On the security vulnerabilities of the virtual force approach to mobile sensor deployment

N. Bartolini, G. Bongiovanni
Department of Computer Science
Sapienza University of Rome, Italy
{bartolini, bongio}@di.uniroma1.it

T. La Porta
Networking and Security Research Center
Pennsylvania State University, USA
tlp@cse.psu.edu

S. Silvestri
Department of Computer Science
Sapienza University of Rome, Italy
silvestris@di.uniroma1.it

***Abstract*—In this paper we point out the vulnerabilities of the virtual force approach to mobile sensor deployment, which is at the basis of many deployment algorithms. For the first time in the literature, we show that some attacks significantly hinder the capability of these algorithms to guarantee a satisfactory coverage.**

An attacker can compromise a few mobile sensors and force them to pursue a malicious purpose by influencing the movement of other legitimate sensors. We make an example of a simple and effective attack, called Opportunistic Movement, and give an analytical study of its efficacy. We also show through simulations that, in a typical scenario, this attack can reduce coverage by more than 50% by compromising a number of nodes as low as the 7%.

We propose SecureVF, a virtual force deployment algorithm able to neutralize the above mentioned attack. We show that under SecureVF malicious sensors are detected and then ignored whenever their movement is not compliant with the moving strategy provided by SecureVF. We also investigate the performance of SecureVF through simulations, and compare it to one of the most acknowledged algorithms based on virtual forces. We show that SecureVF enables a remarkably improved coverage of the area of interest, at the expense of a low additional energy consumption.

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

I. Introduction

Many solutions have been proposed to solve the problem of deploying mobile sensors to cover over 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.

Not only are mobile networks prone to the same security attacks of static networks, but they can also be attacked by exploiting specific vulnerabilities of mobility controlled devices. In this paper, for the first time in the literature, we investigate the vulnerabilities of the virtual force approach. Under VFA, by compromising a subset of nodes, an attacker can influence the movement of legitimate sensors during the deployment and relocation phases, and impede the fulfillment of coverage goals. We introduce a new kind of attack specifically tailored for mobile sensor deployment algorithms based on virtual forces, the *Opportunistic Movement* (OM) attack. According to such an attack, malicious nodes honour the communication protocol but move to positions in which they can exert virtual forces that impede the correct positioning of legitimate sensors. These malicious nodes follow the purpose of the attacker, for instance by creating an unmonitored corridor or isolating a part of the network, thus forcing legitimate nodes to position themselves over a limited portion of the AoI. As a clarifying example, we show the effect of an OM attack where malicious sensors form a barrier which impedes the spreading of legitimate sensors over the AoI.

This work has been partially supported by the NSF project CNS-0916171.

We analytically characterize the effect of this attack on a network executing a general virtual force based algorithm. The analysis shows that by compromising a small fraction, as low as 7%, of legitimate nodes, the attacker is able to reduce the portion of the AoI covered by legitimate sensors by more than 50%.

We propose an algorithm, called SecureVF, which is based on a general formulation of the virtual forces provided by the VFA and provides a set of rules to determine the presence of malicious sensors and neutralize the attack. This set of rules is independent of the particular force formulation and can be applied to any VFA model.

We show that under SecureVF malicious sensors are detected as soon as their movements violate the rules of the deployment algorithm. We perform extensive simulations in order to validate the analytical model and experimentally investigate the SecureVF ability to counteract the OM attack, in comparison with a previous solution based on VFA [3].

The original contributions of this paper are the following:

- We investigate the vulnerabilities of mobile sensor deployment algorithms based on VFA and propose a very simple and effective attack, called OM (Opportunistic Movement), to this approach.
- We provide an analytical model to estimate the effects of the OM attack on a general VFA solution.
- We propose a new algorithm based on VFA, called SecureVF to counteract the OM attack.
- Through simulations, we confirm the results provided by our analytical model and we study the efficacy of SecureVF to resist to the OM attack.

II. The problem of security in mobile sensor deployment algorithms

Prior work on security in wireless networks study several important problems that may affect static networks as well as mobile networks.

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. Such compromised nodes may perform several types of attacks in a mobile sensor network, influencing the behavior of legitimate nodes. A huge amount of work deals with the problem of confidentiality and integrity of communications [9], [10], [11]. Other works address the problem of the sybil attack, under which a malicious node may pretend to be many (sybil) nodes [12]. The problem of false position claims is also dealt with in several previous works [13], [14].

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. By contrast, in this paper we show that even if the best security mechanisms are in place to counteract the above mentioned attacks, it is still possible to severely compromise the functionality of a mobile sensor network, by adopting attacks which are specifically designed to compromise movement assisted deployment. In particular, we introduce the OM attack, specifically tailored to compromise deployment algorithms based on virtual forces. The OM attack does not exploit any of the security vulnerabilities previously described and works even if the network is endowed with top notch network security mechanisms. Instead, the OM attack exploits vulnerabilities that are inherent to the specific protocol that is required to let sensors coordinate with each other and spread throughout the AoI.

We show that, by compromising very few nodes and by performing the OM attack, the attacker can easily preclude a complete coverage of the AoI, by creating uncovered areas or corridors, thus impeding the network to fulfil its coverage requirements.

III. Adversary model and goals

We consider an adversary which introduces some malicious 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 easily be recognized by legitimate nodes, as they are able to send valid messages, since each of them has a valid ID and makes use of the legitimate cryptographic information. The attacker can thus exploit these corrupted nodes to perform malicious attacks to prevent a successful network deployment over the AoI.

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 [15].

We assume that malicious nodes can collude with each other by performing coordinated movements and communications in order to influence the movements of legitimate sensors.

In the next sections we first introduce a generalized VFA based algorithm which models several algorithms previously proposed in the literature. We then introduce the OM attack.

IV. A general virtual force based algorithm

In this section we introduce a Generalized Virtual Force algorithm (GVF) which generalizes several algorithms previously proposed in the literature.

We make the typical assumptions found in the literature 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.

The GVF algorithm is round based and sensors are loosely synchronized. Each round has two phases. During the first phase sensors exchange information such as their position and ID. In the second phase, each sensor calculates the virtual force acting on itself on the basis of the gathered information and moves towards the so calculated destination.

The calculation of the virtual forces acting on a sensor is executed 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 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 $r_f \leq R_{tx}$.

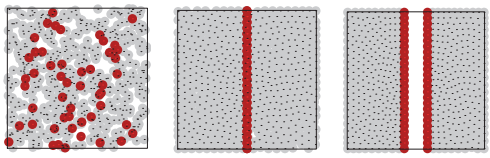
We hereby define the *area of influence* of a sensor s the area in which s exerts its virtual force on other sensors. Due to homogeneity, the area of influence of a sensor s is also the area from which other sensors exert a force on s. This area includes all the points at a distance lower then r_f from 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 [1], [2], [3], [4].

V. The Opportunistic Movement attack

The OM attack is defined on the basis of the adversary model described in Section III. 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*. From such a configuration, malicious nodes start the attack by communicating with legitimate sensors and by gradually adjusting their position so as to exert forces on legitimate sensors that cause their movement away from a specific area of interest to the attacker.

Since malicious nodes move silently to their position in the attack configuration, they are not detected by legitimate



(a) (b) (c)

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

sensors in this initial phase of the attack. After the formation of the attack configuration malicious nodes move according to the attacker strategy but communicate according to the communication protocol provided by the deployment algorithm. By communicating their position at each round, malicious sensors 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 Figure 1(a). They perform an initial silent movement so as to form two superimposed barriers, as shown in Figure 1(b). 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 Figure 1(c).

The opportunistic movement attack is a general attack which can be performed in may ways, by realizing different attacking configuration and adopting different moving strategy of malicious nodes.

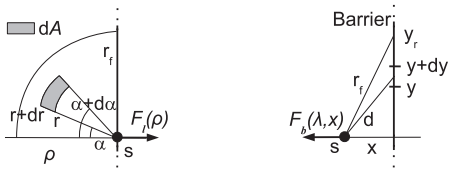
We now define the Barrier Opportunistic Movement (BOM) attack, a specific type of OM attack, which is able to severely reduce the coverage provided by the network while requiring only few sensors to be compromised.

According to the BOM attack, malicious nodes form a linear barrier whose edges intersect the borders 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 Figures 3(a-c). 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, such as in the attack shown in Figures 5(a-b).

The size of the area in which the legitimate sensors can be confined without crossing the barrier depends on the density of legitimate and of malicious sensors. In the following section we provide an analytical model to estimate the impact on network coverage that can be achieved by performing the BOM attack under the use of GVF.

VI. 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.



(a) (b)

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

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

A. Force exerted by uniformly distributed 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. Figure 2(a) 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 minor radius r and major radius $r + \mathrm{d}r$, such that it forms an angle α with the horizontal axis passing on s and it spans over an angle $\mathrm{d}\alpha$.

The area $\mathrm{d}A$ can be approximated as $r\mathrm{d}r\mathrm{d}\alpha$, 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 $F(r)\cos(\alpha) \cdot \rho\mathrm{d}A$. Hence, we can obtain the force acting on s by integrating on α and r:

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

B. Force exerted by the barrier

Let us consider a barrier of equally spaced malicious sensors with 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 Figure 2(b). 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 $\lambda\mathrm{d}yF(d)\frac{x}{d}$.

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:

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

C. Estimation of the effect of the BOM attack

The analytical model provided in the previous Sections VI-A and VI-B 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 it if the force exerted by the barrier is stronger than the one provided by legitimate sensors. 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 due to the movement of the barrier. Thus, as long as no legitimate sensor crosses the barrier, the density ρ of legitimate sensors over the MA increases and corresponds to the ratio $N/|\mathrm{MA}|$, as these sensors gradually adjust their positions so as to reach a uniform distribution.

We assume that the distance of the barrier from frontline sensors is larger than x_{max}, where x_{max} is the minimum distance from the barrier at which the force exerted by the barrier is maximised[1].

The BOM attack reaches its maximum effect when the density of legitimate sensors is so high that its pressure balances the force exerted by the barrier over frontline sensors, therefore $F_l(\rho) = F_b(\lambda, x_{max})$, and the monitored area MA reaches its minimum value *mMA*. *mMA* can therefore be calculated as follows:

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

A similar approach can be used to estimate how many legitimate sensors are needed in order to ensure that *mMA* is not smaller than a given value.

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

[1]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 $F(d)$.

VII. Analytical model for the PDND algorithm

We now apply the model described in the previous Section VI to the PDND algorithm [3]. PDND is considered one of the best algorithms 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 IV, 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:

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

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:

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

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

VIII. 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 VFA based deployment algorithms by providing additional phases, namely *movement verification phase* and *trusted neighbors communication phase*.

A. Assumptions

SecureVF is designed on the basis of the adversary model introduced in Section III. In particular, we assume that public key cryptography is in use in order to guarantee integrity and authentication of the exchanged messages. Notice that the use

of public cryptography in sensor networks is now commonly assumed [16], [15].

We also assume the presence of a location verification protocol [17], [18]. In particular, we assume that a node is able to verify the position of every other node located in its radio proximity. 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. Such an assumption is generally valid: the communication range of sensors is typically 75m-100m in outdoor environments [19], while it is generally assumed that $r_f < 3R_s$ [3], and R_s seldom exceeds a few meters [20]. 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 with a high level of noise, the distance r_f can be reduced accordingly.

We also assume a maximum moving distance per round of $1/2\,(R_{tx} - r_f)$. This ensures that any two sensors in the area of influence each other at a given round will not loose connectivity at the successive round[2].

Finally, similar to previous works on mobile sensor deployment, we assume that nodes are endowed with low cost GPS [21], [22] and that they are loosely synchronized.

B. Nomenclature

Let $C^t(s)$ be the set of sensors in the area of influence of the sensor s at round t. We also denote with $N^t(s)$ the set of sensors which are in communication range with s. Since $r_f < R_{tx}$, $C^t(s) \subseteq N^t(s)$.

According to SecureVF, in order to calculate the force acting on itself, 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 rount t is referred to as $N^t_{untrusted}(s)$. Finally, the position of sensor s at the current round is denoted with $pos^t(s)$.

C. The algorithm

SecureVF extends VFA based solutions with mechanisms for malicious node discovery and isolation. In particular, 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. Notice that we do not consider localization errors of the GPS positioning system or of the location verification algorithm. SecureVF can be extended to take into account these aspects.

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

Position communication phase (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 lies about its position, s immediately marks it as untrusted.

Movement verification phase (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 thanks to the assumption $R_{tx} > 2r_f$. As a result, a sensor s verifies, for each sensor q in $C^{t-1}(s)$, not yet in $N^t_{untrusted}(s)$, the correctness of the movement of q at the previous round[3].

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). Two inconsistencies can be detected by s. 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.

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, the sensor 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. 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.

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

Algorithm SecureVF, executed by 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 phase:
20 Calculate F^t(s) on the basis of N^t_trusted(s);
21 Move according to F^t(s);
```

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

D. Security analysis

The assumption $R_{tx} \geq 2r_f$ has direct implication in terms of capability of a legitimate sensor to detect a malicious behavior. Indeed, all the sensors located in the area of influence of a sensor s can only be affected by sensors located in the transmission range of s, whose position is therefore verifiable by s.

We recall that, by assumption, sensors can rely on mechanisms to also detect false identities. False identities and position claims are treated by SecureVF according to standard techniques. The algorithm SecureVF lets legitimate sensors mark as untrusted and then ignore other sensors performing these malicious activities, as soon as they are discovered.

As a consequence of the above assumptions the only possibilities for a malicious sensor m to influence the deployment of the network are the following:

- *Type 1: malicious set formation.* m classifies some legitimate sensors as untrusted and moves according to a maliciously formed set of trusted sensors;
- *Type 2: malicious movement.* m performs a movement which is not compliant to the force calculated on the basis of its set of trusted sensors.

Notice that a legitimate sensor never classifies another legitimate sensor as malicious, due to the correct behaviour of both sensors in terms of set formation and movements.

Let us consider malicious behavior of type 1. Such a behavior is detected by the legitimate sensors located in the area of influence of m which have not been included in its trusted set. Since such legitimate sensors know that they have behaved correctly, there is no legitimate reason for m to include them in its untrusted set. Notice that, the detection of this type of malicious behavior ensures that m cannot justify a null movement by advertising an empty trusted set.

We now address malicious behavior of type 2 by providing the following lemma.

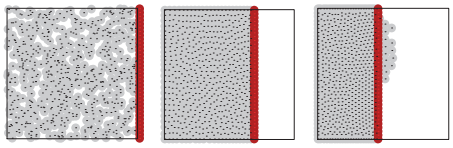
Lemma 1. *Given a legitimate sensor s and a malicious sensor m located in its area of influence at round t, s is capable of detecting a malicious behavior of type 2 performed by m.*

Proof: (Sketch) Since m is in the area of influence of s, the assumption that $R_{tx} \geq 2r_f$ implies that s knows the nodes belonging to the set $C^{t-1}(m)$ that influence the movement of m. As the maximum moving distance is $1/2\,(R_{tx} - r_f)$, s is still in communication with m at round $t+1$, so it can verify whether m moved in compliance with the force calculated on the basis of its trusted set or not. ■

Thanks to the above lemma we can assert that SecureVF is capable to neutralize a BOM attack. Indeed, since under the BOM attack the movement of malicious sensors is regulated by the attacking strategy, this movement is unlikely compliant with the trusted sets of malicious sensors, and especially if a malicious movement occurs for several consecutive rounds. Therefore, under the BOM attack, malicious nodes perform a malicious behavior of type 2. Thanks to lemma 1, legitimate sensors in the barrier proximity detect a malicious movement of type 2 within their area of influence and consequently ignore the malicious sensors performing it. An immediate consequence of this lemma is that SecureVF is able to neutralize also attacks with multiple barriers, such as the one depicted in Figure 1 intended to form an unmonitored corridor.

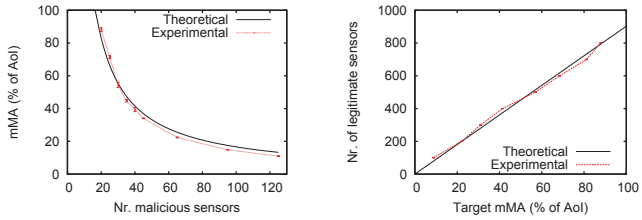
More in general, SecureVF is effective against all attacks characterized by the fact that all malicious nodes perform malicious behavior. Nevertheless, it is possible to formulate attacks in which the set of malicious sensors is split in two teams. The first team performs malicious movements, as in the BOM attack, while the second team surrounds the first team but moves in compliance to the VFA algorithm. If the layer of the second team is sufficiently thick, legitimate nodes 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



(a) (b) (c)

Fig. 3. BOM attack on a network running the algorithm PDND



(a) (b)

Fig. 4. mMA by increasing the number of malicious sensors (a) and minimum number of legitimate sensors to ensure a target mMA (b).

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.

IX. Experimental results

In this Section we experimentally confirm the results provided by the analytical model proposed in Section VI and we study the efficacy of SecureVF to counteract the BOM attack. In order to do so, we developed a simulator on the basis of the Wireless Module of the Opnet simulation environment [23]. We use the following simulation parameters: $R_s = 5m$, $R_{tx} = 25m$, $r^* = \sqrt{3}R_s$, $r_f = 1.2r^*$, moving speed $1m/s$, size of the AoI $150\text{x}150m^2$.

A. Validation of the analytical model

In this Section we verify through simulations the capability of the analytical model 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 we defined as the portion of the AoI in which the legitimate sensors are confined without crossing the barrier. In particular, we use the analytical model of the impact of the BOM attack described in Section VI-C to answer the following questions and confirm the results through simulations: (Q1) Given a number of legitimate sensors, which is the minimum monitored area (mMA) as a function of the number of malicious sensors deployed? (Q2) Given a number of malicious sensors, how many legitimate sensors are needed in order to ensure that the MmMA is not smaller than a certain value?

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. Figure 3(a) shows the considered scenario with 500 legitimate sensors (black dots) and 35 malicious ones (red dots). The grey and red circles are the sensing areas of legitimate and malicious sensors, respectively. Malicious sensors perform the BOM attack by moving the barrier from the right to the left. When the barrier starts moving across the AoI legitimate sensors are repelled, resulting in a reduction of the monitored area (Figure 3(b)). 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 pass through it (Figure 3(c)).

We performed two sets of the experiments in which the results obtained through the analytical model are compared to those 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% of legitimate sensors cross the barrier.

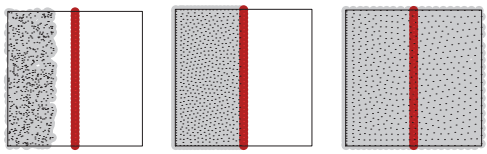
In the first set of experiments, we deploy 500 legitimate sensors and we increase the number of malicious sensors. Figure 4 (a) shows the results of the first set of experiments. The theoretical analysis shows a good fit with the experimental curve. The results show that even a small number of malicious nodes can cause serious damage to the network. As a numerical example, the attacker is able to reduce the mMA to less than 50% of the AoI by compromising only the 7% 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, shown in Figure 4 (b), we deploy 30 malicious sensors and show the minimum number of legitimate sensors that are necessary to balance the effect of the barrier when the size of the mMA corresponds to a given target value. Also in this case the analytical model shows a good fit with simulations in predicting the effect of the BOM attack. In order to achieve a mMA larger than 80% 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.

B. Performance evaluation of SecureVF

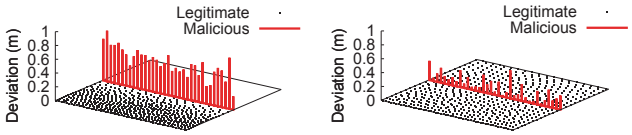
In this section we experimentally show the efficacy of SecureVF against the BOM attack and evaluate its performance. Notice that, SecureVF can adopt the force formulation of any deployment algorithm modeled by the GVF algorithm introduced in Section IV. In the experiments SecureVF is based on the force formulation of PDND, and we compare it to the basic version of PDND.

In order to evaluate the performance of SecureVF, we consider a scenario where malicious sensors form a barrier that splits the AoI in two halves. Legitimate sensors are randomly deployed on the left side of the barrier. Malicious sensors do not move but honor the communication protocol according to the OM attack. In particular, under SecureVF, malicious sensors advertise a trusted set that contains every node in their



(a) (b) (c)

Fig. 5. Initial deployment (a), final deployment under PDND (b) and under SecureVF (c)



(a) (b)

Fig. 6. SecurVF: average deviation measured by the legitimate sensors

area of influence to avoid easy detection due to a malformed set.

Before showing the experimental results, we give an example of the execution of PDND and of SecureVF in the considered scenario. In the example 500 legitimate sensors and 35 malicious sensors are deployed. Figure 5(a) shows the initial sensor distribution while Figures 5(b) and (c) show the final deployment achieved under PDND and SecureVF, respectively. Since the barrier density is sufficiently high, no legitimate sensor is able to cross the barrier under PDND. On the contrary, SecureVF is able to cover the AoI: legitimate sensors detect the malicious movement of the sensors on the barrier which is not compliant with their trusted set. As a result, malicious sensors are ignored and full coverage is achieved.

Figures 6 (a-b) show a 3D representation of the execution of SecureVF. The z-axis shows the average deviation, with respect to the correct movement, measured by legitimate sensors in their area of influence.

The measured deviation is higher when all legitimate sensors are still on one side of the barrier, since the force exerted on the malicious nodes is higher. Such a deviation decreases as the legitimate sensors cross the barrier due to the presence of legitimate nodes on both sides which partially balances the forces exerted on malicious nodes, making the choice of remaining still closer to the correct movement.

We now show the performance comparison between PDND and SecureVF in the considered scenario. We also show the performance of PDND in absence of malicious sensors (PDND-Free in the figures). We studied several performance metrics by increasing the number of legitimate sensors while the number of malicious sensors is 35. Such metrics are related to legitimate sensors only.

Figure 7(a) shows the coverage achieved by the two algorithms. SecureVF is not affected by the presence of malicious sensors. Legitimate sensors are able to detect the incorrect behaviour of malicious sensors and consequently ignore them. As a result, SecureVF achieves the same coverage of PDND-Free. On the contrary, the PDND algorithm is strongly affected by the presence of the barrier. The malicious sensors are able to impede the spread of legitimate sensors by confining them to the left side of the barrier. Coverage increases when the number of legitimate sensors is sufficiently high to let some of them cross the barrier.

Figures 7(b-c) depict the traversed distance and the number of movements, respectively. SecureVF requires the same traversed distance and number of movements of PDND-Free, showing that our algorithm introduces no overhead in terms of movements since it behaves as if malicious sensors were not present. Notice that, when the number of legitimate sensors is close to the minimum to achieve full coverage (400 sensors in our setting) more movements are required to the sensors in order to find their final positions due to poor redundancy. Under PDND, both metrics start increasing as soon as there are sufficient legitimate sensors to let some of them cross the barrier.

In Figures 7(d) 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], [24], [22], [25]: receiving a message costs 1eu, sending a message 1.125eu, 1m movement and starting/stopping a movement cost the same as 300 messages.

SecureVF has a higher communication cost with respect to PDND. Indeed, under SecureVF each sensor has to communicate at each round its trusted set. Thus, the cost of communications increases with the sensor density. Nevertheless, SecureVF does not introduce overheads in terms of movements, which are known to be the most energy demanding activity of mobile sensors. As a result, the energy consumption under SecureVF is close to the one of PDND-Free. The overhead introduced by SecureVF allows legitimate sensors to detect malicious sensors and to ignore them, thus letting the network achieve its coverage goals.

Figure 7(e) shows the termination time of the algorithms, that is the time at which the sensors stop moving. SecureVF shows a higher termination time than PDND-Free. This is due to the longer communication phase of each round, necessary to exchange the trusted set. Nevertheless, the termination time of PDND-Free is on average only 30% lower than the one of SecureVF. PDND terminates earlier because a large fraction of legitimate sensors is confined on the left side of the barrier.

The experiments reported above may seem to show that, by deploying 1000 sensors, PDND achieves better performance than SecureVF. Nevertheless, it is sufficient to slightly increase the number of malicious sensors in order to make 1000 sensors unable to cross the barrier. In order to show this, we perform some experiments by fixing the number of legitimate sensors to 1000 and by increasing the number of malicious sensors. Figure 7(f) shows the obtained results, pointing out that, by compromising only 5% of the legitimate sensors, coverage drops to the 50% of the AoI.

In order to further motivate the advantage of SecureVF with respect to PDND, we experimentally calculate the minimum

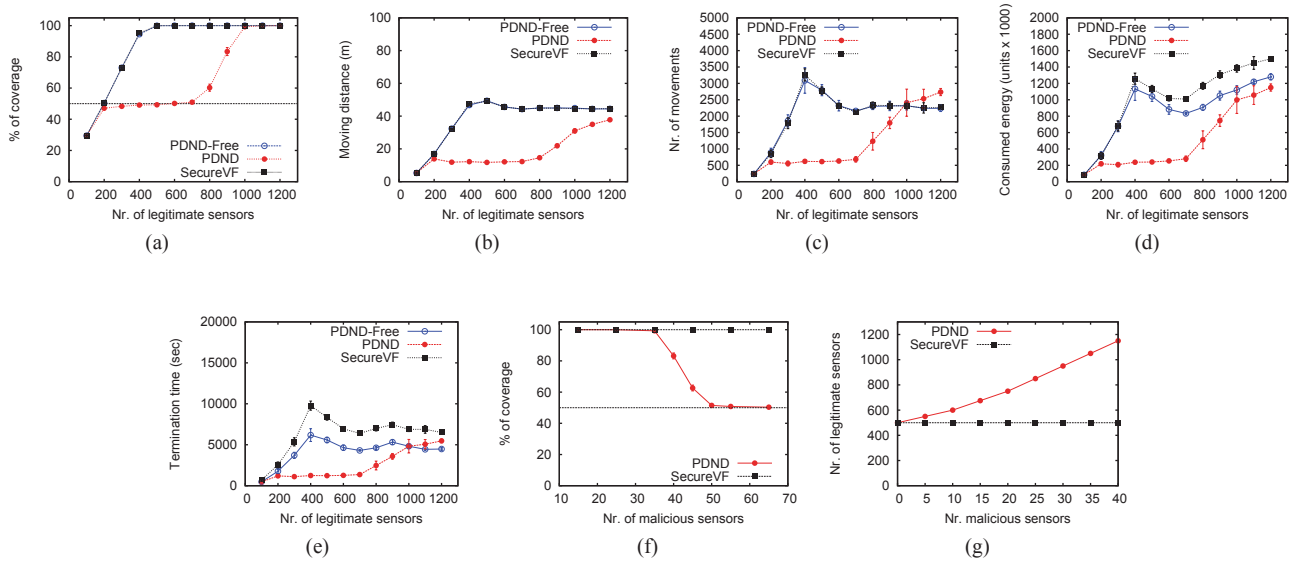


Fig. 7. Coverage (a), traversed distance (b), number of movements (c), consumed energy (d), termination time (e). Coverage achieved with 1000 legitimate sensors (f) and minimum number of sensors need to achieve full coverage (g).

number of legitimate sensors required to achieve full coverage of the AoI by increasing the number of malicious sensors. Figure 7(g) shows the obtained results. SecureVF requires 500 sensors to achieve full coverage independently of the number of malicious sensors deployed. By contrast, PDND shows a linear increase in the number of necessary legitimate sensors to cover the AoI with respect to the number of malicious sensors.

X. Conclusions

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 SecureVF to counteract the OM attack. We show that SecureVF enables the detection of malicious behaviours and we investigate its performance through simulations. Results show that under SecureVF the coverage goals of the network are achieved at the expense of a low additional energy consumption.

References

[1] N. Heo and P. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Trans. on Syst., Man and Cyb.*, vol. 35, no. 1, 2005.

[2] J. Chen, S. Li, and Y. Sun, "Novel deployment schemes for mobile sensor networks," *Sensors*, 2007.

[3] K. Ma, Y. Zhang, and W. Trappe, "Managing the mobility of a mobile sensor network using network dynamics," *IEEE Trans. on Paral. and Distr. Syst.*, vol. 19, no. 1, 2008.

[4] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," *ACM Trans. on Emb. Comp. Syst.*, vol. 3, no. 1, 2003.

[5] M. Garetto, M. Gribaudo, C.-F. Chiasserini, and E. Leonardi, "A distributed sensor relocation scheme for environmental control," *ACM/IEEE MASS*, 2007.

[6] M. R. Pac, A. M. Erkmen, and I. Erkmen, "Scalable self-deployment of mobile sensor networks; a fluid dynamics approach," *IEEE IROS*, 2006.

[7] M. Lam and Y. Liu, "Two distributed algorithms for heterogeneous sensor network deployment towards maximum coverage," *IEEE ICRA*, 2008.

[8] S. Poduri and G. S. Sukhatme, "Constrained coverage for mobile sensor networks," *IEEE ICRA*, 2004.

[9] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," *IEEE Symposium on 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," *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," *IEEE PerCom*, 2005.

[12] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," in *ACM IPSN*, 2004.

[13] K. Liu, N. Abu-Ghazaleh, and K. Kang, "Location verification and trust management for resilient geographic routing," *Elsevier JPDC*, vol. 67, no. 2, 2007.

[14] S. Capkun, K. Rasmussen, M. Cagalj, and M. Srivastava, "Secure location verification with hidden and mobile base stations," *IEEE Trans. on Mobile Computing*, vol. 7, no. 4, 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 Selected Areas in Communications*, vol. 28, no. 5, 2010.

[16] K. Xing and X. Cheng, "From time domain to space domain: Detecting replica attacks in mobile ad hoc networks," *IEEE INFOCOM*, 2010.

[17] J. Gao, R. Sion, and S. Lederer, "Collaborative location certification for sensor networks," *ACM Trans. on Sensor Networks*, vol. 6, no. 4, 2010.

[18] Y. Wei, Z. Yu, and Y. Guan, "Location verification algorithms for wireless sensor networks," *IEEE ICDCS*, 2007.

[19] Crossbow, "Telosb datasheet," *www.willow.co.uk/TelosB_Datasheet.pdf*.

[20] MAXBOTIX, "sonar datasheets," *http://www.maxbotix.com/uploads/LV-MaxSonar-EZ1-Datasheet.pdf*.

[21] G. Wang, G. Cao, and T. La Porta, "Movement-assisted sensor deployment," *IEEE Trans. on Mobile Computing*, vol. 5, no. 6, 2006.

[22] N. Bartolini, T. Calamoneri, T. La Porta, and S. Silvestri, "Autonomous deployment of heterogeneous mobile sensors," *IEEE Trans. on Mobile Computing*, vol. 10, no. 6, 2011.

[23] "Opnet technologies inc." *http://www.opnet.com*.

[24] G. Sibley, M. Rahimi, and G. Sukhatme, "Mobile robot platform for large-scale sensor networks," *IEEE ICRA*, 2002.

[25] 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, 2010.