

MENG INDIVIDUAL PROJECT

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

There is an Impostor Among Us:
Defending Robot Swarms from Bad Actors

Author:
Akshat Tripathi

Supervisor:
Prof. Andrew Davison

Second Marker:
Prof. Antoine Cully

June 3, 2023

TODO

Abstract

TODO

Acknowledgements

Contents

1	Introduction	5
1.1	Objectives	5
1.2	Challenges	5
1.3	Contributions	5
2	State of the Art	6
2.1	Multi-Robot Systems	6
2.1.1	Competitive vs Collaborative Behavior	6
2.1.2	Static vs Dynamic Coordination	6
2.1.3	Explicit vs Implicit Communication	6
2.1.4	Homogeneous vs Heterogeneous Robots	6
2.1.5	Centralised vs Decentralised Decision Making	7
2.2	Robot Web	7
2.2.1	Factor Graphs	7
2.2.2	Belief Propagation	9
2.2.3	Gaussian Belief Propagation	9
2.2.4	Lie Theory	9
2.2.5	Putting it all together	9
2.2.6	Evaluation	10
2.3	Security Issues	11
2.3.1	Denial of Service	11
2.3.2	Identity-Based Attacks	11
2.3.3	Physical Attacks	12
2.4	Wifi Fingerprinting	12
2.4.1	Guaranteeing spoof-resilient multi-robot networks	13
2.4.2	Lightweight Sybil-Resilient Multi-Robot Networks by Multipath Manipulation	14
2.4.3	Conclusions	14
2.5	Proof of Work	15
3	An Investigation into the Security of RobotWeb	16
3.1	The behaviour of an ideal attacker	16
3.2	A simplified system	16
3.3	Theoretical properties of the RobotWeb under attack	18
3.3.1	Measuring the strength of an attack	18
3.3.2	Crafting the perfect message	19
3.3.3	Scaling back up to n-dimensional space	20
3.3.4	The Force-Based model of attacks	20
3.4	Hypotheses	20
3.4.1	There is Strength in Numbers	21
3.4.2	Long Memories are Detrimental	21
3.4.3	Attacks are Contagious	21
3.4.4	Young Robots are the most Vulnerable	22
3.4.5	Bounds on μ can be slowly escaped	22
3.4.6	Bounds on Λ can be quickly escaped	22
3.5	Experimental evidence	23
3.5.1	Experimental setup	23

3.5.2	Testing hypothesis 1	24
3.5.3	Testing hypothesis 2	25
3.5.4	Testing hypothesis 3	25
3.5.5	Testing hypothesis 4	26
3.5.6	Testing hypothesis 5	27
3.5.7	Testing hypothesis 6	27
4	Defence against the Dark Arts	29
4.1	The Limits of Individualism	29
4.2	A Group-Based Defence	30
4.2.1	Inductive proof	30
4.2.2	Experimental confirmation	31
4.2.3	Strategies to improve performance	31
4.2.4	Exits, Mergers and Acquisitions	31
5	Conclusion	32
5.1	Ethical Considerations	32
A	First Appendix	33
	Bibliography	35

List of Figures

2.1	Example factor graph	7
2.2	Robot Web factor graph	10
2.3	RobotWeb’s robustness to garbage measurements	10
3.1	Single Variable in a Factor Graph	17
3.2	Representative factor graph around a single variable	17
3.3	The negligible effect of using time-windowed history	21
3.4	Paths used to evaluate attackers	24
3.5	The “strength in numbers” effect: as the number of robots increases, the effectiveness of an attack decreases. Note the outliers in 3.5(a) occur as the weakened attacker is unable to instantly seize control of its victims.	24
3.6	The effect of History Size on ATE: a longer history size is detrimental when attackers are present, yet provides little value otherwise.	25
3.7	The contagion effect: attacked robots slowly spread their attack out to their neighbours	26
3.8	The effect of a startup delay on the ATE, also investigating the effect of the history on this.	26
3.9	Attackers escaping a 5cm μ bound: the 60cm difference between the actual and target path is bridged by n <i>onramp segments</i> of equal lengths	27
3.10	A Sybil attack on the RobotWeb: as the number of Sybils increases, so does the influence of the attacker upon the rest of the RobotWeb	28

Chapter 1

Introduction

1.1 Objectives

The field of distributed robotics is a growing one, partly due to the growth of the number of applications involving it (autonomous vehicles and drone delivery systems) and partly because of increased interest in areas which would greatly benefit from it, such as the Lunar Gateway Project, or the Mars 2020 Perseverance Mission.

An open problem in this distributed robotics is that of effective inter-robot communication. Inter-robot communication provides several benefits to distributed robotics; it allows robots to 1. acquire a more accurate view of their environment, 2. gain access to a larger map of the world, 3. improve their path planning by incorporating others' plans and 4. carry fewer resources, such as sensors and instead rely on others in the swarm.

A subset of distributed robotics research is dedicated to finding ways to allow such communication in situations where some of the robots may act with hostility. The source of this hostility could be unnatural, where a bad actor solely seeks to disrupt the system, or it could be a natural consequence of competition in the environment, such as a self-driving car trying to prevent others from overtaking it.

This hostility must be protected against if the benefits of communication and collaboration are to be seized. This is the principal focus of this Master's Thesis.

1.2 Challenges

1.3 Contributions

Currently, the main contributions of this paper can be found in the background research section and some of the preliminary results on the effectiveness of attacks on robot networks. This is subject to change with time.

Chapter 2

State of the Art

The first half of this chapter will provide the theoretical background for this thesis. First, we will discuss the field of multi-robot systems, providing the reader with an understanding of how the field has evolved and how it may further evolve. Next, we examine RobotWeb [1], the research that this thesis seeks to build upon. Finally, we discuss various security issues present in the field to arrive at the research question for this thesis.

2.1 Multi-Robot Systems

The study of multi-robot systems concerns itself with studying how to allow multiple robots to operate in the same environment [2]. Multi-robot systems have several advantages over single-robot systems; they are more effective, efficient, flexible, and resilient [3]. These robots can behave competitively or collaboratively, coordinate statically or dynamically, communicate explicitly or implicitly, consist of homogeneous or heterogeneous robots, and make decisions centrally or decentrally [4].

2.1.1 Competitive vs Collaborative Behavior

Multiple robots which share a common goal are considered to be behaving collaboratively, whereas if each robot aimed to complete its own goal at the expense of all others, it would be said to be behaving competitively [4]. Examples of collaboration range from teams of robots constructing a lunar habitat [5] to exploring unknown environments [6].

2.1.2 Static vs Dynamic Coordination

If a multi-robot system operates using a set of predetermined rules, then it can be said to be coordinating itself statically. A possible set of rules would be that each robot must maintain a certain distance between it and all others. Dynamically coordinated multi-robot systems would instead make decisions whilst performing the task and may communicate to do so [4].

2.1.3 Explicit vs Implicit Communication

Most multi-robot systems communicate explicitly by sending messages to each other via a hardware communication interface, for example, a wifi module [4]. However, there is still a sizeable minority of approaches that send messages through their environment (implicit communication) and rely on others to sense these messages to receive them. An example of implicit communication is found in [7], where the authors use it to allow a team of robots to play a game of football for the RoboCup Simulation League [8].

2.1.4 Homogeneous vs Heterogeneous Robots

Multi-robot systems can either contain robots with identical hardware, which are known as homogeneous systems, or individual robots may have different hardware, making them heterogeneous

systems. Heterogeneous systems allow a greater degree of specialisation within a multi-robot system but also add additional decision-making complexity.

2.1.5 Centralised vs Decentralised Decision Making

A multi-robot system is said to have centralised decision-making if all robots communicate with a central agent, which may or may not be a robot itself, to receive instructions. Centralised schemes perform better with smaller groups of robots and in static environments, they also introduce a single point of failure in the central agent [4]. Decentralised schemes, however, avoid vesting authority into a central agent and instead treat each agent as an equal part of the system, which allows them to avoid the problems associated with centralised schemes. However, decentralised schemes lose the guarantee that they will converge to an optimal solution, as decisions are made with incomplete information. In addition to centralised and decentralised schemes, multi-robot systems may also be organised in a hierarchical manner, where some robots would be chosen as local leaders, but no global leader would exist.

2.2 Robot Web

This thesis seeks to build upon the work done in “A Robot Web for Distributed Many-Device Localisation” [1], which describes a method for *heterogeneous* robots in a *decentralised* multi-robot system to *collaborate* via *explicit communication* to localise *dynamically*.

Robots in the Robot Web move along predefined paths, estimating their location via internal odometry. When a robot senses another, it communicates its measurement to the other robot, and then both robots use the measurement to update their location estimates. Since we live in an imperfect world, each sensor measurement carries with it a small amount of noise, which is reflected in the Robot Web by a degree of uncertainty attached to each robot’s location estimate and represented by a Gaussian distribution.

This section will introduce the reader to the core concepts used in the Robot Web and assemble them to provide the reader with an understanding of how the Robot Web functions and some of its limitations.

2.2.1 Factor Graphs

A factor graph is an undirected bipartite graph used to represent the factorisation of a probability distribution $p(X)$. A probability distribution can be said to be factorised if it is written in the form:

$$p(X) = \prod_i f_i(X_i) \quad (2.1)$$

The nodes of a factor graph can either represent variables (X_i) or factors (f_i). There are several different ways to draw factor graphs, but we will use the one defined in [9], where factors are drawn as squares and variables are drawn as circles.

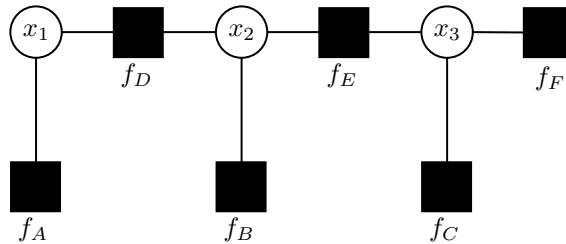


Figure 2.1: An example of a factor graph

The above factor graph represents the following factorisation:

$$p(X_1, X_2, X_3) = f_A(X_1)f_B(X_2)f_C(X_3)f_D(X_1, X_2)f_E(X_2, X_3)f_F(X_3) \quad (2.2)$$

Assuming that each variable takes discrete values, suppose we wanted to find the probability that $X_1 = z$ for some value of z using the above factor graph. Then we would need to find:

$$p(X_1 = z, X_2, X_3) = \sum_{i=X_2} \sum_{j=X_3} p(X_1 = z, X_2 = i, X_3 = j) \quad (2.3)$$

And by 2.2 we get:

$$p(X_1 = z, X_2, X_3) = \sum_{i=X_2} \sum_{j=X_3} f_A(z)f_B(i)f_C(j)f_D(z, i)f_E(z, j)f_F(j) \quad (2.4)$$

which can be rearranged to form:

$$p(X_1 = z, X_2, X_3) = f_A(z) \sum_{i=X_2} \left(f_D(z, i)f_B(i) \left(\sum_{j=X_3} f_E(z, j)f_C(j)f_F(j) \right) \right) \quad (2.5)$$

Similarly, if we wanted to find the probability that $X_2 = z$ for some z we would need to find:

$$p(X_1, X_2 = z, X_3) = f_b(z) \left(\sum_{i=X_1} f_D(i, z)f_A(i) \right) \left(\sum_{j=X_3} f_E(z, j)f_C(j)f_F(j) \right) \quad (2.6)$$

Noticing how the sum over X_3 in both 2.5 and 2.6 is the same, we may want to “cache” the result when dealing with large factor graphs, to improve performance. To do this we can associate calculations with nodes in the factor graph. We call these associations “messages”.

The general form of a message from variable i to factor j is the product of the messages from all other neighbouring factors [10]. Put formally:

$$m_{x_i \rightarrow f_j} = \prod_{s \in N(i) \setminus j} m_{f_s \rightarrow x_i} \quad (2.7)$$

The general form of a message from factor j to variable i is the product of the messages from all other neighbouring variables and the factor applied to all other variables except i [10]. Put formally:

$$m_{f_j \rightarrow x_i} = \left(\sum_{X_j \setminus x_i} f_j(X_j) \right) \left(\prod_{k \in N(j) \setminus i} m_{x_k \rightarrow f_j} \right) \quad (2.8)$$

Finally, the marginal value of a variable is simply the product of all incoming messages to it [10].

$$p(x_i) = \prod_{s \in N(i)} m_{f_s \rightarrow x_i} \quad (2.9)$$

2.2.2 Belief Propagation

The above equations are used by the Belief Propagation algorithm, an iterative message-passing algorithm used to calculate the marginal value for each variable in a factor graph [10]. Each iteration of Belief Propagation has 3 phases:

1. Variables send messages to each of their neighbouring factors 2.7.
2. Factors send messages to each of their neighbouring variables 2.8.
3. Each variable updates its “belief” (its estimated marginal value) 2.9.

The original Belief Propagation algorithm was designed to be used in tree-like graphs, i.e. graphs without loops [10]. However, empirical evidence has shown that “Loopy-BP” can still converge to provide useful results in a variety of problem domains [10].

2.2.3 Gaussian Belief Propagation

A special case of the Belief Propagation algorithm is Gaussian Belief Propagation, which applies to problems where all variables follow a Gaussian distribution, and all factors are Gaussian functions of their inputs.

Under Gaussian Belief Propagation, each message can be interpreted as a Gaussian and so must contain sufficient information to produce one. A naive way of achieving this is to include a mean vector and a covariance matrix in each message. However, this approach is computationally expensive as it requires a full matrix multiplication whenever messages are multiplied which is an order $O(n^3)$ operation. An alternative approach is to use the *canonical form* of the multivariate Gaussian distribution.

The canonical form uses an *information vector* (η) and a *precision matrix* (Λ) defined as follows:

$$\eta = \Sigma^{-1}\mu \qquad \Lambda = \Sigma^{-1}$$

where Σ is the covariance matrix and μ is the mean vector. Now multiplying messages is made more efficient as it only requires the addition of both messages’ η and Λ values, making it an order $O(n^2)$ operation in the worst case. A further performance improvement can be made by recognising that the precision matrix is a sparse matrix [10].

2.2.4 Lie Theory

Lie theory is a subset of group theory focussed on studying *Lie groups*. Lie theory is a vast and abstract field, from which we only need to borrow a few concepts. The first is that positions and rotations can be represented as Lie groups, for example, the group $SO2$ represents a rotation in 2D space and the $SE3$ group represents a rigid motion in 3D space. The second core concept is the *tangent space* which allows small deviations to be applied to the Lie group uniformly regardless of the value it operates on. This concludes our whirlwind tour of Lie theory, we invite the reader to read [11] for a more detailed tutorial.

2.2.5 Putting it all together

Now that we have covered all of the prerequisites to understanding how the Robot Web operates, we shall now demonstrate how they can be assembled into the Robot Web.

Every robot in the Robot Web needs to estimate its current location at all times, this is called localisation. One simple localisation method is to use odometry, which uses internal sensors to measure its displacement from its previous location. Since no sensor is perfect, this introduces a small amount of noise, which can be accurately modelled using a Gaussian distribution. The Robot Web simulates odometry using a factor graph, each known position of the robot maps to a pose variable, and the variables of each pair of successive positions are connected by an odometry factor.

On every timestep, the robot performs an iteration of Gaussian Belief Propagation to estimate its current position.

The Robot Web further improves the accuracy of robots' locations by allowing robots to measure each other using external sensors. When a robot senses another, it creates a factor in its factor graph between its and the other's latest pose variables. When each robot wants to send a message to another, it publishes the message to its **Robot Web Page**, which the other robot will eventually read and use to update its location estimate.

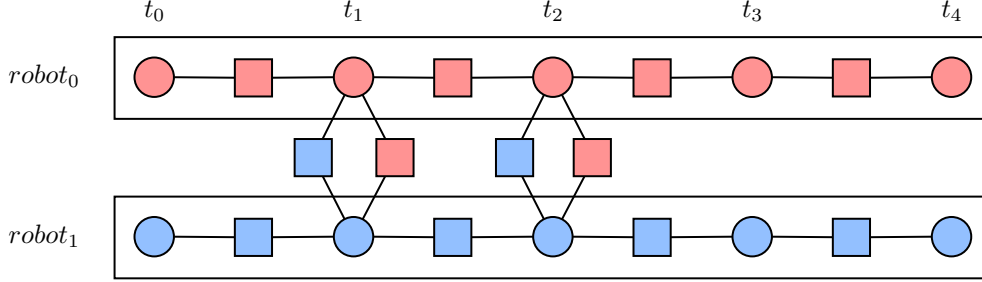


Figure 2.2: An example of a factor graph in the Robot Web. Each robot's variables are connected by odometry factors. At times t_1 and t_2 , both robots sense each other, and so exchange measurements by creating inter-robot-measurement factors on the graph.

The Robot Web represents the locations and sensor measurements of all robots using general Lie groups, rather than any specific group. This has the consequence that any type of sensor or robot can be a part of the Robot Web. For example, a drone moving in 3D space can interact with a car moving on a plane.

2.2.6 Evaluation

The Robot Web has been shown to improve the accuracy of robot localisation, and most importantly the inter-robot measurements have been shown to provide further improvements over schemes where robots would only measure landmarks.

Furthermore, the Robot Web has proven to be robust to a large number of faulty inter-robot sensors reporting random measurements, with this robustness lasting until 70-80% of inter-robot sensors reported corrupted measurements.

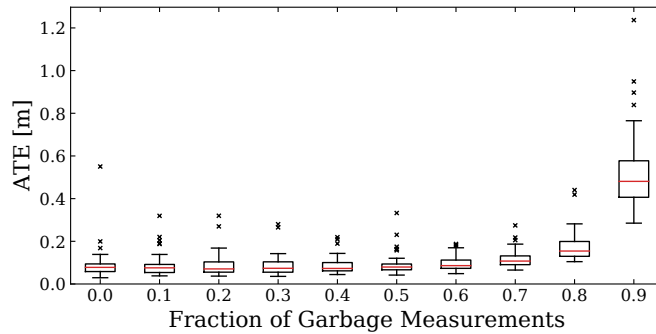


Figure 2.3: A graph showing that the RobotWeb is robust to up to 70-80% of “garbage” measurements, where faulty sensors report random measurements. ATE refers to the average Absolute Trajectory Error measured over 50 runs in an environment with 50 robots and 10 beacons running for 100 timesteps. Taken from [1, Figure 5]

Although the Robot Web is robust to many inter-robot sensors reporting random measurements, it is not robust to a bad actor which may instead report incorrect measurements designed to worsen the localisation of other members of the Robot Web. Possible attacks include but are not limited to:

1. Sending messages with extremely low standard deviations, to lull others into a false sense of security.
2. Sending these messages whilst assuming the identity of another robot.
3. Sending these messages from many nonexistent robots, also known as a Sybil attack.

2.3 Security Issues

In this section, we will discuss several general security issues that can arise in robot networks. We will focus on issues that affect the accuracy of a robot's internal model of the world. This excludes attacks which may result in an attacker gaining control over a robot, yet includes attacks performed by hijacked robots.

2.3.1 Denial of Service

A denial of service attack seeks to deny service. In a robot network such as the Robot Web, this would prevent one or more robots from being able to access messages sent by their peers, and essentially cut them off from the network.

The simplest way for an attacker to perform a DoS attack is to use a signal jammer, which can be constructed using off-the-shelf equipment [12]. This would continuously transmit signals within the range of frequencies allowed by the communication medium, both interfering with and irrecoverably corrupting any messages sent. More sophisticated attackers may craft harder-to-detect jammer attacks by mimicking legitimate messages or only transmitting when it senses communication [12]. In addition to these, there exist a whole host of jamming attacks (and defences) targetting specific communication protocols.

Another form of a DoS attack would be to disconnect specific robots from the network, using the network protocol's existing defences. For example, by convincing others that the target is a bad actor, triggering their defences to remove the target from the network.

2.3.2 Identity-Based Attacks

Identity-based attacks can wreak havoc on robot networks. Using the model defined by Douceur [13], we can describe a robot network as one consisting of E entities (robots) each claiming at least one identity i from the set of all identities I . When the network is under an identity-based attack, at least one of the following two properties will hold:

1. Two entities, $e1, e2$ will present the same identity i
2. The number of identities in I will exceed the number of entities in E .

If only the former holds, then the network is under a spoofing attack, whereas if only the latter holds, then the network is under a Sybil attack. Note: there is no guarantee that only one of these will hold at a time, and so we must be prepared to defend against both simultaneously.

Spoofing Attacks

Devices exchange information by sending packets or frames of data. For our purposes, we can ignore the differences between packets and frames, and use the terms interchangeably. Packets are used to encapsulate the data sent with relevant metadata, such as the source and destination IDs of the packet. This metadata is the main target of spoofing attacks.

In a spoofing attack, the attacker first finds the identity of a legitimate device, for example by first intercepting packets and then extracting the source ID from them. After this point, the attacker sends misleading packets impersonating the target.

This can have several benefits for the attacker. Firstly, they could covertly inject misinformation into the network by impersonating an already trusted robot. Secondly, they could trigger defences in the network to flag target for misinformation and remove it.

Sybil Attacks

In a Sybil attack, the attacker will create many fake identities to gain undue influence on the network. In a robot network, this would allow them to indirectly influence the actions of victim robots. For example, one could lead two self-driving cars to conclude that their best course of action is to crash, by claiming many false identities would be hit if they were not to.

Douceur [13] proves Sybil attacks are always possible in distributed systems where there is no central arbiter of truth. They present and prove four lemmas about identities in large-scale distributed systems. The first two lemmas are concerned with entities which directly validate the identities presented to them, whilst the second two lemmas involve entities which rely upon other, potentially untrustworthy, identities for validation. The lemmas are as follows:

1. If an attacker has ρ times as many resources as the weakest entity, then they can successfully present up to $\lfloor \rho \rfloor$ distinct identities.
2. If an entity doesn't validate all identities simultaneously, then an attacker can present an unbounded number of distinct identities.
3. If an entity trusts q other identities to validate an identity, then at least f attackers are required to perform the attack, where $f \geq q$ or the resources commanded by the attackers exceed $q + f$ times the weakest entity's resources.
4. If all c non-attackers don't coordinate when they validate identities, and an entity again trusts q other identities for validation, then even a weak attacker can present $\lfloor \frac{c}{q} \rfloor$ identities.

We plan to solve this problem by treating the physical world as a central arbiter of truth, albeit with some degree of error, due to imperfections in sensors.

2.3.3 Physical Attacks

Finally, since this thesis focuses on the security of robot networks, we will discuss the possibility of physical attacks on the system, and our limited ability to protect against them. We define a physical attack as any attempt to compromise the ability of a robot to perform its task. This includes colliding with the robot but also includes more subtle tactics, such as obscuring the robot's sensors. Physical attacks could also be used similarly to Sybil attacks, where several attackers would surround a robot and feed it misinformation.

In this thesis, we will not attempt to protect against attacks where several robots would physically collide, since this would require heavy hardware modifications to existing robots. Instead, we will focus on occlusion attacks and physical Sybil attacks, at the very least allowing a robot to detect them.

Several different techniques have been explored in preventing the types of identity-based attacks discussed in this chapter. In the 2nd half of this chapter, we will examine and evaluate these. We broadly group these approaches into 2 main groups; the first uses the physical characteristics of signal propagation to bind an identity to an entity, whilst the second exploits the fact that no entity can perform an unlimited amount of computation.

2.4 Wifi Fingerprinting

There are many ways for devices to communicate wirelessly, many of which use radiowaves. Wifi is the name of a family of networking protocols that allow for this, it derives from the IEEE 802.11 standard. In order to use Wifi, a device must have at least one wireless antenna which can transmit and receive within the bands specified in the IEEE 802.11 standard; usually 2.4GHz and 5GHz.

When a device sends a packet using Wifi, it transmits a radio signal for a given number of nanoseconds from its antenna. This signal will attenuate as it travels further and further through space. After a certain distance, also known as the communicating range of the antenna, the signal will fade into background noise. The signal leaves the antenna in all directions simultaneously, as a

radio wave. Eventually, a small part of this wave will reach the receiver’s antenna and the packet will be decoded out of it.

Other parts of the wave will either diffract around corners, reflect off some large objects, or scatter off many small objects [14]. Consequently, this means that a receiver may receive a packet from several different directions simultaneously, that is that it could encounter different parts of the same wave from different directions at the same time. This phenomenon is known as multipath scattering.

Multipath scattering has two interesting properties; it is practically impossible to predict, ahead of time, the distribution of signals around a receiver and it is unlikely that two receivers will observe the same signal propagation with sufficient multipath scattering. These properties allow for devices to “fingerprint” every packet they receive, such that two different transmitters cannot have the same fingerprint unless they are simultaneously located at the same place.

However, implementing multipath scattering-based algorithms have some technical constraints; namely that the receiving antenna must be able to measure the signal in each direction, and that a sufficient amount of multipath scattering must occur. Usually, these algorithms struggle in outdoor environments, where the environment may not provide objects for multipath scattering to occur.

The following papers discuss methods to circumvent these constraints, focussing on securing robot networks from spoofing and Sybil attacks.

2.4.1 Guaranteeing spoof-resilient multi-robot networks

Gil et al, [15] provides an interesting approach to some of the aforementioned problems, most notably the problem of requiring expensive hardware to allow the receiving antenna to measure the signal in each direction. They do this by inventing an algorithm that allows them to build a “virtual spoofer sensor” only using commercially available wifi hardware, which creates a “spatial fingerprint” from each transmission in the network. They use the output from this sensor to calculate a confidence metric α indicating their algorithm’s confidence that a robot’s identity is its entity. Finally, they characterise the theoretical performance of the “virtual spoofer sensor” and provide empirical evidence to support their claims by undertaking several experiments.

The authors build the “virtual spoofer sensor” by building upon *Synthetic Aperture Radar* (SAR) techniques, which allow a single antenna to be used to simulate an antenna array. SAR involves moving the antenna to different locations and taking snapshots of the signals received. These snapshots are then combined using signal-processing techniques to emulate a multi-antenna array [16]. The “spatial fingerprint” calculated, is then compared to the fingerprints of other clients, and clients with identical fingerprints are assumed to be Sybil attackers.

The authors evaluate their algorithm in the context of the following problem statement. Given an environment with several “clients”, each expecting service from mobile “servers”, dynamically find the optimal layout for the servers such that each “client” is served. A subset of clients are assumed to be malicious and are carrying out Sybil attacks in order to influence the “servers” to move closer to them.

The authors perform four experiments to validate their hypotheses:

1. They compare the performance of the “virtual spoofer sensor” in both an indoor and simulated outdoor environment, as expected, finding that multipath scattering is more effective in indoor settings, but also that adding a single reflector to the environment vastly improves performance.
2. They compare the effect of a stationary, moving, and power-scaling Sybil attacker on the ability of the “virtual spoofer sensor” to correctly classify agents, resulting in no false negatives, but many false positives.
3. They evaluate their system on the multi-agent coverage problem [17], finding that it can provide near-optimal results even when there are $3\times$ more spoofed agents.
4. They apply their system to a drone delivery problem, where the “server” needs to visit each real “client” to deliver a package and again find that their system provides near-optimal

results when there are $3\times$ more spoofed agents.

2.4.2 Lightweight Sybil-Resilient Multi-Robot Networks by Multipath Manipulation

Huang et al. [18] take a different approach to Gil et al. [15]; instead of relying upon the environment to provide multipath scattering, they actively cause it by using backscatter tags. This offers two main advantages over Gil et al.: 1. the environment has a lesser effect on the performance of the Sybil attack detector and 2. the robots' antennae no longer need to move when capturing a fingerprint. Using the captured signal information, the robots again compute a fingerprint per transmission, normalise it to mitigate the effects of any power scaling attacks, and finally compare the normalised fingerprint to those of all others, treating any identities with sufficiently similar fingerprints as Sybil attackers.

Backscatter tags scatter signals that they encounter by rapidly absorbing and reflecting them. Backscatter tags also simplify the fingerprinting process, since robots are no longer required to perform small movements for SAR, nor must the software engage in expensive linear algebra to construct a multi-antenna array.

A key property of backscatter tags is that they operate passively and don't require a power supply. This reduces the cost of implementing this scheme as tags can be simply and inexpensively attached to robots. Another useful property is that the backscattering of the final signal is highly correlated with the positions of the tags, transmitting and receiving antennae, meaning that if two identities have very similar backscattering patterns, they are likely to originate from the same entity.

When a robot receives a signal, it receives a raw signal, which is first smoothed out with a moving average filter, to create the backscattered signal. Then it decodes a message out of the backscattered signal, and uses it, with signal processing techniques to deduce how much each tag contributed to the backscattering, and when each tag was "activated". Then the robot does more filtering to remove any backscattering from the environment, the resulting signal will be used to construct a signature for the transmission.

The authors then evaluate their implementation in both an indoor office environment and an outdoor rooftop environment. They find that their method is virtually indifferent to the surrounding environment, as they measure an average true positive rate of 97.6% and an average false positive rate of 5.1%.

2.4.3 Conclusions

Although both sets of authors extensively test their system, they make some problematic assumptions, which may be exploited by attackers.

1. They do not account for Sybil attacks using multiple antennae, which could transmit the same message at the same time, but with variable power levels. Each antenna would create a different multipath scatter, and if the relative powers between them were varied, then they could theoretically construct many false fingerprints.
2. They also do not account for collaboration between different attackers, which would function similarly to the previous vulnerability, where geographically distributed attackers could simultaneously send the same message, with different power scales, creating another set of false fingerprints. This method could produce a larger range of false identities but may encounter synchronisation problems between attackers.
3. Attackers could leverage methods similar to Huang et al. and physically augment their antennae to manipulate their multipath scattering, for example, one could create moveable barriers to prevent some scattering from occurring.
4. Both sets of authors assume that any noise from the environment will not be malicious; Gil et al. assume that it will follow a Gaussian distribution, whilst Huang et al. assume that standard signal processing techniques would be sufficient to filter it out. Both of these assumptions fail to account for the possibility that coordinated attackers emit noise above background levels, but not so high that it would seriously interfere with transmissions. This

would disrupt every transmission’s signature and would prevent any pair of transmissions from looking alike.

2.5 Proof of Work

Proof of Work is underpinned by the insight that no entity in a network can perform unlimited computation. This leads to defences reliant on the idea that generating identities should be computationally expensive, to prevent any entity from being able to create an unbounded number of them. A PoW identity generation scheme would express an identity as the solution to some puzzle, and thus the identity can be validated by checking if it is a valid solution to the puzzle. This imposes two constraints: 1. the puzzle must be hard to solve and 2. the solution to a puzzle must be easy to verify, which means that, formally, the puzzle belongs to the *NP-Complete* class of problems.

A problem with this strategy is that it requires wasting computational resources, which may be in short supply for the embedded systems used in robotics. Furthermore, it requires that normal robots constantly pay a steep price for their security even without the presence of attackers.

Gupta et al. [19] design an iterative algorithm (**GMCom**) that solves the 2 problem, where if attackers spend T resources, whilst J new non-attacking identities are presented, then each non-attacker only needs to spend $O(\sqrt[3]{TJ} + J)$ resources. They refer to this property as the *assymetry* of the algorithm, as attackers must spend many more resources than non-attackers.

GMCom organises identities into a group, where a subset of them form a committee. When a new identity tries to join the group, it must first solve a puzzle set by the committee. Occasionally, the committee will seek to purge all attackers from the group, by issuing a *purge puzzle*, which must be solved by all identities before a new committee is formed, otherwise the identity will be purged from the group. This limits the wasted computation that good entities must perform since they only need to solve a single puzzle when entering a group, and occasionally thereafter, whilst an attacker would need to solve puzzles for each identity it claims, and would need to solve them repeatedly to avoid being purged.

However, this approach is not without its limitations. The authors assume that attackers will always only be able to command a fraction, α , of the total resources of the system, however, the heterogeneous nature of robots means that this may not always be guaranteed. For example, in a drone delivery system, each drone would likely only possess a small amount of computational power, but an attacker may attack the system using their desktop computer.

Chapter 3

An Investigation into the Security of RobotWeb

The goal of this chapter is to understand how attackers will affect the functioning of the RobotWeb. First, we will posture about the aims of attackers, and how they would pursue them. Then we will dedicate the remainder of the chapter to answering the question of how successful they can be; first theoretically then empirically.

3.1 The behaviour of an ideal attacker

The RobotWeb allows robots to improve their localisations; these are likely to be used for navigational or control purposes. If a robot believes that it is too far left, its navigation/control system will instruct it to move right. This link between localisation and action is the prize sought after by attackers. If an attacker wishes for a robot to move right, it must simply convince it that it has moved too far left.

In addition to compromising others' localisations, an attacker will wish to keep its localisation uncompromised. This would mean that it would ignore any messages from robots that it has attacked, lest it inadvertently misinforms itself. However, this will come at a cost to the attacker, as it has fewer good sources of information about its localisation. If an attacker attacks all robots around it, it would effectively cut itself out of the RobotWeb, which leads us to believe that attackers would seek to perform targeted attacks. Here the attackers would either only attack a small number of robots, or attack robots at key times. Motivated attackers, seeking to attack all robots, may instead use a private version of the RobotWeb, where their attacks don't affect them.

Attackers will also wish to prevent their attacks from being easily identified, as those can be easily defended against. If their attacks are defended against, the overall effectiveness of an attacker is greatly diminished. Thus it stands to reason that attackers would aim to blend into the crowd, where their messages are indistinguishable from good messages. This means that an attacker is unlikely to try to convince robots that they are 100s of kilometers away, instead opting for more subtle differences.

Finally, when multiple attackers are present in an environment, it would be best for them to collaborate rather than compete. Competing attackers are likely to interfere with each others' attacks and thus reduce their overall efficacy. For this reason, we will ignore situations where this occurs.

3.2 A simplified system

Any theoretical investigation of the RobotWeb would require the investigator to analyse its backbone, the factor graph. However, the investigator would soon find themselves ensnared by complexity; caught in an intricate web of variable and factor nodes; messages constantly scurrying

between them. Worse yet, the web would constantly be spun and unspun as robots moved closer and further from each other. In the face of this complexity, it becomes clear that simplification is needed.

To start, we choose to limit our investigation to a single variable, chosen arbitrarily, in the factor graph. Fortunately, the properties of Belief Propagation allow us to reason about this variable without loss of generality. Equation 2.9 shows that the belief of a given variable is solely dependent on its connected factors.

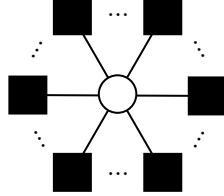


Figure 3.1: A single variable connected to a number of factors

One problem still remains with the above setup - a single variable can be connected to any number of factors. As a further simplification, we can group the factors together based on their shared characteristics, which include their origin (if they are internal to the robot or not), their type (what kind of sensor they represent) and their intentions (whether they will help or hinder the robot). For our purposes, we will split the set of factors F , into distinct sets G and B based on whether the factors are “good” or “bad” for the robot. Good factors aim to steer the variable towards the ground truth value, while bad factors aim to steer it away. Notice that the sets G and B form a cover of F , that is $F = G \cup B$.

We then take this a step further and replace each set of factors with a single “representative factor”.

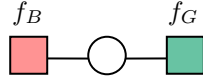


Figure 3.2: The above factor graph using “representative factors”. f_G and f_B respectively represent the sets of good factors (G) and bad factors (B).

We calculate the messages from each of these representative factors as such:

$$m_{f_G \rightarrow x_i} = \prod_{g \in G} m_{f_g \rightarrow x_i} \quad m_{f_B \rightarrow x_i} = \prod_{b \in B} m_{f_b \rightarrow x_i} \quad (3.1)$$

Thus the belief of the variable becomes $p(x) = m_{f_G \rightarrow x} m_{f_B \rightarrow x}$, and since $G \cup B$ covers every factor connected to the variable, we can show that this replacement can be made without altering the variable’s final result by equation 2.9.

$$p(x_i) = \prod_{s \in N(i)} m_{f_s \rightarrow x_i} \quad (2.9)$$

For completeness, we now present the equations for contents of the messages $m_{f_G \rightarrow x_i}$ and $m_{f_B \rightarrow x_i}$. Each message has an information vector η and a precision matrix Λ .

$$\eta_G = \sum_{g \in G} \eta_g \quad \Lambda_G = \sum_{g \in G} \Lambda_g \quad (3.2)$$

$$\eta_B = \sum_{b \in B} \eta_b \quad \Lambda_B = \sum_{b \in B} \Lambda_b \quad (3.3)$$

3.3 Theoretical properties of the RobotWeb under attack

3.3.1 Measuring the strength of an attack

From the above setup, we will now aim to quantify the strength of an attack. For mathematical convenience and ease of understanding, we will derive these equations in 1 dimension, and later present the n-dimensional forms.

To measure the strength of an attack we want to understand the impact of the bad factors on the variable's final belief. To do this we shall use the Kullback-Leibler (KL) divergence between the variable's belief when it is safe and when it is under attack. The KL divergence is a measure of how far the distribution Q is from the distribution P . The further the belief distribution under attack (BDA) is from the distribution suggested by the good factors, the stronger we say the attack is. Similarly, the closer the BDA is to the distribution suggested by the bad factors, the stronger the attack.

The KL divergence between 2 distributions of continuous random variables is:

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} \log \left(\frac{p(x)}{q(x)} \right) dx \quad (3.4)$$

However a special case exists for Normal distributions, $N_1(\mu_1, \sigma_1^2)$ and $N_2(\mu_2, \sigma_2^2)$.

$$D_{KL}(N_1||N_2) = \log \left(\frac{\sigma_2}{\sigma_1} \right) + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2} \quad (3.5)$$

In 1 dimension the messages sent from f_G and f_B are:

$$m_{f_G \rightarrow x_i} = (\eta_G, \Lambda_G) = \left(\frac{\mu_G}{\sigma_G^2}, \frac{1}{\sigma_G^2} \right) \quad (3.6)$$

$$m_{f_B \rightarrow x_i} = (\eta_B, \Lambda_B) = \left(\frac{\mu_B}{\sigma_B^2}, \frac{1}{\sigma_B^2} \right) \quad (3.7)$$

From 2.9 we see that the BDA is equal to:

$$m_{f_G \rightarrow x} m_{f_B \rightarrow x} = (\eta_G + \eta_B, \Lambda_G + \Lambda_B) \quad (3.8)$$

$$= \left(\frac{\mu_G}{\sigma_G^2} + \frac{\mu_B}{\sigma_B^2}, \frac{1}{\sigma_G^2} + \frac{1}{\sigma_B^2} \right) \quad (3.9)$$

$$= \left(\frac{\mu_G \sigma_B^2 + \mu_B \sigma_G^2}{\sigma_G^2 \sigma_B^2}, \frac{\sigma_B^2 + \sigma_G^2}{\sigma_G^2 \sigma_B^2} \right) \quad (3.10)$$

The above distributions follow the following Normal distributions:

$$G \sim N \left(\frac{\eta_G}{\Lambda_G}, \frac{1}{\Lambda_G} \right) \quad (3.11)$$

$$B \sim N \left(\frac{\eta_B}{\Lambda_B}, \frac{1}{\Lambda_B} \right) \quad (3.12)$$

$$BDA \sim N(\mu_{BDA}, \sigma_{BDA}^2) \quad (3.13)$$

$$\sim N \left(\frac{\eta_G + \eta_B}{\Lambda_G + \Lambda_B}, \frac{1}{\Lambda_G + \Lambda_B} \right) \quad (3.14)$$

$$\sim N \left(\frac{\mu_G \sigma_B^2 + \mu_B \sigma_G^2}{\sigma_G^2 + \sigma_B^2}, \frac{\sigma_G^2 \sigma_B^2}{\sigma_G^2 + \sigma_B^2} \right) \quad (3.15)$$

So now we take the KL divergence between the good factors' distribution G and BDA. To make this easier to follow we derive each term in the sum separately.

First deriving the log term we get:

$$\log \left(\frac{\sigma_{BDA}}{\sigma_G} \right) = \log \left(\sqrt[2]{\frac{\sigma_{BDA}^2}{\sigma_G^2}} \right) \quad (3.16)$$

$$= \frac{1}{2} \log \left(\frac{\sigma_{BDA}^2}{\sigma_G^2} \right) \quad (3.17)$$

$$= \frac{1}{2} \log \left(\frac{\sigma_G^2 \sigma_B^2}{\sigma_B^2 + \sigma_G^2} \times \frac{1}{\sigma_G^2} \right) \quad (3.18)$$

$$= \frac{1}{2} \log \left(\frac{\sigma_B^2}{\sigma_B^2 + \sigma_G^2} \right) \quad (3.19)$$

Next, we derive the fractional term:

$$\frac{\sigma_G^2 + (\mu_G - \mu_{BDA})^2}{2\sigma_{BDA}^2} = \frac{\sigma_G^2 + \left(\mu_G - \frac{\mu_G \sigma_B^2 + \mu_B \sigma_G^2}{\sigma_B^2 + \sigma_G^2} \right)^2}{2 \frac{\sigma_G^2 \sigma_B^2}{\sigma_B^2 + \sigma_G^2}} \quad (3.20)$$

$$= \left(\sigma_G^2 + \left(\frac{\mu_G \sigma_G^2 + \mu_G \sigma_B^2 - \mu_G \sigma_B^2 - \mu_B \sigma_G^2}{\sigma_B^2 + \sigma_G^2} \right)^2 \right) \times \frac{\sigma_B^2 + \sigma_G^2}{2\sigma_G^2 \sigma_B^2} \quad (3.21)$$

$$= \left(\sigma_G^2 + \left(\frac{\mu_G \sigma_G^2 - \mu_B \sigma_G^2}{\sigma_B^2 + \sigma_G^2} \right)^2 \right) \times \frac{\sigma_B^2 + \sigma_G^2}{2\sigma_G^2 \sigma_B^2} \quad (3.22)$$

$$= \frac{\sigma_G^2 (\sigma_G^2 + \sigma_B^2)^2 + (\mu_G \sigma_G^2 - \mu_B \sigma_G^2)^2}{(\sigma_B^2 + \sigma_G^2)^2} \times \frac{\sigma_B^2 + \sigma_G^2}{2\sigma_G^2 \sigma_B^2} \quad (3.23)$$

$$= \frac{\sigma_G^2 (\sigma_G^2 + \sigma_B^2)^2 + \sigma_G^4 (\mu_G - \mu_B)^2}{(\sigma_B^2 + \sigma_G^2)^2} \times \frac{\sigma_B^2 + \sigma_G^2}{2\sigma_G^2 \sigma_B^2} \quad (3.24)$$

$$= \frac{(\sigma_G^2 + \sigma_B^2)^2 + \sigma_G^2 (\mu_G - \mu_B)^2}{2\sigma_B^2 (\sigma_B^2 + \sigma_G^2)} \quad (3.25)$$

Thus the KL divergence between G and BDA is:

$$\frac{1}{2} \log \left(\frac{\sigma_B^2}{\sigma_B^2 + \sigma_G^2} \right) + \frac{(\sigma_G^2 + \sigma_B^2)^2 + \sigma_G^2 (\mu_G - \mu_B)^2}{2\sigma_B^2 (\sigma_B^2 + \sigma_G^2)} - \frac{1}{2} \quad (3.26)$$

Similarly, the KL divergence between B and BDA is:

$$\frac{1}{2} \log \left(\frac{\sigma_G^2}{\sigma_B^2 + \sigma_G^2} \right) + \frac{(\sigma_G^2 + \sigma_B^2)^2 + \sigma_B^2 (\mu_G - \mu_B)^2}{2\sigma_G^2 (\sigma_B^2 + \sigma_G^2)} - \frac{1}{2} \quad (3.27)$$

From this, we notice a disturbing detail - the KL divergence is quadratically affected by the μ_B . This suggests that the attacker's power is unbounded, so long as it chooses an appropriate σ_B^2 . This means that an attacker can suggest absurd localisations and have its victims believe them. For example, an attacker can convince its victims that they are on the opposite side of the planet!

3.3.2 Crafting the perfect message

From the results derived in the previous section, we know that, theoretically, an attacker will seek to increase its μ_B to ∞ and decrease its σ_B^2 to 0. However, in a real-life scenario, it is unlikely that the attacker would fully exploit these powers, for two reasons.

1. Robots are unlikely to believe *incredibly* incorrect values - no sensible robot would believe that it has moved 500km away in the past 3 seconds.
2. Robots don't have infinite numerical precision, so large values of Λ_B ($\frac{1}{\sigma_B^2}$) would cause overflow errors, which would prevent the attacker from controlling the robot's belief.

Given these restrictions, attackers would set their values of μ_B to believable values, whilst choosing the largest value of σ_B^2 possible. We will now suppose that the attacker wishes to move its victim's belief from μ_G to μ_T . As before, the attacker sends a message with the form described in equation 3.7.

So from equation 3.15, we see that:

$$\mu_T = \frac{\mu_G \sigma_B^2 + \mu_B \sigma_G^2}{\sigma_G^2 + \sigma_B^2} \quad (3.28)$$

Which can be rearranged into the form:

$$\mu_B = \frac{\mu_T (\sigma_G^2 + \sigma_B^2) - \mu_G \sigma_B^2}{\sigma_G^2} \quad (3.29)$$

Which can be used to calculate the μ_B that an attacker would send given a minimum σ_B^2 . Thus the attacker would send the following message:

$$\left(\frac{\mu_T (\sigma_G^2 + \sigma_B^2) - \mu_G \sigma_B^2}{\sigma_G^2 \sigma_B^2}, \frac{1}{\sigma_B^2} \right) \quad (3.30)$$

3.3.3 Scaling back up to n-dimensional space

For completeness, we now present the KL divergence between the BDA and G distributions in k-dimensions.

The KL divergence between 2 k-variate Normal distributions, $N_1(\mu_1, \Sigma_1)$ and $N_2(\mu_2, \Sigma_2)$ is:

$$KL[P||Q] = \frac{1}{2} \left[tr(\Sigma_2^{-1} \Sigma_1) + (\mu_2 - \mu_1)^T \Sigma_2^{-1} (\mu_2 - \mu_1) - k + \log \left(\frac{\det \Sigma_2}{\det \Sigma_1} \right) \right] \quad (3.31)$$

Using this with the messages sent from f_G and f_B , we find that the KL divergence between G and BDA is:

$$\frac{1}{2} \left[tr(\Lambda_B \Lambda_G^{-1}) + (\mu_{BDA} - \mu_G)^T (\Lambda_G + \Lambda_B) (\mu_{BDA} - \mu_G) + \log \left(\frac{\det \Lambda_G}{\det (\Lambda_G + \Lambda_B)} \right) \right] \quad (3.32)$$

where

$$\mu_{BDA} = (\Lambda_G + \Lambda_B) (\eta_G + \eta_B)$$

3.3.4 The Force-Based model of attacks

From the above analysis, we believe that attacks can be reasoned about using a “Force-Based Model”. Here we think of all messages sent to a variable as forces pulling the variable to particular localisations. Every good message will pull the variable towards the ground truth value, whilst every bad message will pull it towards an alternative localisation. The strength of a message’s “pull” corresponds to the eigenvalues of its Λ matrix, where the higher the eigenvalues, the stronger the pull is.

If two messages pull the robot towards the same localisation, then their forces will be additively combined, increasing their overall strength. However, if the messages pull the robot to diametrically opposed localisations, then their forces will cancel out. Once all forces have been resolved, the variable will move to the localisation suggested by the resultant force. The strength of the variable’s conviction is determined by the strength of this resultant force.

3.4 Hypotheses

Now using the “Force-Based Model”, we form 6 hypotheses about the behaviour of the entire RobotWeb under attack. Later, we will experimentally verify each of these hypotheses.

3.4.1 There is Strength in Numbers

As the number of good robots rises, the effectiveness of an attack will fall. We believe that this can be seen through the lens of the “Force-Based Model” where, as the number of good robots increases, so does the overall number of good messages, which in turn increases the overall force pulling the robots towards the truth. This will increase the resistance against any given attack, decreasing the strength of the resultant force.

However, it is unlikely that a large number of good robots will be able to nullify attacks, as the attacker will likely respond by increasing the strength of its messages. Since the good robots cannot freely increase Λ , they will once again conform to the attacker’s wishes.

3.4.2 Long Memories are Detrimental

One little discussed feature of the RobotWeb so far has been its time-windowed history. Instead of remembering every past position that that robot has had, it instead chooses to only remember the past h positions. This improves the performance of robots in the RobotWeb, as they store fewer pose variables and thus perform fewer floating-point operations when updating beliefs. In the original paper, Murai et al. show that the impact of time-windowed history on the average performance of each robot is negligible.

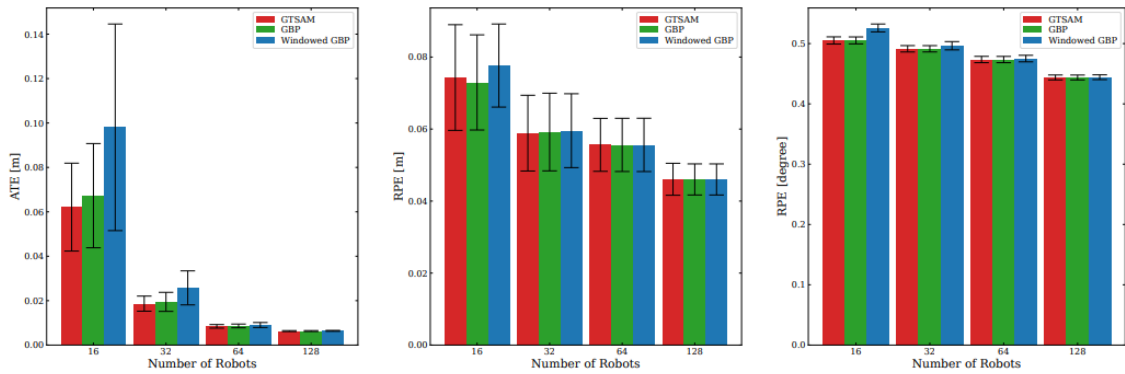


Figure 3.3: Graphs showing a negligible decrease in performance between using a full history vector and only using the previous 5 pose variables. Taken from [1, Figure 3]

However, we believe that keeping the full position history not only harms the computational performance of a robot but also amplifies the strength of attacks. If an attacker can successfully attack a robot at time t , then the pose variable at t , v_t will contain a μ close to the attacker’s target μ , and its Λ will be high. Then in the next step, this corrupted v_t will pull the robot towards the attacker’s target instead of towards a ground truth. As the size of this history increases, so does the total pull it will have on the robot.

Note: During the early stages of an attack, we believe that longer histories will serve as a small counterbalance to the attacker, but as their strength is limited, they will be unable to significantly prevent an attack. Thus we believe that they are more harmful than helpful.

3.4.3 Attacks are Contagious

Similarly to the previous hypothesis, we can assume that any other variable connected to v_t will be attacked by it. Thus a victim of an attack will unintentionally attack all those it contacts, leading us to hypothesise that attacks have a degree of contagion.

Attackers may not be able to prevent or limit contagion, as they would need to strongly pull unintentional victims back to a ground truth value. Since no robot has absolute certainty about the locations of any other, the attacker will pull them to values close to their ground truth. Yet, as there is no mechanism to decrease Λ , the attacker will still make the other robots overconfident about their locations, which may bias them in the long term.

3.4.4 Young Robots are the most Vulnerable

Robots which have only recently joined the RobotWeb (and hence can be considered “young”) are unlikely to have strong beliefs about their localisation. As they participate in the RobotWeb, the strength of their beliefs will increase. However, these initial weak beliefs will translate to a weaker pull towards the ground truth, and provide a weaker resistance to attackers.

We believe that this effect may be strongly linked to the robot’s history size, as once the history size has been exhausted, the robot will only make marginal gains from experience.

3.4.5 Bounds on μ can be slowly escaped

An intuitive defence against the attackers is setting an upper bound on how far a message can suggest the robot is from its current position belief. However, we believe that this approach is ineffective, and can be easily evaded by attackers. In this subsection, we will lay out how this defence would work and how it can be bypassed.

For the upper bound to be effective, it must occupy a “Goldilocks zone” - it cannot be too large or too small. If the upper bound is too large, it would present attackers with ample opportunities to control the robot’s belief, especially since they aim to suggest somewhat plausible positions. If the upper bound is too small, it would effectively prevent the robot from listening to any dissenting messages from other good robots.

Now suppose that at time t , the robot is at position μ_t and has an upper bound of ϵ . It would then evaluate each incoming message and reject it if its proposed position μ'_t is a distance ϵ away from μ_t . After filtering out all “bad” messages, the robot would use the remaining messages to determine μ_{t+1} .

Being aware of this scheme, an attacker would seek to incrementally attack the robot. It would start by proposing a position μ''_t that lies between μ_t and its target location μ'_t , such that μ''_t is accepted by the robot. This would successfully shift the robot’s position estimate μ_{t+1} to μ''_t . In the next iteration, the robot would reject any proposals a distance of ϵ away from $\mu_{t+1} = \mu''_t$. Hence over several iterations, the robot’s position estimate would slowly shift to μ'_t , and thus the attacker would be able to escape the bound.

3.4.6 Bounds on Λ can be quickly escaped

As shown previously, the strength of an attack is highly dependent on its ability to propose arbitrary values of Λ . This leads to another intuitive defence strategy - robots set an upper bound on the Λ values that they receive. We believe that this strategy is effective when there is only a single identity spreading misinformation, counterbalanced by many others sending reliable information. In this subsection, we will provide a justification for this belief as well as a strategy that an attacker could take to bypass the upper bound on Λ .

When sending a message, each robot decides the strength of the message’s pull. Good robots limit their strength to the accuracy of their sensors, while attackers will set their strength to the fullest extent. This once again makes the case for the use of a reasonable upper bound on the strength of a message, similar to the one described above.

However, we argue that this upper bound is unenforceable in practice, for an attacker can simply “split” their message into several weaker parts. So far in our analysis, we have treated Λ_B as if it were sent in a single message, however from equation 3.3 we can see that it may also be the result of several bad robots sending messages. If the upper bound on an individual message’s Λ is λ , then an attacker could simply send several messages from several different identities to arrive at a strong Λ_B , in a Sybil Attack. Here the strengths of each Sybil’s message will increase the overall strength of the attacking force, while each message’s Λ will fly under the bound.

Aside: Attacks are uncorrelated

Considering the “Force-Based Model” of variables, it stands to reason that in a Sybil Attack, the messages sent by individual identities will not be correlated with each other, as that would greatly simplify the detection and prevention of attacks. Instead, we believe that Sybil Attackers would

send messages with wildly different μ values, that would resolve to the actual attack that the attacker intends.

Consequently, it becomes impossible to detect whether a message belongs to an attacker based on its contents.

3.5 Experimental evidence

Now, we will evaluate the above hypotheses in a simulated environment. Our goal is to show that the predicted phenomena can arise, rather than guaranteeing their emergence. The reasoning behind this is that an attack only needs to be successful in a single situation for it to be considered potent.

3.5.1 Experimental setup

We will use an adapted version of the simulator detailed in [1, RobotWeb] for our experiments. Each simulation will contain a single attacking robot and n normal robots. Each robot will move along a predefined path, and aim to localise well to that path - this will be challenged by the attacker, which will attempt to trick them into localising to a path of its choosing.

Furthermore, each robot in the simulation will carry 2 sensors; one for odometry and one for measuring others. The odometry sensor will be used by a robot to estimate its current location using its previous location and its motion. The range-bearing sensor will be used to measure other robots - it will measure the distance between the 2 robots and the bearing of the other robot from this one. All robots' range-bearing sensors have a range limit; however, attackers will ignore this as they always know where they want their victims to be.

The simulation has several configurable parameters which we will vary in our experiments. We detail the most important ones below.

Parameter	Description	Default values
Dataset	The physical path travelled by each robot.	See below
Attacker Dataset	The path attackers attempt to trick the robot into localising onto	See below
History size	The number of past pose variables used by each robot.	1
N iterations per step	The number of GBP iterations performed every time the robot moves.	10
SE2 Prior	The accuracy of each robot's original position.	$\sigma_x = 1e^{-8}$ m $\sigma_y = 1e^{-8}$ m
Between SE2	The accuracy of each robot's odometry sensor.	$\sigma_x = 0.1$ m $\sigma_y = 0.1$ m $\sigma_\theta = 0.01$ rad
Sensor range limit	The furthest a range-bearing sensor can detect.	0.25 m
Sensor noise model	The noise associated with the range and bearing measurements.	$\sigma_r = 0.05$ m $\sigma_\theta = 0.1$ rad
Attacker confidence	The confidence claimed by the attacker.	$\sigma_x = 0.001$ m $\sigma_y = 0.001$ m

Path design

Since the RobotWeb currently hasn't been deployed, we are faced with a dearth of real-world paths for our experiments. This leaves us with the choice of either inventing a realistic path or using a

simple, artificial path. For the following experiments, we have chosen to use simple paths as we aim to show that these phenomena can arise. Later we will test our defences on more realistic paths.

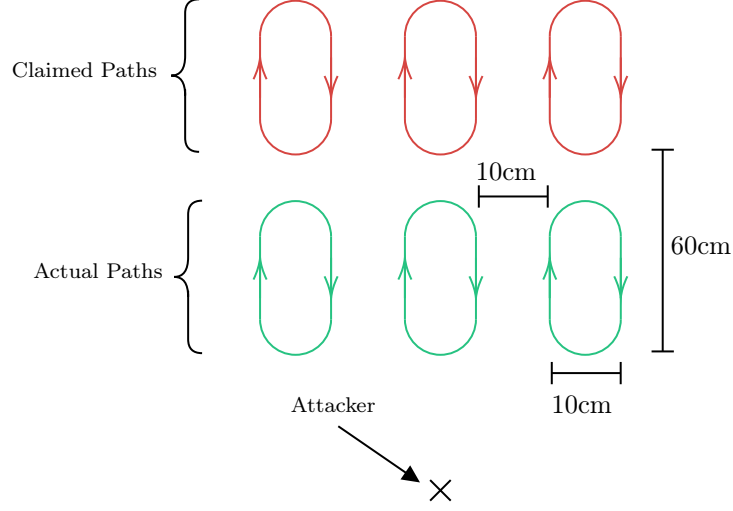


Figure 3.4: An example of the types of paths we use to test our hypotheses. In this example, there are 3 good robots and 1 attacking beacon. The attacker aims to shift the beliefs of its victims 60cm upwards.

We choose to use the above paths to test. The robots will move along the green path, whilst the attackers will try to convince them that they are instead moving along the red path. For these experiments, the attacker will take the form of a beacon placed at the bottom of the map.

3.5.2 Testing hypothesis 1

The first hypothesis, 3.4.1, predicts the existence of a “strength in numbers” effect, where the effectiveness of an attack is reduced as the number of good robots increases. We test this by varying the number of robots attacked and subsequently measuring the Average Trajectory Error (ATE) per robot. For a fair comparison, we extend the range of each robot’s sensors to allow for every robot to sense every other robot. We repeat this experiment 10 times.

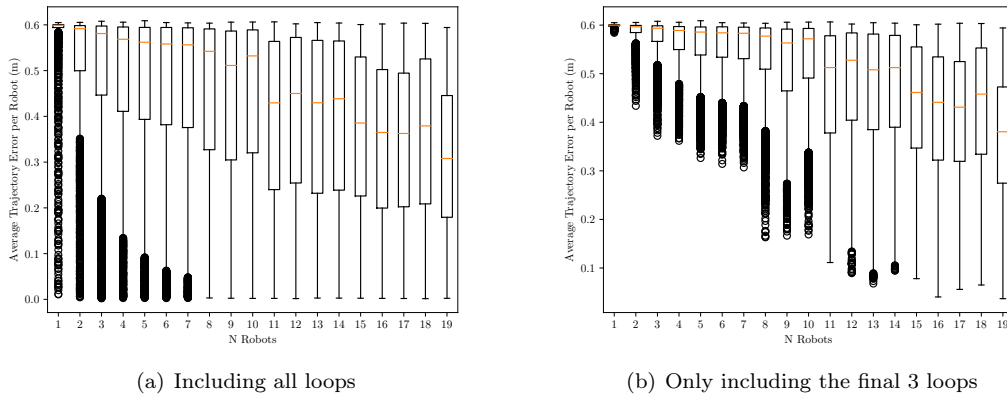


Figure 3.5: The “strength in numbers” effect: as the number of robots increases, the effectiveness of an attack decreases. Note the outliers in 3.5(a) occur as the weakened attacker is unable to instantly seize control of its victims.

Looking at 3.5, we can see that a weak “strength in numbers” effect does exist, as the ATE decreases whilst the number of good robots increases. However, this effect was only observable

after the attackers had been weakened, such that their $\sigma_X = \sigma_y = 1\text{m}$ - which is a lower confidence level than all normal robots’.

Since this effect is only visible when attackers have been weakened, it is unlikely to present a significant hurdle to real-world attackers, unless they are vastly outnumbered. Therefore it is not a viable defence strategy.

3.5.3 Testing hypothesis 2

The second hypothesis, 3.4.2, suggests that robots keeping longer histories are more vulnerable to attacks. We test this using 2 robots; an attacker and its victim. We vary the victim’s history size and measure its ATE. We again weaken the attacker’s attacks by reducing its confidence such that $\sigma_x = \sigma_y = 10\text{ m}$. We repeat this experiment 10 times.

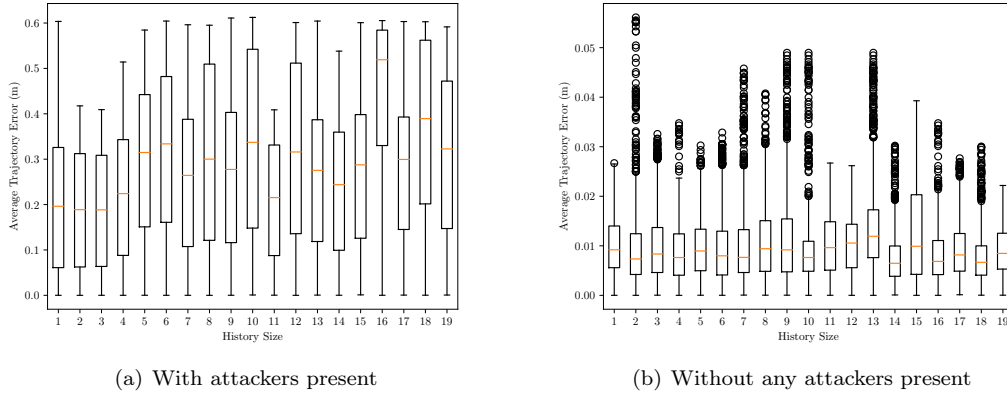


Figure 3.6: The effect of History Size on ATE: a longer history size is detrimental when attackers are present, yet provides little value otherwise.

Looking at 3.6(a) reveals that longer histories do harm robots’ localisation, but only weakly. This is likely because the past poses are fully attacked by then, and so can’t attack the robots further. However, 3.6(b) tells a different story about the value of longer histories. Here we see a negligible impact of longer history sizes on the ATE.

These results show that there is no advantage to be gained by using a longer history vector, which leads us to conclude that the history size should be kept at 1 to reduce the strength of attackers. This is likely to serve as an additional benefit as it reduces the overall computation required by members of the RobotWeb.

3.5.4 Testing hypothesis 3

The third hypothesis, 3.4.3, predicts that attacks will spread without the active involvement of the attacker, where the victims inadvertently will attack other robots. We test this using 12 robots; an attacker, its victim and 10 other robots. We devise a special configuration for this experiment where the attacker only communicates with its victim, and only adjacent robots can communicate with each other. This provides us with a daisy chain communication pattern, where the victim’s attack must take several “hops” to reach the ends of the chain. We then measure the ATE of each robot over 10 runs.

Looking at 3.7(a), we can see that the fewer the number of hops from a robot to the victim, the higher its ATE will be, thus proving the existence of a contagion effect. However, we also see that the effect of the contagion decays, as the number of hops increases. This is likely because each victim does not *try* to attack others, only doing so inadvertently, so the amount of influence it exerts on its neighbours is considerably lower than the influence the attacker exerts upon it. As a victim 1 hop away is less influenced, it too exerts less of an influence upon its neighbours. The overall effect of this is that the contagion is unlikely to spread rapidly and uncontrollably.

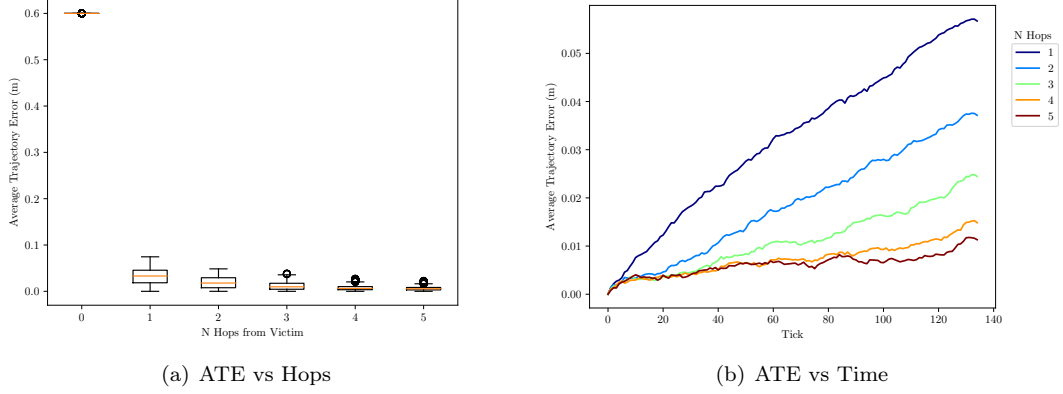


Figure 3.7: The contagion effect: attacked robots slowly spread their attack out to their neighbours

Looking at 3.7(b), we also see that the ATE of each affected robot increases linearly, which implies that given enough time, the contagion can overwhelm the RobotWeb.

This effect may somewhat deter real-world attackers from relying on any robots in the vicinity of their victims, lest they fall prey to their own attack, yet it is unlikely to act as an effective deterrent. A motivated attacker would instead find alternative means of localisation.

3.5.5 Testing hypothesis 4

The fourth hypothesis, 3.4.4, posits that young robots which have recently joined the web will be the most vulnerable to attacks. We test using 2 robots; an attacker and its victim. We introduce a delay between when the victim starts and when it is attacked. We then measure the ATE of the victim across 10 runs. We also vary the victim's history size, as we believe that there is likely a link between the optimum startup delay of a robot and the size of its history. We once more weaken the attacker's attacks by reducing its confidence such that $\sigma_x = \sigma_y = 1\text{m}$.

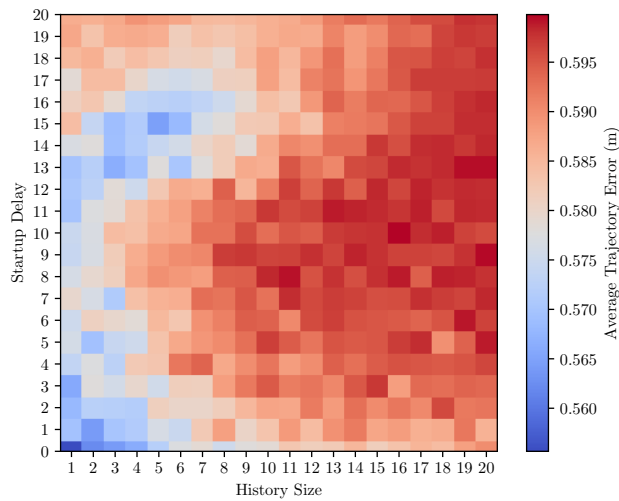


Figure 3.8: The effect of a startup delay on the ATE, also investigating the effect of the history on this.

Note: This section is still a TODO because I realised that I made a mistake in the experiment.

3.5.6 Testing hypothesis 5

The fifth hypothesis, 3.4.5, predicts that attackers can overcome bounds on μ , by performing incremental attacks. We test this using 2 robots; an attacker and its victim. For the victim, we introduce a μ bound defence of 0.05m, so that it only believes messages which are within a 0.05m radius from its current position estimate. For the attacker, we add an onramp composed of several segments, to transition the victim to the attacker’s desired path. We vary the number of the onramp’s segments and measure the victim’s Average Trajectory Error (ATE) over 10 runs.

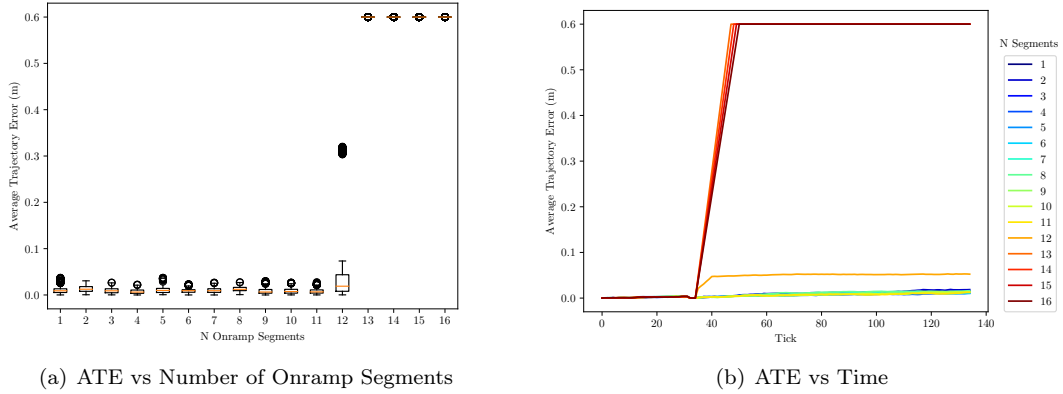


Figure 3.9: Attackers escaping a 5cm μ bound: the 60cm difference between the actual and target path is bridged by n *onramp segments* of equal lengths

Analysing the ATE against the number of segments, in 3.9(a), we see that a cliff emerges, where once the segments are short enough, the ATE significantly increases, indicating a successful attack. Thus confirming that an attacker can easily overcome bounds on μ .

Interestingly enough, we also see that the cliff starts when there are 12 segments. Here each segment would have a length of $\frac{0.6}{12} = 0.05\text{m}$, which is just outside the bound. This happens due to noise in the victim’s localisation, which leads it to occasionally accept the attacker’s messages. Every time the victim accepts one of these messages, it’s pulled closer to the attacker’s desired path, which in turn makes it more likely to accept the attacker’s future messages.

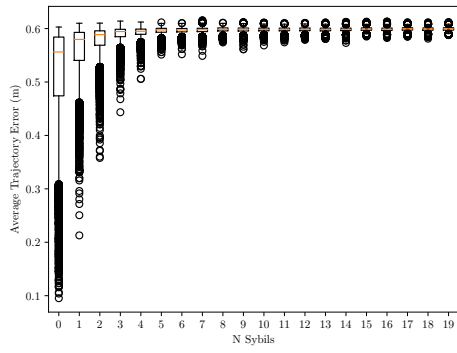
Looking at the victim’s ATE over time in 3.9(b), we see that the victim’s ATE only increases once it is firmly on the onramp. In the case where there are 12 segments, we see that the victim only partially follows the target path, which is once again likely due to its noisy localisation.

These results mean that although the μ bound defence can be evaded, it still hinders the attacker, which must both create an onramp and wait for its victim to traverse it.

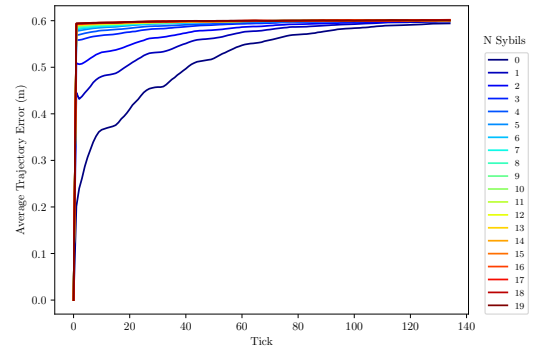
3.5.7 Testing hypothesis 6

The sixth and final hypothesis, 3.4.6, asserts that an attacker can carry out an attack without manipulating its confidence level, instead the attacker would send many messages under many fake identities. We test this using 12 robots; an attacker and its 11 victims. We also prevent the attacker from altering the confidence level of any of the messages it sends. We vary the number of false identities that the attacker creates, and measure the average ATE across each victim. We repeat this experiment 10 times.

From 3.10(a), we see that the ATE quickly converges around 0.6m as the number of Sybils increases. We also see the spread of ATEs narrow. Both these facts confirm that the Sybil attack is highly effective against the RobotWeb. Notably, since the confidence level of messages sent from a Sybil is equal to that of those sent by normal robots, they blend in perfectly with normal messages. This means that a robot cannot determine the true state using messages alone. Furthermore, unlike the previous experiment, when we analyse the ATE over time, we see that the effect of this attack has no delay, which increases its potency.



(a) caption



(b) caption

Figure 3.10: A Sybil attack on the RobotWeb: as the number of Sybils increases, so does the influence of the attacker upon the rest of the RobotWeb

Chapter 4

Defence against the Dark Arts

This chapter serves as a dual to the previous chapter. Here we will use our understanding of the RobotWeb under attack to design and evaluate a collective defence strategy. First, we will discuss the limitations of individualistic defence Strategies. Then we shall outline the conceptual backbone of our defence, and evaluate it in realistic scenarios. Finally, we will evaluate some improvements to this defence.

4.1 The Limits of Individualism

When finding ways to defend robots from attack, one may intuitively reach for approaches where a single robot can defend itself - and these approaches are not without their merits. First, they guarantee that each robot would *always* be able to defend itself from attack, as it only depends on itself. Secondly, they serve as effective deterrents against attacks, as an attacker wouldn't attempt an attack without any hope of success. Essentially allowing for full and fearless participation in the RobotWeb.

However, we argue that these approaches are not possible in the RobotWeb as it stands today. In an entirely trustless system, a robot would be entirely reliant on its odometry sensors to verify the veracity of incoming messages. Yet as we have seen in 3.4.5 and 3.4.6, such measures can be easily circumvented. Thus we can conclude that a robot must find a method to trust others. For one robot to be trusted by another, it must be able to credibly signal its trustworthiness to the other. This can only happen if robots are penalised for spreading misinformation and only for spreading misinformation.

One such approach would be to use a reputation system, where messages from robots with a higher reputation are given a higher weight than those with lower reputations. A robot's reputation would increase with every correct message it sends, and decrease with every incorrect message. Robots would decide the correctness of a message by measuring how well it aligns with their current beliefs - the better the alignment the more likely the message is to be correct. Reputation systems come in 2 distinct flavours, local and global.

In a global reputation system, all robots in the RobotWeb would collectively track one another's reputations, so discovered attackers would be universally distrusted. However, attackers could also attempt to lower the reputations of good robots by claiming that *they* themselves are victims of the good robots. This presents a problem, as now robots need a mechanism to verify misinformation claims, yet if such a mechanism existed then they would not need to rely on a global information system. This leads us to conclude that a global reputation system is not a viable solution here.

Alternatively, in a local reputation system, each robot maintains its own set of beliefs about the reputations of those it encounters. This eliminates the need for multiple robots to reconcile their reputation beliefs and thus prevents attackers from attacking the reputations of others. However, this still falls prey to problems that haunt all reputation systems. Firstly, an attacker can reset its reputation by simply changing its identity. Secondly, an attacker can amass a significant amount

of reputation through the use of Sybils, which it would then use to attack.

Another approach would be to use resources to provide credible signals akin to Proof of Work and Proof of Stake systems. As shown in the background section, schemes using computational resources fall flat due to the heterogeneity of the RobotWeb - some robots may run on micro-controllers whilst others feature GPUs. Schemes backed by financial resources also face the same problem as global reputation systems, as there is still no trusted method to solve disputes.

Finally, we believe that any motivated attacker is likely to treat any incurred penalties as prices, which poses a significant threat to safety as robots are physical entities. Thus we look to systems with a degree of partial trust for our solution.

4.2 A Group-Based Defence

Given that individual robots are unable to defend themselves against attacks, we now organise them into groups. Robots form these groups using prior information that allows them to trust one another. For example, robots with the same owner can all trust each other. The nature of this prior information is left as an implementation detail as several suitable approaches exist.

Each robot in a group can have one of 2 roles - it can be a leader or a follower. Leaders choose to ignore all messages they receive from all untrusted robots. Crucially, this also includes their followers, as a follower under attack will unwittingly attack all others due to the contagion effect. Since leaders are unable to receive messages from other sources, their localisation accuracy will fall over time. This is combatted through term limits, where once a leader has served its term, it will revert to a follower and trigger an election.

Meanwhile, a follower relies on its leaders to filter misinformation. It first combines all the messages it receives from its leaders using 2.9 to obtain a baseline estimate. Then for each message it receives, it calculates the KL divergence between the message and its baseline estimate, if the KL divergence falls outside a bound, it will reject the message. Note, a follower is unwaveringly obedient to its leader and can reject messages from its odometry factors if they fall outside the bound.

At first glance this may seem similar to the μ bound defence discussed earlier however, this is not the case. Unlike the μ bound defence, the baseline estimate here cannot be influenced by an attacker as it originates from the leaders, which cannot be attacked as they ignore all untrusted incoming messages. If an onramp attack occurs, the attacker will not be able to convince its victim past a certain point, as the baseline estimate will not shift with it. A Sybil attack is similarly rendered useless as it will fail to move the robot's belief outside the bound of the baseline estimate.

4.2.1 Inductive proof

We will now formally prove the resiliency of this defence for both leaders and followers.

Lemma 1: Combining 2 correct localisations will always result in yet another correct localisation.

Lemma 2: Combining n incorrect localisations will never result in a correct localisation

Leaders

At time $t = 0$, a leader will start with a correct localisation.

Now assuming that at time $t = k$, the leader has a correct localisation.

It will receive messages only from other leaders, which are guaranteed to have correct localisations. Combining these messages, the leader will arrive at another correct localisation. Since the localisation at time $t = k + 1$ is dependent on the localisation at time t and the messages sent at time t , the leader now has a correct localisation.

Followers

At time $t = 0$, a follower will also start with a correct localisation.

Now assuming that at time $t = k$, the follower has a correct localisation.

It will first receive a series of correct localisation messages from the leaders, which it will combine into another correct localisation. Then it will receive a series of messages which could have correct or incorrect localisations. From this, it will only allow the correct localisations through, and so its next localisation will be correct.

4.2.2 Experimental confirmation

It's testin time Realistic scenarios Attacks be attacking

4.2.3 Strategies to improve performance

4.2.4 Exits, Mergers and Acquisitions

Chapter 5

Conclusion

5.1 Ethical Considerations

“A new device merely opens a door; it does not compel one to enter”
(Lynn White [20])

From the discovery of coffee leading to the “Age of Enlightenment” to the invention of Boolean algebra creating our modern digital age, history has repeatedly shown us that it is impossible to fully understand the implications of new discoveries and nascent technology. With this in mind, we provide a short discussion of potential ethical issues which, we believe, may arise as a consequence of the research conducted in this thesis.

As with all research enabling autonomous robotics, we must consider potential military applications. Many militaries today already use unmanned aerial combat vehicles in their operations, if they were to incorporate this research, they may be able to improve their performance by allowing them to share information securely. However, we do not believe that this is likely to occur as militaries tend to have highly centralised structures, where each robot would have prior knowledge about other trusted robots in the network. Whereas our research focuses on providing security to untrusted robots in decentralised networks.

Another potential misuse of this research would be enhancing the capabilities and security of surveillance robots. In this scenario, an authoritarian regime would use robots to continuously monitor their citizens. The robots would communicate with others in their immediate surroundings to coordinate their search and could be vulnerable to cyber attacks where several are hijacked. The hijacked robots would then send incorrect messages to prevent certain areas from being searched. However, it is unlikely that this research would be an ideal candidate for implementing such a dystopia, as a single party would own the robots and would find it much simpler to implement centralised security measures.

Alongside the unethical misuses of this research, there exist several scenarios where it would confound unethical groups. One intended use case is to implement a common robotic infrastructure for autonomous robots owned by many different parties. Here the decentralised nature of the infrastructure would provide asymmetric robustness against hackers or governments seeking to unilaterally disrupt and destroy the infrastructure as they would not be a single point of failure.

In conclusion, there are many scenarios where this research may be misused to the detriment of humanity, yet we are not convinced that this research would be the most appropriate in those examples. Furthermore, given how this research seeks to defend against bad actors, we believe it is more likely to be a benefit to humanity.

Appendix A

First Appendix

Bibliography

- [1] Murai R, Ortiz J, Saeedi S, Kelly PHJ, Davison AJ. A Robot Web for Distributed Many-Device Localisation. CoRR. 2022;abs/2202.03314. Available from: <https://arxiv.org/abs/2202.03314>.
- [2] Farinelli A, Iocchi L, Nardi D. Multirobot Systems: A Classification Focused on Coordination. IEEE transactions on systems, man, and cybernetics Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society. 2004 11;34:2015-28.
- [3] Roldán-Gómez JJ, Barrientos A. Special Issue on Multi-Robot Systems: Challenges, Trends, and Applications. Applied Sciences. 2021;11(24). Available from: <https://www.mdpi.com/2076-3417/11/24/11861>.
- [4] Yan Z, Jouandeau N, Ali A. A Survey and Analysis of Multi-Robot Coordination. International Journal of Advanced Robotic Systems. 2013 12;10:1.
- [5] Stroupe A, Huntsberger T, Okon A, Aghazarian H, Robinson M. Behavior-based multi-robot collaboration for autonomous construction tasks. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems; 2005. p. 1495-500.
- [6] Burgard W, Moors M, Fox D, Simmons R, Thrun S. Collaborative multi-robot exploration. In: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065). vol. 1; 2000. p. 476-81 vol.1.
- [7] Pagello E, D'Angelo A, Montesello F, Garelli F, Ferrari C. Cooperative behaviors in multi-robot systems through implicit communication. Robotics and Autonomous Systems. 1999;29(1):65-77. Available from: <https://www.sciencedirect.com/science/article/pii/S0921889099000391>.
- [8] Kitano H, Asada M, Kuniyoshi Y, Noda I, Osawa E. RoboCup: The Robot World Cup Initiative. In: Proceedings of the First International Conference on Autonomous Agents. AGENTS '97. New York, NY, USA: Association for Computing Machinery; 1997. p. 340-347. Available from: <https://doi.org/10.1145/267658.267738>.
- [9] Kschischang FR, Frey BJ, Loeliger HA. Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory. 2001;47(2):498-519.
- [10] Ortiz J, Evans T, Davison AJ. A visual introduction to Gaussian Belief Propagation. CoRR. 2021;abs/2107.02308. Available from: <https://arxiv.org/abs/2107.02308>.
- [11] Solà J, Deray J, Atchuthan D. A micro Lie theory for state estimation in robotics. CoRR. 2018;abs/1812.01537. Available from: <http://arxiv.org/abs/1812.01537>.
- [12] Pelechrinis K, Iliofotou M, Krishnamurthy S. Denial of Service Attacks in Wireless Networks: The Case of Jammers. Communications Surveys & Tutorials, IEEE. 2011 06;13:245-57.
- [13] Douceur JR. The sybil attack. In: International workshop on peer-to-peer systems. Springer; 2002. p. 251-60.
- [14] Hidayab M, Ali AH, Abas Azmi KB. Wifi signal propagation at 2.4 GHz. In: 2009 Asia Pacific Microwave Conference; 2009. p. 528-31.
- [15] Gil S, Kumar S, Mazumder M, Katabi D, Rus D. Guaranteeing spoof-resilient multi-robot networks. Autonomous Robots. 2017;41(6):1383-400.

- [16] Kumar S, Gil S, Katabi D, Rus D. Accurate Indoor Localization with Zero Start-up Cost. In: Proceedings of the 20th Annual International Conference on Mobile Computing and Networking. MobiCom '14. New York, NY, USA: Association for Computing Machinery; 2014. p. 483–494. Available from: <https://doi.org/10.1145/2639108.2639142>.
- [17] Cortes J, Martinez S, Karatas T, Bullo F. Coverage control for mobile sensing networks. IEEE Transactions on Robotics and Automation. 2004;20(2):243-55.
- [18] Huang Y, Wang W, Wang Y, Jiang T, Zhang Q. Lightweight Sybil-Resilient Multi-Robot Networks by Multipath Manipulation. CoRR. 2019;abs/1912.04613. Available from: <http://arxiv.org/abs/1912.04613>.
- [19] Gupta D, Saia J, Young M. Peace Through Superior Puzzling: An Asymmetric Sybil Defense. In: 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS); 2019. p. 1083-94.
- [20] Thorndike L. Medieval Technology and Social Change. By Lynn White, jr..(New York: Oxford University Press. 1962. Pp. ix, 194, 10 plates. \$6.00.). Oxford University Press; 1962.