Stealthy Attacks on Pheromone Swarming

JANIECE KELLY
Department of Computer Science
Texas State University
jek44@txstate.edu

SETH RICHTER
Department of Computer Science
LeTourneau University
sethrichter@letu.edu

MINA GUIRGUIS
Department of Computer Science
Texas State University
msg@txstate.edu

***Abstract*— In multi-agent systems, digital pheromone swarming algorithms are used to coordinate agents to achieve complex and intelligent behaviors. Studies have shown that pheromone swarming systems are versatile, efficient and resilient to failures, and thus are applicable in various scenarios such as border control, area coverage, target tracking, search and rescue, etc. Due to their reliance on wireless communication channels – which are vulnerable to interference and jamming attacks – it becomes important to study the security of these systems under malicious conditions. In this paper, we investigate the security of pheromone swarming under different types of jamming attacks. In particular, we expose new types of stealthy attacks that aim to maximize the damage inflicted on the swarm while reducing the risk of exposure. Unlike complete Denial of Service (DoS) attacks, the attacks exposed select which signal to interfere with based on the current state of the swarm. We have assessed the impact of the attacks through new metrics that expose the tradeoff between damage and cost. Our results show that the exposed attacks are more potent than traditional DoS-like attacks. Our results are obtained from simulation experiments and real physical implementation using a number of iRobot Create robots in our Mobile Cyber-Physical Systems lab.**

I. INTRODUCTION

In nature, pheromones are a chemical means of communication used by swarming insects (e.g., ants, bees) to organize themselves without relying on any centralized commands. By depositing and detecting pheromone scent signals, insects can coordinate attacks, communicate the location of a food source, and efficiently organize themselves for travel. Swarm intelligence – the characteristics and behavior of swarming insects – has been extensively studied and applied to artificial multi-agent systems [1], [2]. Research areas such as particle swarm optimization and ant colony optimization take advantage of the self-organizing behavior of insects to solve various optimization problems.

In a digital world, virtual pheromones can be used to control unmanned vehicles and autonomous robots. Digital pheromone swarming systems use less intelligent agents and require less manpower than systems with traditional control, but have demonstrated greater adaptability, autonomy, and robustness against failure. Digital pheromones have been shown to be especially useful in the control of unmanned air (UAV) and ground (UGV) vehicles for target tracking, search and rescue missions, border control, among others [3]–[6].

While insects share information by sensing a pheromone flavor, agents in a digital pheromone system communicate virtual pheromones by exchanging messages over a network. Recent advances in wireless technologies have offered convenient communication channels between agents. Their shared mediums, however, present serious challenges due to the possibility of intentional interference by adversaries. It has been shown that a determined jammer can easily bring down the whole system [7]–[10]. This has prompted research in the area of intelligent and reactive jammers that aim to minimize the jamming time to avoid detection [11]–[13]. While anti-jamming techniques (e.g., frequency hopping) can help with non-adversarial noise, wireless systems remain vulnerable to a powerful attacker that can jam a wide range of spectrums.

One of the most recent security incidents occurred in December 2011 when an American Lockheed Martin RQ-170 Sentinel UAV was captured by Iranian forces. It was claimed that the attack had taken place by first jamming signals to the UAV and then by feeding false location information to land the UAV [14]. While this incident involved only a single agent, it becomes important to study the impact of jamming when multiple agents are needed in different scenarios.

In this paper, we study the impact of jamming attacks on the overall operation of the agents in a swarming system. We identify vulnerabilities in digital pheromone swarming systems where an attacker can capitalize on jamming wireless signals between agents (in a distributed system) and/or between an agent and a base station (in a centralized system). We adopt an aggressive attack model in which the adversary knows the current state of the system and can interfere with any signal he/she wishes to. Furthermore, we consider a stealthy attack that jams a minimal number of signals to evade detection, thus prolonging the effect of the attack.

Research Questions: In this paper we seek to answer the following questions:

- How susceptible are pheromone swarming methods to attacks that can select *when* and *which* signal to jam?
- What metrics are valuable measures to assess the damage caused by an attack and the cost of mounting the attack?
- How can we assess whether a digital pheromone system is under attack?

Contributions: This work makes the following contributions:

1) We identify new classes of stealthy attacks that target pheromone swarming methods whereby attacking decisions are based on the current state of the system (e.g., location of the agents and their pheromone maps).
2) We develop new metrics that capture the tradeoffs be-

tween the damage inflicted on the system and the cost incurred in mounting the attack.

3) We assess the impact of the identified attacks through real implementation in various settings with real robots in our Mobile Cyber-Physical Systems lab.

Paper organization: This paper is organized as follows. In Section II we describe related work. Section III covers background material on pheromone swarming methods and identifies the stealthy attacks along with the assessment metrics. In Section IV we present our simulation and implementation results and we conclude the paper in Section V.

II. Related Work

The work in this paper relates to two main areas of research: (1) pheromone swarming methods and (2) security.

Pheromone swarming applications: Pheromone swarming methods have been shown to be effective in applications such as path planning [6], coordination and control of unmanned vehicles [3], [4], [15], maintaining communication in a mobile ad-hoc network [5], among many others. These applications are vital in environments in which it is hazardous or impossible to involve humans directly (e.g., harsh weather conditions, chemical leaks, malicious territories, and disaster impacted sites). For example, in [15] Sauter et al. have demonstrated the use of pheromone swarming algorithms to control a heterogeneous set of air and ground unmanned vehicles. Digital pheromones released from the ground robots are used for the command and control of iRobot Mig 117 Bravo target drones to perform surveillance and target identification. Similarly in [4], algorithms based on digital pheromones are used to control AAI Aerosonde Mk 4.1 UAVs and modified Pioneer 3-AT UGVs in broad area surveillance and base protection scenarios. Most of the work done in this area did not consider the presence of adversaries. However, in many scenarios and in particular military ones, an adversary may exploit a situation to his/her own benefit to inflict further damage.

Security: A deployment of agents is susceptible to various attacks ranging from the physical capture of individual agents to jamming communication between the agents. Due to the shared nature of the wireless channels, jamming has been shown as a very effective technique in disrupting communication whether mounted as a complete Denial of Service (DoS) attack [7]–[9] or in a more intelligent manner through exploiting the networking protocols (e.g., MAC layer) [11]–[13]. While many studies have focused on fault tolerance issues due to random failures (e.g., [16]), the communication aspect has received less attention. In particular, the jamming attacks mentioned above target the networking protocols regardless of the overall application and thus their effect remains unclear on the operation of pheromone swarming systems. In military applications, for example, attacks on communication can be detrimental if UAVs are diverted into enemy territory due to a blocked pheromone map communication or if a surveillance mission is thwarted due to a communication failure about a target area location. In [17], Winfield et al. investigated different hazards that can occur in robot swarms. One of the those hazards is the complete failure of the communication module. This has the effect of the agent wandering off and not participating in the swarm. In this study, we do not consider a complete failure of the communication module, but rather the interference with a subset of the communication. We believe this work is the first to expose stealthy attacks that judiciously decides whether to jam communication signals or not based on the state of the swarm (i.e., location of the agents and their pheromone maps).

III. A General Model

In this section, we cover some background material on pheromone swarming methods. Then we describe our exposed attacks along with our assessment metrics.

A. Background

In digital pheromone swarming methods, pheromones typically have three main characteristics: (1) they can be pumped from a particular location, (2) they evaporate over time, and (3) they propagate to surrounding areas. Pheromones can be represented as different "flavors" to carry different information or instructions. For example, agent movement can be coordinated using two pheromone "flavors"– attractive and repulsive – by having agents move toward attractive pheromones and away from repulsive pheromones. As an agent moves, it deposits repulsive pheromones at its location to repel other agents from the area, resulting in dispersive behavior. The repulsive pheromones evaporate over time, allowing agents to revisit an area once the pheromones have dissipated. A "pheromone map" denotes the agent's view of the pheromone levels in its environment.

Pheromones propogate with a *propogation constant* α and decay with an *evaporation constant d*. The propogated pheromone strength s at a given time t is:

$$s(t) = S(t)/\alpha^x \tag{1}$$

where *S(t)* is the pheromone's current strength at its source location and *x* is the distance from the source location. Since the pheromone evaporates over time, its strength at its source location for a given time *t* is:

$$S(t) = S_0 - t \cdot d \tag{2}$$

where S_0 is the pheromone's initial strength and *t* is the time passed since the pheromone was deposited.

One of the main strengths of pheromone swarming systems is the probabilistic nature of agent movement. Each agent independently decides how to move by computing movement probabilities based on the pheromones in the surrounding environment. In particular, agents have low probabilities of moving into areas with repulsive pheromones and high probabilities of moving into areas with attractive pheromones.

By adjusting the pheromone levels at particular areas in an environment, one can influence the movement probabilities. For example, attractive pheromones can be placed in specific locations to create Areas of Interest (AOIs) that the agents will visit more often than other areas. Similarly, repulsive pheromones can be placed in specific locations to indicate obstacles and barriers the agents need to avoid (e.g., country borders).

For more information about pheromone swarming, we refer the reader to the following references [2], [3].

B. Stealthy Attacks

A jamming attack involves interfering with a wireless signal in order to prevent its reception and can be accomplished using existing devices. While some of these devices are illegal in some countries, many are commercially available.

Adversary Model: We consider an aggressive attack model in which the state of system is assumed to be known to the adversary. The state of the system encompasses the locations of the agents and their pheromone maps. We assume that the adversary can interfere with any signal he/she wishes to based on the state of the agents. Furthermore, we consider a smart adversary that jams the least number of signals to avoid detection. We aim to identify attacks that cause the agents to make inefficient, incorrect, repetitive and/or hazardous movements.

In this work, we discuss four types of jamming attacks:

1) **Complete (C) Attack:** The complete (C) attack is a Denial of Service (DoS) attack in which all the communications between the agents are jammed. Complete attack causes the agents to work independently rather than as a swarm and serves as a basis for comparison.
2) **Half Proximity (HP):** In this attack, outgoing communications to any agents within a specified radius of the broadcasting agent are jammed, but communications to agents outside of the radius are allowed.
3) **Full Proximity (FP):** In this attack, outgoing communications to all agents are jammed if there is at least one agent within a specified radius of the broadcasting agent.
4) **Probability Threshold (PT):** In this attack, jamming of an outgoing communication occurs only if it would result in a difference in pheromone level maps above a certain threshold between the local maps at the agents and the real map that reflects the true pheromone levels (which we denote by *T*).

Figure 1 illustrates which signals are jammed during the two types of stealthy proximity attacks in comparison to the complete attack. Figure 2 shows the criteria for jamming signals during a probability threshold attack in a system implemented with a server.

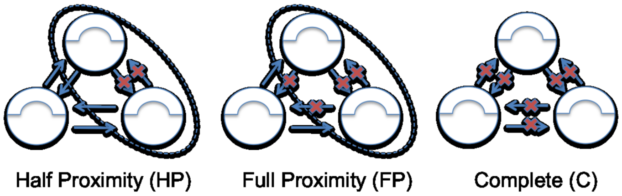


Fig. 1. A simplified representation of proximity and complete attacks for a system of 3 agents. The oval represents the attack radius and the x represents a jammed communication.

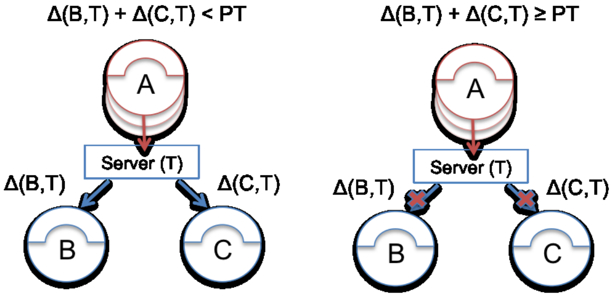


Fig. 2. A probability threshold attack representation for a system of 3 agents. Δ(C,T) represents the difference between an agent's local map (C) and the truth map (T). If the resulting cumulative map differences are above a threshold PT, the signal will be jammed.

The concept behind a PT attack is to only jam signals if it will cause sufficient damage to the system in terms of local map deviations from a truth map *T*. Methods of implementation may vary, but the schematic shown in Figure 2 uses a server to store a truth map of actual pheromones on the map. To predict the damage to the system, the expected local pheromone map X_i is computed for each agent i under the assumption that the signal is not received. For each map X_i, a value $\Delta(X_i$,T) is computed to represent what the deviation of each local map will be if the signal is jammed. The sum of expected map deviations Σ_i $\Delta(X_i$,T) is compared to predecided threshold "PT". If Σ_i $\Delta(X_i$,T) $<$ PT, the expected damage is below the acceptable damage threshold and the signal should not be jammed. If Σ_i $\Delta(X_i$,T) $\geq$ PT, the expected damage meets or exceeds the threshold and the signal should be jammed.

C. Assessment Metrics

To assess the impact of attacks, capturing their stealthy nature, we follow the definition of *attack potency*, π, proposed in [18], whereby an attack's potency is the ratio of the *damage* caused to the *cost* incurred in mounting the attack.

$$\pi = \frac{\text{Damage}}{\text{Cost}} \tag{3}$$

Damage and cost in Equation 3 can be instantiated based on different metrics depending on the application studied. In this paper, we focus on three forms of damage: (1) collisions, (2) pheromone map differences and (3) the number of steps the system fails to complete before the first collision compared to no attack. A collision occurs when multiple agents attempt to occupy the same grid location at the same time. Pheromone map differences capture the difference in movement probabilities between the local maps at the agents – which may reflect outdated information due to the absence of new signals – and the real pheromone map. The idea is to cause the agents to deviate from their normal swarming behavior by repetitive and inefficient coverage. The motivation behind these instantiations of damage is to challenge the system by causing unnecessary movements, disproportionate coverage of the areas of interest, and/or potential expensive repairs due to collision. We instantiate the cost of the attack as the number of signals jammed by the adversary. This notion of cost captures the exposure risk the attacker is willing to take. We are interested in identifying attacks that cause damage with the least amount of cost. With our instantiation of cost, the potency reflects the damage per signal attacked.

IV. Experimental Results

In this section, we present results from our simulation experiments performed in two general setups and from our implementation experiments using the iRobot Create robots.

A. Experimental Design

For both our simulated and physical experiments we designate attractive pheromones as positive integers, repulsive pheromones as negative integers, and the absolute value of an integer as the strength of repulsion or attraction. The physical area where the agents are located is divided into a grid. Depositing a pheromone in a grid cell is handled virtually by adding the new pheromone's integer value to the grid cell's existing value. Each agent stores a list of the pheromone locations and levels on the grid and can broadcast this list to other agents.

As an agent moves, it deposits a repulsive pheromone at its old grid location and sends its pheromone list to a server (Figure 3). The server acts as an aggregation point which maintains a truth map of all incoming pheromone data, and disseminates the pheromone lists received to other agents on the grid. The implementation of the server allows us to interrupt communications between the agents without having to physically jam wireless signals and assess the damage. In order to decide each move, an agent first computes how each pheromone in its pheromone list propogates across the grid. A pheromone is at full strength at its source location and decreases according to Equation 1 radially outward from the source until the strength value reaches 0. The agent sums the propogated pheromone strength values for each grid cell to determine a cumulative pheromone strength gradient.

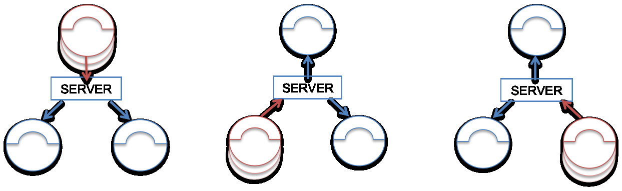


Fig. 3. A map communication schematic for a system of 3 agents. After moving one step, an agent will communicate its pheromone list to the server, which disseminates the list to every other agent in the system.

After computing the pheromone strength gradient, the agent probabilistically decides which cell to move to. Agents are able to move in four directions: left, right, forward, and backward. The *propagated pheromone effect* e_i is calculated for each of the four possible directions of movement based on the cumulative pheromone strength s in the destination cell:

$$e_i = B^s \tag{4}$$

where B is a positive scaling constant. Although the strength value s may be negative, e_i is a scaled value and will always be positive. The probability p_i that an agent will move to a particular adjacent cell is based on the cell's pheromone effect e_i weighted against the pheromone effect of the other adjacent cells:

$$p_i = \frac{e_i}{E} \tag{5}$$

where E is the sum of e_i values for all cells the agent could potentially move to.

To ensure that our implementation exhibited the adaptive and decentralized qualities of a swarming system, we set behavior goals and observed the actual swarming behavior of the agents. For a grid with no AOIs, agents must cover the grid evenly. For a grid with AOIs, agents must cover all targets proportionally according to the attractive pheromone strengths. The desired grid coverage must be observed regardless of the number of agents or targets, and can only be controlled by changing pheromone strengths or evaporation rates.

We performed a series of characterization tests to examine swarming behavior as we varied the negative pheromone strength from -20 to -70, the positive pheromone strength from 20 to 220 and the evaporation constant from 1 to 10. We selected a propogation constant α = 2, evaporation constant d = 4 for simulation and d = 1 for the physical implemenation, and negative pheromone strength -44 so that in a 10x10 grid any given negative pheromone will initially affect a large portion of the grid but completely dissipate within 11 steps. These values were chosen because they resulted in the most even grid coverage. We selected a postive pheromone strength of 70 because this strength resulted in a 3:1 visitation ratio of AOI to non-AOI cells on the grid while still maintaining even grid coverage for non-AOI areas.

B. Simulation Results – Setup A

For setup A, we experimented with all four types of attacks on a 10x10 grid. Configurations are represented as *R*T, where R is the number of agents/robots in the system and T is the number of targets/AOIs (e.g., 3R0T is a system of 3 robots and no targets). Each experiment runs for 1000 steps. Based on our initial results, we concluded that HP attacks were not as potent as the other attacks due to the redundancy of the aggregation point-based communication [19]. Agents outside of the attack radius still receive communications and pass them on to all other agents, so any map differences would correct themselves after only one step rather than compound upon themselves the longer the robots were within the radii. Thus, we do not present results for HP.

Our focus of damage metrics are collisions and map differences. Map differences constitute a very basic measure of deviation from ideal movement without specifying an exact type of damage that actually occurs. Map differences capture inefficiencies due to redundant work being done by the agents as well as additional power consumption. Collisions, however, are a specific measurement of a type of damage that can occur.

Figure 4 shows the cost of each type of attack computed as the number of jammed signals. In all the configurations we studied, the C attack had the highest cost. For a 3R0T

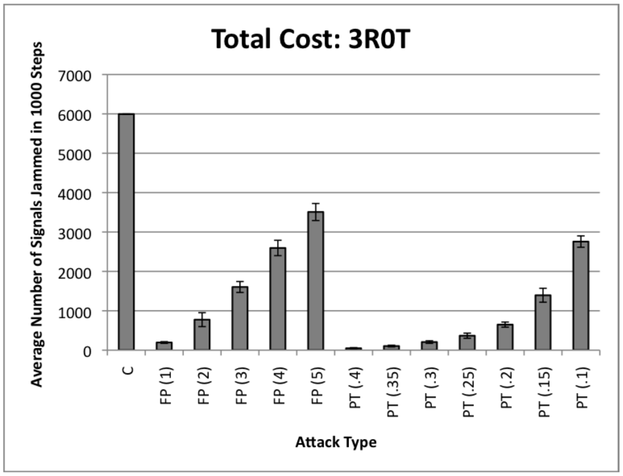


Fig. 4. Average number of signals jammed over 10 independent runs for complete (C), full proximity (FP) and probability threshold (PT) attacks. The FP attack radius varied from 1 to 5. The PT attack threshold varied from 0.4 to 0.1.

configuration, the FP attack ranged from around 10% to 50% of the cost of the C attack as the attack radius increased and the PT attack ranged from around 5% to 48% of the cost of the C attack as the probability threshold decreased. Results are averaged over 10 independent runs and the standard deviation is indicated by the small error bars.

Figure 5 presents results from a 3R0T configuration under different attack types. The number next to the attack type indicates the radii for FP and threshold for PT. Figure 5 (a) shows the potency measured as the number of collisions per signal jammed. The FP attack was 1.5x to 13.5x more potent than the C attack depending on the radius of attack chosen. The PT attack was 1.8x to 14.7x more potent than the C attack depending on the probability threshold chosen. In terms of collisions, every choice of radius and threshold resulted in a higher potency than the C attack, and was significantly higher for smaller radii and larger thresholds. Figure 5 (b) shows the potency measured as the difference in pheromone maps per signal jammed. The FP attack was 1.3x to 1.4x more potent than the C attack, depending on the attack radius chosen. Of the seven probability thresholds tested in the PT attack, the five highest thresholds were 1.1x to 1.4x more potent than the C attack. The two lowest thresholds, 0.1 and 0.15, were less potent.

Figure 5 (c) shows the number of collisions that occurred over 1000 steps. With increasing attack radii, the FP attack caused from 45% up to 99% of the collisions a C attack caused. With decreasing probability thresholds, the PT attack caused from 10% up to 87% of the collisions a C attack caused. Figure 5 (d) shows the map differences over 1000 steps. The FP attack caused from 4% (at the lowest attack radius) to 77% (at the highest attack radius) of the map differences caused by the C attack. The PT attack caused from 1% (at the highest probability threshold) to 38% (at the lowest probability threshold) of the map differences caused by the C attack.

Figure 6 shows results from varying the number of robots and targets in a configuration. Adding robots and targets did not notably affect the potency of the attacks.

Figure 7 shows the coverage of different grid cells by agents when a target is placed at location (8,8). Interestingly, the target is visited more often while under attack than when no attack is present (shown by the solid line). This occurs because when agents fail to communicate, they remain aware of the attractive pheromones present in the original grid but become less aware of repulsive pheromones deposited by other agents. Thus, jamming attacks can only cause robots to visit targets more often than they naturally would have, and never less often. This, however, causes inefficiencies since duplicate work is done by the agents.

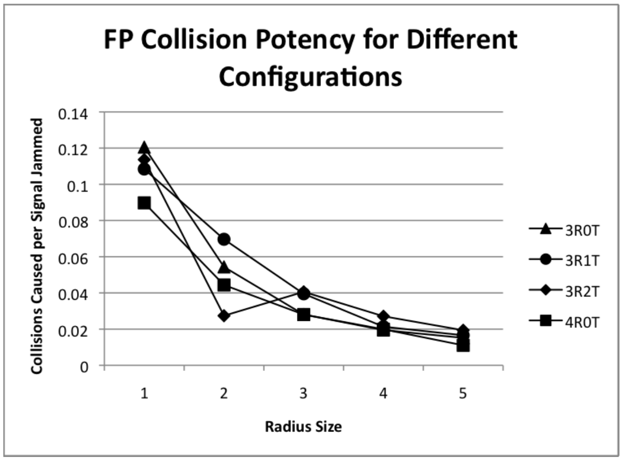


Fig. 6. Potency (collisions per signal jammed) for FP attacks in 3R0T, 3R1T, 3R2T, and 4R0T configurations.

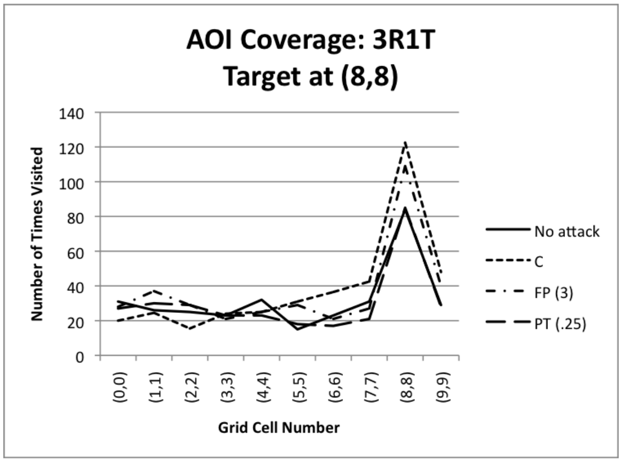
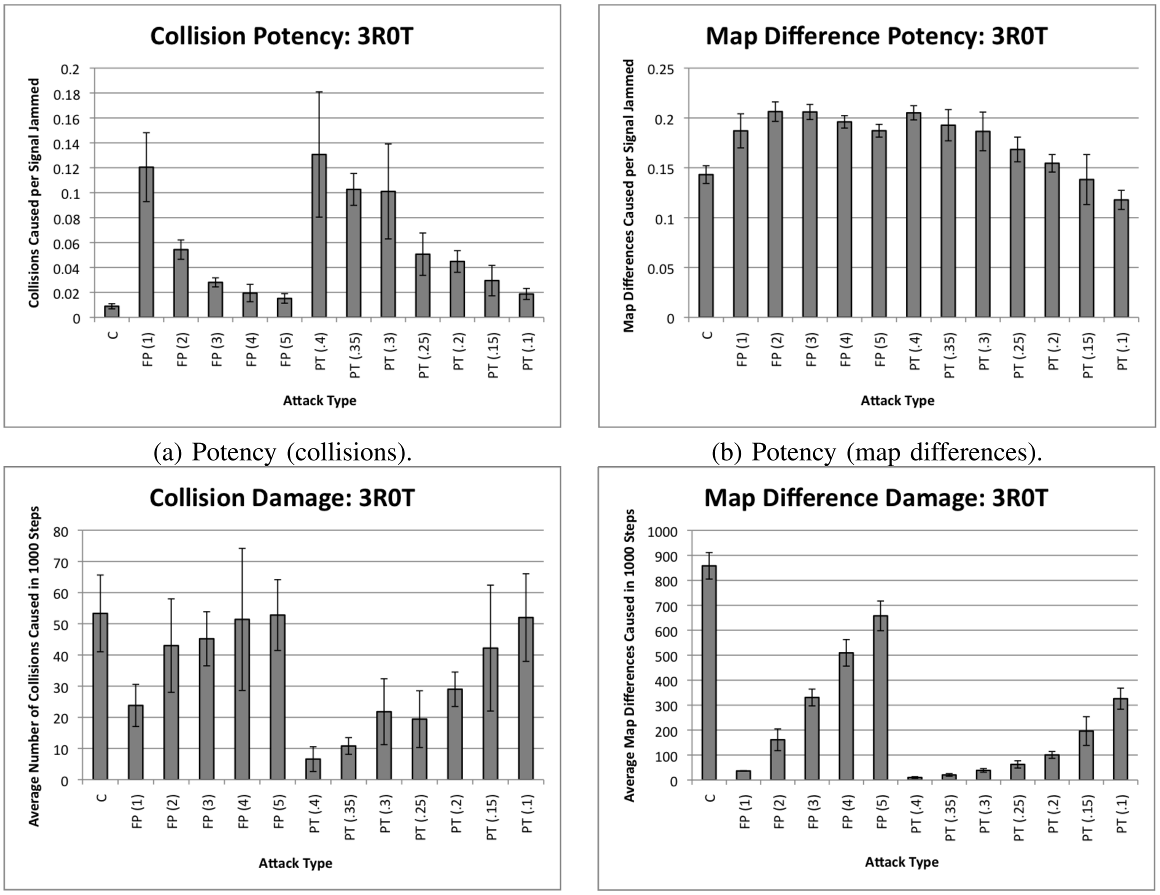


Fig. 7. Frequency of visits to various locations on a 10x10 grid for no attack (N), C, FP with a radius of 3, and PT with a threshold of 0.25.

C. Simulation Results – Setup B

Setup B is composed of 10 agents on a 15x15 grid. We vary the number of targets from 0 (10R0T) up to 20 (10R20T). Targets are placed at random on the grid with no more than one target per grid cell. We present results comparing FP and PT to C and random attacks. In these experiments, we perform 100,000 steps.



(a) Potency (collisions).

(b) Potency (map differences).

(c) Damage (collisions).

(d) Damage (map differences).

Fig. 5. Potency (top row) and Damage (bottom row) for a system with 3 agents and no targets under different attack policies.

Figure 8 shows the potency, measured as the number of collisions per signal jammed, for various configurations under different types of attacks. One can see that there is always one of the identified attacks that is more potent than the C attack. Moreover, FP and PT are more effective in systems with fewer targets. As the number of targets increases, the impact of the attacks seems to level off.

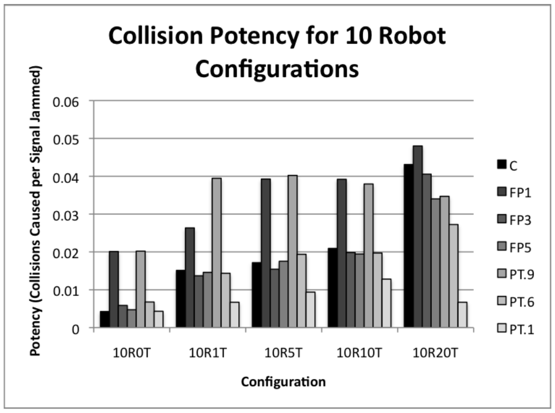


Fig. 8. Potency (collisions per signal jammed) for various configurations with 10 robots.

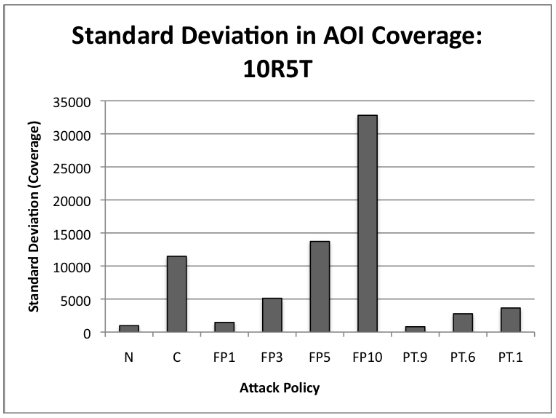


Fig. 9. Standard deviation in coverage for the target location with 10 agents and 5 targets under different attack policies.

Figure 9 reflects another aspect that is not captured by the potency metric. In this figure we show the standard deviation in the number of visits to the target locations using configuration 10R5T. Ideally, all targets should be visited equally. Notice that all attacks cause more variance when compared to no attack (N). When comparing this figure with the potency metric, one can see that attacks that do not necessarily cause high potency, tend to cause more variance. This dimension of damage is important to capture since it reflects the degree of unbalance in coverage.

To determine whether the state-based nature of our stealthy attacks influenced effectiveness, we compared the stealthy attacks to a random attack that does not consider the state of the system. A random attack jams a specified percentage of signals, but selects *which* signals to jam at random. Figures 10 and 11 show the collisions and map differences caused, respectively, by the stealthy and random attacks in a system of 10 robots. The results show that our state-based attacks cause more damage than random attacks for similar costs incurred.

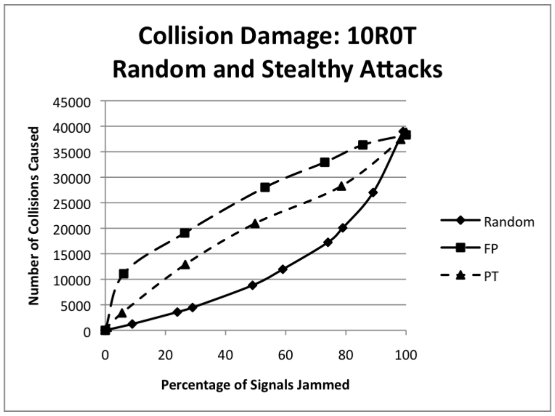


Fig. 10. Collision damage for a system with 10 agents and no targets under different attack policies.

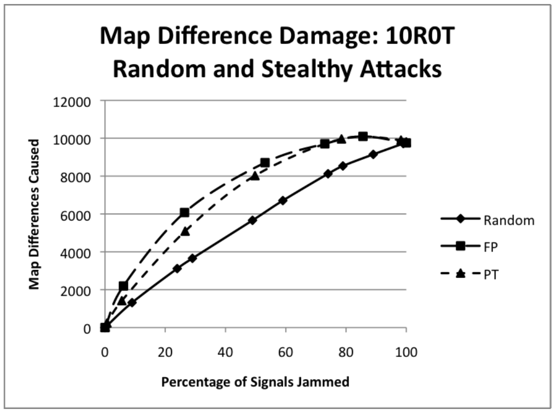


Fig. 11. Map differences for a system with 10 agents and no targets under different attack policies.

D. Implementation Results

While simulation results allowed us to test a wide variety of configurations, a physical implementation is necessary to observe any inconsistencies in behaviors due to the asynchronous nature of the wireless communications between the agents and the server. Since there is no guaranteed order of communication, the server may receive multiple updates from one robot, but only one update from another. In addition, the amount of time it takes to move in each of the four possible directions varies. It takes twice as long to turn an agent 180 degrees as it does to turn the agent 90 degrees. As a result, it is important to determine whether timing inconsistencies and asynchronous movements are detrimental to the performance of stealthy attacks.

Our physical experiments are carried out using netbooks mounted on iRobot Create robots on a 4x6 grid. Movement around the grid is observed for up to 100 steps under four attack types and two grid configurations. The four attack types are no-attack, complete attack, full proximity attack with radius 1 and probability threshold attack with threshold 0.2. The two grid configurations are no target and one target at grid position (3,5). Eight independent experiments were performed per scenario for each of the two grid configurations and the results presented are an average over the eight experiments.

When not under attack, no collisions were observed for the entire 100 steps both with and without a target. Under attack types, collisions were observed as soon as within 4 steps and as late as 53 steps without a target. With a target, collisions were observed within 3 to 95 steps. Since no attack resulted in no damage and required no cost, it is used as a basis of comparison. Damage is measured as the number of steps the system fails to complete compared to the no-attack scenario. For example, if the first collision occurs after 30 steps while under attack and 100 steps were completed while not under attack, the damage is 70 failed steps. The cost is measured as the number of signals jammed in order to cause the first collision. Attack potency, shown in Figure 12, is the number of failed steps per signal jammed.

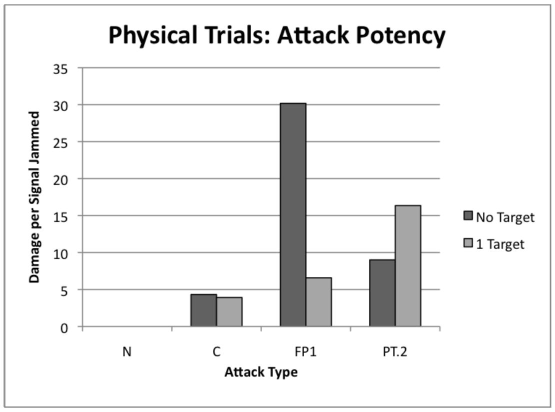


Fig. 12. Average attack potency measured as the number of steps until the first collision divided by the signals jammed leading up to the first collision.

Of the three attack types tested on a target-less grid, FP caused the earliest collisions and jammed the fewest signals. On average, C caused a collision after 23 steps by jamming 46 signals, FP after 14 steps by jamming 3 signals and PT after 22 steps by jamming 14 signals. FP causes greater damage, incurs a lower cost and is more potent than C. PT does not cause greater damage than C, but it does incur a lower cost and as a result is still more potent than C.

When a target is present on the grid, the movement probabilities of each robot are altered according to the permanent positive pheromone gradient and grid coverage is no longer even. Compared to the potency on a grid without targets, the attack potency of C decreased 9%, the potency of FP decreased 78% and the potency of PT increased 81% on a grid with one target. As shown in Figure 12, both FP and PT were still more potent than C when a target was added to the grid.

Grid coverage is an interesting damage metric because simple swarming systems may not have collision avoidance or damage recovery strategies in place. If a collision is catastrophic, early collisions mean failure to complete even the most basic mission (e.g., exploring a grid entirely). When not under attack, the target-less grid in our implementation was completely covered within 32 steps on average. When under attack, "early" collisions were observed in 75% of C trials, 100% of FP trials and 63% of PT trials.

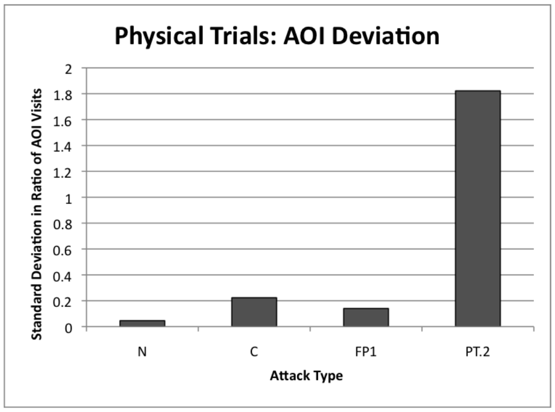


Fig. 13. Standard deviation in coverage ratio of the target location.

Figure 13 shows the standard deviation in the target coverage ratio over the eight runs. The target coverage ratio is measured as the number of times the target is visited compared to the total number of steps completed before the first collision. Under no attack the deviation is small, suggesting that the target is visited consistently and proportionately. Under attack conditions, more variance in target coverage suggests unpredictable coverage of the target.

V. Conclusions

In this paper, we identified multiple jamming attacks that were significantly more effective at causing damage to digital pheromone swarming systems than traditional DoS attacks. These attacks judiciously decide which signal to interfere with based on the state of the agents. We have shown the potency level in terms of collision damage can be 14 times that of DoS attacks. We have also assessed the impact of the attacks on our implementation test-bed and found that the potency level of the stealthy attacks in terms of time until the first collision can be 7.5 times that of DoS attacks. We have identified a number of key parameters that affect the attack potency such as the radius of attack, probability threshold, and the number of agents and targets present. In all the cases studied, there was always a stealthy attack that was more potent than DoS attacks.

Choosing an effective attack type depends on the scenario and the type of detection mechanisms employed in the system. If an attack's effect is detected according to an increase in the number of collisions, a jamming attack similar to PT should be chosen since it can cause a wide range of map differences with a small number of collisions. If an attack's effect is detected by increased map differences, a jamming attack like FP should be used since it causes a greater number of collisions within a narrow range of map differences. The impact of damage type can also be considered. If a collision would be catastrophic, such as for a system of UAVs, a jamming attack with a high collision potency can be chosen. If the system is protected with collision recovery techniques, however, an attack that targets coverage may be selected.

Acknowledgment

This research is funded by NSF REU award #1156712 that is co-funded by the DoD and by NSF award #1149397.

References

[1] H. Parunak, "Go to the Ant: Engineering Principles from Natural Multi-agent Systems," *Annals of Operations Research*, vol. 75, pp. 69–102, 1997.

[2] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, USA, 1999.

[3] J. Sauter, R. Matthews, H. Parunak, and S. Brueckner, "Performance of Digital Pheromones for Swarming Vehicle Control," in *Proceedings of the Fourth International joint Conference on Autonomous Agents and Multiagent Systems*. ACM, 2005, pp. 903–910.

[4] J. Sauter, R. Matthews, J. Robinson, J. Moody, and S. Riddle, "Swarming Unmanned Air and Ground Systems for Surveillance and Base Protection," in *Proceedings of AIAA Infotech Aerospace Conference*, 2009.

[5] H. Parunak, S. Brueckner, and J. Odell, "Swarming Coordination of Multiple UAV's for Collaborative Sensing," in *Proceedings of Second AIAA" Unmanned Unlimited" Systems, Technologies, and Operations Conference*, 2003.

[6] H. Parunak, M. Purcell, and R. OConnell, "Digital Pheromones for Autonomous Coordination of Swarming UAVs," *American Institute of Aeronautics and Astronautics*, vol. 1001, pp. 48105–1579, 2002.

[7] W. Xu, K. Ma, T. Wade, and Y. Zhange, "Jamming Sensor Networks: Attacks and Defense Strategies," *IEEE Network*, 2006.

[8] L. Sang and A. Arora, "Capabilities of Low-power Wireless Jammers," in *Proceedings of INFOCOM*, Rio de Janeiro, Brazil, April 2009.

[9] K. Pelechrinis, M. Iliofotou, and V. Krishnamurthy, "Denial of Service Attacks in Wireless Networks: The Case of Jammers," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 2, 2011.

[10] Y.W. Law, M. Palaniswami, L.V. Hoesel, J. Doumen, P. Hartel, and P. Havinga, "Energy-Efficient Link-Layer Jamming Attacks gainst Wireless Sensor Network MAC Protocols," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 1, pp. 1–38, 2009.

[11] D. Thuente and M. Acharya, "Intelligent Jamming in Wireless Networks with Applications to 802.11b and other Networks," in *Proceedings of IEEE MILCOM*, Washington, DC, October 2006.

[12] M. Acharya, T. Sharma, D. Thuente, and D. Sizemore, "Intelligent Jamming in 802.11 b Wireless Networks," in *Proceedings of OPNETWORK*, Washington, D.C, 2004.

[13] M. Wilhelm, I. Martinovic, J. Schmitt, and V. Lenders, "Short Paper: Reactive Jamming in Wireless Networks: How Realistic is the Threat?," in *Proceedings of the fourth ACM conference on Wireless network security*, 2011.

[14] Wikipedia, "IranU.S. RQ-170 Incident," `http://en.wikipedia.org/wiki/Iran%E2%80%93U.S._RQ-170_incident`.

[15] J. Sauter, R. Matthews, H. Parunak, and S. Brueckner, "Demonstration of Digital Pheromone Swarming Control of Multiple Unmanned Air Vehicles," *Proceedings of AAAI Infotech@ Aerospace*, 2005.

[16] K Dailey, "The FMEA Handbook," 2004.

[17] A. Winfield and J. Nembrini, "Safety in Numbers: Fault-tolerance in Robot Swarms," *International Journal of Modelling, Identification and Control*, vol. 1, no. 1, pp. 30–37, 2006.

[18] M. Guirguis, A. Bestavros, and I. Matta, "Exploiting the Transients of Adaptation for RoQ Attacks on Internet Resources," in *Proceedings of the ICNP*, Oct 2004.

[19] K. Xing, S.S.R. Srinivasan, M.J.M. Rivera, J. Li, and X. Cheng, "Attacks and Countermeasures in Sensor Networks: A Survey," *Network Security*, pp. 251–272, 2010.