Attack Recovery for Cyber-Physical Systems

Fanxin Kong[1] and Zifan Wang[2]
[1] University of Notre Dame, fkong@nd.edu
[2] Syracuse University, zwang345@syr.edu

Abstract. Cyber-physical systems (CPSs) rely on computing components to control physical objects, and have been widely used in real-world life-critical applications. However, a CPS has security risks by nature due to the integration of many vulnerable subsystems, which adversaries exploit to inflict serious consequences. Among various attacks, sensor attacks pose a particularly significant threat, where an attacker maliciously modifies sensor measurements to drift system behavior. There is a lot of work in sensor attack prevention and detection. Nevertheless, an essential problem is overlooked: recovery–what to do after detecting a sensor attack, which needs to safely and timely bring a CPS back. We aim to highlight the need to investigate this problem, outline its four key challenges, and provide a brief overview of initial solutions in the field.

Keywords: cyber-physical systems · sensor attack · attack recovery

1 Introduction

Cyber-physical systems (CPSs) integrate computing units and physical components, involving a feedback control loop. The computing units indicate control signals for actuators and guide the physical components. These systems heavily rely on multiple sensors that constantly monitor internal system states (e.g. speed and position) and observe environmental conditions (e.g. obstacles).

Given the significance of sensors in CPSs, sensor attacks become a threatening risk. An attacker can alter sensor data to negatively interfere with the physical world. Compromised sensors may give false readings and misperceive the targets, and thus deceive the controllers to yield misleading control demands and cause serious consequences.

Extensive efforts have been made to defend against sensor attacks, most of which focus on prevention and detection. On the one hand, attack prevention assumes that CPSs might be under attack and aims to minimize the impact of attacks by proactive measures, for instance, attack-resilient sensor fusion, state estimation, and hidden attack defense [6]. On the other hand, attack detection usually allows attacks (if occur) to affect the system and identifies attacks using the caused impact. For example, many works perform detection by tracking anomalies between sensor measurements and expected values, based on system models [12] or sensor correlation [2, 4].

Despite these efforts, a critical problem–what to do after detecting an attack, remains elusive. We name this post-detection problem as sensor attack recovery. Addressing this problem is essential because a CPS may keep drifting if continuing to act on the malicious sensor data. Misleading control demands caused by the attack may eventually drive the system to unsafe states if no proper actions

are taken. The recovery needs to stop the drifting and reverse the negative impact caused by the attack [7–10]. Despite this importance, much less attention is paid to sensor attack recovery, compared to attack detection [1].

We thus attempts to draw attention from both researchers and practitioners to investigate the sensor attack recovery problem. For this goal, we present i) the recovery problem description in Section 2, ii) four key challenges to address this problem in Section 3, iii) initial works that have been done so far in Section 4.

2 Problem Description

Sensor attack recovery is a post-detection problem that aims to bring a system back to target states after detecting a sensor attack. Note that the recovery problem here is to restore the physical state of a system, which is different from the cyber recovery that restores cyber states such as values of variables [11].

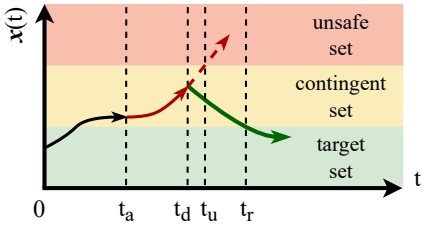

Fig. 1. Attack Recovery Workflow.

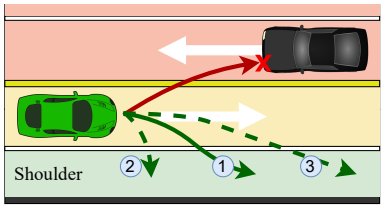

Fig. 2. An Illustrative Example.

As shown in Fig. 1, the physical state space of a system is divided into three sets: unsafe, contingent, and target sets. The unsafe set is a set of physical states that a system must not reach; otherwise, serious consequences may occur. The target set is defined as a set of states that the recovery needs to bring a system into after the detection of an attack. The contingent set refers to a set of states that a system can stay in when there are no attacks but cannot when sensors are compromised. The contingent and target sets together are called the safe set.

We consider two operational modes: normal and recovery. A system operates in the normal mode when there are no attacks, and switches to the recovery mode after attack detection. In the recovery mode, appropriate recovery methods need to be applied and drive the system to the target set. Fig. 1 illustrates the transition from the normal to recovery modes. The system operates normally before t_a. An attack starts at t_a and is detected after some time, i.e. at t_d. Then, the system transits to the recovery mode, and is steered back to the target set at t_r, as shown by the green solid curve. Note if no recovery is applied, the system may keep drifting and reach unsafe set at t_u, shown by the red dotted curve.

Fig. 2 shows an example to illustrate and motivate the recovery problem. Consider an autonomous vehicle running on a two-lane road. The unsafe set is the opposite lane (the red region), the contingent set is the system's own lane (the yellow region), and the target set is the shoulder (the green region). Note that if the vehicle stays in the contingent set after attacks, it may end up being unable to sense obstacles ahead due to the attack and crashing them. Thus, it needs to be steered to the target set and stopped there for safety.

3 Key Challenges

Sensor Data Credibility. The first challenge is data credibility, that is, after detecting a sensor attack, how to determine trustworthy data that can be used for recovery. Two primary sources are typically considered: measurements from uncompromised sensors and historical data recorded before an attack starts.

For the first source, the attack detector needs to localize those reliable sensors and use their readings for further system state estimation and recovery while excluding interference from corrupted sensors. Failure to differentiate between attacked and normal sensors can lead to recovery failure due to mixed readings. False positives (correct sensors labeled as attacked) limit data for precise estimation, while false negatives (corrupted sensors determined as normal) cause risky state assessment and recovery failure. Regarding the second source, historical sensor data, it's essential to precisely determine the attack onset. Reliable sensor measurements before the attack can initiate state estimation and recovery. Accurate attack onset diagnosis is crucial, as an erroneous diagnosis risks unreliable and hazardous state estimation, undermining recovery efforts.

System State Estimation. The second challenge is system state estimation. System estimation means using (a part of the) sensor readings to compute the real/true system states, such as locations and speeds. Fast, accurate, and efficient system state estimation is of great importance in system recovery, because the recovery needs to know the current physical state of a vehicle in order to control it to the target set. Under attacks, state estimation becomes significantly challenging. Attacks may have been launched quite some time ago, and the system can only use old historical readings of a part of the sensors.

Solutions in this context depend on two factors: the proportion of sensors that can be deemed trustworthy and the freshness of the historical data obtained from these reliable sensors. Ample redundancy and relatively new data enable the system to reconstruct its state with ease. However, in many instances, the reconstructions of the system state carry a significant degree of uncertainty because of cost control (reducing redundancy) and the first challenge (sensor data credibility). When uncertainty cannot be effectively reduced, a prudent approach is to act conservatively. This entails utilizing the most pessimistic estimations and generating control sequences that prioritize safety over other considerations.

Recovery Functionality. The third challenge is how to safely recover a system to the target set. Safe recovery differs from the normal system running because a control sequence that is benign for an unattacked system may not be applicable for systems under attack. Thus inappropriate recoveries that derive from normal system runnings may drive a system to an unsafe set and cause danger.

We need to consider proper physical constraints and safety requirements, and the recovery should generate control sequences that do not violate them. Further, due to the unpredictability of attacks, a system may sit in different states when attacks are detected. The recovery needs to be able to dynamically check and compute in order to accommodate this unpredictability.

Satisfying the safety requirements is crucial for recovery functionality. If with the knowledge of accurate system dynamics, model-based recovery methods can embed the requirements in the controller design. By contrast, if without such knowledge, data-driven methods such as machine learning techniques can be applied. However, making data-driven methods satisfy the safety requirements is already very difficult for a system running in normal mode. Developing such methods in the context of sensor attacks will be even more challenging.

Recovery Speed. The last but not least challenge is the recovery speed. Recovery speed refers to how fast it is to bring a vehicle to the target set. Determining appropriate recovery speed is tricky due to two aspects as follows.

Generally, fast recovery is key, since uncertainties accumulate over time, potentially causing recovery failure. Uncertainty stems from two sources: accumulated sensor errors and unforeseen environmental factors, e.g., sudden obstacles.

On the other hand, one should not take "as fast as possible" for granted. Users' experience is also a factor. For example, as shown in Figure 2, trajectory 2 denotes a very fast recovery, while trajectory 1 is moderate, and trajectory 3 is slow. Trajectory 2 immediately reaches the target set with a sharp turn, thus discomforting the passengers and causing other safety issues. Hence, moderate recovery speed may be more reasonable, balancing the above aspects.

Several steps are needed to settle this challenge. First, recovery systems should compute a recovery deadline, beyond which the system will fall into an unsafe set. After calculating the deadline, the recovery system should select the best target recovery speed considering objectives, scenarios, and user preferences. At last, the recovery systems should dynamically adjust the recovery speed according to changing environments.

4 State-of-the-art Recovery Solutions

Shallow Recovery. Shallow recovery refers to the recovery that still uses the original controller in the recovery mode; while deep recovery refers to the recovery that has specific controllers in the recovery mode. This subsection presents shallow recovery papers and the next subsection does deep recovery ones.

Ref. [5] is regarded as the sperm work that investigates the sensor attack recovery problem. It proposes a new concept of physical state recovery, where the essential operation or behavior is defined as rolling the system forward from consistent historical system states. This work develops a procedure that leverages historical data to recover failed system states. It also designs a checkpointing protocol that defines how to record system states for recovery. Specifically, the protocol introduces a sliding window that accommodates attack detection delay to improve the correctness of stored states. A similar idea has also been applied to various systems including chemical processes, robotic vehicles, and power systems. These works rely on the system model to do state estimation. Different from them, Ref. [3] develops a data-driven attack recovery framework: a deep learning-based prediction model that exploits the temporal correlations estimates system states after attack detection.

Deep Recovery. Deep recovery refers to the recovery that has specific controllers in the recovery mode. The works above do not have such controllers, instead mainly perform state estimation in the presence of sensor attacks. Thus, they are unable to manage the recovery process. To address attack recovery, Ref. [7] proposes a formal method-based attack-recovery control that includes the removal of poisoned data, estimation of the current state, prediction of reachable states, and online design of a new controller to recover the system. This work is regarded as the first one that deeps into the recovery control level. Specifically, it solves a reach-avoid problem for a Linear Time-Invariant (LTI) system while considering in-negligible uncertainties. The computed recovery control is guaranteed to work on the original system if the LTI model has less behavioral difference with system dynamics than an error bound. In order to run online with limited computational resources, this work builds a linear programming restriction with constrained safety and target specifications and then finds a solution through a linear programming solver. The results show that the proposed recovery can steer a system back to a target state in a safe and real-time manner.

Ref. [9] significantly improves [7] by considering recovery trajectory oscillation, maintainable time, and computational overhead. Specifically, a real-time recovery system that addresses the recovery speed was provided, using Linear-Quadratic Regulator (LQR) based recovery control calculator that concerns timing and safety constraints can smoothly steer a system back to a target state set before a safe deadline and maintain the system state in the set once it is driven to it. Also, a checkpointer, a state reconstructor, and a deadline estimator are designed to realize the in-time recovery. Compared to [7], Ref. [9] can significantly reduce the oscillation of recovery trajectory and maintain the system in the target set for a while. The cost of this improvement is the increased computational overhead, which is, however, acceptable by the experimental results.

While Ref. [7,9] focus on linear systems, Ref. [10] addresses non-linear system recovery, which is more practical in real-world CPS applications. It incorporates a state predictor that leverages Flow*, a tool specifically designed for efficient non-linear reachability analysis. In addition, it regularly updates linear approximations based on the current state estimate, ensuring a high degree of accuracy within a small range. Compared to Ref. [7,9], another feature of Ref. [10] is that it can leverage uncorrupted sensor data to enhance recovery performance. Once attacked sensors are diagnosed, the recovery control sequence generator uses uncorrupted sensor measurements as feedback at each activation, which prevents the uncertainty from exploding in the uncorrupted dimensions.

Toolbox. At last, a CPS attack recovery toolbox is developed in [8]. It mimics a general CPS, where sensors gather system state data transmitted to observers. Environmental uncertainties or sensor attacks can distort these observed states, affecting estimates. Controllers use these estimates to generate control signals, applied to plant simulators that update system states based on dynamics. The toolbox accommodates various attack strategies, detectors, and recovery controllers, making it a versatile platform for sensor attack recovery experiments. More information is available at: https://sim.cpsec.org.

5 Conclusion

We introduce CPS sensor attack recovery, highlighting its distinction from prevention and detection. Driving CPSs towards target sets is the motivation for recovery. We present four major challenges and present current solutions. A comprehensive view of attack recovery is offered. We highlight the importance and challenges of CPS sensor attack recovery. Despite some preliminary work, the field requires more development to achieve robust attack recovery solutions.

Acknowledgement. This work was supported in part by NSF CNS-2333980.

References

1. Akowuah, F., Kong, F.: Physical invariant based attack detection for autonomous vehicles: Survey, vision, and challenges. In: 2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD). pp. 31–40. IEEE (2021)
2. Akowuah, F., Kong, F.: Real-time adaptive sensor attack detection in autonomous cyber-physical systems. In: 2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS). pp. 237–250. IEEE (2021)
3. Akowuah, F., Prasad, R., Espinoza, C.O., Kong, F.: Recovery-by-learning: Restoring autonomous cyber-physical systems from sensor attacks. In: 2021 IEEE 27th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA). pp. 61–66. IEEE (2021)
4. He, T., Zhang, L., Kong, F., Salekin, A.: Exploring inherent sensor redundancy for automotive anomaly detection. In: 2020 57th ACM/IEEE Design Automation Conference (DAC). pp. 1–6. IEEE (2020)
5. Kong, F., Xu, M., Weimer, J., Sokolsky, O., Lee, I.: Cyber-physical system checkpointing and recovery. In: 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS). pp. 22–31. IEEE (2018)
6. Liu, M., Zhang, L., Lu, P., Sridhar, K., Kong, F., Sokolsky, O., Lee, I.: Fail-safe: Securing cyber-physical systems against hidden sensor attacks. In: 2022 IEEE Real-Time Systems Symposium (RTSS). pp. 240–252. IEEE (2022)
7. Zhang, L., Chen, X., Kong, F., Cardenas, A.A.: Real-time attack-recovery for cyber-physical systems using linear approximations. In: 2020 IEEE Real-Time Systems Symposium (RTSS). pp. 205–217. IEEE (2020)
8. Zhang, L., Liu, M., Kong, F.: Simulation and security toolbox for cyber-physical systems. In: 2023 IEEE 29th Real-Time and Embedded Technology and Applications Symposium (RTAS). pp. 357–358. IEEE Computer Society (2023)
9. Zhang, L., Lu, P., Kong, F., Chen, X., Sokolsky, O., Lee, I.: Real-time attack-recovery for cyber-physical systems using linear-quadratic regulator. ACM Transactions on Embedded Computing Systems (TECS) (2021)
10. Zhang, L., Sridhar, K., Liu, M., Lu, P., Chen, X., Kong, F., Sokolsky, O., Lee, I.: Real-time data-predictive attack-recovery for complex cyber-physical systems. In: 2023 IEEE 29th Real-Time and Embedded Technology and Applications Symposium (RTAS). pp. 209–222. IEEE (2023)
11. Zhang, L., Wang, Z., Kong, F.: Optimal checkpointing strategy for real-time systems with both logical and timing correctness. ACM Transactions on Embedded Computing Systems (TECS) (2023)
12. Zhang, L., Wang, Z., Liu, M., Kong, F.: Adaptive window-based sensor attack detection for cyber-physical systems. In: Proceedings of the 59th ACM/IEEE Design Automation Conference (DAC). pp. 919–924 (2022)