Secure Control: Towards Survivable Cyber-Physical Systems*

Alvaro A. Cárdenas Saurabh Amin Shankar Sastry
University of California, Berkeley

Abstract

In this position paper we investigate the security of cyber-physical systems. We (1) identify and define the problem of secure control, (2) investigate the defenses that information security *and* control theory *can provide, and (3) propose a set of* challenges *that need to be addressed to improve the* survivability *of cyber-physical systems.*

1 Introduction

Cyber-Physical Systems (CPS) integrate computing and communication capabilities with monitoring and control of entities in the physical world. These systems are usually composed by a set of networked agents, including: sensors, actuators, control processing units, and communication devices; see Fig.1.

While some forms of CPS are already in use, the widespread growth of wireless embedded sensors and actuators is creating several new applications –in areas such as medical devices, autonomous vehicles, and smart structures– and increasing the role of existing ones –such as Supervisory Control and Data Acquisition (SCADA) systems.

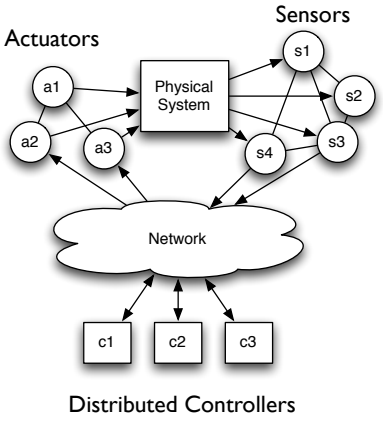

Figure 1. The general architecture of cyber-physical systems

Many of these applications are *safety-critical*: their failure can cause irreparable harm to the physical system being controlled and to people who depend on it. SCADA systems, in particular, perform vital functions in national critical infrastructures, such as electric power distribution, oil and natural gas, water and waste-water distribution systems, and transportation systems.The disruption of these control systems could have a significant impact on public health, safety and lead to large economic losses.

While most of the effort for protecting CPS systems (and SCADA in particular) has been done in *reliability* (the protection against random failures), there is an urgent growing concern for the protection against malicious cyber attacks [21, 9, 31, 12].

In this paper we study the problem of *secure control*. We first characterize the properties required from a secure control system, and the possible threats. Then we analyze what elements from (1) information security, (2) sensor network security, and (3) control theory can be used to solve our problems. We conclude that while these fields can give necessary mechanisms for the security of control systems, these mechanisms alone are not sufficient for the security of CPS.

In particular, computer security and sensor network security have focused on prevention mechanisms, but do not address how a control system can continue to function when under attack. Control systems, on the other hand, have strong results on robust and fault-tolerant algorithms against well-defined uncertainties or faults, but there is very little work accounting for faults caused by a malicious adversary. Therefore, we conclude the paper by outlining some challenges in secure control.

2 Securing CPS: Goals and Threats

The estimation and control algorithms used in CPS are designed to satisfy certain **operational goals**, such as, closed-loop stability, safety, liveness, or the optimization of a performance function. Intuitively, our **security goal** is to protect these *operational goals* from a malicious party attacking our cyber infrastructure.

Security, however, also needs to deal with non-operational goals. For example, if the measurements collected by the sensor network contain sensitive private information we must ensure that only authorized individuals can obtain this data.

2.1 Security Goals

In this section we study how the traditional security goals of *integrity*, *availability*, and *confidentiality* can be interpreted for CPS.

Integrity refers to the trustworthiness of data or resources [4]. A lack of integrity results in **deception**: when an authorized party receives false data and believes it to be true [13]. Integrity in CPS can therefore be viewed as the ability to maintain the *operational goals* by *preventing*, *detecting*, or *surviving deception attacks* in the information sent and received by the sensors, the controllers, and the actuators.

Availability refers to the ability of a system of being accessible and usable upon demand [13]. Lack of availability results in

*This work was supported in part by TRUST (Team for Research in Ubiquitous Secure Technology), which receives support from the National Science Foundation (NSF award number CCF-0424422) and the following organizations: AFOSR (#FA9550-06-1-0244) Cisco, British Telecom, ESCHER, HP, IBM, iCAST, Intel, Microsoft, ORNL, Pirelli, Qualcomm, Sun, Symantec, Telecom Italia, and United Technologies.

1545-0678/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDCS.Workshops.2008.40

denial of service (DoS) [11]. While in most computer systems a temporary DoS attack may not compromise their services (a system may operate normally when it becomes available again), the strong *real-time* constraints of many cyber-physical systems introduce new challenges. For example, if a critical physical process is unstable in open loop, a DoS on the sensor measurements may render the controller unable to prevent irreparable damages to the system and entities around it.

The goal of availability in CPS is therefore, to maintain the *operational goals* by *preventing* or *surviving DoS attacks* to the information collected by the sensor networks, the commands given by the controllers, and the physical actions taken by the actuators.

Confidentiality refers to the ability to keep information secret from unauthorized users. A lack of confidentiality results in **disclosure**, a circumstance or event whereby an entity gains access to data for which it is not authorized [13].

The use of CPS in commercial applications has the potential risk of violating a users' **privacy**: even apparently innocuous information such as humidity measurements may reveal sensitive personal information [14]. Additionally, CPS used for medical systems must abide with federal regulations such as the Health Insurance Portability and Accountability Act (HIPAA), which mandates the protection of a patient's data.

Confidentiality in CPS must prevent an adversary from inferring the *state* of the physical system by eavesdropping on the communication channels between the sensors and the controller, and between the controller and the actuator.

While confidentiality is an important property in CPS, we believe that the inclusion of a physical system and real-time automated decision making does not affect current research in mechanisms for enforcing confidentiality. Therefore in the remaining of this paper we focus only on *deception* and *DoS* attacks.

2.2 Summary of Attacks

A general abstraction of CPS can be seen in Fig. 2. Let y represent the sensor measurements, and u the control commands sent to the actuators. A controller can usually be divided in two components: an **estimation** algorithm to track the state of the physical system given y, and the **control** algorithm which selects a control command u given the current estimate.

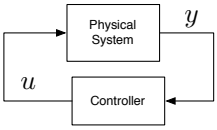

Figure 2. Abstraction of CPS

Attacks to CPS (Fig. 3) can be summarized as follows: A1 and A3 represent *deception attacks*, where the adversary sends false information $\tilde{y} \neq y$ or $\tilde{u} \neq u$ from (one or more) sensors or controllers. The false information can include: an incorrect measurement, the incorrect time when the measurement was observed, or the incorrect sender id. The adversary can launch these attacks by obtaining the *secret key* or by compromising some sensors (A1) or controllers (A3).

A2 and A4 represent *DoS attacks*, where the adversary prevents the controller from receiving sensor measurements. To launch a DoS the adversary can jam the communication channels, compromise devices and prevent them from sending data, attack the routing protocols, etc.

A5 represents a direct attack against the actuators or an external physical attack on the plant. From an algorithmic perspective we cannot provide solutions to these attacks (other than detecting them). Therefore, significant efforts must be placed in *deterring* and *preventing* the compromise of actuators and other direct attacks against the physical system, by for example, securing the physical system, monitoring cameras etc. Although these attacks are more devastating, we believe that a risk-averse adversary will launch cyber-attacks A1-A4 because (1) it is more difficult to identify and prosecute the culprits, (2) it is not physically dangerous for the attacker, and (3) the attacker may not be constrained by geography or distance to the network.

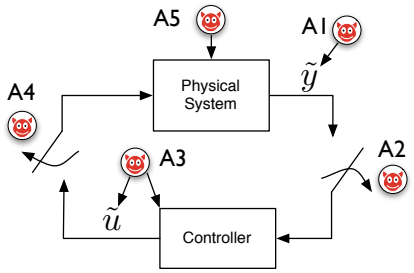

Figure 3. Attacks

3 Related Work in Information Security

The security of SCADA, and CPS in general, is a recognized concern; but, is there a new research problem? After all, the research fields of computer and sensor network security have developed mature technologies and design principles for protecting systems against cyber-attacks. In this section we study the results from these fields and identify their shortcomings.

3.1 Information Security: What can help?

Information security has developed mature technologies that can help us secure CPS. We divide their tools in three: *proactive* mechanisms, *reactive* mechanisms, and *design and analysis principles*.

3.1.1 Proactive Mechanisms

An important tool for securing distributed systems is **authentication**. Authentication schemes prevent humans and devices from impersonating another entity in the system. **Access control** prevents unauthorized access to the system: it prevents outsiders (unauthenticated principals) from gaining access to the network, while imposing and enforcing proper restrictions on what insiders (authenticated principals) can do. **Accountability** can be maintained by keeping audit logs of the actions by authenticated entities.

Secure communications between two *honest* entities is achieved with the help of **Message authentication codes** or **digital signatures** (they can detect when messages have been tampered by a *third party*). **Message freshness** can also be guaranteed by the use of *timestamps* (which require secure time-synchronization protocols) or by challenge and response mechanisms.

Additionally, **verification** tools and **software security** can test the correctness of the system design and implementation, thereby limiting the number of vulnerabilities.

The security of CPS also depends on **sensor network security** [24]. Most of the efforts for the security of sensor networks have

focused on designing a secure communication infrastructure in the presence of malicious insiders. The main results include efficient algorithms for: (1) bootstrapping security associations and key management [10, 25] to build a trusted infrastructure, (2) secure communication [17, 20] and (3) secure routing protocols [18, 23].

Finally, there are several security design principles that can be useful for designing secure control systems [27, 3]. **Redundancy**, for example, is a way to prevent a single-point of failure. **Diversity** is a way to prevent that a single attack vector can compromise all the replicas (the added redundancy). And the **separation of privilege** principle is a design guideline to limit the amount to privileges that a corrupted entity can have.

3.1.2 Reactive Mechanisms

Because we can never rule out successful attacks, security engineering has recognized the importance of **detection** and **response** [29, 2, 3].

Intrusion detection systems are, in general, application-dependent,s an attack is defined in the context of an application. While they are useful in many cases, they are not perfect: false alarms and missed detections are impossible to avoid because detecting malicious logic is an undecidable problem [1].

On the other hand, while response and recovery are often the most important aspect of security engineering, it is one of the most neglected [2]. A typical example of a response mechanism is *key revocation* [6].

3.1.3 Design and Analysis Principles

In security research, when we say that a system is *secure*, we usually mean that the system is *secure as long as our adversary model and trust assumptions are satisfied in practice*.

In general, the **adversary model** is a way of restricting the *scope* of the problem. A careful balance must be kept when defining the adversary model. On one hand, restrictive adversary models, such as assuming that an attacker will follow a Bernoulli distribution when performing DoS attacks, will limit the applicability of our analysis (why would an adversary select such a distribution? What is the incentive?). On the other hand, sometimes these restrictive assumptions are useful to start modeling the adversary, in the hopes of giving us better insights into the nature of the problem, and of how to start obtaining better models in time. As long as the adversary assumptions are explained clearly, we believe that defining a problem with a restrictive adversary is a reasonable first step.

An essential part of security analysis is also in identifying the entities or systems that we **trust**. Trust is generally defined as *accepted dependence* [3]; i.e., trusted systems are systems we rely on. For example, if in Fig 3 we do not trust the actuators, there is very little we can do to secure the system! A human, device or system is **trustworthy** if we have *evidence* to believe they can be trusted.

3.2 Information Security: What is missing?

Althought the security mechanims we have mentioned can improve the security of CPS, n practice, they can often be subverted: inevitable human errors, software bugs, misconfigured devices, and design flaws create many vulnerabilities that attackers can use to launch successful attacks. This is specially undesirable as most CPS are saftey critical, so they must continue to function even when under attack [16]. To capture this notion we define survivability for CPS.

Definition 1: Survivability is the ability of the CPS to maintain or provide graceful-degradation of CPS *operational goals* when under attack.

We now argue that previous research in computer security has not considered the tools necessary to study CPS survivability in a theoretical way.

Claim 1: *Proactive* mechanisms in sensor network security have focused on *integrity* and *availability* from a communication-network point of view. They have not considered how *deception* and *DoS* attacks affect the application layer service; i.e., how successful attacks affect our *estimation* and *control* algorithms –and ultimately, how they affect the physical world.

Claim 2: Intrusion *detection* systems have not considered algorithms for detecting *deception* attacks against *estimation* and *control* algorithms. In particular, previous detection of deception attacks launched by compromised sensor nodes assume a large number of redundant sensors [33]: they *have not considered the dynamics of the physical system* and how this model can be used to detect a compromised node. Furthermore, there has not been any detection algorithm to identify deception attacks launched by compromised controllers.

Claim 3: Most intrusion *response* mechanisms in security involve a human in the loop. Because CPS use *autonomous, real-time decision making algorithms* for controlling the physical world, they introduce new challenges for the design and analysis of secure systems: a response by a human may impose time delays that may compromise the safety of the system. Therefore, we must design autonomous and real-time detection and response algorithms for safety-critical applications.

Claim 4: We need to define security with respect to an adversary model. Previous research has not studied rational adversary models against CPS.

4 Related work in automatic control

The architecture of CPS in Fig. 1 indicates a spatially distributed system in which the system, sensors, actuators, and controllers coordinate their operation over a communication network to achieve some performance goal. A typical problem in control theory is to design a control policy to ensure that under the feedback-loop, an open-loop unstable system remains stable. The nature of such systems impose several constraints on the design of control algorithms.

First, the constraints imposed by communication networks such as limited capacity, random delay, packet loss and intermittent network connectivity can cause DoS. Under DoS the actuator may fail to receive certain packets from the controller that are critical to stabilize an open-loop unstable system. As a result the system may enter a state from which it might be impossible to stabilize it. If the information content of measurement and/or control packets is compromised it may lead to implementation of incorrect control policies. These factors strongly indicate the need to incorporate network characteristics in the design of control algorithms. Such problems are studied in **robust networked control systems** [28, 15].

Secondly, the sensors and actuators are vulnerable to random failures. To enable desired operation under failure modes, we need to introduce appropriate redundancies at the design stage. Such techniques also aim at reconfigurable control and graceful performance degradation in the event of failure thus limiting the negative effects that failure can cause. Research in **fault tolerant control** addresses these issues [5].

Lastly, the system components may be typically located in open and may be limited in transmission power and memory. This moti-

vates the need of designing distributed algorithms that can perform a global task with local information exchange and limited computation at nodes. Research in **distributed estimation**, which falls in the more general area of **consensus problems**, addresses these problems [22]. By way of suitable examples, we now discuss some of the current state-of-the-art in each of these fields.

4.1 Robust networked control systems

We begin by considering a scenario where the system and remote estimator communicate over a communication network. The estimator's goal is to generate recursive state estimates based on measurements sent by the sensor. Under perfect communication there is no loss of data and packets arrive at the estimator instantaneously. Under perfect integrity, the measured data is not compromised. For this ideal case, the Kalman filter is the optimal estimator.

Let us recall the basic Kalman filter in systems theory for a discrete-time linear dynamical system

$$x_{k+1} = Ax_k + w_k, \quad y_k = Cx_k + v_k \tag{1}$$

where, $k \in \mathbb{N}$, $x_k, w_k \in \mathbb{R}^n$ denote the state vector and state noise respectively, $y_k, v_k \in \mathbb{R}^p$ denote the output vector and measurement noise respectively. Here, x_0 is the initial state with Gaussian random vector with zero mean and covariance Σ_0, and w_k and v_k are independent Gaussian random vectors with zero mean and covariance $Q \geq 0$ and $R > 0$ respectively. It is known that under the assumption that (A, C) is detectable and (A, Q) is stabilizable, the estimation error covariance of the Kalman filter converges to a unique steady state value from any initial condition.
The optimal estimate of $x_k, k \in \mathbb{N}$ and the error covariance matrix given the past measurements $Y_k = \{y_0, \ldots, y_{k-1}\}$ are denoted by $\hat{x}_{k|k-1} = E[x_k|Y_k]$ and $P_{k|k-1} = E[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^\top]$ respectively. Starting with $x_{0|-1} = 0$ and $P_{0|-1} = \Sigma_0$, the update equations for basic Kalman filter can be computed as

$$\begin{aligned}
\hat{x}_{k+1|k} &= A\hat{x}_{k|k} \\
P_{k+1|k+1} &= AP_{k|k}A^\top + Q \\
\hat{x}_{k+1|k+1} &= \hat{x}_{k|k} + F_{k+1}(y_{k+1} - C\hat{x}_{k+1|k}) \qquad (2)\\
P_{k+1|k+1} &= (I - F_{k+1}C)P_{k+1|k} \qquad (3)
\end{aligned}$$

where, $F_{k+1} = P_{k+1|k}C^\top(CP_{k+1|k}C^\top + R)^{-1}$ is the Kalman gain matrix.

4.1.1 Quantifying robustness

From a Quality of Service (QoS) point-of-view, every raw measurement y_k sent over the communication network may not arrive at the remote estimator. In particular, packets may be dropped when the network is congested. Although this situation is not necessarily adversarial, its effect is similar to a DoS attack. (Refer to A2 in Fig. 3). This has motivated researchers to design Kalman filters that take into account the history of packet losses. Two widely used packet loss models are: the Bernoulli model and the Gilbert-Elliot model. The Bernoulli model describes the packet loss process by independent and identically distributed Bernoulli random variables. The Gilbert-Elliot model considers that the network state evolves according to a Markov chain. This model can represent bursty packet losses.

The authors in [28] have studied the performance of the Kalman filter under Bernoulli loss model. The packet loss is modeled as a random process $\gamma_k \in \{0, 1\}$ where, $\gamma_k = 1$ signifies successful transmission and $\gamma_k = 0$ signifies lost packet. Let $Pr(\gamma_k = 1) = \lambda_k$. Given the past measurements, $\{\gamma_0, \ldots, \gamma_{k-1}\}$ and $\{y_l : \gamma_k = 1, \forall l \leq k-1\}$, the update equations (2),(3) get modified as

$$\hat{x}_{k+1|k+1} = \hat{x}_{k|k} + \gamma_{k+1}F_{k+1}(y_{k+1} - C\hat{x}_{k+1|k}) \tag{4}$$
$$P_{k+1|k+1} = (I - \gamma_{k+1}F_{k+1}C)P_{k+1|k} \tag{5}$$

Since both $\hat{x}_{k+1|k+1}$ and $P_{k+1|k+1}$ become functions of γ_k, they are random variables. For $Pr(\gamma_k = 1) = \lambda \in (0, 1]$, it was shown that depending of properties of system (1), there exists a critical value of packet dropout probability λ_c above which the expected value of error covariance $E[P_{k|k}]$ becomes unbounded.

Several extensions to this formulation have been proposed by various researchers [15]. However, a key assumption in the design of such estimators is that the QoS parameters for the network are known (e.g., the packet drop probability). This is a restrictive assumption and requires inference of QoS parameters. Recently, researchers have started considering **state of the communication network** as a stochastic event that depends on its QoS. These approaches estimate the state of the network together with the system state.

4.1.2 Increasing robustness

In addition to uncertain QoS parameters, other important statistics such as the distribution of measurement noise may not be fully known. This situation is similar to a deception attack. (Refer to A1 in Fig. 3.) If the actual values of these parameters deviate significantly from what is assumed in the design, the estimation performance might degrade catastrophically. A promising approach to design algorithms that are robust to parameter variations is the minimax approach or robust estimation. Minimax approaches to design estimators can be viewed as a game in which the performance of the estimator depends on the elements of a set of estimators and an uncertainty set that includes the set of possible values the unknown parameters can assume. We now briefly discuss the main idea behind **minimax or robust estimation**.

Let $\mathcal{U}$ denote the space of estimators and $\mathcal{V}$ denote the space of uncertain parameters. The estimator performance defined as a function of estimation error $\mathcal{J} = \mathcal{J}(u, v)$ is to be minimized by $u \in U$ and maximized by $v \in V \subset \mathcal{V}$. Here, $U \subset \mathcal{U}$ is the set of admissible estimators and $V \subset \mathcal{V}$ is the set of admissible uncertainties. Thus, the triple $(\mathcal{J}, U, V)$ define a zero-sum game. For a given $v \in V$, the estimator $u^*(v)$ is optimal if $\mathcal{J}(u^*(v), v) = \inf_{u \in U} \mathcal{J}(u, v)$. On the other hand, $v^*(u)$ is worst-case uncertainty point for $u \in U$ if $v^*(u) \in \arg\max_{v \in V} \mathcal{J}(u, v)$. An estimator u_L is *minimax* or best-case robust filter for the game $(\mathcal{J}, U, V)$ if

$$u_L \in \arg\min_{u \in U} \sup_{v \in V} \mathcal{J}(u, v).$$

Similarly, an uncertainty point v_R is worst-case point for the game $(\mathcal{J}, U, V)$ if

$$v_L \in \arg\max_{v \in V} \inf_{u \in U} \mathcal{J}(u, v).$$

The pair $(u_L, v_L) \in U \times V$ is called a saddle point solution to the game $(\mathcal{J}, U, V)$ if

$$\mathcal{J}(u, v_L) \leq \mathcal{J}(u_L, v_L) \leq \mathcal{J}(u_L, v) \tag{6}$$

If (u_L, v_L) is a saddle point solution to the game $(\mathcal{J}, U, V)$, then by equation (6) one can conclude that no estimator other than u_L gives

the best performance at v_L and that v_L is the worst-case uncertainty point for u_L. Thus, u_L is the desired minimax estimator. Many researchers have worked on minimax estimation problems [32, 26] which essentially reduces to finding conditions for existence of saddle point solution.

4.2 Fault-tolerant control

The components of CPS are vulnerable to random failures and service degradation. In the area of automatic control, **fault detection and diagnosis (FDD)** methods as well as **fault tolerant control (FTC)** designs have been developed in order to increase the reliability and maintainability of systems prone to failures [5]. The main goal of a FTC system is to maintain stability and ensure an acceptable performance level under normal operating conditions as well as under component malfunctions by employing appropriate physical and/or analytical redundancies.

FTC approaches can be broadly classified into two categories: passive and active. In **passive FTC**, a limited number of faulty configurations are taken into account while designing the controller. Once designed, the passive controller can compensate for the anticipated configurations without any FDD schemes or reconfigurable control design. Thus in effect, passive FTC can be viewed as robust control design for limited number of failures. From a performance viewpoint, the passive FTC approaches are conservative.

Consider the following stabilization problem example [30]:

$$\dot{x} = Ax + Bu, \quad y_1 = C_1 x, \ldots, y_n = C_m x$$

with $x \in \mathbb{R}^n, u \in \mathbb{R}^q, y_i \in \mathbb{R}, i = \{1, m\}$. The measurements y_i denote the output of sensor i that can potentially fail. Stoustrup and Blondel [30] show that if (A, B) is stabilizable and (C_i, A), $i = \{1, m\}$ are detectable, there exists a fault tolerant compensator that simultaneously stabilizes the system in the case when at most one of the m sensors fail.

In contrast to passive FTC, **active FTC** has a **fault detection and isolation (FDI)** unit in the control system. Generally speaking, an active FTC system has four components [5]: FDD unit, reconfiguration mechanism, reconfigurable controller and reference governor. The FDD unit estimates the system state and fault parameters based on measurement and control data. Upon detection of a fault, the FDD activates the reconfiguration mechanism and the reconfigurable controller (RC) designs the control parameters to ensure stability and acceptable performance. The RC can switch between one of the several predesigned control laws or can synthesize a new control law in real-time. In addition, the RC also ensures that the trajectory tracking goal provided by reference governor is achieved. In the event of performance degradation, the reference governor may be used to adjust control inputs. The main problems in the design of active FTC systems are: (1) RC design, (2) FDD schemes that are sensitive to faults and robust to model uncertainties and operating condition variations, (3) reconfiguration mechanism to suitably recover normal operating performance.

4.3 Distributed Estimation

Recent activity in distributed coordination and control of cooperative agents has resulted in advances in distributed estimation schemes [22]. These distributed estimation techniques explicitly consider communication constraints of participating agents and aim at scalability to large network sizes by using consensus algorithms. We briefly discuss distributed Kalman filter (DKF) of [22] to motivate the application of similar algorithms for estimation problems in CPS.

Consider a system similar to equation (1) in which $x_k \in \mathbb{R}^n$ denote the state vector and $y_k, v_k \in \mathbb{R}^{mp}$ denote the output vector of $p-$dimensional measurement obtained from from m sensors. Olfat-Saber [22] shows that for a sensor network monitoring a process of dimension n with m sensors, m micro-Kalman filters which are embedded in each sensor can jointly arrive at same estimate of $\hat{x}$ via a consensus approach. The nodes of the sensor network solve two consensus problems to jointly calculate the average inverse error covariance matrix and average measurements at every iteration of the DKF.

4.4 Control for CPS: What is missing?

The field of automatic control is more mature in comparison to information security; however, despite great achievements in the field of nonlinear and hybrid systems theory, robust, adaptive, game-theoretic and fault-tolerant control, much more needs to be done for design of secure control algorithms to ensure survivability of CPS. We propose that there is a need of further research in the following areas of control theory.

Claim 5: We need to design novel *robust control and estimation algorithms* that consider more **realistic attack models** from a security point-of-view. As identified in Section 3.2, these attack models should model deception and DoS attacks. Under the influence of such attacks, these algorithms should optimize the worst-case performance. **Game theoretic** techniques developed in economics for modeling rational adversaries might also be useful for this task.

Claim 6: In addition to the state of the system to be controlled, the *state of communication network* should be jointly estimated. Approaches to estimate the *indicators of QoS and integrity* of the communication network based on available network data should be developed. The estimated state of the network should be used to design *transmission policies* for sensors and actuators as well as *scheduling policies* for controllers to optimize performance.

Claim 7: Physical and analytical redundancies should be combined with security principles (e.g., diversity and separation of duty) to **adapts** or **reschedules** its operation during attacks. For example, under sensor faults or when only intermittent sensory information is available, the system should be able to operate using open-loop control for a sufficient amount of time.

Claim 8: A notion of **trustworthiness** should be associated with different components of CPS and **trust management** schemes should be designed when the above redundancies are in place. For example, if the trustworthiness metric of a component deviates significantly from the trust that is associated with the component, then the component may be regarded as insecure and its contribution toward the operation of DNCS may be restricted or discarded.

5 Conclusions

Definition 2: The **Secure control** problem refers to any of the algorithms and architectures designed to *survive* deception and DoS attacks against CPS under a well-defined adversary model and trust assumptions. As we have discussed, there are many research challenges to achieve our secure control objectives. Some of them are,

Challenge 1: In the design and analysis of secure control algorithms we need to introduce a **trust analysis** of the CPS architecture, and **realistic and rational adversary models** that can launch *deception* or *DoS* attacks against CPS.

Challenge 2: Design new **proactive** *algorithms* and *architectures* that are **robust** against a given adversary model and that provide provable performance bounds (to understand the limits of the resiliency of the algorithms).

Challenge 3: Design **reactive** algorithms and architectures for **real-time detection and response** for a given adversary model.

Challenge 4: In the design of these new algorithms we need to study how attacks affect the performance of the estimation and control algorithms –and ultimately, how they affect the real world– by incorporating the dynamical models of the systems being monitored and controlled.

We hope that this paper, and these challenges and definitions provide enough motivation for future discussions, and interest for analytical work in secure control.

References

[1] L. Adleman. An abstract theory of computer viruses. In *Advances in Cryptology–Proceedings of CRYPTO '88*, pages 354–374, New York, NY, USA, 1990. Springer-Verlag New York, Inc.

[2] R. Anderson. *Security Engineering*. Wiley, 2001.

[3] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–32, January-March 2004.

[4] M. Bishop. *Computer Security, Art and Science*. Addison-Wesley, 2003.

[5] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and fault-tolerant control*. Springer-Verlag, September 26 2003.

[6] H. Chan, V. D. Gligor, A. Perrig, and G. Muralidharan. On the distribution and revocation of cryptographic keys in sensor networks. *IEEE Trans. Dependable Secur. Comput.*, 2005.

[7] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium Research in Security and Privacy*, 2003.

[8] H. Chan, A. Perrig, and D. Song. Secure hierarchical in-network aggregation in sensor networks. In *Proceedings of the ACM Conference on Computer and Communications Security*, pages 278–287, 2006.

[9] J. Eisenhauer, P. Donnelly, M. Ellis, and M. O'Brien. *Roadmap to Secure Control Systems in the Energy Sector*. Energetics Incorporated. Sponsored by the U.S. Department of Energy and the U.S. Department of Homeland Security, January 2006.

[10] L. Eschenauer and V. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 41–47, 2002.

[11] V. D. Gligor. A note on denial-of-service in operating systems. *IEEE Transactions on Software Engineering*, pages 320–324, May 1984.

[12] A. Greenberg. *America's Hackable Backbone*. Forbes, http://www.forbes.com/logistics/2007/08/22/scada-hackers-infrastructure-tech-security-cx_ag_0822hack.html, August 2007.

[13] N. W. Group. Internet security glossary. http://rfc.net/rfc2828.html, May 2000.

[14] J. Han, A. Jain, M. Luk, and A. Perrig. Don't sweat your privacy: Using humidity to detect human presence. In *Proceedings of 5th International Workshop on Privacy in UbiComp (UbiPriv'07)*, September 2007.

[15] J. P. Hepanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, January 2007.

[16] I. John H. Marburger and E. F. Kvamme. Leadership under challenge: Information technology R&D in a competitive world. An assessment of the federal networking and information technology R&D program. Technical report, President's Council of Advisors on Science and Technology, August 2007.

[17] C. Karlof, N. Sastry, and D. Wagner. Tinysec: A link layer security architecture for wireless sensor networks. In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems*, November 2004.

[18] C. Karlof and D. Wagner. Secure routing in sensor networks: Attacks and countermeasures. In *Ad Hoc Networks, vol 1, issues 2–3 (Special Issue on Sensor Network Applications and Protocols), pp. 293-315, Elsevier*, September 2003.

[19] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. In *ACM Trans. Programming Languages and Systems*, July 1982.

[20] M. Luk, G. Mezzour, A. Perrig, and V. Gligor. Minisec: A secure sensor network communication architecture. In *Sixth International Conference on Information Processing in Sensor Networks (IPSN 2007)*, April 2007.

[21] U. S. G. A. Office. Critical infrastructure protection. Multiple efforts to secure control systems are under way, but challenges remain. Technical Report GAO-07-1036, Report to Congressional Requesters, 2007 2007.

[22] R. Olfat-Saber. Distributed kalman filter with embedded consensus filter. In *Proceedings of CDC and ECC*, Seville, Spain, 2005.

[23] B. Parno, M. Luk, E. Gaustad, and A. Perrig. Secure sensor network routing: A clean-slate approach. In *Proceedings of the 2nd Conference on Future Networking Technologies (CoNEXT 2006)*, December 2006.

[24] A. Perrig, J. A. Stankovic, and D. Wagner. Security in wireless sensor networks. *Commun. ACM*, 47(6):53–57, 2004.

[25] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. E. Culler. Spins: Security protocols for sensor networks. *Wireless Networks*, 8(5):521–534, September 2002.

[26] I. R. Peterson and A. V. Savkin. *Robust Kalman filtering for signals and systems with large uncertainties*. Birkhuser, Boston, February 1999.

[27] J. H. Saltzer and M. D. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, September 1975.

[28] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry. Foundations of control and estimation over lossy networks. *Proceedings of the IEEE*, 95(1):163–187, January 2007.

[29] B. Schneier. Managed security monitoring: Network security for the 21st century. *Computers & Security*, (20):491–503, 2001.

[30] J. Stoustrup and V. D. Blondel. Fault tolerant control: a simultaneous stabilization result. *IEEE Transactions on Automatic Control*, 49(2):305–310, February 2004.

[31] R. J. Turk. Cyber incidents involving control systems. Technical Report INL/EXT-05-00671, Idao National Laboratory, October 2005.

[32] S. Verdu and H. V. Poor. On minimax robustness: A general approach and applications. *IEEE Transactions on Information Theory*, 30(2):328–340, March 1984.

[33] Q. Zhang, T. Yu, and P. Ning. A framework for identifying compromised nodes in sensor networks. In *Proceedings of the 2nd International Conference on Security and Privacy in Communication Networks (SecureComm 2006)*, August 2006.