Towards Synthesis of Platform-aware Attack-Resilient Control Systems *

Extended Abstract

Miroslav Pajic[1] Nicola Bezzo[1] James Weimer[1] Rajeev Alur[1]
Rahul Mangharam[1] Nathan Michael[2] George J. Pappas[1] Oleg Sokolsky[1]
Paulo Tabuada[3] Stephanie Weirich[1] Insup Lee[1]

[1]School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104
{pajic, nicbezzo, weimerj}@seas.upenn.edu
{rahulm, pappasg}@seas.upenn.edu
{sokolsky, alur, sweirich, lee}@cis.upenn.edu

[2] Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
nmichael@cmu.edu

[3]Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, CA 90095
tabuada@ee.ucla.edu

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Unauthorized access (e.g., hacking, phreaking)* ; C.3 [**Special-purpose and Application-based Systems**]: [Process control systems, Real-time and embedded systems]

Keywords

Attack-resilient control systems, cyber-physical system security

1. INTRODUCTION

Over the past decade, the design process in the automotive industry has gone through a period of significant changes. Modern vehicles present a complex interaction of a large number of embedded Electronic Control Units (ECUs), interacting with each other over different types of networks. Furthermore, there is a current shift in vehicle architectures, from isolated control systems to more open automotive architectures that would introduce new services such as remote diagnostics and code updates, and vehicle-to-vehicle communication. However, this increasing set of functionalities, network interoperability, and system design complexity may introduce security vulnerabilities that are easily exploitable.

Typically, modern vehicular control systems are not built with security in mind. As shown in [7], using simple methods an attacker can disrupt the operation of a car to either disable the vehicle or hijack it, giving the attacker the ability to control it instead. This problem is even more emphasized with the rise of vehicle autonomy; consequently, criticality analysis for various automotive components will have to be completely re-done.

To address these issues, we have introduced a design framework for development of high-confidence vehicular control systems that can be used in adversarial environments. The framework employs system design techniques that guarantee that the vehicle will maintain control, possibly at a reduced efficiency, under several classes of attacks. This comprehensive end-to-end approach to the development of vehicular control systems can be extended for the use in most networked control systems that may be subject to a variety of externally-originating attacks.

The overview of the development framework is shown in Fig. 1. To protect against the set of attacks that is as extensive and diverse as possible, we combine control-level techniques and code-level techniques:

• During the control design phase, it is necessary to address attacks on the environment of the controller, such as attacks on sensors, actuators, communication media (i.e., the network) and computational resources available to the controller. In this phase we build upon ways to introduce redundancy within the control loop, as well as methods for attack detection and identification. We utilize security-aware attack-resilient estimators that identify attacks and allow the controller to pursue a mitigation strategy. Therefore, we refer to these as *Control-level defenses*.

• In the system development phase, the framework employs *Code-level defenses* that prevent injection of malicious code into the operation of the controller itself. This is achieved by providing secure code synthesis for the derived controllers, with the goal to use a formal representation of the execution and code generation semantics to remove the uncertainty from the code generation process.

2. ATTACK-RESILIENT CONTROL SCHEMES

In this phase, we have built on the work from [3, 4] where methods for compressed sensing and error correction over the reals were used to derive a technique to develop secure state estimators when system sensors or actuators are under attack. We assume that control design for the nominal case, when no attacks are present, has

*This material is based on research sponsored by DARPA under agreement number FA8750-12-2-0247. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

HiCoNS'13, April 9–11, 2013, Philadelphia, Pennsylvania, USA.
ACM 978-1-4503-1961-4/13/04.

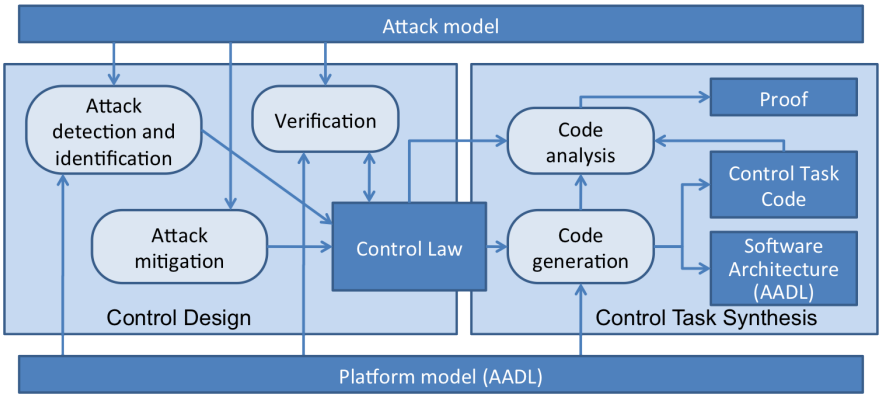

Figure 1: Overview of the approach.

already been performed. Hence, we focus on state estimation techniques that extend conventional state estimators to compensate for exogenous attacks or to provide indications to the controller that the system is under attack.

Most of the existing schemes for attack detection and identification (e.g., [8, 3, 4]) are based on the knowledge of the exact plant model. However, as the environmental conditions might affect some of the model parameters, we combine the resilient estimator with a controller scheme we have introduced in [9], which is resilient to both attacks and limited perturbations in model parameters. Thus, the overall controller design (shown in Fig. 2) guarantees that in the event the model becomes inaccurate, we can maintain a minimum performance level for the closed-loop system.

Our framework has been illustrated on a cruise control case study, in which a vehicle employs redundant sensor measurements (e.g., encoders, GPS, IMU) to maintain a predefined velocity even in the case of attacks. In this regard, we have created a simulator, based on the recently developed Robotic Operating System (ROS) [2], to emulate the dynamics of a real unmanned ground vehicle. Using this system, we have been able to demonstrate that our resilient control strategy can guarantee a safe performance when less than half of the sensors are under attack.

3. SAFE CODE GENERATION

Existing methods for designing secure control systems do not offer coordinated control-level and code-level defenses. Tools like Matlab allow us to model control laws and generate code, but security properties of the generated code are not well studied. Our aim is to prove that behavior of the control algorithm is preserved by code generation and that no vulnerabilities are introduced in the process.

The code of control tasks is executed on top of the underlying communication and computation platform. Therefore, proofs of control code execution need to take properties of the platform into account. We plan to precisely specify the services provided by the platform and use these specifications in proof construction. We use

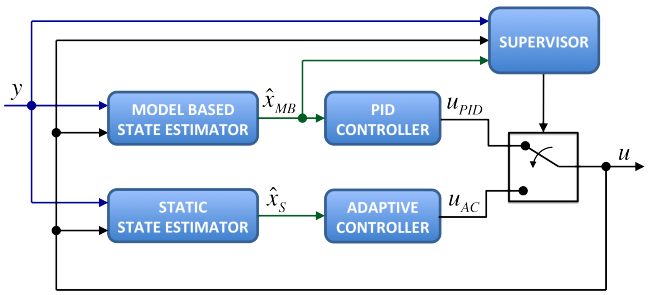

Figure 2: Diagram of the resilient controller.

architectural modeling to describe both the structure of the control software and the capabilities of the platform that runs it. To achieve this goal we utilize the Architecture Analysis and Design Language (AADL) [5], developed for modeling embedded control system architectures.

Code generation is performed by analyzing dependencies between expressions in the control law and generating code for the individual expressions in the topological order of dependencies. In this respect, code generation is similar to the one performed by the Simulink Real-Time Workshop tool [1]. Platform-dependent aspects — that is, access to specific sensors and actuators, handling of timers, etc. — should be "weaved in" afterwards. Currently, we perform this part manually. Our plan, however, is to enhance the code generator to perform this automatically based on the AADL model, using an approach similar to that of Ocarina [6].

4. REFERENCES

[1] Real-Time Windows Target - Run Simulink models on a PC in real time. http://www.mathworks.com/products/rtwt/.

[2] Robotic Operating System. http://www.ros.org.

[3] H. Fawzi, P. Tabuada, and S. Diggavi. Secure state-estimation for dynamical systems under active adversaries. In *49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 337–344, 2011.

[4] H. Fawzi, P. Tabuada, and S. Diggavi. Security for control systems under sensor and actuator attacks. In *Proceedings of the 51st IEEE Conference on Decision and Control*, 2012.

[5] P. Feiler, B. Lewis, and S. Vestal. The SAE AADL standard: A basis for model-based architecture-driven embedded systems engineering. In *Workshop on Model-Driven Embedded Systems*, 2003.

[6] J. Hugues, B. Zalila, L. Pautet, and F. Kordon. From the prototype to the final embedded system using the Ocarina AADL tool suite. *ACM Transactions on Embedded Computing Systems*, 7(4):42:1–42:25, 2008.

[7] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. Experimental security analysis of a modern automobile. In *2010 IEEE Symposium on Security and Privacy (SP)*, pages 447 –462, 2010.

[8] F. Pasqualetti, F. Dörfler, and F. Bullo. Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 2012. submitted.

[9] J. Weimer, N. Bezzo, M. Pajic, G. Pappas, O. Sokolsky, and I. Lee. Resilient adaptive control with application to vehicle cruise control. In *Workshop on Control of Cyber-Physical Systems*, 2012. submitted.