IMHOTEP-SMT: A Satisfiability Modulo Theory Solver For Secure State Estimation*

Yasser Shoukry[1], Pierluigi Nuzzo[2], Alberto Puggelli[2],
Alberto L. Sangiovanni-Vincentelli[2], Sanjit A. Seshia[2],
Mani Srivastava[1], and Paulo Tabuada[1]

[1] Department of Electrical Engineering, University of California at Los Angeles
[2] Department of Electrical Engineering and Computer Sciences, University of California at Berkeley

Abstract

This paper presents IMHOTEP-SMT, a solver for the detection and mitigation of sensor attacks in cyber-physical systems. IMHOTEP-SMT receives as inputs a description of the physical system in the form of a linear difference equation, the system input (control) signal, and a set of output (sensor) measurements that can be noisy and corrupted by a malicious attacker. The output is the solution of the secure state estimation problem, i.e., a report indicating: (i) the corrupted sensors, and (ii) an estimate of the continuous state of the system obtained from the uncorrupted sensors. Based on this estimate, it is then possible to deploy a control strategy, while being resilient to adversarial attacks. The core of our tool relies on the combination of convex programming with pseudo-Boolean satisfiability solving, following the lazy satisfiability modulo theory paradigm. We provide an empirical evaluation of the tool scalability, and demonstrate its application to attack detection and secure state estimation of electric power grids.

1 Introduction

In cyber-physical systems (CPS), software and hardware components collect data from physical processes via sensors in real time, and process them to make decisions that can be both safety-critical and security-critical. In this context, adversarial attacks on sensor measurements can easily lead to life-threatening situations: examples of incidents are the infamous Stuxnet malware targeting industrial SCADA devices [Lan11], the injection of false data in smart grids [LNR09], and sensor spoofing attacks to automotive anti-lock braking systems [SMTS13]. Detecting and mitigating such attacks is key to the safe and secure deployment of CPS. One approach is to algorithmically determine which sensors are under attack while estimating the state of the physical system, a problem that is also known as *secure state estimation*.

Detecting and mitigating attacks on sensory data is, in general, a combinatorial problem [PDB13], which has been typically addressed by either brute force search, suffering from scalability issues [PDB13, CWH15], or via convex relaxations using algorithms that can terminate in polynomial time [FTD14, ST13] but are not necessarily sound.

In this paper, we present IMHOTEP-SMT, a tool implementing, to the best of our knowledge, the first sound and complete[1] algorithm for sensor attack detection and mitigation in linear dynamical systems [SPN+15, SNP+14].[2] IMHOTEP-SMT takes as input a mathematical description of the system

*This work was partially sponsored by the NSF award 1136174, by DARPA under agreement number FA8750-12-2-0247, by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA, and by the NSF project ExCAPE: Expeditions in Computer Augmented Program Engineering (award 1138996).

[1] In the context of decision procedures on the reals, in this paper, we resort to the notion of δ-completeness proposed in [GAC12].

[2] IMHOTEP-SMT along with several application examples can be downloaded from `http://nesl.github.io/Imhotep-smt/`.

dynamics along with its input signals and output (sensor) measurements. The tool can operate in two modes: (i) offline *system specification*, and (ii) online *state estimation*. In the first mode, IMHOTEP-SMT analyzes and diagnoses the mathematical description of the dynamical system, by characterizing its *security index*, that is, the maximum number of attacked sensors that the system can tolerate. In the second mode, IMHOTEP-SMT performs attack detection and secure state estimation at runtime, as new sensor measurements become available, by solving a new problem instance at every time interval. Instrumental to the performance of our tool is a satisfiability modulo theory (SMT) based approach that efficiently tackles the combinatorial aspects of the problem [SNP+14].

The current version of IMHOTEP-SMT assumes that the CPS is modeled by a linear dynamical system with bounded noise; its extension to support both Gaussian noise and nonlinear dynamical systems is ongoing work. Differently from our previous work [SNP+14, SPN+15], in this paper, we focus on the implementation aspects and the usability of IMHOTEP-SMT, rather than its theoretical underpinnings. The contributions of this paper can then be summarized as follows:

- We introduce IMHOTEP-SMT, a tool that implements a novel approach to SMT solving, specifically tailored to the problem of sensor attack detection and mitigation in linear dynamical systems. In this application domain, IMHOTEP-SMT improves on the performance of other existing solvers, such as ISAT [FHT+07], DREAL [GKC13], and Z3 [DMB08]. We provide the details of the implementation of IMHOTEP-SMT; its theoretical foundations, as well as the proof of correctness of the core algorithms, are instead discussed in our previous publication [SNP+14].
- We propose two new algorithms that enhance the usability of IMHOTEP-SMT, by facilitating the problem specification process. As detailed in Appendix A, the proposed algorithms include: (i) fast computation of the upper bound of the security index, and (ii) analysis of the system structure to detect which sensors are the most vulnerable and need to be physically secured.
- We provide new benchmarks to compare the performance of IMHOTEP-SMT with other SMT solvers, and demonstrate, for the first time, its effectiveness on a "smart" grid case study.

In the sequel, we detail the two operating modes of IMHOTEP-SMT (Sec. 2 and 3), report the results of our empirical evaluation (Sec. 4), and then draw some conclusions.

2 Configuring IMHOTEP-SMT: Offline System Specification

As a first step, IMHOTEP-SMT must be configured offline with the parameters of the CPS model to be controlled. The configuration parameters are listed below.

1) System model. IMHOTEP-SMT detects attacks on a linear discrete-time dynamical system. When there is no measurement noise or sensor attacks, this system has the following form:

$$x^{(t+1)} = Ax^{(t)} + Bu^{(t)}, \qquad y^{(t)} = Cx^{(t)}, \tag{1}$$

where $x^{(t)} \in \mathbb{R}^n$ is the state vector of the CPS at time $t \in \mathbb{N}$, $u^{(t)} \in \mathbb{R}^m$ is the input, and $y^{(t)} \in \mathbb{R}^p$ is the observed output. The matrices A, B, and C have appropriate dimensions and represent how the system states evolve over time, how the inputs affect the system states, and which states are sensed by the sensors, respectively.

2) Bounds on measurement noise. In real-world systems, sensors are affected by noise. To model this effect, IMHOTEP-SMT receives as input the parameter `noise_level`, which is the upper bound on the magnitude (norm) of the additive noise $\psi^{(t)}$ on the sensor measurements $y^{(t)} = Cx^{(t)} + \psi^{(t)}$. This

bound is used to discriminate between noise and attacks. If `noise_level` is set lower than the actual noise magnitude in the system, IMHOTEP-SMT will treat noise as an attack. If instead `noise_level` is set higher than the actual noise level, the small attack signals will be regarded as noise.

The bounded noise assumption above does not limit the capabilities of our tool, since it follows from the underlying physics of each sensor. In fact, each sensor is characterized by a dynamic range (the maximum and minimum bounds on the signal measured out of the sensor) as well as a signal-to-noise ratio (the maximum bound on the noise as a fraction of the dynamic range of the sensor) requirement for accurate digitalization of the sensor output by the analog-to-digital converter. Moreover, the maximum bound on sensor noise can be characterized by testing the sensor via a controlled experiment with adequate calibration equipment, e.g. by setting the input signal of the sensor to a specific value, and then measuring the maximum error induced by the sensor on the expected signal.

3) Safe sensors. Based on the physical implementation of the system, some sensors may not be physically accessible to the attacker and can be assumed to be safe *a priori*. This information is used by IMHOTEP-SMT to facilitate the attack detection task.

4) Security index. The security index of a system (denoted as $\bar{s}$) is defined as the maximum number of attacked sensors for which state estimation is feasible. This index depends on the structure of the system (captured by the matrices A and C). The computation of the security index is, however, combinatorial in nature [ST13]. Therefore, in the current implementation of IMHOTEP-SMT, we ask the user to specify an initial guess for the security index and compute $\bar{s}$ only if needed. Specifically, the tool proceeds as follows:

1. It computes a theoretical upper bound on the security index (which can be computed in linear time). Details on how to compute this upper bound are given in Appendix A.
2. If the security index specified by the user is higher than this upper bound, IMHOTEP-SMT further analyzes the system structure to provide suggestions on which sensors need to be physically "secured" for the state estimation problem to be feasible. This procedure is also discussed in Appendix A.
3. If the user's "guess" on the security index is instead smaller or equal than the theoretical upper bound, the tool can still perform, if requested by the user, the combinatorial test known as *sparse observability test* [SNP+14] (see Appendix A). We note that the process of calculating and checking the security index needs to be performed only once in the offline mode of the tool.

3 Online State Estimation With IMHOTEP-SMT

Fig. 1 shows the interface between IMHOTEP-SMT and the CPS under attack. At runtime, the tool receives the stream of input signals fed to the physical system along with the stream of corrupted measurements from the sensors. As described in [SNP+14], the goal of IMHOTEP-SMT is to search for an estimate of the system state $\hat{x}$ and an assignment for the binary indicator variables $b_1, b_2, \ldots, b_p$ that satisfy the following formula[3]:

$$\phi ::= \left\{ \bigwedge_{i\in\{1,\ldots,p\}} \neg b_i \Rightarrow \|Y_i - F_i U - \mathscr{O}_i \hat{x}\|_2 \leq \|\Psi_i\|_2 \right\} \quad \wedge \quad \left\{ \sum_{i\in\{1,\ldots,p\}} b_i \leq \bar{s} \right\}, \tag{2}$$

[3] $\|z\|_2 \in \mathbb{R}$ denotes the 2-norm of a real valued vector $z \in \mathbb{R}^n$, i.e., $\|z\|_2 = \sqrt{\sum_{i=1}^{n} z_i^2}$

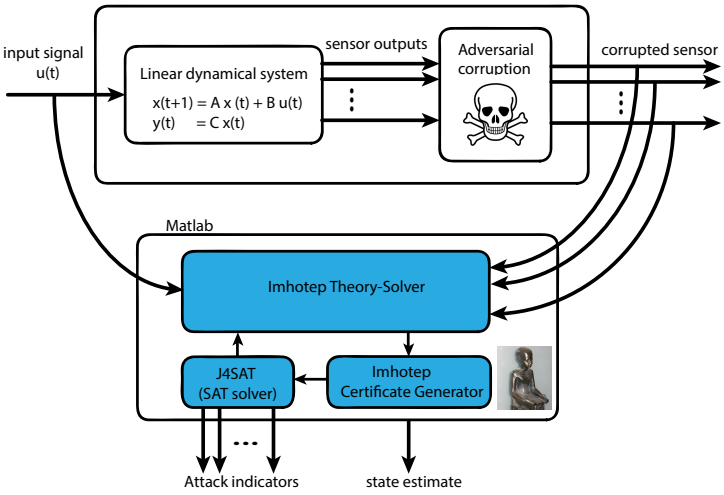


Figure 1: Illustration of the online operation of IMHOTEP-SMT

where $b_i = 0$ if the ith sensor is attack free and $b_i = 1$ otherwise. The measurement stream collected from the ith sensor is denoted by Y_i , while U stores the input stream, and $\bar{s}$ is the security index of the system, as obtained from the offline configuration step. The matrices F_i and $\mathcal{O}_i$ characterize the effect of the inputs and of the state on the outputs, respectively, and can be calculated from the system model (defined in (1)) as follows:

$$\mathcal{O}_i = \begin{bmatrix} C_i \\ C_iA \\ \vdots \\ C_iA^n \end{bmatrix} \qquad F_i = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ C_iB & 0 & \dots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ C_iA^{\tau-2}B & C_iA^{\tau-3}B & \dots & C_iB & 0 \end{bmatrix} \tag{3}$$

where C_i is the ith row of the matrix C. Finally $\|\Psi_i\|_2$ is the bound on the measurement noise which is given as an input to the tool in the offline configuration mode as discussed in Section 2.

The first conjunction of constraints in the formula ϕ requires that *all* attack-free sensors agree on one state estimate $\hat{x}$. This is enforced by asking that, for all attack-free sensors, the mismatch between the measured outputs and the outputs consistent with the dynamics and the estimate $\hat{x}$, represented by the norm of $(Y_i - F_iU - \mathcal{O}_i\hat{x})$, can be explained as noise. The second inequality enforces the cardinality constraint on the number of attacked sensors.

Both the streams of input and output signals are stored into internal buffers of appropriate length. Once the buffers are full, the tool performs its computations to find the sensors under attack and an estimate of the system state, and generates a security report with this information. Then, the current input and output samples stored in the buffers are replaced with newly incoming samples and the calculations are repeated.

IMHOTEP-SMT assumes that the time between successive samples in the input and output streams (as well as the sampling time of the mathematical model) are all compatible with the computation time of the solver. Moreover, the input and output streams are assumed to be synchronized and to follow the sampling time of the system model. While the computation time of the solver depends, in general, on the size of the system, we show in Sec. 4 that IMHOTEP-SMT scales better than previously proposed approaches, which makes it a better solution for real-time attack detection.

Internal Architecture and Operation Principle. As shown in Fig. 1, IMHOTEP-SMT combines a Theory solver with a pseudo-Boolean SAT solver, which can efficiently reason about cardinality constraints over Boolean variables as the one in (2). The pseudo-Boolean SAT solver (currently implemented using the SAT4J solver [BP10]) starts by determining a candidate set of sensors that are attack-free, which is then passed to the Theory solver (implemented in MATLAB[4]). The Theory solver uses the output stream from the sensors that are assumed to be attack-free and the input stream to calculate the optimal state estimate $\hat{x}$. This is done by formulating and solving a set of convex optimization problems. The estimate $\hat{x}$ is then used to modify the original guess of the pseudo-Boolean SAT solver until the actual attacked sensors are found.

For a given assignment on the indicator variables $b_1 \dots b_p$ from the pseudo-Boolean SAT solver, if the state estimate $\hat{x}$ calculated by the Theory Solver does not satisfy the constraints on the right side of the implications in (2), the *Certificate Generator* module generates a succinct unsatisfiability certificate (or counterexample). Inspired by the approach in [NPSSV10], we adopt a lazy SMT paradigm that exploits the specific structure of the secure state estimation problem together with convex programming to generate customized, yet stronger certificates, and enhance the execution time of our tool. Specifically, the certificate generator exploits the underlying convex geometry of the constraints in the formula ϕ (Equation 2) to generate a compact certificate, i.e. a smaller set of sensors whose measurements are conflicting, and rule out any Boolean assignment that contains this conflicting set. This certificate is encoded as a pseudo-Boolean predicate of the form $\sum_{i\in\mathbb{I}} b_i \geq 1$, where $\mathbb{I}$ is the set of sensor indices that are conflicting. Such a predicate informs the pseudo-Boolean SAT solver that at least one of the sensors indexed by $\mathbb{I}$ can not be regarded as being attack-free. The smaller the cardinality of $\mathbb{I}$, the higher is the amount of information provided to the SAT solver to reduce its search space of all possible assignments over $b_1 \dots b_p$.

Our certificate generation algorithm relies on the following theoretical guarantees: (1) there always exist compact certificates for the secure state estimation problem, which prevent enumerating all possible assignments; (2) there exists an upper bound on the number of calls to the theory solver when these certificates are used. Although this theoretical upper bound is conservative [SNP+14], it is still much tighter than the one obtained by enumerating all the possible assignments.

4 Tool Evaluation

We evaluated IMHOTEP-SMT on the problem of estimating the state of an electric power grid, in which some measurement units are influenced by an adversarial attack. Smart grids are indeed an important example of CPS for which attacks have been recently documented [LNR09]. We consider the IEEE 14-bus power network shown in Fig. 2, composed of 5 synchronous generators and 14 buses. The state of each generator includes rotor angle and frequency. The overall system has 35 sensors: 14 sensors measure the real power injections at every bus, 20 sensors measure the real power flows along every branch, and one sensor measures the rotor angle of generator 1. The matrices A, B, and C modeling the power network are derived in [PDB13], where it is also shown that the rotor angle sensor must be secured for the system to have non-zero security index. The attacker is assumed to pick a random set of 14 sensors.

After configuring IMHOTEP-SMT offline with the parameters above, we simulated the power network using MATLAB and directed the corrupted sensor outputs to IMHOTEP-SMT. Fig. 2 (right) shows the error in the state estimation of IMHOTEP-SMT when compared with the error of a traditional least squares state estimator. IMHOTEP-SMT was able to completely isolate the corrupted sensors and correctly construct the state of the power grid.

[4] We could easily integrate the SAT solver with the Theory solver, since SAT4J is implemented in Java, and MATLAB can seamlessly incorporate Java libraries into its code.

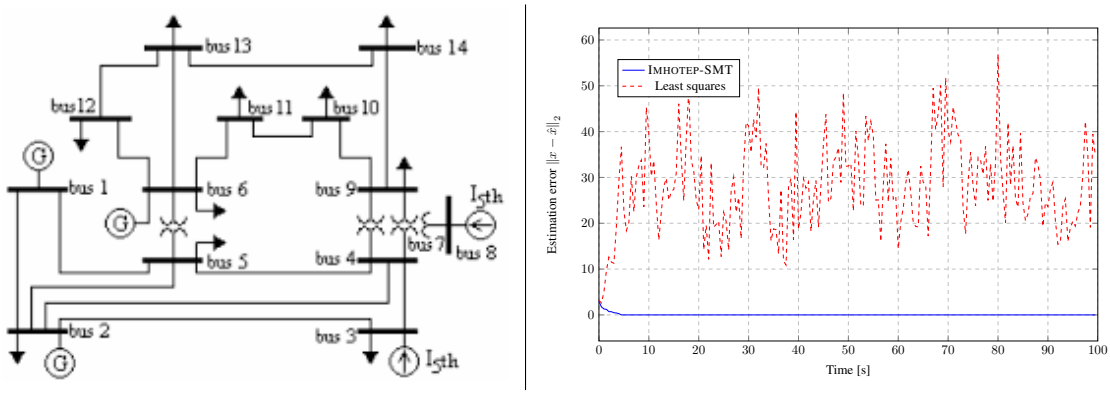


Figure 2: The IEEE 14-bus power network (left) and the estimation error (right) when IMHOTEP-SMT is used versus a standard least squares estimator.

To assess the scalability of IMHOTEP-SMT, we randomly generated a set of matrices for systems of increasing size. For each of these systems, we configured IMHOTEP-SMT, provided it with the system inputs and outputs for state estimation, and recorded the execution time. All the experiments were executed on an Intel Core i7 3.4-GHz processor with 8 GB of memory. Fig. 3 reports the numerical results in two test cases. In the right figure, we fix the number of sensors $p = 20$ and increase the number of system states from $n = 10$ to $n = 150$. In the left figure, we fix the number of states $n = 50$ and increase instead the number of sensors from $p = 3$ to $p = 150$. In both cases, half of the sensors are attacked. As evident from Fig. 3, increasing n has a small effect on the overall execution time, which reflects the fact that the number of constraints to be satisfied does not depend on n. Conversely, as the number of sensors increases, the number of constraints increases, hence the execution time of IMHOTEP-SMT.

When executed on the randomly generated instances of the secure state estimation problem, IMHOTEP-SMT favorably compares with other, more generic, state-of-the-art SMT solvers for non-linear constraints on the reals, e.g. ISAT [FHT+07] or DREAL [GKC13] (based on interval constraint propagation), which may result into inaccurate solutions (e.g. by returning `UNKNOWN`) or longer execution times. We believe that the improvement in runtime performance of IMHOTEP-SMT is mostly motivated by the generation of compact certificates for our specific problem. Overall, our experiments in Fig. 3 show that IMHOTEP-SMT, relying on the combination of pseudo-Boolean reasoning with convex programming, always outperforms the other approaches, and scales nicely with respect to both n and p.

While Z3 [DMB08] can provide support for nonlinear polynomial arithmetic, it also suffers from incompleteness or termination issues[5], and returned `UNKNOWN` on some of the problem instances as the one in (2) in our experiments. Therefore, to compare with Z3, while guaranteeing completeness and termination, we opted for an alternative, linear formulation of the original problem. In fact, another formulation can be proposed for (2), which replaces 2-norms with infinity-norms, and can generate a set of linear constraints that can be handled by SMT solvers for the linear theory over real numbers (e.g., including Z3). However, this linear formulation generates $n \times p + 1$ instead of $p + 1$ constraints, where n is the number of state variables, and p the number of sensors. As shown in Fig. 3, such a large number of constraints inevitably impairs the scalability of this approach.

Finally, IMHOTEP-SMT shows again superior performance when compared with other non SMT-

[5]As also reported by the official Z3 website, `http://research.microsoft.com/en-us/um/redmond/projects/z3/arith-tutorial/`

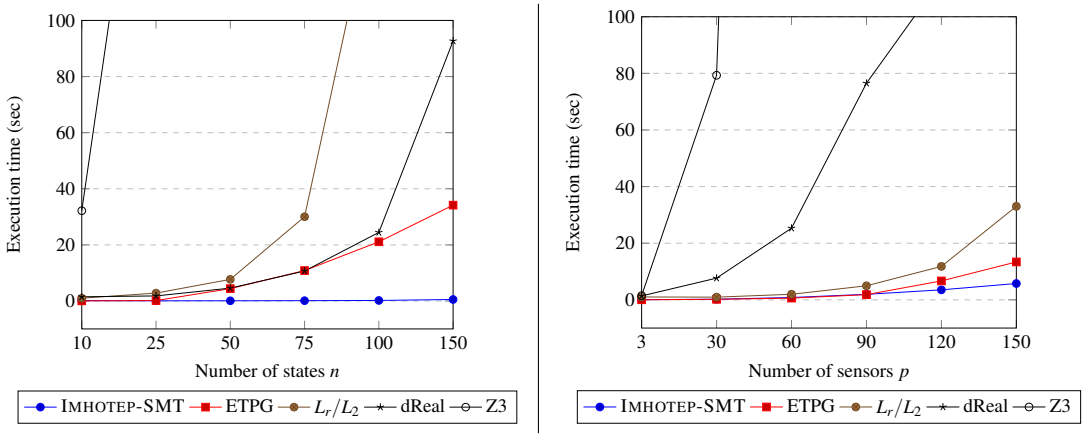


Figure 3: IMHOTEP-SMT scales better than previously proposed approaches.

based tools implementing the ETPG [ST13] and L_r/L_2 algorithms [FTD14]. Because these tools depend on convex relaxations of the original secure state estimation problem, they also led to incorrect results in some of our experiments.

5 Conclusions

We have presented IMHOTEP-SMT, a tool that uses an SMT-based approach to efficiently solve the secure state estimation problem for dynamic systems in which sensor measurements are affected by noise and corrupted by malicious attacks. IMHOTEP-SMT is able to detect the corrupted sensors and retrieve an estimate of the actual state, which is key to the deployment of control strategies for system resiliency. We have applied IMHOTEP-SMT to the detection of attacks in electric power grids and we have shown with numerical experiments that it outperforms other state-of-the-art approaches to secure state estimation.

References

[BP10] Daniel L. Berre and Anne Parrain. The Sat4j library, release 2.2. *Journal on Satisfiability, Boolean Modeling and Computation*, 7:59–64, 2010.

[CWH15] Micelle S. Chong, Masashi Wakaiki, and Joao P. Hespanha. Observability of linear systems under adversarial attacks. In *The 2015 IEEE American Control conference (ACC)*, 2015. accepted.

[DCvdW03] Jean-Michel Dion, Christian Commault, and Jacob van der Woude. Generic properties and control of linear structured systems: a survey. *Automatica*, 39(7):1125 – 1144, 2003.

[DMB08] Leonardo De Moura and Nikolaj Björner. Z3: An efficient SMT solver. In *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, TACAS'08/ETAPS'08, pages 337–340, Berlin, Heidelberg, 2008. Springer-Verlag.

[FHT+07] Martin Franzle, Christian Herde, Tino Teige, Stefan Ratschan, and Tobias Schubert. Efficient solving of large non-linear arithmetic constraint systems with complex Boolean structure. In *JSAT Special Issue on SAT/CP Integration*, pages 209–236, 2007.

[FTD14] Hamza Fawzi, Paulo Tabuada, and Suhas Diggavi. Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Transactions on Automatic Control*, 59(6):1454–1467, June 2014.

[GAC12] Sicun Gao, Jeremy Avigad, and Edmund M. Clarke. δ-complete decision procedures for satisfiability over the reals. In *Proceedings of the 6th International Joint Conference on Automated Reasoning*, IJCAR'12, pages 286–300, Berlin, Heidelberg, 2012. Springer-Verlag.

[GKC13] Sicun Gao, Soonho Kong, and Edmund M. Clarke. dReal: An SMT solver for nonlinear theories over the reals. volume 7898 of *Lecture Notes in Computer Science*, pages 208–214. Springer Berlin Heidelberg, 2013.

[Lan11] Ralph Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security and Privacy Magazine*, 9(3):49–51, 2011.

[LNR09] Yao Liu, Peng Ning, and Michael K. Reiter. False data injection attacks against state estimation in electric power grids. In *Proceedings of the 16th ACM conference on Computer and communications security*, CCS '09, pages 21–32, New York, NY, USA, 2009. ACM.

[NPSSV10] Pierluigi Nuzzo, Alberto Puggelli, Sanjit A Seshia, and Alberto L. Sangiovanni-Vincentelli. CalCS: SMT solving for non-linear convex constraints. In *Formal Methods in Computer-Aided Design (FMCAD), 2010*, pages 71–79, Oct 2010.

[PDB13] Fabio Pasqualetti, Florian Dorfler, and Francesco Bullo. Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11):2715–2729, Nov 2013.

[SMTS13] Yasser Shoukry, Paul D Martin, Paulo Tabuada, and Mani B Srivastava. Non-invasive spoofing attacks for anti-lock braking systems. In *Workshop on Cryptographic Hardware and Embedded Systems*, G. Bertoni and J.-S. Coron (Eds.): CHES 2013, LNCS 8086, pages 55–72. International Association for Cryptologic Research, 2013.

[SNP+14] Yasser Shoukry, Pierluigi Nuzzo, Alberto Puggelli, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia, and Paulo Tabuada. Secure State Estimation Under Sensor Attacks: A Satisfiability Modulo Theory Approach. *ArXiv e-prints*, December 2014. [online] `http://arxiv.org/abs/1412.4324`.

[SPN+15] Yasser Shoukry, Alberto Puggelli, Pierluigi Nuzzo, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia, and Paulo Tabuada. Sound and complete state estimation for linear dynamical systems under sensor attack using satisfiability modulo theory solving. In *Proc. IEEE American Control conference*, 2015. to appear.

[ST13] Yasser Shoukry and Paulo Tabuada. Event-Triggered State Observers for Sparse Sensor Noise/Attacks. *ArXiv e-prints*, September 2013. [online] `http://arxiv.org/abs/1309.3511`.

A Computing the Upper Bound on the Security Index

In this section we provide implementation details about the algorithms we use in the offline configuration phase of IMHOTEP-SMT in order to compute an upper bound for the system security index, and suggest which sensors should be "secured", since they are most likely to compromise, if corrupted, the estimation of the system state.

A state of the linear dynamical system (defined in (1)) is called *secure* whenever it can be estimated from measurements collected out of multiple sensors. More precisely, if an attacker is able to corrupt s sensors, then the necessary condition for system state to be secure is that the system has at least $2s+1$ sensors [ST13]. For example, consider an unmanned vehicle using three GPS sensors for navigation. If the attacker is able to corrupt one of these GPS readings (e.g., $s = 1$), the vehicle can still estimate the actual information on its position, by applying a majority voting scheme over its sensors. On the other hand, if the vehicle is equipped with only two sensors, such a majority voting scheme is apparently inapplicable. The combinatorial test known as *sparse observability test* [ST13] generalizes this idea to heterogeneous sensors (i.e., sensors that measure different physical quantities) and multi-dimensional

system states, by enumerating all possible cases in which all the states can be estimated using different subsets of sensors to secure the system. The security index $\bar{s}$ is then computed by determining the worst case subset of sensors which, once corrupted, precludes the estimation of any of the states. Our objective in this appendix is to construct an upper bound on the security index denoted by $\tilde{s}$ (i.e., $\bar{s} \leq \tilde{s}$).

As a consequence of the definitions above, a necessary condition for a system to be secure is that, for each individual state, it is possible to find multiple sensors that are able to correctly estimate it. Hence, by representing the system as a directed acyclic graph (DAG), where source nodes are sensors and target nodes are states, we can detect which states are more vulnerable to attacks as well as an upper bound on the security index. Observability properties of systems that are abstracted as graphs have been investigated in the control theory literature [DCvdW03] in terms of "structural observability". In the sequel, we provide details on our algorithms.

A.1 Constructing Structural Abstractions

We recall that the system has a total of p sensors. Then, a structural abstraction of the system can be computed from the observability matrix $\mathscr{O}_i$ (defined in (3)) of the ith sensor. In the attack-free case, this matrix acts as the linear map from the system state to the measurements of the ith sensor, i.e., $Y_i = \mathscr{O}_i x$.

We start by abstracting the observability matrix by considering only its pattern. Given the matrix $\mathscr{O}_i$, we define the pattern of this matrix, denoted by $[\mathscr{O}_i]$, as the matrix obtained from $\mathscr{O}_i$ by marking each non-zero element, as in the following example:

$$\mathscr{O}_i = \begin{bmatrix} 5 & 0 \\ 4 & 2 \end{bmatrix} \qquad \Rightarrow \qquad [\mathscr{O}_i] = \begin{bmatrix} \times & 0 \\ \times & \times \end{bmatrix}.$$

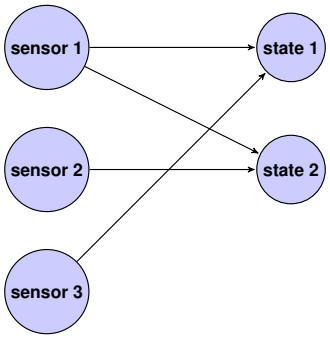
We note that $[\mathscr{O}_i]$ is a $\tau \times n$ matrix, where n is the number of states and τ is the length of the window over which the measurements are collected. We then proceed by building an *influence graph* that illustrates how states influence sensor measurements. The construction of an influence graph can be summarized as follows:

1. Construct a DAG that consists of two partitions having, respectively p and n nodes.
2. For the ith node in the first partition, draw an edge connecting to the jth node of the second partition if the jth column of $[\mathscr{O}_i]$ has at least one non-zero element.

Intuitively, the nodes in the first partition represent sensors while the nodes in the second partition represent the system states. An edge exists between a sensor node and a state node whenever the pattern matrix $[\mathscr{O}_i]$ shows that the state influences the measurements collected from ith sensor. For example, consider a system having two states x_1 and x_2 and three sensors Y_1, Y_2 and Y_3, whose associated observability matrices have the following patterns:

$$[\mathscr{O}_1] = \begin{bmatrix} 0 & \times \\ \times & \times \end{bmatrix}, \qquad [\mathscr{O}_2] = \begin{bmatrix} 0 & \times \\ 0 & \times \end{bmatrix}, \qquad [\mathscr{O}_3] = \begin{bmatrix} \times & 0 \\ \times & 0 \end{bmatrix}$$

Then, the corresponding influence graph will be:



By computing the number of inflow edges for each node of the state partition of the influence graph, we can then determine the most vulnerable, i.e. the least observable, state. Let $\mathscr{I}_j$ be the number of inflow edges for the jth state. The upper bound on the security index is then computed as:

$$\widetilde{s} = \left\lfloor \frac{\min_j \mathscr{I}_j - 1}{2} \right\rfloor,$$

which ensures that each state is measured using at least $2\widetilde{s}+1$ sensors. In our example, $\widetilde{s}$ is zero, which means that, if any sensor is under attack, at least one of the states cannot be estimated correctly.

A.2 Providing Suggestions on Vulnerable Sensors

We can use the influence graph shown in the previous section to provide suggestions on which sensors may need to be physically secured when the user assumes a security index higher than the theoretical bound. Intuitively, sensors which influence the states with a low "structural" observability are vulnerable, and may need an additional layer of security.

Let o_i denote the number of outflow edges from the ith sensor. Let also sense(j) be the set of sensors that have edges with the jth state. IMHOTEP-SMT declares the following sensors:

$$i^* = \arg\max_{i \in \mathbb{I}} o_i, \qquad \text{where} \qquad \mathbb{I} = \text{sense}(\arg\min_j \mathscr{I}_j)$$

as the sensors that needs to be secured. Intuitively, this is the sensor that measures the worst case observable state and as many other states as possible.