Chapter 11
Resilient wireless sensor networks for cyber-physical systems

Waseem Abbas[*], Aron Laszka[†], and Xenofon Koutsoukos[‡]

Abstract

Owing to their low deployment costs, wireless sensor networks (WSN) may act as a key enabling technology for a variety of spatially distributed cyber-physical system (CPS) applications, ranging from intelligent traffic control to smart grids. However, besides providing tremendous benefits in terms of deployment costs, they also open up new possibilities for malicious attackers, who aim to cause financial losses or physical damage. Since perfectly securing these spatially distributed systems is either impossible or financially unattainable, we need to design them to be resilient to attacks: even if some parts of the system are compromised or unavailable due to the actions of an attacker, the system as a whole must continue to operate with minimal losses. In a CPS, control decisions affecting the physical process depend on the observed data from the sensor network. Any malicious activity in the sensor network can therefore severely impact the physical process, and consequently the overall CPS operations. These factors necessitate a deeper probe into the domain of resilient WSN for CPS. In this chapter, we provide an overview of various dimensions in this field, including objectives of WSN in CPS, attack scenarios and vulnerabilities, the notion of attack resilience in WSN for CPS, and solution approaches towards attaining resilience. We also highlight major challenges, recent developments, and future directions in this area.

1 Introduction

A wireless sensor network (WSN) is a collection of sensor devices organized into a wireless network. Traditionally, wireless sensor networks have been used as cost-effective means of monitoring spatially distributed processes and phenomena. Their potential uses include military applications, such as battlefield surveillance and chemical attack detection, environmental applications, such as forest-fire detection and precision agriculture, and health applications, such as monitoring human physiological data [1].

A cyber-physical system (CPS) is an integrated system of *computational elements* and *physical processes*, in which the physical processes are controlled by the computational elements [2]. Since the computational elements must have reliable information about the evolving state of the physical processes in order to control them, every practical cyber-physical system has to include *sensor devices*. These sensor devices monitor the physical processes, providing the computational elements with information that can be used for various tasks, such as state estimation and fault identification.

[*] Vanderbilt University, Nashville, TN 37212, USA, e-mail: waseem.abbas@vanderbilt.edu

[†] Vanderbilt University, Nashville, TN 37212, USA, e-mail: aron.laszka@vanderbilt.edu

[‡] Vanderbilt University, Nashville, TN 37212, USA, e-mail: xenofon.koutsoukos@vanderbilt.edu

Finally, the output of the computational elements is fed into actuator devices that can influence the physical processes in the desired way, which closes the loop between the physical and cyber parts of the system.

In the case of *spatially distributed physical processes*, however, the sensing task can prove to be challenging, as the sensor devices may need to be deployed over a larger area. For example, in order to provide intelligent traffic control for a whole city, we must have reliable information about the current traffic situation in various parts of the city. In order to have such information, we must deploy a large number of traffic sensors over a vast area. With wired sensors, the cost of deployment could be prohibitively high and, in some cases, it may even be physically or legally impossible. Consequently, wireless sensor networks, whose deployment is much simpler and more cost-effective, may act as a key enabling technology for spatially distributed cyber-physical systems.

The rest of the chapter is organized as follows. In the remainder of this section, we illustrate the role of WSN in the context of CPS along with information-security goals in CPS. In Section 2, various applications of WSN for CPS are stated along with examples. An overview of different attack scenarios and vulnerabilities in WSN along with instances of such attacks in practical networks is provided in Section 3. In Section 4, the notion of attack resilience in WSN is discussed along with the modeling issues and related challenges. Different approaches towards making WSN resilient against attacks, as well as a couple of detailed examples, are presented in Section 5. Finally, some future directions in this field are outlined in Section 6.

1.1 Cyber-physical systems and sensor networks

Monitoring and surveillance applications. Traditional sensor network applications focus on acquiring, transmitting, and fusing data. In these applications, the physical and cyber parts do not form a closed loop, or, in some cases, form a closed loop which includes a human element. See Figure 1 for a simple illustration of the system architecture of such applications.

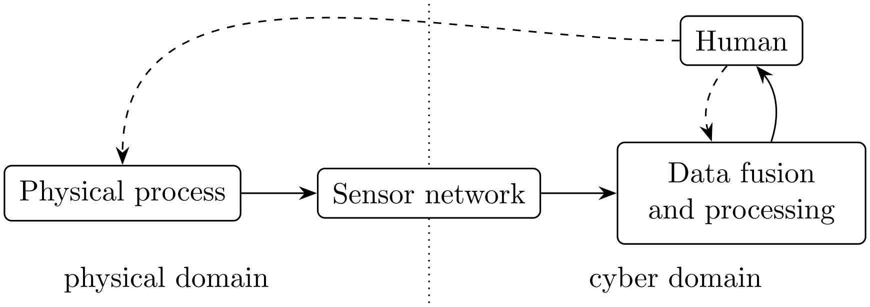

Figure 1: Wireless sensor networks for monitoring and surveillance applications

For example, in a typical habitat-monitoring application [3], sensors measure environmental properties, such as light, temperature, humidity, and barometric pressure, and transmit their data through the sensor network to a gateway. Then, the gateway transmits the data through a transit network to a base station, which provides wide-area network (WAN) connectivity. Finally, the processed data is displayed on a user-friendly interface to scientists.

As another example, in a forest-fire surveillance application [4], sensors collect temperature, humidity, and illumination data, and transmit it through the sensor network to a gateway node. The gateway node then forwards the data to some middleware, which stores the measurements in a database server and calculates forest-fire risk levels from real-time and historical data. Finally, the results are displayed in a web application, and, if a forest fire is detected, alarms are automatically sent to fire stations or nearby residents.

Cyber-physical systems. In cyber-physical systems, on the other hand, the physical processes and computational elements are tightly integrated: physical processes, sensors, controllers, and actuators form a *closed loop.* Note that cyber-physical systems can still be supervised by human operators; however, there is a closed, real-time control loop which does not contain a human element. See Figure 2 for a simple illustration of the architecture of cyber-physical systems using wireless sensor networks.

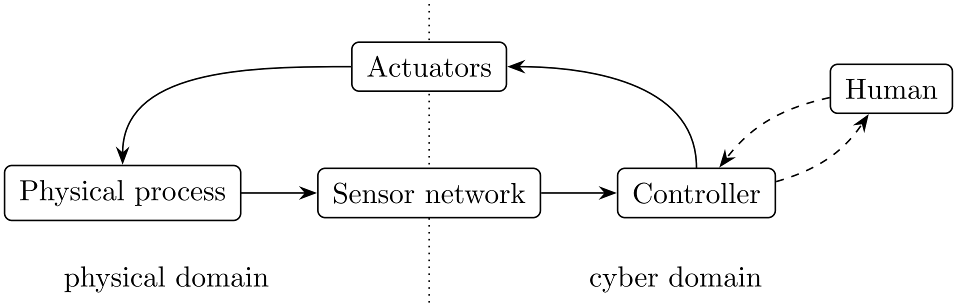

Figure 2: Wireless sensor networks for cyber-physical systems

Since sensor networks in cyber-physical systems are part of closed, real-time control loops, ensuring their security is more critical than in traditional sensor network applications. In a CPS, malicious sensor data will result in incorrect control decisions, which are immediately executed by the actuators. Consequently, an attacker who has compromised a sensor network has some level of control over the physical process and may cause physical damage or financial losses using malicious control. For example, in a smart electric grid, an attacker who can tamper with real-time power consumption data may be able to cause physical damage by simulating a rapid increase in consumption.

Therefore, security is a crucial issue for wireless sensor networks in cyber-physical systems. In the following subsection, we summarize the traditional goals of information security and how they can be applied to cyber-physical systems. For a general overview of WSN in CPS, we refer readers to the other chapters in this book and to the survey of Wu *et al.* [5].

1.2 Information-security goals and cyber-physical systems

Traditionally, the three key goals of information security are *confidentiality*, *integrity*, and *availability* (CIA). For cyber-physical systems, however, these properties are often listed in reverse order (AIC) to emphasize that, in many CPS, availability and integrity requirements have priority over the confidentiality objective [6, 7].

Availability. Availability means ensuring that a system remains in an operable state even if it is attacked. Providing availability for CPS requires ensuring not only that every element of the closed control loop remains operable, but also that the connections between them remain functional. Moreover, most CPS have strict timing constraints, which the system must satisfy even in the case of an attack.

Consequently, providing availability for a CPS entails defending against a variety of attacks. For example, the devices have to be protected against physical attacks, such as physical node destruction and wireless jamming, which can prove to be very challenging for spatially distributed systems. Furthermore, the system has to be protected against cyber attacks as well, in which an attacker exploits some software or protocol vulnerability in order to compromise the system. This can also be very challenging, since the lifetime of many CPS is measured in decades, and deploying software updates is usually a difficult process.

Integrity. Integrity means ensuring that information cannot be modified in an unauthorized and undetected manner. In a CPS, providing integrity means protecting the sensor and control data from modification attacks, which can be achieved by using a message authentication scheme. However, even though there are a number of message authentication schemes which are considered to be secure, providing integrity for a CPS remains challenging. Firstly, many CPS contain legacy devices, which were not designed for security. Consequently, in order to enable message authentication on these devices, their software has to be updated, which can be a difficult process in a CPS. Moreover, these legacy systems have limited computational capabilities and communication bandwidth, which may be insufficient for computing cryptographic functions and transmitting message authentication codes.

Confidentiality. Confidentiality means ensuring that information is accessible only to authorized entities. In CPS, confidentiality is usually viewed as the least important security goal, since sensor and control data rarely constitutes sensitive information. For example, in an intelligent traffic-control system, sensor data usually measures the current traffic situation, while control data adjusts the traffic signals; both of these are easily observable even without any access to the system.

However, in some CPS, sensor data can include sensitive personal information from end-users. In these systems, we must provide privacy, for which providing confidentiality is a necessary condition. For example, privacy issues are a very important concern for smart metering [8–10]. Nevertheless, for most wireless sensor network applications, confidentiality is not a major concern. For this reason, we will focus on availability and integrity in the remainder of this chapter.

2 Objectives of WSN for CPS

Wireless sensor networks are used for a wide variety of applications in CPS. In this section, we give an overview of some of these applications, especially in the context of security and resilience of CPS, and present two specific applications in more detail.

2.1 Main objectives

Surveillance and monitoring. Continuous monitoring and surveillance of physical processes is one of the primary services offered by a WSN. In CPS, this sensory data is then utilized to regulate and control various system modalities in real time. For instance, traffic management through the use of traffic controls, route advisories, and road pricing has been made possible by the observations collected by the sensing devices installed at various locations in traffic-flow networks. Similarly, in the healthcare industry, WSN are in use to enable early detection of clinical deterioration through real-time patient monitoring [11]. The role of WSN in CPS is not just limited to monitoring physical processes by collecting data, but also impacts on the process by being part of the decision-making framework. Smart agriculture, intrusion detection in industrial systems, target tracking in security applications, regulation of necessary conditions in production processes, and energy management systems in buildings are a few instances that have greatly benefited from WSN deployment.

State estimation. In order to dynamically control a system, information about its evolving states is required, which can be acquired through state estimation techniques. Typically, states are estimated based on the noisy measurements collected from various sensing devices and the physical model of the process while minimizing (or maximizing) a certain criterion, such as the weighted least square, the maximum likelihood, and the minimum variance. For instance, in power systems, estimation of system variables such as voltages, relative phasor angles, etc. is required throughout the system nodes for power flow and contingency analysis.

In the case of cyber-physical systems that extend over a large area, reliable state information can be acquired by deploying sensor nodes at various locations within the system, and then processing their observations alongside the dynamical model of the system. For instance, assume that the dynamics or the evolution of states during a process are modeled by a linear time-varying system as

$$x_{k+1} = A_k x_k + B_k u_k + N_k w_k. \tag{1}$$

Here, x_k is a state vector representing n state variables of the system at time instant k, $A_k \in \mathbb{R}^{n\times n}$ is a time-varying matrix that models state transitions, $u_k \in \mathbb{R}^m$ is the control input applied to the system, $B_k \in \mathbb{R}^{n\times m}$ is a control input matrix, and $w_k \in \mathbb{R}^o$ is process noise.

Sensors deployed at various points within the network make observations y_k, which can be represented in a compact way as

$$y_k = C_k x_k + v_k, \tag{2}$$

where $C_k \in \mathbb{R}^{p\times n}$ is a matrix that maps the true state space to the observed space, and $v_k \in \mathbb{R}^p$ is the measurement noise.

The state estimation problem is that, given a measurement sequence and an input sequence up to time k, i.e., $\{y_i, u_i : 0 \leq i \leq k\}$, what is the best possible estimate $\hat{x}_{k+t}$ of x_{k+t}? For $t = 0$, the problem is referred to as the *filtering* problem. *Kalman filtering* (KF) and its extensions are the most widely used tools for state estimation in physical processes using sensor observations (e.g., [12,13]).

Fault detection and localization. Critical infrastructure networks, such as water distribution networks, provide good examples of CPS. These networks are prone to faults and failures, which, if not taken care of in a timely way, might cause tremendous losses. Thus, sensors measuring various system attributes are deployed throughout the network to observe system failures once they occur. For instance, in water networks, flow and pressure sensors are deployed to monitor any abrupt changes in these variables that might correspond to pipe bursts or leakages. If the objective is just to detect the failure without localizing the failure point, then the problem is the *detection* problem. However, if it is desired to uniquely identify each fault, the problem is referred to as the *localization* problem.

In the context of detection and identification of faults, efficient sensor placement is a crucial design problem and has been an active research area. If $\mathcal{S}$ is the set of all available sensors, $c(s)$ is the cost of placing sensor $s \in \mathcal{S}$, and $f : 2^{\mathcal{S}} \to \mathbb{R}$ is a set function that encapsulates the objective of sensor placement, such as detection or identification of specific failures, while assigning a weight to a sensor configuration $S \subseteq \mathcal{S}$, then a typical sensor placement problem can be stated as

$$\max_{S \subseteq \mathcal{S}} f \quad \text{s.t.} \quad \sum_{s \in S} c(s) \leq M. \tag{3}$$

Here M is the allowed budget for the cost of sensor placement. In the context of sensor placement, faults can refer to a wide variety of scenarios. For instance, in water networks, faults can be pipe bursts, leakages, malicious introduction of contaminants, etc.

Short- and long-term forecasting. Integration of demand, generation, and storage is key to efficient – accessible, reliable, and flexible – operation of modern infrastructure networks, such as smart grids [14, 15]. Precise short- and long-term forecasting of generation and demand variables through intelligent and adaptive elements is required to achieve this objective. Using continuous traces of usage patterns that are unique to each premise, along with the database of accumulated observations, thanks to the web of WSN, better design and control of demand response actions can be instantiated. In fact, in many applications, a more proactive approach, which relies on control decisions based on the possible future states of a system, is rather an essential requirement. Moreover, accurate forecasting of system parameters not only leverages planning for growth and changes, but also creates better awareness among customers of their own consumption patterns.

The forecasting problem of state variables in the short and long term is very much related to the state estimation discussed above. Using the system model as given in (1) and (2), if states are estimated for some time in the future, i.e., $\hat{x}_{k+t}$ for $t > 0$, then the problem becomes a forecasting problem, also referred to as the *prediction* problem.

Theft detection and anti-fraud in utility networks. As a result of the integration of WSN with advanced metering infrastructure technologies, high-resolution and fine-grained data is now available to utility service providers such as electricity distribution companies. Using data analytic approaches, utilities are aiming to regulate as well as implement theft detection and anti-fraud mechanisms, which could not be realized through traditional physical checks, to minimize non-technical losses [16, 17].

A simple formulation of theft detection of the quantity under consideration is as follows. Assume that there are two types of consumers, genuine and fraudulent. The goal is to detect fraudulent consumers based on the measurements collected from sensing or metering devices. Let y_t^i be the measurement of the quantity consumed by consumer i at time t, where $t \in \{1, \dots, K\}$, and K is the time interval between billing cycles. Moreover, the probability density functions of y_t^i corresponding to genuine and fraudulent customers are p_g and p_f, respectively. The hypotheses, that consumer i is genuine (H_g), and consumer i is fraudulent (H_f), can then be evaluated using the likelihood ratio test:

$$\frac{\prod_{t=1}^{K} p_g(y_t^i)}{\prod_{t=1}^{K} p_f(y_t^i)} \underset{H_g}{\overset{H_f}{\gtrless}} \gamma. \tag{4}$$

Here, γ reflects the service provider's tradeoff between missing a fraudulent consumer detection and an incorrect detection.

2.2 Examples

In what follows, we discuss two example applications of WSN for CPS.

Smart grid. An electrical (or power) grid is a system for generating, transmitting, distributing, and controlling electricity [18]. Traditionally, these systems are used to carry power in one direction, from a few central generators to a large number of consumers. *Smart grids*, on the other hand, use two-way flows of both electricity and information in order to provide widely distributed and automated energy-delivery networks. Compared to traditional grids, smart grids can deliver power in more efficient ways by utilizing modern information technologies. Furthermore, smart grids can respond to a broad range of events that may occur anywhere in the grid; for example, they may change the power flow automatically when a medium-voltage transformer fails unexpectedly.

Reliable and online information, which can be obtained from sensors monitoring the grid, is crucial to the operation of a smart grid [19]. Since the installation and maintenance costs of wired monitoring systems may be very high, wireless sensors may act as a key enabling technology. Wireless sensor networks can provide a cost-effective sensing and communication platform, enabling the grid to respond to events and changing conditions in a timely and proactive manner.

Intelligent traffic control. Traditionally, traffic signals were controlled by disconnected devices with fixed schedules, which did not have the ability to adapt to the varying traffic situation. However, in recent years, these controllers have been connected both with each other and with sensor devices. This has led to the formation of intelligent traffic-control systems, which provide substantial benefits in terms of wasted time, environmental impact, and public safety [20]. Unfortunately, the installation and maintenance costs of wired sensing systems can be prohibitively high, which hinders large-scale deployment [21]. Since sensor nodes with integrated sensing, computing, and wireless communication capabilities offer much lower installation costs, WSN may revolutionize the field of traffic control.

The direct goal of traffic monitoring is to provide an accurate estimate of the current traffic situation. However, in order to be able to perform proactive, dynamic traffic control, we also need to be able to forecast traffic [22]. In the literature, a large number of models have been proposed for this prediction problem. For example, Xie *et al.* [23] recently proposed to use Gaussian processes and demonstrated their applicability using real-world data.

3 Attacks against sensors

In this section, we present an overview of various attacks that can be mounted against a WSN, which may cause serious damage if successful. The consequences of these attacks may range from deterioration or even complete disruption of services offered by the WSN to complete failure of the physical process whose control depends on the sensory data.

3.1 Types of attacks

In the literature on wireless sensor networks, a multitude of attacks have been studied. We divide these attacks into two main groups: denial-of-service attacks and data falsification attacks.

3.1.1 Denial-of-service (DoS) attacks

Denial-of-service attacks try to break the availability security property of the system. Keeping in view the fact that, in cyber-physical systems, the CIA (confidentiality–security–integrity) security paradigm is often modified to AIC (availability–integrity–confidentiality), DoS attacks in sensor networks, which directly impact availability of services, become a prominent threat to CPS operations. These attacks can be both physical and cyber.

Typically, a classification of various DoS attacks is based on different layers in the protocol stack of the sensor nodes. Details of such a classification can be found in [24–26].

Physical layer attacks. *Wireless jamming* is a typical physical layer attack used by an adversary to put a certain number of nodes in the system out of service by disrupting valid communication between sensor nodes. Various *active* and *reactive* jamming strategies can be used by an adversary for this purpose [27]. Active jamming revolves around the theme of keeping the channel busy for longer periods irrespective of the traffic patterns on the channel, and includes constant jamming, deceptive jamming, and random jamming strategies. Active jamming is relatively easier to detect. Reactive jammers, on the other hand, sense traffic on the communication channel and transmit when the channel is active, and remain idle in the case when the channel is free. Consequently, they become hard to detect. *Physically destroying a sensor* is also a physical layer DoS attack that renders the node out of service. Similarly, *tampering* with the hardware or software configurations of nodes to put them out of service also results in the denial of service.

Link/medium access control (MAC) layer attacks. An adversary might attack the link layer (or media access control (MAC) layer) by introducing malicious collisions, often referred to as *collision attacks*, resulting in repeated retransmissions of the frames associated with various MAC protocols. This not only decreases the network throughput, but also targets sensor nodes' power supplies through *denial-of-sleep* attacks, in which the batteries of nodes are exhausted much earlier than expected due to a large number of retransmissions.

Network and routing layer attacks. Some of the DoS attacks on the network layer involve *spoofing* or altering the routing information to cause routing disruptions, with detrimental consequences on the overall network performance, such as extending or shortening the source routes, increasing end-to-end latency, which is crucial in CPS keeping in view the significant requirement of making decisions in real time, decreasing network throughput, etc. *Black hole attacks* and *sinkhole attacks* are examples in which compromised nodes establish an important role by becoming a part of several routes and then dropping packets through them. Another attack with serious ramifications is a *wormhole attack* [28, 29], in which an attacker receives a message from one point and then replays it from another point in the network after passing it from the source to the new point through a low-latency link (wormhole link). In many protocols, nodes broadcast hello messages to determine one-hop neighbors for routing purposes. An attacker may compromise a node, and through a high-power transmitter may inform other nodes that it is their one-hop neighbor which can provide a superior route to the base station. As a result, many nodes attempt to route their traffic through the compromised node even though the compromised node does not lie within their radio range. This sort of attack is often referred to as a *hello flood attack*.

Transport layer attacks. At the transport layer, which is mainly responsible for maintaining reliable end-to-end connections between nodes, some protocols require nodes at either end of the connection to maintain state. An attacker may bombard a node implementing such a protocol with new connection requests without ever completing them, also known as a *flooding attack*, which might exhaust the connection buffer of the node. In another attack, called a *desynchronization attack*, an attacker disrupts the existing connection by repeatedly spoofing the message to desynchronize the endpoints and cause retransmissions.

Application layer attacks. If sensors are not transmitting observations at fixed intervals, but are rather triggered by some physical activity such as responding to an event, then an adversary can create an attack by *overwhelming* a sensor by artificially generating events. This may result in battery exhaustion and bandwidth consumption. Moreover, on the application layer side, sensors typically utilize simple and not highly sophisticated operating systems to run various applications. As a result, software vulnerabilities of sensor nodes can be exploited by an attacker through *mal-packets* [30] or *sensor worm attacks* [31, 32], in which specially crafted data can exploit memory-related vulnerabilities and use application code in a sensor to further propagate in the network, resulting in malicious behavior and node failures. Propagation of such worms is typically done either by scanning vulnerable devices, or by spreading through topological neighbors [33].

3.1.2 Data falsification attacks

Data falsification attacks try to breach the integrity security property of the system through modifying sensor data or injecting false/fake data. These are typically cyber attacks. An example of such a scenario is a *byzantine failure*, in which different information and falsified data is passed to different nodes from an attacked sensor, which might result in an incorrect state estimation or subjugation of control information.

In many CPS, such as power grids, estimating the state of unknown system variables by analyzing measurements obtained through various metering devices is a crucial process in controlling the overall system operations. An attacker, in the case of access to these meters, may successfully inject malicious measurements, also known as *false data injection attacks* [34, 35], and exploit the configuration of a power system, thus misleading the state estimation process. The false information can include an incorrect observation, incorrect timing information, and an incorrect sender identity. An instance of an attack where an adversary illegitimately claims multiple identities, either by impersonating other nodes or by generating an arbitrary number of node identities, is a *sybil attack* [36] or *node replication attack* [37]. These attacks are particularly effective in thwarting the redundancy mechanisms in sensor networks.

In the context of control of CPS, integrity refers to the trustworthiness or authenticity of the sensors and controllers. By compromising the integrity, an adversary can cause a *deception attack* in which a controller or a node within the network relies on and believes the false information received from one or multiple sensing devices [38]. In a similar context, *stealthy deception attacks* have been studied for the water SCADA (supervisory control and data acquisition) systems in [39, 40]. Another data falsification attack is a *replay attack*, in which an attacker attempts to control the system by compromising a set of sensors, recording their observations for a certain time, and then replaying them while injecting exogenous control inputs [41, 42]. In sensor networks, data collected from various sensing nodes is eventually aggregated, in either a distributed or centralized manner. The control action of the controller is based on the aggregated data received. Thus, if attackers have a high-level knowledge of the aggregation schemes and their parameters, they can conduct a *collusion attack* [43], in which aggregated data is modified due to the false data injected through a set of compromised nodes. This may lead to a control decision that can damage the physical plant, causing financial or physical losses.

3.2 Examples of attacks and vulnerabilities in CPS

Stuxnet. Stuxnet is a computer worm targeting programmable logic controllers (PLCs), which was discovered in June 2010 [44]. The worm is widely regarded as a milestone in cyber security, since it was the first malware that was designed to cause physical damage. Although it has not been confirmed who developed the worm, it is believed that the developers had nation-state support and backing, owing to the highly sophisticated design of the worm, and its usage of an unprecedented four zero-day exploits and two digital certificates stolen from separate well-known companies [45]. To infect computers, the worm is able to propagate through both infected removable drives, such as USB flash drives, and local area networks. These propagation vectors allow it to infect sensitive control systems, which are traditionally supposed to be secured by the “air-gap” (i.e., by not connecting the sensitive systems to the public Internet).

Stuxnet reportedly targeted Iranian uranium-enrichment facilities at Natanz [46]. Once it has infected a computer controlling a PLC that meets specific configuration requirements, the worm performs a man-in-the-middle attack, which fakes industrial process control sensor signals. Next, the worm tries to damage uranium centrifuges by increasing and decreasing their rotor speeds well above and below their normal operating speed, probably in order to cause damaging vibrations. Even though the attack did not result in the total destruction of the centrifuges, it drastically decreased their lifetime: due to the attack, around one-fifth of Iran’s nuclear centrifuges were reportedly destroyed.

Sensys traffic sensors. Sensys Networks is a company that supplies wireless traffic-detection and integrated traffic-data systems, and has deployed more than 50 000 devices in 45 US states and 10 countries. In 2014, a cyber security researcher, Cesar Cerrudo, discovered that the Sensys Networks VDS240 wireless vehicle-detection systems are vulnerable to multiple attacks [47, 48]. These systems comprise magnetic sensors embedded in roadways, which transmit traffic-flow data over a wireless channel to nearby access points or repeaters. These in turn pass the data to traffic signal controllers, allowing intelligent traffic control based on real-time traffic data.

However, the lack of proper security mechanisms allows attackers to tamper with traffic data or compromise the sensor devices. Firstly, the wireless traffic between sensors and access points is unencrypted, and an attacker can intercept and replay traffic data [49]. More importantly, the sensors accept software updates without checking the integrity of the supplied software code, which allows an attacker to compromise the devices. Such compromises enable the attacker to take complete control and send fake data to traffic-control systems. Furthermore, it might also be possible to develop worms that infect vulnerable sensors and then propagate to nearby devices on wireless channels, infecting all sensors in a large area starting from a single device [47].

Even though the vulnerabilities do not allow direct control of traffic signals, an attacker may cause traffic jams and problems at intersections. By supplying fake traffic data, it is possible to influence the timing of traffic lights, to have electronic signs to display incorrect instructions, and to cause ramp meters to allow cars on the freeways slower or faster. Since the discovery of these vulnerabilities, Sensys Networks has released software updates to remediate the vulnerabilities identified in their traffic sensors [49].

Oil-pipeline explosion. In August 2008, the Baku–Tbilisi–Ceyhan oil pipeline in eastern Turkey exploded, which the Turkish government publicly blamed on a mechanical failure [50]. Even though the pipeline was monitored by sensors and cameras, the blast did not trigger any distress signals and the cameras failed to capture the combustion. The complete failure of the alarm systems is very surprising, since critical indicators, such as pressure and flow, were transmitted via both a terrestrial wireless system and backup satellite links to a central control room.

However, according to experts familiar with the incident, the explosion was actually caused by a cyber attack [51]. The attackers allegedly used a vulnerability in the communication software of the cameras to gain entry to the system, which enabled them to access operational controls, without compromising the main control room. By gaining access to smaller industrial control systems at a few valve stations, the attackers were able to shut down the alarms, cut off communications, and super-pressurize the oil in the pipeline. Since no evidence of a physical bomb was found, it is very likely that the explosion was created by the high pressure alone.

Vulnerable traffic signals. Intelligent traffic-control systems not only provide benefits in terms of wasted time and resources, but also provide new opportunities for attackers. While traditional hardware systems were susceptible only to attacks based on direct physical access, systems now may be vulnerable to attacks through wireless interfaces, or even to remote attacks through the Internet. A recent case study by Ghena *et al.* [20] analyzed the security of traffic infrastructure in cooperation with a road agency located in Michigan. The agency operates around a hundred traffic signals, which are all part of the same wireless network, but the signals at every intersection operate independently of the other intersections. The study found three major weaknesses in the road agency's traffic infrastructure: lack of encryption for the network, lack of secure authentication due to the use of default usernames and passwords on the devices, and vulnerability to known exploits. Owing to the hardware-based failsafes, an attacker cannot put traffic lights into an unsafe configuration, but may be able to cause disastrous traffic congestion.

4 Notion of attack resilience

In CPS, as a result of the integration of the cyber and physical domains, physical processes are directly influenced by the integration of IT systems comprising sensing elements and communication networks. Consequently, the physical dynamics of the system can be manipulated through cyber means. This is advantageous, on the one hand, as it stipulates enhanced monitoring and efficient control. However, on the other hand, it raises new threat channels, against which the system needs to be secured. To account for modeling uncertainties and physical disturbances in the control of physical processes, several tools have been developed in classical control, including robust control and stochastic control. However, the added dimension of cyber domain vulnerabilities, which are capable of sabotaging the overall operations and control of CPS, pose new challenges and make these tools insufficient. Thus, there is an imminent need for a systematic and thorough analysis of security and resilience against faults, failures, and adversarial actions in CPS.

4.1 Security and resilience

Cyber security (or computer security) is traditionally concerned with preventing attacks from succeeding. *Security* can be attributed as the ability of a system to avert attacks and remain protected against malicious behavior. *Resilience*, on the other hand, relates to the behavior of a system once its security has been compromised. More specifically, resilience can be thought of as a system's ability to recover online in case of an attack and continue to operate with minimal and tolerable disruption, i.e., to gracefully degrade its operations. Note that security is a *pre-event* property, whereas resilience is a *post-event* attribute.

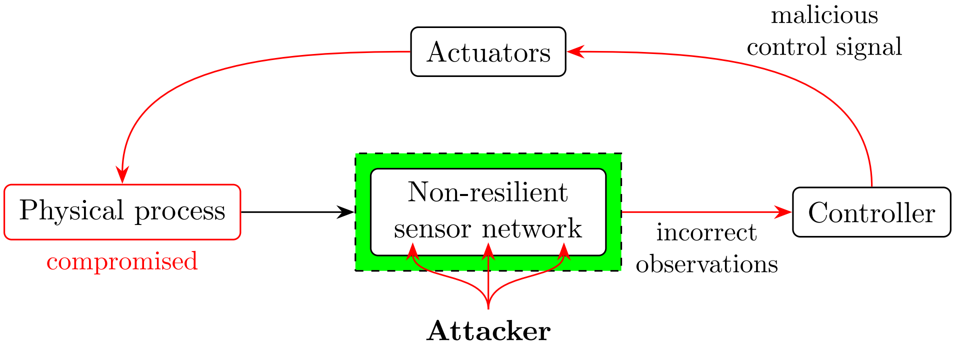

Figure 3: Consequences of attacks against non-resilient WSN in CPS

Since perfect security against all attacks is – in practice – either impossible or financially unattainable, there is always a possibility of a successful attack or malicious intrusion. Recent examples, such as the Stuxnet worm, have shown that determined and resourceful attackers can penetrate even highly secured and secluded systems. In CPS, a successful attack might cause serious physical damage to the system and disrupt services, such as energy supply and water distribution, thereby incurring execrable financial losses. Since control decisions in CPS are made in real time by controllers based on observations received from a WSN, resilience is a critical issue in CPS. Consequently, any compromise or disruption in WSN due to malicious attacks or failures of sensor nodes would severely impact the physical process as illustrated in Figure 3. Thus, *a resilient WSN is imperative for a resilient CPS.*

In light of the above discussion, merely securing WSN against attacks, such as the ones mentioned in Section 3, is not sufficient to ascertain the operational normalcy in CPS. In fact, a resilient design of WSN that maintains the task of transmitting correct sensor data in a timely manner to the controller on the physical side, despite adversarial attacks, is a crucial requirement. In a broader perspective, the resilience property of networks in general, and for WSN in CPS in particular, can be attributed to the following three factors:

- functional correctness of the network (design)
- ability to sustain under reliability failures (faults)
- ability to survive against security failures (attacks).

4.2 Random failures and intentional attacks

In this section, we formulate general problems for resilient WSN designs. As mentioned above, resilience in WSN can be against *random failures*, which correspond to sensor faults and device malfunctions, and *intentional attacks*, which correspond to sensors being attacked by an adversary.

A very important aspect of the resilient design problem is the *tradeoff* between cost and resilience: in practice, making a system more resilient generally entails incurring additional costs. For example, if we achieve resilience through redundancy, we have to deploy additional devices and links, which all have deployment and maintenance costs. As another example, if we achieve resilience by changing the architecture of a system from the most efficient one to a more resilient one, then we incur efficiency loss, which we can model as a form of cost. Consequently, finding the optimal design in practice means striking the right balance between maximizing resilience and minimizing cost.

However, in many cases, we are given a fixed *budget M* for building a system, which means that we can choose only from those designs whose cost does not exceed the budget. Consequently, in these cases, we have to find the most resilient design from the set of designs that satisfy the budget constraint. In other words, we have to solve a *constrained optimization* problem instead of a tradeoff problem.[1] In this chapter, we will formulate the resilient design problem in this way, that is, as a constrained optimization problem based on an exogenous budget value. Note that, given a method for solving the constrained optimization problem, one can solve the tradeoff problem in practice by searching for the budget value that optimizes resilience and cost.

Random failures. First, we discuss failures that are not caused by adversarial action, which we call *random failures*. These include, for example, software crashes due to random faults in static random-access memory (SRAM) and wireless-link failures due to extreme weather conditions. Note that, since our focus is on attack resilience, we disregard systematic failures in this chapter, and present only a simplistic model of random failures. The common characteristic of these failures is that they occur independently of defensive and adversarial actions, i.e., the probability of each event is exogenous to the problem.

Now, we present a simple formulation of the failure-resilient design problem. We let $\mathcal{F}$ denote the set of possible failures in the system, and for each failure $f \in \mathcal{F}$, we let β_f denote the probability of the given failure occurring. Assuming that the failures are independent, we can express the probability of a set F of failures occurring as

$$\Pr[F] = \left(\prod_{f \in F} \beta_f \right) \left(\prod_{f \notin F} (1 - \beta_f) \right).$$

Next, we let $\mathcal{D}$ be the set of possible designs, we denote the cost of design d by $C(d)$, and we denote the loss sustained by a system based on design $d \in \mathcal{D}$ when failures in F occur by $\text{Loss}(d, F)$. Then, if our goal is to minimize the expected amount of losses sustained by the system,

[1] We can also formulate the resilient design problem as the problem of minimizing cost while achieving a threshold level of resilience. Since this formulation is computational-complexity-wise equivalent to having a budget constraint for most problems, we do not discuss it in detail.

the failure-resilient design problem can be formulated as

$$\begin{aligned}
&\operatorname*{argmin}_{d \in \mathcal{D} \,:\, C(d) \leq M} \mathrm{E}[\mathrm{Loss}(d, \cdot)] && (5)\\
&\quad = \operatorname*{argmin}_{d \in \mathcal{D} \,:\, C(d) \leq M} \sum_{F \subseteq \mathcal{F}} \Pr[F] \, \mathrm{Loss}(d, F) && (6)\\
&\quad = \operatorname*{argmin}_{d \in \mathcal{D} \,:\, C(d) \leq M} \sum_{F \subseteq \mathcal{F}} \left(\prod_{f \in F} \beta_f \right) \left(\prod_{f \notin F} (1 - \beta_f) \right) \mathrm{Loss}(d, F). && (7)
\end{aligned}$$

Intentional attacks. Contrary to random failures, intentional attacks never follow an exogenous probability distribution. In fact, adversaries can take into consideration the design of a system and the deployed defensive countermeasures, and strike at the weakest point. Consequently, the occurrence of attacks is endogenous to the model, since they depend on the chosen design, which makes the attack resilience problem more complex.

Similarly to the design options, the adversary's actions are also costly. For example, to launch a physical attack, such as wireless jamming, the adversary may have to buy special equipment and spend effort to carry out the attack. As another example, to launch a cyber attack, the adversary has to discover a security vulnerability in the system, which again requires spending effort or hiring experts. However, in practice, adversaries have limited amounts of resources available for their attacks. Consequently, similarly to the defender's problem, we can model the adversary's decision as a constrained optimization based on an exogenous budget constraint B. Note that, as an alternative, we can also model the adversary's decision as a tradeoff problem.

Finally, we need to formulate the adversary's objective. Here, we will assume a worst-case adversary whose goal is to maximize the losses sustained by the system. Let $\mathcal{A}$ be the set of possible adversarial actions, let the cost of action $a \in \mathcal{A}$ be $c(a)$, and let the loss sustained by the system based on a design d when the adversary carries out the actions in A be denoted by $\mathrm{Loss}(d, A)$. Then, for a given design d, the adversary's decision problem can be formulated as

$$\operatorname*{argmax}_{A \subseteq \mathcal{A} \,:\, \sum_{a \in A} c(a) \leq B} \mathrm{Loss}(d, A), \tag{8}$$

and the attack-resilient design problem is

$$\operatorname*{argmin}_{d \in \mathcal{D} \,:\, C(d) \leq M} \left(\max_{A \subseteq \mathcal{A} \,:\, \sum_{a \in A} c(a) \leq B} \mathrm{Loss}(d, A) \right). \tag{9}$$

4.3 Challenges

In this subsection, we discuss the main challenges in solving attack-resilient problems for WSN as formulated above.

Computational complexity. According to the widely accepted Cobham–Edmonds thesis [52], solving a computational problem is feasible only if we can solve the problem in polynomial time. However, in many attack-resilient CPS problems, the set of possible designs grows exponentially as the size of the problem increases. For instance, in a sensor placement problem, the number of feasible placements is an exponential function of the number of locations at which sensors can be placed. Consequently, the number of designs to choose from quickly becomes “astronomical” as the size of the problem increases. As an example, if we can place at most 100 sensors in 500 possible locations, then the number of feasible subsets is approximately 2×10^{107}, more than the number of atoms in the observable universe, which is only 10^{80}. For such problems, since an exhaustive search is computationally infeasible, we have to find a more intelligent, polynomial-time algorithm.

Unfortunately, many attack-resilient design problems are NP-hard, which means that they cannot be solved in polynomial time (given that P $\neq$ NP, which is a very widely accepted conjecture). To tackle such computationally challenging problems, in practice we have to use approximation and heuristic algorithms. For example, the optimal design of a WSN that is used for pipeline leakage detection in a water distribution network is NP-hard even without an attacker. Let the set of possible sensor locations be $\mathcal{S}$, let the cost of placing a sensor at location $s \in \mathcal{S}$ be $c(s)$, let L be the set of possible leakages (e.g., the set of pipeline sections that may be damaged), and let the set of leakages detected by a sensor at location s be $L_s \subseteq L$. Then, we can show that it is NP-hard to determine whether there exists a placement S that can detect all leakages and whose cost $\sum_{s \in S} c(s)$ does not exceed our budget M.

We prove NP-hardness by reducing a well-known NP-hard problem, the set cover problem, to the above decision problem. Given a base set U, a set $\mathcal{E}$ of subsets of U, and a number k, the set cover problem is to determine whether there exists a set $\mathcal{C} \subset \mathcal{E}$ such that $\mathcal{C}$ covers all elements of U (i.e., for every $u \in U$, there is an $E \in \mathcal{C}$ such that $u \in E$) and $|\mathcal{C}| \leq k$. We can reduce an instance of the set cover problem to our decision problem as follows. Let the set of possible leakages L be U, let the set of possible sensor locations $\mathcal{S}$ be $\mathcal{E}$, let the set of leakages L_s detected by a sensor at location s be the corresponding subset from $\mathcal{E}$, let the cost of placement for all locations be $c(s) = 1$, and let the budget M be k. Then, it is easy to see that a feasible sensor placement detecting all leakages exists if and only if there exists a set cover.

Estimating model parameters. In order to apply theoretical results to a real-world system, we must be able to map the parameters of the model to real-world data, which can prove to be challenging. Firstly, we need to be able to quantify the potential losses that arise from the various security incidents that might happen. These values can be quantified either as financial damage due to physical losses (e.g., increase in operating costs due to suboptimal control or cost of replacing damaged devices) or as liability/penalty to be paid (which can be estimated from past settlements). Note that, if we formulate the design problem using a budget constraint, we only need to be able to compare the possible outcomes to each other; hence, we have to estimate only the relative losses for the various incidents.

Secondly, we need to be able to estimate the cost of the attacker’s actions, and, if we model the attacker’s decision using a budget constraint, the value of this budget constraint as well. The latter can be very challenging. If we underestimate the attacker’s budget, our system will not be resilient to more powerful attacks and may suffer intolerable losses. On the other hand, if we overestimate the attacker’s budget, the resilience of our system against the actual attacks will be suboptimal. We can elude this problem by modeling the attacker’s decision as a tradeoff problem; however, in this case, we need to be able to compare the cost of the attacker’s actions to the potential loss values.

Finally, we need to be able to estimate the budget and the cost of the various design choices and defensive countermeasures. This is a relative easy task, since the budget (if there is one) is given, and we can assume that a defender knows the various deployment and maintenance costs of its system.

Modeling attackers. In order to design optimal attack-resilient systems, we must be able to model all the possible actions that an attacker may take. For physical attacks, this task is generally tractable, as an attacker is always bound by the laws of physics. For cyber attacks, however, establishing bounds on what an attacker might do can be more challenging. Cyber attacks are usually possible due to vulnerabilities that were unintentionally introduced into the software, that is, cyber attacks usually happen because one or more of our assumptions on how a system may be used have already failed. Consequently, we have to be careful about what further assumptions we make. We can overcome this paradox by using more abstract, higher-level attack models. For example, the attacker's actions can correspond to compromising a node or the integrity of a link, abstracting away from the specific types of exploits that an attacker might use and the specific access rights that it might gain. However, such higher-level models increase the difficulty of estimating the cost and success probability of the attacker's actions.

So far, we have assumed rational attackers with complete information. On the one hand, this is a very robust model security-wise, since it is based on pessimistic assumptions, which is in accordance with the principles of security (e.g., Kerckhoffs' principle [53]). On the other hand, these assumptions may be too strong in practice, where attacks are planned by human adversaries, who have bounded rationality and limited information. Modeling human adversaries' irrationality, biased decisions and perceptions, limited observations and computational capabilities, etc. using mathematical tools can be very challenging. These "imperfections" may be modeled most naturally using the game theory nomenclature – see, for example, the work of Pita *et al.* [54] and Freund *et al.* [55].

5 Attack-resilient design

In Section 4, we discussed the notion of attack resilience in WSN for CPS, and presented various models and challenges in solving attack-resilient problems for WSN. In this section, we discuss different approaches towards attack-resilient design of WSN, and also present two example problems in detail.

5.1 Approaches for attack resilience

Attack resilience can be achieved in a WSN using multiple approaches, ranging from resilient sensor placement to resilient data-aggregation algorithms. Here, we discuss three main approaches, noting that one may find many others in the literature. For example, attack resilience can be achieved by using resilient routing algorithms [56–58].

Sensor placement. We can mitigate attacks that compromise or incapacitate individual nodes via attack-resilient sensor placement, i.e., by choosing the locations of the sensor nodes so that the resulting WSN is attack-resilient. Such attacks have multiple adversarial effects, which we must take into account when planning node placement. Firstly, sensors impaired by an attack will no longer supply correct observational data, which can affect the controller and, consequently, the physical process of the CPS. To moderate losses arising from missing observations, we have to place sensors in such a way that the remaining sensors can supply adequate information in the case of an attack, which we call *resilient coverage*. Secondly, as the impaired nodes may have been used for data forwarding, the connectivity of the sensor network can also decrease, which necessitates designing the network topology to be resilient. Here, we will focus on resilient coverage, and we will discuss attack-resilient network topologies later. For a general survey on node placement in wireless sensor networks, we refer the reader to the survey of Younis and Akkaya [59].

The most widely used objective for resilient coverage is k-coverage: we say that a sensor network has k-coverage if every location is within the sensing range of at least k sensors [60]. In other words, a sensor network that has k-coverage can withstand attacks that impair at most $k-1$ sensor nodes. The k-coverage metric is widely used because it has the appeal of simplicity as an objective. For example, Wang and Tseng [61] study placement schemes for providing k-coverage for an area with the minimum number of nodes.

However, k-coverage may be too simplistic for practical applications. The main disadvantage of k-coverage is that it treats coverage as a binary property, i.e., as something that either is or is not provided. In practice, however, the amount of information gathered can decrease more gradually as more and more sensor nodes are impaired. For example, having multiple observations for the same location may provide more accurate measurements, which may lead to more efficient control. As another example, when we cannot observe a location, we may be able to infer missing values from observations taken at nearby locations, which allows us to retain control of the physical processes despite the attack. Consequently, for many applications, more fine-grained metrics are necessary. For instance, Dhillon and Chakrabarty [62] study coverage for surveillance applications using a probabilistic objective, which is based on the uncertainty associated with sensor detections.

For optimal resilient coverage for CPS, we must go one step further and incorporate *how sensor data is used* into our objective function. In other words, we have to place sensor nodes in such a way that, when some are impaired by an attack, the remaining nodes can provide data that will result in acceptable control decisions.

Network topology. Designing resilient network topologies for sensor networks, capable of maintaining a certain level of some global performance measure in the presence of adversarial attacks, is crucial to the overall resilient CPS design. In the case of node or edge removal attacks, which can be random or strategic, and may correspond to node destruction, exhaustion, or jamming, the objective is to design and control the network structure to preserve structural properties of the graph to a reasonable extent. The basic premise of resilient network topology design is as follows (e.g., [63–66]). A set of sensor network performance measures, such as energy consumption, connectivity, distance or communication delay between nodes, throughput, etc., is considered. These performance goals are then translated into network topology-based parameters, such as node or vertex connectivity, persistence, centrality-based measures, distance-based measures, Kirchhoff index, etc. Using tools such as combinatorial optimization, network topology is then optimized with respect to the network-based measures under node or edge removal conditions. An important consideration in attack-resilient sensor network design for CPS, especially in selecting the sensor network performance-based measure and its translation into the network topology-based metric, is to incorporate the consequences of the sensor network being in a closed loop, as shown in Figure 2.

Data aggregation. In general, sensors are spatially distributed in a WSN, collecting data from many vantage points, which is then aggregated, such as average, median, maximum, minimum, etc., rather in a distributed manner. In a CPS, in-network computations are performed and control decisions are taken in real time based on the aggregated data observed from the sensor network. Several aggregation algorithms for sensor networks have been reported in the literature. However, since the majority of these schemes were not originally designed to be resilient against malicious attacks, such as data falsification attacks, data aggregation results can be easily manipulated even if a small subset of nodes are compromised. For instance, in a simple linear consensus algorithm designed to ensure that all sensors converge to an average of their initial observations, a single misbehaving node can result either in convergence to a value far from the average, which might be a significant issue in safety-critical processes, or in no convergence at all. In this direction, designing *resilient* data aggregation algorithms that can withstand the compromise of a subset of nodes and maintain a certain notion of correctness in computations is a way to go [67].

Some of the approaches to achieve resilient data aggregation include *preprocessing* sensory data before applying aggregation schemes, such as selecting a subset of received observations for the algorithm rather than using all of the available data (e.g., [68, 69]). Another approach is to *detect* if received data is from a valid sensor or a compromised node before applying an aggregation scheme (e.g., [70–72]). In yet another approach, the idea is to relate the resilience of the data aggregation algorithm against a certain number of malicious attacks to the underlying network topology (e.g., [68, 69]). For instance, if some connectivity conditions, say $\mathcal{C}$, are satisfied in the underlying graph structure, then the data aggregation algorithm will give correct results even if a certain number of nodes, say $\mathcal{A}$, are compromised, where $\mathcal{A}$ depends on $\mathcal{C}$.

5.2 Examples

Resilient sensor placement for prediction. Now, we discuss the resilient sensor placement problem for prediction and forecasting applications in more detail.

We let $\mathcal{S}$ denote the set of locations at which sensors can be placed, and let Y denote the target variable that we have to predict. For example, in a traffic-control CPS, the set $\mathcal{S}$ can model the road segments on which traffic-flow sensors can be placed, and variable Y can represent the future traffic situation. The design choice in this problem is to select a subset $S \subseteq \mathcal{S}$ of locations at which sensors are placed, such that $|S| \leq M$, and the goal is to predict Y as accurately as possible. We formalize this goal as minimizing the posterior variance $\sigma^2_{Y|S}$ of the target variable Y given observations received from the sensors placed at S. Quantifying how inaccurate our predictions are using variance is reasonable, since minimizing variance minimizes the mean-squared error of predictions, and variance is also logarithmically proportional to the uncertainty of the target variable measured in entropy. Then, without an attacker, the sensor placement problem can be expressed as

$$\operatorname*{argmin}_{S \subseteq \mathcal{S} \,:\, |S| \leq M} \sigma^2_{Y|S}. \tag{10}$$

Next, we introduce an adversary who can launch denial-of-service type attacks that impair sensor nodes. We assume that the adversary has a limited budget B, which we express as a constraint on the number of impaired sensors. Formally, the adversary can attack any subset $A \subseteq S$ such that $|A| \leq B$. Finally, we assume that the adversary is a worst-case attacker, i.e., its goal is to maximize loss. Then, the resilient placement problem can be expressed as

$$\operatorname*{argmin}_{S \subseteq \mathcal{S} \,:\, |S| \leq M} \left(\max_{A \subseteq S \,:\, |A| \leq B} \sigma^2_{Y|(S \setminus A)} \right). \tag{11}$$

So far, we have not discussed what model is used for predicting variable Y from the observations taken at locations S. Now, we assume that a Gaussian-process-based regression model [73] is used (such models have previously been applied successfully to, for example, traffic forecasting [23]). In this model, each possible sensor location is modeled as a random variable, which represents the (*a priori* unknown) observations that would be taken at the location. Next, let $\mathbf{\Sigma}$ be the (prior) covariance matrix of the target variable and the random variables representing the locations $\mathcal{S}$. Then, the posterior variance of Y given observations at S is

$$\sigma^2_{Y|S} = \sigma^2_Y - \mathbf{\Sigma}_{YS}\mathbf{\Sigma}^{-1}_{SS}\mathbf{\Sigma}_{SY}, \tag{12}$$

where σ^2_Y is the prior (i.e., without any observations) variance of Y, $\mathbf{\Sigma}_{YS}$ (or $\mathbf{\Sigma}_{SY}$) denotes the submatrix of $\mathbf{\Sigma}$ formed by row Y (or column Y) and the columns (or rows) in set S, and $\mathbf{\Sigma}_{SS}$ denotes the principal submatrix formed by rows and columns in S.

Unfortunately, this sensor placement problem is computationally hard, even without an adversary. Formally, determining whether there exists a subset of locations $S \subseteq \mathcal{S}$ such that the variance $\sigma^2_{Y|S}$ is less than or equal to a given threshold is an NP-hard problem in general [74]. Consequently, to tackle this problem, either we have to focus on certain special cases, which can be solved efficiently, or we have to use approximation algorithms or heuristics. Das and Kempe [74] showed that the non-resilient variant of the problem (i.e., without an attacker) can be solved in polynomial time for certain covariance matrices. For example, if we represent the covariance matrix as a graph, in which nodes correspond to variables and edges correspond to pairs of variables with non-zero covariance, and this graph is a tree, then the optimal solution can be found using dynamic programming. By generalizing this result, we can show that the optimal solution of the resilient problem can also be found using dynamic programming if the graph corresponding to the covariance matrix is a tree.

However, many practical instances of the problem do not fall into these special cases, which necessitates using some heuristic or approximation algorithm. One of the most commonly used heuristics is the greedy algorithm, since it acts as an approximation algorithm for problems where the objective function is a submodular set function. Unfortunately, the above objective function is not submodular. However, for the non-resilient variant of the problem, Das and Kempe [74] provided a provable bound on the quality of the solution yielded by the greedy algorithm, based on quantifying how close the objective function is to being submodular. Similarly to the special case results, we can generalize this result as well, and we can provide a bound on the greedy algorithm for the resilient problem.

Resilient consensus algorithm. As an example of data aggregation in sensor networks in the presence of adversaries, we consider a resilient consensus problem.

A sensor network in which each sensor is connected to a subset of other sensors can be modeled by an undirected graph $G(V, E)$, where V represents the set of sensor nodes, and E represents the set of communication links between nodes through which they exchange information with their neighbors. Each sensor node i has a state value at a given time k, denoted by $x_i(k)$, which can be a sensor measurement, position variable, opinion, etc. Based on the state values of neighbor nodes, each sensor updates its state with an objective that all nodes eventually converge to a common state value. However, some nodes are compromised and act maliciously by not updating their states as per the defined update law. Consequently, their objective is to prevent the normal nodes from reaching consensus.

Thus, our objective is to design an update rule for the normal nodes such that they all converge to a common value even in the presence of a number $\mathcal{A}$ of adversarial nodes. More precisely, a resilient consensus protocol needs to be designed that can achieve the following two objectives:

- As $k \to \infty$, then $x_i(k) = x_j(k) = x$ for all the normal nodes i, j (*agreement condition*).
- Let $x_{\min}(0)$ and $x_{\max}(0)$ respectively be the minimum and maximum of the initial values of the normal nodes, then $x_{\min}(0) \leq x_i(k) \leq x_{\max}(0)$, for all k and for any normal node i (*safety condition*).

Assume that $\mathcal{N}(i)$ represents the neighbors – nodes that are directly connected to node i in G – of node i. In a typical approach to solve linear consensus in a distributed way assuming that no misbehaving node exists, each node updates its state by taking some weighted average of the states of its neighbors. In the case of an undirected graph G, this approach guarantees consensus as long as the graph is connected. However, in the case of misbehaving nodes, if there is even one such node, consensus might not be achieved.

There are approaches for resilient consensus, such as the weighted mean-subsequence-reduced (W-MSR) algorithm [68], in which each node takes a weighted average of the states of a subset of its neighbors, selected in some clever way. If the graph satisfies certain connectivity conditions, then the proposed algorithm guarantees resilient consensus in the presence of a number $\mathcal{A}$ of adversaries, where $\mathcal{A}$ is related to the connectivity of the graph. Though this approach solves the resilient consensus problem, at the same time it requires the graph to be very highly connected, even for a very small number of adversarial attacks. Thus, in the cases of a higher number of misbehaving nodes and sparse networks, the application of this approach is limited.

In another approach [75], the notion of *trusted nodes* is introduced and a resilient consensus protocol in the presence of trusted nodes is proposed to address the limitations of the above scheme. The basic idea is to distribute a set of trusted nodes, which are nodes with a higher level of security against attacks and their state values can be trusted, in such a way that consensus is achieved in the presence of *any* number of adversaries. As previously, a node updates its state by taking a weighted average of its own state and the states of its selected neighbors. Under this approach, to achieve consensus in the presence of any number of adversarial nodes, it is shown in [75] that the set of trusted nodes should form a *connected dominating set*. In other words, resilient consensus can be achieved for arbitrary $\mathcal{A}$ whenever each node is connected to at least one trusted node, and the set of trusted nodes induce a connected subgraph, as illustrated in Figure 4. However, this approach relies on the fact that trusted nodes are completely protected against attacks. Thus, both approaches have their own merits and limitations.

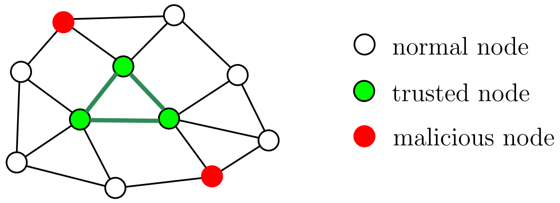

Figure 4: Each normal node is connected to a trusted node, and the set of trusted nodes induces a connected subgraph

6 Conclusions and future directions

Resilience in wireless sensor networks is a critical issue, as these networks may act as a key enabling technology in CPS, but at the same time they pose substantial security challenges. The importance of this issue is elevated by the fact that WSN in CPS are part of a closed control loop, and thus control decisions may be manipulated by attackers that have gained unauthorized access to the WSN. In this chapter, we provided an overview of key topics related to the resilience of WSN in CPS. First, we discussed potential applications of WSN in CPS, followed by an overview of possible attacks against WSN. Then, we formalized the notion of attack resilience in WSN, and discussed the main challenges to attack-resilient design. Moreover, we presented various approaches to achieve attack resilience, along with detailed examples. Finally, in the remainder of this chapter, we outline directions for future research in the area of resilient WSN for CPS.

Intrusion detection systems. Intrusion detection systems (IDS) are useful as they provide a way to detect an attack before it can cause damage. This early detection mechanism provides a way to take preventive measures before the system sustains any substantial losses, thereby elevating the attack resilience of the system. IDS have been successfully deployed in various industrial control systems; however, their utility has not been widely explored in the domain of WSN for CPS.

One of the key obstacles in this regard is that the devices in the sensor network are resource-constrained. For instance, to maximize the lifetime of the network, battery power must be conserved. However, IDS can be computationally expensive, which limits their deployment in WSN. As a result of this limitation, we may not be able to deploy IDS at every single node and may not be able to run it at all times. Therefore, we need placement and scheduling of IDS in WSN that maximize the probability of attack detection while satisfying the resource constraints of the devices and the network. In the context of CPS, a key challenge here is that this optimization – placement and scheduling – should also take into account the characteristics of the physical process.

Moving target defense. We can make a WSN more resilient to attacks by minimizing the attackers' ability to learn and exploit the system parameters. One of the ways to achieve this objective is by continuously changing the defending strategy, i.e., system configurations and defensive countermeasures. For instance, in the IDS scheduling problem discussed above, if we use a predictable schedule, an attacker might be able to time its attack to avoid detection by learning the schedule on each node. On the other hand, if we use an unpredictable schedule, an attacker will not be able to time its attack, and we will be able to detect it with high probability. Nonetheless, to the best of our knowledge, moving target defense approaches have not been applied to WSN for CPS.

Comprehensive design. A defining feature of CPS is the tight integration of physical and cyber elements, which form a closed control loop. The attack resilience of each element – physical and cyber – can be optimized individually. In this approach, the optimization of each element depends on the design and parameters of the other element, and considers them to be exogenous to the problem. Consequently, in this approach, even though each element is individually optimal, the system as a whole is not necessarily optimal. Therefore, instead of optimizing each element of the closed control loop individually, we have to devise an approach for optimizing the system as a whole.

Acknowledgment

This work was supported in part by the National Science Foundation under Award CNS-1238959.

References

[1] Akyildiz I.F., Su W., Sankarasubramaniam Y., Cayirci E. Wireless sensor networks: a survey. *Computer Networks*, 38(4): 393–422, 2002.

[2] Lee E.A. Cyber physical systems: design challenges. In *Proceedings of the 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, pp. 363–369. IEEE, 2008.

[3] Mainwaring A., Culler D., Polastre J., Szewczyk R., Anderson J. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97. ACM, 2002.

[4] Son B., Her Y., Kim J.G. A design and implementation of forest-fires surveillance system based on wireless sensor networks for South Korea mountains. *International Journal of Computer Science and Network Security*, 6(9): 124–130, 2006.

[5] Wu F.J., Kao Y.F., Tseng Y.C. From wireless sensor networks towards cyber physical systems. *Pervasive and Mobile Computing*, 7(4): 397–413, 2011.

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

[7] Sridhar S., Hahn A., Govindarasu M. Cyber-physical system security for the electric power grid. *Proceedings of the IEEE*, 100(1): 210–224, 2012.

[8] McDaniel P., McLaughlin S. Security and privacy challenges in the smart grid. *IEEE Security Privacy*, 7(3/May): 75–77, 2009.

[9] Rial A., Danezis G. Privacy-preserving smart metering. In *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society (WPES'11)*, pp. 49–60. ACM, 2011.

[10] Sankar L., Rajagopalan S.R., Mohajer S., Poor H.V. Smart meter privacy: a theoretical framework. *IEEE Transactions on Smart Grid*, 4(2): 837–846, 2013.

[11] Alemdar H., Ersoy C. Wireless sensor networks for healthcare: a survey. *Computer Networks*, 54(15): 2688–2710, 2010.

[12] Haykin S.S. (ed.) *Kalman Filtering and Neural Networks*. John Wiley, 2001.

[13] Olfati-Saber R. Distributed Kalman filtering for sensor networks. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pp. 5492–5498. IEEE, 2007.

[14] Hernandez L., Baladrón C., Aguiar J., *et al.* A survey on electric power demand forecasting: future trends in smart grids, microgrids and smart buildings. *IEEE Communications Surveys & Tutorials*, 16(3): 1460–1495, 2014.

[15] Mirowski P., Chen S., Ho T.K., Yu C.N. Demand forecasting in smart grids. *Bell Labs Technical Journal*, 18(4): 135–158, 2014.

[16] Amin S., Schwartz G., Cardenas A., Sastry S. Game-theoretic models of electricity theft detection in smart utility networks: providing new capabilities with advanced metering infrastructure. *IEEE Control Systems*, 35(1): 66–81, 2015.

[17] McLaughlin S., Holbert B., Fawaz A., Berthier R., Zonouz S. A multi-sensor energy theft detection framework for advanced metering infrastructures. *IEEE Journal on Selected Areas in Communications*, 31(7): 1319–1330, 2013.

[18] Fang X., Misra S., Xue G., Yang D. Smart grid – the new and improved power grid: a survey. *IEEE Communications Surveys & Tutorials*, 14(4): 944–980, 2012.

[19] Gungor V.C., Lu B., Hancke G.P. Opportunities and challenges of wireless sensor networks in smart grid. *IEEE Transactions on Industrial Electronics*, 57(10): 3557–3564, 2010.

[20] Ghena B., Beyer W., Hillaker A., Pevarnek J., Halderman J.A. Green lights forever: analyzing the security of traffic infrastructure. In *Proceedings of the 8th USENIX Workshop on Offensive Technologies (WOOT'14)*, pp. 1–10. USENIX Association, 2014.

[21] Tubaishat M., Zhuang P., Qi Q., Shang Y. Wireless sensor networks in intelligent transportation systems. *Wireless Communications and Mobile Computing*, 9(3): 287–302, 2009.

[22] Smith B.L., Demetsky M.J. Traffic flow forecasting: comparison of modeling approaches. *Journal of Transportation Engineering*, 123(4): 261–266, 1997.

[23] Xie Y., Zhao K., Sun Y., Chen D. Gaussian processes for short-term traffic volume forecasting. *Journal of the Transportation Research Board*, 2165(1): 69–78, 2010.

[24] Raymond D.R., Midkiff S.F. Denial-of-service in wireless sensor networks: attacks and defenses. *IEEE Pervasive Computing*, 7(1): 74–81, 2008.

[25] Wang Y., Attebury G., Ramamurthy B. A survey of security issues in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 8(8): 2–23, 2006.

[26] Wood A., Stankovic J.A. Denial of service in sensor networks. *Computer*, 35(10): 54–62, 2002.

[27] Xu W., Ma K., Trappe W., Zhang Y. Jamming sensor networks: attack and defense strategies. *Network, IEEE*, 20(3): 41–47, 2006.

[28] Maheshwari R., Gao J., Das S.R. Detecting wormhole attacks in wireless networks using connectivity information. In *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM'07)*, pp. 107–115. IEEE, 2007.

[29] Poovendran R., Lazos L. A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks. *Wireless Networks*, 13(1): 27–59, 2007.

[30] Gu Q., Ferguson C., Noorani R. A study of self-propagating mal-packets in sensor networks: attacks and defenses. *Computers & Security*, 30(1): 13–27, 2011.

[31] Francillon A., Castelluccia C. Code injection attacks on Harvard-architecture devices. In *Proceedings of the 15th ACM Conference on Computer and Communications Security*, pp. 15–26. ACM, 2008.

[32] Yang Y., Zhu S., Cao G. Improving sensor network immunity under worm attacks: a software diversity approach. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 149–158. ACM, 2008.

[33] Wang Y., Wen S., Xiang Y., Zhou W. Modeling the propagation of worms in networks: a survey. *IEEE Communications Surveys & Tutorials*, 16(2): 942–960, 2014.

[34] Liu Y., Ning P., Reiter M.K. False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security*, 14(1): 13, 2011.

[35] Xie L., Mo Y., Sinopoli B. False data injection attacks in electricity markets. In *Proceedings of the 1st International Conference on Smart Grid Communications*, pp. 226–231. IEEE, 2010.

[36] Newsome J., Shi E., Song D., Perrig A. The sybil attack in sensor networks: analysis and defenses. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, pp. 259–268. ACM, 2004.

[37] Parno B., Perrig A., Gligor V. Distributed detection of node replication attacks in sensor networks. In *IEEE Symposium on Security and Privacy*, pp. 49–63. IEEE, 2005.

[38] Amin S., Cárdenas A.A., Sastry S.S. Safe and secure networked control systems under denial-of-service attacks. In *Hybrid Systems: Computation and Control*, pp. 31–45. Springer, 2009.

[39] Amin S., Litrico X., Sastry S., Bayen A.M. Stealthy deception attacks on water SCADA systems. In *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control*, pp. 161–170. ACM, 2010.

[40] Amin S., Litrico X., Sastry S., Bayen A.M. Cyber security of water SCADA systems – Part I: Analysis and experimentation of stealthy deception attacks. *IEEE Transactions on Control Systems Technology*, 21(5): 1963–1970, 2013.

[41] Mo Y., Chabukswar R., Sinopoli B. Detecting integrity attacks on scada systems. *IEEE Transactions on Control Systems Technology*, 22(4): 1396–1407, 2014.

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

[43] Rezvani M., Ignatovic A., Bertino E., Jha S. Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks. *IEEE Transactions on Dependable and Secure Computing*, 12(1): 98–110, 2015.

[44] Kushner D. The real story of Stuxnet. *IEEE Spectrum*, 50(3): 48–53, 2013.

[45] Kaspersky Lab. Kaspersky Lab provides its insights on Stuxnet worm, 24 September 2010. See `http://www.kaspersky.com/about/news/virus/2010/Kaspersky_Lab_provides_its_insights_on_Stuxnet_worm`. Accessed: 1 May 2015.

[46] Kelley M.B. The Stuxnet attack on Iran's nuclear plant was "far more dangerous" than previously thought. *Business Insider*, 20 November 2013. See `http://www.businessinsider.com/stuxnet-was-far-more-dangerous-than-previous-thought-2013-11`. Accessed: 27 April 2015.

[47] Cerrudo C. Hacking US (and UK, Australia, France, etc.) traffic control systems. *IOActive Labs Blog*, 30 April 2014. See `http://blog.ioactive.com/2014/04/hacking-us-and-uk-australia-france-etc.html`. Accessed: 3 May 2015.

[48] Zetter K. Hackers can mess with traffic lights to jam roads and reroute cars. *Wired*, 30 April 2014. See `http://www.wired.com/2014/04/traffic-lights-hacking/`. Accessed: 3 May 2015.

[49] ICS-CERT. *Sensys Networks Traffic Sensor Vulnerabilities*, 28 October 2014. Advisory (ICSA-14-247-01A). See `http://ics-cert.us-cert.gov/advisories/ICSA-14-247-01A`. Accessed: 3 May 2015.

[50] Mouawad J. Conflict narrows oil options for West. *The New York Times*, 14 August 2008. See `http://www.nytimes.com/2008/08/14/world/europe/14oil.html`. Accessed: 3 May 2015.

[51] Robertson J., Riley M.A. Mysterious '08 Turkey pipeline blast opened new cyberwar. *Bloomberg*, 10 December 2014. See `http://www.bloomberg.com/news/articles/2014-12-10/mysterious-08-turkey-pipeline-blast-opened-new-cyberwar`. Accessed: 27 April 2015.

[52] Cobham A. The intrinsic computational difficulty of functions. In *Proceedings of the 1964 Congress for Logic, Methodology, and the Philosophy of Science*, pp. 24–30. North-Holland, 1965.

[53] Kerckhoffs A. La cryptographie militaire. *Journal des Sciences Militaires*, IX(January): 5–38, 1883.

[54] Pita J., Jain M., Tambe M., Ordóñez F., Kraus S. Robust solutions to Stackelberg games: addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence*, 174(15): 1142–1171, 2010.

[55] Freund Y., Kearns M., Mansour Y., Ron D., Rubinfeld R., Schapire R.E. Efficient algorithms for learning to play repeated games against computationally bounded adversaries. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 332–341. IEEE, 1995.

[56] Deng J., Han R., Mishra S. A performance evaluation of intrusion-tolerant routing in wireless sensor networks. In *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN)*, pp. 349–364. Springer, 2003.

[57] Al-Karaki J.N., Kamal A.E. Routing techniques in wireless sensor networks: a survey. *Wireless Communications*, 11(6): 6–28, 2004.

[58] Akkaya K., Younis M. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3): 325–349, 2005.

[59] Younis M., Akkaya K. Strategies and techniques for node placement in wireless sensor networks: a survey. *Ad Hoc Networks*, 6(4): 621–655, 2008.

[60] Cardei M., Wu J. Coverage in wireless sensor networks. In *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, eds Ilyas M., Mahgoub I., pp. 422–433. CRC Press, 2004.

[61] Wang Y.C., Tseng Y.C. Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage. *IEEE Transactions on Parallel and Distributed Systems*, 19(9): 1280–1294, 2008.

[62] Dhillon S.S., Chakrabarty K. Sensor placement for effective coverage and surveillance in distributed sensor networks. In *Proceedings of the 2003 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1609–1614. IEEE, 2003.

[63] Abbas W., Egerstedt M. Robust graph topologies for networked systems. In *Proceedings of the 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pp. 85–90. Elsevier, 2012.

[64] Dekker A.H., Colbert B.D. Network robustness and graph topology. In *Proceedings of the 27th Australasian Conference on Computer Science*, vol. 26, pp. 359–368. Australian Computer Society, 2004.

[65] Laszka A., Buttyán L., Szeszlér D. Designing robust network topologies for wireless sensor networks in adversarial environments. *Pervasive and Mobile Computing*, 9(4): 546–563, 2013.

[66] Santi P. Topology control in wireless ad hoc and sensor networks. *ACM Computing Surveys*, 37(2): 164–194, 2005.

[67] Wagner D. Resilient aggregation in sensor networks. In *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 78–87. ACM, 2004.

[68] LeBlanc H.J., Zhang H., Koutsoukos X., Sundaram S. Resilient asymptotic consensus in robust networks. *IEEE Journal on Selected Areas in Communications*, 31(4): 766–781, 2013

[69] Sundaram S., Hadjicostis C.N. Distributed function calculation via linear iterative strategies in the presence of malicious agents. *IEEE Transactions on Automatic Control*, 56(7): 1495–1508, 2011.

[70] Marano S., Matta V., Tong L. Distributed detection in the presence of byzantine attacks. *IEEE Transactions on Signal Processing*, 57(1): 16–29, 2009.

[71] Olfati-Saber R., Franco E., Frazzoli E., Shamma J. Belief consensus and distributed hypothesis testing in sensor networks. In *Networked Embedded Sensing and Control. Lecture Notes in Control and Information Science*, eds Antsaklis P.J., Tabuada P., pp. 169–182. Springer, 2006.

[72] Vempaty A., Tong L., Varshney P. Distributed inference with byzantine data: state-of-the-art review on data falsification attacks. *IEEE Signal Processing Magazine*, 30(5): 65–75, 2013.

[73] Rasmussen C.E., Williams C.K.I. *Gaussian Processes for Machine Learning.* MIT Press, 2006.

[74] Das A., Kempe D. Algorithms for subset selection in linear regression. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pp. 45–54. ACM, 2008.

[75] Abbas W., Vorobeychik Y., Koutsoukos X. Resilient consensus protocol in the presence of trusted nodes. In *Proceedings of the 7th International Symposium on Resilient Control Systems.* pp. 1–7. IEEE, 2014.