Title
Gordian

Journal
ACM Transactions on Cyber-Physical Systems, 4(4)

ISSN
2378-962X

Authors
Weber, Matthew
Jin, Baihong
Lederman, Gil
et al.

Publication Date
2020-10-31

DOI
10.1145/3386568

Peer reviewed

Gordian: Formal Reasoning-based Outlier Detection for Secure Localization

MATTHEW WEBER, BAIHONG JIN, and GIL LEDERMAN, University of California, Berkeley, USA
YASSER SHOUKRY, University of California, Irvine, USA
EDWARD A. LEE, SANJIT SESHIA, and ALBERTO SANGIOVANNI-VINCENTELLI, University of California, Berkeley, USA

Accurate localization from Cyber-Physical Systems (CPS) is a critical enabling technology for context-aware applications and control. As localization plays an increasingly safety-critical role, location systems must be able to identify and eliminate faulty measurements to prevent dangerously inaccurate localization. In this article, we consider the range-based localization problem and propose a method to detect coordinated adversarial corruption on anchor positions and distance measurements. Our algorithm, Gordian, rapidly finds attacks by identifying geometric inconsistencies at the graph level without requiring assumptions about hardware, ranging mechanisms, or cryptographic protocols. We give necessary conditions for which attack detection is guaranteed to be successful in the noiseless case, and we use that intuition to extend Gordian to the noisy case where fewer guarantees are possible. In simulations generated from real-world sensor noise, we empirically show that Gordian's trilateration counterexample generation procedure enables rapid attack detection even for combinatorially difficult problems.

CCS Concepts: • **Computer systems organization** → *Sensor networks*; *Reliability*;

Additional Key Words and Phrases: Secure localization, noisy sensor network localization, approximate graph embedding, structural rigidity, semidefinite programming, semidefinite relaxation, satisfiability modulo convex optimization

ACM Reference format:
Matthew Weber, Baihong Jin, Gil Lederman, Yasser Shoukry, Edward A. Lee, Sanjit Seshia, and Alberto Sangiovanni-Vincentelli. 2020. Gordian: Formal Reasoning-based Outlier Detection for Secure Localization. *ACM Trans. Cyber-Phys. Syst.* 4, 4, Article 43 (June 2020), 27 pages.
https://doi.org/10.1145/3386568

This work was supported in part by the TerraSwarm Research Center, one of six centers administered by the STARnet phase of the Focus Center Research Program (FCRP) a Semiconductor Research Corporation program sponsored by MARCO and DARPA; NSF award #1845194; Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST); the Industrial Cyber-Physical Systems Research Center (iCyPhy); and by the following companies: Denso, Ford, Siemens, and Toyota.
Authors' addresses: M. Weber, B. Jin, G. Lederman, E. A. Lee, S. Seshia, and A. Sangiovanni-Vincentelli, University of California, Berkeley, EECS, Berkeley, CA, 94720; emails: matt.weber@berkeley.edu, bjin@eecs.berkeley.edu, gilled@berkeley.edu, sseshia@eecs.berkeley.edu, alberto@berkeley.edu; Y. Shoukry, University of California, Irvine, EECS, Irvine, CA, 92697.

2378-962X/2020/06-ART43 $15.00
https://doi.org/10.1145/3386568

1 INTRODUCTION

Accurate real-world location data from Cyber-Physical Systems (CPS) are already crucial for control and location-based applications and stand to become even more important as systems with dynamic context-aware functionality become commonplace. However, many applications that depend on location data are subject to dangerous failure-modes when the CPS providing location information fails. For example in the hospital tracking context, a localization system failure could be life-threatening. Likewise, a platoon of military vehicles trying to localize themselves with respect to each other while some of the vehicles are being hijacked cannot afford location errors. Due to demonstrated vulnerabilities in Global Navigation Satellite System (GNSS) spoofing [Humphreys et al. 2008], an adversary may mislead or attempt to crash an autonomous vehicle or robot in a similar ploy.

Fortunately, many localization systems with sensors embedded in the environment resemble distributed sensor networks (e.g., Lazik et al. [2015]), where a measurement failure at a particular sensor may be isolated from other sensors. This distributed sensing can provide a great deal of redundancy in position estimation to aid in detecting and eliminating location attacks and coordinated sensor failures. We explore this scenario through the lens of range-based localization: Distance measurements are known between sensors, only some of which have known locations, and we seek to localize the entire network in the presence of adversarially corrupted measurements.

We propose Gordian,[1] an attack and coordinated sensor failure detection algorithm. Detecting and mitigating attacks on sensor data is, in general, a combinatorial problem [Pasqualetti et al. 2013], which has been typically addressed by either brute force search, suffering from scalability issues [Pasqualetti et al. 2013], or via convex relaxations using algorithms that can terminate in polynomial time [Fawzi et al. 2014] but are not necessarily sound. However, recent advances in combinatorial search techniques and, in particular, those in Satisfiability Modulo Theories (SMT) solvers, showed combinatorial problems can be cast into smaller problems that are solved efficiently. We demonstrate in this article that our algorithm modeled after these new Satisfiability Modulo Convex (SMC) solving techniques [Shoukry et al. 2017] are an effective way of solving range-based attack detection problems.

This article presents the following contributions:

- Sufficient topological and combinatorial conditions for attack detection in a noiseless network.
- Gordian, the first provably sound and complete coordinated attack detection algorithm over well-formed noiseless networks.
- A novel trilateration counterexample-generation procedure for Gordian's SMT solving architecture that makes combinatorially intractable attack detection problems practically solvable in reasonable time.
- A localization algorithm appropriate for noise on both distance measurements *and* anchor coordinates.
- A generalization of the standard graph-embeddability problems studied in noiseless localization, to a notion of *approximate embeddability*, appropriate for noisy distance measurements and anchor coordinates.
- A convex decision procedure for testing approximate embeddability of noisy networks at a desired confidence level, enabling the extension of Gordian to the noisy case.

[1] The name Gordian is a reference to a legend of Alexander the Great in which he "untied" the impossible Gordian Knot by slicing it in half with his sword.

In Section 2, we introduce conventional and secure localization algorithms, define our formal model of localization problems, and introduce rigidity theory. Next, in Section 3, we define our threat model, the attack detection problem, and identify an attack tolerance property for which we prove sufficient topological and combinatorial conditions. We present GORDIAN's architecture in Section 4, prove that the algorithm is sound and complete, and elucidate the algorithms for embeddability testing and counterexample generation. Section 5 extends localization, embeddability testing, and GORDIAN to the noisy case, which we evaluate in Section 6. We conclude in Section 7.

2 BACKGROUND AND RELATED WORK

Researchers have proposed a wide variety of localization schemes [Liu et al. 2007], including techniques as diverse as RF signal strength and fingerprinting [Honkavirta et al. 2009; Savvides et al. 2001], propagation time of an ultrasonic pulse [Lazik et al. 2015; Priyantha et al. 2000], and range-free techniques [He et al. 2003]. Many of these techniques assume complete trust of the entire localization system. The field of secure localization goes a step further to explore methods that work in the presence of malicious attacks.

2.1 Localization Networks

We apply a simple and common model [Anderson et al. 2010; Biswas and Ye 2004; Eren et al. 2004; Liu et al. 2008; Moore et al. 2004; Savvides et al. 2001; So and Ye 2007; Yang et al. 2013a] of the range-based localization problems, where Incomplete pairwise distance measurements are known between devices. Some devices, referred to as anchors, have known positions and other unlocalized devices do not.

More formally, we assume a set of n nodes representing sensors $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ with $S = \{1, 2, \ldots, n\}$ being their index set. $S_a \subseteq S$ represents anchor nodes, defined as nodes with *a priori* measured locations $p_a : S_a \to \mathbb{R}^2$. The rest, $S_x = S \setminus S_a$, have unspecified locations. An undirected graph $G = (S, E)$ is a natural model for the topology of a sensor network where the sensor nodes are treated as graph nodes and edges E represent measured internode distances. Let $E_a \subseteq E$ represent the edges (i, j) such that both $i, j \in S_a$ and $E_x = E \setminus E_a$. The weighted extension of G is given as $G_d = (S, E, W)$ where $W : E \to \mathbb{R}^+$. Combining all of the above, we formally define:

Definition 2.1 (Localization Network). A *localization network* N is a tuple (S, E, W, p_a) such that S is a set of nodes, E is a set of edges, $W : E \to \mathbb{R}^+$, $p_a : S_a \to \mathbb{R}^2$, where $S_a \subseteq S$.

Let the function $p : \mathcal{S} \to \mathbb{R}^2$, be a "placement," assigning coordinates $p(s_i) \in \mathbb{R}^2$ to each sensor node.

Now assume p^* is a placement representing ground truth positions of the nodes. The Euclidean distance from node s_i to s_j is given by $d_{ij} = \|p^*(s_i) - p^*(s_j)\|_2$. With $\restriction$ as the set restriction operator, we are now equipped to pose the *exact* (noiseless) *localization* problem as follows:

PROBLEM 2.2 (EXACT LOCALIZATION). *Given a localization network $N = (S, E, W, p_a)$ with an unknown ground truth placement p^* s.t. $W(e_{ij}) = d_{ij}$ and $p_a = p^* \restriction_{S_a}$, find p^*.*

Figure 1(a) shows an example localization network and placement that will be a long-running example in this article. It has six anchors (nodes $1 - 6$) represented by squares and one non-anchor (node 7) represented by a circle. Lines in the diagram represent measured internode distances, where the length of a measurement (i.e., $W(e_{xy})$) from s_x to s_y corresponds to the length of the line from node x to node y. The placement of the nodes is intended to correspond to their position on the page. In other words, this is an embedded graph, not an abstract graph representation. Lines from anchors to anchors are fainter than the lines to node 7, because we wish to draw attention to

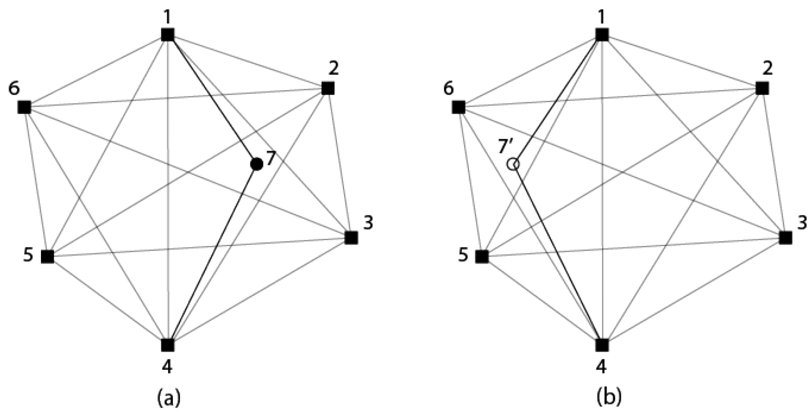

Fig. 1. Two different embeddings, (a) and (b), for the same localization network that is rigid but not GGR.

the latter. Figure 1(a) is an example of an embedding (also called a realization), because all the measurements and anchor positions are *consistent* with the placement p, i.e., $\forall e_{ij}\ d_{ij} = \|p(s_i) - p(s_j)\|_2$ and $\forall s_i \in S_a\ p_a(s_i) = p(s_i)$. We say a localization network N is consistent when it is embeddable, i.e., there exists a p that is consistent with respect to N. Were Figure 1(a) to be an *inconsistent* placement, one or more of the lines in the diagram would fail to meet up with a node at its endpoints or an anchor node would not be located at its designated position. This is the case in Figure 3. We use the terms "consistent" and "embeddable" interchangeably in this article.

2.2 Rigidity

Perhaps the most important question to ask about a localization network in the context of attack detection is whether or not p^* is the unique consistent placement with respect to N. This property is known as *unique localizability.*

If a localization network has two distinct embeddings in the absence of attack, an adversary does not have to take any action to create ambiguity in the localization result, as is the case for the two embeddings shown in Figure 1. The hollow circle in Figure 1(b) for node 7′ represents the alternative consistent placement for $p(7) \neq p^*(7)$. Even without attacks, there are two ambiguous localization solutions.

Eren et al. [2004] identified Global Generic Rigidity (GGR) as the link between sensor network localization, the mathematical concept of a framework, and the theory of structures from mechanical, civil engineering, and physics. Intuitively, a framework is a collection of solid rods connected at flexible joints. A rigid framework will not deform when perturbed in a planar direction. A GGR framework is not only immune to planar deformation, it cannot be discontinuously manipulated (such as the three-dimensional flip of node 7 along the line connecting nodes 1 and 4 in Figure 1) into another configuration. Eren et al. also proved a network N is uniquely localizable if and only if it has at least three anchors in general position and its corresponding grounded graph (the framework obtained by treating a localization network's graph of measurements, G, as rods) is GGR. For simplicity, we will write that N is GGR when its grounded graph is GGR [Eren et al. 2004].

A globally rigid framework can be identified by topological conditions: G is *3-connected* (at least three edges must be removed to partition G) and G is *redundantly rigid* (any one edge may be removed from G and the resulting G' is still rigid) [Jackson and Jordan 2003]. We further assume that the frameworks under discussion are generic, i.e., the coordinates of p are algebraically independent over the rationals. Essentially, the generic requirement forces the framework's p to not be entirely co-linear or otherwise degenerate. The GGR property is efficiently testable by a randomized algorithm [Gortler et al. 2010]. We refer the interested reader to Anderson et al. [2010] and Eren et al. [2004] for a more thorough presentation on rigidity and localization.

2.3 Localization Algorithms

Localization algorithms attempt to solve the exact localization Problem (2.2). From a complexity perspective, just determining if a graph has an embedding (that preserves the d_{ij}) is known to be NP hard [Saxe 1979]. Researchers tackle intractability through two broad classes of methods: local methods that enable each node to determine its location from its neighbors without seeing the big picture and global methods that simultaneously localize all nodes from a perspective external to the network. Gordian makes use of both.

The most basic local algorithm, provided by Eren et al. is *iterative trilateration*, but it is only possible for a specific kind of *trilateration graph*. The definition of a trilateration graph is given in Eren et al. [2004], but informally it can be thought of as a localization network that admits the following localization procedure: Begin with a set of three nodes $\{s_i, s_j, s_k\}$ with known locations (initially these can be anchor points). Find a node s_n that is currently without known location and is connected to three nodes s_i, s_j, and s_k in the known set. Draw three circles centered $p(i)$, $p(j)$, and $p(k)$ with radius d_{in}, d_{jn}, and d_{kn}, respectively. The value of $p(n)$ is given by the unique intersection of the circles. Add s_n to the known set, and repeat the above procedure until all nodes are in the known set.

Eren et al. prove trilateration graphs are uniquely localizable (an important property for Gordian's counterexamples phase—see Algorithm 2) and can be localized in polynomial time. However, the procedure is incomplete in the sense that it fails to localize classes of uniquely localizable graphs such as bipartite and wheel graph networks [Moore et al. 2004].

Alternatives to the node-centric localization methods discussed above use some sort of optimization framework to localize all nodes at once [Anderson et al. 2010; Biswas et al. 2006; Biswas and Ye 2004; So and Ye 2007]. Although actual formulations vary, these approaches frame localization as an optimization problem and (very broadly speaking) aim to minimize the sum of some sort of squared errors resembling

$$\underset{p(i),\, i \in S_x}{\operatorname{argmin}} \sum_{(i,j) \in E_x} \frac{\left| \|(p(i) - p(j)\|_2^2 - W(e_{ij})^2 \right|}{W(e_{ij})^2}$$

$$s.t. \quad \forall i \in S_a \; p(i) = p_a(i), \tag{2.1}$$

where $p(i) \in \mathbb{R}^2$ is a decision variable representing the estimated location of sensor i, and $W(e_{ij})$ is the measured distance between i and j. Since distances are squared, the denominator $W(e_{ij})^2$ normalizes the influence of large edges in optimization.

These methods commonly rely on a relaxation of the general optimization problem in Equation (2.1) from a non-convex program in two dimensions to a Semidefinite Program (SDP) in a higher dimensional space (refer to Equation (4.3) for the statement of the Biswas-Ye SDP relaxation (BY-SDP) used by Gordian). The relaxation was first proposed by Biswas and Ye [2004] with good empirical performance, then proved to have important theoretical connections to rigidity and unique localizability [So and Ye 2007]. Most significantly, it is shown [So and Ye 2007, Theorem 4.2] that a two-dimensional graph is uniquely localizable if and only if the max-rank BY-SDP solution is rank 2.

In this article, we seek to address the *attack detection* and *secure localization* problems with our Gordian algorithm. According to Zeng et al.'s survey [Zeng et al. 2013], secure localization methods in the literature fall into three broad categories: *prevention methods*, which prevent the sensor network from collecting bad data in the first place; *detection methods*, which identify and remove bad localization data; and *filtering methods*, which are robust to bad localization as part of the localization procedure. Under this taxonomy, Gordian is a centralized detection method designed to identify and eliminate distance-measurement attacks (i.e., attacks that corrupt inter-node distance measurements) and anchor position change attacks before localization.

Prevention schemes usually require special-purpose hardware on nodes, or specific ranging techniques. SeRLoc [Lazos and Poovendran 2005], for example, requires directional antennas. Other techniques assume the sensing mechanism used in ranging prohibits an adversary from shrinking range measurements. In the case of RF time-of-flight sensing, an adversary would have to break the speed of light to decrease the measured distance between nodes. This property, known as distance bounding, is the key requirement for Verifiable Multilateration [Capkun and Hubaux 2005] inspired techniques. Detection methods in the literature such as Liu et al. [2005] focus on identifying malicious nodes by catching them in a lie. In Liu et al.'s scheme, some sensors with known position pretend to be unlocalized. If the ranging information they receive from another node is incorrect, foul play is evident. Similarly, Liu et al. propose a voting-based scheme for distance measurement attack detection in which nodes vote to determine which measurements are consistent with their observations [Liu et al. 2008]. Filtering methods attempt to perform accurate localization in the presence of attacks. Li et al. [2005] propose localization via least median squares (LMS) instead of the more typical least squares (LS) approach. If attacks always appear as statistical outliers, Li et al.'s method will filter them out.

2.4 Related Statistical Methods

Statistical filtering techniques for localization (which are distinct from the above-mentioned filtering algorithms for secure localization) have recently been proposed for dynamic (i.e., mobile) systems [Freundlich et al. 2017; Ke Zhou and Roumeliotis 2011; Vander Hook et al. 2015]. Unlike the static (stationary) localization algorithms we have so far discussed, which assume all information is available at once, the dynamic problem domain allows for the iterative localization of mobile sensors/robots with new information appearing over time. One of the advantages of the dynamic formulation is the ability to actively move robots/sensors to maximize information gain [Ke Zhou and Roumeliotis 2011].

Some statistical filtering methods may be sufficiently general to be applicable in the static case as well. However, it is easy to see a statistical filtering algorithm for localization that attempts to minimize some notion of error at each time step becomes a solution to a variant of Problem (2.1) (which minimizes error for a single set of measurement data) if it only has one time step. Therefore, in the static secure localization context of this article, we consider the static application of a statistical filtering algorithm to be a particular type of solution to Problem (2.1), rather than a fundamentally different kind of localization algorithm.

It is, however, worth mentioning that some statistical filtering algorithms, such as Franco et al.'s [Di Franco et al. 2017], are tolerant to noise and outlier measurements, in ways both similar and different to a secure localization algorithm. Franco et al.'s approach for non-line-of-sight errors is to model measurement noise as a random variable with a Gaussian-mixture probability density function. As distance measurements are sampled over multiple time steps, the algorithm jointly infers noise parameters along with the localization result using a generalization of the classical technique of multidimensional scaling [Kruskal 1964]. While interesting, this algorithm is incomparable to Gordian in two respects: First, Gordian uses geometry in the static context to detect inconsistencies rather than the statistical patterns of measurements across multiple time steps. And second, Gordian assumes an adversarial rather than Gaussian-mixture attack model, meaning the kinds of attacks Gordian can deal with do not have to come from any particular distribution. It may be interesting to consider combining the two approaches in the future.

3 LOCALIZATION ATTACK DETECTION

As described in Section 2.2, even in the absence of attacks there are certain graph properties that must be present in the localization network for the localization problem to be well-posed. It should be no surprise then that stronger properties are required for the attack detection problem to be well-posed in the presence of attacks. Although conditions for the number of required non-malicious anchors are presented in Zhong et al. [2008], and rigidity conditions for outlier detection are addressed in Yang et al. [2013a, 2013b], this article is the first, to the best of our knowledge, to give conditions and a systematic procedure for attack detection in the presence of adversarially *coordinated* sensor failures.

3.1 Threat Model

We consider two agents: a *localization system* that attempts to solve Problem (2.2) and an *adversary* that interferes by corrupting some of the measurements seen by the localization system with the goal of causing the system to incorrectly localize one or more sensors to the wrong locations. Thus, the adversary has potentially tampered with the localization system's ranging measurements[2] $m_{ij} = W(e_{ij}) * (1 + \delta_{ij})$ and anchor positions $a_i = p_a(i) + \alpha_i$. The $\delta_{ij} \in \mathbb{R}$ and $\alpha_i \in \mathbb{R}^2$ values are controlled by the adversary.

For now, we assume a noiseless scenario: If an edge measurement m_{ij} (or an anchor position a_i) is attack-free or "clean," then $m_{ij} = d_{ij}$ ($a_i = p^*(i)$). Furthermore, we assume from the system's perspective there is no otherwise distinguishing feature between clean and corrupted edges. We formalize an attack as an *attack profile* $\hat{\eta} = (b, c, m, a)$ where b is the set of attacked edges, c the set of attacked anchors, m the new weights on the attacked edges, and a the new locations of attacked anchors.

A localization network under attack profile $\hat{\eta}$ is modified with these observed values in the expected way: The graph G_d is weighted by m_{ij} instead of d_{ij} and anchor positions are given by $p_a(i) = a_i$ instead of $p_a(i) = p^*(i)$. The localization network N under attack $\hat{\eta}$ will be denoted $\hat{\eta}(N)$, where $\hat{\eta}(\cdot)$ is the *application* of an attack profile to a localization network. For example, in Figure 2, the localization network N depicted in (a) has the correct distance between nodes 2 and 7: $||p(2) - p(7)||_2^2$. But (b) depicts an embedding of the attacked localization result $\hat{\eta}(N)$, where attack profile $\hat{\eta} = (\{(2, 7)\}, \emptyset, \{||p(2) - p(7')||_2^2\}, \emptyset)$.

3.2 Problem Formulation

In this section, we introduce formal notation for discussing attacks in a localization network. If the system can identify the attack and remove all corrupted data from the localization network it observes, the system is free to solve Problem (2.2) using an ordinary, insecure, localization algorithm. We name this prior task the *attack detection* problem.

Given the graph (S, E) of a localization network, we define an *attack hypothesis* $\eta = (b, c)$ where $b \subseteq E$, $c \subseteq S_a$. Intuitively, this is a guess as to which anchors and edges are under attack. Clearly, for each true attack, an attack profile $\hat{\eta} = (b, c, m, a)$ induces an attack hypothesis $\eta = (b, c)$ by simply "remembering" the identities of attacked edges/anchors and "forgetting" the values of the attack. A natural partial order is defined on attack hypotheses, where $(b, c) \preceq (b', c')$ iff $b \subseteq b'$ and $c \subseteq c'$ (as set inclusion). We will say that $\hat{\eta}$ (or η) is an $(\bar{s}, \bar{t})$*-attack profile (hypothesis)* if $|b| \leq \bar{s}$, $|c| \leq \bar{t}$.

[2]Ranging measurements are corrupted multiplicatively (not additively) to avoid cross terms between distance and error when squaring $W(e_{ij})^2$ in the noisy case (Section 5).

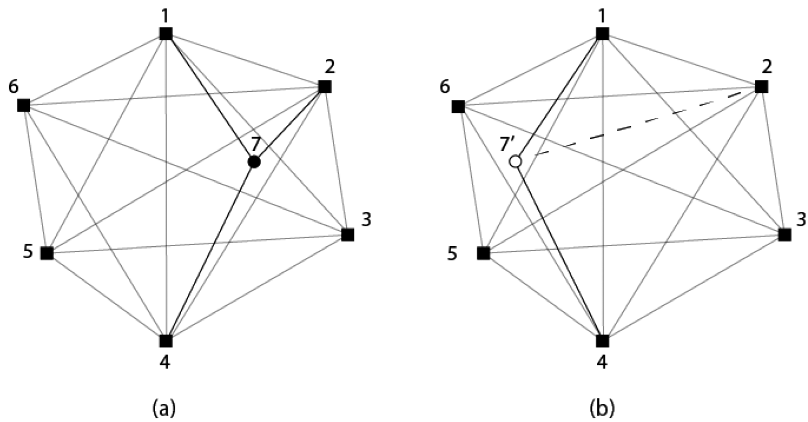

Fig. 2. An illustration of unique embeddings for a GGR localization network before attack (a) and after (b). The attack on edge (2,7), represented by the dashed line, still yields a consistent embedding in (b). This localization network is susceptible to undetectable attacks.

The space of all $(\bar{s}, \bar{t})$-attack profiles (hypotheses) for a localization network N will be denoted $\hat{H}_N(\bar{s}, \bar{t})$ $(H_N(\bar{s}, \bar{t}))$, or, if N is clear from the context, just $\hat{H}(\bar{s}, \bar{t})$ $(H(\bar{s}, \bar{t}))$. If $\hat{\eta}(N) = N$, we say that $\hat{\eta}$ is *trivial* with respect to N. It is trivial, because the application of $\hat{\eta}$ makes no difference to N.

Given a localization network N and an attack hypothesis η, we denote $N \setminus \eta$ the localization network with the hypothesized attacked edges and anchors removed. Since an attack profile $\hat{\eta}$ induces an attack hypothesis η, we will sometimes abuse notation and write $N \setminus \hat{\eta}$ for an attack profile $\hat{\eta}$, which means we remove from N the attack hypothesis induced by $\hat{\eta}$. We pose the attack detection problem for secure localization as follows:

Problem 3.1 (Noiseless Attack Detection). *For a localization network N and an attack profile $\hat{\eta}$, given only $\hat{\eta}(N)$, find a hypothesis $\zeta \in H(\bar{s}, \bar{t})$ such that $\eta \preceq \zeta$ (where η is the hypothesis induced by $\hat{\eta}$).*

The crux of successful attack detection is to make the attacks stand out in some way from the clean measurements of the localization network. If corrupted edges and anchors make up only a small part of a localization network they can be identified as the minimal part of the network that is incompatible with the rest. This forms the general idea behind voting-based attack detection schemes, e.g., Liu et al. [2008]. Drawing inspiration from Yang et al.'s rigidity-based outlier detection technique [Yang et al. 2013a], we use the *embeddability* of a localization network to define an $(\bar{s}, \bar{t})$-attack tolerant (abbreviated as $(\bar{s}, \bar{t})$-AT) localization network that admits incompatibility-based attack identification for up to $\bar{s}$ distance measurement attacks and $\bar{t}$ anchor position attacks.

Unlike other secure localization algorithms, our approach seeks to provably identify *all* attacks in networks meeting requirements we will lay out in Theorem 3.6. As such, $\bar{s}$ and $\bar{t}$ are assumptions that must be made in advance regarding the maximum number of attacks to search for in the network. To see why it is necessary to make such assumptions about the network, consider how attack detection is impossible if $\bar{s}$ and $\bar{t}$ are the size of the network. Consistency-based outlier detection would be impossible, because an attack detection algorithm could simply return the entire network as the attack, and nothing would actually be detected. Instead $\bar{s}$ and $\bar{t}$ are best interpreted as parameters representing assumptions made by an attack detection algorithm. If $\bar{s}$ and $\bar{t}$ are too small, an attack detection algorithm may miss attacks. Theorem 3.6 gives guidance on how large $\bar{s}$ and $\bar{t}$ may be in a given network for guaranteed attack detection success.

Definition 3.2 (Sub-localization Network). A localization network $N' = (S, E', W', p'_a)$ is a sub-localization network of $N = (S, E, W, p_a)$ (denoted $N' \subseteq N$) when $E' \subseteq E$, $W' = W \restriction E'$, and $p'_a = p_a \restriction S'_a$. If in addition, $|E \setminus E'| \leq \bar{s}$ and $|S_a \setminus S'_a| \leq \bar{t}$, we write $N' \subseteq_{\bar{s},\bar{t}} N$.

Given an attack profile (hypothesis) $\hat{\eta}$ on N, we define its restriction $\hat{\eta} \restriction_{N'}$ by simply removing any attacks on edges or anchors that are not in N'. By an abuse of notation, we will sometimes (as in Definition 3.3) write $\hat{\eta}(N')$ instead of $\hat{\eta} \restriction_{N'} (N')$. A sub-localization network $N' \subseteq N$ also induces an attack hypothesis, which we denote $\eta_{N'}$, such that $N' = N \setminus \eta_{N'}$. We now have the terminology to define a desirable property for a localization network that ensures consistency-based attack detection methods, such as Gordian, will be successful:

Definition 3.3 ($(\bar{s}, \bar{t})$-Attack Tolerance). A localization network N is $(\bar{s}, \bar{t})$-attack tolerant when for every $(\bar{s}, \bar{t})$ attack profile $\hat{\eta} \in \hat{H}(\bar{s}, \bar{t})$, for every $(\bar{s}, \bar{t})$ attack hypothesis $\eta \in H(\bar{s}, \bar{t})$, and for $N' = N \setminus \eta$ (in which case $N' \subseteq_{(\bar{s},\bar{t})} N$), the following condition is true: $\hat{\eta}(N')$ is consistent $\rightarrow \hat{\eta}$ is trivial with respect to N', i.e., $\hat{\eta}(N') = N'$.

Note that if $\hat{\eta}$ contains a non-trivial attack on an edge or anchor in N that has been removed from N' via η, the attack on that edge or anchor is trivial with respect to N'. Intuitively, the definition says that if we guess an incorrect $(\bar{s}, \bar{t})$ attack hypothesis to remove from N, but did not remove *all* of the non-trivial attacked edges and anchors, the application of the real (non-trivial) attack $\hat{\eta}$ to the resulting sub-localization network will necessarily leave it inconsistent. Such a network possesses enough redundancy in its edges and anchors that its clean information can "flag" the presence of a non-trivial attack profile through inconsistency, even if an incorrect attack hypothesis has been removed from the network. In other words, the only way to remove $\bar{s}$ edges and $\bar{t}$ anchors from the network to result in a consistent network is to remove the actual attack. Therefore, attacks in an $(\bar{s}, \bar{t})$-AT localization network can be identified by consistency checking. Clearly, GGR is an insufficient criterion, as shown in Figure 2(b) where the attacked localization network is still embeddable; attack tolerance requires something more.

3.3 Conditions for Attack Tolerance

We begin by first considering what it takes for a localization network to be $\bar{s}$-AT in the absence of anchors (and anchor attacks). In our analysis an $\bar{s}$-redundantly generically globally rigid (abbreviated as $\bar{s}$-GGR) network is a localization network that remains GGR after the removal of up to any $\bar{s}$ edges from its graph. Such a network may be identified by repeated use of Gortler et al.'s randomized algorithm [Gortler et al. 2010] on sub-localization networks.[3] We start with the following lemma about $\bar{s}$-GGR localization networks, illustrated by Figure 3:

Lemma 3.4. *Let N be a consistent $\bar{s}$-GGR localization network and $\hat{\eta} \in \hat{H}(\bar{s}, 0)$. $\hat{\eta}(N)$ is consistent if and only if $\hat{\eta}$ is trivial.*

Proof. Clearly, if $\hat{\eta}$ is trivial, $\hat{\eta}(N) = N$ is consistent. However, suppose $\hat{\eta}(N)$ is consistent. $\hat{\eta}(N) \setminus \hat{\eta}$ is also consistent (removing edges cannot make it any less consistent) and satisfies $\hat{\eta}(N) \setminus \hat{\eta} = N \setminus \hat{\eta}$ once the attack has been removed. Observe, $N \setminus \hat{\eta}$ is GGR, since we removed no more than $\bar{s}$ edges from N. It therefore has exactly one realization (up to isometry), which induces consistent edge weights on both N and $\hat{\eta}(N)$. If some edge of N or $\hat{\eta}(N)$ differed from the unique realization, that edge would be inconsistent. Therefore, $N = \hat{\eta}(N)$ and $\hat{\eta}$ must be trivial. □

The first part of this proof relies upon the obvious fact that making no modification to a consistent N results in a consistent N. The second part of the proof uses the intuition that if we were to

[3]There are $\binom{|E|}{(\bar{s})}$ sub-networks to consider with $\bar{s}$ edges removed. This can easily be done in parallel. If all are GGR, the original network is $\bar{s}$-GGR.

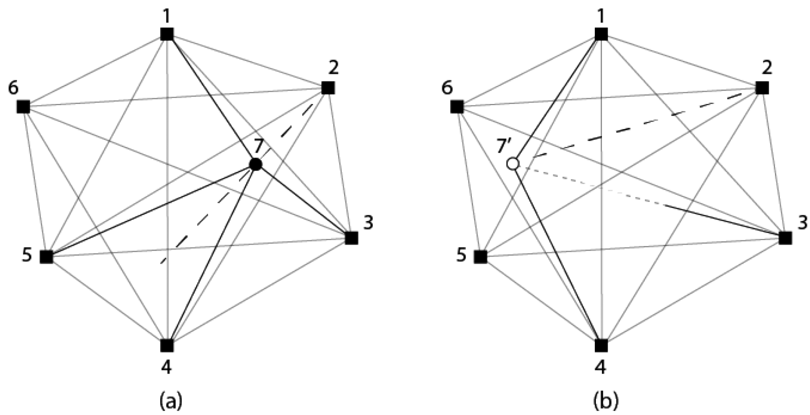

Fig. 3. Adding a redundant edge (3,7) to the example in Figure 2(b) satisfies Lemma 3.4. Neither the placement in (a) nor the placement in (b) is an embedding: (a) is incompatible with edge (2,7) and (b) is incompatible with (3,7).

remove the attacked edges from $\hat{\eta}(N)$, the resulting network would still be GGR and possess only a single, unique realization. This means the attacked edges we removed must either disagree with the unique realization (and be inconsistent) or agree with the unique realization (and be trivial).

Theorem 3.5. *If N is consistent and $2\bar{s}$-GGR, then N is $(\bar{s}, 0)$-AT.*

Proof. In accordance with Definition 3.3, let $\hat{\eta} \in \hat{H}(\bar{s}, 0)$ and $\eta \in H(\bar{s}, 0)$, respectively, be a typical attack profile and attack hypothesis on N. Also, let $N' = N \setminus \eta$. As per the condition in Definition 3.3, we must show a consistent $\hat{\eta}(N')$ implies a trivial $\hat{\eta}$. So assume $\hat{\eta}(N')$ is consistent. Observe N' is consistent, because removing edges from N does not affect consistency and that with $\bar{s}$ fewer edges, N' must be at least $\bar{s}$-GGR. Therefore, Lemma 3.4 applies to N' and $\hat{\eta}$, showing $\hat{\eta}$ is trivial under the assumption, which is the desired result. □

The proof of Theorem 3.5 is a direct application of Lemma 3.4 to the sub-localization network obtained by removing $\bar{s}$ edges from $\hat{\eta}(N)$. This is why the "$2\bar{s}$-GGR" property appears: $\hat{\eta}(N)$ must remain $\bar{s}$-GGR with $\bar{s}$ fewer edges for Lemma 3.4 to be relevant. The above results do not consider anchor attacks, but more importantly, they do not rely on the placement (or existence) of anchors at all and are true up to isometry. We can therefore state the following:

Theorem 3.6. *If N is consistent, $2\bar{s}$-GGR, and $|S_a| \geq 2\bar{t} + 3$, then N is $(\bar{s}, \bar{t})$-AT.*

Proof. Analogously to Theorem 3.5, consider an attack profile $\hat{\eta} \in \hat{H}(\bar{s}, \bar{t})$, attack hypothesis $\eta \in H(\bar{s}, \bar{t})$, and $N' = N \setminus \eta$. As before, we assume $\hat{\eta}(N')$ is consistent and seek to show $\hat{\eta}$ is trivial with respect to N'. We will separately demonstrate the triviality of edge attacks in $\hat{\eta}$ and anchor attacks in $\hat{\eta}$. We refer to the edge attacks in $\hat{\eta}$ as $\hat{\eta}_{\bar{s}}$ and the anchor attacks in $\hat{\eta}$ as $\hat{\eta}_{\bar{t}}$.

First consider edge attacks. Lemma 3.4 applies as in Theorem 3.5 to show $\hat{\eta}_{\bar{s}} \in \hat{H}(\bar{s}, 0)$ is trivial. Even without considering the anchor information in N, a non-trivial edge attack in $\hat{\eta}_{\bar{s}}$ violates our assumption of edge consistency for $\hat{\eta}(N')$.

Next consider anchor attacks. We have already established N' is consistent and has exactly one realization up to isometry (it is $\bar{s}$-GGR, and in particular, GGR). As such it would take 3 consistent anchors to uniquely localize N'. N has at least $2\bar{t} + 3$ anchors and N' with $N' \subseteq_{(\bar{s}, \bar{t})} N$ has at least $\bar{t} + 3$ anchors. As $\hat{\eta}$ contains no more than $\bar{t}$ anchor attacks, $\hat{\eta}_{\bar{t}}(N')$ must have at least 3 unattacked anchors. These 3 anchors establish a unique consistent embedding for $\hat{\eta}_{\bar{t}}(N')$.

Therefore, a non-trivial anchor attack in $\hat{\eta}_t$ violates the consistency assumption for $\hat{\eta}(N')$. Since $\hat{\eta}_{\overline{s}}$ and $\hat{\eta}_{\overline{t}}$ have been shown to be trivial, the full $\hat{\eta}$ must be trivial as well. □

The proof of Theorem 3.6 uses the "$2\overline{s}$-GGR" condition of Theorem 3.5 to ensure trivial edge attacks and introduces a new anchor condition to ensure N always has enough unattacked anchors left to uniquely localize the network after $\hat{\eta}$ and η. Similar to edge attacks in Lemma 3.4, anchor attacks in a uniquely localizable network must either be trivial or inconsistent with the network's unique embedding.

Theorem 3.6 gives sufficient conditions for attack tolerance. These conditions may be used as a guide for sensor network engineers to determine when a consistency-based attack detection algorithm like Gordian is guaranteed to be successful.[4] A full characterization of the necessary and sufficient conditions for $(\overline{s}, \overline{t})$-AT attack detection are beyond the scope of this article. The challenge of this characterization lies in evaluating the attack tolerance of a localization network that is not sufficiently redundantly GGR but makes up the difference with anchor information. For an extreme example, consider a sparsely connected localization network with no anchor attacks where every node is an anchor. Edge attacks are detectable but not by rigidity.

ALGORITHM 1: Attack Detection

procedure `AttackDetection` (`localization network` $\hat{\eta}(N)$, $\overline{s}$, $\overline{t}$)
  for $e_{ij} \in E$ and $a_k \in S_a$ **do**
    Declare pseudo-Boolean indicator variables b_{ij} and c_k // 1 represents corrupted, 0 represents clean
  end for
  $C \leftarrow (\sum_{(i,j)\in E} b_{ij} \leq \overline{s}) \wedge (\sum_{k \in S_a} c_k \leq \overline{t})$ // C is a set of pseudo-Boolean SAT clauses
  while Satisfiable(C) **do**
    `AttackHypothesis` $\zeta \leftarrow$ GetSatisfyingAssignment(C)
    (`TestResult`, `EdgeResidues`) $\leftarrow$ EmbeddabilityTest($\hat{\eta}(N) \setminus \zeta$)
    if `TestResult` = `IsEmbeddable` **then**
      return ζ // $\eta \leq \zeta$
    else
      `NewC` $\leftarrow$ GenCounterexamples ($\hat{\eta}(N) \setminus \zeta$, `EdgeResidues`)
      $C \leftarrow C \cup$ `NewC` $\cup (\bigvee_{(i,j)\in\zeta} b_{ij} \vee \bigvee_{k\in\zeta} c_k)$ // C on next iteration includes counterexamples and ζ
    end if
  end while
  return Failure

4 THE GORDIAN ALGORITHM

Gordian is an algorithm designed to efficiently perform attack detection on finite localization networks, guaranteed to succeed on $(\overline{s}, \overline{t})$-AT localization networks. Gordian's design is inspired by the design of Imhotep [Shoukry et al. 2015], Shoukry et al.'s lazy SMT solver for secure state estimation in the presence of attacks. Like Imhotep, Gordian identifies a combinatorial attack identification sub-problem that can be isolated from an otherwise convex optimization problem.

4.1 High-level Design

We present Gordian's high-level process for solving the attack detection Problem (3.1) in Algorithm 1. Gordian's main steps are also graphically depicted in Figure 4 and summarized here.

[4] Refer to Theorem A.1 in the Appendix for a proof of Gordian's soundness and completeness given an $(\overline{s}, \overline{t})$-AT localization network.

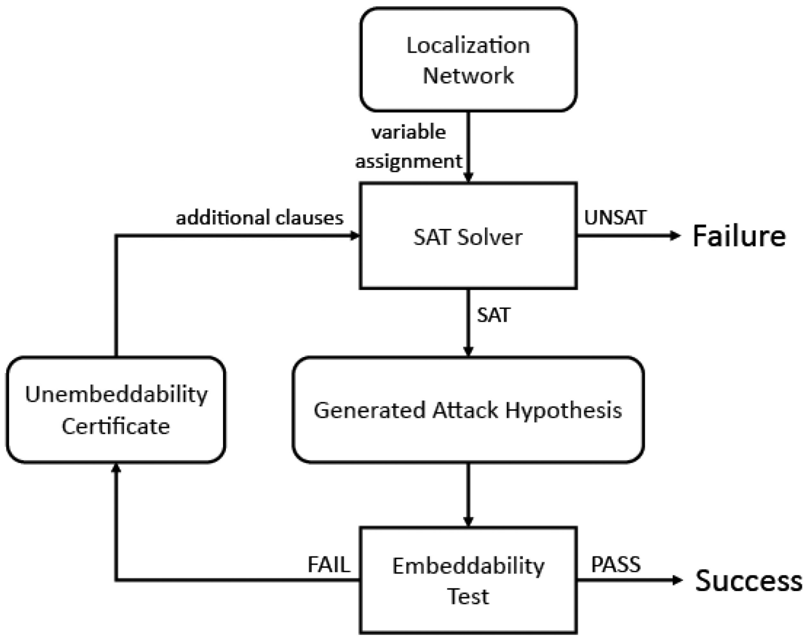

Fig. 4. An illustration of the steps and overall flow of Algorithm 1.

Given $(\overline{s}, \overline{t})$-AT localization network N, GORDIAN first assigns a Boolean variable to represent the corrupt/clean status for each of N's distance measurements and anchor positions. Next, GORDIAN assembles clauses made up of these variables to form a Boolean satisfiability (SAT) problem[5] in such a way that a satisfying assignment to the variables represents a plausible attack hypothesis. GORDIAN solves the SAT problem and tests the attack hypothesis ζ obtained as the SAT solution. This is accomplished by testing the consistency of $N' = N \setminus \zeta$, a network with the hypothesized attacked edges removed. If N' is consistent, by Definition 3.3 all remaining attacks are trivial and ζ may be returned as the Problem (3.1) solution. If N' is not embeddable, GORDIAN attempts to find small $N'' \subseteq N'$ that are also not embeddable. Algorithm 2 attempts to find these N'' and uses them to construct new SAT counterexample clauses that direct future iterations of GORDIAN to test attack hypotheses containing edges and anchors from N''. By Theorem A.1 in the Appendix, assuming a well-posed input, GORDIAN will always terminate in the "Success" zone of Figure 4 with a Problem (3.1) solution.

We walk through an iteration of GORDIAN's main loop on the long-running example[6] in Figure 3. GORDIAN first assigns Boolean variables to the anchors a_1, a_2, a_3, a_4, a_5, a_6 and the edges $e_{(1,2)}$, $e_{(1,3)}, \ldots e_{(4,7)}$. In our convention, the literal x refers to the positive occurrence of variable x and x' as its negation. The clauses, C are initialized with $e_{(1,2)} + e_{(1,3)} + \cdots e_{(4,7)} \leq \overline{s} = 1$ and $a_1 + a_2 + a_3 + a_4 + a_5 + a_6 \leq \overline{t} = 0$ to encode the assumption there are no more than $\overline{s}$ corrupted edges and no more than $\overline{t}$ corrupted anchors. The SAT solver immediately finds a satisfying assignment by setting every variable to false, signifying an empty attack hypothesis ζ. No edges or anchors are removed from $\hat{\eta}(N)$ on this iteration. An embeddability test is performed, and due to the

[5]Technically, the constraints on line 4 of Algorithm 1 make this a pseudo-Boolean satisfiability problem that is translatable into a Boolean SAT problem.
[6]The example in Figure 3 is just 1-GGR and not technically a well-formed input for $\overline{s} = 1$. GORDIAN can still run on such a localization network, just without the guarantee of correctness from Theorem 3.6.

inconsistency on edges connected to node 7, it fails. Gordian goes to the Trilateration Counterexamples (GenCounterexamples) step starting with a randomly selected high residue edge,[7] which happens to be (1,2). Gordian randomly selects edges (1,6) and (2,6) to complete the initial triangle. The induced sublocalization network for nodes 1, 2, and 6 is embeddable, so Gordian randomly selects node 7 to extend the counterexample. The induced sublocalization network for nodes 1, 2, 6, and 7 is still embeddable, so Gordian tries again by randomly selecting node 3. Finally, the induced sublocalization network for nodes 1, 2, 3, 6, and 7 is *not* embeddable, so Gordian learns the clause $(a_1 \vee a_2 \vee a_3 \vee a_6 \vee e_{(1,2)} \vee e_{(1,3)} \vee e_{(1,6)} \vee e_{(1,7)} \vee e_{(2,3)} \vee e_{(2,6)} \vee e_{(2,7)} \vee e_{(3,6)} \vee e_{(3,7)})$, which contains every edge measurement and anchor position in the counterexample. On the next iteration of SAT solving, the new clause will lead to an attack hypothesis that sets at least one of those variables to true, shrinking the search space for the true attack.

4.2 Attack Hypothesis Generation

Assume we begin with a finite $(\bar{s}, \bar{t})$-AT localization network N that is corrupted by $\hat{\eta} \in \hat{H}(\bar{s}, \bar{t})$ (denoted $\hat{\eta}(N)$) as input. To model an attack hypothesis ζ on $\hat{\eta}(N)$, Gordian assigns pseudo-Boolean indicator variables b_{ij} and c_k to the edges and anchors of its input. The attack hypothesis is a tuple of the edges and anchors for which the pseudo-Boolean variables were set to 1. More formally $\zeta = (\{e_{ij} : b_{ij} = 1\}, \{a_k : c_k = 1\})$. This model allows attack hypotheses to be generated by a pseudo-Boolean satisfiability (SAT) solver; Initially only given the constraint there are fewer than $\bar{s}$ and $\bar{t}$ attacks, but over time accumulating counterexample clauses learned from EmbeddabilityTest and GenCounterexamples.

4.3 Embeddability Test

We will show in this section how to frame the embeddability (and localization) problem as a convex program, relying on a key result in the literature [So and Ye 2007]: Uniquely Localizable (UL) localization networks with no attacks can always be localized in the plane using the (BY-SDP) technique [Biswas et al. 2006] presented below. We take the contrapositive of this result to identify inconsistent localization networks.

Let the unknown $p(i)$ for $i \in S_x$ be decision variables and define $X = [p(1), p(2), \ldots, p(|S_x|)] \in \mathbb{R}^{2\times|S_x|}$ as the matrix of decision variables obtained by stacking the first and second coordinates of the $p(i)$. Also, let $v_i \in \{0,1\}^n$ be a unit column vector whose ith component is 1 and all other components 0, and $a_j \in \mathbb{R}^2$ be the position of anchor node j.

We define $g_{ij} = [v_i - v_j \; 0]^\top$ if both s_i and s_j are sensors, and $g_{ij} = [v_i \; - a_j]^\top$ if either of s_i and s_j is an anchor. Now, the squares of sensor-sensor distances and sensor-anchor distances can be uniformly represented as

$$\|p(i) - p(j)\|^2 = \left| g_{ij}^\top \begin{bmatrix} \mathbf{X}^\top\mathbf{X} & \mathbf{X}^\top \\ \mathbf{X} & \mathbf{I}_2 \end{bmatrix} g_{ij} \right|, \tag{4.1}$$

where $\mathbf{I}_2$ is a 2×2 identity matrix. With this representation for the aggregate $\|p(i) - p(j)\|^2$, we can set up an optimization problem of the form in Equation (2.1). It is worthy to note that the sensor set and the anchor set vary with the attack hypothesis. A hypothetically attacked anchor

[7] An edge residue is defined in Problem (4.2). It can be interpreted as a measure of the difference between the nominal edge distance from $W(e_{ij})$ and the edge distance obtained in the embedding.

will be treated as a sensor with unknown location in the optimization problem.

$$\begin{aligned} \underset{\mathbf{X},\mathbf{Y}}{\text{minimize}} \quad & \sum_{(i,j)\in E} \underbrace{\frac{\left| g_{ij}^\top \begin{bmatrix} \mathbf{Y} & \mathbf{X}^\top \\ \mathbf{X} & \mathbf{I}_2 \end{bmatrix} g_{ij} - W(e_{ij})^2 \right|}{W(e_{ij})^2}}_{residue_{(i,j)}} \\ \text{subject to} \quad & \mathbf{Y} = \mathbf{X}^\top \mathbf{X}. \end{aligned} \tag{4.2}$$

We observe that the objective in Problem (4.2) is a summation over the residues on each edge (i, j) that will attain a minimum of zero, i.e., all residues are zeros, if there exists an embedding. However, Problem (4.2) is not convex, because the constraint $\mathbf{Y} = \mathbf{X}^\top\mathbf{X}$ restricts the rank of $\mathbf{Y}$ to be 2.

Biswas and Ye's solution [Biswas et al. 2006] to this dilemma is to lift the feasible set to a higher dimension by relaxing the constraint to $\mathbf{Y} \succeq \mathbf{X}^\top\mathbf{X}$ [Biswas and Ye 2004]. This yields the following SDP[8] solvable in a polynomial number of iterations by interior point methods [Alizadeh 1995]. We refer to this relaxation as the BY-SDP:

$$\begin{aligned} \underset{\mathbf{X},\mathbf{Y}}{\text{minimize}} \quad & \sum_{(i,j)\in E_x} \frac{|g_{ij}^\top \mathbf{Z} g_{ij} - W(e_{ij})^2|}{W(e_{ij})^2} \\ \text{subject to} \quad & \mathbf{Z} \doteq \begin{bmatrix} \mathbf{Y} & \mathbf{X}^\top \\ \mathbf{X} & \mathbf{I}_2 \end{bmatrix} \succeq 0. \end{aligned} \tag{4.3}$$

Conceptually, this relaxation allows the solver to localize each sensor in a higher-dimension space $\mathbb{R}^{|S_x|}$ instead of in $\mathbb{R}^2$ [Biswas et al. 2006]. If the residues are all zero and $\mathbf{Y}$ has rank 2, up to some numerical errors, we can certify that the localization network is embeddable in $\mathbb{R}^2$. If localization is desired, the component of $\mathbf{Z}$ corresponding to $\mathbf{X}$ can be read off as the projection of the high-dimensional solution back down to the plane of the anchors.

4.4 Trilateration Counterexamples

Like many verification algorithms, the worst-case time complexity of Gordian is huge in theory but much faster in practice. Without an efficient way to reduce the search space of attack hypotheses, Algorithm 1 reduces to a brute force search over all $\binom{|E|}{(\bar{s})} \cdot \binom{|S_a|}{(\bar{t})}$ candidates for ζ. Furthermore, each iteration testing an attack hypothesis requires solving a Boolean satisfiability problem (the canonical NP problem) and solving an SDP, which in the worst case cannot be solved in polynomial time, because it may require an exponential number of bits to express its solution [Alizadeh 1995]. Solvers for SAT and SDP problems are popular, because they make these theoretically intractable problems usually tractable in practice. Therefore, in our complexity analysis, we treat SAT and SDP solvers as constant time oracles for their respective problems.

Theorem 4.1. *Algorithm 1's worst-case time complexity is $O(|E|^{\bar{s}} \cdot |S_a|^{\bar{t}})$, assuming oracles for SAT and SDP.*

Proof. Setting up Algorithm 1 (lines 2–4) is linear in $|E| + |S_a|$. The main loop (lines 5–12) executes at most $\binom{|E|}{(\bar{s})} \cdot \binom{|S_a|}{(\bar{t})}$ times, contributing the $|E|^{\bar{s}} \cdot |S_a|^{\bar{t}}$ term. Each iteration of the loop solves a Boolean satisfiability problem (line 6) and an embeddability test SDP (line 7) that are solved by $O(1)$ oracles. The only other potentially expensive step is *GenCounterexamples* (line 11), which

[8]The normalizing factor $W(e_{ij})^2$ in the denominator is not explicitly given by Biswas and Ye, but it is implicitly equivalent to the arbitrary multiplicative weights in Biswas et al. [2006].

in Algorithm 2 performs a constant number of iterations of operations polynomial on $|E| + |S_a|$ and invokes the SDP oracle. □

In our experiments, the Boolean satisfiability problems posed on Algorithm 1, line 6, are easy to solve and make up a trivial amount of GORDIAN's execution time. While the SDP embeddability tests (Algorithm 1, line 7, and Algorithm 2, line 9) make up a large portion of GORDIAN's execution time, that time is more in line with interior point method's polynomially bounded number of iterations rather than the pathological examples that produce an exponential worst case.

The remaining high-complexity task that solvers cannot address is the brute force search over all $\binom{|E|}{\bar{s}} \cdot \binom{|S_a|}{\bar{t}}$ candidates for ζ. Fortunately, when Algorithm 1 finds $\hat{\eta}(N) \setminus \zeta$ unembeddable, often only a small $\hat{\eta}(N') \subset \hat{\eta}(N) \setminus \zeta$ causes unembeddability. If Algorithm 2 can find an $\hat{\eta}(N')$, Algorithm 1 can reduce its search space dramatically by learning a counterexample clause constructed by taking the "or" of the Boolean variables for nodes and edges in $\hat{\eta}(N')$. With the new clause, all attack hypotheses on future iterations must offer an explanation of why N' was unembeddable, focusing the search.

Our heuristic approach, presented in Algorithm 2, for finding small $\hat{\eta}(N')$ is motivated by the observation that high residue values from localization tend to occur in the *vicinity* of the attacks in the graph. As clean trilateration localization networks are GGR, Eren et al.'s iterative trilateration method [Eren et al. 2004] is the basis for Algorithm 2. Starting with three nodes, Algorithm 2 extends a candidate $\hat{\eta}(N')$ one node at a time until $\hat{\eta}(N')$ is determined to be unembeddable[9] or a maximum subgraph size (eight nodes in our implementation) is reached. We memorize calls to the EMBEDDABILITYTEST procedure to save runtime.

IMPORTANCESAMPLINGBYRESIDUE produces a starting point for a trilateration subgraph by weighting every edge in the graph by its residue, normalizing weights into a probability distribution summing to 1, and then sampling an edge from that distribution. IMPORTANCESAMPLINGBYCONNECTINGRESIDUES is a similar algorithm that samples nodes with at least three connections to the current trilateration subgraph. The weight assigned to a node is equal to the sum of edge residues on edges connecting it to the trilateration subgraph.

5 NOISY GORDIAN

Real-world sensors are imperfect, introducing small errors into their measurements that should not qualify as attacks. We now consider the Noisy Localization and Noisy Attack Detection problems where small errors are expected on uncorrupted values. With an extension of GORDIAN's embeddability test (Section 4.3) to this noisy case, we show how Noisy GORDIAN can be used to identify violations of sensor noise assumptions by swapping the noiseless embeddability test (Algorithm 1, line 7, and Algorithm 2, line 9) with the approximate embeddability test described in Section 5.2.

5.1 Noisy Localization Networks

To review our previous terminology, for localization network $N = (S, E, W, p_a)$, the Exact Localization Problem (2.2) presupposes the weights $W(e_{ij}) = d_{ij}$ and anchor positions $p_a = p^* \restriction_{S_a}$. An Attack Profile $\hat{\eta} = (b, c, m, a)$ substitutes $m_{ij} = d_{ij} * (1 + \delta_{ij})$ in place of weight $W(e_{ij})$ for $e_{ij} \in b$ and anchor positions $a_i = p^*(i) + \alpha_i$ for $p_a \in c$.

We now introduce Noise Profiles $\hat{\nu} = (b, c, m, a)$ as a specialized form of Attack Profile, where δ_{ij} and α_i are not arbitrarily determined by an adversary, but instead drawn from zero-mean probability distributions as $\delta_{ij} \sim \Theta_{ij}$ and $\alpha_i \sim \Phi_i$ with the symbol $\sim$ standing for the "is distributed

[9]Non-zero BY-SDP residues are sufficient for showing unembeddability.

ALGORITHM 2: Trilateration Counterexamples

procedure `GenCounterexamples` (`localization network` N, `EdgeResidues`)
  Parameters: `NumberOfIterations` and `MaxSubgraphSize`
  $C \leftarrow \emptyset$ // C is the set of counterexamples
  for 1 : `NumberOfIterations` **do**
    $(i, j) \leftarrow$ ImportanceSamplingByResidue(`EdgeResidues`)
    `ThirdNodeCandidates` $\leftarrow \{k : k \in S \& (i,k), (j,k) \in E\}$
    $k \leftarrow$ ImportanceSamplingByConnectingResidues(`ThirdNodeCandidates, EdgeResidues`)
    $N' \leftarrow$ NewSubLocalizationNetworkInducedBy$_N(\{i, j, k\})$
    while $|S'| <$ `MaxSubgraphSize` **do**
      if EmbeddabilityTest$(N') =$ `NotEmbeddable` **then**
        $C \leftarrow C \cup (\bigvee_{(i,j)\in E'} b_{ij} \vee \bigvee_{k \in S'_a} c_k)$ // At least one edge or anchor in N' is corrupted
        break
      end if
      `NextNodeCandidates` $\leftarrow \{n : n \in S, \exists\, i, j, k \in S' \text{ s.t. } (i,n), (j,n), (k,n) \in E\}$
      `n` $\leftarrow$ ImportanceSamplingByConnectingResidues(`NextNodeCandidates, EdgeResidues`)
      $N' \leftarrow$ NewSubLocalizationNetworkInducedBy$_N(S' \cup n)$
    end while
  end for
  return C

according to" relation. A noise profile may be applied to a noiseless localization network N as $\hat{\nu}(N)$ and may be composed with an (adversarial) attack profile $\hat{\eta}$ to represent a both corrupted and noisy localization network as $\hat{\eta}(\hat{\nu}(N))$.

Since our noise assumptions give anchor positions the possibility of error (like in GNSS sensors), we give the "softened" anchor constraints version of Equation (2.1). The new constant λ weights the relative significance of distance measurements against anchor positions in the objective. Coefficients weighting relative significance of observations in SDP localization problems date back to its formulation by Biswas et al. [2006]. In our experimental analysis, we somewhat arbitrarily use $\lambda = 5$, but this value could certainly be different in applications with less or more reliable anchors.

$$\underset{p(i),\, i\in S}{\text{minimize}} \sum_{(i,j)\in E} \frac{\left| \|(p(i) - p(j)\|_2^2 - W(e_{ij})^2 \right|}{W(e_{ij})^2} + \lambda \sum_{k\in S_a} \|(p(k) - p_a(k)\|_2^2. \tag{5.1}$$

Observe the change from S_x and E_x in Equation (2.1) to S and E in the first summation of Equation (5.1). S changes because anchor positions are now decision variables. E changes because inter-anchor measurements are useful when anchors can "float." When anchor positions were hard constraints, the inter-anchor distance measurements taken by sensors were irrelevant and hence excluded via E_x.

5.2 Approximate Embeddings

We cannot simply extend Problem (2.2) to the noisy case by treating $\hat{\nu}(N)$ as input in place of N, because even attack-free noisy localization networks are almost never consistent. As Anderson et al. observe, solving Problem (2.2) for a noisy GGR localization network is equivalent to finding the solution to an over-determined system of polynomial equations (the distance constraints), and such solutions do not in general exist [Anderson et al. 2010]. Therefore, we define noisy localization as the solution to the optimization problem in Equation (2.1).

Problem 5.1 (Noisy Localization). *Given $\hat{\nu}(N)$, where $\hat{\nu}$ is a noise profile and N is a localization network, find the solution to Equation (2.1).*

Framing noisy localization directly as an optimization problem is common in the literature (e.g., Biswas and Ye [2004], Savvides et al. [2001], and Wang et al. [2008]). In this article, as we consider anchor noise, we instead use Equation (5.1) and solve the similar Problem (5.2).

Problem 5.2 (Noisy Localization with Soft Anchors). *Given $\hat{v}(N)$, where $\hat{v}$ is a noise profile and N is a localization network, find the solution to Equation (5.1).*

Problems (5.2) and (5.1) are not given in terms of finding p^*; however, Anderson et al. show (a statement similar to) Problem (5.1) has useful properties such as a unique minimum for a uniquely localizable N when noise is sufficiently small [Anderson et al. 2010]. Therefore, a solution to Problems (5.2) and (5.1) can be thought of as an *approximate* solution to Problem (2.2).

We seek to generalize the notion of a noiseless embedding to an *approximate* embedding in an analogous way by examining residues. As a noiseless embedding has zero residues, an approximate embedding should have low residues. But how low?

Consider $N = (S, E, W, p_a)$ and a ground truth placement p^* s.t. $W(e_{ij}) = d_{ij}$ and $p_a = p^* \restriction_{S_a}$. In the remainder of this section, we explain how to set residue thresholds for approximate embeddability in such a way that p^* is an approximate embedding of $\hat{v}(N)$ at high confidence in the absence of an attack. If some N' were not approximately embeddable in this way, with high confidence $N' \neq \hat{v}(N)$, implying N' contains an attack.

We have already defined an edge $residue_{(i,j)}$ from Equation (4.2). Now with noise on anchors, we may also define an anchor $residue_i$ as $\|(p(k) - p_a(k)\|_2^2$. Note for both edges and anchors, residues are defined with respect to a particular placement p.

Definition 5.3 (γ-β-Embedding). A placement p is a γ-β-Embedding with respect to localization network N if

(1) $\sum_{(i,j)\in E} residue_{(i,j)} \leq \gamma_e$.
(2) $\sum_{k\in S_a} residue_k \leq \gamma_a$.
(3) $\forall (i,j) \in E\ residue_{(i,j)} \leq \beta_{ij}$.
(4) $\forall k \in S_a\ residue_k \leq \beta_k$.

Conditions (1) and (2) appear in the the objective function for noiseless embeddability testing with soft anchor constraints in Equation (5.1). The β parameter used for conditions (3) and (4) captures the idea that bounded noise distributions Θ and Φ should yield bounded residues. The choice of particular conditions for Definition 5.3 was motivated by the following observations about the residues at p^*:

Since $d_{ij}^2 = \|(p^*(i) - p^*(j)\|_2^2$, and d_{ij}^2 appears in every term of noisy $W(e_{ij})$, it can be factored out, which yields

$$residue_{ij} = \frac{\left|d_{ij}^2 - m_{ij}^2\right|}{m_{ij}^2} = \frac{\left|d_{ij}^2 - (d_{ij}^2(1+\delta_{ij})^2)\right|}{d_{ij}^2(1+\delta_{ij})^2} = \frac{\left|1 - (1+\delta_{ij})^2)\right|}{(1+\delta_{ij})^2}. \quad (5.2)$$

The same is true for $p^*(i)$ in

$$residue_k = \|p^*(k) - (p^*(k) + \alpha_k)\|_2^2 = \|\alpha_k\|_2^2. \quad (5.3)$$

These factored residue expressions contain no instances of d_{ij} nor p^*, yet represent the value of residues when $p = p^*$. Therefore, *it is not necessary to know p^* to calculate the residues at p^**. Noisy p^* residues may be determined directly from Θ and Φ.

These observations enable the selection of γ and β values G and B such that p^* is G-B-embeddable at high confidence. Monte Carlo simulation is a simple and effective method for settling on G and B values appropriate for Θ and Φ. The algorithm is straightforward: For a sufficiently large number of trials, sample δ and α values and compute $\sum_{(i,j)\in E} residue_{(i,j)}$ and $\sum_{k\in S_a} residue_k$. Looking at the histogram of residue sums across the trials, pick a high percentile (e.g., the 99.9th percentile) and use the corresponding values for G_e and G_a. The same algorithm works for B_e and B_a, and as we show in Section 6 is applicable to arbitrary Θ and Φ, such as data sets from real-world sensors.

5.3 Approximate Embeddability Test

We next show how an SDP formulation for noisy localization may be adapted to G-B-embeddability testing via a constraint satisfaction problem (CSP) with a semidefinite relaxation.

The BY-SDP procedure (4.3) was originally intended as a solution to Problem (5.1) for noisy localization networks and is empirically effective with noisy measurements [Biswas and Ye 2004]. We give the soft-anchor variant of the method for Problem (5.2) in Equation (5.4). This alternative formulation can be intuitively understood as treating anchor nodes as unlocalized nodes with an anchor residue term in the objective penalizing the anchor's distance from its nominal position. Formally, a modification must be made to matrix $\mathbf{X}$ by including anchor nodes as decision variables. Here $\mathbf{X} = [p(1), p(2), \ldots, p(|S|)] \in \mathbb{R}^{2\times|S|}$. Let $g_{ij} = [v_i - v_j \; 0]^\top$ *for all* s_i and s_j. Separate the other case of g from before to $f_i = [v_i \;\; -a_j]^\top$ for $s_k \in S_a$. The definitions of $\mathbf{Y}$ and $\mathbf{Z}$ are as before, but with respect to the new $\mathbf{X}$.

$$\underset{\mathbf{Z}\succeq 0}{\text{minimize}} \sum_{(i,j)\in E} \frac{|g_{ij}^\top \mathbf{Z} g_{ij} - W(e_{ij})^2|}{W(e_{ij})^2} + \lambda \sum_{k\in S_a} f_k^\top \mathbf{Z} f_k \tag{5.4}$$

The first summation expresses inter-node residues changed to sum over E in place of E_x, because inter-anchor measurements provide information when anchor positions are not hard constraints. The second summation expresses anchor residues $(\|(p(i) - p_a(i)\|_2^2)$. As before, the semidefinite relaxation of Z changes this from a nonconvex rank-constrained problem to an SDP problem. To obtain the location estimate, read $\mathbf{X}^*$ off the minimizer $\mathbf{Z}^*$.

The important thing about Equation (5.4) from an approximate embeddability testing perspective is that it contains expressions for each edge and anchor residue. This is everything we need to test the G-B-embeddability of network N:

$$\sum_{(i,j)\in E} \frac{\left|g_{ij}^\top \mathbf{Z} g_{ij} - W(e_{ij})^2\right|}{W(e_{ij})^2} \leq G_e \tag{5.5a}$$

$$\sum_{k\in S_a} f_k^\top \mathbf{Z} f_k \leq G_a \tag{5.5b}$$

$$\frac{\left|g_{ij}^\top \mathbf{Z} g_{ij} - W(e_{ij})^2\right|}{W(e_{ij})^2} \leq B_{ij}, \forall (i,j) \in E \tag{5.5c}$$

$$f_k^\top \mathbf{Z} f_k \leq B_k, \forall k \in S_a, \tag{5.5d}$$

If the above constraints are not feasible with respect to any $\mathbf{Z} \succeq 0$, we can then conclude that the given N is not G-B-embeddable; however, if the constraints are feasible, we cannot be sure whether N is truly G-B-embeddable in two dimensions or simply appears as such due to the relaxation of the rank constraint on Z.

5.4 Noisy Attack Detection

Noisy attack detection generalizes from the noiseless case.

Problem 5.4 (Noisy Attack Detection). *For a localization network N, noise profile $\hat{v}$, and attack profile $\hat{\eta}$, given only $\hat{\eta}(\hat{v}(N))$, find a hypothesis $\zeta \in H(\bar{s}, \bar{t})$ such that $\hat{\eta}(\hat{v}(N)) \setminus \zeta$ is G-B-embeddable.*

We implemented Noisy Gordian by replacing the embeddability test in Section 4.3 with Monte Carlo calculation of G and B and the CSP[10] in Equation (5.5). As such, the algorithm for Noisy Gordian is Algorithm 1 only with a noisy embeddability test at Algorithm 1, line 7, and Algorithm 2, line 9. Noisy Gordian solves Problem (5.4) up to the SDP relaxation of its embeddability test. By the selection of parameters G and B, such a hypothesis exists (when $\zeta = \hat{\eta}$) with high confidence for $\hat{\eta}(\hat{v}(N)) \setminus \zeta$, and lines 7 and 8 of Algorithm 1 ensure Noisy Gordian will not terminate until it has found one. We classify non-trivial attacks (larger in magnitude than plausible noise from $\hat{v}$) into two categories: *G-B-effective* and *G-B-compatible*.

- ***G-B*-Effective Attacks**: $\hat{\eta}$, yielding G-B-unembeddable $\hat{\eta}(\hat{v}(N))$.
- ***G-B*-Compatible Attacks**: non-trivial $\hat{\eta}$, yielding G-B-embeddable $\hat{\eta}(\hat{v}(N))$.

When Noisy Gordian solves Problem (5.4) it identifies G-B-effective attacks. Any unidentified G-B-compatible attacks are limited in size by the uncorrupted part of the localization network: If p is a G-B-embedding of $\hat{\eta}(\hat{v}(N))$, p must also be a G-B-embedding of $\hat{\eta}(\hat{v}(N)) \setminus \hat{\eta}$. This is obvious with respect to Definition 5.3 where all conditions true for the residues of $\hat{\eta}(\hat{v}(N))$ must also hold for the subset of those residues in $\hat{\eta}(\hat{v}(N)) \setminus \hat{\eta}$. Therefore, "$G$-$B$-compatible" attacks can only be as bad as the worst G-B-embeddable p of $\hat{\eta}(\hat{v}(N)) \setminus \hat{\eta}$, which the adversary does not influence.

Put simply, for geometric configurations or noise conditions where you would not expect a "good" noisy localization result, you should not expect a "good" attack detection result (i.e., with limited G-B compatible attacks). In the world of navigation systems, Geometric Dilution of Precision (GDP) [Langley 1999] describes how the geometry of noisy range measurements can affect localization precision for a single sensor. Even for GGR localization networks, poor geometry on measurements may result in poor localization precision and poor attack detection. The theoretical aspects of this subject are worthy of further research, but for now, we estimate the role of geometry and network topology empirically in our experimental evaluation by searching for maximum size G-B-compatible attacks on our benchmarks.

6 RESULTS

As the correctness of Gordian and related algorithms are proved in the previous sections, our empirical evaluation is focused on evaluating performance. In this section, we evaluate the runtime of noiseless and noisy Gordian.

6.1 Experimental Setup

We implemented Gordian in Matlab 2016b using the Yalmip toolbox [Löfberg 2004] to model the Semidefinite Programming (SDP)s. Our implementation uses MOSEK [Mosek 2011] and SAT4J's pseudo-Boolean solver [Le Berre and Parrain 2010] as the underlying SDP and Boolean Satisfiability Problem (SAT) solvers. Our testing platform has a quad-core Intel Core i7-4700MQ CPU and 16 GB memory.

[10] Equation (5.5) is a constraint satisfaction problem and does not produce meaningful residues as expected by line 7 of Algorithm 1 in Equation (5.5). As a workaround, we generate residues by first solving a noisy localization problem (5.4) whenever residues are required.

We generated benchmark graphs by randomly dropping nodes onto a 15-by-15 grid and connecting those points with inter-node distances within a fixed connection radius.[11] Attacks consist of $\bar{s}$ randomly selected edges and $\bar{t}$ randomly selected anchors. In each benchmark, $2\bar{t} + 3$ nodes are randomly determined to be anchors.

6.2 Noiseless Gordian

Our goal in experimental evaluation of the Noiseless Gordian algorithm is to evaluate the efficiency of Gordian with noiseless embeddability tests on localization networks with random attacks. We selected benchmarks for our evaluation with the competing goals of demonstrating Gordian's performance on a range of problem sizes while also systematically evaluating the effects of changing $\bar{s}$ and $\bar{t}$ parameters on a medium-size problem. The result for each benchmark is listed in Table 1.

In addition to the BY-SDP noiseless embeddabilty test, we also evaluate a (potentially faster) noiseless embeddability test built from Wang et al.'s alternative *edge*-centric (ESDP) relaxation of Equation (4.2).

Like the BY-SDP's rank condition for embeddability, Theorem 4.2 from Wang et al. [2008] asserts that in the absence of attack, when the diagonal elements of $\mathbf{Y} - \mathbf{X}^\top \mathbf{X}$ are zero, all nodes have been localized to their true locations. Since an unembeddable localization network cannot simultaneously localize nodes to their true locations and achieve zero residues, the combination of zero residues and zero $trace(\mathbf{Y} - \mathbf{X}^\top \mathbf{X})$ certifies an embeddable ESDP solution.

In our trials of Gordian, BY-SDP vs ESDP embeddability tests usually result in roughly comparable runtimes, in line with the expectation of the SDP trending faster on dense localization networks with fewer nodes and the ESDP trending faster on sparse localization networks with more nodes [Wang et al. 2008]. Benchmarks 31 and 36 show instances where the lower quality ESDP relaxation achieved a much faster runtime than the BY-SDP. Gordian's counterexample generation procedure is random and weighted heuristically by residues. We suspect the particular residues generated by the BY-SDP on these trials directed the counterexample generation process to consider particularly unproductive counterexamples. On other benchmarks such as 17 and 22, the situation was reversed and ESDP residues were particularly unhelpful relative to BY-SDP residues.

Numerical errors and solver inaccuracies were significant challenges for implementation of embeddability tests in our noiseless experiments where picking thresholds for "nearly zero" and "nearly rank two" required fine-tuning. However, these concerns are dwarfed by measurement error in the noisy case, which we consider for the remainder of our evaluation.

6.3 G-B Embeddability

We identified two options for relating additive SurePoint noise ($m_{ij} = d_{ij} + \delta'_{ij}$) to the multiplicative noise model ($m_{ij} = d_{ij}(1 + \delta_{ij})$) assumed for Equation (5.2): Obtain an equivalent $\delta_{ij} = \delta'_{ij}/d_{ij}$ or directly estimate d_{ij} as $d_{ij} = m_{ij} - \delta'_{ij}$ to calculate residue estimates from $residue_{ij} = |d^2_{ij} - m^2_{ij}|/m^2_{ij}$ and the m_{ij} values on hand. We use this last approach in our evaluation to determine G values from Monte Carlo simulation.

[11] Due to scaling by $W(e_{ij})$ in the denominator of edge residues, a short edge with large noise can overpower the other residues. To prevent this problem in our evaluation, we enforce a minimum measurement length of 1 m before noise and 0.3 m after noise.

Table 1. Gordian Runtime on BenchMarks (BM) Using Either BY-SDP or ESDP Embeddability Tests in the Noiseless Case, Maximum Undetectable Attacks in the Noisy Case, and Noisy Runtime with Accuracy

BM	#Nodes	#Edges	$\overline{s}$	$\overline{t}$	BY-SDP (s)	ESDP (s)	$+\delta'_{ij}$	$-\delta'_{ij}$	$\|\alpha_i\|$	Noisy (s)	Correct	Nearly Correct
1	20	176	1	2	65.87	81.67	2.1	2.8	1.6	142.40	100%	0%
2	20	170	2	2	73.54	89.65	2.2	1.9	1.4	199.23	100%	0%
3	20	170	2	3	66.22	64.74	1.9	2.0	1.5	197.77	100%	0%
4	20	160	1	3	61.01	50.87	4.5	*	1.5	152.96	80%	20%
5	20	171	4	2	74.96	91.53	2.8	1.7	1.7	354.38	60%	0%
6	20	167	4	3	74.06	93.39	1.9	2.9	1.7	*	0%	40%
7	30	310	0	1	24.91	36.64	*	*	1.7	61.93	100%	0%
8	30	354	0	2	21.67	36.53	*	*	1.7	76.18	100%	0%
9	30	350	0	3	30.54	50.11	*	*	1.5	109.47	100%	0%
10	30	324	0	4	45.84	61.68	*	*	1.4	167.29	100%	0%
11	30	289	1	0	41.60	90.04	2.6	*	*	60.93	100%	0%
12	30	345	1	1	41.01	53.03	1.8	2.3	1.7	201.3	100%	0%
13	30	316	1	2	54.95	62.84	2.0	2.3	1.9	154.14	80%	20%
14	30	282	1	3	55.96	110.51	1.9	2.3	1.5	237.22	100%	0%
15	30	359	1	4	124.58	67.56	4.4	*	1.5	1,259.79	40%	60%
16	30	327	2	0	77.01	98.16	1.9	2.8	*	160.36	100%	0%
17	30	312	2	1	69.78	1,338.69	2.0	2.3	1.7	143.38	100%	0%
18	30	324	2	2	51.21	80.55	2.1	2.2	1.9	*	0%	100%
19	30	318	2	3	58.98	85.61	2.1	1.9	1.6	599.25	80%	0%
20	30	327	2	4	69.59	153.50	2.0	2.1	1.7	418.84	100%	0%
21	30	353	3	0	90.81	61.95	2.1	2.0	*	80.64	40%	0%
22	30	316	3	1	53.39	5,160.47	1.9	2.0	3.8	129.16	100%	0%
23	30	328	3	2	61.23	81.90	2.0	2.3	1.8	377.02	60%	40%
24	30	322	3	3	79.79	95.43	2.1	1.9	1.6	445.48	80%	0%
25	30	341	3	4	155.56	282.91	1.6	1.6	1.5	696.68	100%	0%
26	30	358	4	0	51.92	113.87	2.1	2.1	*	157.69	80%	0%
27	30	352	4	1	110.54	143.50	2.0	1.9	1.6	423.38	100%	0%
28	30	359	4	2	78.94	155.10	2.0	2.4	1.6	278.66	100%	0%
29	30	369	4	3	163.39	208.32	2.4	2.3	1.6	480.68	100%	0%
30	30	372	4	4	106.03	240.63	2.2	2.0	1.6	521.96	100%	0%
31	30	397	7	0	2,140.79	211.23	2.1	1.8	*	218.74	80%	0%
32	30	411	6	0	104.62	174.55	2.8	2.2	*	446.83	80%	0%
33	30	381	6	2	190.13	259.04	1.9	2.1	1.8	811.40	80%	0%
34	30	368	0	5	46.00	76.81	*	*	1.6	142.25	100%	0%
35	30	372	2	7	160.67	201.96	2.3	2.0	1.4	645.57	80%	0%
36	40	716	5	7	1,186.59	454.86	2.2	2.0	1.5	2,671.00	100%	0%
37	40	723	5	3	267.82	382.80	1.9	2.0	1.6	1,786.09	100%	0%
38	40	501	2	2	145.65	142.42	2.0	1.9	1.6	422.64	100%	0%
39	60	908	2	2	404.21	303.91	2.0	3.6	1.8	1,398.83	60%	40%
40	70	1,070	2	2	722.18	754.97	2.0	1.3	1.7	1,383.09	80%	0%
41	80	1,258	2	2	366.71	319.39	2.5	*	1.6	969.74	80%	0%

Noiseless runtimes are averages over six trials. For comparison, a brute-force (no trilateration counterexamples) BY-SDP trial took 3,193.20 seconds on benchmark 1 and over 72 hours on benchmark 2 without terminating. δ' and α values represent maximum undetectable attacks. A * in the table is a placeholder for an impossible to collect value, either because there were no attacked edges/anchors in the benchmark or all attacked edges in the benchmark were too short to determine an effective negative attack. Noisy runtimes are averages over the fully correct runtimes found over five trials. Nearly correct trials had at most one misidentified attack. Benchmarks 6 and 18 had no fully correct (all attacks identified) runtimes and hence no data point. Our explanation for this is below.

The fourth section of Table 1 addresses the performance of the *G-B* embeddability test[12] from Equation (5.5). As discussed in Section 5.4, noise levels and the particular geometry of a benchmark determine whether an attack is *G-B* compatible (not detectable) or *G-B* effective (detectable). As we can only evaluate the runtime of GORDIAN when it has effective attacks to detect, we first set out to estimate the minimum size of an effective attack on our benchmarks.

This was accomplished by starting with the attack-free and noise-free underlying N of a benchmark, applying SurePoint and anchor noise $\hat{\nu}(N)$, arbitrarily selecting a single attacked edge/attacked anchor and corrupting it with a positive edge attack ($m_{ij} = d_{ij} + \delta'_{ij}$), a negative edge attack ($m_{ij} = d_{ij} - \delta'_{ij}$), or an anchor attack ($a_i = p_a(i) + \alpha_i$) to produce $\hat{\eta}(\hat{\nu}(N))$. Starting at the noise bound, we incrementally increased the magnitude of the attack ($|\delta'_{ij}|$ or $\|\alpha_i\|$) until the resulting $\hat{\eta}(\hat{\nu}(N))$ became not *G-B*-Embeddable. The results in the fourth section of Table 1 are averages across five trials of this process. This methodology produced useful estimates for the minimum size of a *G-B*-effective attack in these benchmarks given our noise conditions.

6.4 Noisy Gordian

For noisy trials, we model Θ with data obtained from Lab 11, the authors of the SurePoint Localization System [Kempke et al. 2016]. SurePoint uses DecaWave DW1000 ultra-wideband transceivers to achieve highly accurate ranging data in a real-world environment. SurePoint achieves a median *additive* error of 0.08 m with −0.59 m and 0.31 m as 95th percentile errors after proccessing.

A noise profile $\hat{\nu}$ was determined in the following way: Anchors were moved in a uniformly random direction a distance sampled from a truncated normal distribution clipped at 1.0 with mean 0 and standard deviation 0.2. Bounded additive noise was sampled uniformly at random from the SurePoint data set after eliminating the 405 out of 73,414 outlier[13] values of noise with absolute error value greater than 0.7. We used these thresholds (1.0 and 0.7) to calculate B_{ij} and B_i values and Monte Carlo simulation to determine the 99.9th percentile of residue sums to set G_e and G_a values for *G-B* Embeddability and Noisy Gordian experiments.

As discussed, noisy GORDIAN is only able to detect *G-B*-effective attacks. Therefore, for each edge attack, we set $\delta'_{ij} \in [5, 6]$ and for each anchor attack $\|\alpha_i\|_2 \in [5, 6]$ so all attacks would likely be over the threshold of *G-B* effectiveness. Edge attacks were given a 50% chance of being positive or negative.[14] In this way, $\hat{\eta}$ was fixed for each benchmark. We generated five noise profiles $\hat{\nu}$ and for each profile evaluated a trial of Noisy GORDIAN on each $\hat{\eta}(\hat{\nu}(N))$ of our benchmarks.

The final section of Table 1 contains the results of this experiment. The average runtimes across trials yielding fully correct attack detection results are presented in the first column. This allows a nearly apples-to-apples comparison against runtimes in the noiseless case where correct attack detection results are always produced. Overall, Noisy GORDIAN is significantly slower than corresponding noiseless problems but is still much faster than brute force.

In the majority of trials, Noisy GORDIAN successfully identified the full attack. When Noisy GORDIAN failed in our experiments, either it confused *no more than one* uncorrupted edge or anchor with a corrupted edge or anchor, or the *G-B*-embeddability test misclassified an embeddable sublocalization network as unembeddable. In the latter case, Noisy GORDIAN learns a counter example

[12]We implemented and evaluated an ESDP-style approximate embeddability test, too, but determined the ESDP's weaker relaxation has too many false negatives, i.e., localization networks that only appear embeddable under the ESDP relaxation, to justify its use.

[13]We eliminate outliers from the noise distribution to ensure that the only outliers in our experiments are the attacks we deliberately apply to our benchmarks.

[14]If a negative attack is impossible because d_{ij} is small and edges cannot have negative length, it is changed to a positive attack.

Table 2. Noisy GORDIAN (Noisy) Compared to Voting Attack Detection Algorithm

BM	#Nodes	#Edges	$\bar{s}$	$\bar{t}$	Noisy (s)	Voting (s)	(G) Misses	(V) Misses	(G) False Detections	(V) False Detections
1	20	176	1	2	142.40	254.51	0	1	0	21.2
2	20	170	2	2	199.23	173.48	0	0	0	6
3	20	170	2	3	197.77	174.17	0	0.4	0	6.4
4	20	160	1	3	152.96	214.54	0.2	0.6	0.2	16.8
5	20	171	4	2	354.38	302.25	0	0.4	0	24
6	20	167	4	3	*	370.53	1	1.2	1	36.4
7	30	310	0	1	61.93	180.25	0	0	0	0
8	30	354	0	2	76.18	202.30	0	0	0	0
9	30	350	0	3	109.47	238.20	0	0.2	0	6
10	30	324	0	4	167.29	458.24	0	0.6	0	32.2
11	30	289	1	0	60.93	298.85	0	0	0	15.8
12	30	345	1	1	201.3	203.39	0	0	0	0
13	30	316	1	2	154.14	305.91	0.2	1	0.2	16.2
14	30	282	1	3	237.22	275.36	0	0.8	0	16.8
15	30	359	1	4	1,259.79	446.46	0.6	0.6	0.6	32.6
16	30	327	2	0	160.36	207.40	0	0	0	0
17	30	312	2	1	143.38	567.32	0	0.2	0	44.4
18	30	324	2	2	*	432.84	1	0.4	1	32.2
19	30	318	2	3	599.25	266.04	0	0.8	0	11.4
20	30	327	2	4	418.84	393.54	0	1	0	30
21	30	353	3	0	80.64	244.48	0	0	0	0
22	30	316	3	1	129.16	350.41	0	1	0	20.6
23	30	328	3	2	377.02	639.00	0.4	0.4	0.4	48.4
24	30	322	3	3	445.48	1,076.90	0	1.4	0	100
25	30	341	3	4	696.68	336.31	0	1	0	18.2
26	30	358	4	0	157.69	924.76	0	0	0	76.8
27	30	352	4	1	423.38	255.20	0	0	0	0
28	30	359	4	2	278.66	571.74	0	0	0	40.8
29	30	369	4	3	480.68	392.10	0	0.6	0	9.8
30	30	372	4	4	521.96	770.83	0	0.8	0	64.4
31	30	397	7	0	218.74	411.49	0	0	0	2.2
32	30	411	6	0	446.83	571.55	0	0	0	19.8
33	30	381	6	2	811.40	1,186.01	0	0.4	0	75
34	30	368	0	5	142.25	351.32	0	0.8	0	14.4
35	30	372	2	7	645.57	571.86	0	0.8	0	35.6
36	40	716	5	7	2,671.00	849.41	0	0.6	0	22
37	40	723	5	3	1,786.09	1,288.64	0	0.2	0	52.6
38	40	501	2	2	422.64	464.87	0	0.6	0	8.2
39	60	908	2	2	1,398.83	4,126.23	0.4	1.6	0.4	180.2
40	70	1,070	2	2	1,383.09	3,307.42	0	0.8	0	91.6
41	80	1,258	2	2	969.74	5,406.52	0	0.4	0	97.6

Benchmarks and Noisy runtimes are the same as in Table 1. Results are averages over five trials. "Misses" for Noisy GORDIAN (G) and Voting (V) are given as the average count (over non-error runs) of true attacked edges and anchors missing from attack detection results. "False Detections" for Noisy Gordian (G) and Voting (V) are given as the average count (over non-error runs) of detected edges and anchors missing from true attack detection results.

clause that makes the SAT problem on line 5 of Algorithm 1 unsatisfiable (which is always a risk considering the 99.9% confidence intervals on G). This was what happened with benchmarks 6, 19, 21, 24, 26, 31, 32, and 33. By repeating the tests for benchmark 6 with a bigger (99.99%) confidence interval, the results improved to 0% correct and 80% nearly correct. In summary, *over all experiments, Noisy* GORDIAN*either produced a correct result, a nearly correct result with a single error, or failed while indicating no result could be found at all.*

6.5 Comparison to Voting-based Attack Detection Algorithms

We implemented Liu et al.'s anchor attack detection algorithm from Liu et al. [2005] and Liu et al.'s Enhanced Greedy Algorithm (EARMMSE) for edge attack detection from Liu et al. [2008] for comparison against Noisy Gordian. Our implementation of EARMMSE used the noisy embeddability test presented in this article. We refer to the combination of these two algorithms (anchor attack detection before edge attack detection) as "Voting."[15] We selected these algorithms for comparison against Noisy Gordian, because they are two prominent secure localization algorithms from the literature that together offer a solution to Problem (5.4) and are therefore more directly comparable to Noisy Gordian than other prevention or filtering algorithms (referring to the taxonomy for secure localization discussed in Section 2.3) from the literature.

The results of this comparison over the noisy benchmarks are given in Table 2. On all but the largest benchmarks, average runtimes for Voting were roughly comparable to Noisy Gordian. We also evaluated the average number of false negative edges and anchors (misses) and false positive edges and anchors (false detections) reported by the algorithms.[16] The results show Noisy Gordian modestly outperformed Voting on misses and significantly outperformed Voting on false detections. All missed detections we observed from Voting were the result of missed anchors. In our experiments, the EARMMSE algorithm either quickly found and eliminated the true attacked edges or removed huge chunks of the network until it eventually removed the correct edge. The most dramatic instance we observed of EARMMSE removing edges from the network was a run of benchmark 41 in which 470 edges, more than a third of the entire network, were detected and eliminated as candidate attacks.

7 CONCLUSION

We have presented Gordian, an attack detection algorithm at the distance-graph abstraction level. In the noiseless case, we proved Gordian sound and complete for $(\bar{s}, \bar{t})$-AT input. By generalizing localization network embeddability testing to approximate embeddability testing in the noisy case, we likewise extend Gordian to Noisy Gordian. We evaluate our algorithms on noisy attack detection problems constructed from the error distribution of a real-world localization system, demonstrating the practical utility of our formal reasoning–based outlier detection scheme.

Gordian and Noisy Gordian consistently finished combinatorially difficult attack detection benchmarks with a high success rate on the order of minutes instead of days. These algorithms are appropriate as a periodic check to identify bad actors in a long-running sensor network or general network of cyber-physical systems. Once adversarial data are removed, localization can be made more trustworthy and more accurate.

A APPENDIX

Theorem A.1. *Given a finite $(\bar{s}, \bar{t})$-AT localization network $N = (S, E, W, p_a)$ and a ground truth placement p^* s.t. $W(e_{ij}) = d_{ij}$ and $p_a = p^* \restriction_{S_a}$ and $\hat{\eta} \in \hat{H}(\bar{s}, \bar{t})$, Algorithm 1 is a sound and complete procedure for solving Problem (3.2) when given input $\hat{\eta}(N)$.*

Proof. (Soundness) The algorithm returns with an attack hypothesis $\eta_{N'}$ only if $\hat{\eta}(N')$ is consistent. But, since N was $(\bar{s}, \bar{t})$-AT, that means by definition that $\hat{\eta}$ is trivial with respect to N', and so $\hat{\eta} \subseteq \eta_{N'}$.

[15] Liu et al. refer to a different algorithm in Liu et al. [2008] as "Voting," but we feel this is an appropriate and evocative name for two algorithms that rely upon a consensus between uncorrupted nodes to detect attacks.

[16] This analysis is not averaged over Noisy Gordian runs, which reported an error instead of a result. As discussed in Section 6.4, such trials can be rerun with a higher confidence interval on G.

(Completeness) Let η be the attack hypothesis induced by $\hat{\eta}$. If on line 4 we choose an $(\bar{s}, \bar{t})$-attack hypothesis $\zeta \geq \eta$, $\hat{\eta}(N) \setminus \zeta$ is consistent and the algorithm ends. Denote by C_i the set of clauses at iteration i and let $SAT(C_i)$ be the set of satisfiable assignments to C on iteration i. Since C_0 simply bounds the number of attacks, clearly $\eta \in SAT(C)$. C_{i+1} is constructed $C_{i+1} \supseteq C_i$ (line 12) to forbid a reassignment on future iterations matching ζ (line 12 last term). Specifically, $\zeta \in SAT(C_i)$, but $\zeta \notin SAT(C_{i+1})$, and $SAT(C_{i+1}) \subset SAT(C_i)$. $SAT(C_0)$ is finite, and so, we just have to show that if $\eta \in SAT(C_i)$ and the algorithm does not end, $\eta \in SAT(C_{i+1})$.

On every iteration where the algorithm does not end, we add clauses to C of the form $(\bigvee_{(i,j)\in \mathsf{E}} b_{ij} \vee \bigvee_{k \in \mathsf{Sa}} c_k)$ for some inconsistent sub-localization network. At least one of the b_{ij}, c_k must be in the true attack hypothesis η or the sub-localization network would be consistent (the unattacked localization network was consistent, and removing edges or anchors can not make it less consistent). And so η satisfies the new added clause. □

ACKNOWLEDGMENT

Special thanks to Lab 11 for sharing their SurePoint localization data.

REFERENCES

Farid Alizadeh. 1995. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.* 5, 1 (Feb. 1995), 13–51. DOI: https://doi.org/10.1137/0805002

Brian D. O. Anderson, Iman Shames, Guoqiang Mao, and BariÅ Fidan. 2010. Formal theory of noisy sensor network localization. *SIAM J. Disc. Math.* 24, 2 (Jan. 2010), 684–698. DOI: https://doi.org/10.1137/100792366

P. Biswas, T. C. Liang, K. C. Toh, Y. Ye, and T. C. Wang. 2006. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Trans. Automat. Sci. Eng.* 3, 4 (Oct. 2006), 360–371. DOI: https://doi.org/10.1109/TASE.2006.877401

Pratik Biswas and Yinyu Ye. 2004. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*. ACM Press, 46. DOI: https://doi.org/10.1145/984622.984630

Srdjan Capkun and J.-P. Hubaux. 2005. Secure positioning of wireless devices with application to sensor networks. In *Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, Vol. 3. IEEE, 1917–1928.

Carmelo Di Franco, Amanda Prorok, Nikolay Atanasov, Benjamin Kempke, Prabal Dutta, Vijay Kumar, and George J. Pappas. 2017. Calibration-free network localization using non-line-of-sight ultra-wideband measurements. In *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'17)*. ACM Press, 235–246. DOI: https://doi.org/10.1145/3055031.3055091

Tolga Eren, O. K. Goldenberg, Walter Whiteley, Yang Richard Yang, A. Stephen Morse, Brian D. O. Anderson, and Peter N. Belhumeur. 2004. Rigidity, computation, and randomization in network localization. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 4. IEEE, 2673–2684.

Hamza Fawzi, Paulo Tabuada, and Suhas Diggavi. 2014. Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Trans. Automat. Control* 59, 6 (June 2014), 1454–1467. DOI: https://doi.org/10.1109/TAC.2014.2303233

Charles Freundlich, Soomin Lee, and Michael M. Zavlanos. 2017. Distributed active state estimation with user-specified accuracy. *Arxiv:1706.01576 [cs]* (June 2017).

Steven J. Gortler, Alexander D. Healy, and Dylan P. Thurston. 2010. Characterizing generic global rigidity. *Amer. J. Math.* 132, 4 (2010), 897–939.

Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek Abdelzaher. 2003. Range-free localization schemes for large scale sensor networks. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*. ACM, 81–95.

Ville Honkavirta, Tommi Perala, Simo Ali-Loytty, and Robert Piche. 2009. A comparative survey of WLAN location fingerprinting methods. In *Proceedings of the 6th Workshop on Positioning, Navigation and Communication (WPNC'09)*. IEEE, 243–251.

Todd E. Humphreys, Brent M. Ledvina, Mark L. Psiaki, Brady W. O'Hanlon, and Paul M. Kintner Jr. 2008. Assessing the spoofing threat: Development of a portable GPS civilian spoofer. In *Proceedings of the ION GNSS International Technical Meeting of the Satellite Division*, Vol. 55. Retrieved from https://gps.mae.cornell.edu/humphreys_etal_iongnss2008.pdf.

Bill Jackson and Tibor Jordan. 2003. *Connected Rigidity Matroids and Unique Realizations of Graphs*. Technical Report. Eotvos University, Budapest, Hungary.

Ke Zhou and S. I. Roumeliotis. 2011. Multirobot active target tracking with combinations of relative observations. *IEEE Trans. Rob.* 27, 4 (Aug. 2011), 678–695. DOI: https://doi.org/10.1109/TRO.2011.2114734

Benjamin Kempke, Pat Pannuto, Bradford Campbell, and Prabal Dutta. 2016. SurePoint: Exploiting ultra wideband flooding and diversity to provide robust, scalable, high-fidelity indoor localization. ACM Press, 137–149. DOI: https://doi.org/10.1145/2994551.2994570

J. B. Kruskal. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29, 1 (Mar. 1964), 1–27. DOI: https://doi.org/10.1007/BF02289565

Richard B. Langley. 1999. Dilution of precision. *GPS WORLD* (1999). https://www.semanticscholar.org/paper/Dilution-of-Precision-Langley/c04cd3dc971734ebe2ba1aa113e417e8b28b248c.

Patrick Lazik, Niranjini Rajagopal, Oliver Shih, Bruno Sinopoli, and Anthony Rowe. 2015. ALPS: A Bluetooth and ultrasound platform for mapping and localization. ACM Press, 73–84. DOI: https://doi.org/10.1145/2809695.2809727

Loukas Lazos and Radha Poovendran. 2005. SeRLoc: Robust localization for wireless sensor networks. *ACM Trans. Sens. Netw.* 1, 1 (2005), 73–100.

D. Le Berre and A. Parrain. 2010. The Sat4j library, release 2.2: System description. SAT 7, 2–3 (2010), 59–64. DOI: 10.3233/SAT190075

Zang Li, Wade Trappe, Yanyong Zhang, and Badri Nath. 2005. Robust statistical methods for securing wireless localization in sensor networks. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*. IEEE Press, 12.

Donggang Liu, Peng Ning, and Wenliang Du. 2005. Detecting malicious beacon nodes for secure location discovery in wireless sensor networks. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*. IEEE, 609–619.

Donggang Liu, Peng Ning, An Liu, Cliff Wang, and Wenliang Kevin Du. 2008. Attack-resistant location estimation in wireless sensor networks. *ACM Trans. Inf. Syst. Secur.* 11, 4 (July 2008), 1–39. DOI: https://doi.org/10.1145/1380564.1380570

Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. 2007. Survey of wireless indoor positioning techniques and systems. *IEEE Trans. Syst., Man Cyber., Part C (Applic. Rev.)* 37, 6 (Nov. 2007), 1067–1080. DOI: https://doi.org/10.1109/TSMCC.2007.905750

Johan Löfberg. 2004. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the IEEE International Symposium on Computer Aided Control Systems Design*. IEEE, 284–289.

David Moore, John Leonard, Daniela Rus, and Seth Teller. 2004. Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*. ACM, 50–61.

Mosek. 2011. *MOSEK Related Publications*. Technical Report. Retrieved from https://download.mosek.com/docs/reports/publications.pdf.

Fabio Pasqualetti, Florian Dorfler, and Francesco Bullo. 2013. Attack detection and identification in cyber-physical systems. *IEEE Trans. Automat. Control* 58, 11 (Nov. 2013), 2715–2729. DOI: https://doi.org/10.1109/TAC.2013.2266831

Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. 2000. The cricket location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*. 32–43.

Andreas Savvides, Chih-Chieh Han, and Mani B. Srivastava. 2001. Dynamic fine-grained localization in ad-hoc wireless sensor networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom'01)*. 166–179. https://doi.org/10.1145/381677.381693

J. B. Saxe. 1979. *Embeddability of Weighted Graphs in K-space is Strongly NP-hard*. Carnegie-Mellon University, Dept. of Computer Science. https://books.google.com/books/about/Embeddability_of_Weighted_Graphs_in_K_sp.html?id=vClAGwAACAAJ.

Yasser Shoukry, Pierluigi Nuzzo, Alberto Puggelli, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia, Mani Srivastava, and Paulo Tabuada. 2015. IMHOTEP-SMT: A satisfiability modulo theory solver for secure state estimation. In *Proceedings of the 13th International Workshop on Satisfiability Modulo Theories (SMT'15)*.

Yasser Shoukry, Pierluigi Nuzzo, Alberto Sangiovanni-Vincentelli, Sanjit A. Seshia, George J. Pappas, and Paulo Tabuada. 2017. SMC: Satisfiability modulo convex optimization. In *Proceedings of the 10th International Conference on Hybrid Systems: Computation and Control (HSCC'17)*.

Anthony Man-Cho So and Yinyu Ye. 2007. Theory of semidefinite programming for sensor network localization. *Math. Prog.* 109, 2–3 (Jan. 2007), 367–384. DOI: https://doi.org/10.1007/s10107-006-0040-1

Joshua Vander Hook, Pratap Tokekar, and Volkan Isler. 2015. Algorithms for cooperative active localization of static targets with mobile bearing sensors under communication constraints. *IEEE Trans. Rob.* 31, 4 (Aug. 2015), 864–876. DOI: https://doi.org/10.1109/TRO.2015.2432612

Zizhuo Wang, Song Zheng, Yinyu Ye, and Stephen Boyd. 2008. Further relaxations of the semidefinite programming approach to sensor network localization. *SIAM J. Optim.* 19, 2 (Jan. 2008), 655–673. DOI: https://doi.org/10.1137/060669395

Zheng Yang, Lirong Jian, Chenshu Wu, and Yunhao Liu. 2013a. Beyond triangle inequality: Sifting noisy and outlier distance measurements for localization. *ACM Trans. Sens. Netw.* 9, 2 (Mar. 2013), 1–20. DOI: https://doi.org/10.1145/2422966.2422983

Zheng Yang, Chenshu Wu, Tao Chen, Yiyang Zhao, Wei Gong, and Yunhao Liu. 2013b. Detecting outlier measurements based on graph rigidity for wireless sensor network localization. *IEEE Trans. Vehic. Technol.* 62, 1 (Jan. 2013), 374–383. DOI: https://doi.org/10.1109/TVT.2012.2220790

Yingpei Zeng, Jiannong Cao, Jue Hong, Shigeng Zhang, and Li Xie. 2013. Secure localization and location verification in wireless sensor networks: a survey. *J. Supercomput.* 64, 3 (June 2013), 685–701. DOI: https://doi.org/10.1007/s11227-010-0501-4

Sheng Zhong, Murtuza Jadliwala, Shambhu Upadhyaya, and Chunming Qiao. 2008. Towards a theory of robust localization against malicious beacon nodes. In *Proceedings of the 27th Conference on Computer Communications*. IEEE, 1391–1399. Retrieved from http://ieeexplore.ieee.org/abstract/document/4509792/.

Received June 2018; revised August 2019; accepted January 2020