Chapter 1

MODELING SECURITY IN CYBER-PHYSICAL SYSTEMS*

Mike Burmester, Emmanouil Magkos and Vassilis Chrissikopoulos

Abstract We propose a framework for modeling the security of cyber-physical systems in which the behavior of the adversary is controlled by a threat model that captures both the cyber aspects (with discrete values) as well as the physical aspects (with continuous values) of such systems in a unified way. In particular, it addresses combined (dependent) vector attacks, and synchronization/localization issues. The framework identifies the cyber-physical features specified by the security policies that need to be protected, and can be used for proving formally the security of cyber-physical systems.

Keywords: Cyber-physical systems, threat models, protocols for treaty verification.

1. Introduction

The rapid growth of information and communication technologies has prompted the expansion of network computer systems that address real-world applications, including physical and social applications. This has led to the integration of computing and communication technologies with physical processes, under the name of *cyber-physical systems* (CPS). CPS capture novel aspects of networked systems that include integrating distributed computing systems with monitoring and controlling entities in the physical environment. For example in real-time control systems a hierarchy of sensors, actuators, and control processing components are connected to centralized control stations. Other examples include smart grid systems and supervisory control and data acquisition (SCADA) systems that monitor power, gas/oil transportation, water and waste-

*Part of this material is based upon work by the first author supported by the National Science Foundation under Grant No. DUE 1027217.

water distribution. Such systems used to be stand-alone networks in physically protected locations, using proprietary technology. Nowadays software, hardware and communication technologies are used to extend their connectivity and improve their operations.

Prior work on control systems, by focusing on reliability and resilience, *i.e.*, by protecting CPS against random, independent or benign faults and failures of cyber/physical components [8, 30], fails to adequately address integrity, confidentiality and denial-of-service threats [21, 14, 24, 43, 33, 13]. In addition, traditional computer and network security approaches do not address in a unified way how systems outlive malicious attacks (survivability) or how they recover after an attack (recoverability) [22, 24, 33, 52].

Securing a CPS goes well beyond securing the individual system components separately. A motivated and high-skilled attacker may use a multi-vector attack that exploits the weaknesses of the separate components of the system, *e.g.*, the physical and cyber components, none of which may pose a serious threat for the corresponding component. The combined effect, however, may be catastrophic (the attack vectors may be dependent). An example of a multi-vector attack is the Stuxnet attack [23] that targeted nuclear centrifuges: in this attack a worm that uses zero-day exploits spreads to Windows machines via LANs or USB disks, carrying a malware payload that infects and reprograms programmable logic controllers. An insider SCADA attack on a sewage treatment system in Maroochy Shire, Queensland, Australia, caused 80,000 liters of raw sewage to be released into local rivers and parks [48]. Another example of a multi-vector attack is the Slammer SQL worm attack, in which a private computer network at the Davis-Besse nuclear power plant in Oak Harbor, Ohio, was infected [38].

There have been many efforts to ensure the security of CPS. These are primarily based on extending mechanisms already used to protect the separate components (cyber and physical) of these systems. However there is no formal security model for CPS that addresses security in a unified framework, and that deals with software threats, hardware threats, network threats and physical threats, possibly combined—although there is a lot of work in the literature highlighting the difficulties in securing physical systems, in particular with regards to timing attacks [27, 49, 37], non-interference [25], execution monitoring [35, 36, 27]. One of our goals in this article is to address this issue. The approach we shall use is to extend the traditional Byzantine faults model for cryptographical applications to cyber-physical systems.

In the *Byzantine faults* paradigm a cyber system is represented by a set of linked abstract machines (a graph), some of which may be faulty,

and the messages exchanged are represented by formal expressions. The adversary is active and has full control of the faulty components: the adversary can eavesdrop (wiretap), intercept, and corrupt any message and is only limited by the constraints of the cryptographic methods used. In particular the adversary may be computationally unbounded, polynomially bounded, or bounded by the inability to solve a particular "hard" problem. To get reliable (robust) communication for a system with n components with an adversary that is computationally unbounded, the number of faulty components t should be less than $n/2$ (for a fully connected system). This model focuses on the protocol layer and deals with attacks at this layer resulting from interactions between an active attacker and the system parties in a possibly unbounded number of parallel sessions. There are variants of this model in which the power of the adversary is restricted, *e.g.*, the adversary may be passive.

A slightly different model was proposed by Herzberg et al. in [29]. In this case the adversary is computationally bounded and the faulty components are periodically repaired—*e.g.*, compromised keys are refreshed. Security is assured if throughout the life-time of the system the adversary is restricted to compromising $t < n/2$ components of the system (at different times, different components may be compromised), where n is the number of components. This model captures a physical aspect of the system, if we regard the faults as being caused by adversarial operators (insiders) and assume that components are periodically repaired.

Traditional threat models are restrictive and do not adequately capture the security of CPS. In particular, they typically exclude survivability and recovery. For example, abnormal behavior may be tolerated by a CPS: a system may transition to critically vulnerable (*i.e.*, unsafe) states, but eventually converge to a safe state—in the course of time, or with a probability. Furthermore, in the Byzantine faults model the number of faulty components cannot be reduced; in a physical system however, nodes may become non-faulty in a dynamic way, *e.g.*, after sporadic human intervention or because of *Nature*.

Our contribution. The contributions of this article are to:

1 Discuss the inadequacies of traditional adversary models for CPS.

2 Present a high level threat model that captures adversarial behavior in CPS and that addresses multi-vector threats of multi-component systems.

3 Show how this adversarial threat model can be used to secure a typical CPS.

2. A threat model for cyber-physical systems

A cyber-physical system (CPS) is a finite state system consisting of several networked components, some of which may be cyber while others are physical. It can be modeled by a finite, hybrid timed automaton $\mathcal{A}$ [7, 3, 28] *with faults*, which is a tuple

$$(\tau, A, Q, q_0, D, \mathcal{F}),$$

with $\tau\colon t_1, t_2, \ldots$ a strict monotone unbounded *time schedule* of positive real numbers, A a finite set of actions that includes a special symbol "$\perp$", $Q \neq \emptyset$ a finite set of states that is partitioned into *safe* states Q_s, *critical* states Q_c, and *terminal* states Q_t, $q_0 \in Q_s$ an initial state, and $D \subset Q \times Q \times A$ a transition function that is *time triggered*: for $(q, q', a) \in D$ and $t_i \in \tau$,

$$D(t_i):\ q \xrightarrow{t_i,a} q'$$

describes the transition that action a causes at time t_i. Critical states are unsafe states from which the system can recover; terminal states are unsafe states from which the system cannot recover. $\mathcal{F}$ is the *faults distribution* of the CPS, that corresponds to component failure. The transition function D is deterministic when $a \in A\backslash\{\perp\}$ and probabilistic when $a = \perp$. When $a = \perp$ the posteriori state q' is selected by Nature using the distribution $\mathcal{F}$.

A *timed execution* of $\mathcal{A}$ is a path that starts at time t_1 from state q_0:

$$r:\ q_0 \xrightarrow{t_1,a_1} q_1 \xrightarrow{t_2,a_2} q_2 \xrightarrow{t_3,a_3} q_3 \cdots \longrightarrow q_{i-1} \xrightarrow{t_i,a_i} q_i \cdots,$$

and traverses the states q_i instantiated by actions a_i at time t_i.

The parties involved in a CPS are those specified by the system (*e.g.,* the operators), the adversary (an entity that controls all parties that do not adhere to the system policies/specifications), and Nature (the Environment). We use the Game Theory paradigm to model Nature. In particular,

a) Nature uses the probability distribution $\mathcal{F}$ to select from among her strategies for component failure randomly.

b) Nature controls the temporal and location aspects of all events and schedules the state transitions in a timely manner according to the time schedule τ.

c) Nature resolves concurrency issues, by linking events to their *real* start time: if two events take place during $(t_{i-1}, t_i]$ then Nature will schedule them according to the order in which they occurred.[1]

The threat model for a CPS must capture those features of the system that may lead to system failure and the adversarial intent. System failure can result from actions by Nature, the adversary or both (the adversary can manipulate Nature, *e.g.,* in a terrorist attack). The adversary can be *passive* or *active*. Passive adversaries are restricted to eavesdropping on communication channels; active adversaries can additionally modify the contents of the communication channels and use compromised components to undermine the security of the system.

Our threat model restricts the adversary to exploiting specific system vulnerabilities. These are identified by:

a) The *security policies* of the system (*e.g.,* availability of services, resilience, privacy of records, etc),

b) *Vulnerability assessments*, and

c) *Grey-box penetration testing.*

The vulnerabilities involve the components of the system, such as the control systems and the embedded systems and the communication channels, but may also involve the system $\mathcal{A}$ as a whole. The security goal of $\mathcal{A}$ is to prevent the adversary from exploiting these features.

Let $V = \{v_1, v_2, \ldots, v_m\}$ be the set of identified features of the states of Q that are vulnerable and need to be protected. The features v_i are vectors with discrete and/or continuous values. The vulnerabilities of a CPS may be time-dependent. That is, the adversary may only be able to access $v_i \in V$ some times t_i.[2] To identify the vulnerabilities at time t_i we use the function:

$$f : \ (\tau, Q) \ \mapsto \ (\tau, 2^V); \quad (t_i, q_j) \ \rightarrow \ f_i(q_j) \in V,$$

that specifies the vulnerabilities of state q_j the adversary can exploit at time t_i. The threat model of $\mathcal{A}$ is defined by a *timed vulnerabilities transition function* $D_f(\tau)$:

$$D_f(t_i) : \ f_{i-1}(q_{j-1}) \quad \stackrel{t_i, a}{\longrightarrow} \quad f_i(q_j),$$

that specifies the priori and posteriori features of an adversarial exploit/attack during $(t_{i-1}, t_i]$ (Fig. 1). In a passive attack the adversary can eavesdrop on the priori $f_{i-1}(q_{j-1})$-features and the posteriori $f_i(q_j)$-features, and no more. In an active attack the adversary can also cause the transition $D_f(t_i)$, and exploit the priori and posteriori features. We assume that the adversary may have prior knowledge of the vulnerabilities $v_j \in V$ of the system and the structure of $D_f(t_i)$, but not necessarily their values v_j.

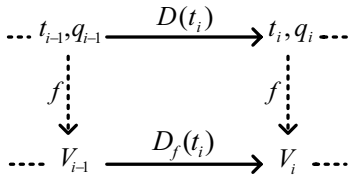

Figure 1. The mapping f that identifies the priori/posteriori vulnerabilities of the states q_{i-1}/q_i of the transition $D_f(\tau)$.

DEFINITION 1 *An adversary that is restricted to the vulnerabilities of the transitions $D_f(\tau)$ is called a* $D_f(\tau)$-adversary. *We say that the automaton $\mathcal{A}$ is* $D_f(\tau)$-tolerant *if it operates as specified in the presence of a $D_f(\tau)$-adversary.*

The specifications for the automaton $\mathcal{A}$ typically require that the system should never enter into a terminal state, and that it should not stay in a critical state for longer than a certain time period. $D_f(\tau)$-*tolerance* guarantees resilience against adversaries that try to exploit the vulnerabilities $v \in V$ of $\mathcal{A}$ and cause it to transition to a state that violates its specifications.

Traditional threat models for cyber systems such as the Byzantine faults model [20] do not capture physical aspects/features/behavior. For example, the state of a system that uses a wireless medium for communication (such as a sensor and/or RFID system) contains discrete values extracted from continuous values (*e.g.,* RF waveforms). There are several attacks that exploit such physical system aspects. For example:

- *Online man-in-the-middle relay attacks* [6, 32] in which the adversary interposes between parties and relays messages, and

- *Side Channel and Power Analysis attacks* [42] in which the adversary exploits information leaked during the physical implementation of protocols.

Both attacks are at the physical layer and are typically excluded from cyber threat models (and their security analysis [11]). To protect against such attacks, one needs physical layer mechanisms (such as temporal and/or location mechanisms, screening, etc).

To motivate our approach we show how the threat transition function $D_f(\tau)$ is used to model the vulnerabilities of some cyber and cyber-physical systems.

2.1 The Byzantine faults model

The Byzantine model [20] assumes a system with n (cyber) components and an adversary that may compromise up to $k < n$ components. In this case the identified vulnerabilities are $f(t_i, q_j) = V_j$, where $V_j \subseteq V$ is a set of $j \in [0 : k]$ faulty components of q_j. The threat transition function is

$$D_f(t_i) : V_j \xrightarrow{t_i,a} V_s,$$

where $V_j \subseteq V_s$. That is, an adversary that has compromised the components of V_j is restricted to attacking those states with $V_s \supseteq V_j$. This defines the allowable system transitions that the adversary can exploit. Note that for this model, faulty components cannot recover.

For the model proposed by Herzberg et al. [29], discussed in the Introduction, the state of the system is repaired/refreshed at regular intervals. This re-labels the faulty components. We then get, $f(t_i, q_j) = z$, $0 \leq z \leq k$, the *number* of faulty components. For this model the vulnerabilities transition function is

$$D_f(t_i) : z \xrightarrow{t_i,a} z',$$

where z' can be any number in $[0 : k]$, if the system has been refreshed during $(t_{i-1}, t_i]$. Otherwise, $0 \leq z \leq z' \leq k$. In this threat model the transitions allow for a reduction of the number of faulty components: for example, if at some point in time the number of faulty components is $z \leq k$, then in the next time period there may be no faulty component (if faulty components are replaced by with non-faulty components). This captures the behavior of certain types of physical faults, *e.g.*, faults that can be fixed.

Such models are typical of physical systems that may tolerate critical state levels provided the system can recover, *e.g.,* provided the faults are fixed and their duration is short. In this cyber-physical model the duration is enforced by Nature, and cannot be manipulated by the adversary.

In the Byzantine model the adversary controls the communication channels of the system: which messages are sent, which messages are received, to whom or by whom, as well as which messages get compromised through faulty components (devices and/or channels). This applies to our model as well, when the communication channels are identified as vulnerabilities.

2.2 Threat transitions for network traffic

For this model $f(t_i, q) = (z_1, \ldots, z_n, Z)$, where z_i is the number of packets sent by node N_i, $i = 1, \ldots, n$, in a network domain and Z is

the traffic volume in that domain (in packets) during the time interval $(t_{i-1}, t_i]$. We distinguish three cases for z_i:

$$c_1 : \ z_i \leq a, \quad c_2 : \ a < z_i \leq b, \quad c_3 : \ b < z_i,$$

with $0 \leq a < b$, where a is an upper bound for normal use and b the maximum tolerated value for packet transmissions (permitted for short periods only); and three cases for Z:

$$C_1 : \ Z \leq A, \quad C_2 : \ A < Z \leq B, \quad C_3 : \ B < Z,$$

with $0 \leq A < B$, where A is a threshold for domain traffic and B the maximum tolerated level.

States for which the constraint C_3 holds are terminal, and will lead to domain shutdown. Similarly, nodes that violate the constraint c_3 are denied access to the domain.[3] States for which C_2 holds are critical. The thresholds a, A are such that $Z \leq A$ if, and only if, for all nodes N_i we have $z_i \leq a$. The system specifications require that when the state of the system is critical ($A < Z \leq B$) then all nodes N_i for which $z_i > a$ reduce the number of packets sent to $\leq a$ at time t_i. Finally, states bound by C_1 are *safe*, provided all the z_i are bounded by the constraints c_1 or c_2. For this model, the vulnerabilities transition function

$$D_f(t_i) : \ (z_1, \ldots, z_n, Z) \quad \xrightarrow{t_i, a} \quad (z'_1, \ldots, z'_n, Z')$$

requires that priori and posteriori states are not terminal, and that if a priori state is critical then the posteriori state must be safe (so $z_i > a$ implies $z'_i \leq a$). This restricts the adversary to attacking states for which the traffic volume Z is bounded by A over time. $D_f(\tau)$-tolerance is achieved by requiring that, whenever the traffic volume exceeds A, all nodes N_i for which $z_i > a$ reduce the number of packets sent to $z'_i \leq a$ at time t_i.

This model addresses attacks in which the adversary may try to exploit the dependence between the vulnerabilities z_i and Z: *e.g.*, when some nodes send $z_i : b \geq z_i > a$ packets (constraint c_2) and the traffic load is critical (constraint C_2). This behavior is checked by restricting the adversary to transitions that lead to states with lesser traffic load. The network is allowed to stay in a critical state for short periods (one time interval in this case). This is a *temporal* feature that captures a physical security aspect that is normally excluded from the threat model of cyber systems, and highlights one of the main differences between cyber and physical systems.

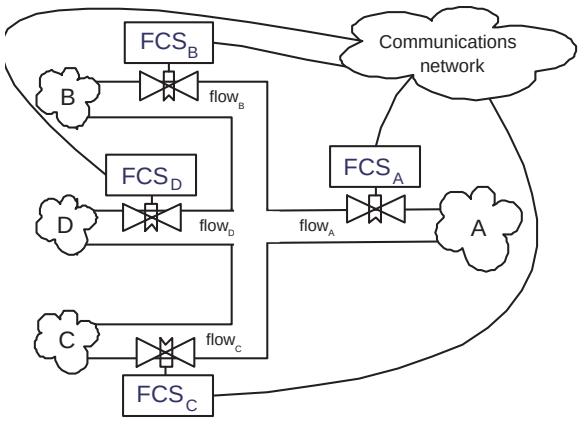

Figure 2. The Russia-Ukraine natural gas grid with subnetworks: B (North EU), D (Ukraine) and C (South EU).

3. Protecting a natural gas grid

This model is motivated by the Russian-Ukrainian dispute over the price of natural gas and the cost of its transportation, which threatened the gas supplies to the European Union (EU) from 2005 to 2010 [17]. Russia provides approximately one quarter of the natural gas consumed in the EU, and 80% of this is piped across Ukraine to reach the EU. Ukraine manages the natural gas grid within Ukraine. For this service Ukraine is allocated a certain amount of natural gas, drawn from the pipeline grid.

The Russia-Ukraine grid starts in Russia and branches in Ukraine, with one branch going to the EU while the other is for domestic supplies.

In this paper we consider an application for which the EU pipeline has two branches, one for North EU (subnetwork B), the other for South EU (subnetwork C)—see Figure 2 (a slightly different application was investigated and analyzed in [1]). Subnetwork A supplies the natural gas from Russia, and subnetwork D provides Ukraine with its allocation. We shall refer to this as the *RU natural gas grid*, or simply the *RU grid*.

We denote by $flow_A$, $flow_B$, $flow_C$ and $flow_D$ the flows of the subnetworks A, B, C and D respectively. The Ukraine entitlement $flow_D$ is 10% of the natural gas that flows to North EU and 5% of the natural gas that flows to South EU. That is,

$$flow_D = 10\% flow_B + 5\% flow_C.$$

Natural gas flows are regulated by Flow Controller Systems (FCS) which automate and control the gas flowing through the pipeline, and enforce flow agreements. A FCS has sensors, actuators, an embedded programmable logic controller (PLC) and a transceiver. The PLC controls the flows in the pipeline and communicates with neighboring FCS to regulate flows. It can execute commands that raise or lower the flows. In the RU grid three flow controllers: FCS_A, FCS_B and FCS_C are controlled by Russia, and regulate the flows coming from Russia and going to North and South EU respectively. A fourth flow controller FCS_D, controlled by Ukraine, regulates the natural gas allocated to Ukraine. All four controllers are located in Ukraine.

SAF: Safety specifications.

- The value of $flow_i$, $i \in \{A, B, C, D\}$, should not exceed the critical threshold flow level and normally be within a safe range—the thresholds and ranges are system parameters.

- $0 \leq flow_A - flow_B - flow_C - flow_D < \varepsilon$, where ε is a small flow variation corresponding to typical gas leakages/fluctuations—a system parameter.

SEC: Security specifications.

- *Flow privacy*: The values of $flow_A$, $flow_B$ and $flow_C$ should not be inferable from signals transmitted by the flow controllers FCS_A, FCS_B and FCS_C.

- *Flow integrity/verifiability*: At all times: $flow_D - 10\% flow_B - 5\% flow_C < \varepsilon$. Furthermore Ukraine should be able to verify correctness.

Threat model, the $D_f(\tau)$-adversary. The vulnerabilities that are identified by the system specifications of the RU grid concern the flows $flow_A$, $flow_B$, $flow_C$, $flow_D$ (SAF) and its communication channels (SEC). In particular:

$$f(state) = (flow_A, flow_B, flow_C, flow_D, z_5, z_6),$$

with $z_5 = flow_A - flow_B - flow_C - flow_D$ and $z_6 = 20 flow_D - 2 flow_B - flow_C$.

The constraints for the safe, critical and terminal flow levels are specified by:

$$
\begin{array}{lll}
c_1 : 0 \leq flow_A < y_1, & c'_1 : y_1 \leq flow_A < y'_1, & c''_1 : y'_1 \leq flow_A, \\
c_2 : 0 \leq flow_B < y_2, & c'_2 : y_2 \leq flow_B < y'_2, & c''_2 : y'_2 \leq flow_B, \\
c_3 : 0 \leq flow_C < y_3, & c'_3 : y_3 \leq flow_C < y'_3, & c''_3 : y'_3 \leq flow_C, \\
c_4 : 0 \leq flow_D < y_4, & c'_4 : y_4 \leq flow_D < y'_4, & c''_4 : y'_4 \leq flow_D, \\
c_5 : 0 \leq z_5 < \varepsilon, & c'_5 : \varepsilon \leq z_5, & \\
c_6 : 0 \leq z_6 < \varepsilon, & c'_6 : \varepsilon \leq z_6, &
\end{array}
$$

where y_i, y'_i, $i = 1, 2, 3, 4$, are system parameters with $y_2 + y_3 + y_4 \leq y_1 < y'_1 \leq y'_2 + y'_3 + y'_4$. States that are bound by the constraints c_i, $i \in [1:6]$, are safe. States bound by c'_i, $i \in [1:4]$ are critical and require an action to reduce flows: $flow_A$, $flow_B, flow_C$ and $flow_D$ should be reduced proportionately to the levels of the constraints c_i, $i \in [1:5]$, while maintaining c_6. Finally, states for which one of $c''_1, c''_2, c''_3, c''_4, c'_5, c'_6$ holds are terminal. When c''_1, c''_2, c''_3, or c''_4 hold, the flow in one of the subnetworks of the pipeline grid exceeds the safety levels. When c'_5 holds then the pipeline grid has a leakage that exceeds the safety levels. When c'_6 holds, the flow of natural gas to Ukraine exceeds the allowed levels (contractual allocation). If the system transitions to a terminal state, then it will shut down with all flows reduced to zero.

The security specifications SEC require that Ukraine should not have access to the values of the flows to South EU and North EU. Also, that Ukraine should be able to verify that it gets its correct allocation of natural gas.

Verification with privacy. Several cryptographic mechanisms can be used to support an application in which one party (Ukraine) can verify the correctness of a particular value (its gas allocation) without getting any additional information about other component values (the gas flows to South EU and North EU).

Clearly Ukraine may get such information by using covert channels, *e.g.,* by accessing the pipelines directly, or accounts/receipts and payments made by South EU and North EU, if these are available. Our threat model does not address security aspects that are not part of the security specifications, and assumes that the RU agreement protocol is based only on readings taken at FCS_B, FCS_C and FCS_D. However, if covert channels are an issue, then the system vulnerabilities must take this feature into account—this extends the scope of an $D_f(\tau)$-adversary, and $D_f(\tau)$-tolerance requires additional protection.[4]

We now describe a cryptographic protocol that can be used by Ukraine to verify the correctness of flows while providing privacy to Russia. The security of this protocol reduces to the Decision Diffie-Hellman (DDH) assumption.

DEFINITION 2 *The DDH-assumption.* Let G_q be a cyclic group of prime order q with generator g. The DDH-assumption concerns the indistinguishability of tuples of type $\langle g, g^x, g^y, g^{xy}\rangle$, $0 \leq x, y < q$, called *DH-tuples*, from general tuples $\langle g, g^x, g^y, g^z\rangle$, $0 \leq x, y, z < q$. Let D_0 be the set of DH-tuples and D_1 the set of non-DH tuples (with $z \neq xy \bmod q$). A *DDH-distinguisher* $\mathcal{D}$ is a probabilistic polynomial-time algorithm (in the length $|q|$ of q) that on input a tuple $T \in D_i$, i a random bit, will predict its type with probability better than 1/2. More specifically, there is a constant $\alpha > 0$, such that for sufficiently large q: on input a tuple T selected uniformly from $\mathcal{D}_i$, i a random bit,

$$\mathsf{Pr}[\mathcal{D}(T) = \mathsf{type}(T) = i \mid T \in D_i] > \frac{1}{2} + |q|^{-\alpha},\ i \in \{0,1\}$$

(here $|q|^{-\alpha}$ is a *non-negligible* quantity). The DDH-assumption is that for some families of groups G_q (including the one considered below) there is no DDH-distinguisher. For more details see [9].

The Flow Verification protocol uses a family of multiplicative integer groups $Z_p^*(\cdot)$ whose modulus is a 'safe' prime p, that is $p = 2q+1$, where q is a prime. Let $g \in Z_p$ have order q and G_q be the subgroup generated by g. Set $b = flow_B$, $c = flow_C$, $d = flow_D$. We shall assume that b, c are rounded to an integer value and that $2b + c << q$.

A Flow Verification protocol for the RU grid

FCS$_B$: Read flow b; select t_b uniformly from Z_q; compute $y_b = g^{\,2bt_b}$.
Send to FCS$_C$: y_b.

FCS$_C$: Read flow c and message y_b; select s_c, t_c uniformly from Z_q;
Compute $x_c = g^{s_c}$, $y_c = {y_b}^{t_c} = g^{2bt_bt_c}$. Send to FCS$_B$: y_c.

FCS$_B$: Read message y_c; compute $z_b = {y_c}^{{t_b}^{-1}} = g^{\,2bt_c}$.
Send to FCS$_C$: z_b.

FCS$_C$: Read z_b; compute $z_c = z_b^{\,t_c^{-1}\cdot s_c} \cdot {x_c}^c = g^{\,(2b+c)s_c}$.
Send to FCS$_D$: (x_c, z_c).

FCS$_D$: Read flow d and (x_c, z_c); compute $z_d = x_c^{20d}$.
If $z_d = z_c$ then send to the verifier: valid.

This protocol captures correctness[5] because: $20d = 2b + c$. It uses a *one-way homomorphic function*, whose security reduces to the Decision Diffie-Hellman (DDH) assumption, as we shall see.

DEFINITION 3 *One-way homomorphic functions.* A mapping $F : G \rightarrow H$ from a group $G(+)$ to a group $H(\cdot)$ is a *one-way homomorphism* if:

a) F *is one-way:* that is, it is infeasible for a probabilistic polynomial-time algorithm to invert any $y = F(x)$.

b) $F(x + y) = F(x) \cdot F(y)$, for all $x, y \in G$.

In the Flow Verification protocol the Flow Controllers FCS_B and FCS_C generate the proof. The verifier (Ukraine) employs FCS_D to verify the proof.

We shall assume that the flow controllers of the RU grid are *tamper-resistant*, that FCS_B and FCS_C are managed by Russia, and that FCS_D is managed by Ukraine. That is, even though all three FCSs are located in Ukraine, Ukraine has physical access only to FCS_D, with Russia having access to FCS_A, FCS_B and FCS_C. Also that the embedded programmable logic controllers (PLC) are trusted and autonomous. The components of the FCS can be checked by all parties concerned[6] prior to deployment. We shall also assume that the communication channels between the FCS are reliable and authenticated; this can be achieved by employing redundancy and using cryptographic authentication mechanisms: either Message Authentication Codes (MAC) or digital signatures. Digital signatures must be used for validation. The communication can be over fixed lines, or wireless.

The $D_f(\tau)$-adversary can be any party other than the prover. For our application the adversary is an insider (possibly Ukraine) who knows the value of $(2b + c)$ (this should be $20d$, where d is the amount of natural gas allocated to Ukraine). The goal of the adversary is to undermine the privacy of the flows b,c.

THEOREM 4 *Suppose that:*

1 FCS_A, FCS_B, FCS_C *and* FCS_D *are tamper-resistant;*

2 *The communication channels linking* FCS_A, FCS_B, FCS_C *and* FCS_D *are reliable and authenticated.*

Then the RU pipeline grid will tolerate an $D_f(\tau)$*-adversary.*

PROOF. The first requirement implies that the adversary cannot access the inner state of the FCS (*e.g.,* the values of b, c or the randomness

t_b, s_c, t_c used to compute their outputs). The second that transmissions are reliable and the origin of messages can be established.

The embedded programmable logic controllers of the FCS can be designed to enforce $D_f(\tau)$-tolerance since we are assuming that: (i) they are not faulty, (ii) their communication channels are trusted, and (iii) the system is autonomous. Insider threats on the FCS are thwarted because the system is autonomous with tamper-proof components. □

THEOREM 5 *Suppose the RU pipeline grid is $D_f(\tau)$-tolerant, and that:*

1 There are no covert channels that leak the values b, c.

2 The DDH-assumption holds.

Then the Flow Verification protocol is correct and provides privacy for the flows b, c against an eavesdropping $D_f(\tau)$-adversary who knows the value of the flow d.

PROOF. The first assumption states that the $D_f(\tau)$-adversary cannot find the values of b, c by using some other means, external to the protocol, *e.g.*, by accessing directly the pipelines, or monitoring the EU gas consumption/payments. Correctness follows from the fact that $20d = 2b+c$.

For the privacy of b, c, suppose that an eavesdropping $D_f(\tau)$-adversary $\mathcal{E}$ can access the values

$$g^{2bt_b},\ g^{2bt_c}, g^{2bt_bt_c},\ \text{ and } g^{s_c},\ g^{(2b+c)s_c}$$

of the communication channels of the RU-grid. Since we are assuming that $\mathcal{E}$ knows the value of d, and $20d = 2b+c$, the last two values do not contribute any additional knowledge regarding the values of b and c. To prove the privacy of b in the presence of $\mathcal{E}$ we consider an experiment $\mathsf{Priv}_{\mathcal{E}}^{\mathsf{eav}}$ in which $\mathcal{E}$ chooses two values of b: b_0, b_1, and is then given the obfuscated tuple

$$T_{b_i} = \langle g, g^{2b_it_b}, g^{2b_it_c}, g^{2b_it_bt_c}\rangle$$

of one of these, where i is random uniform bit. In this experiment the adversary $\mathcal{E}$ (a probabilistic polynomial-time algorithm) must find which one of b_0, b_1 was used in T_{b_i}.[7] Of course $\mathcal{E}$ can toss a coin to guess which one was encrypted. He will succeed with probability $1/2$. Suppose that $\mathcal{E}$ can find the correct value b_i with probability $1/2+\varepsilon$, $\varepsilon = \varepsilon(|q|)$. We shall show that ε is negligible (in $|q|$) by reducing $\mathcal{E}$ to a DDH-distinguisher $\mathcal{D}$.

Let $T = \langle g, g^x, g^y, g^z\rangle$ be the G_q-tuple input to the distinguisher. $\mathcal{D}$ must decide if this is a DH-tuple (that is, if $z = xy \bmod q$), or not. For

this purpose $\mathcal{D}$ queries $\mathcal{E}$ for two values b_0, b_1 and then computes the tuple

$$T'_{b_i} = \langle g, g^{2b_i x}, g^{2b_i y}, g^{2b_i z} \rangle,$$

where i is a random bit. The distinguisher $\mathcal{D}$ gives T'_{b_i} to the adversary $\mathcal{E}$ in the experiment $\mathsf{Priv}^{\mathsf{eav}}_{\mathcal{E}}$, instead of T_{b_i}. If $\mathcal{E}$ predicts that the value b_i was used in the computation of T'_{b_i} then $\mathcal{D}$'s prediction is that T is a DH-tuple. $\mathcal{D}$ outputs 1. Otherwise $\mathcal{D}$ tosses a coin, and bases its prediction on the outcome of the toss (0 or 1). It is easy to see that the probability that $\mathcal{D}$ will distinguish DH-tuples (output 1) is $1/2 + \varepsilon/2$, since $\mathcal{E}$ will succeed with probability ε whenever T is a DH-tuple. Then by the DDH-assumption, $\varepsilon/2$ and hence ε must be negligible (in $|q|$). This completes the proof. □

The Flow Verification protocol is a proof that the (cyber) equation: $20d - 2b - c = 0$ holds, whereas for correctness we have to show that the (physical) inequality: $0 \leq 20d - 2b - c < \varepsilon$ holds. For this particular application there is a simple fix, however in general using a cyber mechanism (cryptography) to secure a physical system may be inadequate, and we may have to use hybrid security mechanisms.

To show that the proof is valid we first sandbox the Flow Verification protocol to separate it from the $D_f(\tau)$-tolerance supporting mechanisms. We then calculate the value of flows by using a unit of measurement for which: $1/4$ unit $> \varepsilon$. We take integer values and map these to Z_q. For example, if the flow measurement is x, it is first reduced by using a new measurement unit to get x' units, and then it is reduced to its integer value $x'' = \lceil x' \rceil$ in Z_q. This approach is good enough for applications in which the fluctuations in measured values are small. Observe that the exact flow values x, as measured at the FCSs, are used to prove $D_f(\tau)$-tolerance.

A change of flow in the flow controller FCS_A will only register at one of the flow controllers FCS_B, FCS_C, or FCS_D, at a later time [41]. To deal with time dependencies of flows, the values of $flow_A$, $flow_B$, $flow_C$, $flow_D$ are time-stamped, and when verifying the values of flow allocations such delays should be taken into account.

4. Related work

In a CPS, distributed computing components interact with the physical environment. Several approaches have been proposed for modeling a CPS: A *hybrid automaton* [2, 28, 39] is a formal model that combines finite state transition systems with discrete variables (whose values capture the state of the modeled discrete or cyber components) and continuous variables (whose values capture the state of the modeled continuous

or physical components). In another related formalism, *timed automata* [3, 31] can model timing properties of CPS. Such machines are finite automata with a finite set of real-valued clocks. They accept timed words, *i.e.*, infinite sequences in which a real-time of occurrence is associated with each symbol.
Hybrid process algebras [5, 18] are a powerful tool used for reasoning about physical systems and provide techniques for analysis and verification of security protocols for hybrid automata. *Bond graphs* [46] are used to synthesize mixed component systems, with electrical, mechanical, hydraulic, thermal and more generally, physical components. Bond graphs are domain independent, allow free composition, and allow efficient analysis and classification of models, permitting rapid determination of various types of feasibility or acceptability of candidate designs. *Genetic programming* [34] is an evolutionary algorithm-based methodology inspired by biological evolution. It is a powerful tool for finding computer programs that perform a user-defined task. When combined with bond graphs it provides for better synthesis of complex mixed component systems. *Hybrid bond graphs* [44] combine bond graphs with hybrid automata to provide a uniform, physics-based formal model that incorporates controlled and autonomous mode changes as idealized switching functions.

Security and survivability goals, threats and attacks on CPS control systems, as well as proactive/reactive mechanisms for robust distributed control and distributed consensus in the presence of deception and DoS adversaries are summarized in [14, 13]. A survey of vulnerabilities, failures and attacks on real-time distributed control systems, as well as of mitigation recovery strategies is given in [33]. A taxonomy of attacks against energy control systems was also given in [24]. Data replay threats on control systems are studied and formulated in [43]. A comprehensive (though, informal) threat model and a taxonomy of attacks against sensor networks in SCADA systems was given in [15], while an emphasis on monitoring and intrusion/anomaly detection methodologies and automatic response for control systems, as well as a formalism of the anomaly detection problem is given on [13]. In [13] risk assessment formalisms are proposed for measuring the possible damages caused by cyber attacks on control systems.

In [30] failures and fault tolerance in distributed CPS are modeled, where such CPS are modeled as distributed algorithms executed by a set of agents and the continuous dynamics of the CPS are abstracted as discrete transitions. An informal attack model for energy control systems is given in [24], where attacks are related to the vulnerabilities they exploit and the damages they cause. Finite state machine models

based on *Petri nets* have also be proposed to describe cyber attacks [51]. Other attack models also include *attacks trees* [45], where the root node denotes the goal of an attacker and a path from leaf nodes to the root node denotes an attack instance, *i.e.*, the steps for completing the attack [50]; attack trees are criticized in [12]. A model using *graph theory* for expressing control system failures and attacks is also given in [12]. In [16] a language for modeling multistep attack scenarios on process control systems was proposed, enabling correlation engines to use these models to recognize attack scenarios.

In another realm, *stochastic* approaches were initially proposed for modeling the different probabilities with which failures occur in distributed computing systems [4]. *Game theoretic* techniques and formalisms for modeling attacks and defense strategies in CPS were given in [40]. There, the game is between an attacker and the defender of a CPS system, where the attacker tries to disrupt either the cyber or the physical system components. Finally, access control and information flow-based policies for CPS security are analyzed in [1, 26], while in [26] a framework to enforce information flow policies in CPS in order to obfuscate the observable effects of a system is presented.

5. Conclusion

We proposed a threat framework for cyber-physical systems. This is based on the traditional Byzantine paradigm for cryptographic security in which the basic security features and requirements as specified by the security policies are used to identify system vulnerabilities. This model allows for a formal analysis and a security proof using existing cryptographic methodologies.

Notes

1. We are assuming that only a countable number of events are related to the execution of $\mathcal{A}$, so their start time is a sparse subset of the *real-time* set (the positive real numbers).

2. An insider may only be able to access system software while it is serviced/upgraded.

3. For this model the system can easily recover from a shutdown.

4. The challenge of preventing covert channels should not be underestimated, particularly in cases where it is possible to collect information leaked from third parties (*e.g.,* through payments made). The issue here is that such information cannot be used to violate the treaty (though it may provide side information). A similar issue, but strategic, is discussed in Footnote 6.

5. The values of the flows must be securely linked to the time and location of their reading; time-stamps should be included in all messages.

6. The Strategic Arms Limitation Treaty SALT II between the United States and the Soviet Union (1977–1979) sought to curtail the number of Inter Continental Ballistic Missiles (ICBM) to 2,250 on each sides. This would involve installing tamper-resistant sensor control units in the ICBM silos to detect the presence of missiles. The sensors were to be used to verify the number of deployed ICBM. Both parties would have access to this information,

but to no other information, particularly regarding the location of the responding silos [47, 19, 10].

7. Indistinguishability of obfuscated data by a polynomial-time adversary captures a strong aspect of privacy, and is the basis for semantic security.

References

[1] R. Akella, H. Tang, and B. McMillin, Analysis of information flow security in cyber-physical systems, *International Journal of Critical Infrastructure Protection*, vol. 3(3-4), pp. 157–173, 2010.

[2] R. Alur, C. Courcoubetis, T. Henzinger, and P. Ho, Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems, *Hybrid Systems*, LNCS vol. 736, pp. 209–229, Springer-Berlin Heidelberg, Germany, 1992.

[3] R. Alur and D. Dill, A theory of timed automata, *Theoretical Computer Science*, vol. 126(2), pp. 183–235, 1994.

[4] Ö. Babaoğlu, On the reliability of consensus-based fault-tolerant distributed computing systems, *ACM Transactions on Computer Systems*, vol. 5(4), pp. 394–416, 1987.

[5] J. Baeten, B. Beek, P. Cuijpers, M. Reniers, J. Rooda, R. Schiffelers, and R. Theunissen, Model-based engineering of embedded systems using the hybrid process algebra Chi, *Electronic Notes in Theoretical Computer Science*, vol. 209, pp. 21–53, 2008.

[6] S. Bengio, G. Brassard, Y. Desmedt, C. Goutier, and J. Quisquater, Secure implementations of identification systems, *Journal of Cryptology*, vol. 4(3), pp. 175–183, 1991.

[7] J. Bengtsson and W. Yi, Timed Automata: Semantics, Algorithms and Tools, *Lectures on Concurrency and Petri Nets*, LNCS vol. 3098, pp. 87–124, Springer-Berlin Heidelberg, Germany, 2003.

[8] M. Blanke, M. Kinnaert, J. Schröder, and J. Lunze, *Diagnosis and fault-tolerant control*. Springer-Verlag Berlin Heidelberg, Germany, 2006.

[9] D. Boneh, The Decision Diffie-Hellman Problem, *Proceedings of the Third International Symposium on Algorithmic Number Theory*, pp. 48–63, 1998.

[10] M. Burmester, Y. Desmedt, T. Itoh, K. Sakurai, H. Shizuya, and M. Yung, A progress report on subliminal-free channels, *Proceedings of the First International Workshop on Information Hiding*, pp. 157–168, 1996.

[11] M. Burmester, T. Le, B. Medeiros, and G. Tsudik, Universally composable RFID identification and authentication protocols, *ACM*

Transactions on Information and System Security, vol. 12(4), pp. 1–33, 2009.

[12] J. Butts, M. Rice, and S. Shenoi, Modeling control system failures and attacks–the Waterloo campaign to oil pipelines, *Proceedings of 4th Annual IFIP Working Group 11.10 International Conference on Critical Infrastructure Protection*, pp. 43–62, 2010.

[13] A. Cárdenas, S. Amin, Z. Lin, Y. Huang, C. Huan, and S. Sastry, Attacks against process control systems: risk assessment, detection, and response, *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pp. 355–366, 2011.

[14] A. Cárdenas, S. Amin, and S. Sastry, Secure control: Towards survivable cyber-physical systems, *Proceedings of the 28th IEEE International Conference on Distributed Computing Systems Workshops*, pp. 495–500, 2008.

[15] A. Cárdenas, T. Roosta, and S. Sastry, Rethinking security properties, threat models, and the design space in sensor networks: A case study in SCADA systems, *Ad Hoc Networks*, vol. 7(8), pp. 1434–1447, 2009.

[16] S. Cheung, U. Lindqvist, and M. Fong, Modeling multistep cyber attacks for scenario recognition, *Proceedings of the 3rd DARPA Information Survivability Conference and Exposition*, pp. 284–292, 2003.

[17] E. Chow and J. Elkind, Where East meets West: European gas and Ukrainian reality, *The Washington Quarterly (Center for Strategic and International Studies)*, vol. 32(1), pp. 77-92, 2009.

[18] P. Cuijpers, J. Broenink, and P. Mosterman, Constitutive hybrid processes: a process-algebraic semantics for hybrid bond graphs, *Simulation*, vol. 84(7), pp. 339–358, 2008.

[19] W. Diffie, The national security establishment and the development of public-key cryptography, *Designs, Codes and Cryptography*, vol. 7(2), pp. 9–11, 1995.

[20] D. Dolev, The Byzantine generals strike again, *Journal of Algorithms*, vol. 3(1), pp. 14–30, 1982.

[21] J. Eisenhauer, P. Donnelly, M. Ellis, and M. O´Brien, Roadmap to secure control systems in the energy sector, Energetics Incorporated, Columbia, Maryland, USA (`http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/roadmap.pdf`), 2006.

[22] R. Ellison, D. Fisher, R. Linger, H. Lipson, T. Longstaff, and N. Mead, Survivable network systems: An emerging discipline,

Technical report, CMU/SEI-97-TR-013, Carnegie Mellon, Software Engineering Institute, Pittsburgh, USA, 1997.

[23] N. Falliere, L. Murchu, and E. Chien, W32.stuxnet dossier, version 1.4, *Symantec Security Response* (`http://www.symantec.com`), February 2011.

[24] T. Fleury, H. Khurana, and V. Welch, Towards a taxonomy of attacks against energy control systems, *Proceedings of the 2nd Annual IFIP Working Group 11.10 International Conference on Critical Infrastructure Protection*, pp. 71–85, 2009.

[25] T. Gamage and B. McMillin, Enforcing information flow properties using compensating events, *Proceedings of the 42nd Hawaii International Conference on System Sciences*, pp. 1–7, 2009.

[26] T. Gamage, B. McMillin, and T. Roth, Enforcing information flow security properties in cyber-physical systems: A generalized framework based on compensation, *Proceedings of the 34th Annual IEEE Computer Software and Applications Conference Workshops*, pp. 158–163, 2010.

[27] K. Hamlen, G. Morrisett, and F. Schneider, Computability classes for enforcement mechanisms, *ACM Transactions on Programming Languages and Systems*, vol. 28(1), pp. 175–205, 2006.

[28] T. Henzinger, The theory of hybrid automata, *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science*, pp. 278–292, 1996.

[29] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, Proactive secret sharing or: How to cope with perpetual leakage, *Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '95)*, pp. 339–352, 1995.

[30] T. Johnson, Fault-tolerant distributed cyber-physical systems: Two case studies, Masters Thesis, University of Illinois, Department of Electrical and Computer Engineering, Urbana, USA, 2010.

[31] D. Kaynor, N. Lynch, R. Segala, and F. Vaandrager, The theory of timed I/O automata, *Synthesis Lectures on Distributed Computing Theory*, vol. 1(1), pp. 1–137, 2010.

[32] C. Kim, G. Avoine, F. Koeune, F. Standaert, and O. Pereira, The swiss-knife RFID distance bounding protocol, *Proceedings of the 11th International Conference on Information Security and Cryptology (ISC '08)*, pp. 98–115, 2008.

[33] R. Kisner, W. Manges, T. Mcintyre, J. Nutaro, J. Munro, P. Ewing, M. Howlader, P. Kuruganti, and M. Olama, Cybersecurity through

real-time distributed control systems, Technical report, Oak Ridge National Laboratory (ORNL), Oak Ridge, Tennessee, USA, 2010.

[34] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, London, England, 1992.

[35] L. Lamport, Proving the correctness of multiprocess programs, *IEEE Transactions on Software Engineering*, vol. 3(2), pp. 125–143, 1997.

[36] L. Lamport, Proving possibility properties, *Theoretical Computer Science*, vol. 206(1-2), pp. 341–352, 1998.

[37] L. Lamport, Real-time model checking is really simple, *Proceedings of the 13th Advanced Research Working Conference on Correct Hardware Design and Verification Methods*, pp. 162–175, 2005.

[38] E. Levy, Crossover: Online pests plaguing the offline world, *IEEE Security & Privacy*, vol. 1(6), pp. 71–73, 2003.

[39] N. Lynch, R. Segala, F. Vaandrager, and H. Weinberg, Hybrid I/O automata, *Proceedings of the DIMACS Workshop on Verification and Control of Hybrid Systems*, pp. 496–510, 1995.

[40] C. Ma, N. Rao, and D. Yau, A game theoretic study of attack and defense in cyber-physical systems, *Proceedings of the 1st IEEE International Workshop on Cyber-Physical Networking Systems*, pp. 708 – 713, 2011.

[41] B. McMillin, personal communication, 2012.

[42] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks - revealing the secrets of smart cards*, Springer-Verlag New York, USA, 2007.

[43] Y. Mo and B. Sinopoli, Secure control against replay attacks, *Proceedings of the 47th Annual Allerton Conference on Communication, Control, and Computing*, pp. 911–918, 2009.

[44] I. Roychoudhury, M. Daigle, P. Mosterman, G. Biswas, and X. Koutsoukos, A method for efficient simulation of hybrid bond graphs, *Proceedings of the International Conference on Bond Graph Modeling (ICBGM 2007)*, pp. 177–184, 2007.

[45] B. Schneier, Attack trees, *Dr. Dobbs journal*, vol. 24(12), pp. 21–29, 1999.

[46] K. Seo, Z. Fan, J. Hu, E. Goodman, and R. Rosenberg, Toward an automated design method for multi-domain dynamic systems using bond graph and genetic programming, *Mechatronics*, vol. 13(8–9), pp. 851–885, 2003.

[47] G. Simmons, Personal communication, 1993.

[48] J. Slay and M. Miller, Lessons learned from the Maroochy water breach, *Proceedings of the 1st Annual IFIP Working Group 11.10 International Conference on Critical Infrastructure Protection*, pp. 73–82, 2007.

[49] H. Tang and B. McMillin, Security property violation in CPS through timing, *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops (ICDCS '08)*, pp. 519–524, 2008.

[50] C. Ten, C. Liu, and G. Manimaran, Vulnerability assessment of cybersecurity for SCADA systems, *IEEE Transactions on Power Systems*, vol. 23(4), pp. 1836–1846, 2008.

[51] R. Wu, W. Li, and H. Huang, An attack modeling based on hierarchical colored Petri nets, *Proceedings of the 1st International Conference on Computer and Electrical Engineering (ICCEE '08)*, pp. 918–921, 2008.

[52] K. Xiao, S. Ren, and K. Kwiat, Retrofitting cyber physical systems for survivability through external coordination, *Proceedings of the 41st Hawaii International Conference on Systems Science (HICSS '08)*, pp. 454–466, 2008.