available. DRM behaves similarly to PDND when there are not enough malicious sensors to cause random movements of DRM. On the contrary, when random movements occur, the provided coverage is not affected by the number of malicious sensors deployed and it only depends on the setting of k_{DRM}. Under SecureVF malicious sensors are detected and ignored, independently of their number, as a result SecureVF achieves full coverage in all the considered scenarios.

For space limitations, we do not show the results related to traversed distance, number of movements, energy consumption and termination time obtained by increasing the number of malicious sensors. We summarize these results here. PDND has the worst performance. It shows an increasing trend in all the cited metrics as the number of malicious sensors increases. Since the OM attack has more impact when more malicious sensor are available, legitimate sensors traverse more distance, perform more movements, consume more energy and the algorithm converges later. DRM shows stability problems when malicious sensors are more than 70, for the setting $k_{\mathrm{DRM}} = 1.2$. This is due to the density of malicious sensors on the barrier which is sufficiently high to let a legitimate sensor perform a random movement when it is in the barrier proximity. SecureVF is not affected by the number of malicious sensors deployed, it provides full coverage achieving the best performance in terms of all the considered metrics with respect to PDND and DRM.

10.2.2 Donut Attack

In this set of experiments we consider malicious sensors to be initially deployed at center of the AoI and then perform a radial movement in order to create a circular coverage hole. Also in this scenario, the attacker stops the movement as soon as 3 percent of legitimate sensors have entered the circular zone. Legitimate sensors are randomly deployed over the area. In the experiments we deploy 50 malicious sensors and we increase the number of legitimate sensors.

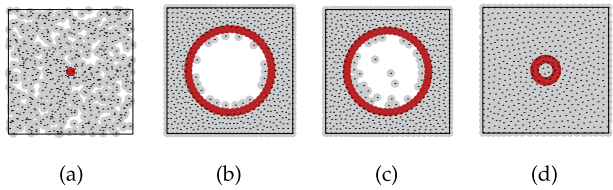

Fig. 9. Donut attack: Initial deployment (a), final deployment under PDND (b), DRM (c) and SecureVF (d).

The considered initial configuration is shown in Fig. 9a where 600 legitimate sensors and 50 malicious sensors are deployed. Figs. 9b, 9c, and 9d show the final deployment achieved under PDND, DRM with $k_{\mathrm{DRM}} = 1.5$, and SecureVF, respectively. Also in this experimental scenario the OM attack is able to significantly reduce the area covered by PDND and DRM.[7] Under SecureVF legitimate sensors detect the malicious movement and then ignore malicious sensors, achieving full coverage.

Fig. 10a shows the coverage achieved by PDND, DRM and SecureVF. Also in this scenario, the coverage provided by the network under PDND is severely reduced. DRM mitigates the effect of the attack, resulting in a higher coverage than PDND. Notice that, when more than 700 legitimate sensors are present, the coverage of DRM with $k_{\mathrm{DRM}} = 1.5$ and PDND are similar. This is due to the setting of the density threshold which is sufficiently high to make random movements unlikely, and thus legitimate sensors penetrate the circular area because of the magnitude of the force exerted on frontline sensors by legitimate sensors. In this scenario, the OM attack has less impact on the network with respect to the barrier attack for a given number of malicious sensors. This is due to the longer perimeter such sensors have to cover while performing the attack. SecureVF detects and ignores malicious sensors also in this case, achieving the maximum coverage in all the considered scenarios.

Figs. 10b, 10c, 10d, and 10e show the traversed distance, the number of movements, the consumed energy and the termination time, respectively. PDND has a decreasing behavior in all the above mentioned metrics because, similar to the barrier case, the impact of the attack decreases as more legitimate sensors are available. DRM shows an increasing trend in the considered metrics because the initial density of malicious sensors is very high and causes all legitimate sensors located in proximity of the donut to move randomly. For the setting $k_{\mathrm{DRM}} = 1.5$ the performance converge to the one of PDND as the number of legitimate sensors increases. SecureVF outperforms PDND and DRM by achieving full coverage while requiring a lower traversed distance, less movements, less energy and a shorter termination time.

Also in this scenario we perform a set of experiments by randomly deploying 600 legitimate sensors and increasing the number of malicious sensors. Fig. 10f shows the coverage achieved by the algorithms. The coverage of PDND

7. Note that, under DRM random movements may continue for a while even after the barrier stops. Therefore, more than 3 percent of legitimate sensors may have crossed the circular barrier when the algorithm terminates, as shown in Fig. 9c.

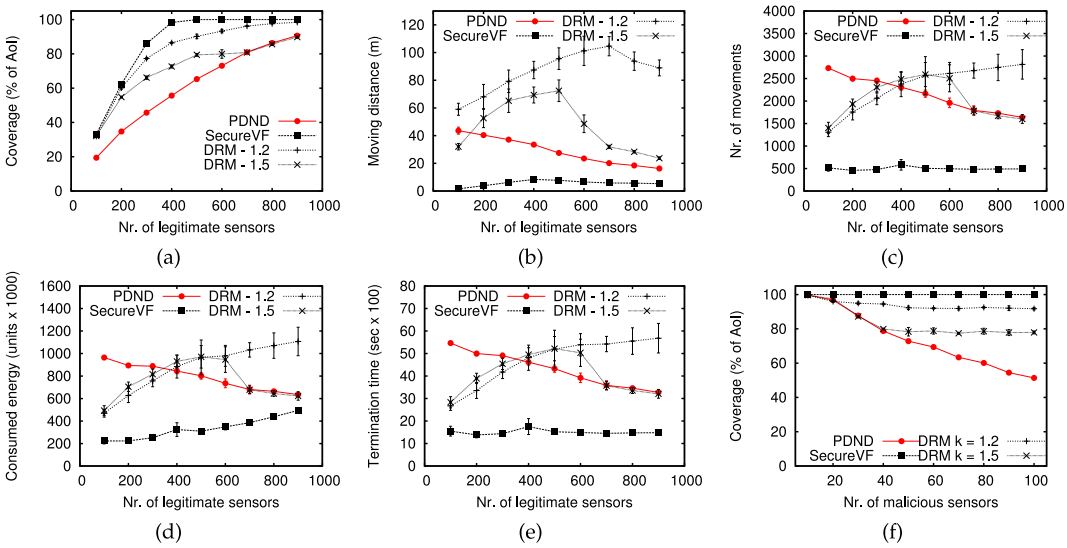

Fig. 10. Donut attack: coverage (a), traversed distance (b), number of movements (c), consumed energy (d), termination time (e). Coverage achieved with 600 legitimate sensors.

rapidly drops as more malicious sensors are present. Similar to the barrier attack case, DRM achieves a coverage which depends on the setting of $k_{\texttt{DRM}}$. SecureVF is not affected by the number of malicious sensors deployed and it achieves full coverage in all the considered cases. We do not show results related to the other performance metrics for space limitations. We observed similar trends with respect to the one discussed for the barrier attack case.

10.3 Robustness to Localization Inaccuracies

In this section we study the robustness of SecureVF with respect to localization inaccuracies, as described in Section 8.4. We consider a scenario where legitimate sensors are initially deployed on a stripe located on the left side of the AoI, while malicious sensors form a static barrier which splits the AoI in two halves. The initial deployment is shown in Fig. 11a.

Fig. 11b shows a zoomed snapshot of the execution of SecureVF with $\epsilon_{\texttt{MAX}} = 1$m. The white bars show the average normalized RMS of the deviation vectors (see Section 8.4) measured by legitimate sensors. Notice that, we show the normalized RMS as a vector to highlight the bias in the deviation vectors of malicious sensors. As the figure points out, the normalized RMS of legitimate sensors remains small and has a random direction. On the contrary, the biased movement of malicious sensors cause their normalized RMS to grow and eventually exceed the threshold $T_{\texttt{max}}\epsilon_{\texttt{Tot}}$, as described in Section 8.4. As a result, malicious sensors are detected and ignored.

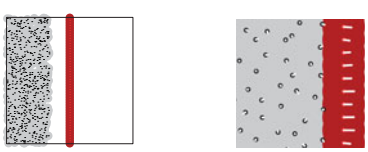

Fig. 11. Localization errors: initial deployment (a), average normalized RMS (b).

Notice that the PDND algorithm, on which the force formulation of SecureVF is based, does not guarantee convergence in the presence of localization inaccuracies, i.e., sensors may be unable to stop their movement. As a result, we studied the performance of SecureVF over time. In the experiments we set $T_{\texttt{max}} = 3$ and we consider different settings of the maximum error $\epsilon_{\texttt{MAX}}$.

Fig. 12a shows the coverage over time under different settings of $\epsilon_{\texttt{MAX}}$. Higher values of $\epsilon_{\texttt{MAX}}$ lengthen the time required by SecureVF to cover the AoI, because more deviation vectors are needed to detect malicious behaviors. Nevertheless, SecureVF is able to achieve complete coverage under all the considered settings. As expected, when $\epsilon_{\texttt{MAX}} > 0$, the algorithm does not converge, as a result the traversed distance increases as shown in Fig. 12b. Oscillation control mechanisms can be introduced to ensure termination as described in [1].

11 Conclusions and Open Problems

In this paper we pointed out, for the first time in the literature, the security vulnerabilities of deployment algorithms based on VFA. We introduce the OM attack, specifically tailored for mobile sensor deployment algorithms. We analytically studied a particular type of OM attack, where malicious sensors form a barrier, showing its detrimental effect on network coverage. We propose two approaches to counteract the OM attack, DRM and SecureVF.

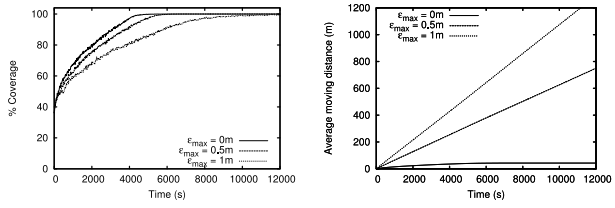

Fig. 12. Localization errors: coverage (a) and traversed distance (b) over time.

This new area of research still remains vastly unexplored. In the following we discuss some of the open research problems.

New attacker strategies can be designed, which may have different goals instead of reducing the network coverage and may exploit attacks such as Sybil attack and node cloning.

Other deployment schemes, besides VFA, have been proposed to deploy mobile sensors, such as Voronoi based [28] and pattern based [21]. These schemes do not take into account potential security vulnerabilities and thus are prone to security attacks which need to be further investigated.

ACKNOWLEDGMENTS

This work was partially supported by the William E. Leonhard Chair at the Pennsylvania State University.

REFERENCES

[1] N. Heo and P. Varshney, "Energy-Efficient Deployment of Intelligent Mobile Sensor Networks," *IEEE Trans. Systems, Man and Cybernetics*, vol. 35, no. 1, pp. 78-92, Jan. 2005.
[2] J. Chen, S. Li, and Y. Sun, "Novel Deployment Schemes for Mobile Sensor Networks," *Sensors*, vol. 7, pp. 2907-2919, 2007.
[3] K. Ma, Y. Zhang, and W. Trappe, "Managing the Mobility of a Mobile Sensor Network Using Network Dynamics," *IEEE Trans. Parallel and Distributed Systems*, vol. 19, no. 1, pp. 106-120, Jan. 2008.
[4] Y. Zou and K. Chakrabarty, "Sensor Deployment and Target Localization in Distributed Sensor Networks," *ACM Trans. Embedded Computing Systems*, vol. 3, no. 1, pp. 61-91, 2003.
[5] M. Garetto, M. Gribaudo, C.-F. Chiasserini, and E. Leonardi, "A Distributed Sensor Relocation Scheme for Environmental Control," *Proc. IEEE Int'l Conf. Mobile Adhoc and Sensor Systems (MASS)*, 2007.
[6] M.R. Pac, A.M. Erkmen, and I. Erkmen, "Scalable Self-Deployment of Mobile Sensor Networks: A Fluid Dynamics Approach," *Proc. IEEE Int'l Conf. Intelligent Robots and Systems (IROS)*, 2006.
[7] M. Lam and Y. Liu, "Two Distributed Algorithms for Heterogeneous Sensor Network Deployment Towards Maximum Coverage," *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, 2008.
[8] S. Poduri and G.S. Sukhatme, "Constrained Coverage for Mobile Sensor Networks," *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, 2004.
[9] H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks," *Proc. IEEE Symp. Security and Privacy*, 2003.
[10] W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney, "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge," *Proc. IEEE INFOCOM*, 2004.
[11] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. Shantz, "Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks," *Proc. Third IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom'05)*, 2005.
[12] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil Attack in Sensor Networks: Analysis & Defenses," *Proc. Third Int'l Symp. Information Processing in Sensor Networks (IPSN'04)*, 2004.
[13] K. Liu, N. Abu-Ghazaleh, and K. Kang, "Location Verification and Trust Management for Resilient Geographic Routing," *J. Parallel and Distributed Computing*, vol. 67, no. 2, pp. 215-228, 2007.
[14] S. Capkun, K. Rasmussen, M. Cagalj, and M. Srivastava, "Secure Location Verification with Hidden and Mobile Base Stations," *IEEE Trans. Mobile Computing*, vol. 7, no. 4, pp. 470-483, Apr. 2008.
[15] Y. Zeng, J. Cao, S. Zhang, S. Guo, and L. Xie, "Random-Walk Based Approach to Detect Clone Attacks in Wireless Sensor Networks," *IEEE J. Selected Areas in Comm.*, vol. 28, no. 5, pp. 677-691, June 2010.
[16] I. Khalil and S. Bagchi, "Stealthy Attacks in Wireless Ad Hoc Networks: Detection and Countermeasure," *IEEE Trans. Mobile Computing*, vol. 10, no. 8, pp. 1096-1112, Aug. 2011.
[17] I.M. Khalil, A. Khreishah, F. Ahmed, and K. Shuaib, "Dependable Wireless Sensor Networks for Reliable and Secure Humanitarian Relief Applications," *Ad Hoc Networks*, vol. 13, part A, pp. 94-106, 2014.
[18] M. Poturalski, P. Papadimitratos, and J.-P. Hubaux, "Towards Provable Secure Neighbor Discovery in Wireless Networks," *Proc. Sixth ACM Workshop Formal Methods in Security Eng. (FMSE'08)*, pp. 31-42, 2008.
[19] R. Shokri, M. Poturalski, G. Ravot, P. Papadimitratos, and J.-P. Hubaux, "A Practical Secure Neighbor Verification Protocol for Wireless Sensor Networks," *Proc. Second ACM Conf. Wireless Network Security (WiSec'09)*, pp. 193-200, 2009.
[20] N. Bartolini, G. Bongiovanni, T.La Porta, S. Silvestri, and F. Vincenti, "Voronoi-Based Deployment of Mobile Sensors in the Face of Adversaries," *Proc. IEEE Int'l Conf. Comm.*, 2014.
[21] N. Bartolini, T. Calamoneri, E. Fusco, A. Massini, and S. Silvestri, "Push & Pull: Autonomous Deployment of Mobile Sensors for a Complete Coverage," *Wireless Networks*, vol. 16, no. 3, pp. 607-625, 2010.
[22] P.K. Dutta, A.K. Arora, and S.B. Bibyk, "Towards Radar-Enabled Sensor Networks," *Proc. Fifth ACM Int'l Conf. Information Processing in Sensor Networks (IPSN'06)*, pp. 467-474, 2006.
[23] G. Yan, S. Olariu, and M.C. Weigle, "Providing Vanet Security through Active Position Detection," *Computer Comm.*, vol. 31, no. 12, pp. 2883-2897, 2008.
[24] R.J. Fontana, E. Richley, and J. Barney, "Commercialization of an Ultra Wideband Precision Asset Location System," *Proc. IEEE Conf. Ultra Wideband Systems and Technologies (UWST)*, pp. 369-373, 2003.
[25] S. Capkun and J.-P. Hubaux, "Secure Positioning of Wireless Devices with Application to Sensor Networks," *Proc. IEEE INFOCOM*, vol. 3, pp. 1917-1928, 2005.
[26] Crossbow, "Telosb Datasheet," www.willow.co.uk/TelosB_Datasheet.pdf, 2013.
[27] MAXBOTIX, "Sonar Datasheets," http://www.maxbotix.com/uploads/LV-MaxSonar-EZ1-Datasheet.pdf, 2013.
[28] G. Wang, G. Cao, and T. La Porta, "Movement-Assisted Sensor Deployment," *IEEE Trans. Mobile Computing*, vol. 5, no. 6, pp. 640-652, June 2006.
[29] N. Bartolini, T. Calamoneri, T.La Porta, and S. Silvestri, "Autonomous Deployment of Heterogeneous Mobile Sensors," *IEEE Trans. Mobile Computing*, vol. 10, no. 6, pp. 753-766, June 2011.
[30] T. Beran, R.B. Langley, S.B. Bisnath, and L. Serrano, "High-Accuracy Point Positioning with Low-Cost GPS Receivers," *Navigation*, vol. 54, no. 1, pp. 53-63, 2007.
[31] Memsic, "Mts420/400 Datasheet," http://pharos.ece.utexas.edu/wiki/images/c/c0/Mts420-datasheep.pdf, 2014.
[32] J.M. Borwein, D. Nuyens, A. Straub, and J. Wan, "Random Walks in the Plane," *Discrete Mathematics and Theoretical Computer Science*, special volume for FPSAC, pp. 191-202, 2010.
[33] "Opnet Technologies Inc." http://www.opnet.com, 2014.
[34] G. Sibley, M. Rahimi, and G. Sukhatme, "Mobile Robot Platform for Large-Scale Sensor Networks," *Proc. IEEE Int'l Conf. Robotics and Automation*, 2002.

Novella Bartolini graduated with honors in 1997 and received the PhD degree in computer engineering in 2001 from the University of Rome, Italy. She is now an assistant professor at the University of Rome. She was researcher at the Fondazione Ugo Bordoni in 1997, visiting scholar at the University of Texas at Dallas in 1999-2000, and research assistant at the University of Rome 'Tor Vergata' in 2000-2002. She was program chair and program committee member of several international conferences. She has served on the editorial boards of *Elsevier Computer Networks and ACM/Springer Wireless Networks*. Her research interests lie in the area of wireless mobile networks and web based systems. She is a member of the IEEE.

Giancarlo Bongiovanni obtained the Laurea degree in Scienze dell'Informazione at the University of Pisa, Italy, in 1973. In 1974 and 1975, he was with Selenia S.p.A., working in the field of parallel computer architectures for signal processing. From 1976 to 1981, he had a research position at the University di Pisa, Italy, working on combinatorial structures and magnetic bubble memories. In 1980, he was a visiting scientist at the IBM T. J. Watson Research Center (Yorktown Heights, New York,USA), working on algorithms for data transmission with SS/TDMA satellite systems. From 1982 to 1987, he had a research position at the University "La Sapienza" of Rome, Italy, working mainly on VLSI architectures. From 1987 to 1991, he was a full professor at the University of l'Aquila, Italy. Since 1991, he has been a full professor at the University "La Sapienza" of Rome, Italy. At "La Sapienza," he has served as chair of the Department of Computer Science, coordinator of the Ph.D. program in computer science, and president of the Council of Computer Science degrees. His current research interests are focused mainly on distributed architectures, computer networks and sensor networks.

Thomas F. La Porta is a Distinguished Professor in the Department of Computer Science and Engineering at Penn State University. He joined Penn State in 2002. Dr. La Porta is the director of the Institute for Networking and Security Research. Prior to joining Penn State, he was with Bell Laboratories starting in 1986. There he was the director of the Mobile Networking Research Department where he led various projects in wireless and mobile networking. He is an IEEE Fellow, Bell Labs Fellow, received the Bell Labs Distinguished Technical Staff Award in 1996, and an Eta Kappa Nu Outstanding Young Electrical Engineer Award in 1996. He also won Thomas Alva Edison Patent Awards in 2005 and 2009. Dr. La Porta received the B.S.E.E. and M.S.E.E. degrees from The Cooper Union, New York, New York, and the Ph.D. degree in electrical engineering from Columbia University, New York, New York. He was the founding editor-in-chief of the *IEEE Transactions on Mobile Computing*, and served as editor-in-chief of *IEEE Personal Communications Magazine* for three years. His research interests include mobility management, signaling and control for wireless networks, mobile data and sensor systems, and network security.

Simone Silvestri is a post-doctoral researcher in the Department of Computer Science and Engineering at Pennsylvania State University. Before joining Penn State, Dr. Silvestri was a post-doctoral researcher in the Department of Computer Science at Sapienza University of Rome, Italy, where he also received the Ph.D. in computer science in 2010. He served as program committee member of several international conferences. His research interests lie in the area of interdependent networks, network recovery and inference, green small cell networks, network tomography, sensor networks and complex web systems. He is a member of the IEEE.