SAND2017-7025C

Trustworthy Design Architecture: Cyber-Physical System

Sung Choi, Adrian Chavez, Marcos Torres
Sandia National Laboratories, Albuquerque, NM
Cheolhyeon Kwon, Inseok Hwang
Purdue University, West Lafayette, IN

Abstract- **Conventional cyber defenses require continual maintenance: virus, firmware, and software updates; costly functional impact tests; and dedicated staff within a security operations center. The conventional defenses require access to external sources for the latest updates. The whitelisted system, however, is ideally a system that can sustain itself freed from external inputs. Cyber-Physical Systems (CPS), have the following unique traits: digital commands are physically observable and verifiable; possible combinations of commands are limited and finite. These CPS traits, combined with a trust anchor to secure an unclonable digital identity (i.e., digitally unclonable function [DUF] – Patent Application #15/183,454; CodeLock), offers an excellent opportunity to explore defenses built on whitelisting approach called "Trustworthy Design Architecture (TDA)." There exist significant research challenges in defining what are the physically verifiable whitelists as well as the criteria for cyber-physical traits that can be used as the unclonable identity. One goal of the project is to identify a set of physical and/or digital characteristics that can uniquely identify an endpoint. The measurements must have the properties of being reliable, reproducible, and trustworthy. Given that adversaries naturally evolve with any defense, the adversary will have the goal of disrupting or spoofing this process. To protect against such disruptions, we provide a unique system engineering technique, when applied to CPSs (e.g., nuclear processing facilities, critical infrastructures), that will sustain a secure operational state without ever needing external information or active inputs from cybersecurity subject-matter experts (i.e., virus updates, IDS scans, patch management, vulnerability updates). We do this by eliminating system dependencies on external sources for protection. Instead, all internal communication is actively sealed and protected with integrity, authenticity and assurance checks that only cyber identities bound to the physical component can deliver. As CPSs continue to advance (i.e., IoTs, drones, ICSs), resilient-maintenance free solutions are needed to neutralize/reduce cyber risks. TDA is a conceptual system engineering framework specifically designed to address cyber-physical systems that can potentially be maintained and operated without the persistent need or demand for vulnerability or security patch updates.**

I. Introduction

An ICS is a network of interconnected devices that monitor and control physical actions in industrial environments. Technological advancements in computing have been adapted for industrial use in order to save costs, improve resiliency, and make use of standards-based technologies [1]. ICSs differ significantly from traditional enterprise networks in that reliability, timing responses, deterministic behavior, and security and integrity of messaging are much more constrained. Compromise in any one of these functional requirements can have severe consequences, including damage to expensive infrastructure equipment, plant safety, and causing worst case scenario like nuclear power plant core damage resulting in loss of life.

A current trend within industrial networking implements fieldbus protocols using wireless technologies due to significant cost savings in replacing aging wired infrastructure [2]. Other incentives include wireless networking's suitability for hazardous environments or installation on moving equipment where cabling may be easily damaged or restrict the operation of the machinery involved. Unfortunately, TCP/IP-based and wireless technologies were not designed to accommodate industrial functional requirements (e.g., determinism and real-time capabilities). Furthermore, because of the high consequences of failure (i.e., loss of life and expensive equipment) and the real-time constraints imposed by industrial network environments, the security and integrity of communication services provided by traditional TCP/IP-based solutions are not only insufficient but also inadequate. The incompatibility of IT cybersecurity solutions in ICS environment is particularly apparent for the following reasons:

1) ICSs in critical infrastructure cannot afford to go down, not even for a moment. Deploy, patch and update model in IT environment may mean disruption to power, water, transportation (ground, air and sea) and hospital services that could result in deaths and accidents.

2) ICS actuators and sensors have limited computational capabilities. Implementing IT based cybersecurity solutions adds maintenance cost to ICS deployment (i.e., cryptographic key management and device-level identity management, anti-virus and firewall updates, intrusion detection systems, patch management and OS upgrades), all of which requires heavy, computationally intensive load on the legacy kernel that runs the ICS instrumentation.

Cybersecurity industry have explored the option of utilizing "white-listed (*W*)" vs. "black-listed (*B*)" techniques for defensive capabilities for many years [3]. Analogy of *W* is that "no one shall have access to the red button except person X" vs *B* being "everyone but person X can access the red button." In the cybersecurity Industry, this is the same thing

as *W* firewall rule called Deny-All-Permit-by-Exception (DAPE). Typical samples of *B* cybersecurity applications on the other hand are IDS/IPS, anti-virus, vulnerability scans. Most cybersecurity tools are built on *B* due to the extreme complexity involved in formulating *W* rules for a standard software and operating systems (i.e., Windows 8: 50 million lines of code [LOC])[4]. Trying to legitimize on *W* commands on 50 million LOC are, at best, futile as seen by numerous cyber incidents (Yahoo!, Target, LMC, IRS, OPM, etc.). *B* model requires compiling, updating, and testing the latest, ever growing list [5]. For critical infrastructures, such as power plants, maintaining *B* model can be cost prohibitive.

Cyber-physical system (CPS), have significantly reduced, finite number of known commands that results in physically verifiable, kinetic event. This discriminating CPS feature offers a unique opportunity to explore the system exclusively built on *W* techniques for resiliency and security. Furthermore, to enable TDA, we argue that following two security traits/technologies are needed: trust anchor built on integrity and availability of identity and physically verifiable, secured *W* rules. We have explored various forms of authentication techniques [6] and came to conclude that unclonable identities (either digital or hardware) are the essential attributes needed in formulating the trust anchor (e.g., digitally unclonable function [DUF]/CodeLock).

We developed a new class of architectural framework for the CPSs capable of being highly resistant to identity theft, zero day exploits, and untrusted components. Multi-factor, active authentication technologies (i.e., DUF - Patent #62/175,753; PUF/CodeLock) can be used as a combination of "what you know/where you are" and unclonable "what you have" factor.

Successful implementation of TDA will dramatically curtail cybersecurity problems such as an adversary's ability to compromise and take over multiple identities; insider threats posed by untrusted, compromised components; and/or having to halt 24/7 industrial control systems (ICS) operations to apply patch updates. The DUF solution, for example, will remove the adversary's ability to spoof identities; with the proper integration of TDA rules, advanced persistent threats (Trojan horses and logic bombs, bad insiders) may be mitigated: No more mass identity theft for government, private industries, or public organizations. No need to rely on having to trust untrusted supply chains or worry about end-user susceptibility to phishing attacks.

We will conclude with example of potential CPS deployment plan as applied to Unmanned Arial System (UAS). The primary goal is to make conceptual framework for CPS system built on TDA model and theoretically validate the efficacy of TDA as applicable to UAS and thus proposing demonstrable experimental/simulation model for any generic ICS deployment.

II. WHAT IS TDA?

TDA is built on integrating three fundamental ideas: 1) Apply *W* rules as means of controlling the integrity and authenticity of communications between CPS component to component; 2) Use unclonable device authentication as means of a trust anchor to secure the *W* rules; 3) Solve the complexity problem resident in cyber-domain by exclusively utilizing *W* rules within the bounds of CPS functions that can be validated and measured as physical phenomena. These three principles are heavily leveraged to build explicitly *W* CPS architecture.

A. *White vs. Black List*

Replacing the word "software" with a "product," one can easily see that *W* technics are not limited to just software but that the concept of DAPE can indeed be applied to any generic CPSs. In the software industry, the plethora of vulnerabilities found every year [7] suggests that cybersecurity is often compared to an "artisanship" profession rather than a repeatable scientific/engineering profession. The differences between artisanal work and engineering work is well expressed in the SEI (Software Engineering Institute) work on capability maturity models [8]. Levels of maturity range from 1 to 5:

1. Ad-hoc, individual efforts and heroics
2. Repeatable
3. Defined
4. Managed
5. Optimizing (Science) [9]

Artistic enterprises depend solely on individuality and is entirely dependent on the (unique) skills of the individual. Engineering work aims to be objective, independent from one individual's perception and does not require unique skills. It should be reproducible, predictable and systematic.

In this context, traditional cybersecurity defenses are heavily leveraging *B* techniques (i.e., deploy and patch/update, virus updates) requiring continuous aggregation of "known bad behaviors" in order to mitigate the potential threats. The security community often suggests using methods that have artisanal characteristics such as using a particular brand of firewall, IDS or anti-virus. Increased security is often equated to how often and how broadly are the *B* data updated or implemented.

A *B* design essentially creates a list of "bad" inputs, bad characters, or other undesirable things. Unfortunately, it often fails because the enumeration is incomplete, or because the removal of bad characters from the input can result in the production of another bad input which is not caught (and so on recursively). It turns out that in complex systems, such as standard operating system or software, there are more ways to circumvent or fool the *B* mechanism then the black-list themselves. Steve McConnell, in his book "Code Complete" estimates that there are "about 15 – 50 errors per 1000 lines of delivered code." Taking the round number of 25 errors per 1000 lines of codes, Windows 8 would have average of 1.25 million software bugs, a potential built-in cybersecurity flaws [10]. *B* technique fails because they are based on previous experience, and only enumerate on a *known* bad input as well as human mind's inability to catch all possible errors.

When cybersecurity or secure programming is taught in schools and conferences, students are often taught commonly repeated mistakes and how to avoid them. Books on secure programming show lists upon lists of "sins" and errors to avoid. Those are blacklists that we are in effect creating in the minds of students [11]. Furthermore, the recommended development methods (solutions to repeated mistakes) also often take the form of black lists. Risk assessment and threat modeling require expert artisans to imagine, based on past experience, what are likely avenues of attack, and possible damage and other consequences [12]. The results of those activities are dependent upon unique skill sets, are irreproducible (ask different people and you will get different answers), and attempt to enumerate known bad things. They build black lists into the design of software development projects.

In the insurance businesses, risk assessment and threat modeling built on black listing are appropriate and acceptable. The risk of earth quakes, storms and possibilities of accidents are all based on the laws of physics and statistical evidences from the past physical events or experiences. However, in a complex system, such as software industry or ICSs where every version is different and new, *B* techniques are doomed to fail since the possible ways to fail is logarithmically dependent on the complexity level of software. For example, the more lines of codes there are, the more possibilities of failure and greater vulnerable surface areas for the attackers to explore.

There are development and software configuration methods emphasizing guaranteed/provable correctness of the codes. For example, a software solution called AppArmor uses white-listing with fine-grained software capabilities defined as acceptable commands [13]. This corresponds to building a white list of what an application can do with possible extension to limited processing at root level. Unfortunately, it may still be possible for some malicious activity to take place if the *W* rules, as well as possible combinations and permutations of rules that haven't been accounted for. In a very complex system, such as OS kernels and software applications, there are limitations to making a pure *W* rules covering all contingencies. Furthermore, we argue that, unless we can enforce explicit *W* at a OS/kernel level, application white-lists are at best a "permissive white-list.

For example, AppArmor isn't quite a pure white listed system since it is possible to bypass application whitelist by penetrating/compromising OS kernel that runs the AppArmor. Another word, AppArmor allows more than necessary functions; thus, it cannot be considered an "explicitly whitelisted" system. In order for AppArmor to be an "explicitly whitelisted" system, the whitelist must be finite and bound, must match exactly what is safe/secure to do, no more and no less. Applying *W* architecture in an unbounded complex system is logically inconsistent and cannot be achieved. Contrary to software industry, however, when applying *W* architecture on a CPS, even in the case where CPS components use unbound complex systems, such as the Linux kernel or Windows 10 operating system, we have a finite and limited digital commands that bounded by measurable, observable physical phenomena. Digital commands given in CPSs must have kinetic effects. The recommended *W* practice in TDA enumerates on known good inputs, as well as dynamically validating the CPS commands by measuring the physical events as by-product of legitimate CPS commands. For example, if the digital commands do not have the proof of physical origin and authenticity of the *W* identity, no matter how legitimate the command and control data are, it is rejected.

B. *Unclonable Device Authentication*

The standard three factors of authentication (3FA), *what you know, what you have, and what you are*, require users to interact with authenticating mechanism through awkward gestures (iris scan) and hard-to-remember passwords, complex behaviors (symbolic drawing) that distract and impede functional and operational efficiencies in accessing assets. With rising cybercrimes and cyberwarfare [14], many organizations are moving towards the use of two or more factors of authentication [15], hoping to increase the security and trustworthiness of the system/information access. A prime example of two-factor authentication is the use of ATM cards where bank customers are asked to provide something they have, their bank card, and something they know, their PIN number. Even this two-factor authentication was not enough to prevent one of the biggest bank "heist", leveraging the flaws and vulnerabilities of the ATM card use [16].

As biometrics technologies are becoming readily available and cheaper to deploy, "*what you are*" factor is becoming an increasingly popular factor to use in two-factor authentication deployments [17]. Some examples of the biometrics solutions include things like: finger or palm print; iris pattern; voice print; or even one's DNA. Using "*something you are*" is more convenient and has the appearance of greater security then "*what you have*" factors which can be replicated or copied. However, because biometrics are permanent and precise, once compromised, a person's identity can never be revoked and re-issued. The permanence of the biometrics is its weakest link in mass deployment of biometric-based authentication. Also, while biometrics may be more convenient than carrying a swipe card, it's still not convenient enough to make it usable for active authentication where identity is continuously requested and validated by authenticating entity.

Conventional cyber-identities are built around keeping static information (SSN, fingerprints, password, manufacturing serial numbers, etc.) secret. Unfortunately, this method requires exposing the secret when identities are confirmed. Even with encrypted protection, digitized data are inherently susceptible to cryptanalysis, cloning, phishing, and replay attacks. Over 75% of all cyber incidents are due to the compromise of cyber identities [18]. Examples of failed "lookup and compare" authentication are plenty and alarming, cutting across all sectors of businesses and government operations (Yahoo!, Target, LMC, IRS, OPM, etc.). In the case of CPSs, authentication options are even more restricted since we would have to eliminate authentication factors requiring human-in-the-loop options.

Digitally Unclonable Function (DUF) protocol is a system that relies on the possession factor of the "unclonable"

"unique" hardware as a means of verifying the authenticity and integrity of message (i.e., identity, commands). DUF protocol utilizes the unique device-level watermarking behavior to verify the source and truth of the statement [19].

C. *CPS and Finite Complexity*

This "unique, unclonable" hardware identity is used as underlying "trust anchor" where all CPS communications are explicitly checked against pre-defined *W* rules (i.e., who is it from and is it part of the approved message/command?). If the integrity and identity source of message cannot be verified, the received communication is dropped or logged as potential unauthorized penetration attempt. The details of DUF's protocol design and algorithm have been published and presented at a peer-reviewed IEEE conference [20].

In the case where tamper-resistant hardware is not available as CPS components, we propose enhancing the identity trust anchor with a software obfuscation technology called CodeLock [21]. CodeLock works by passing cryptographic key and software through a compiler which results in an obfuscated version of the software using well-established encryption algorithms, such as AES. The obfuscated code is functionally equivalent to the original but cannot be reverse engineered unless the block cipher used to encrypt the software is broken. The following block cipher algorithms can be used and have been demonstrated to work on the CodeLock design: AES-128, AES-256[22]. When the code needs to be executed, an embedded cryptographic key placed in a tamper-resistant computation environment is used as the trust anchor to de-obfuscate and run the code. The trust anchor uses only a small amount of memory and processing power to run the obfuscated code in a protected CPS environment.

III. *Example of CPS TDA Model: Unmanned Aerial System*

ICSs as well as Unmanned Aerial Systems (UASs) are just a subset of CPSs. The differences between UASs and ICSs are that UAS is much simpler, standalone system. ICSs are usually considered part of nation's critical infrastructure. Making and testing architectural modification on a scale of TDA on an operating ICSs could potentially result in unacceptable disruption to production and critical services. Aside from military application of UASs, the services rendered by UASs are generally considered non-critical to industrial manufacturing basis. What is common to both UASs and ICSs, however, are that they share the same technical security challenges: the integrity of communication, managing and protecting the system from cyber attackers that can physically impact safety and security of devices and people. UASs/ICSs can be targeted by attackers who want to steal the information resident in CPSs, sabotage them or commandeer them. As next phase of research, Sandia National Laboratories have worked with Purdue University's Flight Dynamics and Control / Hybrid System (FDC-HS) laboratory to explore a potential test case for the TDA simulation model as applied to UASs.

A. *Differences between UAS (i.e., ICS or CPSs) and Traditional Enterprise Networks*

ICSs differ significantly from traditional enterprise networks in that reliability, timing responses, deterministic behavior, and security and integrity of messaging are much more constrained. Compromise of any one of these functional requirements can have severe consequences. Conventional defenses are largely composed of firewalls, virus checking, intrusion detection and intrusion prevention systems (IDS/IPS), and patch updates. These solutions are ineffective against codes/hardware that have been already integrated into the deployed systems. Furthermore, many of the computing devices used are composed of open source operating systems with potential zero-day-exploit (ZDE) and unchecked vulnerabilities. Running IDS/IPS requires having regularly updated signatures, which places high processing demands on the UAS/ICS platform with limited computing processing power. Applying new signature updates and a patch update introduces an unacceptable disruption to mission operation, yielding an unrealistic deployment strategy. Furthermore, conventional defenses are grossly outmatched by the level of complex attack pathways that are possible—vulnerabilities inherent in standard operating systems and software codes. With millions of lines of codes, there are millions of different paths to compromise.

In addition to the conventional computer security issues that focus on trustworthiness of data flow, the onboard automation of a UAS is closely related to complex physical dynamics, interacting with different environments through sensors and actuators. Their tight integration between cyber and physical resources is the complex nature of many types of CPS (e.g., UAS, autonomous cars, smart power grid, and SCADA) [23]. Although computer security components are key elements in the hardware/software layer, these methods alone are insufficient to assure the health and security of UASs. For example, if the attacker has application layer control at the source node in the UAS, link layer encryption does not protect the UAS. It is also possible for multiple sensor attacks to corrupt the state of a UAS without breaking encryption. To address these conditions, a significant architectural opportunity exists in being able to design and test TDA so that security issues can be resolved within the context of a unified cyber (computing resources) and physical process model for the UAS, as shown in Figure 1.

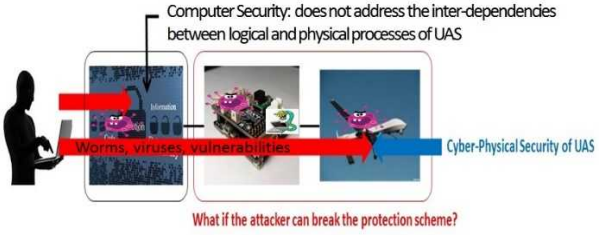

Figure 1. Cyber-Physical Security Problem for UAS

What technology can be used to reduce this problem of complexity, arising from millions of lines of codes and unchecked interaction with physical dynamics? Instead of trying to solve this "needle-in-a-haystack" (complexity) problem with an even more complex solution through

IDS/IPS, what if we design the UAS with the DAPE or *W* model? This design would simplify the system functions to include only what is intended, thus eliminating unintended actions. What if we can build a trusted system out of disparate untrusted components?

B. *Hybrid System Framework*

The cybersecurity problems can be categorized into two main types: (1) complexity problem from the firmware and software source, and (2) integrated components from untrusted supply chains [24]. It is important to note that these two problem types need to be addressed within an integrated cyber-physical process model of a UAS, for which we propose a hybrid system framework that can accurately represent the UAS as a CPS. The hybrid system is a dynamic system with interacting continuous dynamics, which represent the physical behavior of the UAS, and with discrete dynamics, which describe the cyber (logical) behavior of the UAS. This framework allows for a holistic view of UAS security, successfully complementing existing computer-oriented research.

The Purdue's FDC-HS laboratory led by Professor Inseok Hwang has devised a simulation laboratory designed to accurately model variety of CPSs by (i) accounting for both their physical and logical behaviors as well as studying (ii) cybersecurity capabilities, limitations, and potential threats. This hybrid system framework uses "reachability analysis," a solid mathematical foundation that simplifies or automates threat/vulnerability discovery during CPS design, production, testing, and operations. The main idea of the reachability analysis is to compute a set of states for a CPS that can be reached from the given initial states with inputs, attacks, uncertainties, etc. and examine potential security / safety violations based on the computed reachable set. This reachable set represents the operational envelope of a CPS [25]. Analysis of the reachable set is performed to assess the safety and operational capability of the CPS. To address the computational complexity of reachable set computation, the FDC-HS laboratory has developed an algorithm using a linear matrix inequality approximation solution for polyhedral invariant hybrid automaton, and the Markov chain approximation for the hybrid system. The resulting reachability information augments the knowledge on the cyber-physical process, eventually advancing the detection and analysis of untrusted components/sources in a CPS.

C. *Solving Complexity and Untrusted Component Issues for UAS*

The first type of problem (complexity) can be addressed in a straightforward manner - by limiting the number of functional capabilities (e.g., UASs and autonomous cars). For example, even though UAS components may run on Linux kernel with millions of lines of codes, if we know that a component's job is only to receive and respond to "on-off" commands, we can easily utilize DAPE rule to one rule (i.e., on/off command), not millions of possibilities inherent in Linux kernel. We can design the component to reject all other possible commands that can be given to the Linux kernel and only accept "on-off" commands and nothing else. Thus, we need a technology that can provide assurance that commands are coming from preconfigured, pre-approved *W* source(s) and that the command for "on" is really "on" and that it has not been modified to "off" during information transit. This aspect can be augmented by exploring the underlying physical behavior of the UAS. For example, even if the "on" command signal is legitimate, if the identity of the message originator is unknown, this message is dropped.

The second type of problem is building trust out of untrusted components. In order to accept the integrity and confidence of the information source, we must be able to devise a technique to create and distribute identities independent of the untrusted supply chain sources. Furthermore, these dynamically generated identities cannot be cloned or replicated once the identity registration process is complete. This ability to inject a "field generated, unclonable" ID into an untrusted component would deliver secure identity assurance in a platform composed of multiple components from untrusted sources.

1). *UAS Trust Anchor*

In a UAS environment, if an obfuscated version of software is to be run in a control processor/ field programmable gate array (FPGA), then the processor would have to make a call to the obfuscated code to execute. When the obfuscated code is called to execute, it would communicate with the trust anchor to run the obfuscated code. The trust anchor (Figure 2) can be built on a compartmentalized FPGA in place of tamper-resistant hardware, possibly residing on the same hardware as the obfuscated code itself. Without the use of tamper-resistant hardware, DUF's ability to guarantee the unclonability of hardware identity may be lowered. However, a CodeLock-enhanced DUF solution does provide significant protection against remote attackers since attackers would need physical access to the UASs in flight to clone the DUF/CodeLock processor.

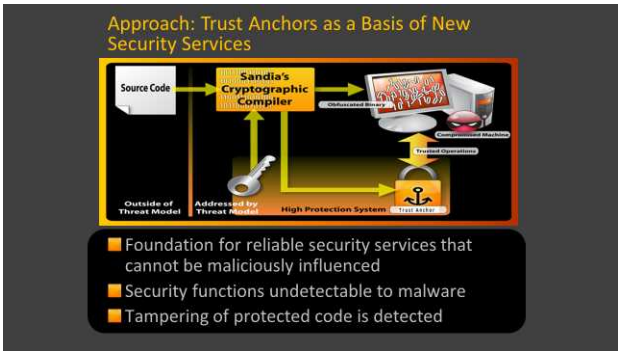

Figure 2. Trust Anchor

The DUF algorithm and its processing pattern is secured/obfuscated by using CodeLock. The DUF's dynamically changing identity would then be secured so that adversaries would have to reverse engineer the encryption algorithm used within CodeLock (AES-128 in our current implementation). A rigorous mathematical proof to prove this claim has been published on this topic [20].

We can further enhance the DAPE model in an online manner based on UAS physical dynamics such that DAPE becomes more rigorous or relaxed according to the physical

feasibility of the UAS. The "permit-by-exception" portion of DAPE can be updated based on the current state of the UAS involved with its physical environment (e.g., physically allowable transmission time, measurement set).

A design for the DUF/DAPE-enabled TDA for UAS would require an appropriate change in the overall system process to be built on the DAPE model. Specifically, we need to develop the relevant high-level operation logic, and the low-level control and estimation algorithms designed to rely on (or place more confidence in) the components permitted by DAPE. For example, we can selectively design an optimal controller for the commands filtered by DAPE. Otherwise, if any untrusted data are delivered without passing through DAPE, we would employ a robust controller to assure the UAS against potential threat vectors at the cost of some performance loss. The main difficulty lies in allowing for reasonable and comparable UAS performance to achieve the given mission, while the overall system process is permitted to use only a limited set of (trusted) components. This problem represents a trade-off between security and performance, especially for a high consequence system that is prone to excessive conservatism.

2). *Security Monitoring System and High Assurance Controller Design*

Research at Purdue University on securing a UAS addresses the development of a security monitoring system and a high assurance controller design, both of which are compatible with DAPE-enabled UAS operations. First, considering the untrusted components by DAPE, the developed monitoring system can assess the safety of the UAS in cases where there are security breaches regardless of detection. The main idea is based on the reachability analysis whereby the safety property is determined in an online manner without the need for specification details about the untrusted resources, thus significantly reducing the complexity of the DAPE model. In addition, the hybrid switching controller has the ability to adapt the control logic in response to a varying DAPE process. The hybrid controller consists of multiple sub-controllers, each of which can be designed for a different DAPE model. The hybrid controller can switch among the sub-controllers to secure and optimize UAS performance with respect to a time-varying DAPE process depending on different operational circumstances.

Research into these areas will focus on improving and comparing system performance (i.e., tolerance and resilience) against the threat vectors inherent in compromised system components and complex operating systems. The study will compare resilience of the developed trustworthy architecture vs. conventional cybersecurity defenses. Two identical UASs with the same physical/kernel components will be examined after cyber threats: one UAS will have a built-in TDA, and the other will have a conventional cybersecurity architecture. Advanced persistent threats (such as "back-doors or logic bombs") or ZDEs can be built in to attack the two systems, and then the survivability of the two architectures can be compared. Detailed comparative demonstration and validation are discussed in Section III.D.

From a system point of view, the UAS cyber threats can be categorized into the following two attack classes, and Figure 3 represents the selective TDA plug-in design against them.

- **Application Logic Security:** Attacks that manipulate the sensor or the environment data, thereby providing false data to the control system. In this case, the control system behaves as programmed without any fault, but some or all of the inputs to the system are corrupted. Some examples of this type of attack include sensory data manipulation, system component state data manipulation, navigational data manipulation, and command and control data manipulation. Figure 3 shows the most likely type of vulnerabilities for each component of the UAS.

- **Control System Security:** Attacks that attempt to prevent the hardware/CPU from behaving as programmed. Some examples of this type of attack include a buffer overflow exploit through an input device, a forced system reset to load malicious code, or a hardware change in the system.

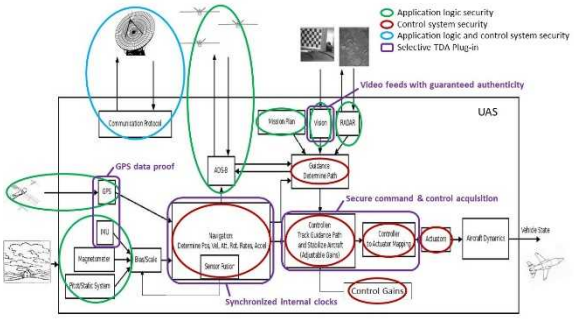

Figure 3. Cyber Attack Classes on UAS and Selective TDA Plug-in Design

D. *Potential Design Demonstrations*

The systems perspective does not need to look deeper into the methods of attacks; rather it focuses on studying the resulting damage/physical event to the UAS due to the malfunctioning components under the assumption that possibilities of corruption in the data flow exist within the UAS. The integration of our TDA architecture, shown in Figure 3, can counter such security vulnerabilities, making distinctive differences under some representative attack scenarios between a UAS with TDA built in and a UAS with conventional cyber security design. This approach provides for a broader impact of cyber threats, allowing for comprehensive and quantitative evaluation of UAS vulnerability, followed by testing of a DUF/DAPE-enabled TDA for a UAS.

1). *UAS TDA Use-Cases*

Following are potential UAS White-listed Use-Cases:

1. A system in which GPS has a built-in Trojan horse to purposely miss feeding GPS coordinates to the central brain of the drone or the command center. The TDA (physical) rules can be built in several different places, e.g.,
    a. Rules can be built into the GPS receiver itself. If the coordinates are generated by a compromised GPS processor, there should be differences of race conditions between the generated coordinates and the coordinates received from satellite. The central drone brain can then distinguish the source of the sensor data and reject it.
    b. Rules can be used to compare the time history of GPS coordinates and the drone's built-in Inertial Measurement Units (IMUs) to validate the accuracy of sensor data.

2. Demonstration of validation assurance and integrity of a messaging source. "Well synchronized" internal clocks will be vital. The drone components that use "accurate time" to validate assurance and integrity of a messaging source will require "authenticity" of time measurements from the "drone clock" to rest of the drone's electronic control units. Any time value "unsigned" by the drone's internal clock would be thrown out.

3. Ensuring a system that receives the command to "drop the bomb" responds only to commands from the commander on the ground, i.e., we want to prevent a compromised drone from dropping bombs according to pre-programmed Trojan horses. With TDA, we can build a secure command and control system that will be able to distinguish differences between "human-generated" vs. "machine-generated" commands.

4. Video feeds with guaranteed authenticity, which would prevent actions based on pre-recorded feeds rather than images that are directly and actively fed from the drone.

2). *Testing and Validation*

Once appropriate scenarios of UAS mission execution are selected, the use-cases need to be tested and validated for their effectiveness through extensive simulations using Software-In-The-Loop (SITL) and/or Hardware-In-The-Loop (HITL) simulation platforms. Depending on the funding source, this work will be developed by Purdue's FDC-HS laboratory as shown in Figure 4. The SITL and HITL testbeds can be used to implement various threat vectors in order to investigate their effects. The simulation platforms are based on an open source architecture, which comprises ArduPilot and PX4 autopilots, JSBSim/jMAVSim for dynamics, Scicoslab for simulation, and FlightGear for visualization. The major benefits of using the open source systems are reduced cost and the ability to modify each component for this research without proprietary issues, which could occur if off-the-shelf products were used. To enhance the fidelity of the simulation result, we are going to build highly modular HITL simulation environments in which each component can be replaced by real physical components for actual flight demonstration.

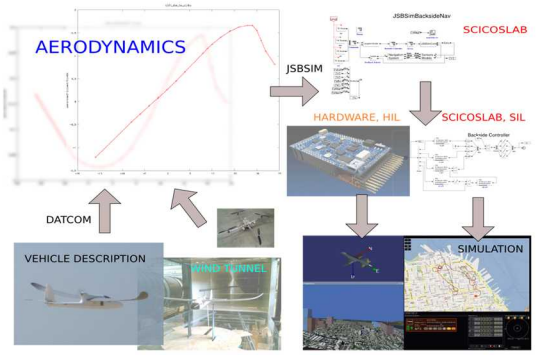

Figure 4. Software-In-The-Loop (SITL) and Hardware-In-The-Loop (HITL) Simulation Platforms

IV. CONCLUSION

In summary, TDA represents a significant departure from current cybersecurity defense postures where the quality of assurance/security is only as good as its latest updated profile of "known bad" things. Every new version of operating system or new software brings with it an undiscovered vulnerabilities and risks that must be monitored and mitigated.

With increasing complexity and diversification of software, cybersecurity built on *B* technique is reaching its limits. TDA model does not rely on building ever growing library of known or anticipated vulnerabilities to secure the system functions; instead TDA is designed to execute on CPS commands exclusively originated from the unclonable digital ID (i.e., the trust anchor). *B* systems, such as IDS or anti-virus applications, assumes that they have the latest and greatest library of data on vulnerabilities and threats. These *B* systems rely heavily on artisanal cybersecurity knowledge or sophisticated heuristic behavior comparisons with no possible end in sight.

TDA for UAS binds the impossible complexity problem that cybersecurity industry has been struggling with by utilizing security anchored in *W* physical CPS behaviors. TDA represents a radical, yet promising, change in system engineering concept that a self-sustaining secure system can be built using explicit *W* technique.

TDA may offer the best chances of overcoming the tsunami of ever increasing cyber threats, vulnerabilities, and costly patch management. For safety-critical CPSs such as ICSs, any delays, down-time or disruption to services have high consequence cost. It is important that cybersecurity

solutions implemented do not hinder or impede safety and security of CPSs.

REFERENCES

[1] T. Campbell, "Industrial Control Systems", Practical Information Security Management, pp. 205-211, 2016.

[2] Q. Wang and J. Jiang, "Comparative Examination on Architecture and Protocol of Industrial Wireless Sensor Network Standards", IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2197-2219, 2016.

[3] "Guidelines for Application Whitelisting in Industrial Control Systems", 2016. [Online]. Available: https://www.iad.gov/iad/library/ia-guidance/security-configuration/industrial-control-systems/guidelines-for-application-whitelisting-industrial-control-systems.cfm.

[4] "Lines of Code", Code.org, 2017. [Online]. Available: https://code.org/loc

[5] U. CERT., "Continuous Diagnostics and Mitigation", 2014. [Online]. Available: https://www.us-cert.gov/sites/default/files/cdm_files/SoftwareAssetManagementImplementation.pdf.

[6] C. Michael, K. van Wyk and W. Radosevich, "Black Box Security Testing Tools | US-CERT", Us-cert.gov, 2005. [Online]. Available: https://www.us-cert.gov/bsi/articles/tools/black-box-testing/black-box-security-testing-tools.

[7] S. Choi and D. Zage, "Addressing insider threat using “where you are” as fourth factor authentication", 2012 IEEE International Carnahan Conference on Security Technology (ICCST), 2012.

[8] M. Paulk, B. Curtis, M. Chrissis and C. Weber, "Capability maturity model, version 1.1", IEEE Software, vol. 10, no. 4, pp. 18-27, 1993.

[9] P. Meunier, "CERIAS - Center for Education and Research in Information Assurance and Security", Cerias.purdue.edu, 2006. [Online]. Available: https://www.cerias.purdue.edu/site/blog/post/what-is-secure-software-engineering/.

[10] A. Leemon, "Mitigate Cyber Threats in Industrial Control Systems with Application Whitelisting", CyberArk, 2016. [Online]. Available: https://www.cyberark.com/blog/mitigate-cyber-threats-in-industrial-control-systems-with-application-whitelisting/.

[11] S. Bacik, "Whitelisting", Infosectoday, 2017. [Online]. Available: http://www.infosectoday.com/Articles/whitelist.htm.

[12] "The IDART™ Methodology", Idart.sandia.gov, 2017. [Online]. Available: http://www.idart.sandia.gov/methodology/IDART.html.

[13] C. Cowan, "Securing Linux Systems with AppArmor", 2015.

[14] Symantec Security Response Team, "2013 internet security threat report," Symantec, Tech. Rep., 2013.

[15] Protecting your linkedin account with two-step verification. [Online]. Available: http://blog.linkedin.com/2013/05/31/protecting-your-linkedin-account-wht-two-step-verification/

[16] How cybercriminals stole $40 million from atms worldwide in just 10 hours. [Online]. Available: http://pctechmag.com/2013/05/how-cybercriminals-stole-40-million-from-atms-worldwide-in-just-10-hours/

[17] S. Choi and D. Zage, "Ephemeral Biometrics: What are they and what do they solve?", 2013 47th International Carnahan Conference on Security Technology (ICCST), 2013.

[18] "Data Breach Investigations Report", 2013. [Online]. Available: http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigations-report-2013_en_xg.pdf.

[19] Choi, S., David Zage, Yung Ryn Choe, and Brent Wasilow. 2015. "Physically Unclonable Digital ID," published in Proceedings of 2015 IEEE International Conference on Mobile Services (MS), 27 June-2 July 2015. SAND2015-3875C. http://ieeexplore.ieee.org/document/7226678/

[20] "Programmable Tamper Responses with Push-Button Protection", Microsemi.com. [Online]. Available: https://www.microsemi.com/document-portal/doc_view/134636-codeseal-programmable-tamper-responses-with-push-button-protection.

[21] Anderson, W. Erik. 2008. On the Secure Obfuscation of Deterministic Finite Automata. SAND2008-3520C. Albuquerque, NM: Sandia National Laboratories. http://ai2-s2-pdfs.s3.amazonaws.com/6bf9/d44610c36c85aa5d7f1159c910bc4291570f.pdf

[22] C. Sun, C. Liu and J. Xie, "Cyber-Physical System Security of a Power Grid: State-of-the-Art", Electronics, vol. 5, no. 3, p. 40, 2016.

[23] Y. Yang, T. Littler, S. Sezer, K. McLaughlin and H. Wang, "Impact of cyber-security issues on Smart Grid", 2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies, 2011.

[24] H. Yoo and T. Shon, "Challenges and research directions for heterogeneous cyber–physical system based on IEC 61850: Vulnerabilities, security requirements, and security architecture", Future Generation Computer Systems, vol. 61, pp. 128-136, 2016.

[25] P. Crosman, "Alert: There are too many cybersecurity alerts", American Banker, 2017. [Online]. Available: https://www.americanbanker.com/news/alert-there-are-too-many-cybersecurity-alerts.