Security Games for Node Localization through Verifiable Multilateration

Nicola Basilico, Nicola Gatti, Mattia Monga, and Sabrina Sicari

Abstract—Most applications of Wireless Sensor Networks (WSNs) rely on data about the *positions* of sensor nodes, which are not necessarily known beforehand. Several localization approaches have been proposed but most of them omit to consider that WSNs could be deployed in adversarial settings, where hostile nodes under the control of an attacker coexist with faithful ones. Verifiable Multilateration (VM) was proposed to cope with this problem by leveraging on a set of trusted landmark nodes that act as verifiers. Although VM is able to recognize reliable localization measures, it allows for regions of undecided positions that can amount to the 40% of the monitored area. We studied the properties of VM as a non-cooperative two-player game where the first player employs a number of *verifiers* to do VM computations and the second player controls a *malicious node*. The verifiers aim at securely localizing malicious nodes, while malicious nodes strive to masquerade as unknown and to pretend false positions. Thanks to game theory, the potentialities of VM are analyzed with the aim of improving the defender's strategy. We found that the best placement for verifiers is an equilateral triangle with edge equal to the power range R, and maximum deception in the undecided region is approximately $0.27R$. Moreover, we characterized—in terms of the probability of choosing an unknown node to examine further—the strategies of the players.

Index Terms—Game theory, localization games, security, wireless sensor networks

1 Introduction

Node localization plays a crucial role in most wireless sensor network (WSN) applications [1], [2] such as environment monitoring and vehicle tracking. Location can also be used to improve routing and saving power and to develop applications where services are location dependent. However, the installation of GPS receivers is often unfeasible for its costs, while the positions of sensor nodes is not necessarily known beforehand. In fact, nodes are often deployed randomly or they even move, and one of the challenges is computing localization at time of operations. Thus, several localization schemes have been proposed (*e.g.*, [3], [4], [5], [6], [7], [8], [9], [10], [11]), but most of the current approaches omit to consider that WSNs could be deployed in adversarial settings, where hostile nodes under the control of an attacker coexist with faithful ones. Wireless communications are easy to tamper and nodes are prone to physical attacks and cloning: thus classical solutions, based on access control and strong authentication, are difficult to deploy due to limited power resources of nodes.

A method, which allows to evaluate node reputation and the related accuracy of the monitored data, is required. In this direction, a well defined approach to localize nodes even when some of them are compromised was proposed in [12] and it is known as *Verifiable Multilateration* (VM). VM computes an unknown position by leveraging on a set of trusted landmark nodes that act as *verifiers*. Although VM is able to recognize reliable localization measures (known as *robust* computations) and sure malicious behaviors, it allows for undecided positions (*unknown nodes*), *i.e.*, cases in which localization data are not enough to mark a node as *robust* or *malicious*. In the undecided region, a malicious node can pretend to be in a position that is different from its true one but still compatible with all verifiers' information. In many cases, a large portion of the monitored field is in the undecided region (even more than 40%, as we show in Section 3) and therefore, if this region is not considered, a large area is wasted. Trivially the number of deployed verifiers can be incremented, but this requires higher costs. In this paper, we resort to *non-cooperative game theory* [13] to deal with the problem of secure localization where a set of verifiers and a number of independent malicious nodes are present. The assumption of independence between malicious nodes will allow us to adopt a two-player game, where the first player (defender) employs a number of *verifiers* to do VM computations and the second player (attacker) controls a single *malicious node*. The defender acts to securely localize the malicious node, while the attacker tries to masquerade it as unknown since, when recognized as malicious, its influence would be ruled out by VM. Scenarios where attackers can form coalitions and collude are not considered here, the reason being the unsuitability of VM for such cases, as we will

- *N. Basilico is with the School of Engineering, University of California, Merced, USA.*
  E-mail: nbasilico@ucmerced.edu
- *N. Gatti is with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy.*
  E-mail: ngatti@elet.polimi.it
- *M. Monga is with the Department of Computer Science, Università degli Studi di Milano, Milan, Italy.*
  E-mail: mattia.monga@unimi.it
- *S. Sicari is with the Dipartimento di Scienze Teoriche e Applicate, Università degli Studi dell'Insubria, Varese, Italy.*
  E-mail: sabrina.sicari@uninsubria.it

discuss more in detail later.

This paper provides two main original contributions. First, we study how the verifiers should be placed in order to minimize the maximum error the attacker might induce if the defender accepted also unknown positions and which is the best pair of positions (where actual and fake positions differ the most) for the malicious node. We accomplish such a task by modeling the situation as an extensive-form game between the verifier player and the malicious node player. Furthermore, we find the minimum number of verifiers needed for assuring a given upper bound over the error the attacker might induce.[1]

Second, we introduce a probabilistic approach according which each node is associated with a probability to be malicious. This can be useful to determine the reputation of the nodes. In many situations, such as when the non-malicious nodes are distributed uniformly over the monitored area, the probability that a non-malicious node appears in the best position a malicious node can pretend to is zero, and therefore the verifiers, once observed the positions of all the nodes, can recognize the malicious node with a probability of one. As a result, the malicious node could be interested in varying its positioning strategy in the attempt to masquerade itself as a non-malicious one, reducing thus the maximum induced error. We model this situation as an extensive-form game with uncertainty and we provide an algorithm to find the best strategies of the two players.

The paper is organized as follow. Section 2 shortly describes VM. Section 3 introduces the secure localization games. Section 4 faces the problem of optimal verifiers placement. Section 5 studies the distribution of the malicious node. Section 6 provides an overview of the related works and Section 7 concludes the paper.

2 Verifiable Multilateration

Multilateration is the main technique used in WSNs to estimate the coordinates of unknown nodes given the positions of some given landmark nodes—known as *anchor* nodes—whose positions are known. The position of an unknown node U is computed by geometric inference based on the distances between the anchor nodes and the node itself. However, the distance is not measured directly; instead, it is derived by knowing the speed of the transmission signal, and by measuring the time needed to get an answer to a beacon message sent to U.

Unfortunately, if this computation were carried on without any precaution, U might fool the anchors by delaying the beacon message. However, since in most settings a malicious node can delay the answer beacon, but not speed it up, under some conditions it is possible to spot malicious behaviors. VM uses three or more anchor nodes to detect misbehaving nodes. In VM the anchor nodes work as *verifiers* of the localization data and they send to the sink node B the information needed to evaluate the consistency of the coordinates computed for U. The basic idea of VM is: each verifier V_i computes its *distance bound* db_i [16] to U; any point $P \neq U$ inside the triangle formed by $V_1V_2V_3$ has necessarily at least one distance to the V_i enlarged. This enlargement, however, cannot be masked by U by sending a faster message to the corresponding verifier.

Under the hypothesis that verifiers are trusted and they can securely communicate with B, the following verification process can be used to check the localization data in a setting in which signals cannot be accelerated:

Step 1. Each verifier V_i sends a beacon message to U and records the time τ_i needed to get an answer;

Step 2. Each verifier V_i (whose coordinates $\langle x_i, y_i \rangle$ are known) sends to B a message with its τ_i;

Step 3. From τ_i, B derives the corresponding distance bound db_i (that can be easily computed if the speed of the signal is known) and it estimates U's coordinates by minimizing the sum of squared errors

$$\epsilon = \sum_i (db_i - \sqrt{(x_U - x_i)^2 + (y_U - y_i)^2})^2 \tag{1}$$

where $\langle x_U, y_U \rangle$ are the coordinates to be estimated[2];

Step 4. B can now check if $\langle x_U, y_U \rangle$ are feasible in the given setting by two incremental tests: (a) δ *test:* For all verifiers V_i, compute the distance between the estimated U and V_i: if it differs from the measured distance bound by more than the expected distance measurement error, the estimation is affected by malicious tampering; (b) *point-in-the-triangle test:* Distance bounds are reliable only if the estimated U is within at least one verification triangle formed by a triplet of verifiers, otherwise the estimation is considered unverified.

If both the δ and the *point-in-the-triangle* tests are positive, the distance bounds are consistent with the estimated node position, which moreover falls in at least one verification triangle. This implies that none of the distance bounds were enlarged. Thus, the sink can consider the estimated position of the node as ROBUST; if instead one of the tests fails, the information at hands is not sufficient to support the reliability of the data. An estimation that does not pass the δ test is considered MALICIOUS. In all the other cases, the sink marks the estimation as UNKNOWN. In an ideal situation where there are no measurement errors, there are neither malevolent nodes marked as ROBUST, nor benevolent ones marked as MALICIOUS. Even in this ideal setting, however, there are UNKNOWN nodes, that could be malevolent or not. In other words, there are no sufficient information for evaluating the trustworthiness of a node position. In fact, U could pretend, by an opportune manipulation of

[1] Preliminary versions of these results are published in [14] and [15].

[2] In an ideal situation where there are no measurement errors and/or malicious delays, this is equivalent to finding the (unique) intersection of the circles defined by the distance bounds and centered in the V_i and $\epsilon = 0$. In general, the above computation in presence of errors is not trivial; we point an interested reader to [17].

delays, to be in a position P that is credible to be taken into account. No such points exist inside the triangles formed by the verifiers (this is exactly the idea behind VM), but outside them some regions are still compatible with all the information verifiers have.

Consider $|V|$ verifiers that are able to send signals in a range R. Let x_U and y_U be the *real* coordinates of U. They are unknown to the verifiers, but nevertheless they put a constraint on plausible fake positions, since the forged distance bound to V_i must be greater than the length of $\overline{UV_i}$. Thus, any point $P = \langle x_i, y_i \rangle$ that is a plausible falsification of U has to agree to the following constraints, for each $1 \leq i \leq |V|$:

$$\begin{cases} (y - y_i)^2 + (x - x_i)^2 < R^2 \\ (y - y_i)^2 + (x - x_i)^2 > (y_U - y_i)^2 + (x_U - x_i)^2 \end{cases} \tag{2}$$

The constraints in (2) can be understood better by looking at Fig. 1, where three verifiers are depicted: the green area around each verifier denotes its power range, and the red area is the bound on the distance that U can put forward credibly. Thus, any plausible P must lay outside every red region and inside every green one (and, of course, outside every triangle of verifiers).

VM is suitable to be used only when the attacker can only *enlarge* distances: this requires the absence of collusion between malicious nodes, since coalitions can undertake joint strategies for speeding up of messages, thus violating the assumptions used to classify nodes as malicious via distance bounds and geometric inference [12]. In this work, we consider settings where malicious nodes are independent. Dealing with collusion would require to adopt VM refinements or to develop alternative methods (see [18] for further discussion).

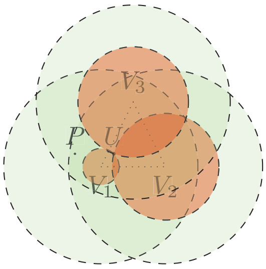


Fig. 1. Plausible falsification region: P is a plausible fake position for U since it lays outside every red region and inside every green one whose radius is R (moreover it is outside the triangle of verifiers).

3 Secure localization games

Our approach models the interaction between independent malicious nodes and verifiers as a non-cooperative game. For the sake of presentation, we restrict our attention to a game played between a group of verifiers and a single malicious node. Handling multiple independent nodes would call for simple extensions and scalings of the model we present here. We will provide some insights along this direction in the following sections. In the game we consider, the malicious node acts to masquerade itself as an unknown node while the verifiers try to face the malicious node at best.

A *game* is described by a tuple ⟨mechanism, strategies⟩: *mechanism* defines the rules of the game in terms of number of players, actions available to the players, outcomes of actions interplay, utility functions of players; *strategies* describe the behaviors of the players during the game in terms of played actions. Strategies can be *pure*, when a player performs one action with a probability of one, or they can be *mixed*, when a player randomizes over a set of actions. The players' strategies define an outcome (if the strategies are pure) or a lottery over the outcomes (if mixed). Players have preferences over the outcomes expressed by utility functions and each player is *rational*. Solving a game means to find a profile of strategies (*i.e.,* a set specifying one strategy for each player) such that the players' strategies are somehow in equilibrium. The most known equilibrium concept is *Nash* [13] where each player cannot improve its utility by unilaterally deviating.

We now formally state our secure localization game by focusing on a setting with $|V| = 3$ verifiers (the minimum number needed to apply VM) and one malicious node. A *secure localization game* is a tuple $\langle \mathcal{Q}, \mathcal{A}, \mathcal{U} \rangle$. Set $\mathcal{Q}$ contains the players: a defender $\mathbf{v}$ who controls the verifiers and a malicious player $\mathbf{m}$ who controls the unknown nodes, thus $\mathcal{Q} = \{\mathbf{v}, \mathbf{m}\}$. Set $\mathcal{A}$ contains the actions available to players. Given a surface $S \subseteq \mathbb{R}^2$, the actions available to $\mathbf{v}$ are all the possible tuples of positions $\langle V_1, V_2, V_3 \rangle$ of the three verifiers with $V_1, V_2, V_3 \in S$, while the actions available to $\mathbf{m}$ are all the possible pairs of positions $\langle U, P \rangle$ with $U, P \in S$ (where U and P are defined in the previous section). We denote by $\sigma_{\mathbf{v}}$ the strategy (possibly mixed) of $\mathbf{v}$ and by $\sigma_{\mathbf{m}}$ the strategy (possibly mixed) of $\mathbf{m}$. Given a strategy profile $\sigma = (\sigma_{\mathbf{v}}, \sigma_{\mathbf{m}})$ in pure strategies, it is possible to check whether or not constraints (2) are satisfied. The outcomes of the game can be $\{\textsc{malicious}, \textsc{robust}, \textsc{unknown}\}$. Set $\mathcal{U}$ contains the players' utility functions, denoted $u_{\mathbf{v}}(\cdot)$ and $u_{\mathbf{m}}(\cdot)$ respectively, that define their preferences over the outcomes. We define (the verifier is indifferent among the outcomes $\{\textsc{malicious}, \textsc{robust}\}$ since both rule out any malicious influence) $u_i(\textsc{malicious}) = u_i(\textsc{robust}) = 0$ for $i \in \mathcal{Q}$, while $u_i(\textsc{unknown})$ can be defined differently according to different criteria. A simple criterion could be to assign $u_{\mathbf{v}}(\textsc{unknown}) = -1$ and $u_{\mathbf{m}}(\textsc{unknown}) = 1$. However, our intuition is that the UNKNOWN outcomes are not the same for the players, because $\mathbf{m}$ could prefer those in which the distance between U and P is maximum. The criterion we adopt in this paper to characterize UNKNOWN outcomes is *maximum deception* where $u_{\mathbf{m}}$ is defined as the distance between U and P, while $u_{\mathbf{v}}$ is defined as the opposite. Other interesting criteria non-explored in this paper are *deception area* where $u_{\mathbf{m}}$ is defined as the size of the region $S' \subseteq S$ such that $P \in S'$ is marked as UNKNOWN, while $u_{\mathbf{v}}$ is defined as the opposite. Another one is the *deception shape* where $u_{\mathbf{m}}$ is defined as the number of disconnected regions $S' \subseteq S$ such that $P \in S'$ is marked as UNKNOWN, while $u_{\mathbf{v}}$ is defined as the opposite. Players could even use different criteria, *e.g.*, $\mathbf{v}$ and $\mathbf{m}$ could adopt the maximum deception criterion and the deception shape respectively. However, when players adopt the same criterion, the game is *zero-sum*, the sum of the players' utilities being zero, and its resolution is tractable.

4 Placement of Verifiers

In this section, we study the optimal placement of the verifiers in order to minimize the maximum deception. The results presented here apply to any node individually trying to fake its position. Therefore, they do not depend on the number of malicious nodes in the environment.

4.1 Maxmin Solution with Three Verifiers

We focus on the case with three verifiers. In our analysis of the game, we consider only the case in which

$$\overline{V_iV_j} \leq R \qquad 1 \leq i, j \leq 3, i \neq j \tag{3}$$

Indeed, if we allow $\overline{V_iV_j} > R$, then there will be several unreasonable equilibria. For instance, an optimal verifiers' strategy would prescribe that the verifiers must be positioned such that only one point satisfies constraints (2). This strategy would assure the verifiers the largest utility (*i.e.*, zero), no UNKNOWN positions being possible. However, this setting is not interesting, since the total area monitored by the verifiers collapses in one point.

At first, we can show that for each action of the verifiers—under the assumption (3)—there exists an action of the malicious node such that this is marked as UNKNOWN. Therefore, there is no verifiers' strategy such that, for all the malicious node's actions, the malicious node is marked as ROBUST or MALICIOUS.

Theorem 1. *For each tuple $\langle V_1, V_2, V_3 \rangle$ such that $\overline{V_iV_j} \leq R$ for all i, j, there exists at least a pair $\langle U, P \rangle$ such that $u_{\mathbf{m}} > 0$.*

Proof: Given V_1, V_2, V_3 such that $\overline{V_iV_j} \leq R$ for all i, j, choose a V_i and call X the point on the line $\overline{V_kV_j}$ $(k, j \neq i)$ closest to V_i. Assign $U = X$. Consider the line connecting V_i to X, assign P to be any point X' on this line such that $\overline{V_iX} \leq \overline{V_iX'} \leq R$. Then, by construction $u_{\mathbf{m}} > 0$. □

We discuss what is the configuration of the three verifiers such that the maximum deception is minimized.

Theorem 2. *Any tuple $\langle V_1, V_2, V_3 \rangle$ such that $\overline{V_iV_j} = R$ for all i, j minimizes the maximum deception.*

Proof: Since we need to minimize the maximum distance between two points, by symmetry, the triangle whose vertexes are V_1, V_2, V_3 must have all the edges with the same length. We show that $\overline{V_iV_j} = R$. It can be easily seen, by geometric construction, that U must be necessarily inside the triangle. As shown in Section 2, P must be necessarily outside the triangle and, by definition, the optimal P will be on the boundary constituted by some circle with center at some V_i and range equal to R (otherwise P could be moved farther and P would not be optimal). As $\overline{V_iV_j}$ decreases, the size of the triangle reduces, while the boundary is unchanged, and therefore $\overline{UP}$ does not decrease. □

Theorem 3. *Let W be the orthocenter of the triangle $V_1V_2V_3$. The malicious node's best responses have polar coordinates*[3] *$U = (\rho = \frac{2\sqrt{3}}{3} - 1, \theta = \frac{3\pi}{2})$ and $P = (\rho = (1 - \frac{\sqrt{3}}{3})R, \theta = \frac{3\pi}{2})$ w.r.t. pole W ($\rho_{V_i} = 0$), for $1 \leq i \leq 3$. The best maximum deception is $\overline{UP} = (2 - \sqrt{3})$.*

Proof: Consider point P in Fig. 2. Any unknown U who pretended to be in P must be located inside the blue circles with centers in the V_i and radii $\overline{V_iP}$, otherwise the fake distance $\overline{V_iP}$ would be greater than the real one $\overline{V_iU}$. Thus, given P, the farthest U is the other intersection of two circles with centers on the endpoints V_i, V_j of the triangle edge at the same distance of P from the edge. The distance $\overline{UP}$ is maximal when the angle w.r.t. pole W ($\rho_{V_i} = 0$) is $\frac{3\pi}{2}$ and P lays on the edge of the power range of $V_k, k \neq i, k \neq j$. In this case $\overline{V_kP} = R$ and $\overline{UP} = 2(R - \frac{\sqrt{3}}{2}R) = (2 - \sqrt{3})R = 0.2679R$. □

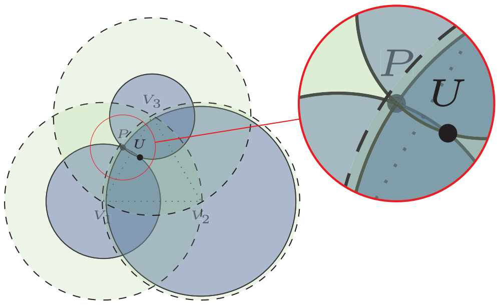


Fig. 2. The farthest U pretending to be in position P.

The value of $u_{\mathbf{m}}$ (*i.e.*, the maximum deception) is then $0.2679R$. In other words, when the verifiers compose an equilateral triangle, a malicious node can masquerade as unknown and the maximum deception is about 27% of the verifiers' range R. Interestingly, with this configuration, the area monitored by all the landmark nodes is the area of the circular triangle [19] $\frac{\sqrt{3}}{4}R^2 + \frac{3}{2}(\frac{\pi}{3} - \sin(\frac{\pi}{3}))R^2$, thus the portion of the region in which VM is effective (*i.e.*, the area of the triangle $V_1V_2V_3$) is $\frac{\frac{\sqrt{3}R^2}{4}}{(\pi - \sqrt{3})\frac{R^2}{2}} \approx 61\%$.

4.2 Maximum Deception with Multiple Verifiers

The results exposed in Section 4.1 are the basis to study situations with multiple (more than three) verifiers. Our main result is the derivation of a bound between the maximum deception and the number of verifiers.

[3] U and P can be expressed more easily with polar coordinates with origin in W, $\rho_V = 0$.

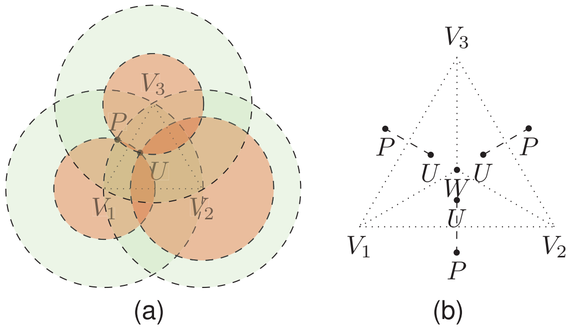
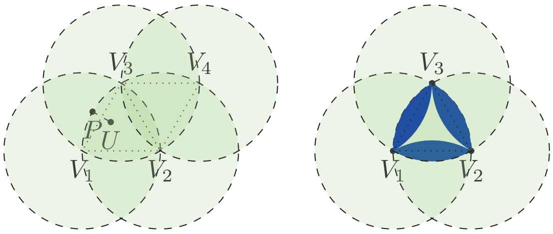


Fig. 3. Malicious node's best responses (maximum deception is $\overline{UP} = 0.2679R$).



(a) Max deception with 4 verifiers is still $26.7\%R$

(b) U positions that allows for plausible Ps

Fig. 4. Impact of verifiers on U ability to fake positions.

Initially, consider the simple situation in which we have four verifiers and they constitute two adjacent equilateral triangles as shown in Fig. 4(a). The maximum deception does not change w.r.t. the case with three verifiers, since some of the best responses depicted in Fig. 3(b) are still available. Indeed, the fourth verifier is useful to rule out only the two positions that are on the edge V_4 faces: on this side any fake P would surely be marked as MALICIOUS (or even ROBUST if $P \equiv U$) since it would be inside the triangle $V_2V_3V_4$. The proof is straightforward. Consider (w.l.o.g.) the triangle $V_1V_2V_3$ in Fig. 4(a). In order for a node not to be marked as MALICIOUS, U must be in the areas depicted in Fig. 4(b). Moreover, any plausible P cannot be neither inside the triangle $V_1V_2V_3$ nor inside the triangle $V_2V_3V_4$, otherwise the node would be marked as MALICIOUS. Indeed, any plausible fake P, given a U in the blue area between V_2 and V_3 (see Fig. 4(b)), cannot be in regions that are outside both the triangles $V_1V_2V_3$ and $V_2V_3V_4$.

The above observation can be leveraged to give a bound over the maximum deception with a given number of verifiers opportunely placed and tuned such that the shape of the area they monitor is a triangle.

Theorem 4. *Given a triangular area, in order to have a maximum deception not larger than $\frac{0.2679R}{2^k}$ we need at least $2 + \sum_{j=0}^{k} 3^j$ verifiers.*

Proof: Consider the basic case with three verifiers (composing an equilateral triangle) with range R and $\overline{V_iV_j} = R$. As shown in Section 4.1 the maximum deception is $0.2679R$. Let us add now three verifiers such that we have four equilateral triangles with edge $\frac{R}{2}$ as shown in Fig. 5. The range of all the verifiers is set equal to $\frac{R}{2}$ (*i.e.*, they could just ignore any beacon message that takes longer than needed to cover the distance $\frac{R}{2}$). Since the edge of the small triangles is now $\frac{R}{2}$, the maximum deception here is $\frac{0.2679R}{2}$ and no U positions are possible in the central triangle $V_4V_5V_6$. Indeed, all the edges of the central triangle are adjacent to the edge of other triangles. This last result allows us not to consider the central triangle when we want to reduce the maximal deception, the malicious node never positioning itself within it. The basic idea is that if we want to halve the maximum deception we need to decompose all the triangles vulnerable to the malicious node by introducing three verifiers. By introducing three new verifiers per triangle we obtain four sub-triangles with an edge that is half of the original triangle and therefore the maximum deception is halved. In general, in order to have a maximum deception of $\frac{0.2679R}{2^k}$, the number of required verifiers[4] is $\frac{3}{2}(1 + 3^k)$, as shown in Table 6(b). In Fig. 6(a), we report an example with $k = 2$ and 15 verifiers. Notice that, when we introduce new verifiers, we need to halve the range. In general, we will have verifiers with multiple different ranges. □

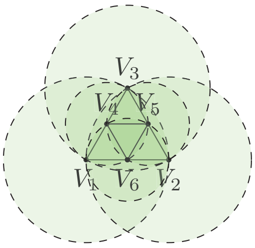
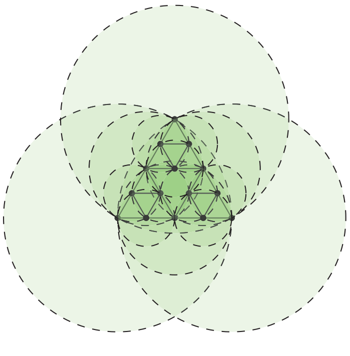


Fig. 5. Max deception with 6 verifiers is $\overline{UP} = \frac{0.2679R}{2}$.

(a) 15 verifiers ($k = 2$) give a max deception $0.06698R$



(b) max deception

k	$\lvert V\rvert$	*max. deception*
0	3	$0.2679R$
1	6	$0.134R$
2	15	$0.06698R$
3	42	$0.03349R$
4	123	$0.01674R$
5	366	$8.372 \cdot 10^{-3}R$

Fig. 6. Max deception is reduced by adding verifiers.

The number of verifiers increases according to the formula $|V|_k = |V|_{k-1} + 3^k$. Asymptotically $\lim_{k\to\infty} \frac{|V|_{k+1}}{|V|_k} = 3$, thus we need to multiply by three the number of verifiers to divide by two the maximum deception. Increasing the number of verifiers requires

[4] The number of vertices in Sierpinski triangle of order k; see [20].

to add new verifiers with a smaller range w.r.t. those already present in the network.

5 Chasing maliciousness

We now consider the case in which we have three deployed verifiers that monitor a given area S and a number n of unknown nodes, among which one is malicious. According to Theorem 3, the malicious node should pretend to be in one of three possible positions. However, if non-malicious nodes can appear in every position of the monitored area with a given probability distribution, excluded for degenerate probability distributions, the probability with which a non-malicious node will appear in the positions that the malicious node must pretend according to Theorem 3 is zero. Therefore, the verifiers, once the positions of all the nodes have been observed, can mark the node in the position prescribed by Theorem 3 as malicious. As a result, the malicious node could be interested in changing its strategy, randomizing over a number of different positions, to masquerade its position as the position of a non-malicious node. To address this problem, we studied a new security game in which *at most one* of the n nodes is malicious and $\mathbf{v}$ might physically inspect one of the unknown nodes to spot if it is malicious. However, we assume that the defender has limited resources and *only one node can be controlled directly,* thus $\mathbf{v}$ is interested in a proper estimation of the probability of maliciousness of the unknown node.

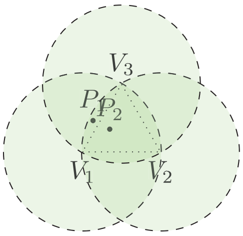


Fig. 7. Two unknown nodes monitored by three verifiers.

Let's analyze the scenario depicted in Fig. 7: we have three verifiers that, according to Theorem 2, were deployed on an equilateral triangle of edge R. They sense the beacons of unknown nodes U_i (not present in the figure) and compute their positions P_i ($1 \leq i \leq n$). At most one U_i is malicious: only if it is, the computed position is fake. The questions are: given a number of nodes, with which probability each node is malicious? Morevoer, what is the strategy of the malicious node that masquerades at best its position?

5.1 Game model

We divide the area S in a finite number $|S|$ of subregions each identified with its centroid $\{\mathbf{y}_1, \dots, \mathbf{y}_{|S|}\}$. This discretization process is necessary to make the game model solvable in practice (a generic game with continuous actions cannot be solved exactly and approximate solutions can be found only with very small problems [13]). We describe the game by referring to its *extensive form*, *i.e.,* players act by alternating their moves, thus the game can be represented by a game tree in which the vertices are decision points of the players and the edges are the actions available to a player at a given decision point. We also introduce a non-strategic player, known as *nature* in the game theoretic jargon, here denoted by $\mathcal{N}$: it plays only the first move and gains no utility in the game. The game is defined by the tuple $\langle \mathcal{Q}, \mathcal{A}, \mathcal{U} \rangle$, where $\mathcal{Q} = \{\mathbf{v}, \mathbf{m}, \mathcal{N}\}$ are the players, $\mathcal{A} = \{\mathcal{A}_i\}_{i \in \mathcal{Q}}$ the actions, and $\mathcal{U}$ the utility functions for players $\mathbf{v}$ and $\mathbf{m}$.

The mechanism defines a game tree $\mathcal{G}$, whose non-terminal vertices are in $\mathcal{V}$ and the terminal ones in $\mathcal{T}$. Function $\iota : \mathcal{V} \to \mathcal{Q}$ returns the player that plays at a given vertex, while function $\chi : \mathcal{V} \to \mathcal{A}$ returns the set of actions available at vertex i to player $\iota(i)$. The game tree is defined by the function $\nu : \mathcal{V} \times \{\mathcal{A}_i\}_{i \in \mathcal{Q}} \to \mathcal{V} \cup \mathcal{T}$ that returns the next vertex given the current vertex and one of the possible actions according to χ and ι. Vertices are grouped in *information sets*: two vertices $i, j \in \mathcal{V}$ are in the same information set $\mathcal{H}$ if $\chi(i) = \chi(j) \wedge \iota(i) = \iota(j)$ and the player knows that it is playing in i or j, but it cannot recognize the specific vertex in which it is playing.

For simplicity, we assume that there is always one malicious node. The structure of the game is as follows.

Nature. In the root vertex, according to function ι, the nature $\mathcal{N}$ plays first; according to χ the available actions $\mathcal{A}_{\mathcal{N}}$ are all the possible assignments $\mathbf{y}_i$ of the $n-1$ (non-malicious) nodes to the sub-regions of S. Players can only observe which regions are occupied by nodes (without distinguishing the nodes), thus we can safely restrict ourselves to $\binom{|S|+(n-1)-1}{n-1}$ actions, the number of combinations (with repetitions) of $n-1$ elements out of $|S|$. $\mathcal{N}$ chooses its action according to a probability distribution Ω over $\mathcal{A}_{\mathcal{N}}$.

Malicious node. According to function ν, each action of $\mathcal{N}$ leads to a different vertex in which, according to function ι, the malicious player $\mathbf{m}$ plays one of the actions determined by χ. We assume that $\mathbf{m}$ cannot observe the nature's moves and therefore all the decision vertices in which the malicious player acts constitute a unique information set $\mathcal{H}_{\mathbf{m}}$. The available actions $\mathcal{A}_{\mathbf{m}}$ are all the possible assignments of the malicious node to a position $\mathbf{x} \in S$ and of its fake position to $\hat{\mathbf{y}} \in S$. As discussed in detail below, we can safely consider just the fake positions $\hat{\mathbf{y}}$ and therefore $|\mathcal{A}_{\mathbf{m}}| = |S|$. We denote by $\sigma_{\mathbf{m}}(\hat{\mathbf{y}})$ the strategy of $\mathbf{m}$, expressing the probability with which $\mathbf{m}$ places the malicious node such that the fake position is $\hat{\mathbf{y}}$.

Verifiers. According to function ν, each action of $\mathbf{m}$ leads to a different vertex in which, according to function ι, the verifier player $\mathbf{v}$ acts one of the actions determined by χ. This player observes the positions of all the nodes without knowing which are malicious or non-malicious, thus equivalent cases ($\mathbf{v}$ distinguishes only the assignments to regions) are grouped in proper information sets.

The number of information sets is $\binom{|S|+n-1}{n}$[5]. We denote the i-th information set by $\mathcal{H}_{\mathbf{v}.i}$. At each information set, the set of actions $\mathcal{A}_{\mathbf{v}}$ available to the verifier player are the choices of a node (to further inspect) in one of the observed positions. We denote by $\sigma_{\mathbf{v}}(\mathcal{H}_{\mathbf{v}.i}, \bar{\mathbf{y}})$ the strategy of $\mathbf{v}$ in information set $\mathcal{H}_{\mathbf{v}.i}$ expressing the probability with which it chooses the node in position $\mathbf{y}$, given that $\mathbf{y}$ is an observed position ($\mathbf{v}$ can adopt a different strategy at each information set). The number of actions changes according to the information set since the number of observed positions can vary (recall that two nodes can be in the same position). The largest number of actions at a given information set is n. Since the positions are not distinguishable by the verifier, the actual node to be inspected is determined according to a uniform probability, *i.e.*, any node will be chosen with probability $\frac{\sigma_{\mathbf{v}}(\mathcal{H}_{\mathbf{v}.i},\mathbf{y})}{|\mathcal{H}|}$. Thus, in all the information sets where all the nodes (non-malicious and malicious) are in the same position, the probability with which any specific node will be chosen is $\frac{1}{n}$. In these information sets, the strategy of the verifier is fixed.

The utility functions are defined on terminal nodes $\mathcal{T}$ and depend on the capture of the malicious node.

- *capture*: $u_{\mathbf{m}} = 0$ and $u_{\mathbf{v}} = 1$;
- *non-capture*: $u_{\mathbf{v}} = 0$ and $u_{\mathbf{m}} = \Delta(\hat{\mathbf{y}})$ where $\Delta(\hat{\mathbf{y}})$ is the max deception given by the fake position $\hat{\mathbf{y}}$.

Notice that the new obtained game is now general-sum. Given that each player behaves as a maximizer of the expected utility, for each fake position $\hat{\mathbf{y}}$ the malicious node will choose the position such that the deception is maximum and therefore, as anticipated above, we can safely reduce the set of actions of the malicious player to exclusively the fake positions. When the players adopt mixed strategies, randomizing over their actions, their utility is defined in expectation over the outcomes as prescribed in [21].

To clarify the notation, Fig. 8 depicts the game tree of the following example.

Example 1. *We have $|S| = 3$ with $S = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ and $n = 2$ (one malicious and one non-malicious). The game formalization (to read together with Fig. 8) follows.*

$$\begin{aligned}
\mathcal{Q} \;&\triangleq\; \{\mathcal{N}, \mathbf{m}, \mathbf{v}\}\\
\mathcal{A} \;&\triangleq\; \{\mathcal{A}_{\mathcal{N}}, \mathcal{A}_{\mathbf{m}}, \mathcal{A}_{\mathbf{v}}\}\\
&\text{where } \mathcal{A}_{\mathcal{N}} = \{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}, \mathcal{A}_{\mathbf{m}} = \{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \hat{\mathbf{y}}_3\}, \mathcal{A}_{\mathbf{v}} = \{\bar{\mathbf{y}}_1, \bar{\mathbf{y}}_2, \bar{\mathbf{y}}_3\}\\
\mathcal{V} \;&\triangleq\; \{\bullet_0, \bullet_{0.1}, \bullet_{0.2}, \bullet_{0.3}, \bullet_{0.1.2}, \bullet_{0.1.3}, \bullet_{0.2.1}, \bullet_{0.2.3}, \bullet_{0.3.1}, \bullet_{0.3.2}\}\\
\mathcal{T} \;&\triangleq\; \forall i \neq j : 1 \le i, j \le 3, \{\bullet_{0.i.j.c}, \bullet_{0.i.j.\bar{c}}\}\\
&\quad\; \forall i : 1 \le i \le 3, \{\bullet_{0.i.i}\}\\
\iota \;&\triangleq\; \bullet_0 \to \mathcal{N}\\
&\quad\; \forall i : 1 \le i \le 3, \bullet_{0.i} \to \mathbf{m}\\
&\quad\; \forall i, j : 1 \le i, j \le 3, \bullet_{0.i.j} \to \mathbf{v}\\
\chi \;&\triangleq\; \bullet_0 \to \mathcal{A}_{\mathcal{N}}\\
&\quad\; \forall i : 1 \le i \le 3, \bullet_{0.i} \to \mathcal{A}_{\mathbf{m}}\\
&\quad\; \forall i, j : 1 \le i, j \le 3, \bullet_{0.i.j} \to \{\bar{\mathbf{y}}_i, \bar{\mathbf{y}}_j\}\\
\nu \;&\triangleq\; \forall i : 1 \le i \le 3, (\bullet_0, \mathbf{y}_i) \to \bullet_{0.i}\\
&\quad\; \forall i, j : 1 \le i, j \le 3, (\bullet_{0.i}, \hat{\mathbf{y}}_j) \to \bullet_{0.i.j}\\
&\quad\; \forall i \neq j : 1 \le i, j \le 3, (\bullet_{0.i.j}, \bar{\mathbf{y}}_j) \to \bullet_{0.i.j.c}\\
&\quad\; \forall i \neq j : 1 \le i, j \le 3, (\bullet_{0.i.j}, \bar{\mathbf{y}}_i) \to \bullet_{0.i.j.\bar{c}}\\
&\quad\; \forall i : 1 \le i \le 3, (\bullet_{0.i.i}, \bar{\mathbf{y}}_i) \to \bullet_{0.i.i}\\
\mathcal{H} \;&\triangleq\; \{\mathcal{H}_{\mathbf{m}}, \mathcal{H}_{\mathbf{v}.1}, \mathcal{H}_{\mathbf{v}.2}, \mathcal{H}_{\mathbf{v}.3}\}\\
&\text{where } \mathcal{H}_{\mathbf{m}} = \{\bullet_{0.1}, \bullet_{0.2}, \bullet_{0.3}\},\\
&\quad\; \mathcal{H}_{\mathbf{v}.1} = \{\bullet_{0.1.2}, \bullet_{0.2.1}\}, \mathcal{H}_{\mathbf{v}.2} = \{\bullet_{0.1.3}, \bullet_{0.3.1}\},\\
&\quad\; \mathcal{H}_{\mathbf{v}.3} = \{\bullet_{0.2.3}, \bullet_{0.3.2}\}\\
u \;&\triangleq\; \forall i \neq j : 1 \le i, j \le 3, \bullet_{0.i.j.c} \to (-, 0, 1)\\
&\quad\; \forall i \neq j : 1 \le i, j \le 3, \bullet_{0.i.j.\bar{c}} \to (-, \Delta(\hat{\mathbf{y}}_i), 0)\\
&\quad\; \forall i : 1 \le i \le 3, \bullet_{0.i.i} \to (-, \frac{1}{2}\Delta(\hat{\mathbf{y}}_i), \frac{1}{2})
\end{aligned}$$

Nature $\mathcal{N}$ places $n - 1 = 1$ non-malicious nodes on a position and then $\mathbf{m}$ acts by choosing the only fake position. In Fig. 8, all the decision points of the malicious player are connected by a dashed line because they all constitute a unique information set $\mathcal{H}_{\mathbf{m}}$. In all the situations in which the two nodes (the malicious one and the non-malicious one) are in the same observed position we have a terminal node because the strategy of the verifier player is determined by a $\frac{1}{2}$ probability of choosing the malicious node (and same probability for the non-malicious one). The expected utilities are $u_{\mathbf{m}} = \frac{1}{2}\Delta(\hat{\mathbf{y}}_i), u_{\mathbf{v}} = \frac{1}{2}$. In the other situations, $\mathbf{v}$ has two possible actions. For instance, the verifier cannot distinguish the situation in which the malicious node has chosen $\hat{\mathbf{y}}_1$ and the non-malicious node has been placed by nature in $\mathbf{y}_2$ from the reverse situation. Therefore, these two situations constitute a unique information set ($\mathcal{H}_{\mathbf{v}.1}$). Consider, as an example, $\hat{\mathbf{y}}_1$ and $\mathbf{y}_2$ as the positions of the malicious and non-malicious nodes respectively, if the verifier chooses $\bar{\mathbf{y}}_1$, the malicious node will be captured ($u_{\mathbf{v}} = 1$), instead, if the verifier chooses $\bar{\mathbf{y}}_2$, the malicious node will not be captured ($u_{\mathbf{m}} = \Delta(\hat{\mathbf{y}}_1)$).

5.2 Solution concepts and equilibrium constraints

We consider in this section two solution concepts for our game model. The maxmin strategy and the Nash equilibrium. The maxmin strategy gives the maximum utility against the strongest opponent, therefore, the utility provided by the maxmin strategy is a lower bound. The maxmin value is known also as the *security level* of a player. The computation of a maxmin strategy is easy, requiring polynomial time in the size of the game. The Nash equilibrium is the most appropriate solution concept for general-sum games, but its computation is not easy in the worst case.[6]

5.2.1 Security levels and strategies

Initially, we focus on the maxmin value of the verifier player. For simplicity, we compute it by finding the minmax strategy of the malicious player (*i.e.*, the dual strategy of the verifier player's maxmin strategy). The verifier player's maxmin value is given by the following

[5] For instance, suppose that there are two nodes (one malicious and one non-malicious) and three possible positions $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}$, the number of information sets is six, one for each of the following possible combinations of two nodes $\{\mathbf{y}_1, \mathbf{y}_1\}, \{\mathbf{y}_1, \mathbf{y}_2\}, \{\mathbf{y}_1, \mathbf{y}_3\}, \{\mathbf{y}_2, \mathbf{y}_2\}, \{\mathbf{y}_2, \mathbf{y}_3\}, \{\mathbf{y}_3, \mathbf{y}_3\}$ since $\mathbf{v}$ cannot distinguish between $\{\mathbf{y}_1, \hat{\mathbf{y}}_2\}$ and $\{\hat{\mathbf{y}}_2, \mathbf{y}_1\}$: thus both choices are equivalent to a generic $\{\mathbf{y}_1, \mathbf{y}_2\}$.

[6] When a game is zero-sum, the pair maxmin strategies is certainly also a Nash equilibrium, while there is no guarantee with general-sum games: computing a Nash equilibrium for a general-sum game is PPAD-complete and, although P ⊆ PPAD ⊆ NP and therefore it could be the case PPAD ⊂ NP, it is commonly believed that computing a Nash equilibrium is exponential in the worst case [13].

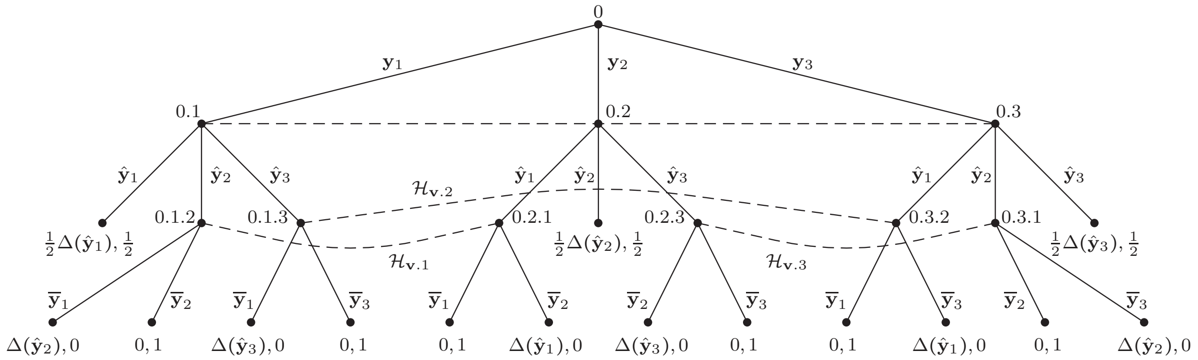

0
$\mathbf{y}_1$
$\mathbf{y}_2$
$\mathbf{y}_3$
0.1
0.2
0.3
$\hat{\mathbf{y}}_1$
$\hat{\mathbf{y}}_2$
$\hat{\mathbf{y}}_3$
$\mathcal{H}_{\mathbf{v}.2}$
$\mathcal{H}_{\mathbf{v}.1}$
$\mathcal{H}_{\mathbf{v}.3}$
0.1.2
0.1.3
0.2.1
0.2.3
0.3.2
0.3.1
$\frac{1}{2}\Delta(\hat{\mathbf{y}}_1), \frac{1}{2}$
$\frac{1}{2}\Delta(\hat{\mathbf{y}}_2), \frac{1}{2}$
$\frac{1}{2}\Delta(\hat{\mathbf{y}}_3), \frac{1}{2}$
$\bar{\mathbf{y}}_1$
$\bar{\mathbf{y}}_2$
$\bar{\mathbf{y}}_3$
$\Delta(\hat{\mathbf{y}}_2), 0$
$0, 1$
$\Delta(\hat{\mathbf{y}}_3), 0$
$0, 1$
$0, 1$
$\Delta(\hat{\mathbf{y}}_1), 0$
$\Delta(\hat{\mathbf{y}}_3), 0$
$0, 1$
$0, 1$
$\Delta(\hat{\mathbf{y}}_1), 0$
$0, 1$
$\Delta(\hat{\mathbf{y}}_2), 0$

Fig. 8. The game tree with $n = 2$ and $|S| = 3$ (see Example 1).

linear mathematical programming problem, where $u^*_{\mathbf{v}}(h)$ denotes the value of the verifier players' optimal action at information set h. This, together with $\sigma_{\mathbf{m}}$, are the variables of the linear programming problem we formulate.

$$\underset{u^*_{\mathbf{v}},\sigma_{\mathbf{m}}}{\text{minimize}} \sum_{h\in\mathcal{H}} u^*_{\mathbf{v}}(h) + \sum_{1\leq i\leq |S|} \frac{1}{2}\Omega(\mathbf{y}_i)\cdot\sigma_{\mathbf{m}}(\hat{\mathbf{y}}_i) \tag{4}$$

$$\text{subject to } u^*_{\mathbf{v}}(h) \geq \sigma_{\mathbf{m}}(\hat{\mathbf{y}}_i)\cdot\Omega(\mathbf{y}_j) \qquad \hat{\mathbf{y}}_i, \mathbf{y}_j \in h, i\neq j \tag{5}$$

$$\sum_{1\leq i\leq |S|} \sigma_{\mathbf{m}}(\hat{\mathbf{y}}_i) = 1 \tag{6}$$

$$\sigma_{\mathbf{m}}(\hat{\mathbf{y}}_i) \geq 0 \qquad 1\leq i\leq |S| \tag{7}$$

The objective function (4) is written for $n = 2$; when n is greater, the second part is more complex: in fact, one has to consider all the cases in which $\mathbf{m}$ plays a fake position already used one or more times by $\mathcal{N}$[7]. Constraints (5) assure that $u^*_{\mathbf{v}}(h)$ is larger than the value given by every action available at h; constraints (6)–(7) assure that strategy $\sigma_{\mathbf{m}}$ is well defined. Interestingly, the malicious player's minmax strategy matches the probability distribution Ω according to which non-malicious nodes are placed, as shown by the following lemma.

Lemma 1. *With one non-malicious node, the malicious player's minmax strategy is $\sigma_{\mathbf{m}}(\hat{\mathbf{y}}_i) = \Omega(\mathbf{y}_i)$ for all $1 \leq i \leq |S|$ and the verifier player's maxmin value is $\frac{1}{2}$.*

Proof: When $\sigma_{\mathbf{m}}(\hat{\mathbf{y}}_i) = \Omega(\mathbf{y}_i)$ for all $1 \leq i \leq |S|$, function (4) is equal to $\frac{1}{2}$ for every possible Ω. Indeed, the second term of the objective function is $\sum_{1\leq i\leq |S|} \frac{1}{2}\Omega(\mathbf{y}_i)\cdot\Omega(\mathbf{y}_i)$, and the first term, by constraint (5), becomes $\sum_{h\in\mathcal{H}} u^*_{\mathbf{v}}(h) = \sum_{1\leq i,j\leq |S|\wedge i\neq j}\Omega(\mathbf{y}_i)\cdot\Omega(\mathbf{y}_j)$. The sum of these two terms is equal to $\frac{1}{2}(\sum_{1\leq i\leq |S|}\Omega(\mathbf{y}_i))^2$ that is, by definition of Ω, equal to $\frac{1}{2}$. It can be easily shown that it is a minimum because, when $\sigma_{\mathbf{m}}(\hat{\mathbf{y}})$ is increased by ξ, reducing for example $\sigma_{\mathbf{m}}(\hat{\mathbf{y}}_j)$, then the objective function increases by $\xi\cdot(1-\frac{1}{2}\Omega(\mathbf{y}_i)-\frac{1}{2}\Omega(\mathbf{y}_j))$ (a strictly positive value). Thus, $\frac{1}{2}$ is the minimum. □

This result shows that the verifier will always choose the malicious node with at least a probability of $\frac{1}{2}$. With more than one non-malicious node, the result is analogous (we omit the proof, being the same of the case with a single non-malicious node).

Lemma 2. *With $n-1$ non-malicious nodes, the malicious player's minmax strategy is $\sigma_{\mathbf{m}}(\hat{\mathbf{y}}_i) = \Omega(\mathbf{y}_i)$ for all $1 \leq i \leq |S|$ and the verifier player's maxmin value is $\frac{1}{n}$.*

Similarly, we can compute the maxmin value for the malicious player by the following linear program:

$$\text{minimize } u^*_{\mathbf{m}} \tag{8}$$

$$\underset{1\leq i\leq |S|}{\text{subject to}}\ u^*_{\mathbf{m}} \geq \Delta(\hat{\mathbf{y}}_i)\cdot\left(\frac{1}{2}\Omega(\mathbf{y}_i) + \sum_{\substack{\{\bar{\mathbf{y}}_i,\bar{\mathbf{y}}_j\}\in\mathcal{H}\\ \bar{\mathbf{y}}_i\neq\bar{\mathbf{y}}_j}}\Omega(\mathbf{y}_j)\sigma_{\mathbf{v}}(h,\bar{\mathbf{y}}_j)\right) \tag{9}$$

$$\sum_{\bar{\mathbf{y}}_k\in h}\sigma_{\mathbf{v}}(h,\bar{\mathbf{y}}_k) = 1 \qquad \forall h\in\mathcal{H} \tag{10}$$

$$\sigma_{\mathbf{v}}(h,\bar{\mathbf{y}}_i) \geq 0 \qquad \bar{\mathbf{y}}_i\in h \tag{11}$$

Constraints (9) force the maxmin value $u^*_{\mathbf{m}}$ to be larger than the expected utility given by every action of the malicious player; constraints (10) and (11) assure that $\sigma_{\mathbf{v}}$ is a well defined strategy. In this case, the verifier player's minmax strategy is not trivial, as instead it is above in the case of the malicious player. Therefore, also the case with more non-malicious nodes is more complicated and requires a different linear formulation of constraints (9) to capture all the possible assignments of non-malicious nodes to positions. In this case, the size of the game tree rises exponentially in the number of non-malicious nodes and therefore, although maxmin strategy can be computed in polynomial time in the size of the game, the maxmin computation is exponential in the number of non-malicious nodes.

5.2.2 Nash equilibria

Under the assumption that each single player knows the payoffs of the opponent and knows the probability distribution associated with which the non-malicious nodes place in the positions, the appropriate solution concept is the Nash equilibrium.

Initially, we study the situation in which there is a single non-malicious node and, subsequently, we discuss how to extend it to the general case.

We can formulate the equilibrium constraints for finding a Nash equilibrium as a mixed-integer linear mathematical programming problem as follows. Call $\mathcal{H}_{\mathbf{v}}$ the

[7] For example, with $n = 3$ the term becomes: $\sum_{1\leq i\leq |S|}\frac{1}{3}\Omega(\mathbf{y}_i\mathbf{y}_i)\sigma(\hat{\mathbf{y}}_i) + \sum_{1\leq i,j\leq |S|\wedge i\neq j}\frac{1}{2}\Omega(\mathbf{y}_i\mathbf{y}_j)\sigma(\hat{\mathbf{y}}_i)$.

set of information sets of the verifier player and h a single information set, defined as the pair of observed positions of two nodes, *i.e.*, $h = \{\bar{\mathbf{y}}_i, \bar{\mathbf{y}}_j\}$; we have:

$$
\begin{aligned}
& u_{\mathbf{m}}(\bar{\mathbf{y}}_i) = \Delta(\mathbf{y}_i)\left(\frac{1}{2}\Omega(\mathbf{y}_i) + \sum_{\substack{\{\bar{\mathbf{y}}_i,\bar{\mathbf{y}}_j\}\in\mathcal{H}\\ \bar{\mathbf{y}}_j\neq\bar{\mathbf{y}}_i}} \Omega(\mathbf{y}_j)\sigma_{\mathbf{v}}(h,\bar{\mathbf{y}}_j)\right) && 1 \leq i \leq |S| \quad (12)\\
& u^*_{\mathbf{m}} \geq u^*_{\mathbf{v}}(\bar{\mathbf{y}}_i) && 1 \leq i \leq |S| \quad (13)\\
& u^*_{\mathbf{m}} \leq u^*_{\mathbf{v}}(\bar{\mathbf{y}}_i) + \Delta_\infty \cdot (1 - s_i) && 1 \leq i \leq |S| \quad (14)\\
& \sum_{1\leq i\leq |S|} s_i \geq 1 && 1 \leq i \leq |S| \quad (15)\\
& s_i \in \{0,1\} && 1 \leq i \leq |S| \quad (16)\\
& \sigma_{\mathbf{v}}(h,\bar{\mathbf{y}}_i) \leq s_i + 1 - s_j && \bar{\mathbf{y}}_i, \bar{\mathbf{y}}_j \in h, i \neq j \quad (17)\\
& \sum_{\bar{\mathbf{y}}_i \in h} \sigma_{\mathbf{v}}(h,\bar{\mathbf{y}}_i) = 1 && \bar{\mathbf{y}}_i \in h, h \in \mathcal{H}_{\mathbf{v}} \quad (18)\\
& \sigma_{\mathbf{v}}(h,\bar{\mathbf{y}}_i) \geq 0 && \bar{\mathbf{y}}_i \in h, h \in \mathcal{H}_{\mathbf{v}} \quad (19)
\end{aligned}
$$

In these formulae s_i are *slack variables* that will be equal to one if the malicious player takes action $\hat{\mathbf{y}}_i$ with positive probability. Parameter Δ_∞ is defined as $\Delta_\infty = \max_{1\leq i\leq|S|} \Delta(\hat{\mathbf{y}}_i)$: thus it plays the role of infinity when on the right side of a '$\leq$' inequality. The size of the above mathematical program is $O(|S|^2)$, both in terms of number of variables and number of constraints.

The above formulation is inspired to the formulation presented in [22] based on a mixed integer-linear problem (MILP), but it is much more compact. In the original formulation, one binary variable per action for each player is needed: in our case $|S| + |S|^2$ binary variables and unfortunately the hardness of a MILP is due to the presence of integer variables.[8] Instead, our formulation uses only $|S|$ binary variables and therefore we dramatically improved the efficiency by $2^{|S|^2}$.

We can state the following theorem.

Theorem 5. *For every solution $(u^*_{\mathbf{m}}, u^*_{\mathbf{v}}, \sigma_{\mathbf{v}}, \mathbf{s})$ of the feasibility problem composed of constraints (12)–(19), we can derive a Nash equilibrium $(\sigma^*_{\mathbf{v}}, \sigma^*_{\mathbf{m}})$ where the verifier player's optimal strategy is defined as $\sigma^*_{\mathbf{v}} = \sigma_{\mathbf{v}}$ and, called*

$$
\tilde{\sigma}_{\mathbf{m}}(\hat{\mathbf{y}}_i) = \begin{cases} 0 & \text{if } s_i = 0 \\ \Omega(\mathbf{y}_i) & \text{otherwise} \end{cases}
$$

the malicious player's optimal strategy is defined as $\sigma^*_{\mathbf{m}}(\hat{\mathbf{y}}_i) = \frac{\tilde{\sigma}_{\mathbf{m}}(\hat{\mathbf{y}}_i)}{\sum_{1\leq i\leq|S|}\tilde{\sigma}_{\mathbf{m}}(\hat{\mathbf{y}}_i)}$.

Proof: Constraints (12) force each $u^*_{\mathbf{v}}(\bar{\mathbf{y}}_i)$ to be equal to the expected utility the malicious player receives from undertaking action $\hat{\mathbf{y}}_i$. Constraints (13) and (14) force $u^*_{\mathbf{m}}$ to be equal to the expected utility of the best actions of the malicious player and force $s_i = 0$ (*i.e.*, they force the malicious player not to play action $\hat{\mathbf{y}}_i$) for each non-optimal action $\hat{\mathbf{y}}_i$ (*i.e.*, those with $u^*_{\mathbf{v}}(\bar{\mathbf{y}}_i) < u^*_{\mathbf{m}}(\hat{\mathbf{y}}_i)$). Constraints (15) assure that the malicious player plays at least one action $\hat{\mathbf{y}}_i$, given that constraints (16) force variables s_i to be binary. Therefore, constraints (12)–(16) constitute the equilibrium constraints for the malicious player, forcing the player to play only its best responses. Thus, every action $\hat{\mathbf{y}}_i$ such that $s_i = 1$ will be played with a positive probability in a Nash equilibrium. Constraints (17) force every strategy $\sigma_{\mathbf{v}}(h, \bar{\mathbf{y}}_i)$ to be zero only if $\bar{\mathbf{y}}_i$ is not played and $\bar{\mathbf{y}}_j$ is played. The basic idea is that, if $\hat{\mathbf{y}}_i$ is not played, then the verifier player will never choose a node in position $\bar{\mathbf{y}}_i$, except for all the cases in which the malicious player will never play both positions $\bar{\mathbf{y}}_i, \bar{\mathbf{y}}_j$ composing an information set. In these last cases (never reachable along the equilibrium path), the strategy of the verifier player is arbitrary. Thus, if only a position $\hat{\mathbf{y}}_i$ of an information set h is played by the malicious player, then the verifier player will play such position with a probability of one and *vice-versa*. Instead, if both positions of an information set are played by the malicious player, then verifier player will randomize over them. The constraints over the malicious player strategies such that verifier player can randomize over the two choices $\bar{\mathbf{y}}_i, \bar{\mathbf{y}}_j$ of an information set is that $\frac{\sigma_{\mathbf{m}}(\hat{\mathbf{y}}_i)}{\Omega(\mathbf{y}_i)} = \frac{\sigma_{\mathbf{m}}(\hat{\mathbf{y}}_i)}{\Omega(\mathbf{y}_j)}$, *i.e.*, it is equiprobable that the malicious player is in both positions. Therefore, since the malicious player strategy can be easily derived from the s_i, it can be omitted from the equilibrium constraint problem. Finally, constraints (18) and (19) force strategies $\sigma_{\mathbf{v}}(h, \bar{\mathbf{y}}_i)$ to be well defined in every information set. □

The above theoretical results show that the strategy of the malicious player is equal to Ω except for a truncation (some positions are not played by the malicious player) and the normalization of the probability to one. Interestingly, some game instances can admit Nash equilibria in pure strategies. We report some examples.

Example 2. *Suppose that $|S| = 3$ and that $\Omega(\mathbf{y}_1) = \Omega(\mathbf{y}_2) = \Omega(\mathbf{y}_3) = \frac{1}{3}$. We report in the following table how the Nash equilibrium strategies change as the values of Δ change in five different settings.*

	# 1	# 2	# 3	# 4	# 5
$\Delta(\hat{\mathbf{y}}_1)$	1.00	1.00	0.00	1.00	2.00
$\Delta(\hat{\mathbf{y}}_2)$	1.00	2.00	2.00	1.00	2.00
$\Delta(\hat{\mathbf{y}}_3)$	1.00	3.00	3.00	4.00	1.00
$\sigma_{\mathbf{m}}(\hat{\mathbf{y}}_1)$	0.33	0.33	0.00	0.00	0.33
$\sigma_{\mathbf{m}}(\hat{\mathbf{y}}_2)$	0.33	0.33	0.50	0.00	0.33
$\sigma_{\mathbf{m}}(\hat{\mathbf{y}}_3)$	0.33	0.33	0.50	1.00	0.33
$\sigma_{\mathbf{v}}(\mathcal{H}_{\mathbf{v}.1}, \bar{\mathbf{y}}_1)$	1.00	0.05	0.00	0.50	0.6
$\sigma_{\mathbf{v}}(\mathcal{H}_{\mathbf{v}.1}, \bar{\mathbf{y}}_2)$	0.00	0.95	1.00	0.50	0.4
$\sigma_{\mathbf{v}}(\mathcal{H}_{\mathbf{v}.2}, \bar{\mathbf{y}}_1)$	0.00	0.00	0.00	0.00	0.75
$\sigma_{\mathbf{v}}(\mathcal{H}_{\mathbf{v}.2}, \bar{\mathbf{y}}_3)$	1.00	1.00	1.00	1.00	0.25
$\sigma_{\mathbf{v}}(\mathcal{H}_{\mathbf{v}.3}, \bar{\mathbf{y}}_2)$	1.00	0.32	0.30	0.00	1.00
$\sigma_{\mathbf{v}}(\mathcal{H}_{\mathbf{v}.3}, \bar{\mathbf{y}}_3)$	0.00	0.68	0.70	1.00	0.00

It can be observed that: when all the positions have the same deception, then (as expected) the strategy of the malicious node is uniform over its action; when a location has a value of zero, the malicious node does not play such an action; when a location has a value remarkably larger than the others' value, the malicious node plays that action with probability of one.

Given that a game can admit multiple Nash equilibria, it can be useful to characterize the range of equilibria. We can achieve this task by exploiting formulation (12)–(19) together with a linear objective function. More pre-

[8] More precisely, MILP techniques create a branch-and-bound tree whose size rises exponentially in the number of binary variables.

cisely, we can study the range of Nash equilibria by finding the equilibrium maximizing the expected utility of the malicious node and the equilibrium minimizing the same objective function. Every other equilibrium will be in the middle. The two extreme equilibria can be found by solving (12)–(19) with the maximization and minimization, respectively, of $u^*_{\mathbf{m}}$. The two resulting mathematical programs are MILP.

With more non-malicious nodes, the mathematical program (12)–(19) is more complicated and depends on n. We provide the program with $n = 3$:

$$\begin{aligned}
u_{\mathbf{m}}(\bar{\mathbf{y}}_i) = \Delta(\mathbf{y}_i) \cdot \Big(\frac{1}{3}\Omega(\mathbf{y}_i)^2 + \sum_{j \neq i} \Omega(\mathbf{y}_j)^2 \cdot \sigma_{\mathbf{v}}(\bar{\mathbf{y}}_i, \bar{\mathbf{y}}_j, \bar{\mathbf{y}}_j)& \\
+2\sum_{j \neq i} \Omega(\mathbf{y}_i) \cdot \Omega(\mathbf{y}_j) \cdot \big(1 - \frac{1}{2}\sigma_{\mathbf{v}}(\bar{\mathbf{y}}_i, \bar{\mathbf{y}}_j, \bar{\mathbf{y}}_i)\big)& \\
+\sum_{j \neq i} \sum_{k \neq j, k \neq i} \Omega(\mathbf{y}_j) \cdot \Omega(\mathbf{y}_k) \cdot \big(1 - \sigma_{\mathbf{v}}(\bar{\mathbf{y}}_k, \bar{\mathbf{y}}_j, \bar{\mathbf{y}}_i)\big)\Big)& \qquad (20) \\
u^*_{\mathbf{m}} \geq u_{\mathbf{m}}(\bar{\mathbf{y}}_i)& \qquad (21) \\
u^*_{\mathbf{m}} \leq u_{\mathbf{m}}(\bar{\mathbf{y}}_i) + \Delta_\infty \cdot (1 - s_i)& \qquad (22) \\
\sum_{1 \leq i \leq |S|} s_i \geq 1& \qquad (23) \\
\sigma_{\mathbf{v}}(\mathbf{y}_i, \mathbf{y}_j, \mathbf{y}_k) = \sigma_{\mathbf{v}}(\mathbf{y}_j, \mathbf{y}_i, \mathbf{y}_k)& \qquad (24) \\
\sigma_{\mathbf{v}}(\mathbf{y}_j, \mathbf{y}_i, \mathbf{y}_i) = \sigma_{\mathbf{v}}(\mathbf{y}_j, \mathbf{y}_j, \mathbf{y}_i)& \qquad (25) \\
\sigma_{\mathbf{v}}(\mathbf{y}_i, \mathbf{y}_j, \mathbf{y}_k) + \sigma_{\mathbf{v}}(\mathbf{y}_i, \mathbf{y}_k, \mathbf{y}_j) + \sigma_{\mathbf{v}}(\mathbf{y}_k, \mathbf{y}_j, \mathbf{y}_i) = 1& \qquad (26) \\
\sigma_{\mathbf{v}}(\mathbf{y}_i, \mathbf{y}_j, \mathbf{y}_i) + \sigma_{\mathbf{v}}(\mathbf{y}_i, \mathbf{y}_j, \mathbf{y}_j) = 1& \qquad (27) \\
\sigma_{\mathbf{v}}(\mathbf{y}_i, \mathbf{y}_j, \mathbf{y}_i) \leq s_i + 1 - s_j& \qquad (28) \\
\sigma_{\mathbf{v}}(\mathbf{y}_i, \mathbf{y}_i, \mathbf{y}_j) \leq s_j + 1 - s_i& \qquad (29) \\
\sigma_{\mathbf{v}}(\mathbf{y}_i, \mathbf{y}_j, \mathbf{y}_k) \leq s_k + 1 - r_{i,j}& \qquad (30) \\
r_{i,j} \geq s_i& \qquad (31) \\
r_{i,j} \geq s_j& \qquad (32) \\
r_{i,j} \leq s_i + s_j& \qquad (33) \\
s_i \in \{0, 1\}& \qquad (34) \\
r_{i,j} \in \{0, 1\}& \qquad (35) \\
1 \leq i, j, k \leq |S| \qquad i \neq j, i \neq k, j \neq k& \qquad (36)
\end{aligned}$$

The above program is similar to (12)–(19). Constraints (20)–(23) are the analogous of (12)–(15). Constraints (24)–(27) enforce the consistency of the verifier's strategies. Constraints (28)–(30) are the analogous of constraints (17). With $n = 3$, an additional auxiliary variable r is necessary. Constraints (31)–(33) enforce $r_{i,j}$ to be equal to the 'or' operator between s_j and s_i.

The game model presented in this section can be easily extended to take into account the presence of multiple malicious independent nodes. Under the assumption that, in such a case, every node would behave in the same way, the results provided in this section keep to hold. Some new features could be required. For example, one could redefine the actions for the verifiers allowing them to inspect multiple position in one single action.

5.3 Empirical evaluations

Differently from the game studied in Section 4, a closed-form solution of the game presented in this section cannot be provided. Hence, we provide here an experimental evaluation of our model.

5.3.1 Simulation setting

We consider the following scenario composed of:

- three anchors acting as verifiers on an equilateral triangle;
- from 1 to 4 non-malicious nodes with uniform probability to appear in the monitored area;
- one malicious node;
- a number $|S|$ of discretized subregions of S.

For reasons of symmetry, we can work directly on the space of the deception Δ instead of the positions $\mathbf{y}$. More precisely, we assume that the malicious node directly chooses the deception Δ associated with its fake position and then it chooses the fake position with uniform probability over all the fake positions with deception Δ. This reduction is safe and allows us to reduce the dimensions of the malicious node's strategy space from two ($S \subset \mathbb{R}^2$) to one ($\Delta \subset \mathbb{R}$).

Given this reduction and the assumption that each non-malicious node can appear uniformly on all the values of the deception, we derive Ω as function of Δ. Ω is a mixed probability distribution, including discrete probability and continuous probability measure. More precisely, when $\Delta = 0$, Ω returns a probability and this probability is the ratio between the area of the triangle whose vertices are the verifiers and the total area monitored by the three verifiers: $\frac{\frac{\sqrt{3}R^2}{4}}{(\pi - \sqrt{3})\frac{R^2}{2}} \approx 0.61$. Indeed, if a non-malicious node appears inside the triangle, it will be perfectly localized and therefore the deception will be zero. Instead, if the non-malicious node appears outside the triangle, its potential deception is not zero. In this case, $\Omega(\Delta) = \frac{6\sqrt{R\frac{1}{4} - \Delta^2 - 2\Delta R\sqrt{3}}}{R^2\frac{3}{2}(\frac{\pi}{3} - \sin(\frac{\pi}{3}))}$. We report in Fig. 9, with blue color, the graph of Ω as Δ varies between $[0, \Delta_\infty]$, we recall that in this case $\Delta_\infty = 0.2679R$.

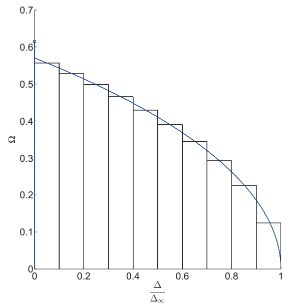
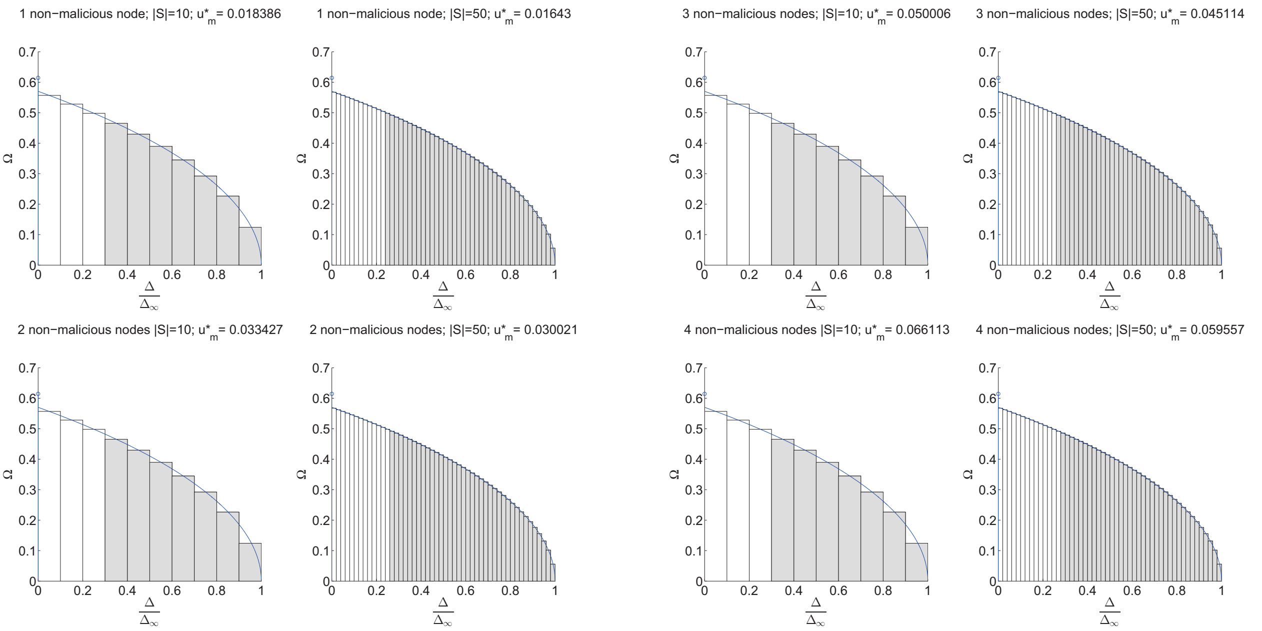


Fig. 9. Probability distribution Ω as function of $\frac{\Delta}{\Delta_\infty}$ (when $\frac{\Delta}{\Delta_\infty} = 0$, Ω is a probability, when $\frac{\Delta}{\Delta_\infty} > 0$, Ω is a probability measure) and a discretization with $|S| = 5$.

We discretize the possible values of Δ in $|S|$ regular intervals. Each interval $[\Delta_i, \Delta_{i+1}]$ is associated with the value of the centroid (*i.e.*, $\Delta = \frac{\Delta_{i+1} + \Delta_i}{2}$) and with probability at the centroid (*i.e.*, $\Omega(\frac{\Delta_{i+1} + \Delta_i}{2})$ —after the discretization the probability is normalized such that the sum of all the probabilities is one). An example of discretization with $|S| = 5$ is reported in Fig. 9.



(a) (b)

Fig. 10. Optimal malicious strategies' supports (gray) for different $|S|$ with 1, 2 (a), 3, and 4 (b) non-malicious nodes.

5.3.2 Empirical results

We provide two empirical results:

- how the strategy of the malicious node varies as $|S|$ and n vary;[9]
- how the expected utility u_m^* of the malicious node at the equilibrium varies as $|S|$ and n vary.

At first, we study how the strategy of the malicious node at the equilibrium changes as discretization grain $|S|$ changes and as the number of non-malicious nodes changes. We searched for a Nash equilibrium with $|S| \in [10, 50]$ and a step of 2 applied to the experimental setting described in the previous section. Given that multiple Nash equilibria can coexist in a single game, each with different properties, we searched a specific Nash equilibrium to have a consistent comparison of the strategies. More precisely, we searched for the Nash equilibrium minimizing the expected utility of the malicious node by solving the mathematical programming problems described in Section 5.2.2 with the objective function $\min\ u_{\mathbf{m}}^*$. In Fig. 10(a) and 10(b) we report the most significant experimental results, with $|S| \in \{10, 50\}$, for different values of n (plots with other values of $|S|$ are omitted due to reasons of space). Each subfigure reports in gray the values of deception that are played by the malicious node with strictly positive probability (the strategy can be easily recovered by assigning each action Δ with the probability $\Omega(\Delta)$ and then normalizing the probabilities to one).

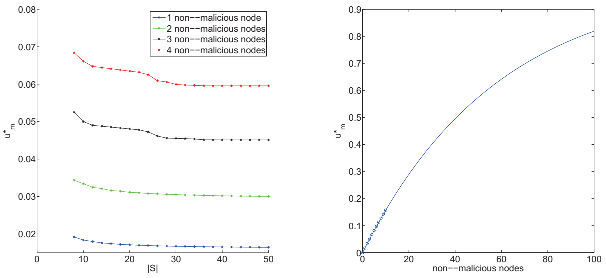
It can be observed that the strategy of the malicious node is characterized by a minimal played deception $\underline{\Delta}$ such that all the deceptions $\Delta < \underline{\Delta}$ are not played, while all the deceptions $\Delta \geq \underline{\Delta}$ are played. Thus, strategies can be conveniently characterized by studying how $\underline{\Delta}$ varies as the values of the parameters change. Initially, we evaluate how $\underline{\Delta}$ varies as $|S|$ varies. When $|S|$ increases, $\underline{\Delta}$ rapidly converges to a stable value. In our experiments, we observed that increasing $|S|$, $\underline{\Delta}$ reduces and the difference in terms of $\underline{\Delta}$ between a given $|S|$ and $|S|+1$ goes to zero. For instance, when there is only one non-malicious node, the difference between $\underline{\Delta}$ with $|S| = 50$ (*i.e.*, $0.26\Delta_\infty$) and with $|S| = 10$ (*i.e.*, $0.32\Delta_\infty$) is about 13%, while the difference between $\underline{\Delta}$ with $|S| = 50$ (*i.e.*, $0.26\Delta_\infty$) and $|S| = 26$ (*i.e.*, $0.28\Delta_\infty$) is about 7%. It can be easily derived that with $|S| = 50$ the exact equilibrium (without discretization) is $\pm 2\%$ w.r.t. the approximate equilibrium (with discretization). $|S| = 50$ is thus is a satisfactory discretization. It can be observed that with $n > 2$ results are similar. We evaluated also how $\underline{\Delta}$ changes as the number n of non-malicious nodes changes. It can be observed that $\underline{\Delta}$ increases as the number of non-malicious nodes increases. Summarily, this is because the probability with which the malicious node will be chosen by the verifiers decreases as the number of non-malicious nodes increases and the malicious node can focus its strategy on larger deceptions (as a result the expected utility increases, as shown also below). The details follow: the optimal verifiers' strategy is to randomize with uniform probability over

[9] We focus on the strategy of the verifiers because, differently from the strategy of the verifiers, it can be conveniently graphically represented. In addition, the strategy of the verifiers can be easily derived from the one of the malicious node: the verifiers randomize uniformly over all the nodes whose deception is played with strictly positive probability by the malicious node.

all the nodes whose deception is played with strictly positive probability by the malicious node. In the case the randomization is over all the nodes, the probability to choose the malicious node is $\frac{1}{n}$ and therefore it reduces as $O(\frac{1}{n})$ as n increases. If $\underline{\Delta}$ increases, the probability that a non-malicious node appears on a deception that is not played by the malicious node increases and therefore the probability with which the malicious node is chosen by the verifiers increases. On the other hand, increasing $\underline{\Delta}$, the malicious node increases its utility. The optimal strategy is given by a tradeoff between the minimization of the probability to be chosen and the maximization of $\underline{\Delta}$. With two non-malicious nodes, $\underline{\Delta}$ increases by $0.02\Delta_\infty$ w.r.t. the case with a single non-malicious node. When the number of non-malicious nodes goes to infinity the detection probability of the malicious node goes to zero, and therefore the optimal strategy of the malicious node is to play Δ_∞ with a probability of one. Indeed, as the number of non-malicious nodes goes to infinity, also the number of nodes at $\Delta = \underline{\Delta}_\infty$ is infinity and therefore the probability to choose the malicious one is zero. From the above considerations, $\underline{\Delta}$ increases monotonically as the non-malicious nodes increase and converges to Δ_∞. (The analysis with $n > 5$ is computationally hard given that the game tree rises exponentially with n and requires the development of *ad hoc* exact and approximate algorithms; this issue is beyond the aim of this paper.)

Finally, we evaluate how the expected utility of the malicious node at the equilibrium changes as $|S|$ and the number of non-malicious nodes changes. Given that multiple equilibria can coexist in a game, we evaluate also the range of the expected utility for all the possible equilibria by finding the Nash equilibrium maximizing the expected utility of the malicious node and that minimizing it. In addition, we evaluate the safety value of the malicious node to compare it w.r.t. the expected utility of the Nash equilibria. Surprisingly, the maxmin value, the value of the best Nash, and value of the worst Nash perfectly overlap for almost all the values of $|S|$ and, when they do not overlap, the difference is very small, being about 1%. This shows that the maxmin value is a very close approximation of the expected utility of the Nash equilibria. In addition, it shows that all the Nash equilibria are essentially the same and these equilibria overlap with the malicious node's maxmin strategy. The result is of paramount importance, because computing the maxmin value is easy, while computing a Nash equilibrium is hard, and therefore, by exploiting the maxmin formulation, the algorithm can scale and solve much larger settings.

Now, we evaluate how the expected utility varies as $|S|$ varies, see Fig. 11(a). It reduces as $|S|$ reduces converging to a given value. As already discussed above, the convergence is relatively fast and at $|S| = 50$ the expected utility results stable. In addition, we anticipated above, the expected utility of the malicious node increases as the number of non-malicious node increases. Given the impossibility to solve settings with a very large number of nodes, even by using the maxmin formulation, we estimate by regression how the expected utility of the malicious node increases. We used an exponential regression, given that, when $n = 1$, the utility of the malicious node is 0 and, when $n \to +\infty$, it is 1. The resulting regression curve is depicted in Fig. 11(b) with $n \in [5, 100]$. It can be observed that the expected utility of the malicious node is relatively small when n is not excessively large, showing that. although the malicious node can pretend a fake position with deception Δ_∞, the fake position the malicious node pretends in average is much smaller (*e.g.*, $\sim 0.2\Delta_\infty$ with $n = 10$) and therefore our approach allows one to dramatically improve the security of VM.



(a) (b)

Fig. 11. (a) Expected utility at the Nash equilibrium as a function of $|S|$ for different numbers of non-malicious nodes. (b) Regression curve for the malicious node's expected utility at the equilibrium.

6 Related Works

The employment of game theoretical tools for security is currently explored in a number of different scientific communities, including computer security, artificial intelligence and robotics, and telecommunications. The most studied scenario consists in the strategic allocation of resources in adversarial settings. Customarily, a security game takes place in an environment where a player (called *attacker*) threatens the engagement of malicious activities and a player (called *defender*) operates in the continuous attempt of detecting them. Customarily, the attacker has a finite number of targets (*e.g.*, nodes, packets, locations) of interest where to start an attack. On the other side, the defender has a finite number of resources per time unit to protect a subset of targets. The impossibility of securing all targets at the same time entails the need for computing a (randomized) resource allocation strategy. Against this background, solving a security game means to apply a solution concept to the corresponding two-player non-cooperative game [13] to derive the optimal strategy for the players.

Starting from the seminal work of von Neumann's hide-and-seek games [23] to date, security games have been adopted in different scenarios. Several works addressed the situation in which the defender controls a *pursuer* with the objective of clearing the environment from the presence of an *evader* which, in turn, is controlled by the attacker [24]. When no assumptions are made over the movement of pursuer and the evader, the games are named infiltration games [25]. When the evader starts from a source and tries to reach a target without being intercepted by the pursuer, the games are named interdiction games [26]. Other interesting variations of security games are search games [27], where the defender's objective is to identify the location of a stationary malicious activity. Opposite situations, where the attacker can move a resource and the defender cannot, come under the name of ambush games [28].

In the specific field of WSNs, four main applications for security are currently studied in the literature (see [29] for a detailed survey): preventing denial of services (DoS) attacks [30], [31], [32], intrusion detection [33], [34], [35], strengthening security [36], [37], coexistence of malicious and non-malicious nodes [38]. In the following we briefly review the main works.

The basic model to prevent DoS attacks is a two-player general-sum non-cooperative game between the attacker node and the WSNs [31]. Given a fixed node i, the attacker's available actions are: attack sensor node i does not attack at all, or attack a different actor sensor node; while the WSNs' available actions are two: defend sensor node i, or defend a different sensor node. The authors resort to the notion of equilibrium to design novel schemes for preventing DoS attacks.

In [33] the authors study sensor network broadcast environment, where malicious sensor nodes can deprive other sensor nodes from receiving a broadcast message. They model the situation as a zero-sum two-player game between the attacker and the intrusion detection system. The attackers' best strategy is to compromise all neighbor sensor nodes of the base station in such a way to achieve maximum payoff. Once the attacker is detected, its payoff goes to zero.

In [36] the authors propose a secure auction based routing protocol by means of the First-Price auction. Both malicious and non-malicious sensor nodes compete to forward incoming packets and, by doing so, each sensor node improves its reputation among other sensor nodes. The sensor nodes decide by themselves to whether to participate in an auction, whereas a malicious sensor node tries its best to win the bid, drop the packets, and corrupt the network. The payoff of each sensor node is calculated based on battery power and reputation.

In [38] the authors analyze the interactions between a malicious sensor node and a non-malicious sensor node in WSNs. The coexistence can give both the malicious and non-malicious sensor nodes different benefits. The authors model the interactions as a pair of games. The first game is a signaling game (*i.e.,* Bayesian extensive-form game). The second game is played when the non-malicious sensor node knows confidently that its opponent is a malicious sensor node.

The work described in this paper proposes a problem different from those above. Indeed, the adversarial secure localization problem appears to be original in the literature. In addition, our work distinguishes from the others on WSN security for the following reasons: the other works propose simple game theoretical models without posing attention on the computation of solution in practice, instead in our work we provide a lot of computational results directed to find equilibria. In terms of computational contributions, our work is closer to [39], [40], [41], [42], where the aim is the development of algorithms to find optimal strategies in large settings, including securing the Los Angeles International Airport, mobile robot patrolling, and malicious packet detection in computer networks.

7 Conclusions

In this paper, we studied a novel game theoretical scenario for WSNs where Verifiable Multilateration is employed to assess the presence of malicious nodes. We built a game theoretical framework where verifiers and malicious nodes compete as rational players. First, we studied the best placement of the verifiers to minimize the maximum deception of the malicious node and we derived the equilibrium prescribing optimal strategies. We studied the case with three verifiers and we extended the result to an arbitrary number of verifiers showing how, as this number increases, the maximum deception of the malicious node decreases. Second, we studied how the malicious node changes its strategy when a number of non-malicious nodes are present. We did this by considering the best strategy for the malicious node when verifiers can inspect one node. To find the equilibrium, we provided a MILP formulation and we experimentally showed that the Nash equilibria of the game almost everywhere coincide with the malicious node's maxmin strategy.

One of the future directions of this work will be along the theoretical analysis our model in the attempt to prove that the malicious node's maxmin strategy corresponds to the optimal strategy at the Nash equilibrium. We also aim at extending our framework to handle multiple malicious nodes, additional security countermeasures, and energy constraints.

Acknowledgments

This work is partially funded by the EU Programme IDEAS-ERC, Project 227977-SMScom.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on wireless sensor network," *IEEE WIREL COMMUN*, vol. 40, no. 8, pp. 102–114, 2002.

[2] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, and H. Yim-Fun, "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards," *COMPUT COMMUN*, vol. 30, no. 7, pp. 1655–1695, 2007.

[3] L. Hu and D. Evans, "Localization for mobile sensor networks," in *MOBICOM*, 2004, pp. 45–57.

[4] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE PERS COMMUN*, vol. 7, no. 5, pp. 28–34, 2000.

[5] S. Čapkun, M. Hamdi, and J.-P. Hubaux, "Gps-free positioning in mobile ad-hoc networks," *CLUSTER COMPUT*, vol. 5, no. 2, pp. 157–167, 2002.

[6] J. Chen, K. Yao, and R. Hudson, "Source localization and beamforming," *IEEE SIGNAL PROC MAG*, vol. 19, no. 2, pp. 30–39, 2002.

[7] L. Doherty, K. Pister, and L. E. Ghaoui, "Convex position estimation in WSNs," in *IEEE INFOCOM*, 2001, pp. 1655–1663.

[8] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher, "Range–free localization schemes for large scale sensor networks," in *ACM MOBICOM*, 2003, pp. 81–95.

[9] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *IEEE GLOBECOM*, 2001, pp. 2926–2931.

[10] V. Ramadurai and M. Sichitiu, "Localization in wireless sensor networks: A probabilistic approach," in *ICWN*, 2003, pp. 275–281.

[11] A. Savvides, H. Park, and M. B. Srivastava, "The bits and flops of the n–hop multilateration primitive for node localization problems," in *WSNA*, 2002, pp. 112–121.

[12] S. Čapkun and J. Hubaux, "Secure positioning in wireless networks," *IEEE J SEL AREA COMM*, vol. 24, no. 2, pp. 221–232, 2006.

[13] Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.

[14] N. Gatti, M. Monga, and S. Sicari, "Localization security in wireless sensor networks as a non–cooperative game," in *IEEE ICUMT*, 2010, pp. 295–300.

[15] ——, "A localization game in wireless sensor networks," in *GAMESEC*, vol. 6442, 2010, pp. 168–179.

[16] S. Brands and D. Chaum, "Distance–bounding protocols," in *EUROCRYPT*, 1994, vol. 765, pp. 344–359.

[17] M. Monga and S. Sicari, "On the impact of localization data in WSNs with malicious nodes," in *SIGSPATIAL*, 2009, pp. 63–70.

[18] R. Poovendran, C. Wang, and S. Roy, *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks (Advances in Information Security)*. Springer-Verlag New York, Inc., 2006.

[19] M. Fewell, "Area of common overlap of three circles," Australian Maritime Operations Division Defence Science and Technology Organisation, Tech. Rep. DSTO-TN-0722, 2006.

[20] M. Wessendorf, "The on–line encyclopedia of integer sequences," http://www.research.att.com/~njas/sequences/A067771, 2002, AT&T Labs Research.

[21] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*. Princeton University Press, 1947.

[22] T. Sandholm, A. Gilpin, and V. Conitzer, "Mixed–integer programming methods for finding Nash equilibria," in *AAAI*, 2005, pp. 495–501.

[23] M. Flood, "The hide and seek game of von neumann," *MANAGE SCI*, vol. 18, no. 5, pp. 107–109, 1972.

[24] M. Adler, H. Räcke, N. Sivadasan, C. Sohler, and B. Vöcking, "Randomized pursuit–evasion in graphs," *COMB PROBAB COMPUT*, vol. 12, pp. 225–244, 2003.

[25] S. Alpen, "Infiltration games on arbitrary graphs," *J MATH ANAL APPL*, vol. 163, no. 1, pp. 286–288, 1992.

[26] A. Wahsburn and K. Wood, "Two–person zero–sum games for network interdiction," *OPER RES*, vol. 43, no. 2, pp. 243–251, 1995.

[27] S. Gal, *Search games*. Academic Press, 1980.

[28] W. Ruckle, R. Fennel, P. Holmes, and C. Fennemore, "Ambushing random walk I," *OPER RES*, vol. 24, pp. 314–324, 1976.

[29] S. Shen, G. Yue, Q. Cao, and F. Yu, "A survey of game theory in wireless sensor networks security," *Journal of Networks*, vol. 6, no. 3, pp. 521–532, 2011.

[30] A. Agah, S. K. Das, and K. Basu, "A game theory based approach for security in WSNs," in *IEEE IPCCC*, 2004, pp. 259–263.

[31] A. Agah, K. Basu, and S. K. Das, "Preventing DoS attack in sensor networks: a game theoretic approach," in *IEEE ICC*, 2005, pp. 3218–3222.

[32] A. Agah and S. K. Das, "Preventing dos attacks in wireless sensor networks: A repeated game theory approach," *International Journal Network Security*, vol. 5, no. 2, pp. 145–153, 2007.

[33] J. M. McCune, E. Shi, A. Perrig, and M. K. Reiter, "Detection of denial–of–message attacks on sensor network broadcasts," in *IEEE SP*, 2005, pp. 64–78.

[34] Y. B. Reddy, "A game theory approach to detect malicious nodes in WSNs," in *IEEE GLOBECOM*, 2009, pp. 259–263.

[35] M. S. Kodialam and T. V. Lakshman, "Detecting network intrusions via sampling: A game theoretic approach," in *IEEE INFOCOM*, 2003.

[36] A. Agah, K. Basu, and S. K. Das, "Security enforcement in wireless sensor networks: A framework based on non-cooperative games," *Pervasive and Mobile Computing*, vol. 2, no. 2, pp. 137–158, 2006.

[37] X. Li and M. R. Lyu, "A novel coalitional game model for security issues in wireless networks," in *IEEE GLOBECOM*, 2008, pp. 1962–1967.

[38] W. Wang, M. Chatterjee, and K. Kwiat, "Coexistence with malicious nodes: a game theoretic approach," in *GAMENETS*, 2009, pp. 277–286.

[39] J. Pita, M. Jain, J. Marecki, F. O. nez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, "Deployed ARMOR protection: the application of a game theoretic model for security at the LAX airport," in *AAMAS*, 2008, pp. 125–132.

[40] N. Basilico, N. Gatti, and F. Amigoni, "Leader–follower strategies for robotic patrolling in environments with arbitrary topologies," in *AAMAS*, 2009, pp. 57–64.

[41] N. Basilico, N. Gatti, and F. Villa, "Asynchronous multi–robot patrolling against intrusions in arbitrary topologies," in *AAAI*, 2010.

[42] O. Vaneky, Z. Yin, M. Jain, B. Boransky, M. Tambe, and M. Pechouceky, "Game–theoretic resource allocation for malicious packet detection in computer networks," in *AAMAS*, 2012.

Nicola Basilico received a M.S. degree in Computer Science in 2007 and a Ph.D. in Information Technology in 2011 from Politecnico di Milano (Italy). In 2012, he received the AIxIA award for the best italian Ph.D. thesis in Artificial Intelligence. He is currently a Post-doctoral Scholar at University of California, Merced. His main research interests are navigation strategies for autonomous mobile robots, probabilitic search and patrolling, security games and algorithmic game theory.

Nicola Gatti received his M.S. degree in Biomedical Engineering and his Ph.D. degree in Information Engineering from Politecnico di Milano (Italy) in 2001 and 2005, respectively. Since 2006, he is Assistant Professor in Computer Science at Politecnico di Milano. His research interests focus on algorithmic game theory, computational microeconomics, artificial intelligence, operations research. In 2011, he received the AIxIA award as best young Italian researcher in artificial intelligence.

Mattia Monga received a M.S. degree in Electronic Engineering and a Ph.D. in Computer and Automation Engineering from Politecnico di Milano, Italy. He is currently an Associate Professor with the Department of Computer Science of Università degli Studi di Milano, Italy. His research interests focus on software engineering, security, and computer science education.

Sabrina Sicari Sabrina Sicari is an Assistant Professor at Università degli Studi dell'Insubria (Italy). She received her master degree in Electronic Engineering in 2002 and her Ph.D. in Computer and Telecommunications Engineering in 2006 from Università degli Studi di Catania (Italy). From September 2004 to March 2006 she has been a research scholar at Politecnico di Milano. Her research interests are on wireless sensor networks, risk assessment methodology and privacy models.