Journal of Communications and Information Networks
Vol.1, No.4, Dec. 2016
DOI: 10.11959/j.issn.2096-1081.2016.041

Review paper

On the architecture and development life cycle of secure cyber-physical systems

SUN Cong, MA Jianfeng, YAO Qingsong

School of Cyber Engineering, Xidian University, Xi'an 710071, China

Abstract: Cyber-physical systems are being confronted with an ever-increasing number of security threats from the complicated interactions and fusions between cyberspace and physical space. Integrating security-related activities into the early phases of the development life cycle is a monolithic and cost-effective solution for the development of security-critical cyber-physical systems. These activities often incorporate security mechanisms from different realms. We present a fine-grained design flow paradigm for security-critical and software-intensive cyber-physical systems. We provide a comprehensive survey on the domain-specific architectures, countermeasure techniques and security standards involved in the development life cycle of security-critical cyber-physical systems, and adapt these elements to the newly designed flow paradigm. Finally, we provide prospectives and future directions for improving the usability and security level of this design flow paradigm.

Key words: cyber-physical system, development life cycle, security architecture, security standard, assessment, synthesis, verification

Citation: SUN C, MA J F, YAO Q S. On the architecture and development life cycle of secure cyber-physical systems[J]. Journal of communications and information networks, 2016, 1(4): 1-21.

1 Introduction

CPSs (Cyber-Physical Systems) have been referred to as the next generation of engineered systems. They tightly integrate computation, communication, and control mechanisms to achieve stability, reliability, robustness, and efficiency in dealing with physical processes of many different application domains[1]. CPSs have been impacting an increasing number of critical infrastructures and sectors of society, including energy, transportation, aerospace, design automation, robotics, medical devices, and next-generation manufacturing.

A CPS integrates a set of computing and communication components and relevant devices that deeply collaborate and interact with the physical world. Embedded computers and their networks usually monitor and control the physical processes with the help of sensors and actuators, and make decisions according to the feedbacks from the physical processes. As an engineered system, its functionality and salient characteristics emerge from the networked

Manuscript received Aug. 17, 2016; accepted Nov. 25, 2016
This work is supported by the National Natural Science Foundation of China (Nos. 61303033, 61303221), the National High Technology Research and Development Program of China (863 Program) (No. 2015AA017203), the Natural Science Basis Research Plan in Shaanxi Province of China (No. 2016JM6034), China 111 Project (No. B16037), and the Special Research Foundation of MIIT (No. MJ-2014-S-37).

interaction of computational and physical components. Compared to traditional embedded systems and real-time systems, the characteristics of CPSs put emphasis on (1) the information processing ability of each physical component, (2) layered and networked structure at large scale, (3) close-loop control and high degree of automation, (4) dynamic reconfigurability. As a result of these characteristics, the design of CPSs requires holistic understanding of hardware, software, networks, and physical processes.

As CPSs are widely integrated into various critical infrastructures, any security breach of these systems may result in disastrous consequences. Cyber-attacks can result in more serious consequences to the physical world than just a digital loss. This situation makes security solutions vital to CPSs. Merging networks and computation with physical processes usually brings significant challenges to the security enforcement of CPSs. In addition to the ordinary cyberspace vulnerabilities, the observation and manipulation on physical processes will also break the confidentiality of cyberspace data and communications, the integrity of physical feedbacks and control commands, and the availability of critical system services.

From a designer's perspective, the lack of theoretical foundations on secure system modeling and heterogeneous components composition makes it hard to incorporate security mechanisms, possibly from different realms, into CPSs. Moreover, security preserving integrations, verification and synthesis on non-functional requirements, and security analysis and assessment become more difficult due to the vastly increasing system size and behavior complexity. Embedding security activities into the development life cycle of CPSs is in great urgency to envisage the above challenges.

To elucidate the deficiencies associated with the development of secure CPSs and the future direction on the development life cycle of such systems, we present a comprehensive survey on the recent developments about the security considerations for the design flow and development life cycle of CPSs, with emphasis on the domain-specific security architectures and the security standards adaptable to this scenario. In contrast to recent literature reviews on domain specific threat identification and countermeasures for CPSs at different scales[2-4], we focus on providing system developers with valuable choices on the architectures and countermeasures useful in the development life cycle of security-critical CPSs. This is more comprehensive than a sketchy description on the relation between the architectural design and the development life cycle[5]. The contributions of this paper are as follows:

1) We propose a fine-grained design flow paradigm for the development of security-critical and software-intensive CPSs.

2) We propose four anticipated prerequisites for the security architectures. The more of these prerequisites that are satisfied, the more practical the security architecture is for practical use. In accordance with the results of a comprehensive check on the satisfaction of these prerequisites, we also bridge the gap between the domain-specific security architectures and the activities in the development life cycle involved in using these architectures.

3) To make the security requirement specification and security assessment more rigorous in the development life cycle of CPSs, we provide a summary of the security standards and guidelines w.r.t these development activities in different CPS sub-domains.

4) We provide a vision on the challenges and future directions of the architectural issue and the development activities of security-critical CPSs.

The rest of this paper is organized as follows. We propose a new paradigm on the design flow of software-intensive and security-critical CPSs in Section 2. In Section 3, we present the prerequisites

on the choice of architectures for the security critical CPSs, as well as the recent development on such security architectures in different CPS sub-domains. In Section 4, we summarize the typical security related activities and techniques for the development of secure cyber physical systems. We present the security standards in several sub-domains in Section 5 and the prospective research challenges and future directions in Section 6. We conclude the paper in Section 7.

2 Design flow for secure CPS development

In this section, we propose a new paradigm for the design flow of software-intensive CPSs (see the black-arrow directed flow in Fig.1). In this paradigm, the concepts show the generic guideline for each step of the development life cycle of CPSs. The advantage of this paradigm is to easily integrate most of the security-related activities and techniques, which is out of the scope of existing design flow paradigms.

The specifications of CPSs usually consist of control theoretic requirements and computational requirements. We merge the physical and environmental models into the functional design to consider the system controls at an early stage of development according to the specification of the system requirements. In the functional design of this V-shaped design flow, we also follow a similar principle as Lee[6,7] to cyberize the physical continuous dynamics to cyber-side discrete event models. Both the environment of CPSs and the physical process to be controlled should be modeled with mathematical equations to derive a control design, that can be expressed, and refined in the following as a control algorithm. In the architecture design phase, the control algorithm should be adapted into a platform-dependent architecture, and the related constraints for the semantics of computational models should be established. Different computational models can be used in this phase, including general models[8-11] and domain-specific models[12,13]. Under several computational models, we can further take a decomposition/refinement procedure to derive fine-grained collaborative components. In the code generation phase, automatic tools may be used to generate or synthesize source code for each individual software component. Both the functional and non-functional properties are verified along with unit testing. These properties are also derived from the requirement specifications and constraints. The integration of components put different functionalities onto the architecture with timing and resource configurations for the hardware, operating system, and COTS (Commercial Off-The-Shelf) components. The system deployment derives an executable system that can be validated, and runtime monitoring can be performed against the system requirements. Integration is usually performed before deployment when the characteristics and behaviors of components are well-known by the integrator. This order is taken by our design flow and some well-known domains, e.g. avionics. Conversely, the integration of medical CPSs is often performed at runtime after deployment with a somewhat loose binding[14].

To facilitate the development of secure CPSs, we extend this design flow with several concepts of security-related activities and techniques, as denoted by the gray elements in Fig.1. First, the security requirement specification mainly focuses on identifying new types of security vulnerabilities in CPSs and establishing the security functions required to mitigate or resist these vulnerabilities. Then, security mechanisms should be designed to satisfy the security functions in a holistic manner within the functional design. Detailed modeling and assessment on the security threats and attackers may help to elucidate these security requirements and designs. The primary work of architectural mitigation is to choose a suitable and platform-compatible

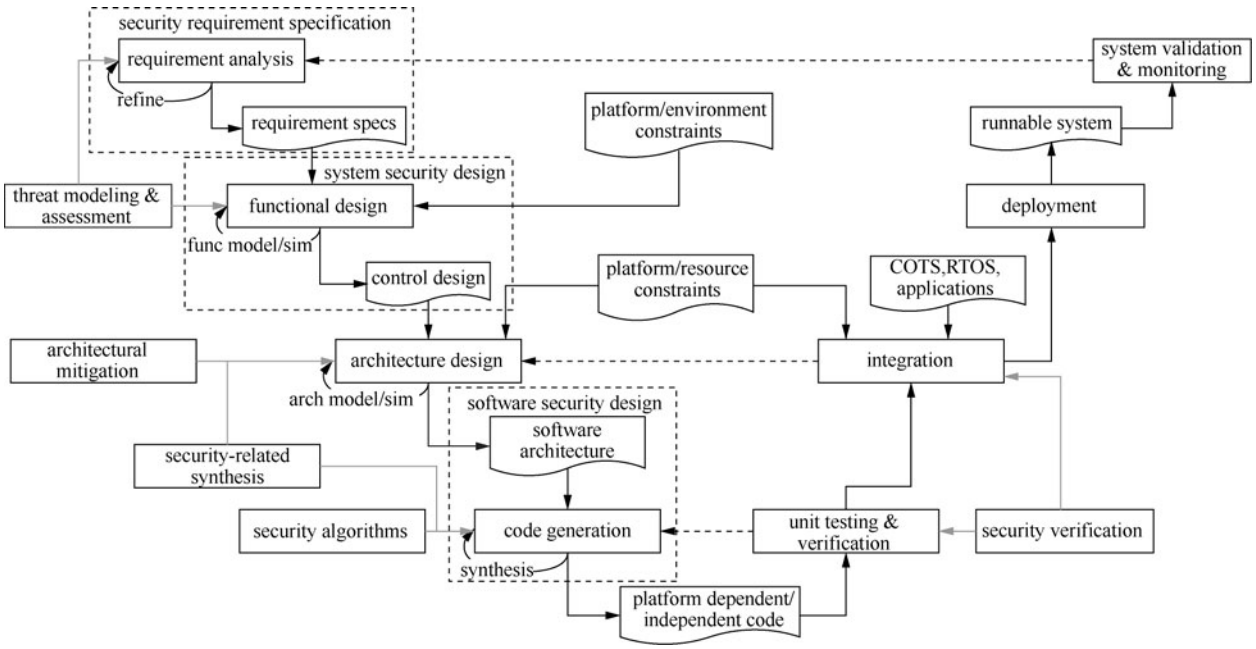


Figure 1 Overall security enhanced design flow of cyber-physical systems

security architecture for cyber-physical system design, possibly from those introduced in Section 3. After choosing a security architecture, there may be iterations on the security design of software, as well as the migration and application of various security algorithms, especially the off-the-shelf cryptographic libraries, to serve the implementation of security mechanisms. On both architectural level and code level, formal verification and synthesis may be applied to ensure the security policies are complied with by the models and programs.

3 Domain-specific security architectures for CPSs

In the context of CPSs, the cyber component (computation and communication) and the physical component (sensor and actuator) are tightly integrated. The vast variety on the regional scale of physical facilitates makes it hard to perform effective monitoring. Meanwhile, the connectivity between cyberspace and the physical world brings in new vulnerabilities to cyberspace infrastructures. The vulnerabilities of CPSs can be classified into three aspects: physical-side, cyber-side, and those caused by cyber-physical integration. A proposed secure architecture for CPSs is presented in Fig.2. In this abstract architecture, we firstly require that the physical processes and the environment are secured. Then, the possibly distributed topology of controllers forces us to ensure the security of cyber networks (i.e. Secure Network 2 in Fig.2). Finally, the connections between the sensor/actuator and cyberspace should resist not only traditional cyber-attacks but also the attacks on cyber-physical integration, e.g., the interception and tampering of the communication between physical-side source and sink nodes. That

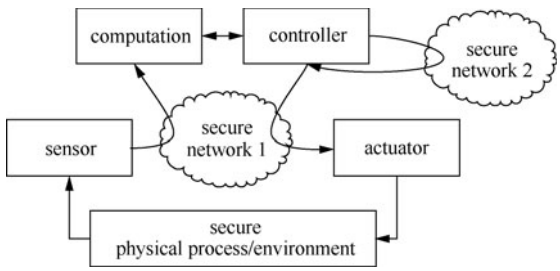


Figure 2 Proposed abstract architecture for secure CPSs

means in Fig.2, the cyber-physical boundary network, i.e., Secure Network 1, should meet different security requirements from the cyber networks, i.e., Secure Network 2.

The mitigation mechanisms of the platform-dependent architecture play a central role in the development of secure CPSs. In our opinion, the chosen security architectures for security-critical CPSs should make great efforts to satisfy the following prerequisites:

1) Control design adaptability. The architecture can adapt the control algorithm to meet the constraints of the platform and its resources.

2) Platform supportable and implementable. The architecture should be supported by general or realized platforms with both hardware and software. Irrespective of whether model-based refinement is allowed, the derived software architecture and components should be implementable, and the tool supports are preferred.

3) Security functionality distributable. In a networked or distributed environment, the security mechanisms can be distributed to different peers or layers as needed to achieve the security objectives.

4) Security should be built as part of the CPS architecture and not as a separate or stand-alone security service provider.

In general, the security architectures may be developed within CPSs horizontally or vertically. The middlewares may be deployed as a horizontal mechanism to play as the interface between physical plants and communication infrastructure for a robust and secure architecture of CPS communications[15]. Context-aware security architecture can be horizontally integrated into the computation component of CPSs[16]. A security engine reasons and enforces policies according to the security contexts from different sources, e.g., the physical environment, the boundary firewall, and the cyberspace system state. Layered architectures[17-19] have considered both dimensions. For the vertical aspect, different attacks and security mitigation strategies can reside at different layers. The functional modules of concrete systems, as well as the security functions to be protected are mapped to different layers. For example, the intrusion detection on the control layer can provide a cross-layer protection for the physical and communication layers, while the cyberspace anti-jamming and anti-eavesdropping mechanisms are predominantly on the communication layer. For the horizontal aspect, decentralized nodes respectively obey the layering, and interact through the network, communication and physical layers.

The design of CPSs can be considered as a cyber/physical co-design progress to trade off characteristics of the physical (e.g.,sampling rate) and computational plants (e.g., scheduling, frame period), and the cyber-side designs are substantially similar to the designs of embedded computing systems[5]. The modeling phase can be rigorous with support from domain-specific modeling languages and related tools. Consequently, the security architectures should be domain-specific as well. In the following section, we present the recent developments on the security architectures in several CPS sub-domains where the security requirements are more intensively addressed than some other sub-domains, e.g. medical CPSs.

3.1 Security architectures for networked infrastructures

Contemporary networks and critical infrastructures are examples of architectures whose physical components and their functionality, configuration, and operation should be protected against cyber-physical attacks. These kinds of architectures cannot be easily mapped to any architecture for the design flow, unless the requirements of CPSs can be decoupled and distributed to each physical component.

In power distribution automation networks or

smart grid, the secure operations are guaranteed by equipping conventional security facilities to provide highly secure key and trust management, authentication, end-to-end message delivery, strict code signatures and system integrity. These security mechanisms include firewall-based perimeter protection, PKI (Public Key Infrastructure), trusted computing technique, software patching, and intrusion detection[20,21]. Special emphasis is placed on secure bootstrapping to prevent rogue field devices from being installed without authorization[21]. Operating systems are usually treated as external circumstance or deployable off-the-shelf component of CPSs. For example, a real-time microkernel operating system has been integrated into a secure ubiquitous CPS architecture with support on inter-process access control, proxy-based inter-controller communication, proxy-separated legacy device support, and TPM-based device authorization[22]. In a multi-core hardware platform, the vertical isolation provided by a microkernel-based runtime environment can be extended to reduce the commonly dependent components to a DTU (Data Transfer Unit) and a stand-alone microkernel, both of which are more easily verified than a holistic system[23]. This architecture can separate trustworthy tasks (controllers) from untrustworthy tasks with considerations on the latency and safety. These security facilities will potentially change the topology and design architecture of the original infrastructures. In particular, when the system depends on a TPM-based platform, the location of the trusted modules and how these modules cooperate with the control processes need to be decided.

SDN (Software-Defined Networking) is a recent popular paradigm with clear separation between networking control and data. Bringing this paradigm into the CPS design can mitigate long-standing attacks, and improve grid control quality and the robustness of grid communication. To eliminate the additional challenges and vulnerabilities brought by SDN itself, further research on techniques, such as potential behavior predication on SDN control messages, data plane traffic monitoring to detect DoS attempts, and SDN rootkit detections is urgently required[24].

Cloud computing systems can sometimes be treated as CPSs. It has been realized that the infrastructure of cloud computing-based data centers may be under physical attacks on the application data and software[25]. Taking a similar principle as MTD (Moving Target Defense), we can establish an architecture to respond to sensor detected physical attacks by deleting, encrypting or migrating user code and data threatened by these attacks. The defense strategies are decided by the infrastructure information, the current status from sensors, the security requirements for VMs, and the scheduled events. A more pervasive manner in which to use clouds in CPSs is to enhance the control system with cloud computing ability, as exemplified by Google's self-driving car[26] and Unmanned Vehicles[27,28]. The security issues addressed in these kinds of architectures are mainly those associated with the confidentiality and integrity of data between clouds and control systems. For example, we should preserve a driver's private data when the vehicle's position is continuously tracked[29]or encrypt the sensor data before computation on control decision is outsourced to the cloud[27]. The security mechanisms should meet the stability and efficiency requirements of the control systems.

3.2 Avionics security architectures

IMA (Integrated Modular Avionics)[30] and MILS (Multiple Independent Levels of Security)[31] are both stand-alone architectures that facilitate the development of safety critical systems with explicit consideration of security. Verified or certified system components are preferred on both architectures for

safety and security reasons.

With the IMA architecture, we can build systems with hard real-time constraints. These systems are built with the support of (a) networked computing modules with different levels of criticality, (b) common interfaces for both hardware and software, and (c) principles of portability across an assemblet common hardware modules[14]. The principles of IMA emphasize 1) integration of functions from different parts of aircraft into a common computing and I/O platform, and 2) separation of computer system resources into isolated partitions. Several IMA systems have been developed by companies such as WindRiver and HoneyWell based on the ARINC-653 software architecture[32], which allows different applications to run at different levels of safety criticality without considering the different security clearances of data.

Compared to IMA, the MILS architecture is security-oriented, and supports compositional assurance and evaluation. It is designed to meet the Common Criteria requirement[33]. MILS can provide a dependable and secure architecture for high assurance applications, as well as a robust framework for separating untrustworthy code from security-critical code. The data and time partitioning provided by MILS can enable convenient component-based development of information assurance applications that satisfies rigorous certification criteria. Although the primary application area is avionics, MILS has also been adapted to automotive scenarios[34,35] and smart grids[36] recently. The Rockwell Collins AAMP7G microprocessor has been certificated by the U.S. National Security Agency as being in compliance with the MILS architecture[37]. PRISM[38] is another application of MILS that allows COTS programs running on separate single-level partitions to cooperate like a multi-level secure application whose behaviors comply with the Bell LaPadula policy.

The separation kernel is in principle the base computing platform of the MILS architecture. It separates a system into several independent virtual machines by dividing the memory into partitions and restricting the flow of information between those partitions. In a MILS environment, a separation kernel acts as a reference monitor, which is non-bypassable, evaluable, always invoked, and tamper proof[39]. SYSGO PikeOS (https://www.sysgo.com/products/pikeos-rtos-and-virtualization-concept/) is a separation kernel-based OS operating system whose hypervisor provides full separation for guest applications (guest OSs, runtime environments, APIs, etc.) on different criticality levels. PikeOS is totally compliant with both the MILS architecture and ARINC-653. Other examples of separation kernels include seL4[40] and XtratuM[41], which both can be used to build MILS architecture and even ARINC-653 compatible systems.

3.3 Automotive security architectures

Federated architecture was once widely adopted in the aeronautics and automotive industries. Although it has subsequently largely been replaced by IMA in avionics, it is still popular in vehicles. Axelsson, et al.[42] proposed a federated embedded system architecture that contains a centralized trusted server to provide security services and plug-in updates for the products. In the proposed system, the trusted server can also check the compatibility between the plug-in and the platform configurations[43]. In this architecture, the decentralized runtime environment for the plug-ins on each control unit has a rigorous interface for the plug-in to access the control unit.

In the established specification, e.g. AUTOSAR (http://www.autosar.org/standards), security is integrated by introducing cryptographic interfaces to ensure secure on-board communications. Integrating microkernel-based partitioning and inter-partition communication facilities of avionics helps to improve

the data-flow separation and the unidirectional control on confidentiality and integrity[44]. When the partitioning and physical isolation are sometimes unavailable for some automotive embedded systems, the safety-critical functions and non-critical management and information components can be transitively connected through gateways[45], whose design should concern the time-triggered characteristics for the data transformation and feather-weighted mechanisms for authentication and encryption. Seifert, et al.[46] also proposed a centralized security gateway to control the communication between ECUs and detect the failures in the on-board networks. Moreover, the AUTOSAR architecture can also hold a trusted server acting as a secure and manageable plug-in database to serve as a dynamic component model[43].

The automotive security should satisfy the resource constraints of hardware, real-time requirements of system functionality, network throughput limitations, and cost-effective requirements. The limitation on computation power and memory resources of ECUs, as well as the message length of bus protocols restrict us to add security mechanisms to the current automotive architecture. Even hardware assisted authentication can only provide partial protection for some automotive components[47]. It is inevitable to shift the remedial security enhancement to a holistic security-build-in design approach. This tendency happens in different sub-domains, e.g. Cybercar system and electric vehicles. The Cybercar system architecture integrates V2V (Vehicle-to-Vehicle), V2I (Vehicle-to-Infrastructure), and V2C (Vehicle-to-Cloud) communications together. In this architecture, the security challenges are closely related to the conditions of car networks and development progress[48]. The internal security of Cybercar is mainly focused on the CAN protocol related security under safety preservation. The security of V2V and V2C involves integrating security hardware accelerators, managing large numbers of keys with PKI, and protecting the owner's privacy. The security of infotainment platform focuses on isolating different kinds of accesses. For the electric vehicles, it is also necessary to secure the V2G (Vehicle-to-Grid) communication because the battery management system can communicate with charging stations through a power line communication protocol[13]. We may resort to Ethernet-based in-vehicle networks in combination with middleware and message filtering to achieve a high-bandwidth and scalable security solution.

It can be seen in the summarization in Tab.1 that, most of the security architectures used in developing domain-specific CPSs have security mechanisms that are tightly bound to the original CPS infrastructures. Only a few of these architectures resort to centralized or single-machine security service. For practical reasons, many security architectures depend on off-the-shelf security infrastructures, standard protocols and hardware. In addition, several representative instantiations have been demonstrated. Meanwhile, most of these architectures are not related to any specific control design. That means the architectures are insensitive to the control algorithms, and cannot effectively resist several control-specific malicious behaviors, e.g., injecting false data into public channels according to the control algorithm.

4 Security-enhanced SDLC (System Development Life Cycle) of CPSs

In different phases of the design flow, various security-related activities and techniques should be used to enhance the development of secure CPSs. Our survey addresses the following topics: (1) security requirement specification and security design methodologies, (2) threat modeling and vulnerability assessment helpful for identifying the security risks and attack consequences, and (3) security verification and synthesis in every possible design

Table 1 Architectural prerequisites meet by the security architectures of cyber-physical systems

security architecture	architectural prerequisites			
	control design adaptability	platform supportable and implementable	security functionality distributable	security build-in architecture
middleware[15]	N/A	√(supported by commercial communication infrastructures)	√(heterogeneous communication between subsystems of CPS gateways/routers)	√
context-aware security arch[16]	√(controller combined with situation predication)	N/A	√(horizontally integrated and located at any computation peer)	√
layered arch 1[17]	√(towards control system security)	N/A	√(distributed security measures at different layers)	√
layered arch 2[18]	√(control commands/operations on cyber network but not on cyber-physical boundary)	N/A	√(distributed gateways, conguration tools and security services)	√
layered arch 3[19]	√(multiple control loops share the platform)	√(CAN or CAN-FD as instance platform)	√(multiple control loops distributed)	√(security level based on estimation error of Kalman lter)
smart grid arch[20]	N/A	√(open PKI and TC implementations)	√(PKI components distributed)	√(trust computing facilitates built in)
secure ADN[21]	N/A	√(standard protocols, PKI and TPM implementations)	√(logical positioning of firewalls)	√
microkernel with proxy-based comm[22]	N/A	√(MINIX kernel and encryption on legacy BAS protocols)	√(different controllers distributed with their own proxy)	√(micro separation kernel based RTOS with TPM)
manycore with microkernel and split applications[23]	N/A	√(L4 microkernel and TPM on manycore	√(secure/public application OS on different VMMs communicating through VPN)	√
cloud-enabled networked controlsys[27]	√(model predictive control framework)	N/A	√(condentiality preserving computation decentralizes transparently on cloud)	√
IMA[30]	N/A	√(ARINC-653 based implementations by WindRiver & HoneyWell)	×(resource isolated locally and computation centralized)	×(no mechanism tightly integrated)
MILS[31]	N/A	√(AAMP7G, SYSGO PikeOS)	√(D-MILS consists of MILS nodes and deterministic network)	√(separation kernel as reference monitor)
federated arch[42, 43]	N/A	√(some AUTOSAR implementation)	×(centralized trusted server)	√
microkernel enhanced AUTOSAR[44]	N/A	√(AUTOBEST microkernel extending AUTOSAR & ARINC-653)	×(similar as IMA)	√(security-sensitive apps into dedicated time partitions)
secure automotive gateway[46]	N/A	√(in-vehicle network simulated on CANoe)	×(centralized gateway)	√
cybercar[48]	N/A	√(support platform-based design)	√(PKI and access control components distributed)	√

stage to ensure the constructed CPSs comply with security properties.

4.1 Security requirement specification and security design

The integrated nature of CPSs usually forces us to use a top-down approach to specify the security requirements and develop the security policies. Obviously the security requirements should be considered from the early stages of control system design. An important challenge is acquisition of sufficient understanding of the security requirements of the processes under control. Towards the analysis of security requirements, Fletcher, et al.[49] proposed a structural object-oriented security requirement analysis, specification, and prioritization process that uses high-order object models to represent CPS assets, and extends activity diagrams with mal-activities and prevention or mitigation options to identify threats posed by both internal and external misusers. Occasionally, security goals are firstly identified to derive the security requirements on the cyber-physical system design. We also have to deal with the conflict between the long lifespan of CPS and the high changeability of security requirements, as well as the vulnerabilities introduced by continuously increasing usage of open process of software design and commercial off-the-shelf components.

In fact, the choice of security design may significantly disrupt control performance, platform scheduling, human factors, etc. A global trade-off among these aspects is necessary in the requirement specification. Pasqualetti, et al.[50] argued that in resource-constrained CPS platforms, security mechanisms and control algorithms need to be co-designed and co-managed with the embedded platform, in order to tradeoff the conflicting control performance and security measures, and to avoid infeasibility when implementing these algorithms on embedded platforms. The design process proposed by Faruque, et al.[51] extends the functional model with aspect-oriented attack models[52]. By carrying out fuzzy attacks between the controller and the environment according to the attack models, we can simulate a disruption on the physical side of CPSs, and the simulation results can be further used to guide the countermeasure choice for the refinement of the design. A deep understanding of the human factors and usage contexts can also help to identify the security properties and guide the design of secure CPSs[53]. Value-sensitive design methods, such as stakehold analyses and value dams and flows, are used to derive domain-specific information from security.

Defining contracts for the interactions and dependencies of components benefits both the design and analysis of CPSs. For instance, design contracts are used to specify the timing and functionality properties for the interactions between control and software designers of CPSs[54]. A contract usually divides the responsibilities between a component and its environment into an assumption fulfilled by the environment and a guarantee that represents a property under the responsibility of the component. More formally, a contract is a pair $C=(A,G)$, where the assumption A and the guarantee G are properties satisfied by the set of all inputs and outputs of a design model. A design model M satisfies a contract $C = (A, G)$ whenever (1) M satisfies the guarantees G in the context of the assumptions A, and (2) M and C have the same set of variables. In the CPS design scenario, the assumptions are most likely on the elements of platform and physical processes, and the guarantees are inclined to the properties for software. Parallel composition and refinement on contracts are also applicable under this scenario. Sangiovanni-Vincentelli, et al.[55] suggested that the model should be rich to include descriptions on functions, performance and safety, and the design principle

should be followed along the entire design flow, especially in the platform dependent architecture design in Fig.1. To address the security aspect, Cimatti, et al.[56] integrated the distributed MILS architecture with contract-based reasoning to ensure global security requirement from locally guaranteed policies on components. The high integrity of MILS components and their connections ensures that verified components satisfy their contracts. Except for the design contracts, analysis contracts[57] are another kind of contracts used for security design. They are usually used to specify dependencies between analyses and for a verification algorithm to detect situations in which analysis produces unsound results or violates assumptions of other analyses. Ruchkin, et al.[58] used analysis contracts to describe the dependencies between sensor security, reliability and control. If the assumptions in these contracts are not satisfied, inter-domain vulnerabilities may exist.

4.2 Threat modeling and assessment

Threat and adversary modeling is the most important technique to make the security requirement specification and security design rigorous, and is of course a vital aspect of the control design and system design of security-critical CPSs. The requirements on security controls should be assigned over the model elements of assets, functions and environmental influences. Moreover, quantitative modeling approaches are preferred because the vulnerability assessments are primarily based on the results of such modeling.

Game theoretical models can usually capture the confliction between attackers and defending facilitates. The objective is to reach a tradeoff between perfect security and the usability of CPSs, and finally to make design proper defense countermeasures. A list of related approaches is given in Tab.2. Most of the listed approaches rely on discrete-time models and stochastic games. The game-theoretical framework

Table 2 Game-theoretical approaches for modeling attack/defense strategies and designing optimal solutions

Ref.	model	game approach	main results
[59]	continuous-time dynamics for the controlled physical process	zero-sum differential game	robust control design at physical layer (perfect robustness vs. performance)
	discrete-time Markov chain for the evolution of cyberstates	zero-sum stochastic game	design of defense mechanisms (perfect security vs. suability)
[60]	discrete-time CMDP (Competitive Markov Decision Process)	infinite-horizon stochastic game	design of response policy according to the optimal solution of game-theoretic maximin problem
[61]	reward functions and solution concepts for SCADA operator and intruder	semi network-form game	design on damage mitigation and intrusion response
[62, 63]	switching among subsystems (consisting of different controller, estimator, and detector) according to the system dynamics and detector information	zero-sum stochastic game (finite or moving horizon)	switching control policy that balances control performance/cost with security overhead (e.g. detection rate for different kinds of active attacks)
[64]	graph-based models	two-player stackelberg security game	design of IDS scheduling schemes to maximize 1)detection probability on attack under battery constraint, 2)the network/battery lifetime while ensuring attack always being detected
[65]	SMP (semi-Markov process)	zero-sum colonel Blotto game	optimal resource allocation between attacker and defender indicating relation between the probability of successful attack and the offensive/defensive resources

over hierarchical systems[59] combines discrete-time Markov chain with continuous-time dynamics, e.g. H^{∞} control, to improve the resilience of control systems under attacks. In this framework, there are two game models respectively on the cyber layer and the physical layer. The physical-layer model specifies the interactions between the physical system and disturbance, whereas the cyber-layer model captures the interactions between an attacker and a defender. EliMet[60] is a hybrid security assessment framework that formulates the response action selection procedure as a game-theoretic maximin problem. It also employs a discrete-time Markov decision process to describe the reciprocal interaction between the adversary and the power system operator or automated response engine. Both the operator and the automated engine are assumed to maximize their own benefit. We can specify the asymmetric interactions between cyber intruder and system operator over control systems with game-theoretic models, to estimate the outcome of the adversarial interaction and guide the design of control[61]. When the choice on subcomponents is available due to switching strategies, the game-theoretic solution on switching policy is discussed to achieve a balance between control cost and security overhead[62,63]. Several other specific game-theoretic models, such as Stackelberg game[64] and Colonel Blotto game[65], have been recently developed to optimize the scheduling of intrusion detection or resource allocation according to the balance between resource constraints and attack detection probability.

The attack models have been developed from the informal graphic structures to the formal models for security verification. In a specific SCADA framework, attack trees help to analyze the impact or consequence of cyber-physical attacks by identifying the adversary objectives and the vulnerabilities on different levels[66]. Attack trees are also useful for capturing the cyber-physical threat model with respect to the line contingencies of power systems[67]. Another application of attack trees is to analyze the risks of mobile transit information apps of urban railway[68]. CyberSAGE[69,70] raises a concept of security argument graph, which is automatically generated to integrate the attacker model with system information, workflows and security goals. The workflow-oriented security assessment produces quantitative results over the graphs. Bayesian attack graphs[71,72] of vulnerabilities and communication links are another kind of graphic model proposed to specify the procedures of successful cyber-attacks and to quantify the probabilities of attacks under different networking attack scenarios. Compared to the graphic models, more formal models can achieve more expressive notion on the actions of simultaneous attackers. Representative examples include the Byzantine faults model extended for cryptographic purposes[73] and Petri nets[74-77]. Chen, et al.[74] proposed to construct Petri net for modeling coordinated cyber physical attacks on large-scale infrastructures from domain-specific smaller Petri nets. SPNs (Stochastic Petri nets) are a type of timed Petri nets in which transitions fire after random-length time slot. Mitchell, et al.[75] used SPNs to analyze the tradeoffs between the design choice on detection/response strength of IDS and the energy consumption or attacker's ability. They also use SPNs to capture the dynamics between adversary behavior and defense for CPSs[76]. Colored Petri nets are useful for modeling information flows as well as building the threat models on smart meters for collaborative intrusion detection[77].

Threat modeling may also be tightly integrated with industrial processes on safety analysis and risk assessment. The SAHARA framework[78] extends inductive hazard analysis and risk assessment with the STRIDE threat model (https://msdn.microsoft.com/library/ms954176.aspx) to quantify the security impact on the safety goals. Some existing security and

safety co-analyses, such as FMVEA and CHASSIS, have been proved to be applicable on CPSs[68,79]. In these analyses, the safety failures and security threats are identified and treated in a holistic manner. Consequently, when conducting these co-analyses, security factors should meet the requirement of safety analyses, e.g. threat catalogues are required by FMVEA to identify threat and failure modes.

The vulnerability assessments are usually based on the results of threat modeling, and various analyses have been developed to assist these procedures. Wang, et al.[80] proposed a CPS simulator to model the behaviors of CPS components under different missions, and leverage symbolic execution to identify mission-critical components w.r.t value-assigned missions. Their approach tries to discover all the attack paths and assesses how vulnerable components collectively disrupt mission-critical components. The framework identifies a set of vulnerable components for the protection of mission-critical components to mitigate with the optimal cost. The impact of vulnerabilities on the reliability of CPSs has also been extensively studied over smart grid infrastructures or their subsystems[70,71,81,82]. The MTTC (Mean Time-To-Compromise) models can help to estimate the average frequency of cyber intrusions through various paths[71]. CPIndex[81]automatically performs cyber-physical contingency analysis with incomplete information for ranking the current status of system. Liu, et al.[82] assessed the pricing cyber attacks on AMI (Advanced Metering Infrastructure) and proposed to mitigate these attacks with support vector regression to predicate the pricing curve and maximum tolerable impact difference for detecting anomaly pricing.

4.3 Security synthesis and verification

Both formal synthesis and verification are promising approaches for enforcing specific properties over CPS models. The principle of synthesis is to automatically construct a system S whose composition with a given environment model E satisfies the property ϕ. S may be chosen from a class of system models. Formal verification is more straightforward. It takes both system model S and environment model E as input, and decides whether their composition, i.e. $S||E$ satisfies the property ϕ. The synthesis can be performed inductively with the help of formal verification. For instance, the abstraction function of the system is guided to be refined by the counterexamples derived by CEGAR-based model checking[83]. In the cyber-physical system scenario, these kinds of approaches have become more difficult because of the incompleteness of the environment modeling when the physical process is considered.

Formal synthesis has been applied to derive system-level simulation models and high-level architectural models that satisfy functional requirements[84] or hard real-time constraints[85]. To address security requirement and attack models, the security-related synthesis can be performed at either the code level or architectural level according to our design flow in Fig.1. Integrating code-level synthesis into the design framework of controllers can help to prevent injection of malicious code into the operation of the controller. The formal specification of execution and the code generation semantics are used by the synthesis procedure to avoid the interference or non-determinism to the code generation process[86]. Code-level synthesis is also applicable to derive a hardware monitor implemented in programmable fabric[87]. In both studies, the synthesized code or the derived code can be verified to ensure that some security properties are satisfied. At the architectural level, another formal framework[88] facilitates automatic synthesis of cost-effective countermeasures. The synthesis procedure iteratively decides whether some selected candidate architecture can resist stealthy attacks under given resource constraints and security requirements, and

terminates when one such architecture is found. The security architecture contains a set of buses or measurements to be secured.

Formal verification has been widely used to decide the correctness of system-environment models and the satisfaction of strictly defined properties. Although different reasoning and abstraction techniques have been applied for long to ensure correctness properties under various domains of CPSs[89-93], they have only recently been addressed to conduct security properties[94-98]. The related techniques, models and security properties are summarized in Tab.3. E# [94] is a programming language that integrates the specification of security-sensitive safety properties and limited forms of liveness properties into CPS modeling to facilitate theorem proving or model checking. The environment processes are modeled as clauses of E# and related event causality graphs. Rahman, et al.[95,96] utilized an SMT solver to verify whether the functionalities of smart grid, e.g. AMI configuration and optimal power flow, can satisfy security constraints, e.g., data freshness, integrity, confidentiality, or limited impact from attacks. The security constraints or properties can be specified as the negation of attack conditions or constraints over functionality under attacks[96, 97]. In an automotive scenario, the system architecture can be transformed to a continuous-time Markov model whose transition rates are determined by the security assessment of each component[98]. The probabilistic model checker can then decide a quantified security value for a defined security property. Without concretizing the formal models for CPSs, several guidelines are suggested to make formal verification more applicable to CPSs. It is suggested by a kind of design-for-verifiability approach[47] to implement critical functions of CPSs on a microkernel or a minimal component to reduce the complexity of formal modeling, and to use assume-guarantee reasoning over this microkernel-based architecture to achieve system-wise verification. Another framework combines verification and synthesis in the design cycle[99]. Security is enforced in this approach by synthesizing in-vehicle and V2V architecture design with timing and latency constraints.

5 Security in standards for CPSs

Standardization is a critical step towards widespread application of both security and safety techniques in industry. When security-oriented standards are enforced to raise the resilience and resistance of systems, the security requirements are usually made more specific and rigorous to benefit the decision of choosing countermeasures and mitigations. NIST SP 800-64[100] has attempted to integrate security

Table 3 Formal verifications of security properties of cyber-physical systems

Ref.	verification tech and tool	system model	property
[94]	model checking in inductive setting or theorem proving	E# code for CPSs (causes clauses & related event causality graphs for environment)	security-sensitive safety property & response-style liveness properties
[95]	constraint satisfaction checking with Z3 SMT solver	logic-based formal model of AMI behavior based on AMI device configurations	security constraints about authenticated communication, secure tunnel, priority delivery, fault-tolerance, and domain boundary protection
[96]	constraint satisfaction checking with Z3 SMT solver	formal model of OPF (Optimal Power Flow)	security constraints on fault data injection, cost limitation, expected attack impact
[97]	Simulink Design Verifier	closed-loop discrete time dynamical system modeled in Simulink	negation of the attack conditions
[98]	probabilistic model checking with PRISM	CTMC (Continuous-Time Markov Chain)	security properties in quantitative form, e.g exploitability rate

considerations into a five-stage development life cycle of general systems instead of CPSs. As a widely used standard for security evaluations, ISO/IEC15408 (Common Criteria)[33] consists of a catalogue of pre-defined security functional requirements that the evaluation target claims to fulfill. It is neither system-centric nor specific to the cyber-physical domain.

Meanwhile, in several CPS sub-domains, the security objectives, requirement specifications or assessment procedures have been standardized, especially in an integrated manner with safety issues. The safety standard IEC 61508[101] related to control systems requires treatment of security threats within hazard analysis in the form of security threats analysis, but no guidelines have been suggested. NIST SP 800-82[102] provides guidelines for establishing secure industrial control systems, including the threats and vulnerabilities of ICS, the boundary security for the network architecture, and the security controls.

In the transportation domain, security requirements have to be considered concomitant with the increasing use of wireless communication in ETCS (European Train Control System). IEC 62443[103] suggests to integrate security in the safety standards. The future draft of EN 50129[104] may include the security requirements from IEC 62443. Extending the safety development process of automotive safety standard ISO 26262[105] with security activities[106] shows a similar viewpoint as our design flow in Fig.1, but the extended process does not address the controlled physical process or the related threat issues of CPSs. Aviation authorities have focused intently on the security vulnerabilities that may affect the safety properties of aircraft. The ARINC 811 standard[107] defines different on-board domains of aircraft and their respective security clearances, i.e., closed, private and public. In fact, the primary security objectives for avionics are the assurances of availability and integrity. EUROCAE and RTCA have issued guidance for the definition of the airworthiness security process, including the specification[108] that addresses certification considerations during the early life cycle stages of aircraft, as well as its methods and considerations[109] that address the assessment of the acceptability of airworthiness security risk and the design and verification of airworthiness security attributes as related to system safety.

In the electric and smart grid domain, NISTIR 7628[110] provides detailed guidelines for the design of cyber-security mechanisms for smart grids. It presents methods for assessing risks in smart grids, and then identifies appropriate security requirements to mitigate these risks. Yoo, et al.[18] listed the security requirements that are required in a heterogeneous CPS environment based on IEC 61850[111], including the ones for software, security functions and policies. The IEC 62351 standard series[112] define end-to-end security mechanisms to protect communication protocols for substation systems, especially with the stronger encryption and authentication mechanisms[113]. Commercial level standard, such as the CIP (Critical Infrastructure Protection) standard[114] developed by NERC, aims at introducing compliance requirements to enforce baseline cyber-security efforts throughout power systems.

6 Challenges and future directions

Despite the advances made in security research of CPSs, there are still several prominent challenges preventing the effective development and wide application of security-critical CPSs.

The future directions on the architectural aspect include at least the following issues. First, self-healing architectures to monitor the security-related states of CPS and perform possible recovery at runtime need to be developed. This may also lead to combined consideration of the security and resilience of CPSs. The second issue is on the application side. We can try to find more applications of general

security-oriented architectures, e.g. MILS and TPM-based architectures, in different sub-domains of CPSs, e.g., medical devices. For the diverse types of processors used in CPSs, using the separation kernels and the associated information flow controls, data isolation, and QoS mechanisms to achieve high-assurance architectures is still an open problem. Third, the security architectures of CPSs should be dynamically reconfigurable to meet the connections of heterogeneous and COTS-enabled components. The dynamical evolution may in addition affect the security measures of systems.

On the design aspect, in order to ensure the robustness of CPSs, the security algorithms are sometimes treated as control optimizations, and are co-designed with the implementation platform under real-time and resource constraints. Therefore, we need online optimization algorithms to adapt the system parameters against attacks. Different artificial intelligence algorithms, such as simulated annealing or heuristic algorithms, are possible solutions to the optimization problems. Trust management between the components of CPSs is also a challenging issue. We need to propose system designs that can continuously assess and manage the trustworthiness of system components, and identify the compromises of components by cyber-attacks. As regards contract-based design and integration, a possible direction is to extend the analyses contracts towards more complicated contracts, such as those describing evolutionary attackers in highly dynamical environment. The contracts can be extended from deterministic to probabilistic, to describe the tendency and probability of uncertain events, e.g. from cryptography and fault tolerance. The corresponding verification needs to be adapted to these kinds of contracts, and more scalable contract verification tools need to be developed.

The future researches on threat modeling and assessment should first focus on the development of a knowledge base of known attack patterns on CPSs. Because empirical studies have shown that most attacks reuse mature attack vectors with minimal modification, this knowledge base can facilitate productive and precise analysis on the attack surface. Second, more comprehensive quantifications of the attack resources, vulnerabilities, resultant impacts, and security performance metrics are expected to assess the security level of CPSs in practical environments. Third and more specifically on game-theoretic models, we may develop different game-theoretic frameworks to consider more complicated intrusion and defensive strategies, that depend on both cyber and physical states. We may use the game theoretic methods and the models controlling the intrusion-defense dynamics to provide resiliency and survivability of CPSs against a wide variety of attacks. Fourth, the newly developed security metrics should be further integrated within the widely used security risk assessment procedures and standards.

For the verification aspect of security-related properties of CPSs, scalability and tool support are still open challenges. In order to formally reason about the perturbation from security properties on the control-centric properties, formal models at the interface of cyber and control layers need to be specified. However, this is still a difficult task.

7 Conclusion

Today, secure cyber-physical system development remains a field in its infancy in terms of research and industrial deployment. Although various security issues have been investigated in the context of its network or system aspect, the challenges imposed by the complexity of security architectures and countermeasures compel us to take a new look at the architectures and development life cycle for secure CPSs. We believe that a combinational study on the domain-specific security architectures and the security

techniques on different phases of development life cycle would enable us to reach a more reasonable and practical solution to the design and implementation of security-critical CPSs under a broad class of multidisciplinary constraints.

References

[1] KIM K, KUMAR P R. Cyber-physical systems: a perspective at the centennial[J]. Proceedings of the IEEE, 2012, 100: 1287-1308.

[2] RUSHANAN M, RUBIN A D, KUNE D F, et al. Sok: security and privacy in implantable medical devices and body area networks[C]// IEEE Symposium on Security and Privacy (S&P), 2014: 524-539.

[3] KONSTANTINOU C, MANIATAKOS M, SAQIB F, et al. Cyber-physical systems: a security perspective[C]//The 20th IEEE European Test Symposium, 2015: 1-8.

[4] STUDNIA I, NICOMETTE V, ALATA E, et al. Survey on security threats and protection mechanisms in embedded automotive networks[C]//The 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop, 2013: 1-12.

[5] WOLF M, FERON E. What don't we know about CPS architectures?[C]//The 52nd Annual Design Automation Conference (DAC), 2015: 80:1-80:4.

[6] LEE E A. CPS foundations[C]//The 47th Design Automation Conference (DAC), 2010: 737-742.

[7] JENSEN J C, CHANG D H, LEE E A. A model-based design methodology for cyber-physical systems[C]//The 7th International Wireless Communications and Mobile Computing Conference, 2011: 1666-1671.

[8] LIU Z, LIU J, HE J, et al. Spatiotemporal UML statechart for cyber-physical systems[C]//The 17th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS), 2012: 137-146.

[9] CANEDO A, FARUQUE M A A. Towards parallel execution of IEC 61131 industrial cyber-physical systems applications[C]// Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012: 554-557.

[10] YOONG L H, ROOP P S, SALCIC Z. Implementing constrained cyber-physical systems with IEC 61499[J]. ACM Trans. Embedded Comput. Syst., 2012, 11(4): 1-22.

[11] TAN Y, VURAN M C, GODDARD S. Spatio-temporal event model for cyber-physical systems[C]//The 29th IEEE International Conference on Distributed Computing Systems Workshops, 2009: 44-50.

[12] ALVES-FOSS J, TAYLOR C, OMAN P W. A multilayered approach to security in high assurance systems[C]//The 37th Hawaii International Conference on System Sciences, 2004: 90302b.

[13] LUKASIEWYCZ M, STEINHORST S, ANDALAM S, et al. System architecture and software design for electric vehicles[C]// The 50th Annual Design Automation Conference (DAC), 2013: 95:1-95:6.

[14] HATCLIFF J, KING A L, LEE I, et al. Rationale and architecture principles for medical application platforms[C]//IEEE/ACM 3rd International Conference on Cyber-Physical Systems (ICCPS), 2012: 3-12.

[15] SHIN D, HE S, ZHANG J. Robust, secure, and cost-effective design for cyber-physical systems[J]. IEEE intelligent systems, 2014, 29(1): 66-69.

[16] WAN K, ALAGAR V S. Context-aware security solutions for cyber-physical systems[J]. MONET, 2014, 19(2): 212-226.

[17] ZHU Q, RIEGER C, BAS_AR T. A hierarchical security architecture for cyber-physical systems[C]//The 4th International Symposium on Resilient Control Systems, IEEE, 2011: 15-20.

[18] YOO H, SHON T. Challenges and research directions for heterogeneous cyber-physical system based on IEC 61850: vulnerabilities, security requirements, and security architecture[J]. Future generation computer systems, 2016, 61:128 - 136.

[19] ZHENG B, DENG P, RAJASEKHAR A, et al. Cross-layer codesign for secure cyber-physical systems[J]. IEEE trans. on CAD of integrated circuits and systems, 2016, 35(5): 699-711.

[20] METKE A R, EKL R L. Security technology for smart grid networks[J]. IEEE trans. on smart grid, 2010, 1(1): 99-107.

[21] TESFAY T T, HUBAUX J, BOUDEC J L, et al. Cybersecure communication architecture for active power distribution networks[C]//Symposium on Applied Computing (SAC), 2014: 545-552.

[22] WANG X, MIZUNO M, NEILSEN M L, et al. Secure RTOS architecture for building automation[C]//The 1st ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy (CPS-SPC), 2015: 79-90.

[23] VOLP M, ASMUSSEN N, H ARTIG H, et al. Towards dependable CPS infrastructures: architectural and operating-system challenges[C]//The 20th IEEE Conference on Emerging Technologies & Factory Automation (ETFA), 2015: 1-8.

[24] DONG X, LIN H, TAN R, et al. Software-defined networking for smart grid resilience: opportunities and challenges[C]//The 1st ACM Workshop on Cyber-Physical System Security (CPSS), 2015: 61-68.

[25] SZEFER J, JAMKHEDKAR P A, CHEN Y, et al. Physical attack protection with human-secure virtualization in data centers[C]// IEEE/IFIP International Conference on Dependable Systems and Networks Workshops, 2012: 1-6.

[26] KEHOE B, PATIL S, ABBEEL P, et al. A survey of research on cloud robotics and automation[J]. IEEE trans. on automation science and engineering, 2015, 12(2): 398-409.

[27] XU Z, ZHU Q. Secure and resilient control design for cloud enabled networked control systems[C]//The 1st ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy (CPS-SPC), 2015: 31-42.

[28] PANDEY P, POMPILI D, YI J. Dynamic collaboration

between networked robots and clouds in resource constrained environments[J]. IEEE trans. on automation science and engineering, 2015, 12(2): 471-480.

[29] WAN J, ZHANG D, SUN Y, et al. VCMIA: a novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing[J]. MONET, 2014, 19(2): 153-160.

[30] CONMY P, NICHOLSON M, MCDERMID J A. Safety assurance contracts for integrated modular avionics[C]//The 8th Australian Workshop on Safety Critical Systems and Software, 2003: 69-78.

[31] ALVES-FOSS J, OMAN P W, TAYLOR C, et al. The MILS architecture for high-assurance embedded systems[J]. IJES, 2006, 2(3/4): 239-247.

[32] COOK A. ARINC 653-challenges of the present and future[J]. Microprocessors and microsystems – embedded hardware design, 1995, 19(10): 575-579.

[33] International Standardization Organization. ISO 15408: information technology-security techniques – evaluation criteria for IT security (common criteria)[S]. 2009, http://www.commoncriteriaportal.org.

[34] CAMEK A G, BUCKL C, KNOLL A. Future cars: necessity for an adaptive and distributed multiple independent levels of security architecture[C]//The 2nd ACM International Conference on High Confidence Networked Systems (HiCoNS), 2013: 17-24.

[35] PITCHFORD M. Applying MILS principles to design connected embedded devices supporting the cloud, multitenancy and App Stores[C]//The 8th European Congress on Embedded Real Time Software and Systems (ERTS), 2016.

[36] BYTSCHKOW D, QUILBEUF J, IGNA G, et al. Distributed MILS architectural approach for secure smart grids[C]//The 2nd International Workshop on Smart Grid Security (SmartGridSec), 2014: 16-29.

[37] HARDIN D S. Invited tutorial: considerations in the design and verification of microprocessors for safety-critical and security-critical applications[C]//Formal Methods in Computer-Aided Design (FMCAD), 2008: 1-8.

[38] OWEN C A, GROVE D A, NEWBY T, et al. PRISM: program replication and integration for seamless MILS[C]//The 32nd IEEE Symposium on Security and Privacy (S&P), 2011: 281-296.

[39] HEITMEYER C L, ARCHER M, LEONARD E I, et al. Formal specification and verification of data separation in a separation kernel for an embedded system[C]//The 13th ACM Conference on Computer and Communications Security (CCS), 2006: 346-355.

[40] KLEIN G, ELPHINSTONE K, HEISER G, et al. Sel4: formal verification of an OS kernel[C]//The 22nd ACM Symposium on Operating Systems Principles (SOSP), 2009: 207-220.

[41] CRESPO A, RIPOLL I, MASMANO M. Partitioned embedded architecture based on hypervisor: the xtratum approach[C]//The 8th European Dependable Computing Conference (EDCC), 2010: 67-72.

[42] AXELSSON J, Kobetski A. Architectural concepts for federated embedded systems[C]//European Conference on Software Architecture Workshops, 2014: 25:1-25:8.

[43] NI Z, KOBETSKI A, AXELSSON J. Design and implementation of a dynamic component model for federated AUTOSAR systems[C]//The 51st Annual Design Automation Conference (DAC), 2014: 94:1-94:6.

[44] ZUEPKE A, BOMMERT M, LOHMANN D. AUTOBEST: a united AUTOSAR-OS and ARINC 653 kernel[C]//The 21st IEEE Real-Time and Embedded Technology and Applications Symposium, 2015: 133-144.

[45] ABDALLAH A, FERON E M, HELLESTRAND G R, et al. Hardware/software codesign of aerospace and automotive systems[J]. Proceedings of the IEEE, 2010, 98(4): 584-602.

[46] SEIFERT S, OBERMAISSER R. Secure automotive gateway-secure communication for future cars[C]//The 12th IEEE International Conference on Industrial Informatics (INDIN), 2014: 213-220.

[47] SAGSTETTER F, LUKASIEWYCZ M, STEINHORST S, et al. Security challenges in automotive hardware/ software architecture design[C]//Design, Automation and Test in Europe (DATE), 2013: 458-463.

[48] KOUSHANFAR F, SADEGHI A, SEUDIE H. EDA for secure and dependable cybercars: challenges and opportunities[C]//The 49th Annual Design Automation Conference (DAC), 2012: 220-228.

[49] FLETCHER K K, LIU X F. Security requirements analysis, specification, prioritization and policy development in cyber-physical systems[C]//The 5th International Conference on Secure Software Integration and Reliability Improvement (SSIRI), 2011: 106-113.

[50] PASQUALETTI F, ZHU Q. Design and operation of secure cyber-physical systems[J]. Embedded systems letters, 2015, 7(1): 3-6.

[51] FARUQUE M A A, REGAZZONI F, PAJIC M. Design methodologies for securing cyber-physical systems[C]// International Conference on Hardware/ Software Codesign and System Synthesis (CODES+ISSS), 2015: 30-36.

[52] WAN J, CANEDO A, FARUQUE M A A. Security-aware functional modeling of cyber-physical systems[C]//The 20th IEEE Conference on Emerging Technologies & Factory Automation (ETFA), 2015: 1-4.

[53] DENNING T, KRAMER D B, FRIEDMAN B, et al. CPS: beyond usability: applying value sensitive design based methods to investigate domain characteristics for security for implantable cardiac devices[C]//The 30th Annual Computer Security Applications Conference (ACSAC), 2014: 426-435.

[54] DERLER P, LEE E A, TRIPAKIS S, et al. Cyber physical system design contracts[C]//ACM/IEEE 4th International Conference on Cyber-Physical Systems (ICCPS), 2013: 109-118.

[55] SANGIOVANNI-VINCENTELLI A L, DAMM W, PASSERONE R. Taming dr. frankenstein: contractbased design for cyber-physical systems[J]. Eur. j. control, 2012, 18(3): 217-238.

[56] CIMATTI A, DELONG R, MARCANTONIO D, et al. Combining MILS with contract-based design for safety and security requirements[C]//Computer Safety, Reliability, and Security

(SAFECOMP) Workshops, 2015: 264-276.

[57] RUCHKIN I, DE NIZ D, CHAKI S, et al. Contract-based integration of cyber-physical analyses[C]//International Conference on Embedded Software (EMSOFT), 2014: 23:1-23:10.

[58] RUCHKIN I, RAO A, de NIZ D, et al. Eliminating inter-domain vulnerabilities in cyber-physical systems: An analysis contracts approach[C]//The 1st ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy (CPS-SPC), 2015: 11-22.

[59] ZHU Q, BASAR T. Game-theoretic methods for robustness, security, and resilience of cyber physical control systems: Games-in-games principle for optimal cross-layer resilient control systems[J]. IEEE control systems, 2015, 35(1): 46-65.

[60] ZONOUZ S A, HAGHANI P. Cyber-physical security metric inference in smart grid critical infrastructures based on system administrators' responsive behavior[J]. Computers & security, 2013, 39: 190-200.

[61] BACKHAUS S, BENT R, BONO J W, et al. Cyberphysical security: a game theory model of humans interacting over control systems[J]. IEEE trans. on smart grid, 2013, 4(4): 2320-2327.

[62] MIAO F, PAJIC M, PAPPAS G J. Stochastic game approach for replay attack detection[C]//The 52nd IEEE Conference on Decision and Control (CDC), 2013: 1854-1859.

[63] MIAO F, ZHU Q. A moving-horizon hybrid stochastic game for secure control of cyber-physical systems[C]//The 53rd IEEE Conference on Decision and Control (CDC), 2014: 517-522.

[64] ABBAS W, LASZKA A, VOROBEYCHIK Y, et al. Scheduling intrusion detection systems in resource bounded cyber-physical systems[C]//The 1st ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy (CPS-SPC), 2015: 55-66.

[65] ZHANG Y, WANG L, XIANG Y, et al. Inclusion of scada cyber vulnerability in power system reliability assessment considering optimal resources allocation[J]. IEEE transactions on power dystems, 2016, 31(6): 4379-4394.

[66] TEN C, MANIMARAN G, LIU C. Cyber security for critical infrastructures: attack and defense modeling[J]. IEEE Trans. on systems, man, and cybernetics, Part A, 2010, 40(4): 853-865.

[67] DAVIS K R, DAVIS C M, ZONOUZ S A, et al. A cyber physical modeling and assessment framework for power grid infrastructures[J]. IEEE trans. on smart grid, 2015, 6(5): 2464-2475.

[68] CHEN B, SCHMITTNER C, MA Z, et al. Security analysis of urban railway systems: the need for a cyber-physical perspective[C]//Computer Safety, Reliability, and Security (SAFECOMP) Workshops, 2015: 277-290.

[69] VU A H, TIPPENHAUER N O, CHEN B, et al. CyberSAGE: a tool for automatic security assessment of cyber-physical systems[C]//The 11th International Conference on Quantitative Evaluation of Systems (QEST), 2014: 384-387.

[70] CHEN B, KALBARCZYK Z, NICOL D M, et al. Go with the flow: toward workow-oriented security assessment[C]//New Security Paradigms Workshop (NSPW), 2013: 65-76.

[71] ZHANG Y, WANG L, XIANG Y, et al. Power system reliability evaluation with SCADA cybersecurity considerations[J]. IEEE trans. on smart grid, 2015, 6(4): 1707-1721.

[72] ZHANG Y, XIANG Y, WANG L. Power system reliability assessment incorporating cyber attacks against wind farm energy management systems[J]. IEEE transactions on smart grid, 2016, PP(99): 1-15.

[73] BURMESTER M, MAGKOS E, CHRISSIKOPOULOS V. Modeling security in cyber-physical systems[J]. International journal of critical infrastructure protection, 2012, 5(3-4): 118 - 126.

[74] CHEN T M, SANCHEZ-AARNOUTSE J C, BUFORD J F. Petri net modeling of cyber-physical attacks on smart grid[J]. IEEE trans. on smart grid, 2011, 2(4): 741-749.

[75] MITCHELL R, CHEN I. Effect of intrusion detection and response on reliability of cyber physical systems[J]. IEEE trans. on reliability, 2013, 62(1): 199-210.

[76] MITCHELL R, CHEN I. Modeling and analysis of attacks and counter defense mechanisms for cyber physical systems[J]. IEEE trans. on reliability, 2016, 65(1): 350-358.

[77] LIU X, ZHU P, ZHANG Y, et al. A collaborative intrusion detection mechanism against false data injection attack in advanced metering infrastructure[J]. IEEE trans. on smart grid, 2015, 6(5): 2435-2443.

[78] MACHER G, SPORER H, BERLACH R, et al. SAHARA: a security-aware hazard and risk analysis method[C]//Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015: 621-624.

[79] SCHMITTNER C, MA Z, SCHOITSCH E, et al. A case study of FMVEA and CHASSIS as safety and security co-analysis method for automotive cyber-physical systems[C]//The 1st ACM Workshop on Cyber-Physical System Security (CPSS), 2015: 69-80.

[80] WANG X, DAVIS M, ZHANG J, et al. Missionaware vulnerability assessment for cyber-physical systems[C]//2015 IEEE TrustCom / BigDataSE/ISPA, 2015: 1148-1153.

[81] VELLAITHURAI C, SRIVASTAVA A K, ZONOUZ S A, et al. CPIndex: cyber-physical vulnerability assessment for power-grid infrastructures[J]. IEEE trans. on smart grid, 2015, 6(2): 566-575.

[82] LIU Y, HU S, HO T. Vulnerability assessment and defense technology for smart home cybersecurity considering pricing cyberattacks[C]//IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2014: 183-190.

[83] CLARKE E M, GRUMBERG O, JHA S, et al. Counterexample-guided abstraction re_nement[C]//The 12th International Conference on Computer Aided Verification (CAV), 2000: 154-169.

[84] CANEDO A, SCHWARZENBACH E, FARUQUE M A A. Context-sensitive synthesis of executable functional models of cyber-physical systems[C]//ACM/IEEE 4th International Conference on Cyber-Physical Systems (ICCPS), 2013: 99-108.

[85] HANG C, MANOLIOS P, PAPAVASILEIOU V. Synthesizing cyber-physical architectural models with realtime constraints[C]// The 23rd International Conference on Computer Aided Verification (CAV), 2011: 441-456.

[86] PAJIC M, BEZZO N, WEIMER J, et al. Towards synthesis of platform-aware attack-resilient control systems: extended abstract[C]//The 2nd ACM International Conference on High Confidence Networked Systems (HiCoNS), 2013: 75-76.

[87] LERNER L W, FRANKLIN Z R, BAUMANN W T, et al. Using high-level synthesis and formal analysis to predict and preempt attacks on industrial control systems[C]//ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA), 2014: 209-212.

[88] RAHMAN M A, AL-SHAER E, KAVASSERI R G. Security threat analytics and countermeasure synthesis for power system state estimation[C]//The 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2014: 156-167.

[89] THOMPSON S, BRAT G P, VENET A. Software model checking of ARINC-653 fight code with MCP[C]//The 2nd NASA Formal Methods Symposium (NFM), 2010: 171-181.

[90] DENMAN W, ZAKI M H, TAHAR S, et al. Towards fight control verification using automated theorem proving[C]//The 3rd NASA Formal Methods Symposium (NFM), 2011: 89-100.

[91] M ULLER C A, PAUL W J. Complete formal hardware verification of interfaces for a exray-like bus[C]//The 23rd International Conference on Computer Aided Verification (CAV), 2011: 633-648.

[92] WEISSMANN M, BEDENK S, BUCKL C, et al. Model checking industrial robot systems[C]//The 18th International Workshop on Model Checking Software (SPIN), 2011: 161-176.

[93] GARLAPATI S, SHUKLA S K. Formal verification of hierarchically distributed agent based protection scheme in smart grid[C]//The 19th International Workshop on Model Checking Software (SPIN), 2012: 137-154.

[94] POROOR J, JAYARAMAN B. Formal analysis of eventdriven cyber physical systems[C]//The 1st International Conference on Security of Internet of Things (SECURIT), 2012: 1-8.

[95] RAHMAN M A, AL-SHAER E, BERA P. A noninvasive threat analyzer for advanced metering infrastructure in smart grid[J]. IEEE trans. on smart grid, 2013, 4(1): 273-287.

[96] RAHMAN M A, AL-SHAER E, KAVASSERI R G. A formal model for verifying the impact of stealthy attacks on optimal power flow in power grids[C]//ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), 2014: 175-186.

[97] TRCKA N, MOULIN M, BOPARDIKAR S D, et al. A formal verification approach to revealing stealth attacks on networked control systems[C]//The 3rd International Conference on High Confidence Networked Systems (HiCoNS), 2014: 67-76.

[98] MUNDHENK P, STEINHORST S, LUKASIEWYCZ M, et al. Security analysis of automotive architectures using probabilistic model checking[C]//The 52nd Annual Design Automation Conference (DAC), 2015: 38:1-38:6.

[99] ZHENG B, LI W, DENG P, et al. Design and verification for transportation system security[C]//The 52nd Annual Design Automation Conference (DAC), 2015: 96:1-96:6.

[100] KISSEL R, STINE K, SCHOLL M, et al. NIST SP 800- 64, Revision 2: security considerations in the system development life sycle[S]. 2008.

[101] International Electrotechnical Commission. IEC 61508: functional safety of electrical/electronic/programmable electronic safety-related systems[S]. 2010.

[102] STOUFFER K, FALCO J, SCARFONE K. NIST SP 800-82: guide to industrial control systems (ICS) security[S]. National Institute of Standards and Technology, 2011.

[103] International Electrotechnical Commission. IEC 62443: industrial communication networks-network and system security-security for industrial automation and control systems[S]. 2009.

[104] European Committee for Standardization. EN 50129: railway applications-communication, signalling and processing systems - safety related electronic systems for signalling[S]. 2003.

[105] International Standardization Organization. ISO 26262: road vehicles-functional safety[S]. 2011.

[106] BURTON S, LIKKEI J, VEMBAR P, et al. Automotive functional safety = safety + security[C]//The 1st International Conference on Security of Internet of Things (SECURIT), 2012: 150-159.

[107] Airlines Electronic Engineering Committee. Arinc 811: commercial aircraft information security concepts of operation and process framework[S]. 2005.

[108] RTCA/EUROCAE. ED-202A/DO-326A: airworthiness security process specification[S]. 2010.

[109] RTCA/EUROCAE. ED-203/DO-356: airworthiness security methods and considerations[S]. 2014.

[110] NIST Smart Grid, Cyber Security Working Group. NISTIR 7628: Guidelines for smart grid cyber security[EB/OL]. http://www.nist.gov/smartgrid/upload/ nistir-7628_total.pdf.

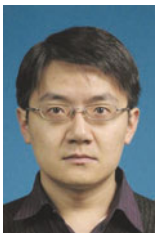
[111] International Electrotechnical Commission. IEC 61850: communication networks and systems in substations[S]. 2003.

[112] International Electrotechnical Commission. IEC 62351: power systems management and associated information exchange - data and communications security[S]. 2013.

[113] CLEVELAND F. IEC TC57 security standards for the power system's information infrastructure-beyond simple encryption[C]//IEEE/PES Transmission and Distribution Conference and Exhibition, 2006: 1079-1087.

[114] North American Electric Reliability Corporation. Critical infrastructure protection (CIP) reliability standards[S]. 2009.

About the authors



SUN Cong [corresponding author] was born in Xingping, Shannxi Province. He received a B.S. degree in computer science from Zhejiang University in 2005, and a Ph.D. degree in computer science from Peking University in 2011. He is currently an associate professor with the School of Cyber Engineering, Xidian University. His research interests include systems software and information security. (Email: suncong@xidian.edu.cn)

MA Jianfeng received a B.Sc. degree in mathematics from Shaanxi Normal University in 1985, and M.Sc. and Ph.D. degrees in computer software and communications engineering from Xidian University in 1988 and 1995, respectively. From 1999 to 2001, he was a research fellow with Nanyang Technological University of Singapore. Currently, he is a professor and Ph.D. supervisor in the School of Computer Science and Technology at Xidian University. He is also Director of Shaanxi Key Laboratory of Network and System Security. His research interests include information and network security, wireless and mobile computing systems, and computer networks. (Email: jfma@mail.xidian.edu.cn)

YAO Qingsong was born in Songzi, Hubei Province. He received a B.S. degree in computer science and technology from Xidian University in 2004, and a Ph.D. degree in computer science and technology from Xi'an Jiaotong University in 2012. He is currently an associate professor with the School of Cyber Engineering, Xidian University. His research interests include network security. (Email: qsyao@xidian.edu.cn)