Editorial: Schizoid Design for Critical Embedded Systems

Recently, I finished teaching a course on cybersecurity of cyber-physical systems. It was a new graduate-level course in which I wanted to simultaneously learn and teach the importance of considering cybersecurity as a design requirement for all critical infrastructures—national power grids, railway and air traffic control—transportation systems in general, water/sewage system of a municipality, and other systems of critical importance for the economic sustenance and growth of an economy. The point I wanted to impress upon the young minds is that cybersecurity as a secondary thought, as an add-on, does not work well, especially in view of many recent incidents of cyberattacks—the poster example being Stux-net and plenty of other examples being abound.

I have written about cybersecurity and its relevance to embedded systems in line with the cautionary messages from almost all government agencies, such as NIST, the U.S. Departments of Energy and Homeland Security, and many other bodies of industry. The technology and the science of cybersecurity of cyber-physical systems, particularly the large-scale embedded systems manifest in our national critical infrastructures, are important and are being developed by researchers. However, the cognizance of the imminent threat to our social and economic order from cyberborne attacks—by the common public, and more importantly by our young engineers, is extremely important. Awareness at all levels—high school through graduate school, manual factory workers to CIOs of companies, consumers to the politicians—cannot be overemphasized.

Interestingly, while teaching the course, and being a novice to the topic, I was learning the literature just ahead of the students. The most important lesson learned through teaching this course is demonstrative. I ended up dedicating almost half of the lectures to machine learning and/or data analytics. Of course, if I teach the course again, I would probably require students to have taken a course in machine learning; however, having not done so initially, I had the onus to introduce machine learning techniques that allows one to build models from massive amounts of data, either synthetic or real.

In the past, when teaching how to design embedded systems with explicit goals of delivering robust, reliable, and resilient system, I depended mostly on model-based design methods. In this paradigm, one depends on the physics and abstract computation models to create models based on modeling precepts—ones learned from standard computer science curriculum. The requirements of the system functionality, as well as the methodology, tools, and familiarity with specific modeling languages, dictated the model construction. This abstract model would then be elaborated with implementation choices and details, and the final model close to the implementation, such as the register-transfer-level models in the case of hardware, would be subject to evaluation. Students were taught how to carry out functional testing, performance measurements, and various other methods to evaluate robustness to input assumptions, reliability to known fault models, and resilience to unknown faults, usually modeled with statistical conditions. More recently, cybersecurity would be considered a concern, and the implementation model might be subject to various evaluations, such as side-channel analysis and vulnerability detection.

Unfortunately, in really large distributed cyber-physical system–based critical infrastructure, one has to take a different take on the design process from a cybersecurity point of view. One has to enter the mind-set of a cyberattacker and create threat models that would constitute the touchstone of secure design process.

For example, one has to consider the various observation or sampling points of the system (e.g., sensors to read or sample the physical environment) and think how an attacker might compromise the unobservability of those points without authentication, what knowledge of the system dynamics or the control mechanism of the system might be actually reconstructed by the attacker, and how little information would be needed to do so. One also has to consider the actuation points of the system and ponder the least number of such actuation points that the attacker may take over to disrupt the dynamics of the system enough to create considerable damage. Another point to consider is how to obfuscate the dynamics of the system when certain sensing or actuation points are compromised. However, the scariest part is to know that a large percentage of attacks are induced by inside attackers who have possible authentication of controlling parts of the system, if not the whole, or can access certain points in the system. In such cases, the attack cannot be avoided, and one is confronted with the question of whether there is enough indication of an ongoing attack in the dynamics of the system itself.

This approach to viewing the system from an adversarial position requires one to topple the paradigm that we usually teach students, over its head, as well as teach them how to build models from data and not just generate data from models. Students should know how to observe the system in action—even through partial observations—and construct a model close enough to the real system model, and use the partial access to create damage because the approximate model allows them to do so.

Almost like a schizophrenic duality, the engineer also has to wear a designer's hat and consider a game in which the observations are obfuscated enough to render it impossible for an attacker to build any useful model to induce clever attacks. In addition, one has to consider that if the attack has not been thwarted, can one construct a dynamics from unobfuscated observations quickly enough so that the difference between the expected dynamics and the real dynamics can trigger alarms to alert system administrators?

Data analytics and machine learning have been used by economists for decades to construct models from observations, as economists make predictions, analyze, and try to thwart disasters. Atmospheric scientists have a similar approach. But as designers, when we have to do put on that hat, we need a paradigm shift in our teaching of the design of embedded systems, particularly the specific ones discussed here.

Now I must focus on the issue at hand. This is the third issue of Volume 14, and first of this volume to include a special issue—*Embedded Platforms for Cryptography in the Coming Decade*. My sincere thanks go to Patrick Schaumont, Maire O'Neill, and Tim Guneyasu for their great leadership during this process. This issue is important to ACM *TECS* because we have not had many submissions on the forward-looking aspect of cryptoengineering, and these editors have worked tirelessly with the reviewers to select five excellent papers from 18 submissions. Some of the papers were deemed suitable for *TECS* but required multiple revisions and will appear in upcoming regular issues.

Also in this issue, we have 15 papers on topics ranging from application mapping and scheduling to many-core processors, power- and energy-constrained embedded systems, concurrency, and real-time applications. Most of these articles present topics at the chip level or application issues close to the hardware.

Once again, submissions related to cybersecurity, machine learning, the Internet of Things, formal methods, and sustainable and green embedded computing are few. I

encourage submissions on all topics associated with embedded computing; however, submissions that focus on the preceding topics would be highly welcome and lead to a greater reflection of the overall state of activities in the field of embedded computing at large.

Sandeep K. Shukla
Virginia Tech
Editor-in-Chief