CHAPTER

11

Security in Smart Cyber-Physical Systems: A Case Study on Smart Grids and Smart Cars

Sandeep Nair Narayanan, Kush Khanna†, Bijaya Ketan Panigrahi†, and Anupam Joshi**

***University of Maryland Baltimore County, Baltimore, MD, United States**
†Indian Institute of Technology Delhi, New Delhi, India

1 INTRODUCTION

Across the globe, cities are expanding in size and infrastructure. The idea of Smart Cities plays a vital role in offering higher efficiency, comfort, awareness, and convenience to end users. Harrison et al. [1] describe the Smart City as an instrumented, interconnected, and intelligent city. They instrument different sectors of smart infrastructure, such as smart energy, smart transportation, smart governance, smart healthcare, smart buildings, and so forth to capture real-world data and interconnect them to share this data among different services. The shared data is then used to make intelligent operational decisions using complex analytics and provide better facilities to end users. Smart Cyber-Physical Systems (CPSs) are essential components of all smart infrastructure. According to the National Science Foundation (NSF),[1] "Cyber-physical systems integrate sensing, computation, control and networking into physical objects and infrastructure, connecting them to the Internet and each other." Smart cars and smart grids are two CPS domains that have demonstrated tremendous growth over the past few decades. However, the capabilities of these CPSs to influence critical infrastructure make them a lucrative target for hackers. Some of the attacks even have a direct impact on the economy of a nation. For example, consider the attack on the Ukrainian smart grids [2]. It left a whole city without heat and electricity in the cold of December for many hours. Similarly in the domain of smart cars, although they are capable of providing efficient transportation and fewer accidents, the potential attacks against them are alarming. The famous Jeep Hack of 2016, in which the researchers manipulated a moving unaltered vehicle on the road, resulted in Chrysler recalling 1.4 million[2] vehicles. Hence, there is an urgent requirement to secure these individual Smart City components. In this chapter, we delve into the general architecture and security of smart grids and smart cars.

Electrical energy is considered the most efficient form of energy for generation, transmission, and energy conversion. The traditional power grid is getting smarter with large-scale automation and integration of new smart components capable of quick decision making, thereby improving the reliability of the entire system. The futuristic "smart grid" is a cyber-physical power system of interconnected smart metering infrastructure that enables autonomous and resilient operation. Due to technological advances, many consumer-centric facilities such as demand-side management, advanced metering infrastructure, dynamic pricing, and incentives have evolved for serving the consumer in a better and more reliable way. The huge amount of data collected from smart meters across the cities will result in better management of resources, ultimately lowering the cost for consumers.

[1] See https://www.nsf.gov/news/special_reports/cyber-physical/.

[2] See https://www.wired.com/2015/07/jeep-hack-chrysler-recalls-1-4m-vehicles-bug-fix/.

https://doi.org/10.1016/B978-0-12-815032-0.00011-1

Another CPS that acquired smart decision-making capability is cars. They developed from being just mechanical devices into machines that can drive autonomously. They can provide safe, efficient, and fast transportation in Smart Cities. Driver-assisted features, such as antilock braking systems (ABS), adaptive cruise control, and blind spot warning systems, are becoming the de facto in many newer cars. For example, the latest version of the Accord from Honda Motor Company has the *Honda Sensing*[3] (Honda's package of smart features such as the Collision Mitigation Braking System, Road Departure Mitigation System, Adaptive Cruise Control, and Lane Keeping Assist System) feature as a default, even for its base version. Current cutting-edge research in this domain focuses on autonomous driving in which a car can navigate you from one location to another without direct human intervention. Waymo[4] (an Alphabet company), Uber,[5] GM motors,[6] Tesla,[7] and so forth are some the automotive giants who are innovating in this area. Waymo's fleet of autonomous cars has already driven more than 4 million[8] combined miles on normal roads.

When the smart features of CPSs were upgraded, they were connected to the cloud environment over the Internet, either directly or indirectly. The lack of connectivity was acting as a security feature in disguise until it became defunct. Moreover, power grids and automobiles existed for a long period, and evolved over the past century. The design considerations for some of the legacy systems were efficiency and speed with lower priority to security features such as authentication and authorization. For example, the Controller Area Network (CAN) bus used in many cars is a simple broadcast bus and messages on it are considered authenticated. An attacker who can sneak a crafted message on this bus can perform many malicious activities. Supporting such legacy devices makes their security complicated and opens many interesting research avenues.

This chapter explores the general architecture of smart grids and smart cars as smart CPSs in Section 2. Sections 3 and 4 then delve into the security in smart grids and smart cars by discussing specific attacks and countermeasures proposed to identify those attacks. Finally, this chapter is concluded in Section 5 by presenting the future outlook on the cybersecurity issues in smart systems, and discussing recommendations.

2 GENERAL SMART CYBER-PHYSICAL SYSTEM ARCHITECTURE

As described in Section 1, smart CPSs are essential components of a Smart City. Each system should be instrumented and interconnected to build it. In this section, we describe the architecture of two such instrumented smart CPSs, smart grids (Section 2.1) and smart cars (Section 2.2).

2.1 Smart Cyber-Physical System: Smart Grids

The union of Information Technology (IT) and Operational Technology (OT) has resulted in higher reliability, efficiency, better services, and ease of use for customers and utilities in every domain. In the recent past, the electrical energy sector is witnessing extensive modernization owing to the convergence of cyber and physical technologies. The main objective of the power system is to generate and transmit electrical energy from the power plants to the consumers reliably and efficiently. A cyber-physical power system can be considered a conventional power grid with integrated communication technologies enabling cyber and physical connections between various power system components. An overview of the electrical power system with a review of state-of-the-art communication technology is presented in later sections.

2.1.1 Electric Power system

The conventional power system can be broadly divided into three core domains: generation, transmission, and distribution. The generating units (power plants) are generally located in remote areas far from the end consumers. To transmit electrical power efficiently, the voltage level at the generation site is stepped up using power transformers. The generation site is also equipped with generation remote terminal units (RTUs), integrated electronic

[3] See https://automobiles.honda.com/safety.

[4] See https://waymo.com/.

[5] See https://www.uber.com/.

[6] See https://www.gm.com/.

[7] See https://www.tesla.com/.

[8] See https://waymo.com/ontheroad/.

devices (IEDs), current and voltage transformers (CTs/PTs), relays, and circuit breakers (CBs) for protection and control of critical generating units. The transmission system transmits high-voltage electrical power to the distribution system, which further distributes electrical energy to the consumers (industries, commercial buildings, hospitals, and households) as shown in Fig. 1. The transmission system is equipped with phasor measurement units (PMUs), RTUs, and IEDs to measure voltage and current for the operation, and to monitor power systems. In the smart grid environment, the measured data from various RTUs, PMUs, and IEDs are received by data concentrating units installed at the substations, and are further transmitted to a control center. The operation of conventional power systems with the integration of communication technology is more transparent as the latter has enabled wide area situational awareness of power system components available at the control center.

2.1.1.1 Power System Operation and Control

Supervisory control and data acquisition (SCADA) systems receive metered data for monitoring the operation of a power system. Based on the measurements received from the SCADA system, a state estimator (SE) is used to obtain an accurate snapshot of the power system in real-time [3]. Power system stability and security studies rely heavily on the estimated states. To ensure the accuracy of the estimated states, an SE generally uses redundant measurements that will filter out noises and telemetry errors [4]. Estimation of voltages at each bus (node) of the power system is the prime objective of an SE. The measurements required for the correct estimation of $V\angle\theta$ (complex voltage) at each bus are real and reactive power injections at buses; real and reactive power flows in the transmission lines; and voltage measurements at generator buses.

For N bus power system, the total number of states to be estimated is $2N - 1$; that is, $N - 1$ angles (one angle is considered as the reference angle) and N voltage magnitudes.

The state vector is given as,

$$\boldsymbol{x} = [\theta_2, \theta_3, \ldots, \theta_N, V_1, V_2, \ldots, V_N]. \tag{1}$$

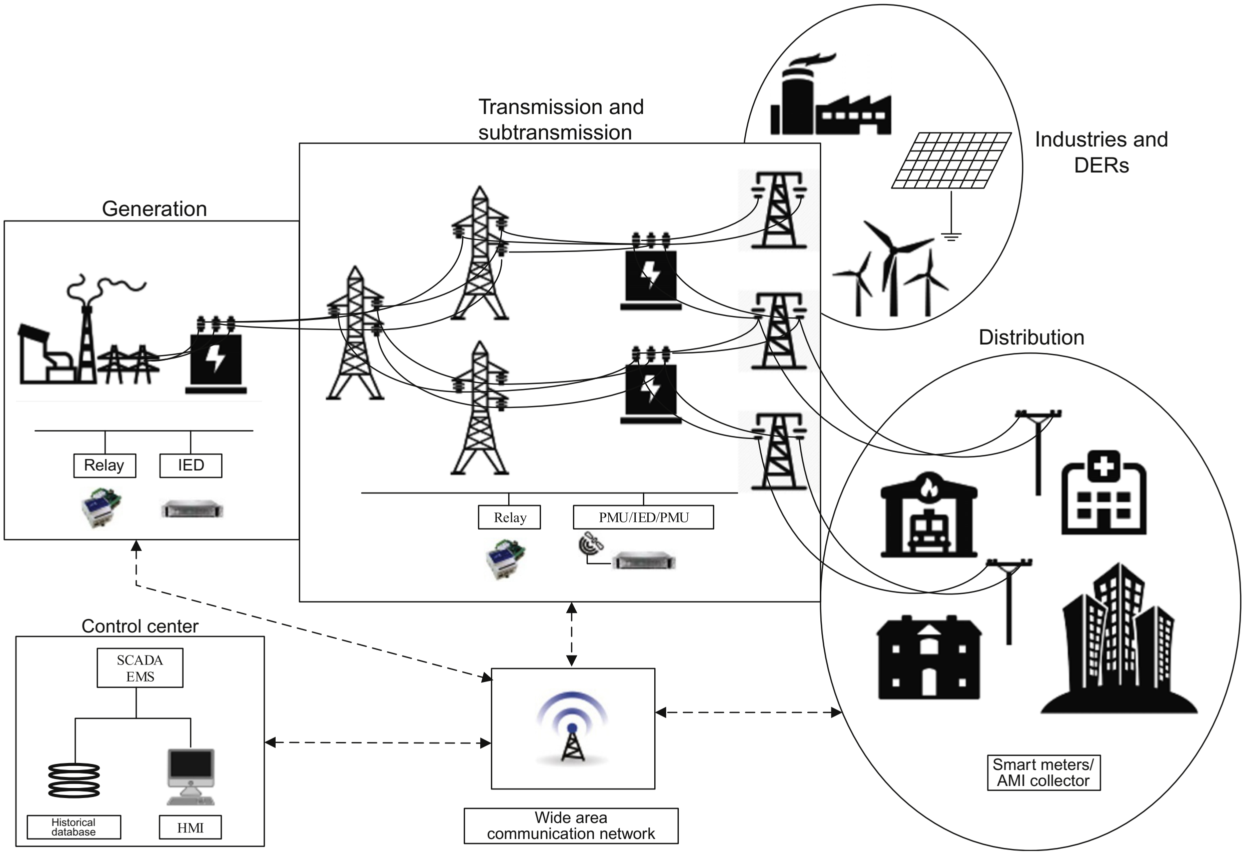


FIG. 1 Complete smart grid schematic.

Once the power system states are estimated, the security and contingency analysis is carried out for the power system's estimated snapshot. The EMS is responsible for unit commitment, economic load dispatch, automatic generation control, and many other critical functions of the power grid. Fig. 2 highlights the importance of SE in the operation and control of power systems.

2.1.1.2 Communication Technology

To transfer information from RTUs, IEDs, and relays to substations; from substations to regional and central load dispatch centers; from market operators to independent system operators (ISOs); and in a smart grid environment, from end consumers to utilities and vice versa, the smart cyber-physical power system uses varied communication technologies. Dedicated cables, optical fiber communication, wireless communication, and power line carriers are some of the communication technologies used in the smart grid. For substation automation (SA), the IEC 61850 Standard enhances the interoperability of devices from different vendors. In addition, IEC 61850 supports high-speed communication of generic object oriented substation event (GOOSE) messages for sending trip commands from relays to CBs, as shown in Fig. 3. IEC 61850 also supports sampled values to send measured voltage and current values from the measurement devices to the merger units. Manufacturing message specification is used to communicate device status and control commands to and from IEDs.

Deployment of PMUs increases the situational awareness of large-scale power systems. PMUs are capable of sampling data at 60–240 Hz. The sampled data is time stamped and synchronized using global positioning systems (GPS). The data from the PMU is transmitted to the phasor data concentrator (PDC) using a standard IEEE C37.118 protocol. PDC further communicates over the Ethernet and local area network of the substation or control center to the SA/SCADA/EMS system. PMUs facilitate voltage and current measurements with a common reference (as data is time stamped), which results in accurate state estimation (SE) and enhances the observability of the power system.

2.2 Smart Cyber-Physical System: Smart Cars

Smart cars are an integral part of new Smart Cities. Apart from providing human convenience, they can alleviate traffic issues, reduce accidents due to human error, provide intelligent and time-bound goods delivery, and so forth. A modern car has a large number of smart subsystems (Fig. 4) and they make a lot of intelligent decisions every second. Some example subsystems are ABS, driver alertness monitoring, blind spot detection, and so forth.

Automobiles as just moving parts, pulleys, and mechanical devices solely controlled by human drivers are antiquated. Advancements in microprocessor technologies resulted in the addition of electronic control units (ECUs) for making fast and smart decisions for many activities. Typically, an ECU will be connected to different sensors (e.g., acceleration sensor, rain sensor, ambient light sensor, wheel speed sensor, RPM sensor, vehicle speed sensor, oxygen sensor, temperature sensor) and actuators (e.g., instrument clusters, spark plugs, airbag inflators) using cables. Even though all automobiles have similar facilities, as of now, they do not have a standardized architecture.

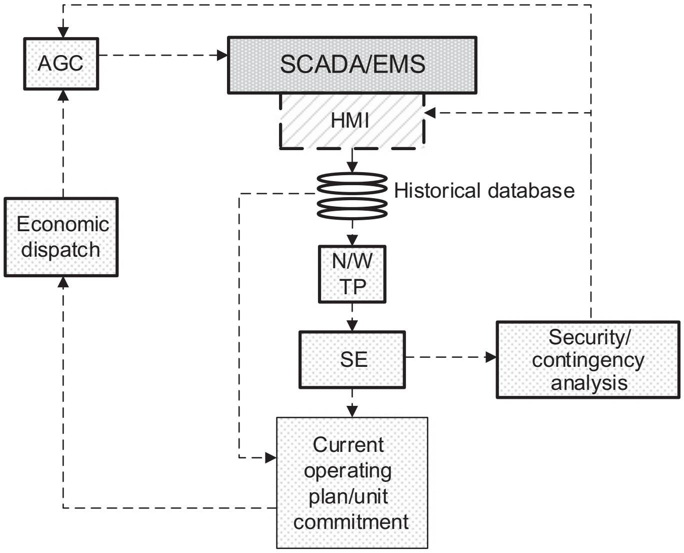

AGC
SCADA/EMS
HMI
Historical database
Economic dispatch
N/W TP
SE
Security/ contingency analysis
Current operating plan/unit commitment

FIG. 2 SCADA/EMS operation flow.

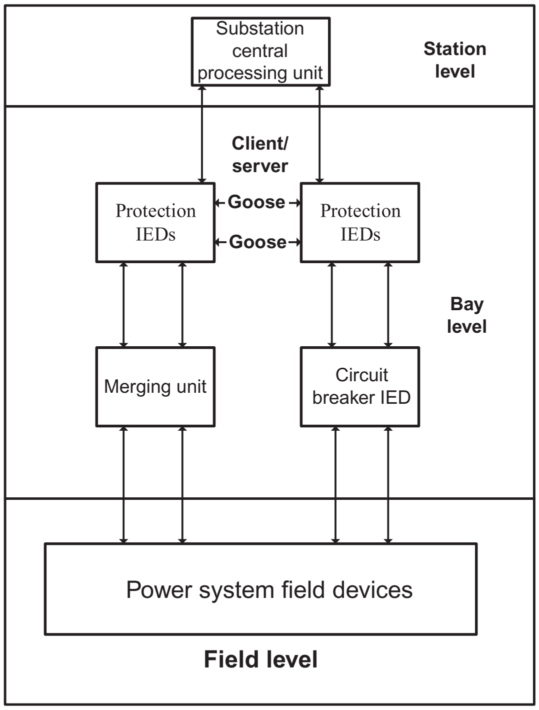


FIG. 3 SA communication architecture (IEC 61850).

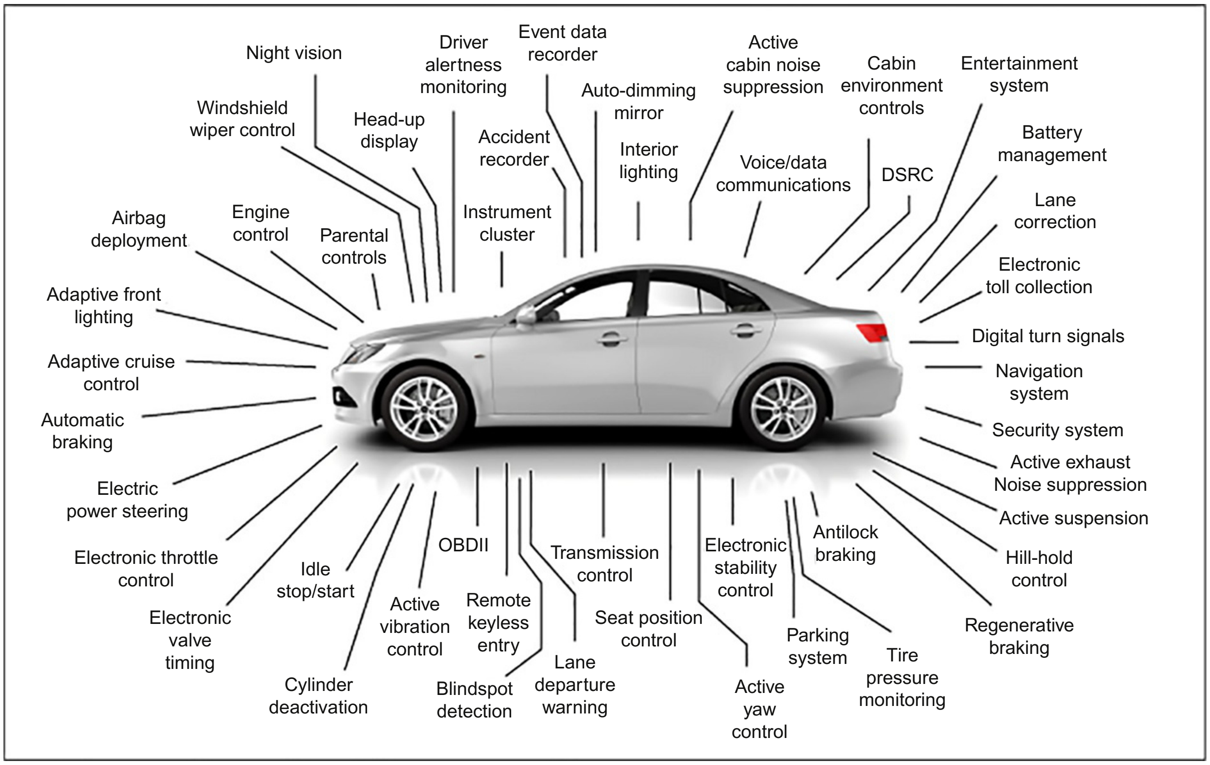
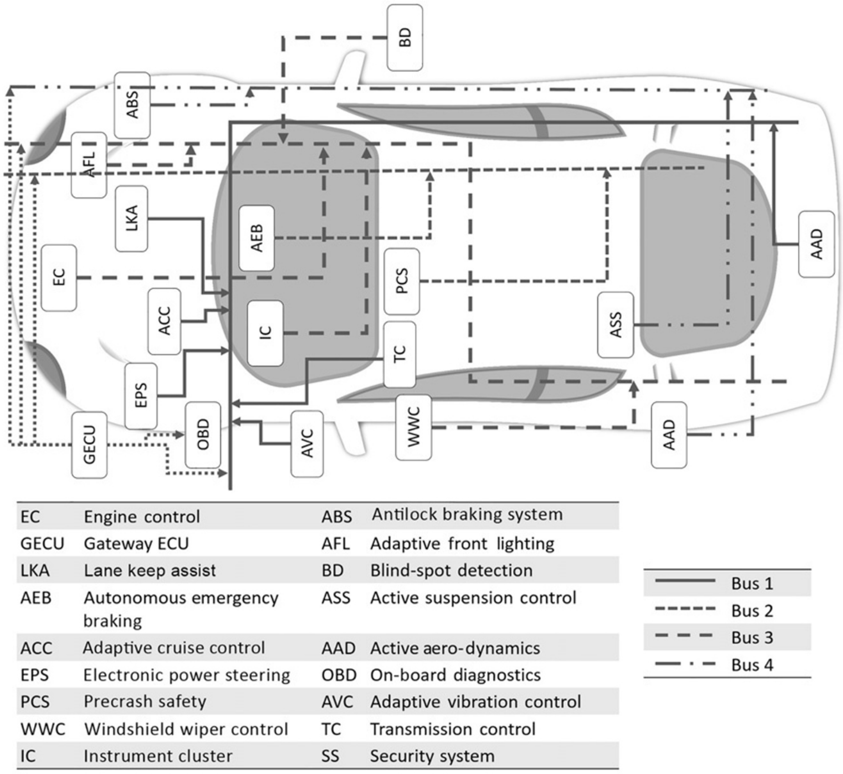


FIG. 4 Modern car subsystems. *(Courtesy: Clemson Vehicular Electronics Laboratory.)*

Different manufacturers such as Honda, Toyota, BMW, Benz, Kia, and General Motors use different types of equipment, internal wiring architecture, and protocols. However, in general, we can see that all of them use one or more buses inside, and most of the subsystems and sensors are connected to them as shown in Fig. 5. A subsystem needs to meet specific constraints according to the task it is assigned to. For example, ABS needs to meet strict time

FIG. 5 General vehicle architecture.



EC	Engine control	ABS	Antilock braking system
GECU	Gateway ECU	AFL	Adaptive front lighting
LKA	Lane keep assist	BD	Blind-spot detection
AEB	Autonomous emergency braking	ASS	Active suspension control
ACC	Adaptive cruise control	AAD	Active aero-dynamics
EPS	Electronic power steering	OBD	On-board diagnostics
PCS	Precrash safety	AVC	Adaptive vibration control
WWC	Windshield wiper control	TC	Transmission control
IC	Instrument cluster	SS	Security system

requirements. Otherwise, the vehicle will not stop at the required location. On the other hand, some subsystems, such as entertainment subsystems, need not meet the time requirement, but they need high-volume data transfer. Different buses are chosen based on these requirements. Attributed to lack of standardization, some sensors are directly connected to the ECU, while others are connected to the common bus. Broadly, each vehicle has at least a common bus and different smart subsystems.

2.2.1 *Common Bus*

Earlier vehicle manufacturers used a point-to-point wiring harness to connect all the components, and it became complicated due to an increase in the number of components. In 1986, Robert Bosch GMBH introduced a lightweight and low-cost serial bus protocol called the CAN for vehicular communication, because none of the existing protocols had the required characteristics. CAN was a broadcast-based protocol in which all the devices were connected to the same bus, and all devices saw all network communications. Error correction and priority maintenance were its important characteristics, along with decreased complexity in the wiring harness. The newer CAN 2.0 specification was published in 1992, and became widely used by different manufacturers. The number of smart systems has increased considerably over the past few decades. According to a report,[9] the average number of such systems in a car increased from 24 in 2002 to 70 in 2013. To cope with this increase in the number of components and speed requirements of different subsystems, newer protocols and buses were introduced. Some of the protocols over the CAN bus are the ISO-TP protocol, CANopen, and so forth. A newer version of CAN called the CAN-FD was introduced in 2011, which offered flexible data rates. Other protocols introduced include the PWM protocol from Ford, the Keyword protocol (KWP2000), the VPW protocol used in GM and Chrysler, the local interconnect network (LIN) protocol,[10] the media oriented system transport (MOST) protocol,[11] and so forth. Each of them has specific characteristics. For example, LIN is a very inexpensive protocol to implement, and uses only a single wire bus. But it supports only up to 20 kbps. On the other hand, a version of the MOST protocol supports up to 150 Mbps, with the associated complexity of implementation. Another high-speed bus with a communication speed of up to 10 Mbps

[9] See http://cvrr.ucsd.edu/ece156/AutomotiveSensors-Review-IEEESensors2008.pdf.

[10] See https://vector.com/vi_lin_spec_download_en.html.

[11] See https://www.mostcooperation.com/publications/specifications-organizational-procedures/request-download/mostspecificationpdf/.

is FlexRay.[12] Due to diminishing support and increasing cost, newer vehicles are moving toward a newer protocol, automotive Ethernet.[13] As of now, a common practice seen among manufacturers is to use different protocols and buses for different subsystems. Typically, MOST or FlexRay are used for high-end systems, CAN is used for mid-range systems, and LIN is used for low-cost devices.

2.2.2 Smart Subsystems

More and more driver-assisted technologies are becoming the de facto in a modern car. Some of the driver-assisted technologies available in a car include ABS, cruise control, lane-assist, power brakes, power steering, central locking systems, and so forth. Technology is now taking the next step to move toward autonomous control in which the subsystems will take over driving as a whole. Broadly, we classify these subsystems into three categories.

1. *Indicative smart subsystems*: Indicative systems identify certain states of the vehicle using their sensory inputs just to alert the driver. The smart blind-spot detection system is an example-indicative system that detects objects in a car's blind spot (the region around the vehicle that is not visible to the driver using rear view mirrors) and alerts the user. It uses sensory inputs such as radar and ultrasonic, and vehicle speed sensors to detect an alien object's presence. On detection, depending on the system, it alerts the user using a display, sound, or haptic warning in the steering wheel or seats. Other examples of indicative systems include lane departure warnings, tire pressure monitoring, and so forth.
2. *Reactive smart subsystems*: Reactive smart systems, on the other hand, take necessary actions to avoid potentially dangerous incidents. A very common example of such a system is an adaptive cruise control system that automatically adjusts the speed of a vehicle for the live road traffic. It uses a headway radar or lidar to detect traffic in front, and utilizes vehicle speed sensors, accelerator pedal position, and brake pedal position to identify the action to be taken for maintaining a safe distance on the road. It will then actuate the throttle or brakes to adjust the speed of the vehicle automatically when the vehicle is in a cruise mode. Other example subsystems include adaptive front lighting, parking systems, auto dimming mirrors, ABSs, airbag deployment, and so forth.
3. *Predictive smart subsystems*: In predictive smart systems, after identifying specific states of the vehicle, they anticipate certain actions from the driver. Instead of directly affecting the state of the car using actuators, they get the vehicle ready for such actions. A typical example of such a system is collision warning with brake support from Ford. In the event of an obstacle ahead and a potential collision, the system not only warns the user about the condition, it precharges the braking systems. This action will help to apply a full application of the brakes with a simple touch on the brake pedal, hence avoiding an imminent collision. The precrash safety feature found in some automobiles is another example for such systems.

3 SECURITY IN SMART CYBER-PHYSICAL SYSTEMS: SMART GRIDS

Cyber threats have now expanded from just targeting computers and smartphones to hacking power grids. In the recent past, many attacks on power grids have been reported.

1. *The attack on the Ukrainian power grid*: On December 23, 2015, a cyberattack on the Ukrainian power grid resulted in a power outage to approximately 225,000 customers. The attackers used spear phishing to gain access to the business networks. Using virtual private networks, the adversary entered the ICS network. Further, the attackers used existing remote access tools for issuing the critical commands directly from the remote station, mimicking an HMI operator. The attackers opened various CBs, which resulted in a system-wide collapse. The illegal entry into the utility's SCADA and control center caused these attacks. The attack also forced the system operators to move to manual mode to prevent further escalation. This attack exposed the vulnerabilities in the existing infrastructure. It is thus learned that an adversary can use spear phishing to install malware in a smart power grid network and use it to launch attacks. The Electricity Information Sharing and Analysis Center (E-ISAC) recommends various countermeasures to prevent attacks such as the one on the Ukraine in the future [2].
2. *Attacks on PMUs*: PMUs are deployed in many remote locations to capture phasors accurately and more frequently compared with traditional SCADA devices for wide area measurement, protection, and control. As the PMUs are time stamped using GPS signals, the measurements are vulnerable to data and GPS spoofing. An

[12] See https://svn.ipd.kit.edu/nlrp/public/FlexRay/FlexRayProtocolSpecificationVersion3.0.1.pdf.

[13] See http://www.opensig.org/about/specifications/.

adversary can cause an erroneous GPS clock offset of the PMU receiver, resulting in false time stamp calculation [5]. The error in time stamp calculation by the PMU can adversely affect the stability of the power system. It is also observed that the errors can even cause the tripping of generators and transmission lines.

It is also worth noting that an adversary need not intrude into the actual PMU device, he/she just needs to modify the timing signal received by the PMU device. The resulting measurements with incorrect timestamps will be communicated to the system operator, resulting in a false estimation of states. It will affect the voltage stability, fault location, and detection in transmission lines [6].

3. *Attacks on state estimation*: Reliable and secure operation of electrical grid banks rely on the precise estimation of the power system's operating state [4]. The integrity of SE depends on the precision of the measurement sensors. The dependence of the SCADA/EMS on the communication technology makes SE susceptible to data integrity attacks. Because SE is the crux of the entire power system operation and control, malicious data injection attacks in the measurement sensors can have catastrophic consequences.

False data injection attacks (FDIAs) can be random or targeted. FDIAs compromise both the confidentiality and integrity of the information obtained from smart meters. In a random-attack scenario, the attacker injects malicious data into random measurement sensors with a goal of making the SE erroneous. The random attacks can still be detected by the control center, but rather imperfectly due to the presence of measurement noise [7]. However, in targeted FDIAs, the goal is to inject predetermined errors in some specific state variables [8]. Such attacks, if injected stealthily on certain specific measurement sensors, are undetectable for the system operator as they bypass the bad data detection (BDD), even without meter noise.

A simple FDIA can be launched by injecting significant errors into the measurements. The errors should be just enough to cause changes in the system state, but also small enough to bypass BDD. The attack can be formulated as

$$J(\hat{\boldsymbol{x}}_{bad}) = \sum_{i=1}^{m} (z_i - h(\hat{\boldsymbol{x}}) + a_i)^2 / \sigma_i^2 \leq \tau. \tag{2}$$

Here a_i is the malicious error injected in the sensor z_i, $h(\hat{x})$ is the measurement function, $\hat{x}$ is the estimated state vector, and τ is the threshold for BDD.

If we assume that the adversary has acquired the network and topology information, a stealthy attack can be launched bypassing BDD [4], that is, $\| \boldsymbol{z} - \boldsymbol{H}\hat{\boldsymbol{x}} \| \leq \tau$, if the attack vector, $\boldsymbol{a} = \boldsymbol{Hc}$, shown as follows:

$$\begin{aligned} \| (\boldsymbol{z} + \boldsymbol{a}) - H(\hat{\boldsymbol{x}} + \boldsymbol{c}) \| \ &= \| \boldsymbol{z} - \boldsymbol{H}\hat{\boldsymbol{x}} + \boldsymbol{a} - \boldsymbol{Hc} \| \\ &= \| \boldsymbol{z} - \boldsymbol{H}\hat{\boldsymbol{x}} \| \leq \tau. \end{aligned} \tag{3}$$

Here, H is measurement Jacobian. An adversary can launch an attack on any state variable, or multiple states, by making sure that the attack vector $\boldsymbol{a}$ satisfies Eq. (3). Even with limited network information or access to limited smart meters, the FDIAs can be launched [9].

FDIAs also impact the real-time power market where SE is used to determine real-time locational marginal prices [10, 11]. Furthermore, coordinated FDIAs with physical attacks can also be launched that are capable of causing system-wide disruptions. The adversary with access to network topology can craft a load redistribution attack, thereby forcing the system operator to change the generator dispatch schedule, which can cause uneconomic operation of power systems, transmission line overloading [12], load shedding [13], and in the worst case, cascaded tripping of power system components [14].

3.1 Countermeasures

For protecting smart grids against cyberattacks, a smart security infrastructure is required. Smart security includes security of both OT and IT. Encryption, digital signature, and authentication codes can be embedded in the information security layer to ensure confidentiality, integrity, and availability of information to both the sender and receiver. IT infrastructure includes servers and routers. Protocols and servers must be secured to maintain secure communication between PMUs, RTUs, IEDs, and the control center. It will also help to avoid malicious code injection and Denial of Service (DoS) attacks. Countermeasures for FDIAs can be categorized as *detection-based defense* and *protection-based defense*.

3.1.1 *Protection of Critical Devices*

The electrical power grid is a huge network of transmission lines, generation plants, and substations. The complex interconnectivity, which continues to grow and expand, has thousands of metering devices located miles apart for continuous monitoring to control power system events in real time. The cost of securing all the metering devices can be massive. To minimize the cost of protection, it is necessary to identify and protect critical components to secure power system operations. Graph theoretical approaches are applied to identify critical metering devices [15, 16]. Metering devices can also be identified and protected by considering the interaction between the attacker and defender. The optimization problem can be modeled for minimizing the number of smart meters protected such that none of the states can be attacked [17].

3.1.2 *Moving Target Defense*

The moving target defense (MTD) can also be applied to deceive adversaries in real time. The topology of the network can be dynamically changed without affecting generator and load dispatch, which will minimize the risks of data integrity attacks. The system operator randomly uses multiple sets of measurements for estimating the states, as shown in Fig. 6. The attacker, unaware of measurement sets used by the control center, fails to launch the attacks [18].

The operator can dynamically change network parameters to detect FDIAs. Distribution FACTS (D-FACTS) devices can be used to slightly modify the line reactances. The integrity of measurements can be validated if a proportional change is observed in the measurements [19].

3.2 Detection of FDIAs

Analyzing smart meter/sensor data along with corresponding power system events helps in detecting FDIAs. The measurement pattern is usually observed and compared with the previous data to detect measurement anomalies. FDIAs may be random or targeted. For the random attack, contrasting patterns in measurements compared with historical measurements could easily separate attack measurement samples from true measurements. However, if the attacked measurements follow the same pattern, a measurement variation-based approach is used for detecting anomalies [20]. The measurement variation-based approach uses real-time and historical measurement variations to calculate relative entropy. For each sample, the relative entropy is compared with a predefined threshold for detecting FDIAs. To further improve detection efficiency, transformation techniques are used on the measurement variations before calculating relative entropy [21].

3.2.1 *Data-Driven Approach*

With more and more automation incorporated at the OT level, a huge amount of data is available at the control center. This data can be used to detect abnormal behavior in these smart CPSs. SCADA receives data from sensors and metering devices that is used for Wide Area Monitoring, Protection and Control (WAMPAC), and is also stored in the historical database. Machine learning approaches can be used for short-term forecasting of power system states to detect anomalies in real time [22]. The approach is effective against detecting measurement anomalies that could cause significant operational impacts.

An artificial neural network is a network designed to replicate the human brain to perform complex tasks. A properly trained neural network can solve complex real-world problems, such as forecasting and estimation. A basic neural network consists of weights, bias, adders, and an activation function. Out of various neural network designs, feed-forward (FF), along with back propagation (BP), is the most popular architecture among researchers. An

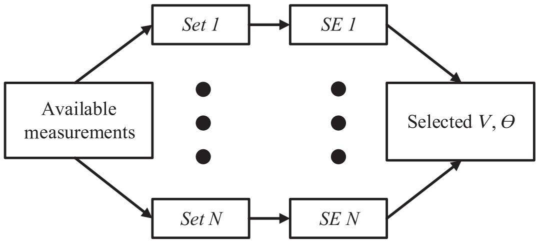


FIG. 6 Simple MTD strategy.

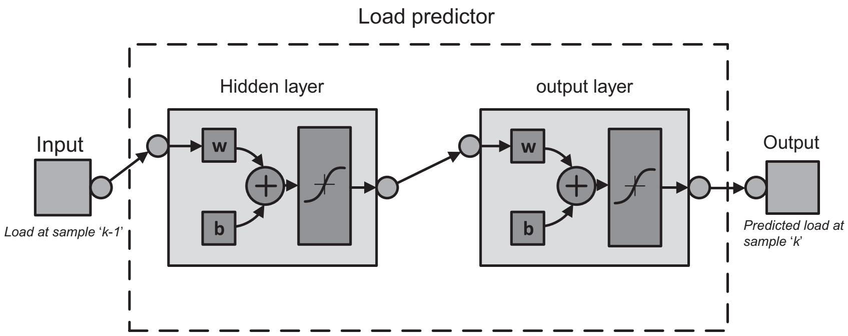


FIG. 7 An artificial neural network for load predictors.

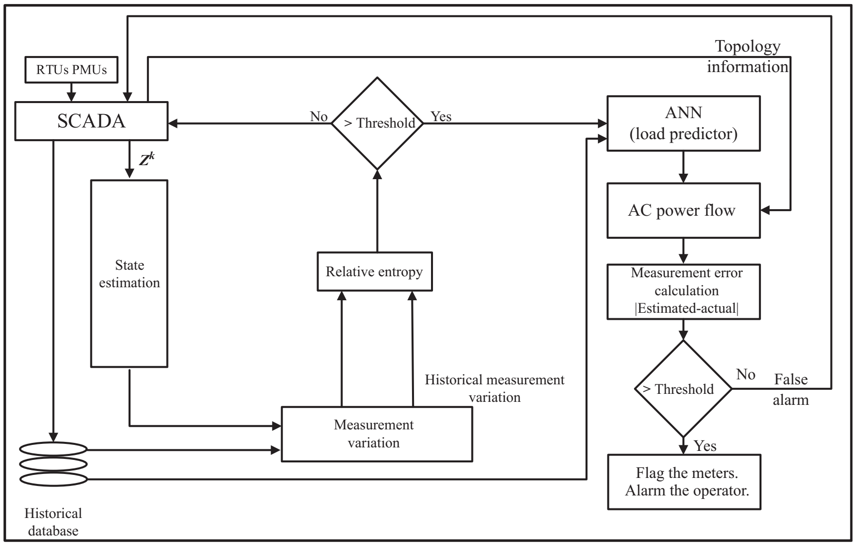


FIG. 8 Schematic for a data-driven approach to detect/identify FDIAs.

FFBP neural network [23] is used for estimating the load based on the loading condition of the preceding sample, as shown in Fig. 7. In the training phase, the neural network maps input-output for a given process. In this learning phase, the weights are optimized such that the error in the output is minimized. Once the training is complete, the network is tested with new inputs, and it is expected to have an accurate output.

For large power systems (with thousands of buses), the forecasting load for each sample can burden the SE process, which runs at 2-minute intervals. Therefore, the relative entropy-based detection strategy [21] can be used to raise an alarm in the event of attack detection. For the sample that is flagged as malicious, the load predictor strategy can be used for identifying compromised sensors [23]. The inputs to the load predictor are weather details (temperature, humidity, pressure, due point) for the present and previous sample, and demand at the previous sample. The output of the load predictor is the predicted demand for the present sample (the one flagged attacked during detection). The measurements for the predicted load are compared with actual measurements available from SCADA. The error is compared with a predefined threshold to identify the corrupted sensors. The complete schematic for the data-driven approach is shown in Fig. 8.

Once the attacked sensors are identified, the system operator can either replace the meters, use a different set of measurements, or discard the measurements, provided that the power system remains observable. The system operator can also use the pseudo-measurements to estimate the states and maintain observability. Moreover, strict measures can also be taken to ensure security once the sensors are identified.

4 SECURITY IN SMART CYBER-PHYSICAL SYSTEMS: SMART CARS

An unwanted aftereffect of automobiles becoming autonomous or semiautonomous is the potential for attackers to confuse or hack it. Researchers and hackers enlist various attacks that are possible on automobiles. We categorize these attacks under three different labels, which are described as follows.

1. *Direct physical access attacks*: In such attacks, the attacker will have complete access to the vehicle, and he or she can modify the software and hardware in it at his or her will. A potential attacker can be your mechanic or a car wash person. Hoppe et al. [24, 25] demonstrated many such attack scenarios. In the first attack scenario, an arbitrary code added to ECU resulted in a 0-day attack in which the car windows opened when its speed exceeded 200 kmph (kilometers per hour). In another attack, they demonstrated how airbag control systems can be removed from a car and could be masked using injected code. Other demonstrations include attacks on gateway ECUs and warning lights. Research from Koscher et al. [26] evaluated a car and performed extensive lab and on-road tests to demonstrate various attacks. On-road tests were performed at a decommissioned airport runway with a lot of safety precautions. Researchers manually introduced packets into the vehicular system to perform the attacks. Among the different attacks performed, some could be manually over-ridden, while others could not be. It is not very surprising that many of the attacks can be done on a running car also. Attacks demonstrated include frequent activation of lock relays and windshield wipers, trunk popping, permanent horn activation, disabling window and key lock relays, temporary RPM increase, idle RPM increase, prevention of braking, uneven engagement of car brakes, falsified speedometer readings, radio volume increase, car alarm honking, engine killing, and so forth. Some of the performed attacks are innocuous, but if used with a malicious intent, they can cause loss of reputation and trust. For example, if the car engine stops suddenly, the reliability aspect will be badly affected.
2. *Indirect physical access attacks*: In indirect physical access, the attacker cannot modify the hardware, but can introduce alien devices, such as CD, USB, and on-board diagnostic (OBD) devices. Devices that are connected to the OBD port of a vehicle with advanced external connectivity are a new trend in automobiles. The OBD port, being connected to the external CAN bus, can fetch various information about the car, including error codes. Most of these tools provide intelligent information about the car, and are connected to the Internet/cloud. Some of the popular devices include Automatic,[14] Progressive Snapshot,[15] Ford Reference VI,[16] Verizon Hum,[17] and so forth and a wide range of corresponding mobile phone applications. Miller and Valasek [27] demonstrated various attacks possible on cars using this OBD port. An OBD port is mandated by federal law in all cars in the United States, which has been in effect since 1996. It is generally used for monitoring emissions statistics by motor vehicle departments, checking the health of a car by mechanics, and so forth. They use ECOM cables to communicate with the CAN bus of two vehicles, the Ford Escape and Toyota Prius, and used tools, such as ECOMCat to read and write to the bus. Their research started with simple attacks such as displaying false data on the dashboard, and simple DoS attacks by overwhelming the bus. Eventually, they showed more advanced attacks such as disabling brakes, applying brakes, killing the engine, and so forth by introducing crafted packets to the CAN bus.
3. *External attacks*: In such attacks, attackers will make use of the external open interfaces or software vulnerabilities on an unaltered/factory condition vehicle. Researchers Miller and Valasek [28] famously hacked into an unaltered Jeep vehicle and controlled it remotely. The attack reportedly resulted in a 1.4 million vehicle recall by Chrysler. The Jeep is an advanced connected vehicle that has mobile connectivity built in using the Sprint network in the United States. Researchers reverse engineered and found that the D-Bus service (a software bus for interprocess communication and remote procedure calls) is running on the port 6667, and can be accessed anonymously. It will also allow anyone to run arbitrary code on the vehicle's head unit. Interestingly, they discovered that vulnerable vehicles can be found by just scanning the port 6667 on IP addresses starting with 21.0.0.0/8 and 25.0.0.0/8. The initial steps in the exploit chain include target identification (the IP address of the vehicle by scanning for port 6667) and exploitation of the OMAP chip in the head unit using D-Bus service. To control non-CAN features such as the radio, only these steps are necessary. However, in order to get control of the vehicle, the attacker can modify the v850 microcontroller firmware. Once done, the hacker is in a position to introduce arbitrary CAN messages to the vehicle, and hence can remotely control the vehicle. Despite fail-safe practices such as shutting down the system in case of irregularity, the attackers were able to perform malicious activities, such as killing the engine while running, locking and unlocking the car, activating the blinkers and windshield wipers, partially controlling the steering wheel (exploiting parking assist features, but only at low speeds), and so forth. Several other reports claim a crypto attack [29] on the keyless entry feature on cars from popular makers, including Volkswagen, Skoda, Volvo, and so forth.

[14] See https://www.automatic.com/.

[15] See https://www.progressive.com/auto/discounts/snapshot/.

[16] See https://shop.openxcplatform.com/ford-reference-vi.html.

[17] See https://www.hum.com/productstext.

4.1 Countermeasures

One of the core problems with the vehicular architecture is the presence of comparatively primitive bus technology, lacking cutting-edge security features, with far more sophisticated subsystems running on top. For example, the CAN bus is a broadcast-based bus in which all connected devices can see all the traffic, and there is no in-built concept of authentication or authorization. Any message on the wire is authenticated by default. This enables an attacker to introduce well-formed packets on to the wire to perform malicious activities. Research enlists two categories of techniques to secure the environment, which are described in the following sections.

4.1.1 Attack Prevention Techniques

The first approach is to prevent the attacks from happening. According to Wolfe et al. [30], a combination of hardware protection techniques, software protection techniques, and secure communication are required to achieve this. Newer protocols (varying in capabilities, speed, implementation overhead, and so forth) and the modification of existing protocols are proposed with this intent. LCAP [31] (Lightweight CAN Authentication Protocol) is one such protocol, with minimum overhead, that is a modification to the existing CAN network. The technique is based on a preshared key and magic number exchanged between the sender and receiver. At the beginning of a drive cycle, all devices perform an initial setup, and will have a session key, HMAC key, and a channel initial magic number, which are used for subsequent message transmissions. For data exchange, the sender will append the magic number to the message and encrypt it using a session key. The receiver will authenticate the message using the magic number after decrypting it. However, this protocol will have the overhead for cryptographic calculations.

CANAuth [32] is another proposed protocol that is backward compatible. This protocol also depends on a preshared key stored in a tamper-proof location with each of the entities. A session key will be generated using the preshared key and it will be used for authenticating messages. In CANAuth, considering the hard constraints on time and message length, the authentication data is transmitted out of band using the CAN+ [33] protocol. Each authentication message will have a 32-bit counter value to prevent replay attacks and an 80-bit signature with this protocol.

LiBrA-CAN [34] is another proposed protocol to prevent attacks that uses CAN-FD[18] (a newer CAN protocol with flexible data rates). LiBrA-CAN presents two new paradigms, namely, key splitting and MAC mixing for message authentication. The protocol can be used in a master oriented flavor (a node with higher computational capabilities is used for authentication), which includes centralized, cumulative, and load balanced authentication schemes, or distributed flavor, which includes two-stage authentication and multimaster authentication schemes. In comparison with other protocols, they have efficient forgery detection using MAC mixing and lesser authentication delays.

Apart from protocol-specific drawbacks such as time constraints, the requirement of preshared keys and higher computational power are some disadvantages of such techniques. Another major drawback for this class of techniques is that we can only protect future vehicles, not the present ones on the road. Moreover, the automotive domain is a huge industry, and it requires a long time to change. Koscher et al. [26] describe these stringent “operational and economic realities” in the automotive industry and explain the importance of a detection strategy for automobiles.

4.1.2 Attack Detection and Mitigation

The second strategy to make the automotive domain safe is attack detection and mitigation. Different companies have come up with automotive intrusion detection systems. Panasonic Corporation is developing an intrusion detection and prevention system[19] (IDPS) that combines host intrusion detection technology (which combines behavioral information), in-vehicle device-type intrusion detection technology (it monitors and detects unauthorized commands in the in-vehicle network, both CAN-based and Ethernet-based), and cloud-type vehicle intrusion detection technology (detects intrusion by analyzing logs collected from different vehicles using machine learning). Karamba Security[20] is another company that develops tools such as *Karamba Carwall* (it generates policies based on factory settings and embeds them into ECUs to continuously validate actions at runtime) and *Karamba SafeCAN* (it authenticates and hardens in-car network communications, helping ECUs to ignore commands from invalid ECUs or physical hacks). Escrypt[21] is yet another company that develops an IDPS for vehicles. Their products, *CycurIDS* and *CycurGUARD*, analyze data from multiple vehicles and detect potential intrusions. We can classify the basic research in this domain into two subcategories based on whether the technique uses a semantic understanding of the underlying data.

[18] See https://can-newsletter.org/assets/files/ttmedia/raw/e5740b7b5781b8960f55efcc2b93edf8.pdf.

[19] See http://itsworldcongress2017.org/wp-content/uploads/2017/11/kishikawa_20171025.pdf.

[20] See https://karambasecurity.com/products.

[21] See https://www.escrypt.com/en/solutions-overview.

4.1.2.1 Statistics-Oriented Techniques

Research enlists different techniques for attack detection and mitigation. One such research direction is by applying machine learning techniques on raw CAN messages without considering the semantics of the messages.

Muter et al. [35] proposed an entropy-based approach for anomaly detection in vehicles. In this technique, various information theoretic measures are used to calculate entropy. Entropy, in general, measures "how much coincidence" of a given dataset, or it represents the abstract representation of randomness. However, due to restricted vehicular network specifications, the amount of randomness (or entropy) is low. Hence, in their approach, they consider an adaptation of the existing entropy-based intrusion detection techniques. To calculate entropy, they use three levels of data abstraction. Binary-level data abstraction consists of raw ones and zeros. At this level, either a bitwise classifier (which considers each bit as an event) or x-bitwise classifier (a combination of x bits is considered as an event) can be used. The next level of abstraction is the signal level, in which an event is generated for every signal value of the CAN message. The assumption is that there are only fixed messages in a vehicular network unlike a general protocol such as TCP or UDP. The final level of abstraction is the protocol level. CAN defines 12 fields for data frames in the base format. The classifier in this level generates an event for every field in the CAN message. The entropy (Eq. 4) and relative entropy (Eq. 5) are calculated using standard information theoretic techniques. Their technique was put against simulated attack scenarios such as *increased frequency*, *message flooding*, and so forth, and showed that most attack scenarios caused variations in the calculated entropy, and could be utilized for detecting anomalies.

$$
\begin{aligned}
H(X) &= \sum_{x \in C_X} P(x) \log \frac{1}{P(x)} \\
&\text{where} \\
&C_x: \ \text{Set of classes for dataset } X, \\
&P(x): \ \text{Probability of } x \text{ in } X.
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
RelEnt(p/q) = \sum_{x \in C_X} p(x) \log \frac{p(x)}{q(x)} \\
&\text{where} \\
&C_x: \ \text{Set of classes for dataset} X, \\
&p(x), q(x): \ \text{Probability distribution over} x \in C_x.
\end{aligned}
\tag{5}
$$

Taylor et al.[36] use long short-term memory (LSTM) to detect anomalies. LSTM is used to learn long-term dependencies in sequences of data. In their research, they considered each bit on the CAN message as a feature for the LSTM and trained the network. Logically, once trained, the network will be able to predict the next CAN message, given a previous sequence of CAN messages. In this technique, they trained a separate LSTM network for each CAN message. For detecting an anomaly, a scalar anomaly score is required. To generate this score, they use the binary loss function, which is defined as

$$
\begin{aligned}
L\left(\hat{b}_k, bk\right) &= -\left(b_k \log\left(\hat{b}_k + \epsilon\right) + (1 - b_k) \log\left(1 - \hat{b}_k + \epsilon\right)\right) \\
&\text{where} \\
&b_k: \ k\text{th bit in the Message at step } i, \\
&\hat{b}_k: \ k\text{th bit's predicted value by the network}, \\
&\epsilon: \ \text{a fixed value that caps the maximum loss.}
\end{aligned}
\tag{6}
$$

The loss function will have a low value for incorrect and middling predictions and very high value for confident and incorrect predictions. The final anomaly score is calculated by combining the bit losses over the entire sequence using various strategies such as *maximum bit loss* (maximum of all bit losses over the entire sequence), *maximum word loss* (maximum in average bit loss over all words), *window max* (maximum in average bit loss over words in a window), *log window max* (log mean of average bit loss over words in a window), *sequence mean* (mean bit loss over the complete sequence), and so forth. To detect anomalies, an empirically found threshold will be used on the calculated scalar value.

In other parallel research, Kang et al. [37, 38] used deep neural networks (DNNs) directly on the bit stream. The motivation behind using a DNN is its ability to model nonlinear relationships. In their training phase, features representing the statistical behavior of the CAN packet are extracted. For feature extraction efficiency, they use each bit in the DATA field (64 bits) of a CAN packet as features. To reduce the dimension, they propose splitting the DATA field into mode information and value information depending on the CAN message ID and avoid using unwanted bits from training. Now using these extracted features, they train a deep neural network with the input layer's size as the number of extracted features, and the output layer with two neurons. In between, the network will have a fixed number of neurons. The two output neurons represent an attack packet and a normal packet, respectively. That is, if the first neuron is activated, it signifies that it is an attack packet, and if the other neuron is activated, it signifies a normal packet. In comparison with an FF network, they were able to get a better performance on their evaluation with data generated using open car test-bed and network experiments (OCTANE) [39].

4.1.2.2 Semantics-Oriented Techniques

All the preceding techniques consider a CAN packet as a flow of structured bit streams of messages. In semantics-oriented techniques, the semantics of the messages are considered. It implies that each CAN message is first interpreted as a state of the automobile, or action performed on it.

Narayanan et al. [40] hypothesize that the flow of packets in a car is not just a flow of independent messages, but they form the state of that vehicle. As a result, we can extract the sequence of observations from this bit stream. For example, $O = O_1, O_2, O_3, \ldots, O_t, \ldots, O_n$ can be a sequence of n observations extracted from a car, where O_t is the observation at time t. Each observation $O_t \in O$ is a vector $O_t = \langle v_{t,1}, v_{t,2}, v_{t,3}, \ldots, v_{t,i}, \ldots, v_{t,m} \rangle$, where $v_{t,i}$ represents the value of a sensor/ECU i at time t. For example, consider only two sensors (the speed sensor and door sensor), and the following sequence of observations from O_1 to O_6 under three different conditions.

Normal Sequence	**Abnormal Sequence$_1$**	**Abnormal Sequence$_2$**
$O_1 \leftarrow \langle v_{1,1} = 0, v_{1,2} = Closed \rangle$	$O_1 \leftarrow \langle v_{1,1} = 20, v_{1,2} = Closed \rangle$	$O_1 \leftarrow \langle v_{1,1} = 20, v_{1,2} = Closed \rangle$
$O_2 \leftarrow \langle v_{2,1} = 0, v_{2,2} = Open \rangle$	$O_2 \leftarrow \langle v_{2,1} = 22, v_{2,2} = Closed \rangle$	$O_2 \leftarrow \langle v_{2,1} = 22, v_{2,2} = Closed \rangle$
$O_3 \leftarrow \langle v_{3,1} = 0, v_{3,2} = Closed \rangle$	$O_3 \leftarrow \langle v_{3,1} = 25, v_{3,2} = Open \rangle$	$O_3 \leftarrow \langle v_{3,1} = 25, v_{3,2} = Closed \rangle$
$O_4 \leftarrow \langle v_{4,1} = 2, v_{4,2} = Closed \rangle$	$O_4 \leftarrow \langle v_{4,1} = 28, v_{4,2} = Open \rangle$	$O_4 \leftarrow \langle v_{4,1} = 85, v_{4,2} = Closed \rangle$
$O_5 \leftarrow \langle v_{5,1} = 5, v_{5,2} = Closed \rangle$	$O_5 \leftarrow \langle v_{5,1} = 30, v_{5,2} = Closed \rangle$	$O_5 \leftarrow \langle v_{5,1} = 28, v_{5,2} = Closed \rangle$
$O_6 \leftarrow \langle v_{6,1} = 7, v_{6,2} = Closed \rangle$	$O_6 \leftarrow \langle v_{6,1} = 34, v_{6,2} = Closed \rangle$	$O_6 \leftarrow \langle v_{6,1} = 25, v_{6,2} = Closed \rangle$

The first column under *Normal Sequence* represents a normal scenario of events in which a passenger enters the car while the speed remains 0. However, consider the sequence of observations under *Abnormal Sequence$_1$*. While the speed is increasing from 20 to 34, a door is reported as open, which is against general logic. If the sensor values are taken separately, they look normal, but taking them together makes it look abnormal. Yet another example is given under the sequence *Abnormal Sequence$_2$* in which the speed is showing a sudden spike from 25 → 85 and then 85 → 28 which is very abnormal in itself.

Hidden Markov models [41] (HMM) are used to find the posterior probability of a sequence. In this technique, an HMM model is trained from the normal sequence of observations extracted from the CAN bit stream. During the anomaly detection phase, the incoming bit stream is first converted to a sequence of observations. Using a window size (representing the number of observations need to be considered at one go), the posterior probability of the sequence is determined. An empirically determined threshold is then used to detect anomalous windows of observations. In the evaluation of the system, different single sensor and multisensor anomalous scenarios were easily detected.

```
HASCOMPONENT (VEHICLE, ENVIRONMENTLIGHTSENSOR) ˆ HASCOMPONENT (VEHICLE,
    VEHICLESPEEDSENSOR) ˆ HASCOMPONENT (VEHICLE, HEADLIGHT) ˆ
    HASSTATEVALUE (ENVIRONMENTLIGHTSENSOR,  "LOW") ˆ HASSTATEVALUE
    (HEADLIGHT,  "OFF") ˆ HASSTATEVALUE (VEHICLESPEED,  "HIGH") ==>
    HASSTATEVALUE (CURRENTVEHICLESTATE,  "ANOMALOUS NO LIGHT")}
```

In another work, Narayanan et al. [42] used semantic web technologies to detect various attacks. They used logical reasoning over a knowledge base and current sensor values to extract context from data. Their proposed architecture has two components, a cross-component inferencing engine and a local context detection (LCD) layer. The cross-component inferencing engine internally uses an ontology to represent the domain knowledge. The ontology is an extended version of IOT-lite ontology with new attributes, such as *senseAttribute* and *hasStateValue*. It is then populated with domain-specific instances and domain-specific rules (specified using SWRL [43]). The LCD layer extracts valid messages from the bit stream and further processes them to generate local contexts. Some examples of local context include *highSpeed*, *normalSpeed*, *suddenAcceleration*, and so forth. Technically, any machine learning technique could be deployed to generate local contexts. Now a reasoner will reason over these local contexts to detect anomalous situations. Domain experts can also write rules on top of this ontology using SWRL. An example SWRL rule that suggests driving without headlights while lighting conditions are poor is presented herein. The main challenge in this technique is to aggregate all the domain knowledge rules, and this research proposes to develop a rule mining engine from available normal data.

5 CONCLUSION AND FUTURE DIRECTIONS

Smart CPSs are essential components of Smart Cities, and their secure operation is critical to maintaining integrity and trustworthiness of the entire infrastructure. Section 2 described the architecture of two such smart CPSs. We can find many similarities between these domains.

- Both CPSs have a large number of sensors connected to the network. In a smart car, the interconnection network is within the car, while the interconnection network of smart grids spans across a much larger landscape.
- Both CPSs take smart decisions after complex calculations on real-world data collected from these sensors.
- Another similarity in both the domains is the constraints they had while developing the system. In both smart cars and smart grids, the need for supporting legacy systems is vital. Otherwise, it will be a substantial financial burden to replace all existing infrastructure. In smart grids, replacing the entire existing infrastructure with more advanced and secure infrastructure is nearly impossible because of this. Similarly in smart cars, to change the ECUs and different ECU components in a car, the century-old automotive industry might need to disrupt its assembly line and subcomponent manufacturers (and bear the associated cost).
- Due to the presence of constraints in their architecture, the system designers did not implement robust, cryptographically secure protocols for their communication. Attackers utilized this as a vulnerability point. For example, because most cars use less secure broadcast buses internally, any message on the CAN is directly authenticated and authorized. The system cannot differentiate between a message from an attacker and a message from a legitimate ECU. Similarly, in the smart grid domain, smart meters at home can act as an entry point for the adversary to trace back to the control center and inject malicious code to manipulate the operation at his or her will.
- Often classic cybersecurity techniques (the same as any other cyber domains) are utilized to access the CPS network. For example, in the Ukrainian power grid attack, the attackers used spear phishing and stealing valid credentials to gain access to the network. Similarly, in the Jeep hack of 2016, researchers used an open Dbus service to gain initial access to the car's internal network.
- A full-scale attack on CPSs can cause financial damage, reputation loss, and large-scale service disruptions. The financial implications of recalling 1.4 million vehicles after the Jeep hack was reported, and large-scale power outages caused by the Ukrainian smart grid hack are authentic testimonies for this.
- Secure protocols and control units are solutions in both these domains. However, the requirement to support legacy systems makes detection strategies vital in both domains. Even after complete migrations to newer and more secure protocols, detection strategies will remain significant, because a determined attacker is always out there trying to find ways to get into the system.

Researchers demonstrated various attacks on these CPS domains, which are detailed in Sections 3 and 4. Many of the current countermeasures proposed to defend attacks in both these domains use a data-driven approach. However, these techniques either consider values from different streams separately, or do not consider the semantics of the system, and overlook certain stealthy attacks. If domain semantics are avoided, a determined attacker can utilize this to find ways past their detection. For example, to evade the detection of fault data injection, a skilled attacker injects errors only in small magnitudes, which will not raise alarm. To see the effect of ignoring the interrelationship between sensors, consider the scenario in which the speed of a car is increasing gradually. This is not an abnormal scenario until there is an obstruction in front of it. Similarly, in the smart grid domain, a fault in a transmission line will cause transience in voltages and currents, but line overloading will exist only when there is more consumption of power in the nearby buses.

Future research on these domains will stem from the fact that they share many characteristics, problems, and scenarios. A study on the adaptability of techniques applicable in one domain to another is an interesting avenue to explore. Another path to consider is the development of a human-like learning and monitoring system. It is clear that the attacker's intention is always to disrupt the normal behavior of the system (in both smart grids and smart cars). For example, in the Ukrainian smart grid attack, the attackers went on to open the CBs for multiple substations together, and in the Jeep hack, hackers killed the engine of a moving car. These situations are improbable in a normal environment, and could be easily detected by a human monitoring the system. In the case of humans, we interpret context from all the measurements and analyze the situation. We can also learn from previous experiences. Holistically, a more robust risk assessment model, and techniques that can learn and adapt on their own are required to protect the individual ecosystems and infrastructure. Exploring more advanced machine learning techniques such as convolutional neural networks and adversarial networks, can develop such systems. Such techniques can also provide common domain-independent learning systems that can automatically extract context from different sensor data and detect attacks.

References

[1] C. Harrison, B. Eckman, R. Hamilton, P. Hartswick, J. Kalagnanam, J. Paraszczak, P. Williams, Foundations for smarter cities, IBM J. Res. Dev. 54 (4) 2010 1–16.

[2] Defense Use Case, Analysis of the Cyber Attack on the Ukrainian Power Grid, Electricity Information Sharing and Analysis Center (E-ISAC), 2016.

[3] F.C. Schweppe, Power system static-state estimation, Part I, II, III, IEEE Trans. Power App. Syst. 89 (1) 1970 120–135.

[4] A. Abur, A. Gomez-Exposito, Power System State Estimation: Theory and Implementation, CRC Press, 2004.

[5] X. Jiang, J. Zhang, B.J. Harding, J.J. Makela, A.D. Domı, et al., Spoofing GPS receiver clock offset of phasor measurement units, IEEE Trans. Power Syst. 28 (3) 2013 3253–3262.

[6] Z. Zhang, S. Gong, A.D. Dimitrovski, H. Li, Time synchronization attack in smart grid: impact and analysis, IEEE Trans. Smart Grid 4 (1) 2013 87–98.

[7] O. Kosut, L. Jia, R.J. Thomas, L. Tong, Malicious data attacks on the smart grid, IEEE Trans. Smart Grid 2 (4) 2011 645–658.

[8] Y. Liu, P. Ning, M.K. Reiter, False data injection attacks against state estimation in electric power grids, ACM Trans. Inf. Syst. Secur. 14 (1) (2011) 13.

[9] X. Liu, Z. Li, Local load redistribution attacks in power systems with incomplete network information, IEEE Trans. Smart Grid 5 (4) 2014 1665–1676.

[10] A.L. Ott, Experience with PJM market operation, system design, and implementation, IEEE Trans. Power Syst. 18 (2) 2003 528–534.

[11] T. Zheng, E. Litvinov, Ex post pricing in the co-optimized energy and reserve market, IEEE Trans. Power Syst. 21 (4) 2006 1528–1538.

[12] X. Liu, Z. Li, Trilevel modeling of cyber attacks on transmission lines, IEEE Trans. Smart Grid (99) 2015. https://doi.org/10.1109/TSG.2015.2475701.

[13] Y. Yuan, Z. Li, K. Ren, Quantitative analysis of load redistribution attacks in power systems, IEEE Trans. Parallel Distrib. Syst. 23 (9) 2012 1731–1738.

[14] K. Khanna, B.K. Panigrahi, A. Joshi, Bi-level modelling of false data injection attacks on security constrained optimal power flow, IET Gener. Transm. Distrib. 11 (14) 2017 3586–3593.

[15] R.B. Bobba, K.M. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, T.J. Overbye, Detecting false data injection attacks on DC state estimation, in: Preprints of the First Workshop on Secure Control Systems, CPSWEEK, 2010, 2010. vol.

[16] S. Bi, Y.J. Zhang, Defending mechanisms against false-data injection attacks in the power system state estimation, in: 2011 IEEE GLOBECOM Workshops (GC Wkshps), IEEE, 2011, pp. 1162–1167.

[17] R. Deng, G. Xiao, R. Lu, Defending against false data injection attacks on power system state estimation, IEEE Trans. Ind. Inf. 13 (1) 2017 198–207.

[18] Y. Yao, Z. Li, MTD-inspired state estimation based on random measurements selection, in: 2016 North American Power Symposium (NAPS), IEEE, 2016, pp. 1–6.

[19] K.R. Davis, K.L. Morrow, R. Bobba, E. Heine, Power flow cyber attacks and perturbation-based defense, in: 2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)2012, pp. 342–347.

[20] G. Chaojun, P. Jirutitijaroen, M. Motani, Detecting false data injection attacks in AC state estimation, IEEE Trans. Smart Grid 6 (5) 2015 2476–2483.

[21] S.K. Singh, K. Khanna, R. Bose, B.K. Panigrahi, A. Joshi, Joint-transformation-based detection of false data injection attacks in smart grid, IEEE Trans. Ind. Inf. 14 (1) 2018 89–97.

[22] A. Ashok, M. Govindarasu, V. Ajjarapu, Online detection of stealthy false data injection attacks in power system state estimation, IEEE Trans. Smart Grid (99) 2017, https://doi.org/10.1109/TSG.2016.2596298.

[23] K. Khanna, B.K. Panigrahi, A. Joshi, AI-based approach to identify compromised meters in data integrity attacks on smart grid, IET Gener. Transm. Distrib. 12 (5) 2018 1052–1066.

[24] T. Hoppe, J. Dittman, Sniffing/replay attacks on CAN buses: a simulated attack on the electric window lift classified using an adapted CERT taxonomy, in: Proceedings of the 2nd Workshop on Embedded Systems Security (WESS), 2007, pp. 1–6.

[25] T. Hoppe, S. Kiltz, J. Dittmann, Security threats to automotive CAN networks-practical examples and selected short-term countermeasures, in: Computer Safety, Reliability, and Security, Springer, 2008, pp. 235–248.

[26] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, et al., Experimental security analysis of a modern automobile, in: 2010 IEEE Symposium on Security and Privacy (SP), IEEE, 2010, pp. 447–462.

[27] C. Miller, C. Valasek, Adventures in automotive networks and control units, in: DEF CON 21 Hacking Conference, Las Vegas, NV, 2013.

[28] C. Miller, C. Valasek, Remote exploitation of an unaltered passenger vehicle, Tech. Rep., Blackhat, 2015.

[29] D. Storm, Hack to steal cars with keyless ignition: Volkswagen spent 2 years hiding flaw. Available from: http://www.computerworld.com/article/2971826/cybercrime-hacking/hack-to-steal-cars-with-keyless-ignition-volkswagen-spent-2-years-hiding-flaw.html.

[30] M. Wolf, A. Weimerskirch, T. Wollinger, State of the art: embedding security in vehicles, EURASIP J. Embed. Syst. 2007 (1) 2007 074706.

[31] A. Hazem, H.A. Fahmy, LCAP-A lightweight CAN authentication protocol for securing in-vehicle networks, in: 10th ESCAR Embedded Security in Cars Conference, Berlin, Germany, 6, 2012. vol.

[32] A. Van Herrewege, D. Singelee, I. Verbauwhede, CANAuth—a simple, backward compatible broadcast authentication protocol for CAN bus, in: ECRYPT Workshop on Lightweight Cryptography, 2011.

[33] T. Ziermann, S. Wildermann, J. Teich, CAN+: a new backward-compatible Controller Area Network (CAN) protocol with up to 16× higher data rates, in: Proceedings of the Conference on Design, Automation and Test in Europe, European Design and Automation Association, 2009, pp. 1088–1093.

[34] B. Groza, S. Murvay, A. Van Herrewege, I. Verbauwhede, LiBrA-CAN: lightweight broadcast authentication for controller area networks, ACM Trans. Embed. Comput. Syst. 16 (3) 2017 90.

[35] M. Müter, N. Asaj, Entropy-based anomaly detection for in-vehicle networks, in: 2011 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2011, pp. 1110–1115.

[36] A. Taylor, S. Leblanc, N. Japkowicz, Anomaly detection in automobile control network data with long short-term memory networks, in: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2016, pp. 130–139.

[37] M.-J. Kang, J.-W. Kang, A novel intrusion detection method using deep neural network for in-vehicle network security, in: 2016 IEEE 83rd Vehicular Technology Conference (VTC Spring), IEEE, 2016, pp. 1–5.

[38] M.-J. Kang, J.-W. Kang, Intrusion detection system using deep neural network for in-vehicle network security, PLoS ONE 11 (6) 2016 e0155781.

[39] C.E. Everett, D. McCoy, OCTANE (Open Car Testbed and Network Experiments): bringing cyber-physical security research to researchers and students, in: CSET, 2013.

[40] S.N. Narayanan, S. Mittal, A. Joshi, OBDSecureAlert: an anomaly detection system for vehicles. in: 2016 IEEE International Conference on Smart Computing (SMARTCOMP), 2016, pp. 1–6, https://doi.org/10.1109/SMARTCOMP.2016.7501710.

[41] L. Rabiner, B. Juang, An introduction to hidden Markov models, IEEE ASSP Mag. 3 (1) 1986 4–16.

[42] S. Nair, S. Mittal, A. Joshi, Using semantic technologies to mine vehicular context for security, in: 37th IEEE Sarnoff Symposium, 2016.

[43] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, et al., SWRL: a semantic web rule language combining OWL and RuleML, W3C Member Submission 21 2004 79.