Voronoi-based Deployment of Mobile Sensors in the Face of Adversaries

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

T. La Porta, S. Silvestri
Department of Computer Science
and Engineering
Pennsylvania State University, USA
{tlp, simone}@cse.psu.edu

F. Vincenti
Department of Computer Science
Sapienza University of Rome, Italy
vincenti@di.uniroma1.it

***Abstract*—Mobile sensor networks enable the monitoring of remote and hostile environments without requiring human supervision. Several approaches have been proposed in the literature to let mobile sensors self-deploy over a region of interest. In this paper we study, for the first time, the vulnerabilities of one of the most referenced approaches to mobile sensor deployment, namely the Voronoi-based approach. We show that, by compromising a small number of sensors, an attacker can influence the sensor deployment causing a significant reduction of the monitoring capability of the network. We propose a secure deployment algorithm called SecureVOR. We formally prove that SecureVOR has guaranteed termination and that it allows legitimate sensors to detect the malicious behavior of compromised nodes. We also show by extensive simulations that SecureVOR is able to fulfill the network monitoring goals even in presence of an attack, at the expense of a small performance overhead.**

I. Introduction

Mobile wireless sensor networks are composed by small and relatively cheap devices with sensing, communication and locomotion capabilities [1]. In order to fully exploit the monitoring capabilities of these networks, distributed deployment algorithms have been designed to let mobile sensors self-deploy over an Area of Interest (AoI). These algorithms can be roughly classified in three major families on the basis of *regular patterns* [2], [3], *virtual force models* [4], [5] or *computational geometry* [6], [7], [8].

Similar to static wireless sensors, mobile sensors lack tamper proof hardware, thus an attacker can capture several nodes, extract their cryptographic material and reprogram them according to its malicious goal. This leads to several security issues, which have been largely studied, such as security of communication [9], [10], false position claims [11], sybil [12] and node replication [13] attacks. Nevertheless, even if the network is endowed with security mechanisms such as the ones cited above, an attacker can still alter the sensor deployment by exploiting the vulnerabilities of the self-deployment phase. As an example, an attacker may reduce the area in which the sensors are deployed, thus creating an unmonitored corridor or an unmonitored zone. In the rest of the paper we will refer to the nodes compromised by an attacker as *malicious sensors* whereas we will refer to the rest of the nodes as *legitimate sensors*.

The above mentioned vulnerabilities of deployment algorithms in mobile sensor networks have not been considered in the literature. Only recently in [14], we investigated the security issues of the Virtual Force Approach (VFA), introducing an attack called Opportunistic Movement (OM) tailored for mobile sensor deployment algorithms. The authors show that the OM attack can severely reduce the provided coverage, even if the network is already deployed. The paper proposes a counter measure also based on virtual forces.

In this paper we study, for the first time, the vulnerabilities of a completely different approach, namely the Voronoi-based approach [6], [7], which is one of the most referenced approaches to mobile sensor deployment. In particular, we tailored the OM attack introduced in [14] to deal with Voronoi-based deployment algorithms.

We show that, unlike with VFA, under the Voronoi approach, the OM attack has no impact on a previously deployed network. Nevertheless, we show that such an attack can significantly limit the coverage provided by the network during the deployment phase. The efficacy of the attack depends on the number of malicious sensors necessary to create a barrier to confine the area that the attacker wants to keep uncovered.

We propose SecureVOR, a new secure Voronoi-based deployment algorithm designed to counteract the OM attack and we show that SecureVOR is able to cover the AoI, even in the presence of the attack, at the expense of a small increment in energy consumption. Furthermore, we prove that SecureVOR has a guaranteed termination.

II. Voronoi-based deployment

In this section we describe the Voronoi-based Deployment Algorithm (VDA) introduced in [6]. The algorithm assumes that a sensor communicates within a distance R_{tx} (*communication radius*), it senses over a circular area of radius R_s (*sensing radius*), with $R_{tx} > 2R_s$. Nodes can move in any direction inside the AoI, are endowed with low cost GPS, and are loosely synchronized.

VDA is round based; each round has two phases: (1) position communication, (2) coverage evaluation and movement. Let us consider a sensor s at a round t. During the position communication phase, s exchanges information with its neighbors regarding positions and IDs. On the basis of the gathered information, s determines the neighbor set $N^{(\mathtt{t})}(s)$.

In the coverage evaluation and movement phase, s calculates its Voronoi polygon $V^{(\mathtt{t})}(s)$, considering the nodes in $N^{(\mathtt{t})}(s)$, and evaluates the coverage of its polygon to decide whether to move or not. In particular, if s detects a coverage hole in $V^{(\mathtt{t})}(s)$ then it determines a point inside its polygon

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

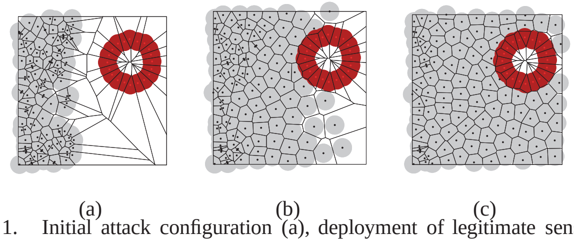

(a)
(b)
(c)

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

where it can contribute a better coverage. In [6] the authors propose two different approaches to determine the destination point of s, namely the *Farthest Vertex* (FV) and the *MiniMax* (MM) approach. FV dictates that a sensor s moves towards the farthest vertex V_F of its polygon and stops at a distance R_s from it. According to MM, s moves to the point that minimizes the maximum distance from the vertices of $V^{(t)}(s)$.

Notice that, s moves to its destination only if its movement provides a better coverage of $V^{(t)}(s)$. According to [6], s can traverse a maximum distance per round $d_{max} = R_{t_x}/2 - R_s$, in order to take account of possible inaccuracies in the distributed construction of Voronoi polygons.

III. The OM attack

The OM attack is a simple and effective attack specifically designed to alter the deployment of mobile sensors. It has been introduced for the first time in [14].

In the following we will assume that network security mechanisms are in place to let each node detect sybil attacks [12], perform location verification [11] and exchange messages in a secure manner [9], [10]. Furthermore, we assume that the attacker cannot create clones of the compromised nodes [13].

According to the OM attack malicious nodes, once deployed in the AoI, 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 in a legitimate way, malicious sensors influence the movement of legitimate nodes.

Let us consider the following example in which the adversary creates an uncovered area over the AoI. Malicious sensors are initially deployed by the attacker around the zone it wants to keep uncovered, as depicted in Figure 1(a). Black dots are legitimate sensors and grey circles their sensing ranges. Similarly, red dots are malicious sensors and red circles their sensing ranges. The figure also shows the Voronoi diagram partitioning the AoI. Legitimate sensors arrive in proximity of the malicious nodes as shown in Figure 1(b). Since malicious sensors advertise their positions according to the rules of the communication protocol, legitimate sensors are not able to penetrate the attacker zone. As a result, the attacker successfully precludes the deployment over the target area as shown in Figure 1(c).

IV. The SecureVOR algorithm

In this Section we introduce SecureVOR, a secure Voronoi-based deployment algorithm. SecureVOR provides a method to detect malicious movements when the deployment is based on VDA and can be applied to both moving strategies FV and MM.

The idea of SecureVOR is to detect malicious nodes by verifying the compliance of their movements to the rules of the deployment algorithm in use. In order to detect malicious movements, each sensor s has to declare at each round the set of its neighbors that it considers as *trusted* and that it is using to determine its polygon. Notice that, a sensor determines this set only on the basis of its local observation since SecureVOR does not require transitive trust among sensors. Neighbor sensors of s locally calculate the polygon of s, based on its stated trusted set, and verify whether its movement is in compliance with the deployment algorithm or not. If a malicious movement is detected, s is marked as untrusted and ignored in further rounds by its neighbors.

According to SecureVOR, a sensor s only considers neighbors at a distance less that $R_{tx}/2$ to calculate its own polygon. We refer to such neighbors at a round t as $Q^{(t)}(s)$. This choice enables s to be in communication with the sensors considered by its neighbors in $Q^{(t)}(s)$ to determine their polygon. Obviously, $Q^{(t)}(s) \subseteq N^{(t)}(s)$.

Among the nodes in $Q^{(t)}(s)$, s takes into account only the sensors that it considers as trusted in order to determine its polygon. We refer to the set of such trusted sensors as $N^{(t)}_{trusted}(s)$ while the set of untrusted sensors discovered until round t is referred to as $N^{(t)}_{untrusted}(s)$. Finally, the position of sensor s at the current round is denoted with $pos^{(t)}(s)$.

SecureVOR is round based similar to VDA. In particular, it comprises four phases, namely: *Position communication*, *Movement verification*, *Trusted neighbors communication* and *Coverage evaluation and movement*. Notice that we do not consider localization errors of the GPS positioning system or of the location verification algorithm. SecureVOR can be extended to take into account these aspects. The pseudo-code of the algorithm is shown as Algorithm SecureVOR.

In the following we assume that $R_{tx} > 4R_s$ and we set $d_{max} = R_{tx}/4 - R_s$. Such an assumption is generally valid as R_{tx} is typically 75m-100m [15], while for most sensors R_s seldom exceeds a few meters [16]. Furthermore, we assume the presence of a signature protocol to guarantee authentication of the exchanged messages.

Position communication (lines 1-3)
At the beginning of a round each sensor communicates its position to the neighbors through a signed message and determines the sets $N^{(t)}(s)$ and $Q^{(t)}(s)$.

Movement verification (lines 4-18)
In this phase, a sensor s verifies the movements of its neighbors to determine $N^{(t)}_{trusted}(s)$ and $N^{(t)}_{untrusted}(s)$. At the first round, $N^{(t)}_{trusted}(s) = Q^{(t)}(s)$ and $N^{(t)}_{untrusted}(s) = \emptyset$ (lines 4-6).

The set of untrusted neighbors at round $t > 1$, $N^{(t)}_{untrusted}(s)$, contains all the sensors of $N^{(t-1)}_{untrusted}(s)$ plus the sensors that were in $Q^{(t-1)}(s)$ and that now are not in communication with s (line 8). [1] Other sensors that are detected as malicious in the current round are added to $N^{(t)}_{untrusted}(s)$ (lines 9-18) as explained in the following.

A sensor s verifies, for each sensor q in $Q^{(t-1)}(s)$, not yet in $N^{(t)}_{untrusted}(s)$, the correctness of its movement at the previous round[2].

[1] SecureVOR imposes that a sensor travels a maximum distance $d_{max} = R_{tx}/4 - R_s$. Hence even if two sensors, at a distance at most $R_{tx}/2$, move in opposite directions, they will stop at a distance from each other less than $R_{tx}/2 + 2(R_{tx}/4 - R_s)$ which is less than R_{tx}. This means that $Q^{(t-1)}(s) \subseteq N^{(t)}(s)$, so if a sensor in $Q^{(t-1)}(s)$ is not in $N^{(t)}(s)$, s can mark it as untrusted.

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

Algorithm SecureVOR, 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 $Q^{(t)}(s)$;
    // Movement verification:
 4  if $t = 0$ then
 5      $N^{(t)}_{untrusted}(s) = \emptyset$;
 6      $N^{(t)}_{trusted}(s) = Q^{(t)}(s)$;
 7  else
 8      $N^{(t)}_{untrusted}(s) = N^{(t-1)}_{untrusted}(s) \cup (Q^{(t-1)}(s) \setminus N^{(t)}(s))$;
 9      for $q \in Q^{(t)}(s)$ s.t. $q \notin N^{(t)}_{untrusted}(s)$ do
10          if $q \notin Q^{(t-1)}(s)$ then $N^{(t)}_{trusted}(s) \leftarrow q$;
11          else
12              if $(s \notin N^{(t-1)}_{trusted}(q) \vee N^{(t-1)}_{trusted}(q) \nsubseteq N^{(t-1)}(s))$
                then
13                  $N^{(t)}_{untrusted}(s) \leftarrow q$;
14              else
15                  Calculate $V^{(t-1)}(q)$;
16                  Calculate $\widehat{pos}^t(q)$;
17                  if $\widehat{pos}^t(q) \neq pos^t(q)$ then $N^{(t)}_{untrusted}(s) \leftarrow q$;
18                  else $N^{(t)}_{trusted}(s) \leftarrow q$;

    // Trusted neighbors communication:
19  Broadcast the list of nodes in $N^{(t)}_{trusted}(s)$;
20  Receive $N^{(t)}_{trusted}(z)$ from any $z \in Q^{(t)}(s)$;
    // Coverage evaluation and movement :
21  Calculate $V^{(t)}(s)$ on the basis of $N^{(t)}_{trusted}(s)$;
22  if $V^{(t)}(s)$ is completely covered then do not move;
23  else Determine destination point and move accordingly.
```

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 itself 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 fabricated the presence of some sensors in $N^{(t-1)}_{trusted}(q)$ which are not physically located in its proximity to justify its movement. Sensor s can detect such malicious behavior because, according to SecureVOR, a sensor q must selects the sensors in $N^{(t-1)}_{trusted}(q)$ among those in $Q^{(t-1)}(q)$. In order to be in $N^{(t-1)}_{trusted}(q)$, a sensor must be at a distance at most $R_{tx}/2$ from q which implies that it is at a distance at most R_{tx} from s, being q at a distance at most $R_{tx}/2$ from s ($q \in Q^{(t-1)}(s)$). More formally $N^{(t-1)}_{trusted}(q) \subseteq Q^{(t-1)}(q) \subseteq N^{(t-1)}(s)$.

If an inconsistency is detected, q is marked as untrusted and will be hereafter ignored by s. 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 14-18). To this aim, s calculates the polygon of q at the previous round $V^{(t-1)}(q)$ on the basis of $N^{(t-1)}_{trusted}(q)$ and $pos^{(t-1)}(q)$. s then compares the current position $pos^{(t)}(q)$, which q has just broadcast in the previous phase with the expected position of q at the current round, $\widehat{pos}^{(t)}(q)$, calculated considering the polygon $V^{(t-1)}(q)$ and $pos^{(t-1)}(q)$. If $pos^{(t)}(q)$ is different from $\widehat{pos}^{(t)}(q)$, s marks q as untrusted.

Trusted neighbors comm. (lines 19-20)
In this phase each sensor s broadcasts a signed message containing 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.

Coverage eval. and movement (lines 21-23)
This phase is the same as in the Voronoi approach as described in Section III, except that each sensor s calculates its Voronoi polygon $V^{(t)}(s)$ on the basis of the sensors in $N^{(t)}_{trusted}(s)$. Furthermore s looks for a destination point p within a distance $d_{max} = R_{tx}/4 - R_s$ instead of $d_{max} = R_{tx}/2 - R_s$.

V. THEORETICAL ANALYSIS

The following Theorem shows that, under VDA, the OM attack has no impact on an already deployed network which provides full coverage of the AoI. Notice that, the theorem holds for both the FV and MM moving strategies.

Theorem 5.1: Under VDA, once legitimate sensors have achieved full coverage of the AoI, the OM attack cannot cause the movement of any sensors.

Proof sketch: Let us consider a legitimate sensor s with neighbors $N(s)$. Since the AoI is completely covered, $V(s)$ is also completely covered, hence s does not move. When the OM attack starts, s has a set of neighbors $\widehat{N}(s)$, which may include some additional malicious sensors, and a polygon $\widehat{V}(s)$. Since $N(s) \subseteq \widehat{N}(s)$ then $\widehat{V}(s) \subseteq V(s)$, thus $\widehat{V}(s)$ is also completely covered, hence s does not move. ■

We now study the capability of SecureVOR to counteract the OM attack and to terminate within a finite number of rounds. We denote by L and M the set of legitimate and malicious sensors, respectively.

Notice that, if a malicious node m moves in compliance to VDA it cannot be detected by SecureVOR, since it 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* of a malicious sensor as a movement which is not in compliance with the deployment rules. Furthermore, given a malicious sensor $m \in M$ performing a malicious movement at round t, we define the set L^t_m as the set of legitimate sensors whose movement can be influenced by the malicious movement of m.

Theorem 5.2: Given a malicious sensor $m \in M$ performing a malicious movement at round t, if $L^t_m \neq \emptyset$ then m is marked as untrusted by at least one sensor in L^t_m at round $t + 1$.

Proof: Since m can influence the movement of the sensors in L^t_m, such sensors consider m as trusted at the current round. Furthermore, since we assume that a node considers only sensors at a distance $R_{tx}/2$ to determine its polygon, $\forall s \in L^t_m$ $d(s, m) < R_{tx}/2$ thus s is able to verify if $N^{(t)}_{trusted}(m)$ is inconsistent. As a result, according to the assumptions made in Section III, the only degree of freedom that m has in order to try to justify its malicious movement without being detected lies in the selection of the nodes to be advertised as trusted.

Notice that all nodes in L^t_m are legitimate and are at a distance less than $R_{tx}/2$ 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 $N^{(t)}_{trusted}(m)$, then such sensors mark m as untrusted at round $t + 1$ and then the theorem is proved.

If, on the contrary, m includes all sensors in L^t_m in $N^{(t)}_{trusted}(m)$, such sensors are in communication range with m at round $t + 1$ since $\frac{R_{tx}}{2} + 2d_{max} < R_{tx}$. As a result, sensors in L_m^t are able to verify the correctness of the current movement of m at the next round. Since m is performing the OM attack, its malicious movement is detected and thus all sensors in L_m^t mark m as untrusted at round $t+1$. ■

In order to prove the termination of SecureVOR we show that, at each round, either at least one malicious sensor is detected or the overall coverage provided by legitimate sensors increases. We first show the convergence of the algorithm and then we discuss the termination.

Definition 5.1: A *network state* is a vector $S =< c_1, \dots, c_{|M|}, s_1, \dots, s_{|L|}, m_1, \dots, m_{|M|} >$ where c_j is the number of legitimate sensors which consider the malicious sensor $m_j \in M$ as untrusted, $s_i \in L$ for $i = 1, \dots, |L|$ and $m_j \in M$ for $j = 1, \dots, |M|$.

We define a function $f : \mathbb{N}^{|M|} \times L^{|L|} \times M^{|M|} \rightarrow \mathbb{N} \times \mathbb{R}_+$ such that given a network state S, $f(S) = (\sum_{j=0}^{|M|} c_j, A_{total})$, where A_{total} is the size of the area covered by legitimate sensors in S. Given two network states S_1, S_2 we say that $f(S_1) \prec f(S_2)$ according to the lexicographic order. Notice that, the function f is upper-bounded by the pair $(|L||M|, AoI)$. In the following, in order to prove the convergence of SecureVOR, we show that at each round the value of such function increases.

Theorem 5.3: The algorithm SecureVOR converges.

Proof: Let us consider a generic state change from round t to round $t+1$. We want to show that $f(S^{(\mathtt{t})}) \prec f(S^{(\mathtt{t+1})})$. We recall that, for a malicious sensor $m \in M$ performing a malicious movement at round t, L_m^t is the set of legitimate nodes whose movement can be influenced by the malicious movement of m. We consider two cases:

Case 1: $\exists m_j \in M$ s.t. $L_{m_j}^t \neq \emptyset$.

Thanks to Theorem 5.2 we know that there exist at least one legitimate sensor at round $t+1$ that marks m_j as untrusted. As a result, $c_j[S^{(\mathtt{t})}] < c_j[S^{(\mathtt{t+1})}]$, hence $f(S^{(\mathtt{t})}) \prec f(S^{(\mathtt{t+1})})$.

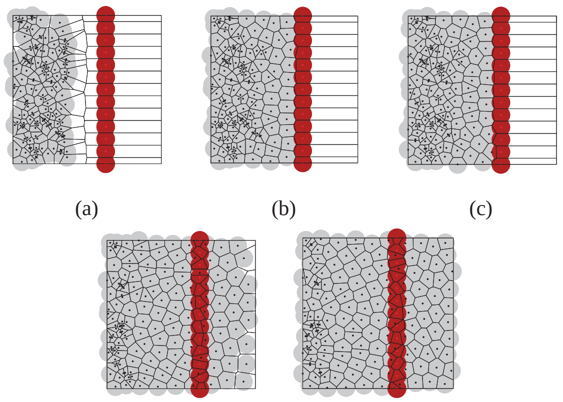
Case 2: $\forall m_j \in M$, $L_{m_j}^t = \emptyset$.

In this case no malicious movement influences the movement of legitimate sensors. As a result no malicious sensor is detected at round $t+1$, hence $\forall\ j = 1, \dots, |M|$, $c_j[S^{(\mathtt{t+1})}] = c_j[S^{(\mathtt{t})}]$. Notice that, if no malicious sensor is detected SecureVOR lets sensors deploy according to the rules of VDA. Since under VDA if at least one sensor moves then the provided coverage increases at each round [7], this holds also under SecureVOR[3]. As a result, if at least one sensor moves then $f(S^{(\mathtt{t})}) \prec f(S^{(\mathtt{t+1})})$.

The function f is upper-bounded and it increases at each round, as a result SecureVOR eventually converges. ■

The above theorem proves that SecureVOR converges, nevertheless the increase in coverage may be infinitesimal and thus the algorithm may require an infinite number of rounds to terminate.

Corollary 1: The algorithm SecureVOR terminates if movements are allowed only if they provide a coverage increase which exceeds a positive minimum threshold ϵ.


(a) (b) (c)

(d) (e)

Fig. 2. Initial deployment of 150 legitimate sensors and 13 malicious sensors (a), final deployment of FV (b), MM (c), SecureFV (d) and SecureMM (e).

The introduction of ϵ ensures fast termination and power saving, at the expense of a small loss in the coverage extension.

VI. Experimental results

In this Section we experimentally study the effects of the OM attack on VDA and the ability of SecureVOR to counteract such an attack. We develop a simulator on the basis of the wireless module of the Opnet simulation environment [17].

In the experiments we consider a squared AoI of size 80m×80m, we set $R_{tx} = 30$m and $R_s = 5$m. Sensors can move at a maximum speed of 1m/s. Under this setting, the maximum moving distance d_{max} under VDA is 10m, while under SecureVOR is 2.5m. We set the threshold $\epsilon = 0.001$.

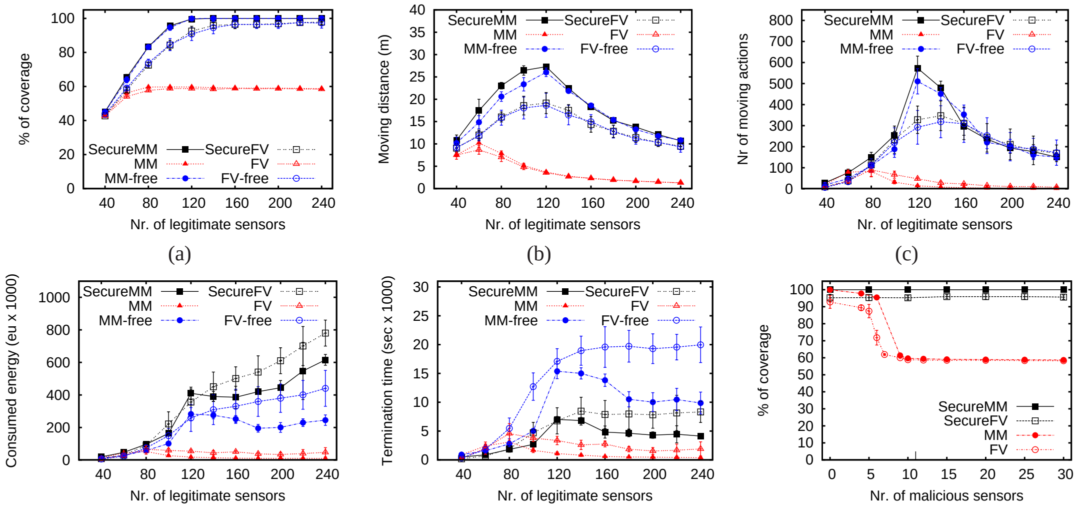
Static barrier of malicious sensors

In this set of experiments we consider a specific type of OM attack where malicious sensors form a static linear barrier, whose edges reach the borders of the AoI, in order to prevent legitimate sensors from spreading over the AoI. Malicious sensors perform the OM attack by periodically advertising their position during the Position communication phase while they remain still during the next phase. Under SecureVOR, each malicious sensor m, in order to avoid being easily detected by the surrounding legitimate sensors, advertises a trusted set $N_{trusted}^{\mathtt{t}}(m) = Q^{\mathtt{t}}(m)$. Legitimate sensors are randomly deployed on the left side of the AoI. In the experiments we set the number of malicious sensors to 13 and we increase the number of legitimate sensors from 60 to 240.

We compare the two moving strategies provided by VDA, namely Farthest Vertex (FV) and MiniMax (MM), with SecureVOR applied to both of them, to which we refer as SecureFV and SecureMM, respectively. For the sake of completeness, we show the results obtained by MM and FV in absence of malicious sensors, called MM - Free and FV - Free in the graphs. An example of the considered scenario with 150 legitimate sensors and 13 malicious sensors is depicted in Figure 2(a). Figure 2(b) and (c) show the detrimental effect of the barrier over FF and MM, while Figures 2(d) and (e) show that both SecureVOR and SecureMM are able to cross the barrier and fulfill the coverage requirements of the network.

Figure 3(a) shows the coverage of the AoI achieved by the considered algorithms. FV and MM are not able to cross the barrier of malicious sensors. As a result, sensors are confined on the left side of the barrier and coverage is always less

[3] In [7] the authors consider an extended VDA which takes into account heterogeneous sensors. Theorem 4.1 of [7] can be applied to our case by considering homogeneous sensors, sensor polygons constructed by considering neighbors at a distance less than $R_{tx}/2$ and $d_{max} = R_{tx}/4 - R_s$. We refer the reader to [7] for more details.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 3. Coverage (a), traversed distance (b), number of movements (c), consumed energy (d), termination time (e). Coverage with 140 legitimate sensors (f).

than 60%. Increasing the number of legitimate sensors does not improve the coverage because according to VDA, only sensors detecting a coverage hole are allowed to move. As a result, by compromising a fixed amount of sensors, the attacker is able to impede the spread of legitimate sensors over the AoI, independently on the number of legitimate sensors deployed.

On the contrary, SecureFV and SecureMM allow legitimate sensors to detect and ignore malicious sensors and cover the AoI as well as FV - Free and MM - Free. In particular, SecureFV and SecureMM neutralize the OM attack as they achieve the same coverage that the network would have achieved in absence of the attack (FV - Free and MM - Free in the figure). FV - Free achieves a lower coverage with respect to MM - Free due to the different moving strategy in use: the movement according to the farthest vertex may lead to non uniform deployment and to a lower coverage with respect to MiniMax, as pointed out in [6]. As a consequence, also SecureFV achieves a lower coverage with respect to SecureMM.

Notice that, since under FV and MM sensors are not able to spread over the AoI, these algorithms achieve lower values of performance metrics such as traversed distance, energy consumption and termination time with respect to the other algorithms. For this reason, in the following we do not discuss such results although we show them in the figures.

Figure 3(b) shows the average distance traversed by sensors. As the figure points out, SecureFV and SecureMM introduce a very small overhead in terms of traversed distance with respect to FV - Free and MM - Free. All algorithms show a peak in the traversed distance. This happens because, when few sensors are available, all sensors move in order to contribute to the achievement of the final coverage, resulting in an increase in the traversed distance as the number of available sensors increases. When more sensors are available, most do not move since, according to VDA, only sensors detecting a coverage hole are allowed to move. As a result, the average traversed distance decreases.

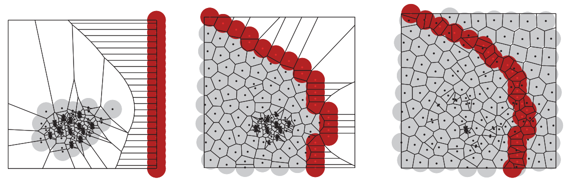
Figure 3(c) shows the average number of moving actions. This is an important metric to evaluate mobile sensor deployment algorithms, since a sensor consumes an high amount of energy to start and stop a movement. Similar considerations with respect to the traversed distance and the peaks in the graphs discussed above can be done. SecureFV and SecureMM introduce a small overhead in terms of number of movements with respect to FV - free and MM - free. Such an overhead is due to the reduced traversed distance per round under SecureVOR which results in an higher number of movements to traverse the same distance.

We now show results related to sensor energy consumption. We adopt the energy cost model commonly used in the literature for mobile sensors [2], [6], [14]. In particular, receiving a message costs 1 energy units (eu), sending a message 1.125eu, traversing one meter costs 300eu and starting/stopping a movement costs as one meter of movement. We consider a cumulative energy consumption metric which takes into account all the above contributions.

Figure 3(d) shows the obtained results. SecureFV and SecureMM show an higher energy consumption with respect to FV - Free and MM - Free. All algorithms incur in an higher communication cost as the sensor density increases. Nevertheless, such an overhead is higher under SecureVOR because of the additional messages required to communicate the trusted neighbor set. Nevertheless, FV - Free and MM - Free on average only consume 31.5% and 43.2% less energy with respect to SecureFV and SecureMM, respectively.

The termination time is shown in 3(e). MM and SecureMM require less time to terminate as the number of available sensors exceed the minimum required for full coverage. Differently, FV and SecureFV are not able to achieve full coverage, thus the termination time does not decrease. Notice that, SecureFV and SecureMM show a shorter termination time with respect to FV - Free and MM - Free. This is due to the shorter maximum traversed distance of SecureVOR which allows shorter movements that are forbidden by VDA. As a result, under VDA sensors move only when a long movement is possible, thus resulting in cascade movements which lengthen the termination time. On the contrary, shorter movements enable sensors to move more in parallel, resulting in a lower termination time for SecureVOR.

In order to further study the performance of the considered algorithms, we performed some experiments by setting the number of legitimate sensors to 140 and by increasing the



(a) (b) (c)

Fig. 4. Dynamic barrier: initial deployment (a), final deployment under MM (b) and SecureMM (c).

number of malicious sensors from 0 to 30. Figure 3(f) shows the achieved coverage. These experiments show that legitimate sensors do not cross the barrier even when a small number of malicious sensors is present. SecureMM and SecureFV are not affected by the number of malicious sensors deployed, since legitimate sensors are able to detect malicious sensors and cover the AoI. Notice that, when no malicious sensor is present SecureFV achieves slightly higher coverage with respect to FV. This is due to the shorter maximum traversed distance which allows some movements under SecureFV that are forbidden under FV, resulting in an higher coverage.

We omit results related to other performance metrics for space limitations. We only mention that SecureMM and SecureFV are not affected by the number of malicious sensors deployed, so they achieve constant values for performance metrics such as traversed distance, energy consumption and termination time. FV an MM are not able to cross the barrier as shown in Figure 3(f), as a consequence they achieve lower values of the considered metrics with respect to the secure algorithms.

We also performed some experiments in absence of malicious sensors and increasing the number of the legitimate nodes. We omit the figures due to space limitations and we summarize the results in the following. SecureFV and SecureMM achieve similar coverage than FV and MM. For both secure algorithms, the traversed distance and the number of starting/stopping actions are within 5% of the values of FV and MM. The energy consumption of SecureFV and SecureMM is higher due to the additional communication required, nevertheless FV and MM consume only 35% less energy. Finally, FV and MM require twice the time of their secure counterpart to terminate due to the serialization of movements, as previously discussed.

Dynamic barrier of malicious sensors

In this set of experiments we consider an OM attack in which malicious sensors are initially deployed outside the AoI and dynamically adapt their position at each round in order to reduce the area in which legitimate sensors deploy. In particular malicious sensors initially form a barrier parallel to an AoI side. At each round, a malicious sensor advertises its position and its trusted set, similar to the previous experiments, but then it may decide to move. Malicious movements are performed in order to not disconnect the barrier, perpendicular to it and are of length d_{max}. In order to avoid moving beyond legitimate sensors, a malicious sensor does not move at the current round if it is at a distance less then $2R_s$ from at least one legitimate sensor.

Despite its simplicity, the above OM attack is able to severely reduce the coverage provided by legitimate sensors under VDA. We consider the initial deployment shown in Figure 4(a) with 150 legitimate and 25 malicious sensors. In this experiment we only consider the MM moving strategy since we obtained similar results than FV. Figure 4(b) shows the effect on the final deployment under MM, whereas Figure 4(c) shows that, under SecureMM, legitimate sensors successfully detect malicious movements and ignore the sensors on the barrier and as a result, they are able to cover the AoI.

VII. Conclusions

In this paper we studied for the first time the vulnerabilities of one of the most referenced approach to mobile sensor deployment, the Voronoi-based approach. We show that, by compromising a small number of sensors, it is possible to severely reduce the area covered by legitimate sensors. We propose a secure Voronoi-based deployment algorithm called SecureVOR. We show that under SecureVOR malicious movements are detected and legitimate sensors terminate the deployment in a finite time. We experimentally study the performance of SecureVOR showing that it is able to achieve the monitoring goals of the network even in presence of an attack, at the expense of a small overhead in terms of energy consumption.

References

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

[2] N. Bartolini, T. Calamoneri, E. G. 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.

[3] Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng, "Efficient placement and dispatch of sensors in a wireless sensor network," *IEEE Trans. on Mobile Computing*, vol. 7, no. 2, pp. 262–274, 2008.

[4] 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.

[5] 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.

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

[7] 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.

[8] M. Ma and Y. Yang, "Adaptive triangular deployment algorithm for unattended mobile sensor networks," *IEEE Trans. on Computers*, vol. 56, no. 7, pp. 946–847, 2007.

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

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

[13] 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.

[14] N. Bartolini, G. Bongiovanni, T. La Porta, and S. Silvestri, "On the security vulnerabilities of the virtual force approach to mobile sensor deployment," *IEEE INFOCOM*, 2013.

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

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

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