Spoof Resilient Coordination in Distributed and Robust Robotic Networks

Venkatraman Renganathan · Kaveh Fathian · Sleiman Safaoui · Tyler H. Summers

Received: date / Accepted: date

Abstract As cyber-physical networks become increasingly equipped with embedded capabilities, they are made vulnerable to malicious attacks with the increased number of access points available to attackers. A particularly pernicious attack is spoofing, in which a malicious agent spawns multiple identities and can compromise otherwise attack-resilient algorithms that rely on assumed network robustness structures. We generalize a class of resilient consensus strategies, known as Weighted Mean-Subsequence-Reduced (W-MSR) consensus, to further provide spoof resilience by incorporating a physical layer authentication. By comparing the physical fingerprints of received signals, legitimate agents can identify and isolate malicious agents that attempt spoofing attacks. In stochastic settings where fingerprint signals are noisy, we quantify worst-case misclassification probability using distributionally robust Chebyshev bounds computed via semidefinite programming. Numerical simulations and experimental results illustrate the effectiveness of the proposed methods. Our framework is applicable to a variety of problems involving multi-robot systems coordinating via wireless communication.

Keywords Spoof Attack · Network Resiliency · Robot Coordination · Graph Robustness

Supplementary Material

Video of the experimental results and simulation is available at https://youtu.be/dcd0EexMnzE.

1 Introduction

As cyber-physical networks become increasingly equipped with embedded sensing, communication, computation, and actuation capabilities, they are made vulnerable to malicious attacks by increasing the number of access points available for attackers. A large and growing literature has emerged on security, resilience, and robustness of the cyber-physical systems in the presence of non-cooperative and adversarial agents Zhu and Basar (2015); Cardenas et al (2009); Banerjee et al (2012); Albert et al (2000). A particularly pernicious attack is spoofing¹, in which a malicious agent spawns multiple non-existent identities or impersonates existing legitimate agents to gain a disproportionate advantage in distributed algorithms that operate on the network. Spoofing is not just an abstract concern; successful attacks have been realized in several critical networks, such as civilian GPS Humphreys et al (2008), global navigation satellite systems Psiaki and Humphreys (2016),

¹ Also known as a “Sybil” attack Douceur (2002).
anti-lock braking systems Shoukry et al (2013), and others.

Autonomous multi robot systems are a rapidly emerging type of cyber-physical network in which many tasks, including coverage, distributed estimation, cooperative manipulation, and formation control, utilize distributed consensus protocols to coordinate agreement on certain quantities of interest Ren et al (2007); Cortes et al (2004). Recent work has developed resilient consensus algorithms that prevent malicious agents from exerting undue influence on the network, effectively by designing sufficient redundancy in the algorithms and underlying network structures Zhang and Sundaram (2013); Zhang (2012); Zhang et al (2015); LeBlanc et al (2013, 2012); Zhang and Sundaram (2012b); Pasqualetti et al (2012); Zhang and Sundaram (2012a). These approaches have recently been applied to multi-robot systems Saulnier et al (2017); Guerrero-Bonilla et al (2017); Saldana et al (2016). However, spoofing attacks can easily compromise these otherwise attack resilient algorithms and network structures, which assume an upper bound on the number of malicious agents in the network. Thus, malicious information easily propagates through the network and can lead to severe performance deterioration or safety constraint violations.

It was argued in Douceur (2002) that any defense against a spoofing attack requires either a trusted central authority to certify (perhaps cryptographically) the identities of all legitimate agents in the network, or a reliable method to distinguish physical fingerprints of signals received from neighboring agents. Reliance on a centralized authority is generally an undesirable feature in distributed multi-robot networks, so we focus here on the latter. Physical fingerprint analysis and discrimination has been used to detect spoofing in specific application contexts Humphreys et al (2008); Gill et al (2015), but not in the context of general distributed algorithms in cyber-physical networks or dynamic multi-robot systems. Due to the noise in the communication channel, there is also a high chance that a legitimate robot can wrongly classify a legitimate neighbor as spoofed and vice-versa. Thus it is important to quantify the worst-case probability of such misclassification scenarios.

**Contributions:** The present paper is a significant extension of our preliminary work in Renganathan and Summers (2017). We propose an approach for spoof resilient coordination in cyber-physical and multi-robot networks. We generalize a class of resilient consensus strategies, known as Weighted Mean Subsequence Reduced (W-MSR) consensus, to provide spoof resilience by incorporating a physical fingerprint analysis of signals received from neighboring agents. Performing physical layer authentication by comparing the physical fingerprints of received signals, legitimate agents can detect and isolate malicious agents that attempt spoofing attacks. Our algorithm achieves resilient consensus despite an arbitrary number of spoofed agents in the network. In stochastic settings where fingerprint signals are noisy, we quantify worst-case misclassification probability using distributionally robust Chebyshev bounds computed via semidefinite programming, using techniques in Vandenberghe et al (2007). Numerical simulations and experimental results using Sphero rolling robots swarms illustrate the effectiveness of the proposed methods. Our framework is applicable to a variety of problems involving multi-robot systems coordinating via wireless communication, including coverage, distributed estimation, and formation control.

The rest of the paper is organized as follows. Section 2 formulates a model for spoofing attacks in multi-robot networks and presents the attack detection technique using a physical fingerprint analysis. Section 3 proposes a generalized spoof resilient W-MSR algorithm by suitably modifying an existing standard W-MSR algorithm. A problem of quantifying misclassification probability is addressed in Section 4, where a semidefinite programming formulation is proposed to arrive at distributionally robust Chebyshev bounds. Numerical simulation results are then presented in Section 5. Section 6 explains the experimental results obtained by implementing the proposed algorithms on a Sphero rolling robot swarm. Finally, Section 7 summarizes the results and discusses future work directions.

## 2 Spoofing Attacks In Multi-Robot Networks

### 2.1 Spoofing Attack - Network Model

Figure 1 illustrates an example of a spoofing attack we aim to address. We model the network with an undirected graph $G$ comprising a node set $V$ representing $m$ agents and edge set $E[t] \subset V \times V$ representing a set of (possibly time-varying) communication links amongst the agents. The node set is partitioned into two disjoint subsets $V = S_l \cup S_m$. The set $S_l$ represents the set of legitimate agents. A malicious agent attempts to disrupt the network by communicating subversive information to neighboring agents and may in addition attempt to perform a spoofing attack by creating multiple non-existent identities. Thus, the set of adversaries $S_a$ is composed of both malicious and spoofed agents, so that $S_a = S_m \cup S_s$, where $S_m$ denotes malicious agents and $S_s$ denotes the agents spoofed by $S_m$. An upper bound of $F$ number of malicious agents is assumed, whereas an arbitrary number of agents could be spoofed.
Spoof Resilient Coordination in Distributed and Robust Robotic Networks

2.2 Consensus Dynamics - Update Model

We associate with each node \( i \in \mathcal{V} \), a state \( x_i[t] \in \mathbb{R} \) at time \( t \in \mathbb{Z}_{\geq 0} \). The state may represent a position or some quantity to be estimated or optimized, depending on the application context. In order to achieve some objective, the nodes interact synchronously by exchanging their state value with neighbors in the network Olfati-Saber et al (2007); Tsitsiklis et al (1986); Jadbabaie et al (2003). Each legitimate node updates its own state over time based on its current state and the state of neighboring agents according to a prescribed rule of the form

\[
x_i[t + 1] = f_i(x_j[t]), \quad j \in \mathcal{J}_i[t] = \mathcal{N}_i[t] \cup \{i\}, \quad i \in \mathcal{S}_i, (1)
\]

where \( \mathcal{N}_i[t] = \{ j \in \mathcal{V} : (j, i) \in \mathcal{E}[t] \} \) is the neighbor set of agent \( i \) at time \( t \), whose states are available to agent \( i \) via communication links. The degree of \( i \) is denoted as \( d_i[t] = \| \mathcal{N}_i[t] \| \), and every node is assumed to have access to its own state at time \( t \). Resilient consensus algorithms Sundaram and Hadjicostis (2011) specify a nonlinear function \( f_i(\cdot) \) that updates the states by suitably modifying which agents in the neighbor set (including \( i \)) \( \mathcal{J}_i[t] \) are included in the update to provide resilience to malicious agents.

2.3 Attack Detection Using Physical Fingerprint Analysis

We imagine a scenario in which the agents in the network communicate amongst themselves using a wireless communication protocol. Physical properties of the received wireless signal profiles are leveraged to detect the spoofing attack. The physical fingerprint of an agent \( j \) received by agent \( i \) is modeled by a \( d \)-dimensional feature vector \( \mathcal{F}_{ij} \in \mathbb{R}^d \) containing physical signal properties, such as angle-of-arrival, time-of-arrival and other features that can be used for discriminating between the signals received from two distinct agents Humphreys et al (2008); Gill et al (2015). Since multiple spoofed agents may be generated by a single distinct malicious agent, the physical fingerprints associated with each of them will be similar. Of course, due to the random nature of wireless communication, there may be noise associated with the fingerprints of received signals. This situation can be modeled by associating a probability distribution with received signal fingerprints.

The neighbor set \( \mathcal{N}_i \) of agent \( i \) includes the set of agents which can transmit signals to agent \( i \). Based on the received signal fingerprints of pairs of neighboring agents, we define a similarity metric

\[
\gamma_{ijk} = \frac{1}{1 + \| \mathcal{F}_{ij} - \mathcal{F}_{ik} \|^2}, \quad j, k \in \mathcal{N}_i, (2)
\]

which quantifies how similar the fingerprint of neighboring agent \( j \) is to that of neighboring agent \( k \), as received by agent \( i \). The authors in Gill et al (2015) deal with a similar setting for a coverage control problem and our development is inspired from their approach. Agent \( i \) computes these similarity metrics for each neighbor pair. From these similarity metrics, a confidence weight \( \alpha_{ij} \in [0, 1] \) can be associated with neighboring agent, which should be close 1 for legitimate neighbors and close to 0 for spoofed and spoofing neighbors. For example, in a deterministic setting,

\[
\gamma_{ijk} = 1 \quad \Rightarrow \quad \alpha_{ij} = 0, \quad \alpha_{ik} = 0,
\]

\[
\gamma_{ijk} < 1 \quad \Rightarrow \quad \alpha_{ij} = 1, \quad \alpha_{ik} = 1,
\]

i.e., the confidence weights for neighbors \( j \) and \( k \) are 0 if the neighbor \( j \) has the same fingerprint as neighbor \( k \), and 1 otherwise. In general, we can write

\[
\alpha_{ij} = \prod_{j \in \mathcal{N}_i, j \neq k} (1 - \gamma_{ijk}). (4)
\]

In a stochastic setting, we define a spoof detection threshold \( \omega \in [0, 1] \). If the likelihood that the physical fingerprints of two neighbors are different is below the threshold, the neighbors are classified as spoofed or spoofing agents, and otherwise they are classified as legitimate; for example,

\[
g(\alpha_{ij}) < \omega \quad \Rightarrow \quad j \text{ is spoofed or spoofing},
\]

\[
g(\alpha_{ij}) > \omega \quad \Rightarrow \quad j \text{ is legitimate}, (5)
\]

where \( g(\cdot) \) is a prescribed detection function. In the stochastic settings, the \( g(\cdot) \) and \( \omega \) could be selected...
based on an assumed model for the probability distributions of the fingerprints and associated bounds on misclassification probability. To recover the deterministic case, we can set $\omega = 0$ and $g(x) = x$.

2.4 Example of a Physical Fingerprint Model

A specific example of a particular physical fingerprint model is discussed in Gill et al. (2015). The fingerprint is modeled by a directional signal strength profile that depends on wireless signal wavelengths, distances and relative angles between directional antennae, multiple possible signal paths, and random channel properties with additive Gaussian noise. Based on this stochastic channel model, similarity and confidence metrics can be explicitly defined, and quantitative bounds can be obtained on the expectation that received signals are coming from spoofed agents. Such models could be used to define spoof resilient algorithms tailored to the specific communication model and perform analyses of probabilistic algorithm properties. Here we focus mainly on deterministic detection settings and stochastic settings with simple thresholding. Computing bounds associated with such specific probabilistic fingerprint models is described in Section 4.

3 Design of a Spoof Resilient Coordination Algorithm

In this section, we describe a coordination algorithm that is resilient to anonymous malicious agents who share adversarial state values and may also attempt to spoof non-existent agent that also share adversarial state value. Since malicious agents do not necessarily attempt to spoof, we build upon recent work on resilient consensus algorithms that do not handle spoofing. These algorithms achieve resiliency by effectively designing and exploiting redundancy in the underlying communication graph. We now review these resilient graph properties and an existing resilient consensus algorithm called Weighted Mean-Subsequence-Reduction (W-MSR) as described in Zhang et al. (2015). We subsequently present our spoof resilient adaptation of W-MSR. We assume throughout that there are at most $F$ malicious agents but that there may be an arbitrary number of spoofed agents.

**Definition:** A set $S$ is $r$-reachable, if it contains a node that has at least $r$ neighbors outside of $S$. The parameter $r$ quantifies the redundancy of information flow from nodes outside of $S$ to some node inside $S$. Intuitively, the $r$-reachability property captures the notion that some node inside the set is influenced by a sufficiently large number of nodes from outside the set.

**Definition:** A graph $D = (V,E)$ on $n$ nodes is said to be $r$-robust, if $r \in \mathbb{Z}_{\geq 0}$, if for every pair of disjoint nonempty subsets of $V$, at least one of the subsets is $r$-reachable.

**Definition:** Given a graph $D$ and a nonempty subset of nodes $S$, we say that $S$ is an $(r,s)$ - reachable set, if there are at least $s$ nodes in $S$, each of which has at least $r$ neighbors outside of $S$, where $r,s \in \mathbb{Z}_{\geq 0}$. i.e., given $X_S = \{ i \in S : |V_i \setminus S| \geq r \}$, then $|X_S| \geq s$.

**Definition:** A graph $D = (V,E)$ on $n$ nodes $(n \geq 2)$ is $(r,s)$-robust, for nonnegative integers $r \in \mathbb{Z}_{\geq 0}, 1 \leq s \leq n$, if for every pair of nonempty, disjoint subsets $S_1$ and $S_2$ of $V$ such that $S_1$ is $(r,s_{r,1})$-reachable and $S_2$ is $(r,s_{r,2})$-reachable with $s_{r,1}$ and $s_{r,2}$ maximal (i.e., $s_{r,k} = |X_{S_k}|$ where $X_{S_k} = \{ i \in S_k : |V_i \setminus S_k| \geq r \}$ for $k = \{1,2\}$), then at least one of the following hold:

1. $s_{r,1} = |S_1|
2. s_{r,2} = |S_2|
3. s_{r,1} + s_{r,2} \geq s$

The $(r,s)$-robustness property introduces information redundancy by specifying a minimum number of nodes that are sufficiently influenced from outside of their set. Note that $(r,s)$-robustness is a strict generalization of $r$-robustness.

3.1 Resilient Asymptotic Consensus

Let $x_M[t]$ and $x_m[t]$ denote the maximum and minimum values of the legitimate nodes at time $t$, respectively. The legitimate agents in the network are said to achieve resilient asymptotic consensus Zhang and Sundaram (2012b) in the presence of a particular threat model if for any initial conditions it holds

- $\exists L \in \mathbb{R}$ such that $\lim_{t \to \infty} x_i[t] = L, \forall i \in S_t$
- the interval $[x_m[0], x_M[0]]$ is an invariant set (i.e., the legitimate values remain in the interval $\forall t$)

Resilient asymptotic consensus has three important properties LeBlanc et al (2013). First, the legitimate nodes must reach asymptotic consensus despite the presence of some misbehaving nodes given a particular threat model and scope of threat (e.g., at most $F$ malicious agents). This is a condition on agreement. Additionally, it is required that the interval containing the initial values of the legitimate nodes is an invariant set for the legitimate nodes; this is a safety condition, where the interval $[x_m[0], x_M[0]]$ is known to be safe. The agreement and safety conditions, when combined, imply a third condition on validity: the converged consensus value lies within the range of initial values of the legitimate nodes.
3.2 The Weighted Mean-Subsequence-Reduced (W-MSR) Algorithm

We now review a class of resilient consensus algorithms described in Zhang et al (2015) that utilize an update rule called as Weighted Mean-Subsequence-Reduced (W-MSR). At every time $t$, each legitimate node $i$ obtains the values of other nodes in its neighborhood. Since there are at most $F$ total malicious nodes in the network, some of node $i$’s neighbors may misbehave; however, node $i$ is unsure of which neighbors may be compromised. To ensure that node $i$ updates its state in a safe manner, we consider a protocol where each node removes the extreme values with respect to its own value. Specifically, the W-MSR algorithm comprises the following steps:

1. At each time $t$, each legitimate node $i \in S_t$ obtains the state values of its neighbors, and forms a sorted list.
2. If there are less than $F$ values strictly larger than its own value, $x_i[t]$, then legitimate node $i$ removes all values that are strictly larger than its own. Otherwise, it removes precisely the largest $F$ values in the sorted list (breaking ties arbitrarily). Likewise, if there are less than $F$ values strictly smaller than its own value, then node $i$ removes all values that are strictly smaller than its own. Otherwise, it removes precisely the smallest $F$ values.
3. Let $\mathcal{R}_i[t]$ denote the set of nodes whose values were removed by legitimate node $i$ in step 2 at time $t$. Each legitimate node $i$ applies the update

$$
x_{i}[t+1] = \sum_{j \in \mathcal{J}_i[t]\setminus \mathcal{R}_i[t]} w_{ij}[t]x_j[t] \quad (6)
$$

where $w_{ij}[t]$ is the weight associated with node $j$’s value by node $i$ at time step $t$. The weights are chosen to satisfy the following conditions:

1. $w_{ij}[t] = 0$ whenever $j \notin \mathcal{J}_i[t], i \in S_t, t \in \mathbb{Z}_{\geq 0}$
2. There exists a constant $\beta \in \mathbb{R}, 0 < \beta < 1$ such that $w_{ij}[t] \geq \beta, \forall j \in \mathcal{J}_i[t], i \in S_t, t \in \mathbb{Z}_{\geq 0}$
3. $\sum_{j=1}^{n} w_{ij}[t] = 1, \forall i \in S_t, t \in \mathbb{Z}_{\geq 0}$

A network being $(F+1, F+1)$-robust is a necessary and sufficient condition for the normal nodes to achieve consensus when no more than $F$ total malicious nodes are present in the entire network Usevitch and Panagou (2017). Then if the underlying graph is $(F+1, F+1)$-robust, under the update protocol specified in equation (6), the legitimate agents in network are guaranteed to achieve resilient asymptotic consensus Zhang et al (2015) despite the presence of at most $F$ malicious agents, but assuming that there are no spoofed agents.

3.3 Spoof Resilient W-MSR Algorithm

A spoofing attack is capable of compromising the $(F+1, F+1)$-robust graph robustness property and W-MSR algorithm above, and hence the network resiliency. Our spoof resilient adaptation of the W-MSR algorithm here is summarized in Algorithm 1. Based on a pairwise comparison of physical fingerprints of signals received from neighboring agents and associated confidence weights, spoofed agents in the network are identified. Achieving resiliency then involves removing the identified spoofed and spoofing agents from the state update if the expectation of some likelihood of their confidence weight is at most equal to the spoofing threshold $\omega$. Thus in a stochastic setting, a spoofing threshold can be employed as explained in the spoof resilient W-MSR extension presented in algorithm 1. In a deterministic setting, we have the following result.

**Theorem 1** Given an undirected network $G = (V, E)$, where $V$ represents the set of agents and $E$ represents the set of communication links between them. Suppose the network is $(F+1, F+1)$-robust, assuming an upper bound of $F$ total malicious agents in the network, some of which may spoof. Then the network achieves resilient asymptotic consensus under Algorithm 1 in the presence of any spoofing attack.

**Proof** In a deterministic setting, the physical fingerprints of signals of spoofed agents are identical to that of the spoofing agent. So any spoofed and spoofing agents are exactly detected and removed from the network state updates in lines 17-20 of the proposed Algorithm 1. Moreover, since the initial underlying graph is $(F+1, F+1)$-robust, the use of the W-MSR protocol for the state updates guarantees resilient asymptotic consensus in the presence of up to $F$ malicious nodes who do not spoof but may behave in other adversarial ways. Thus, the overall protocol is resilient to both an arbitrary number of spoofed agents and up to $F$ non-spoofing malicious agents in the network.

3.4 Choosing a Spoofing Threshold

Since robots have physical extent, there is a non-zero minimum distance between the sensors or receivers located in each robot that are used for discriminating received signals. Due to this, suppose it is known that the physical fingerprints of every pair of legitimate robots
are separated by a distance of at least $D_{\text{min}}$. This suggests a threshold for classifying neighboring agents as malicious or spoofed.

Specifically, the similarity metrics and confidence weights for legitimate neighbors satisfy

$$
\gamma_{ijk} \leq \frac{1}{1 + D_{\text{min}}} \implies \alpha_{ij} \geq \left( 1 - \frac{1}{1 + D_{\text{min}}} \right)^{|N_i|}.
$$

It follows that $\omega = \left( 1 - \frac{1}{1 + D_{\text{min}}} \right)^{|N_i|}$ would be an appropriate spoofing threshold for correctly discriminating between legitimate and malicious or spoofed neighbors. However, fingerprints of received signals may be stochastic and not easily bounded. In this setting there may inherently be a possibility of misclassifying a malicious neighbor as legitimate. In the next section, we explore the problem of quantifying misclassification probabilities in spoofing detection via physical fingerprint authentication.

**Algorithm 1 Spoof Resilient W-MSR (SR-W-MSR)**

```plaintext
procedure SR-W-MSR(\omega)
  // Input: spoofing threshold \( \omega \), convergence threshold \( \epsilon \), initial states \( x[0] \), received signal fingerprints \( F_{ij} \) for each agent at each time
  \( t \leftarrow 0 \)
  while \( \| x[t+1] - x[t] \| \geq \epsilon \) do
    \( i \leftarrow 1 \)
    while \( i \leq |S_i| \) do
      // Iterate through all legitimate nodes
        for each \( j \in N_i \) do
          if \( j \neq k \) then
            \( \gamma_{ijk} = \frac{1}{\| x_i - F_{ij} \|} \)
          end if
          \( \alpha_{ij} \leftarrow \alpha_{ij}(1 - \gamma_{ijk}) \)
        end for
      if \( \alpha_{ij} \leq \omega \) then
        // Spoof attack is detected
          \( Z[t] \leftarrow R[t] \cup \{j\} \)
          \( x_i[t+1] \leftarrow \sum_{j \in Z[t]} x_i[t] \cdot w_{ij}[t] x_j[t] \)
        else
          // Spoof attack is not detected
            \( x_i[t+1] \leftarrow \sum_{j \in Z[t]} x_i[t] \cdot w_{ij}[t] x_j[t] \)
        end if
      end if
      \( i \leftarrow i + 1 \)
    end while
  end while
end procedure
```

4 Quantifying Misclassification Probabilities in Spoofing Detection

In our proposed physical layer authentication technique, each legitimate robot associates a fingerprint with each wireless communication signal received from neighboring agents. This fingerprint is represented as a vector of signal parameters, such as received signal strength and relative bearing. Since communication channels are inherently noisy, the pairwise comparison performed in Algorithm 1 to classify neighbors as legitimate or malicious/spoofed may always have a chance of incorrectly classifying a spoofed neighbor as a legitimate one or vice versa. By associating probability distributions with fingerprints of received signals, misclassification probabilities can be estimated based on assumptions about the distributions.

True fingerprint distributions are unknown. Based on experimental data with sensors or signal receivers, distributions parameters such as mean and covariance can be estimated. It is common to assume that distributions are Gaussian to maintain tractability of certain probabilistic computations. However, the emerging area of distributionally robust optimization has shown that it is often not necessary to make such strong and often unjustifiable assumptions Wiesemann et al. (2000). Here we adopt a distributionally robust approach to compute worst-case probabilities of misclassification over the set of fingerprint distributions with given mean and covariance. These worst-case estimates can be interpreted as generalized Chebyshev bounds Vandenbergh et al. (2007), which extend classical moment-based Chebyshev inequalities to vector-valued random variables.

Let $X_1 \sim P_1(\mu_1, \Sigma_1)$ and $X_2 \sim P_2(\mu_2, \Sigma_2)$ denote the fingerprints and associated probability distributions of two neighbors received by a robot. For illustrative purposes, we will assume that $X_1$ corresponds to a legitimate robot and $X_2$ to a malicious one. We assume that the true distributions $P_1$, and $P_2$ are unknown but that the means and covariance matrices $\mu_1 \in \mathbb{R}^n, \Sigma_1 \in \mathbb{S}_n, \mu_2 \in \mathbb{R}^n, \Sigma_2 \in \mathbb{S}_n$ are known or estimated from received signal data or sensor hardware datasheets (with $\mathbb{S}_n$ denoting the set of symmetric $n \times n$ matrices). The setup is illustrated in Figure 2.

If the malicious agents attempts a spoofing attack, then all fingerprints of the malicious agent and its spoofed identities are distributed according to $P_2$. A misclassification event occurs when $X_1$ is actually closer to $\mu_2$ than $X_2$, in which case the legitimate neighbor would be classified as malicious by the receiving robot. A similar misclassification event occurs when $X_2$ is actually closer to $\mu_1$ than $X_1$, in which case the malicious neighbor would be classified as legitimate by the receiving...
Fig. 2: Illustration of probability distributions of fingerprints of neighboring agents. Inherent errors in fingerprints may cause the receiving robot to misclassify its neighbors as legitimate or malicious. In general, the true distribution is not known, and probabilities must be estimated based on assumed or estimated quantities, such as means and covariance matrices. In multirobot systems, physical fingerprints are closely related to robot positions, so that neighboring robots which are very close together may lead to increased probability of misclassification.

Accordingly, let us define the following sets

\[ C_1 = \{X_1, X_2 \in \mathbb{R}^n \mid \|X_1 - \mu_2\|^2 \leq \|X_2 - \mu_2\|^2\} \]

\[ C_2 = \{X_1, X_2 \in \mathbb{R}^n \mid \|X_2 - \mu_1\|^2 \leq \|X_1 - \mu_1\|^2\}. \]

(7)

Since the fingerprints are independent, the joint random variable \( X = [X_1, X_2] \) has mean and covariance

\[ \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} S_1 & 0 \\ 0 & S_2 \end{pmatrix}. \]

Now, the probability of misclassification for the first event is given by

\[ \sup\{P(X \in C_1)\} = 1 - \inf\{P(X \notin C_1)\}. \]  

(8)

where the supremum and infimum are taken over the set of probability distributions on \( \mathbb{R}^n \) with the given mean and covariance; an analogous expression gives the misclassification probability for the second event \( C_2 \). Utilizing generalized Chebyshev bounds Vandenberghe et al. (2007), the misclassification probabilities can be readily computed by solving a (convex) semidefinite programming problem.

4.1 Computing Misclassification Probability via Semidefinite Programming

**Theorem 2** Consider the joint random fingerprint variable \( X = [X_1, X_2] \) with mean \( \mathbf{E}X = \mu \) and covariance \( \mathbf{EX}X^T = \Sigma \). The worst-case probability of misclassification

\[ \sup\{P(X \in C_1)\} \]

over the set of all distributions of \( X \) with mean \( \mu \) and covariance \( \Sigma \) is given by the optimal value of the following semidefinite program

\[
\begin{align*}
\text{maximize} & \quad \lambda \\
\text{subject to} & \quad \text{tr}(AZ) + 2b^T z \geq 0 \\
& \quad \begin{pmatrix} Z & z \\ z^T \lambda \end{pmatrix} = \begin{pmatrix} \Sigma + \mu^T \mu & \mu^T \\ \mu^T & 1 \end{pmatrix} \\
& \quad \begin{pmatrix} Z & z \\ z^T \lambda \end{pmatrix} \geq 0 \\
\end{align*}
\]

where \( A = \begin{pmatrix} -I_n & 0_n \\ 0_n & I_n \end{pmatrix}, \quad b = \begin{pmatrix} \mu_2 \\ -\mu_2 \end{pmatrix} \) with variables \( \lambda \in \mathbb{R}, Z \in \mathbb{S}_{2n}, z \in \mathbb{R}^{2n} \).

**Proof** The smaller probability of not misclassifying, namely \( \inf\{P(X \notin C_1)\} \), is given by

\[
\begin{align*}
\inf \{P(\|X_1 - \mu_2\|^2 \geq \|X_2 - \mu_2\|^2) \} \\
= \inf \{P((X_1 - \mu_2)^T(X_1 - \mu_2) \geq (X_2 - \mu_2)^T(X_2 - \mu_2)) \} \\
= \inf \{P(X_1^T X_1 - 2\mu_2^T X_1 + \mu_2^2 \geq 0) \} \\
= \inf \{P(X^T \begin{pmatrix} I_n & 0_n \\ 0_n & -I_n \end{pmatrix} X + 2\begin{pmatrix} -\mu_2 \\ \mu_2 \end{pmatrix}^T X \geq 0) \} \\
= \inf \{P(X^T \begin{pmatrix} -I_n & 0_n \\ 0_n & I_n \end{pmatrix} X + 2\begin{pmatrix} \mu_2 \\ -\mu_2 \end{pmatrix}^T X \geq 0) \} \\
\end{align*}
\]

Then applying the main result in Section 2 of Vandenberghe et al (2007) and combining with (8) yields the result.

Also, an analogous semidefinite programming problem can be formulated for the misclassification probability \( \sup\{P(X \in C_2)\} \). In any case, although misclassifications may occasionally occur, signals may be received relatively frequently. The probability of misclassifications persisting over extended periods of time would rapidly decrease, and robots may be able to improve spoofing classifications by considering fingerprints over multiple time periods. Such analysis will be pursued in future work.
Venkatraman Renganathan et al.

(a) Illustration of one-standard deviation uncertainty ellipses for fingerprints of the legitimate and spoofed neighbors.

(b) Misclassification probability decreases as neighbors become further apart.

Fig. 3: The fingerprints are highly noisy, and the received signal strength indicating relative distance is noisier than the bearing signal indicating relative angle.

4.2 Illustrative Example

Suppose the robots evolve in $\mathbb{R}^2$ and the receiving robot has sensors that allow discrimination of received signal strength and bearing sensors to estimate the positions of neighboring robots, which will serve as the fingerprint estimate. The signal strength measurement in the radial direction toward the neighbors has variance 2, and the bearing measurement in the orthogonal direction has variance 1. This corresponds to a highly noisy scenario. For illustrative purposes, suppose the receiving robot is located at origin. The setup is illustrated in Figure 3a.

The relative positions of legitimate and malicious neighbors are varied symmetrically to modulate the misclassification probability. Note that the symmetry in this example implies that the probabilities of both events mentioned previously are identical. The semidefinite program (9) was solved for the varying neighbor positions to compute the worst-case misclassification probability. As the neighbors got further apart from each other as shown in Figure 3a, the misclassification probability decreased as shown in Figure 3b. These estimates of misclassification probability can be sharpened by making stronger assumptions on the fingerprint distributions.

5 Numerical Simulations

We now illustrate our spoof resilient W-MSR algorithm using the 7-node (2,2)-robust network with 6 legitimate agents, 1 malicious agent who spoofs 1 agent, as shown in Figure 4. The objective of the network is to form a hexagonal formation and remain in the safe region, which can be expressed by introducing a constant bias in the consensus update equations. Specifically, at every iteration the desired position of robot $i = 1, \ldots, 6$ is computed via

$$x_i[t+1] = \sum_{j \in \mathcal{N} \setminus \{i\}} w_{ij}[t] (x_j[t] - \bar{x}_j) + \bar{x}_i.$$  \hspace{1cm} (10)

where $\bar{x}_i := [\sin(\theta_i), \cos(\theta_i)]^\top \in \mathbb{R}^2$, $\theta_i := \frac{2\pi(i-1)}{6}$, is a constant bias vector that is used to position each robot at a vertex of the hexagon. Note that due to the addition and subtraction of the same bias vector for each agent, by defining $\bar{x}_i := x_i - \bar{x}_i$ one can see from (10) that $\bar{x}_i$ has the same dynamics as of Eq.(6). The six legitimate agents were given random initial states as shown in Figure 5a. Malicious node 7 performs spoofing attack by spawning a spoofed identity called 8 and sends messages to all his neighbors using both identities. A constant bias of $\bar{x} = 5$ is added to malicious robot’s position in each dimension to obtain spoofed robot’s position. The goal of the malicious and spoofed
robots is to move the legitimate robot formation from safe region to unsafe region. A spoofing attack is evaluated both in deterministic and stochastic settings. In the deterministic setting, all the physical fingerprints are obtained without any noise. As a result, the spoofed nodes are exactly identified and removed from the network using the spoof resilient W-MSR algorithm.

The rest of the Section is organized as follows. Initially the failure of linear consensus protocols under malicious attacks is demonstrated in Subsection 5.1. Then, the W-MSR algorithms is demonstrated to achieve resiliency against the malicious attack in Subsection 5.2. The failure of the W-MSR algorithm under spoofing attack is demonstrated in Subsection 5.3. Implementation of the proposed spoof resilient extension of W-MSR algorithm is presented in Subsection 5.4. Finally other additional observations are discussed in Subsection 5.5.

5.1 Linear Consensus Protocol Fails with 1 Malicious Agent

Consider the network as shown in Figure 4 with spoofed agent not being present. When a linear consensus protocol is employed to obtain a robot formation, the malicious robot is successful in pulling the formation from safe region to the unsafe region as shown in Figure 5b. This clearly shows that standard linear consensus protocols are not resilient against malicious attacks.

5.2 W-MSR Guarantees Resilience Against 1 Malicious Robot

Consider the network as shown in Figure 4 with spoofed agent not being present. When the W-MSR resilient consensus algorithm Zhang et al (2015) is used to obtain a robot formation, the legitimate robots are resilient against the malicious robot and achieve the desired formation in the safe region as shown in Figure 5c. This shows that the W-MSR is resilient against malicious attacks, but without spoofing.

5.3 The W-MSR Algorithm Fails Under a Spoofing Attack

Consider a spoofing attack on the network shown in Figure 4, where agent 7 is malicious. When the malicious agent spoofs a single additional agent identity, the legitimate agents fail to achieve resiliency and hence they are pulled into unsafe region by malicious robots as shown in the Figure 6b. This shows that spoofing attacks are capable of compromising graph robustness properties and thereby the network resiliency.

5.4 Spoof Resilient W-MSR Achieves Desired Formation and Remains Safe

Consider a spoofing attack on the network shown in Figure 4, where agent 7 is malicious spoofing agent 8. The spoofing attack is simulated for first 10 time steps and then the algorithm switches to spoof resilient version guaranteeing spoof resilient formation of legitimate robots in the safe region. The spoof detection needs to happen as early as possible or else the attack is capable of pulling the robot formation to unsafe region. Thus, under the proposed Spoof Resilient extension of the W-MSR algorithm, the legitimate agents achieve resilient formation, as shown in Figure 6c.

5.5 Variations: Delayed and Probabilistic Spoof Detection

In practical settings, it may take a non-trivial amount of time to detect a spoofing attack, and spoofing may not be detected perfectly due to noise and uncertainty in the received signal properties. It has been shown in our previous work Renganathan and Summers (2017) that with longer delays in spoof detection, the malicious agents were able to cause larger deviations from the network’s final consensus value. Further, it was also shown that a malicious agent’s location in the networks affects how influential it can be in perturbing the state of the network; a spoof attack by agent 5 in Figure 4 with lower degree than agent 7 has a smaller transient impact. However, it is also possible to repair the impact of a spoofing attack on the network after delayed detection by maintaining memory of neighboring state transmission histories and subtracting out modifications made by spoofed and spoofing agents. Of course, such a bookkeeping effort would be limited by memory constraints, and may be cumbersome to implement in complicated networks. We are pursuing how this might be achieved in general for future work.

6 Experimental Results

To demonstrate the performance of the spoof resilient W-MSR algorithm, experiments are performed using a robotic platform to show that a team of robots can achieve a desired formation and remain in a safe region under the spoofing attack. In the experiments, each robot represents a node in the network, and the (2, 2)-robust network with six legitimate and one malicious agent, shown in Figure 4, is adopted to represent the communication among the robots. The objective of legitimate robots is to achieve a hexagon formation in a
safe region, while the malicious agent aims to bring the formation to an unsafe region via the spoofing attack.

6.1 Implementation Details

Our experimental setup consists of seven Sphero 2.0 robots, a Logitech C950 webcam, a Bluetooth enabled smartphone, and a Windows OS computer. All routines executed during the experiment are implemented in MATLAB. We emphasize that although this experimental implementation is centralized, the information that is made available to each robot is restricted according to the communication graph shown in Figure 4. The webcam is set up to overlook a confined area in which the Sphero robots are placed and allowed to move. The LEDs of six legitimate robots are set to emit a blue color, while the malicious robot emits a red color. A color based image segmentation routine Corke (2017) is used to detect and track the robots in real-time from the 640 × 480 images that are fetched from the webcam.

To extract the position of robots from the webcam image, let \((x', y')\) denote the pixel coordinates of a robot’s centroid that is returned from the image segmentation routine. The Euclidean coordinates of the image point in the camera coordinate frame is given by \(p' := K^{-1}[x', y', 1]^\top\), where \(K \in \mathbb{R}^{3 \times 3}\) is the camera intrinsic matrix, which can be obtained beforehand through the camera calibration procedure as explained in Bouguet (2018). Let \(\ell_R, \ell_t\) respectively denote the rotation and translation of the world coordinate frame \(\mathbb{F}\) expressed in the camera coordinate frame \(\mathbb{C}\). We assume that the world coordinate frame is chosen such that the plane in which the robots lie is spanned by the \(x-y\) axes, as illustrated in Fig. 7a. Note that extrinsic camera parameters, i.e., \(\ell_R\) and \(\ell_t\), can be found via the PnP algorithm Lepetit et al (2009), where a checkerboard of a known size can be placed on the plane containing the robots to define the global coordinate frame. Let \(n := [0, 0, 1]^\top\) denote the normal vector of the image plane in the camera coordinate frame \(\mathbb{C}\). The distance of the image point \(p'\) along the \(z\)-axis of the camera is given by the dot product \(d' := \langle n, p' \rangle\). Similarly, distance of the plane containing the robots along the camera \(z\)-axis is \(d := \langle n, \ell_t \rangle\). From the pinhole
Spoof Resilient Coordination in Distributed and Robust Robotic Networks

(a) Robot Projection on the image plane for image reconstruction.

Fig. 7: Projection of a robot on the image plane as shown in (a). Their initial positions for testing all algorithms used below is shown in (b).

camera model Ma et al (2012), it follows that the 3D coordinates of the robot in the camera coordinate frame is $c'p = \frac{g}{d}p'$. Consequently, coordinates of the robot in the world frame is given by $wp = cR_w^t(c'p - ct_w)$.

The recovered coordinates from the procedure above are used in a consensus-based formation control strategy to bring the robots to a hexagon formation. As in the numerical simulations, at every iteration the desired position of robot $i = 1, \ldots, 6$ is computed via

$$x_i[t + 1] = \sum_{j \in J_i[t] \setminus \mathcal{R}[t]} w_{ij}[t] (x_j[t] - \bar{x}_j) + \bar{x}_i, \quad (11)$$

where $\bar{x}_i := [\sin(\theta_i), \cos(\theta_i)]^T \in \mathbb{R}^2$, $\theta_i := \frac{2\pi(i-1)}{6}$, is a constant bias vector that is used to position each robot at a vertex of the hexagon. For the demonstration purposes, a constant bias of $\bar{x} = 50$ is added to malicious robot’s position in each dimension to obtain spoofed robot’s position. Given the desired position $x_i[t + 1]$, a low-level PID controller computes the required linear and angular velocity control commands that guide the robot to this locations at the next iteration. These control commands are communicated to the robots via Bluetooth as explained in Sethi and Campa (2018). The motion of the legitimate robots are controlled by the computer and the malicious robot that aims to move the formation to unsafe region is controlled using smartphone. For all the experimental demonstrations, the robot swarms were started from the same positions as shown in the Figure 7b.

6.2 Observations

Using the 7-node (2,2) robust network as shown in Figure 4, the W-MSR algorithm is implemented with seven robots, one of which is malicious and spoofed robot being absent. The algorithm as shown in Figure 8a is successful at ignoring the malicious robot’s intention to move the formation to an unsafe region and rest of the legitimate robots achieve the resilient formation. But when the W-MSR algorithm ran with the malicious robot spoofing one more additional identity, the assumption on the upper bound on the number of malicious agents in the network is compromised. As shown in the Figure 8b, the legitimate robots achieve the desired formation relative to the malicious robot and when the latter moves from the safe to the unsafe region, it is successful in pulling the network. In the Figure 8c, the legitimate robots perform fingerprint comparison test and detect the spoofing attack as early as possible and remove the malicious robot and its spoofed entities from the network. As a result, the legitimate robots achieve the formation in the safe region and remain resilient there unaffected by the spoofing attack.

7 Conclusions

We proposed a spoof resilient consensus algorithm that extends a class of resilient consensus strategies, known as the Weighted Mean-Subsequence-Reduced (W-MSR) consensus, to provide resilience to malicious agents that
may both adversely update state values and spoof non-existent agent identities. Physical fingerprint comparisons of received signals are used by legitimate agents to identify and isolate malicious agents that attempt spoofing attacks. The proposed algorithm using physical fingerprint approach guarantees resiliency despite the presence of a certain number of malicious agents and an arbitrary number of spoofed agents in the network. A probabilistic spoof detection analysis is presented using a semidefinite programming technique to arrive at distributionally robust Chebyshev bounds for probability of misclassification of robots. Experimental results using Sphero robot swarms and numerical simulations demonstrate the effectiveness of the proposed algorithm. The framework is applicable to a variety of problems involving multi-robot systems coordinating via wireless communication, including coverage, distributed estimation, and formation control. Future research involves investigating the spoof resiliency with different fault models and quantifying the worst-case probability of misclassifications persisting over extended periods of time by considering fingerprints over multiple time periods.

References

Ma Y, Soatto S, Kosecka J, Sastry SS (2012) An invitation to 3-d vision: from images to geometric models. Springer 26
Sundaram S, Hadjicostis CN (2011) Distributed function calculation via linear iterative strategies in the presence of malicious agents. IEEE Transactions on Automatic Control 56(7)
Usevitch J, Panagou D (2017) r-robustness and (r,s)-robustness of circulant graphs. In: 56th IEEE Conf. on Decision and Control, Melbourne, Australia.