Gordian SMT: Untangling Ranging Attacks in Noisy Sensor Networks for Secure Localization

Matthew Weber
Baihong Jin
Gil Lederman
Yasser Shoukry
Edward A. Lee
Sanjit A. Seshia
Alberto L. Sangiovanni-Vincentelli

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2017-5
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-5.html

March 17, 2017

Acknowledgement

This work was supported in part by the TerraSwarm Research Center, one of six centers supported by the STARnet phase of the Focus Center Research Program (FCRP) a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

Gordian SMT: Untangling Ranging Attacks in Noisy Sensor Networks for Secure Localization

Matthew Weber, Baihong Jin, Gil Lederman, Yasser Shoukry[*]
Edward Lee, Sanjit Seshia, Alberto Sangiovanni-Vincentelli
University of California, Berkeley

ABSTRACT

Accurate localization is a critical enabling technology for sensor networks and context awareness in the Internet of Things. As localization plays an increasingly safety-critical role in applications, engineers must have confidence in the validity of location data. In this paper we consider the sensor network localization problem with noisy distance measurements and propose a method to detect adversarially corrupted values. Our algorithm, GORDIAN SMT, rapidly finds attacks on distance measurements by identifying geometric inconsistencies at the graph level without requiring assumptions about hardware, ranging mechanisms or cryptographic protocols. We give the necessary and sufficient conditions for which attack detection is guaranteed to be possible in the noiseless case, and present GORDIAN SMT as a sound and complete algorithm for well-posed noiseless input. We extend GORDIAN SMT to the case of noisy measurements where our empirical analysis shows good performance at a run-time several orders of magnitude faster than the naive brute force algorithm.

1. INTRODUCTION

Accurate real-world location data are already crucial input for cyber-physical systems and stand to become even more important in the coming Swarm [18] of ubiquitous computing devices. Location-based services with dynamic functionality for users in different places are an exciting possibility for the Swarm, but are subject to dangerous failure-modes when location information is incorrect. Consider a home automation system complete with smart locks that automatically unlocks the doors when it detects a resident is approaching the house from the outside. A malicious adversary could break into the home if he/she were to trick the system into thinking the resident is outside when in fact the resident is away, or lock the resident outside by "transporting" the resident somewhere distant when he/she is trying to legitimately enter the house. An adversary might crash a driverless car with a similar ploy.

Fortunately, in a ubiquitous computing world there are many sensors embedded in the environment, and many sources of localization data. If used to their full capacity, these sources could provide a great deal of redundancy in position estimation to aid in preventing location-attacks. We explore this scenario in this paper through the lens of sensor network localization: distance measurements are known between Swarm devices (sensors), only some of which have known locations, and we seek to localize the entire network in the presence of corrupted measurements.

We propose GORDIAN SMT[1], an attack detection and localization method , to rapidly identify inconsistencies in localization data. Detecting and mitigating attacks on sensor data is, in general, a combinatorial problem [25], which has been typically addressed by either brute force search, suffering from scalability issues [25], or via convex relaxations using algorithms that can terminate in polynomial time [8] but are not necessarily sound. On the other side, recent advances in combinatorial search techniques and in particular those used in Satisfiability Modulo Theories (SMT) solvers showed how combinatorial problems can be cast into smaller problems that can be solved efficiently. However, current state-of-art SMT solvers cannot handle those problems that arise in the area of localization in sensor networks.

GORDIAN SMT is an attack detection algorithm at the distance-graph abstraction level that requires no custom hardware, no special nodes, and no specific ranging techniques. We empirically show it to be orders of magnitude faster than the brute force equivalent. The

[*] Also affiliated with University of California, Los Angeles

[1] The name GORDIAN SMT 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.

ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

algorithm is sound and complete for a class of noiseless secure localization problems we define in this paper, and can be extended with useful results to the noisy domain.

In Section 2 we introduce conventional and secure localization algorithms, define our formal model of a sensor network and introduce rigidity theory. Next in Section 3 we identify and prove the necessary and sufficient conditions for Gordian SMT's correct behavior in the noiseless case. We present Gordian SMT's architecture in Section 4 and elucidate the algorithms for localization graph embeddability testing and counterexample generation. We extend Gordian SMT to the noisy case in Section 5, prove the soundness of noisy embeddability testing, and discuss the noisy conditions needed for correct Gordian SMT performance. An empirical evaluation is given in Section 6 and we conclude in Section 7.

2. BACKGROUND AND RELATED WORK

Researchers have proposed a wide variety of localization methods [22] including techniques as diverse as RF signal strength and fingerprinting [11, 27], propagation time of an ultrasonic pulse [16, 26], and range-free techniques [10]. 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 Secure Localization in Sensor Networks

According to Zeng et al.'s survey of secure localization [33], 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 SMT is a centralized detection method designed to identify and eliminate range-change attacks (i.e. attacks that corrupt inter-node distance measurements) at the location determination step.

Prevention schemes usually require special-purpose hardware on nodes, or specific ranging techniques. SeRLoc [17] for example requires directional antennas. Other techniques assume the sensing mechanism used in ranging prohibits an adversary from shrinking range measurements (such as RF time of flight). This property, known as distance bounding, is the key requirement for Verifiable Multilateration [6] inspired techniques.

Detection methods in the literature such as [20] 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.

Filtering methods attempt to perform accurate localization in the presence of attacks. Li et al. [19] 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. Similarly, Liu et al. propose an AR-MMSE scheme in which nodes vote for plausible rectangles consistent with their observations [21].

Gordian SMT has advantages over existing secure localization techniques in the following respects: It requires no specialized hardware or assumptions for trust beyond the anchor positions needed for basic localization. Like filtering algorithms Gordian SMT can be used at the location computation stage, but because it detects attacks rather than filtering them, Gordian SMT can be used in conjunction with state-of-the-art trusted localization algorithms such as [3]. Finally, Gordian SMT has theoretical grounding in rigidity theory [7] which allows a deeper analysis of vulnerabilities to which the method is susceptible.

2.2 Rigidity and Unique Localizability

We apply a simple and common model [1,4,7,9,21,24,27,30,32] of a sensor network: Incomplete pairwise distance measurements are known between sensors. Some sensors, referred to as anchors, have known positions and other unlocalized sensors do not. There are numerous (mostly equivalent) formulations of this setup in the literature, but somewhat arbitrarily we have chosen Anderson et al.'s problem statement [1].

We assume a set of N nodes representing sensors $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ with $S = \{1, 2, ..., N\}$ as their index set. $S_a \subseteq S$ represents anchor nodes with *a priori* known locations. The rest, $S_x = S \setminus S_a$, have unknown locations. The distance from node s_i to s_j is given by d_{ij}. An undirected graph $G = (S, E)$ is a natural choice of model for the sensor network where the sensor nodes are treated as graph nodes and weighted edges E represent distance measurements. Let $E_a \subseteq E$ represent the edges (i, j) such that both $i, j \in S_a$ and $E_x = E \setminus E_a$. Define a framework $F := (G, p)$ where $p : \mathcal{S} \to \mathbb{R}^2$, a "placement", assigns coordinates $p(s_i) \in \mathbb{R}^2$ to each sensor node. When $\|p(s_i) - p(s_j)\| = d_{ij}\ \forall (i, j) \in E_x$, the framework is *consistent.* For our purposes, p^* represents the ground truth positions of the nodes, and p represents the positions we are trying to infer. For the remainder of this paper we will make a minor abuse of notation and use $p(i)$ to represent $p(s_i)$.

Perhaps the most important question to ask about a framework in the context of attack detection is whether or not the framework corresponds to a unique embedding. This is known as Unique Localizability (UL). If a framework corresponds to two distinct embeddings in the absence of attack, an adversary does not have to take any action to create ambiguity in the localization result.

Eren et al. [7] were the first to identify rigidity as the link between sensor network localization, the abovementioned mathematical concept of a framework, and the theory of structures from mechanical and civil engineering. The mechanical analogue of rigidity is easiest to visualize: consider a collection of solid metal rods connected to each other at flexible joints. Mechanical and civil engineers are interested to know if such a structure will continuously deform as force is applied to a point. Deformation is probably desirable in a robotic leg, but potentially catastrophic in a sky scraper. If we replace joints with sensors and metal rods with our d_{ij} terms, there is a strong metaphor between distorting a structure and finding an ambiguous embedding of the network.

To formalize this notion, we again borrow the language of [1]. Two frameworks (G, p), and (G, p') are *equivalent* if $\|p(i) - p(j)\| = \|p'(i) - p'(j)\| \; \forall (i, j) \in E$. There is a stronger property: (G, p), and (G, p') are *congruent* when $\|p(i) - p(j)\| = \|p'(i) - p'(j)\| \; \forall i, j$ regardless of whether or not $(i, j) \in E$. Equivalent, but non-congruent frameworks are *ambiguous*. With these definitions in hand, (G, p) is rigid if when (G, p) is equivalent to (G, p') and $\|p(i) - p'(i)\| < \epsilon \; \forall i \in S$ for some positive ϵ, then the two frameworks are congruent.

Intuitively, this states that a rigid network does not have an infinite family of ambiguous embeddings in the neighborhood of the ground truth. However as demonstrated by the rigid frameworks depicted in Figure 1, a rigid framework does not exclude all ambiguities provided they are at least ϵ distant from p. The two frameworks depicted in Figure 1(a) are equivalent to each other, but they are not congruent (e.g. $\|p(2) - p(3)\| > \|p(2') - p(3)\|$). A rigid structure is resistant to planar deformations, but clearly this is insufficient for congruence. Figure 1 represents two varieties of rigid ambiguities: (a) a flip ambiguity obtained by rotating a graph component through a third dimension and (b) a discontinuous flex ambiguity obtained by temporarily removing an edge, deforming the graph, and snapping the removed edge back into place.

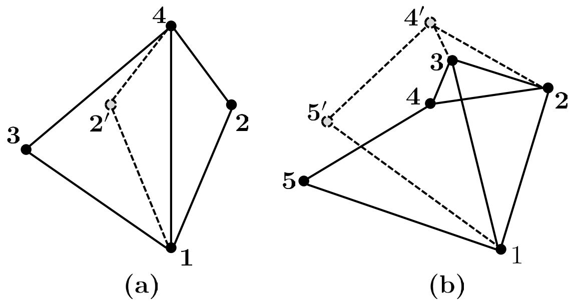


Figure 1: An illustration of ambiguous rigid embeddings from [9], (a) flip ambiguity and (b) discontinuous flex ambiguity.

Whether or not a framework is rigid in two dimensions can be determined by way of the well known Laman Theorem [14] and can be tested in polynomial time with the Pebble Game algorithm [13]. In this paper we will keep the discussion of rigidity at a high level, but we refer the interested reader to [1, 7] for a thorough presentation.

A *globally rigid* framework can be identified by the following 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) [12]. A globally rigid framework is UL if it is also generic, i.e. the coordinates of p are algebraically independent over the rationals. Essentially, the generic requirement forces the framework not to correspond to a ground truth configuration that is entirely co-linear or otherwise degenerate. Such a globally generically rigid framework is UL and ambiguity free.

2.3 Problem Statement

The *exact localization problem* is the following task: Given a graph G along with the $p^*(s_i)$ values for all $i \in S_a$, find a $p : \mathcal{S} \to \mathbb{R}^2$ that assigns coordinates $p(s_i) \in \mathbb{R}^2$, such that $p^*(s_i) = p(s_i) \; \forall i \in S_a$ and $\|p(s_i) - p(s_j)\| = d_{ij} \; \forall (i, j) \in E_x$. We refer to such a p as an *embedding* because if it exists for a G we have succeeded in consistently embedding G in $\mathbb{R}^2$.

For our purposes, the embedding is 2-dimensional because certain technical conditions (specifically, Laman's Theorem) have no complete analogue in three dimensions [7].

Because the frameworks we consider are meant to represent real sensor networks for which there ought to exist a consistent p^*, the exact localization problem should always have at least one solution. Observe that this is not true in general without this assumption: for example, if a framework were to have distance values that violated the triangle inequality. Such a framework has no embedding, but also does not represent any possible ground truth sensor network so we would reject it as an ill-formed input to the exact localization problem.

2.4 Threat Model

In the context of attack detection, we consider two agents: *the system* who attempts to localize the framework by finding a p that corresponds to the ground truth placement of sensors in the world, and *the adversary* who modifies the framework with the goal of causing the system to incorrectly localize one or more sensors to the wrong locations. The threat model we consider in this paper does not allow the system to directly use d_{ij} values from the ground truth distances. Instead, the system has access to indirect measurements $m_{ij} = d_{ij} \cdot (1 + n_{ij}) + a_{ij}$. The n_{ij} terms represent multiplicative noise in the distance obtained from the system's observations of d_{ij}. In the exact (noiseless) case, $n_{ij} = 0$ and in the noisy case, $n_{ij} \in [-h, h]$ for some positive real $0 \leq h < 1$.

The $a_{ij} \in \mathbb{R}$ terms are controlled by the adversary who does not have access to the n_{ij} values. Furthermore for some $k \in \mathbb{N}$ we assume $|\{a_{ij} : a_{ij} \neq 0\}| \leq k$. An edge

(i,j) is "clean" if $a_{ij} = 0$ and "corrupted" when $a_{ij} \neq 0$. We assume from the system's perspective there is no otherwise distinguishing feature between clean and corrupted edges. The attack detection problem is the task of identifying the corrupted measurements and solving the localization problem with the remaining clean (uncorrupted) m_{ij}.

3. ATTACK DETECTION

3.1 Localization Algorithms

Localization algorithms attempt to solve the localization problem (finding a p consistent with the d_{ij}), which may appear difficult considering that the problem of determining if a graph has an embedding (that preserves the d_{ij}) is known to be NP hard [28]. Researchers tackle intractability through two broad classes of methods: local methods that enable each node to determine its location from its neighbors and global methods that simultaneously localize all nodes from an external perspective to the network. GORDIAN SMT 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 [7], but informally it can be thought of as a graph that admits the following localization procedure: Begin with a set of three nodes $\{s_i, s_j, s_k\}$ with known location (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.

An advantage of this procedure is Eren et al. prove trilateration graphs are uniquely localizable, and can be localized in polynomial time. However the procedure is incomplete and fails to localize a broad class of uniquely localizable graphs such as bipartite and wheel (see Figure 2) graph formations. Furthermore this method has significant error problems in the presence of noisy measurements [24]. Other incremental methods such as an iterative procedure for bilateration graphs [9] have been proposed to localize wheel graph formations in sparse networks, but these suffer an exponential blow up when handling wheel graphs, and still fail to localize bipartite graphs.

Alternatives to the iterative localization methods discussed above use some sort of optimization framework to localize all nodes at once [1,3,4,30]. 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

$$\min_{p(i), i \in S_x} \sum_{(i,j) \in E_x} \left| \|(p(i) - p(j)\|^2 - m_{ij}^2 \right|. \qquad (3.1)$$

where $p(i) \in \mathbb{R}^2$ is a decision variable representing the estimated location of sensor i, and m_{ij} is the measured distance between i and j.

These methods commonly rely on a relaxation of the general optimization problem stated in (3.1) from a non-convex program in two dimensions to a Semidefinite Program (SDP) in a higher dimensional space (refer to (4.5) for the statement of the SDP used by GORDIAN SMT). The relaxation was first proposed by Biswas and Ye [4] with good empirical performance, then proved by So and Ye [30] to have important theoretical connections to rigidity and UL. Most significantly, So and Ye show that the optimum value of the SDP is 0 if the problem is UL.

Although a slight departure from the localization algorithms in the rest of this section, Yang et al. demonstrate in [32] how rigidity can be used to identify outlier distance measurements. Their scheme uses the sweeps algorithm from [9] to find inconsistent edges in rigid subgraphs and is a major influence on the direction of our research. However, Yang et al.'s methods are not applicable to intelligent attackers who are able to specifically tune the size of errors to match alternative graph realizations such as in Figure 3.

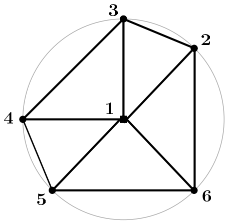


Figure 2: An illustration of a uniquely localizable wheel graph that cannot be localized through iterative trilateration. Image courtesy of [7].

3.2 Attack Tolerance

As mentioned in the discussion on rigidity, there are certain graph properties that must be present for the localization problem to be well-posed even in the absence of attacks. 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. A framework is *k-attack tolerant* (k-AT) when it is well-posed for the attack detection problem (i.e. it is always possible for the system to identify corrupted edges) in the presence of up to k attacks in the absence of noise.

First we provide the intuition. Figure 1(a) is obviously not well-posed for attack detection because as evidenced by the two consistent embeddings, it is not even UL. However, neither is Figure 3, and the latter is UL.

It appears some redundancy is needed in a UL graph to achieve AT, but Figure 4 shows a single redundant edge and this is still not enough to identify the corrupted edge. An attack is evident to the system in Figure 4, but it is unclear if the attack is on $m_{4,7}$ or on $m_{8,7}$.

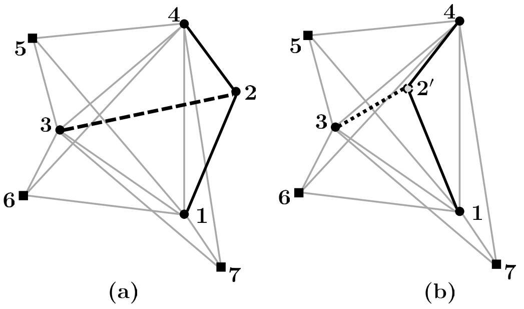


Figure 3: A UL graph that is 0-AT. Observe that a clever adversary corrupting the edge from node 4 to 7 can control which of two consistent placements are realized by the system with no perceptible inconsistencies.

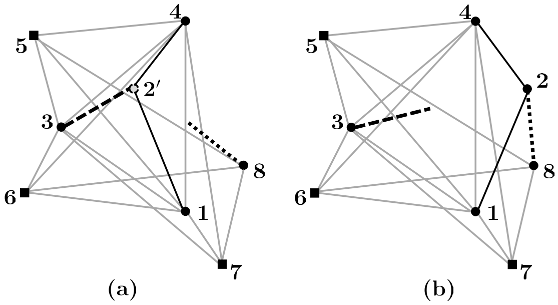


Figure 4: A 0-AT graph that is unembeddable in the presence of one attack. It is impossible to decide whether the red edge or the green edge is corrupted, although it is evident one of the two has been modified.

Clearly, there is a relationship in these graphs between attack tolerance and redundancy in connections. We can formalize this idea. A framework is n redundantly UL (n-UL) when after the removal of any n edges the remaining subgraph is still UL. Armed with this definition, we present the following theorem.

Theorem 1. *A framework F is k-AT if and only if it is $2k$-UL.*

Proof (Necessity). Recall from the definition of UL, that a UL graph is redundantly rigid. Therefore $2k$-UL actually provides $2k+1$ redundancy to the rigidity. We give a counterexample F in Fig. 5with $2k-1$-UL that is not k-attack tolerant. This is evident because the triangular structure in the center is rigid and node A is connected to the rest of the graph by $2k$ edges, implying $2k$ redundant rigidity and (as it is 3-connected and generic), $2k-1$-UL. All k corrupted edges in the graph come out of the totally connected network on the left and attempt to drag node A to its flip position on the other side of the triangle. Another k clean edges are accurately connected to node A in its true position on the right side of the triangle. Like in figure 4, it is impossible to distinguish from G' alone if the k edges from the totally connected network on the left are corrupted and A's true position is to the right of nodes C and B or if the k edges from the totally connected network on the right are corrupted and A ought to be located to the left of nodes C and B. Therefore F is not k-AT. □

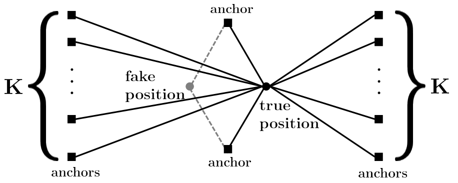


Figure 5: A counterexample graph with $2k-1$-UL that is not k-AT. To see this, remove $2k-1$ edges from the unlocalized node - the result is minimally UL because it is connected to three anchors.

We make use of the following lemma in our proof of the sufficient direction.

Lemma 2. *A framework F that is k-UL with $1 \leq m \leq k$ non-trivial attacks is unembeddable.*

Proof. Construct F' by removing the attacked edges from F. F' is UL and has a unique embedding, $p = p^*$. Now consider a hypothetical embedding for F. It must be consistent with both the clean edges in F' and the attacked edges, but this is impossible because any non-trivial attack is inconsistent with p. Since F' has no other consistent embedding, F is unembeddable. □

Proof (Sufficiency). We give an (inefficient) algorithm for identifying the attacks in F. Construct an F' by removing any k edges from the graph. Maybe all of the attacked edges were removed, maybe not. If some or all of the attacked edges remain, F' is still k-UL, so by Lemma 2 it is unembeddable. If they were all removed, F' is embeddable and the attack is a subset of the removed edges. Enumerate all combinations of k edges and the attack can be identified as the minimal attack hypothesis that yields a consistent framework. □

Our empirical results in section 6 suggest the necessary direction of theorem 1 should be taken with a grain of salt: Gordian SMT is usually effective even for densely connected UL frameworks that aren't $2k$-UL. For these frameworks, the theoretical guarantee of correctness is lost, but the outcome is still usually successful.

4. GORDIAN SMT

Algorithm 1 Attack Detection

```
procedure AttackDetection( Graph)
    for $m_{ij} \in M$ do
        Declare pseudoboolean variable $b_{ij}$        ▷ 1 represents corrupted, 0 represents clean
    $C \leftarrow \sum_{(i,j)\in E_x} b_{ij} \leq k$        ▷ $C$ is the set of pseudoboolean SAT clauses
    while Satisfiable( $C$) do
        AttackHypothesis $\leftarrow$ GetSatisfyingAssignment( C)
        CleanedGraph $\leftarrow$ Graph $\setminus \{m_{ij}$ : AttackHypothesis$(b_{ij}) = 1\}$
        (TestResult, SortedHighResidueEdges) $\leftarrow$ EmbeddabilityTest(CleanedGraph)
        if TestResult = IsEmbeddable then
            return AttackHypothesis        ▷ True attack believed to be subset of AttackHypothesis
        else
            NewC $\leftarrow$ GenCounterexamples(SortedHighResidueEdges)
            if IsEmpty(NewC) then
                NewC $\leftarrow \bigvee_{(i,j)\in \text{CleanedGraph}} b_{ij}$        ▷ Use trivial counterexample as learned conflict clause
            $C \leftarrow C \cup$ NewC
    return Failure
```

4.1 Design: Lazy SMT

Gordian SMT architecture follows the lazy Satisfiability Modulo Theories (SMT) paradigm [2], like Shourky et al.'s lazy SMT solver for secure state estimation in the presence of attacks: Imhotep [29]. Imhotep's chief insight is that identifying attacks in state estimation problems involves a combinatorial attack identification sub-problem that can be isolated from an otherwise convex optimization problem. Gordian SMT can be seen as a application of the Imhotep approach from linear systems to the range-based nonlinear localization problem described in this paper. However, instead of using slack variables to identify bad sensors as in [29], we apply a graph-rigidity specific method outlined in Sec. 4.3.

We present Gordian SMT's high level workflow in algorithm 1: Assume we begin with a graph satisfying the $UL{+}2k$ condition from Theorem 1. Gordian SMT assigns a Boolean variable to each edge of the graph with a 1 indicating that the edge is corrupted and a 0 that the edge is accurate. Initially the SAT solver is only given the constraint there are fewer than k attacks in the graph, but as time progresses it acumulates counterexample clauses learned from the EmbeddabilityTest and GenCounterexamples. In the noiseless problem statement, the SAT solver should always be able to find a satisfying assignment: if it cannot, there is a violation in the assumptions of the algorithm, such as too many corrupted edges.

Gordian SMT's design is modular and does not depend on the implementations of the Satisfiable, EmbeddabilityTest, and GenCounterexamples functions in algorithm 1. Considering any modern out-of-the-box pseudoboolean SAT solver could be used to implement Satisfiable we will instead elaborate on the other more exotic problems.

4.2 Noiseless Embeddability Test

We will show in this section how to frame the embeddability (and localization) problem as a convex optimization problem. An efficient implemenatation matters here because in our experience, embeddability testing takes orders of magnitude longer than SAT solving. At first glance, this may appear odd - that the NP problem (SAT) runs faster than a convex optimization problem - but can be explained by the relative size of the problems.

For clarity, we will first discuss the implementation of the embeddability solver in the noiseless case. We discuss the noisy implementation in section 5.

We rely on a key result from [30]: UL frameworks with no attacks can always be localized in the plane with a zero cost solution from the relaxed SDP localization algorithm from [3]. In consideration of lemma 2, attacked graphs have no consistent embedding and will yield a nonzero cost when tested by the same localization algorithm. Therefore to test embeddability we must simply run the localization algorithm and check the optimal cost value against zero. If we assume the input to Gordian SMT is a noiseless k-attack detectable framework with no more than k corrupted edges, this will yield the desired result.

We outline the SDP procedure: Let the unknown $p(i)$ for $i \in S_x$ be decision variables and define

$$X = [p(1), p(2), ..., p(|S_x|)]$$

as the $2 \times |S_x|$ matrix of decision variables obtained by stacking the first and second coordinates of the $p(i)$. Also, let $\eta_i \in \{0,1\}^n$ be a unit column vector whose i-th component is 1 and all other components 0, and $a_j \in \mathbb{R}^2$ be the position of anchor node j. The pairwise distance between s_i and s_j can be represented as:

$$\|p(i) - p(j)\|^2 = (\eta_i - \eta_j)^T X^T X(\eta_i - \eta_j). \tag{4.1}$$

and the distance between sensor s_i and anchor s_j

$$\begin{aligned}\|p(i)-p(j)\|^2 &= (X\eta_i - a_j)^T(X\eta_i - a_j) \qquad (4.2)\\ &= \begin{bmatrix}\eta_i^T & -a_j^T\end{bmatrix}\begin{bmatrix}X^TX & X^T\\ X & I_d\end{bmatrix}\begin{bmatrix}\eta_i\\ -a_j\end{bmatrix}.\end{aligned}$$

We define $g_{ij} = \begin{bmatrix}\eta_i - \eta_j\\ 0\end{bmatrix}$ if both s_i and s_j are sensors, and $g_{ij} = \begin{bmatrix}\eta_i\\ -a_j\end{bmatrix}$ if either of s_i and s_j is an anchor. Now, the squares of sensor-sensor distance (4.1) and sensor-anchor distance (4.2) can be uniformly represented as

$$m_{ij} = \left| g_{ij}^T \begin{bmatrix}X^TX & X^T\\ X & I_d\end{bmatrix} g_{ij}\right|. \qquad (4.3)$$

where I_d 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 (3.1).

$$\min_{X,Y} \sum_{(i,j)\in E_x} \left|\left(g_{ij}^T \begin{bmatrix}Y & X^T\\ X & I_d\end{bmatrix} g_{ij} - m_{ij}^2\right)\right| \ : \ Y = X^TX. \qquad (4.4)$$

We define the *residue* on edge (i,j), $residue_{(i,j)} = \|p(i)-p(j)\|^2 - m_{ij}^2$, and observe the objective in problem (4.4) is a summation over residues.

We can see (assuming the m_{ij} are noiseless), the objective of (4.4) can attain a minimum of zero. However, (4.4) is not a convex optimization problem, because $Y = X^TX$ expresses a non-convex constraint on the rank of Y. Biswas et al.'s solution to this dilemma is to relax the offending constraint to $Y \succeq X^TX$ [4], yielding problem (4.5) by way of standard manipulations of linear algebra [5].

$$\min_{Z} \sum_{(i,j)\in E_x} \left| g_{ij}^T Z g_{ij} - m_{ij}^2\right| \ : \ Z = \begin{bmatrix}Y & X^T\\ X & I_d\end{bmatrix} \succeq 0. \qquad (4.5)$$

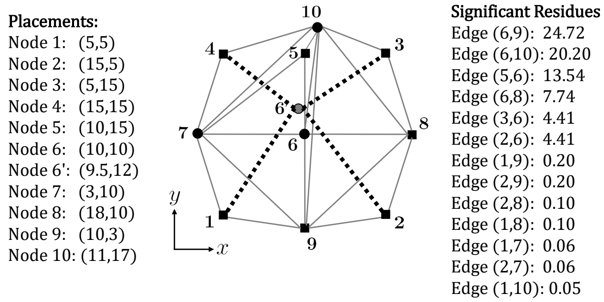
Problem (4.5) is an SDP and can be solved in polynomial time by interior point methods. Conceptually, this relaxation allows the solver to localize each sensor in $\mathbb{R}^{|S_x|}$ instead of $\mathbb{R}^2$ [3]. The component of Z corresponding to X can be read off as the projection of the high dimensional solution back down to the plane of the anchors. This is the localization result.

Finally, we are ready to express the embeddability test: First extract X from Z and compute the residues *with respect to* X. If $\forall (i,j) \in E_x \, residue_{(i,j)} = 0$ the framework has a consistent 2-dimensional embedding and X is it. If $\exists (i,j) \in E_x, residue_{(i,j)} \neq 0$ the situation is as described in lemma 2 where a corrupted edge is inconsistent with p^*. As So and Ye assure us, problem (4.5) always finds 2-dimensional solutions when the input is UL [30] Therefore the only explanation for a nonzero residue is an unembeddable graph[2].

4.3 Trilateration Counterexamples

Minimal counterexample generation is the chief advantage of Gordian SMT over the brute force approach outlined in the proof of theorem 1. As illustrated in algorithm 1, it is always possible to conclude an iteration of attack detection with the trivial counterexample on line 14. Such a conflict specifically disallows the current attack hypothesis from being tested on a later iteration, amounting to a brute force search for the attacked edges. Our empirical evaluations show attack detection can be significantly accelerated with the use of small counterexamples generated by the trilateration graph-based heuristic algorithm presented in algorithm 2.

Our approach is motivated by the observation that high residue values from SDP localization tend to occur in the vicinity of attacked edges in the graph. Biswas et al. suggest large residue values can be directly used to identify faulty measurements [4]. However, a large residue value is not enough to implicate an edge in a coordinated attack. There are examples, such as the graph in Fig. 6, where a cleverly engineered attack causes larger residues on clean edges than corrupted ones.



Placements:	
Node 1:	(5,5)
Node 2:	(15,5)
Node 3:	(5,15)
Node 4:	(15,15)
Node 5:	(10,15)
Node 6:	(10,10)
Node 6':	(9.5,12)
Node 7:	(3,10)
Node 8:	(18,10)
Node 9:	(10,3)
Node 10:	(11,17)

Significant Residues	
Edge (6,9):	24.72
Edge (6,10):	20.20
Edge (5,6):	13.54
Edge (6,8):	7.74
Edge (3,6):	4.41
Edge (2,6):	4.41
Edge (1,9):	0.20
Edge (2,9):	0.20
Edge (2,8):	0.10
Edge (1,8):	0.10
Edge (1,7):	0.06
Edge (2,7):	0.06
Edge (1,10):	0.05

Figure 6: An example showing that cleverly coordinated attacks in problem (4.5) can 1) incur high residues on edges not under attack and 2) spread non-negligible residues over clean edges. Here, squares represent anchors and circles represent sensors. As shown in this figure, the adversary has made a coordinated effort with respect to four attacked edges (in dashed lines) to deceive problem (4.5) into localizing node 6 further to the top of the figure than it ought to be.

Instead of treating large residues as an infallible indicator of corruption, Gordian SMT uses them as a good heuristic indicator of the region of the graph where an attack is taking place. Our algorithm searches for small-sized subgraphs that fail the embeddability test in the vicinity of high-residue edges. The trick is to find subgraphs that should be UL in the absence of an attack

[2] Practical tests should test $residue_{(i,j)} > \epsilon$ for some small $\epsilon > 0$ to account for numerical errors in optimization

due to their connectivity. For this, we use Eren et al.'s iterative trilateration method [7] along with their theoretical guarantee that clean trilateration graphs are UL. When a small trilateration graph fails the embeddability test, it gives a very useful counterexample. We outline this procedure in Alg. 2 where we first add a fast triangle inequality check for three node subgraphs.

If Alg 2 finds no small counterexamples, the entire attack hypothesis can be returned as a counterexample reducing Gordian SMT to a "brute force" search for the attack.

A successful run of GORDIAN SMT terminates with an attack-hypothesis that contains all corrupted edges. However in cases where GORDIAN SMT is looking for up to k attacked edges, but fewer than k attacked edges are present in the graph, it is possible that clean edges might unintentionally be included in a successful attack hypothesis. From a certain perspective, this is an acceptable outcome because the remainder of the graph is UL. But if the specific list of corrupted edges is desired, they can be identified by testing subsets of the successful hypothesis for embeddability until a minimal unembeddable subset is found.

5. ATTACK DETECTION WITH NOISY MEASUREMENTS

As we have shown, localization and attack detection in the absence of noise is a very tidy subject. GORDIAN SMT will always find the attacks in a k-AT framework thanks to the heroics of the noiseless embeddability solver: it just has to solve a localization problem to test a framework for consistency. However, when noise is introduced into the attack detection problem, the localization problems we must ask the embeddability solver to test *usually don't have a solution*! As Anderson et al. observe, solving a noisy UL localization problem 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 [1]. Indeed, Anderson et al. show that it is very unlikely for a noisy localization problem to have an exact solution. To proceed we must redefine localization and attack detection in the presence of noise.

5.1 Noisy Localization

To properly frame the noisy localization problem, we must first articulate the desiderata for noisy localization in the absence of an adversary. Although it is no longer possible to expect a consistent framework with a true embedding, a good localization result ought to find a "good" approximate embedding that matches the observed m_{ij} as well as possible. This suggests an optimization-based localization approach matching the general form of equation (3.1) is an effective strategy. Anderson et al. propose the following goals of such an optimization: "First, if the data are noiseless, the correct sensor positions are returned. Second, when the noise is not great, the solution of the minimization problem is unique and returns sensor position estimates which are not far from the correct values. Third, the errors between the true sensor positions and the estimates returned by solving the minimization problem go to zero continuously as the noise perturbations in the true distances go to zero. " [1]

Anderson et al. use the implicit function theorem to prove these properties hold under certain conditions for the non-convex optimization problem in (5.1) where we replace the m_{ij} terms from equation (3.1) with $d_{ij}^2+n_{ij}$. Their result suggests the efficacy of similar optimization methods under noise, however the specific noise model and optimization statement suffers a significant technical issue: noise is expressed in the square of the distance. The consequence is noise in the distance becomes inseparable from the length of an edge.

$$\min_{p(i),i\in S_x} \sum_{(i,j)\in E_x} \left(\|(p(i)-p(j)\|^2-(d_{ij}^2+n_{ij})\right)^2 \quad (5.1)$$
$$\text{s.t.} \quad p(i)=p^*(i) \quad \forall i\in S_a$$

This same problem is manifested to a lesser degree in (4.5) due to the use of m_{ij}^2 in the objective. In the multiplicative noise model, $m_{ij}^2 = d_{ij}^2\cdot(1+n_{ij})^2$ relates the cost of an edge to its length. Even if we had chosen an additive noise model (for example $m'_{ij}=d_{ij}+n_{ij}$, yielding $m'^2_{ij}=d_{ij}^2+2\cdot d_{ij}\cdot n_{ij}+n_{ij}^2$) there is again an undesired product of d_{ij} and n_{ij}. Errors in longer edges are biased to contribute more cost.

We give an alternative optimization problem for noisy localization in (5.2) with an important technical advantage over formulations in equations (5.1) and (4.5): the effect of multiplicative noise can be disentangled from the length of measurements. Let

$$\min_{p(i),i\in S_x} \quad cost(p) \quad (5.2)$$
$$\text{s.t.} \quad p(i)=p^*(i) \quad \forall i\in S_a,$$
$$cost(p)=\sum_{(i,j)\in E_x}\frac{\left|\|(p(i)-p(j)\|^2-m_{ij}^2\right|}{m_{ij}^2}.$$

The objective in (5.2) is a reasonable choice in its own right. In [3], Biswas et al. define multiplicative γ_{ij} parameters to weight the effect of more significant terms in the localization objective on cost. We can interpret the denominator m_{ij} as a parameter assignment designed to increase emphasis on short edges and reduce the significance of small errors relative to the length of long ones. Furthermore, (5.2) is compatible with the following SDP relaxation (5.3), so we can efficiently solve it.

$$\min_{Z} \sum_{(i,j)\in E_x}\frac{\left|g_{ij}^T Z g_{ij}-m_{ij}^2\right|}{m_{ij}^2} \; : \; Z=\begin{bmatrix} Y & X^T \\ X & I_d\end{bmatrix}\succeq 0. \quad (5.3)$$

5.2 Noisy Embeddability Testing

Algorithm 2 Trilateration Counterexamples

```
procedure GenCounterexamples(SortedHighResidueEdges)
    $C \leftarrow \emptyset$
    for $e = (n_1, n_2) \in$ SortedHighResidueEdges do
        ThirdNodeCands $\leftarrow \{k \mid k \in V', (n_1, k), (n_2, k) \in E'\}$
        for $k \in$ ThirdNodeCands do
            $N \leftarrow \{n_1, n_2, k\}$
            if ObeysTriangleInequality($n_1, n_2, k$) then
                while $|N| <$ MaxSubgraphSize do
                    NextNodeCands $\leftarrow \{n \mid n \in V' \backslash N, \exists i, j, k \in N$ s.t. $(i, n), (j, n), (k, n) \in E'\}$
                    NextNode $\leftarrow$ PickRandomElement(NextNodeCands)
                    $N \leftarrow N \cup$ NextNode
                    Subgraph $\leftarrow$ GetSubgraphFromNodes($N$)        ▷ Include as many edges as possible
                    if EmbeddabilityTest(Subgraph) = NotEmbeddable then
                        $C \leftarrow C \cup$ Subgraph
                        break
            else                                                          ▷ $n_1, n_2, k$ cannot form a triangle
                Subgraph $\leftarrow$ GetSubgraphFromNodes($N$)
                $C \leftarrow C \cup$ Subgraph
    return $C$                                                            ▷ $C$ is the set of counterexamples
```

Just as noisy localization has fewer guarantees than the exact version, noisy embeddability testing (and thus attack detection) is no longer surefire. With noise it is not enough to check edge residues and reject a framework with non-zero values because non-zero values are a side effect of ordinary noise on clean distance measurements. Instead we consider the h bound on the magnitude of noise on a particular measurement. Theorem 3 introduces an upper bound on the optimal cost value of the optimization problem (5.3) that is an effective indicator of corrupted measurements.

The corresponding noisy embeddability testing algorithm is quite simple: solve the SDP formulation in (5.3) and compare the optimal value to $|E_x| \cdot \frac{h}{1-h}$. If we require all input frameworks to represent a consistent ground truth, and multiplicative noise in the square of the distance bounded by h, then $v_{SDP} > |E_x| \cdot \frac{h}{1-h}$ can only be explained by a corrupted edge.

Theorem 3. *Let p^* be the ground truth placement of a consistent framework F with noisy distance measurements m_{ij}. For $(i,j) \in E_x$, we represent the squared measurement as follows: $m_{ij}^2 = d_{ij}^2 \cdot (1 + n'_{ij})$ (note that n'_{ij} is multiplicative in the square of the edge length). Suppose $n'_{ij} \in [-h, h]$ with $0 \leq h < 1$. Then $cost(p^*) \leq |E_x| \cdot \frac{h}{1-h}$.*

Proof. For the ground truth p^*, we have:

$$cost(p^*) = \sum_{(i,j)\in E_x} \frac{\left|d_{ij}^2 - d_{ij}^2 \cdot (1 + n'_{ij})\right|}{d_{ij}^2 \cdot (1 + n'_{ij})}$$

$$= \sum_{(i,j)\in E_x} \frac{\left|n'_{ij}\right|}{1 + n'_{ij}} \leq \sum_{(i,j)\in E_x} \frac{h}{1-h} = |E_x| \cdot \frac{h}{1-h}$$

Therefore, if an instance of problem (5.1) achieves an optimal cost greater then the threshold in Theorem 3 either it does not correspond to a consistent F (and has no consistent ground truth) or an edge has been corrupted. If we require consistent F, the latter is the only possibility.

Of course, we are unable to solve the non-convex problem for v^{opt} efficiently. Rather, we solve for v^{sdp}, the minimum of the SDP problem (5.3) that can be solved in polynomial time by interior point methods. However, since the ground truth placement is a feasible solution to (5.3) , it is clear that we have $v^{\mathrm{sdp}} \leq cost(p^*)$, and so Theorem 3 applies.

Note, in Theorem 3 we use a bound on the noise which is multiplicative in the *square* of the edge length. In our experiments the noise bound is multiplicative in the edge length itself. If the multiplicative noise in the edge length is n_{ij} and the multiplicative noise in the square of the error is n'_{ij}, we have, with m_{ij} as the noisy measurement of true length d_{ij}:

$$\begin{aligned} m_{ij}^2 &= d_{ij}^2 \cdot (1 + n'_{ij}) = d_{ij}^2 \cdot (1 + n_{ij})^2 \\ \Rightarrow \; n'_{ij} &= 2n_{ij} + n_{ij}^2 \end{aligned} \tag{5.4}$$

And so, if we require $n'_{ij} \in [-h, h]$ as in Theorem 3, noting that $x^2 + 2x$ is monotonic in $[-1, \infty)$, we have the slightly asymmetric result for corresponding n_{ij} bounds:

$$\sqrt{1-h} - 1 \leq n_{ij} \leq \sqrt{1+h} - 1 \tag{5.5}$$

5.3 Resilient Frameworks

The noisy embeddability testing algorithm is sound (it never returns false positives), but incomplete (it sometimes misses attacks). To a certain extent, this problem is fundamental: soundness requires the algorithm adopt a conservative stance to allow for worst-case noise, but certain noise assignments below the threshold of detection facilitate low-cost attacks. For example, when noise distributed over the entire framework pushing an unlocalized node in a consistent direction coincides with an attack pushing the node in the same direction, the attack may go undetected. Bad localization results may appear compelling with respect to the cost function even without attacks. Geometric Dilution of Precision [15] is a well known result within the world of navigation systems like GPS that describes how the geometry of noisy range measurements can affect localization precision. Moore et al.'s iterative localization algorithm by the use of robust quadrilaterals [24] applies a similar intuition.

These weaknesses in even non-adversarial localization hint at a flaw in the paradigm of optimization-based localization: a low cost p does not necessarily correspond to a qualitatively good localization result. We term this often implicit assumption to the contrary as *resilience*. Articulating its topological and geometric requirements is beyond the scope of this paper and is a topic for future research. We postulate that the result, will draw upon rigidity and be applicable to noisy attack detection in much the same way the result in theorem 1 uses UL in the noiseless case.

Since the noisy embeddability test does not identify low cost attacked graphs as problematic, GORDIAN SMT may fail to identify all the attacked edges. However, if we may assume redundant resilience, i.e. low cost results are qualitatively good, whatever attacks GORDIAN SMT misses are low cost with respect to a clean subset of the edges and must not be that bad.

6. EMPIRICAL EVALUATION

Our goals in experimental evaluation are two-fold: evaluate the accuracy of Gordian SMT for frameworks with random attacks and varying degrees of noise, and demonstrate the performance advantages of Gordian SMTs trilateration counterexamples in comparison to brute force trivial counterexamples.

We implemented GORDIAN SMT in MATLAB 2014b, using the YALMIP toolbox [23] to model the SDP problem. Our implementation uses SEDUMI [31] and SAT4J's pseudoboolean solver as the underlying SDP and SAT solvers. Our testing platform is a Linux Virtual Machine (VM) running on a server. The VM is allocated with 4 Intel Xeon E5-2667v2 3.3GHz cores and 8GB memory.

All of our experiments begin by randomly generating a grid-like ground truth sensor network such as figure 7 in a 15 unit by 15 unit box. We place four anchors along the corners and use the unit disc graph model to determine the connectivity of nearby nodes. To generate larger graph instances we increase the number of complete rows and columns of unlocalized nodes in the convex hull of the anchors. Gordian SMT problem instances are created by computing $m_{ij}^2 = d_{ij}^2 \cdot (1 + n'_{ij})$ with n'_{ij} values selected uniformly at random in a specified range (refer to equation 5.5 to compute the corresponding n_{ij}). Attacks are generated by randomly picking an edge (i, j) and corrupting $m_{ij}^2 = d_{ij}^2 \cdot (1 + a'_{ij})$, with a'_{ij} outside the noise range. We take care to maintain $0 < m_{ij} < UnitDiscSize$, ensuring attacked edges remain in a plausible range between 0 and the maximum distance over which nodes are connected.

We expect Gordian SMT to have a harder time correctly identifying attacks in noisier and larger graphs because the noisy embeddability testing bound depends on both h and $|E_x|$. Figure 8 shows the effect of noise bounds on the performance (number of detected attacks and and average runtime) of GORDIAN SMT in the presence of two, three, and four attacks. The experiment was performed on the ground truth graph as shown in Figure 7. For each combination of the parameters, we resample noise in the specified range and generate new attacks to create 3 different problem instances. We run GORDIAN SMT over each instance for 3 trials (because the counterexample generation process is random). The average values of the results are reported in Figure 8.

The number of detected attacks in Figure 8 appears to decrease in noisier problems as expected. However it is difficult to interpret the significance of figure 8 with respect to the discussion on resilience in section 5.3. It is possible that the undetected attacks in our experiments are undetected because they cause very little disturbance in the localization result and there is minimal harm leaving them in.

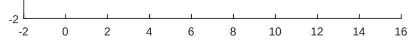


Figure 7: The ground truth with 20 nodes and 166 edges for the noise vs accuracy comparison in Figure 8.

We give an evaluation of worst-case running time for four different ground truth graphs in Table 1. We set a small noise bound, $h = 0.01$, in the square of the distance and report the *worst case* runtime for three trials[3]. It appears Gordian SMT is able to detect attacks in a range of sizes fairly efficiently, but it occasionally struggles (as in the case for Graph A with 2 attacks) to find effective counterexamples. This is still much better than the brute force algorithm that tries $O(\binom{|E_x|}{k})$ attack hypothesis in expectation. As a point of comparison, the runtime for brute force attack detection on Graph B with 2 attacks with two attacks was

[3] We discard trials that find no attacks, because such trials terminate after a single SDP invocation and have an uninteresting (very fast) run-time.

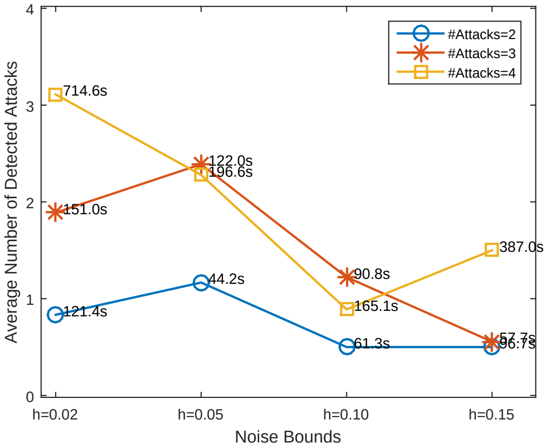


Figure 8: The effect of noise bounds on accuracy.

1146 seconds (the 514th guessed attack hypothesis was successful).

7. CONCLUSION

We have presented Gordian SMT, an attack detection algorithm at the distance-graph abstraction level that requires no custom hardware, no special nodes, and no specific ranging techniques. In the noiseless case we prove Gordian SMT a sound and complete algorithm for detecting up to k attacks in a $2k$-UL framework. Gordian SMT leverages an SMT solving architecture and our trilateration counterexamples algorithm to achieve a over a naive brute force implementation. We leverage embeddability testing by way of the SDP relaxation for localization [4] to detect inconsistent frameworks in the noiseless case. We also prove a bound on the cost of a consistent embedding that facilitates the extension of embeddability testing to the case of multiplicative noise. Finally, we give empirical results demonstrating Gordian SMT's success on realistic noisy input.

In future work we intend to investigate the resilience property that we postulate plays a similar role to unique localizability in the noisy case to guarantee total attack detection of serious attacks. Interesting extensions of the Gordian SMT approach include alternative noise models, alternative cost functions, 3-dimensional localization, and alternative attack models such as malicious anchors.

8. REFERENCES

[1] B. D. O. Anderson, I. Shames, G. Mao, and B. Fidan. Formal Theory of Noisy Sensor Network Localization. *SIAM Journal on Discrete Mathematics*, 24(2):684–698, Jan. 2010.

[2] C. Barrett, R. Sebastiani, S. Seshia, and C. Tinelli. *Handbook of satisfiability*. IOS Press Fairfax, 2009.

[3] P. Biswas, T. C. Liang, K. C. Toh, Y. Ye, and T. C. Wang. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Transactions on Automation Science and Engineering*, 3(4):360–371, Oct 2006.

[4] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54. ACM, 2004.

[5] S. P. Boyd, editor. *Linear matrix inequalities in system and control theory*. Number vol. 15 in SIAM studies in applied mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 1994.

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

[7] T. Eren, O. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. Anderson, and P. N. Belhumeur. Rigidity, computation, and randomization in network localization. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2673–2684. IEEE, 2004.

[8] H. Fawzi, P. Tabuada, and S. Diggavi. Secure Estimation and Control for Cyber-Physical Systems Under Adversarial Attacks. *IEEE Transactions on Automatic Control*, 59(6):1454–1467, June 2014.

[9] D. K. Goldenberg, P. Bihler, M. Cao, J. Fang, B. Anderson, A. S. Morse, and Y. R. Yang. Localization in sparse networks using sweeps. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 110–121. ACM, 2006.

[10] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 81–95. ACM, 2003.

[11] V. Honkavirta, T. Perala, S. Ali-Loytty, and R. Piche. A comparative survey of WLAN location fingerprinting methods. In *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on*, pages 243–251. IEEE, 2009.

[12] B. Jackson and T. Jordan. Connected rigidity matroids and unique realizations of graphs. Technical report, Eotvos University, Budapest, Hungary, Mar. 2003.

Table 1: Worst-Case Runtime Comparison on Benchmarks of Different Sizes under $h = 0.01$

Benchmark $(\lvert S_x\rvert, \lvert E_x\rvert)$	Graph A (13,44)			Graph B (20,109)			Graph C (20,142)			Graph D (29,305)		
#attacks	1	2	3	1	2	3	1	2	3	1	2	3
Worst-case Runtime (s)	26.3	268.8	65.2	177.9	204.3	525.2	146.9	35.3	112.4	59.9	371.9	914.9

[13] D. Jacobs and B. Hendrickson. An Algorithm for Two Dimensional Rigidity Percolation: The Pebble Game. *Journal of Computational Physics*, 137:346–365, 1997.

[14] G. Laman. On graphs and rigidity of plane skeletal structures. *Journal of Engineering mathematics*, 4(4):331–340, 1970.

[15] R. B. Langley. Dilution of Precision. *GPS WORLD*, 1999.

[16] P. Lazik, N. Rajagopal, O. Shih, B. Sinopoli, and A. Rowe. ALPS: A Bluetooth and Ultrasound Platform for Mapping and Localization. pages 73–84. ACM Press, 2015.

[17] L. Lazos and R. Poovendran. SeRLoc: Robust localization for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 1(1):73–100, 2005.

[18] E. A. Lee, B. Hartmann, J. Kubiatowicz, T. Simunic Rosing, J. Wawrzynek, D. Wessel, J. Rabaey, K. Pister, A. Sangiovanni-Vincentelli, S. A. Seshia, D. Blaauw, P. Dutta, K. Fu, C. Guestrin, B. Taskar, R. Jafari, D. Jones, V. Kumar, R. Mangharam, G. J. Pappas, R. M. Murray, and A. Rowe. The Swarm at the Edge of the Cloud. *IEEE Design & Test*, 31(3):8–20, June 2014.

[19] Z. Li, W. Trappe, Y. Zhang, and B. Nath. Robust statistical methods for securing wireless localization in sensor networks. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 12. IEEE Press, 2005.

[20] D. Liu, P. Ning, and W. Du. Detecting malicious beacon nodes for secure location discovery in wireless sensor networks. In *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pages 609–619. IEEE, 2005.

[21] D. Liu, P. Ning, A. Liu, C. Wang, and W. K. Du. Attack-Resistant Location Estimation in Wireless Sensor Networks. *ACM Transactions on Information and System Security*, 11(4):1–39, July 2008.

[22] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080, Nov. 2007.

[23] J. Löfberg. Yalmip: A toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289. IEEE, 2004.

[24] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 50–61. ACM, 2004.

[25] F. Pasqualetti, F. Dorfler, and F. Bullo. Attack Detection and Identification in Cyber-Physical Systems. *IEEE Transactions on Automatic Control*, 58(11):2715–2729, Nov. 2013.

[26] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43, 2000.

[27] A. Savvides, C.-C. Han, and M. B. Srivastava. Dynamic Fine-Grained Localization in Ad-Hoc Wireless Sensor Networks. 2001.

[28] J. B. Saxe. Embeddability of weighted graphs in k-space is strongly np-hard. 1979.

[29] Y. Shoukry, P. Nuzzo, A. Puggelli, A. L. Sangiovanni-Vincentelli, S. A. Seshia, M. Srivastava, and P. Tabuada. IMHOTEP-SMT: A Satisfiability Modulo Theory Solver For Secure State Estimation. In *13th International Workshop on Satisfiability Modulo Theories (SMT)*, 2015.

[30] A. M.-C. So and Y. Ye. Theory of semidefinite programming for Sensor Network Localization. *Mathematical Programming*, 109(2-3):367–384, Jan. 2007.

[31] J. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[32] Z. Yang, L. Jian, C. Wu, and Y. Liu. Beyond triangle inequality: Sifting noisy and outlier distance measurements for localization. *ACM Transactions on Sensor Networks*, 9(2):1–20, Mar. 2013.

[33] Y. Zeng, J. Cao, J. Hong, S. Zhang, and L. Xie. Secure localization and location verification in wireless sensor networks: a survey. *The Journal of Supercomputing*, 64(3):685–701, June 2013.