Secure Abstract and Radar Sensor Patterns

BIJAYITA THAPA, Florida Atlantic University
EDUARDO B. FERNANDEZ, Florida Atlantic University

With the increasing use of the Internet of Things (IoT) and Cyber-Physical Systems (CPSs), sensors are used extensively in vehicle systems, smart health care systems, smart buildings and cities, and many other types of applications. At the same time, the extensive use of sensors and their increasing connectivity make them susceptible to a large variety of threats. In particular, a wide variety of sensors are needed in autonomous vehicles. All these sensors must have a fundamental structure and operations derived from an abstract security pattern (ASP), that defines a prototype conceptual secure sensor. An ASP is a security pattern that describes basic security mechanisms at an abstract level that realizes one or more security policies to handle a threat or comply with a security-related regulation or institutional policy, without any implementation aspects. An efficient way to produce concrete patterns is to start from an ASP, and add to each derived pattern the new forces and security controls needed for their specific context. From ASPs we can derive concrete patterns and we present here a Secure Sensor ASP from which we derive a Secure Radar Sensor pattern.

Categories and Subject Descriptors: D.2.11 **[Software Engineering]** Software Architectures–Patterns; D.5.1, D.4.6 **[Security and Protection]**

General Terms: Design

Additional Key Words and Phrases: abstract security pattern, concrete security pattern, internet of things, cyber-physical system, sensors, radar

ACM Reference Format:

Thapa, B. and Fernandez, E. B., Abstract and vehicle secure sensor patterns. Procs. of the 28th Conf. on Pattern Lang. of Prog. (Plop'21), October 2021, 11 pages.

1. INTRODUCTION

With the increasing use of the Internet of Things (IoT) and Cyber-Physical Systems (CPSs), sensors are used extensively in all kinds of applications; in particular, autonomous vehicle systems, smart health care systems, smart buildings and cities, and many other types. The internet has allowed a wide range of various types of devices to connect, communicate, and collaborate, either among themselves or with more complex systems. At the same time, the wide range of the use of sensors and their increasing connectivity enlarges their attack surface and make them susceptible to a large variety of threats. Because sensors capture physical quantities, attacks to them can produce severe consequences for the control of physical systems, including loss of safety and privacy.

In practice, a variety of sensors are needed. All secure sensors have a fundamental structure and operations derived from an abstract secure sensor pattern, a type of abstract security pattern (ASP). An ASP is a security pattern that describes basic security mechanisms at an abstract level, that realizes one or more security policies to handle a threat or comply with a security-related regulation or institutional policy without any implementation aspects (Fernandez et al., 2008, 2014). From an ASP we can build a family of patterns, connected using UML operations, frequently a generalization hierarchy. We call this structure a Security Solution Frame (SSF) (Uzunov et al. 2015). Starting from an ASP we can produce concrete patterns by adding to each derived pattern the new forces and security controls needed for their specific context, which may add new threats.

While the concept of ASP has been used to describe security mechanisms, we can also talk of Abstract Conceptual Patterns (ACPs), a type of analysis pattern that describes abstract conceptual entities; in this case, conceptual models of sensors. A Sensor ACP would describe those units and operations that every sensor must have. Adding security defenses to a Sensor ACP would produce a Sensor ASP.

Author's address: Bijayita Thapa, Dept. of Computer and Electrical Eng. and Computer Science, Florida Atlantic University, 777 Glades Rd., Boca Raton, FL33431, USA; email: bthapa@fau.edu; Eduardo B. Fernandez (corresponding author), Dept. of Electrical Eng. and Computer Science, Florida Atlantic University, 777 Glades Rd., Boca Raton, FL33431, USA; email: fernande@fau.edu

We present here an ASP for sensors from which we derive a Secure Radar Sensor pattern, part from a sensor-based SSF; the Sensor ASP is derived from a Sensor ACP. These patterns are part of a project on IoT security patterns, where a subproject models autonomous cars. The pattern diagram of Fig. 1 shows a partial ecosystem of these patterns; the patterns in this paper are shown in color. A pattern diagram (Buschmann et al., 1996) shows relationships between patterns; patterns are shown as rounded rectangles where an arc from pattern A to pattern B indicates the contribution of pattern A to pattern B. In Fig. 1, the UML generalization symbol indicates that the Radar Sensor is derived from the Secure Sensor ASP. The intents of the patterns in Fig. 1 are:

Secure Sensor (this paper): A Secure Sensor measures physical values from its environment and converts them into digital data that can be sent to other components in a secure way. This is an ASP, used as reference for the Secure Radar Sensor.

Secure Radar Sensor (this paper): A type of sensors that uses radio waves to determine the distance (range), angle, or velocity of objects, protected against threats that may change their readings.

Secure IoT Thing (Fernandez et al. 2021). An entity in the IoT that controls actuators and receives feedback from sensors where its data and interactions are secured by defending against its known threats.

Secure Cloud. A Cloud Security Reference Architecture (SRA) describes its threats and implements corresponding defenses

Secure Cloud-Based IoT Architecture (Fernandez 2020): Define security protection for data assets and for the communication channels in order to neutralize the system threats and reduce the complexity of managing security. The typical defenses include authentication, authorization, security logger/auditor, and secure channel.

Secure Fog. Fog Computing is a virtualized platform that stands between cloud computing systems and Internet devices, providing to these computational, storage, and networking services and allowing a cloud to control and communicate with these devices and to the devices to send data to the fog or the cloud (Syed et al. 2016).

Secure Actuator. A component of a system that can securely control a physical system (Orellana et al., 2021).

Secure Sensor Node. Describe the architecture of a unit intended to sense, store, and communicate local information about a physical environment. For those purposes the node architecture includes sensors, memory, and communication channels and their data and communication channels are protected (Orellana et al., 2020).

Our audience includes IoT device manufacturers, IoT application designers, developers of cloud/fog systems that may need to interact with IoT systems, security researchers, and students. We use a POSA template for describing this pattern (Buschmann et al., 1996.), extended with sections on security.

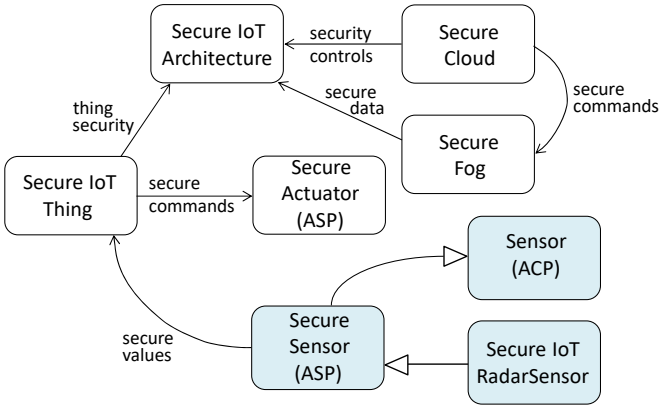

Figure 1: Pattern diagram of IoT secure architecture ecosystem.

2. SECURE SENSOR

2.1 Intent

A Secure Sensor measures physical values from its environment and converts them into digital data that can be sent to other components in a secure way.

2.2 Context

We consider an environment including cyberphysical and IoT systems. In these systems we need to detect physical quantities to be used by a control or a logging system.

Figure 2 presents a class diagram of the sensor ACP. The Sensor (sensing unit), collects a signal representing some physical measure and sends it to an analog-to-digital converter (ADC). The resulting digital signal can be captured through an Interface that communicates with other system components. This interface can also be used to receive commands to activate/deactivate the sensing unit.

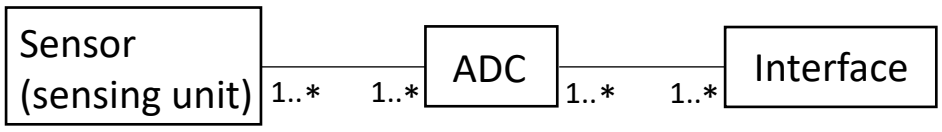

Figure 2. Class diagram of the Sensor ACP.

Fig. 3 shows the sequence diagram of the use case “Send Digital Data”. The scenario is:
Precondition: the sensor has been activated (another use case)..

1. The sensing unit measures an analog physical signal and sends it to the ADC,
2. The ADC converts this quantity to digital data.
3. The ADC sends the digital data to the interface.
4. The interface sends the digital data to other units in the system.

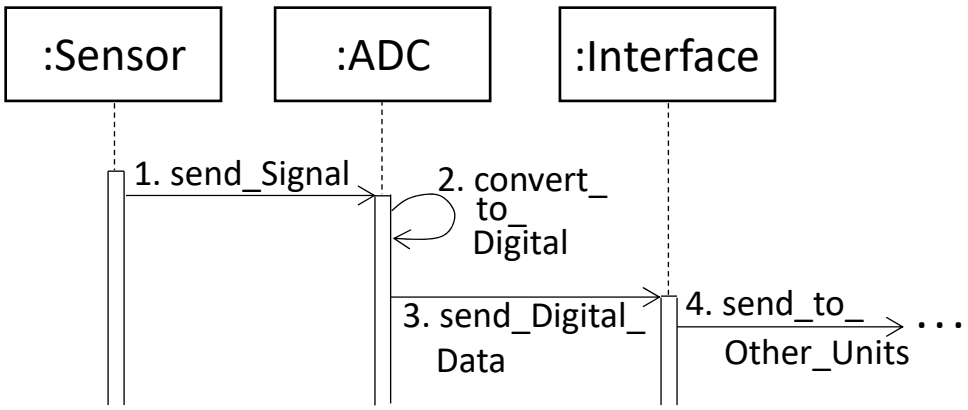

Figure 3. Sequence diagram of use case “Send Digital Data”.

2.3 Problem

How to make sure that all the readings from sensors are legitimate? If the sensor readings are modified or prevented from reaching their controllers they could make the whole system fail. For example, if a sensor in a wearable device is compromised, it could be life threatening to its carrier.

A possible solution to this problem is guided by the following forces:

- *Threat handling*. The solution must be able to stop or mitigate all the identified threats.
- *Diversity of sensor origins*. Applications may use sensors coming from different vendors and possibly with different owners. Some of these owners may be malicious. We also need an effective identification strategy to distinguish each individual sensor.
- *Sensor heterogeneity*. A basic sensor just measures and converts its values into digital values. Depending on their use, sensors may have memory, different interfaces, a processor, a transceiver, etc. The solution must contain the essential features of all these varieties.
- *Vulnerability of sensors*. Many sensors are inexpensive and have low computational power. This means that they may not have elaborated defenses; for example, if they use cryptography, it is a lightweight version. However, we still may need them to be secure to use them in possibly critical applications.
- *Tradeoffs.* There should be tradeoffs between security and cost, functionality, or performance.

2.4 Threats

The threats for sensors are:

- T1. Interception: The attacker intercepts or eavesdrop sensor outputs to gain access to sensitive data.
- T2. Tampering: The attacker modifies the sensor data to produce disruption in the system.
- T3. Flooding: The attacker repeatedly generates requests to prevent the sensor to send its data to the other parts of the system.
- T4.Man-in-the-middle: The attacker performs a man-in-the-middle attack (Umamaheshwari and Swaminathan 2018), intercepting sensor outputs, and changing them, to force actuator devices which interact with a physical environment to perform unwanted actions.
- T5: Continuous activation: The attacker keeps the sensor continuously active to make it exhaust its battery (this attack only applies to wireless sensors).
- T6: Physical damage: An intruder can physically damage the sensors if they are physically accessible.
- T7: Signal jamming: By jamming radio-frequency signals, an attacker can interrupt the transmission of data. Again, this attack only applies to wireless sensors.

2.5 Solution

In the basic sensor we only need the sensor itself, a power supply, an analog-to-digital converter (ADC), and interfaces to the rest of the system. As shown in 2.4, even those simple functions can be threatened, and we need defenses.

2.5.1 Countermeasures and secure system

Fig. 4 shows the class diagram of the Secure Sensor ASP, where the following defenses have been applied:

- T1. Interception: Use a network protocol such as TLS for mutual authentication. This requires adding an Encryptor/Decryptor to the Interface. This defense does not work if the sensor unit can be physically accessible or the sensor does not have enough computational power to support this protocol.
- T2. Tampering: Same defense as for T1.
- T3. Flooding: Use whitelisting (Fernandez et al. 2021). In this way, the sensor only communicates with specific units. This requires adding a Filter to the Interface.
- T4. Man-in-the-middle: TLS will avoid this attack too.
- T5. Continuous activation: Whitelisting can also work here. Again, this requires adding a Filter to the Interface.
- T6. Physical damage: Keep the sensors in a protected area; if not possible, there is no solution except reconfiguration of the network to replace the damaged sensor.
- T7. Signal jamming: Use a fixed connection. Sensors that use radio frequencies such as radars use more elaborated ways (see Section 3.5.1)

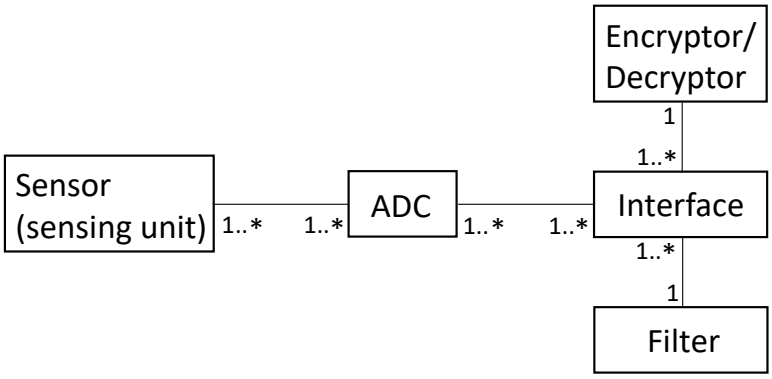

Figure 4. Class diagram of the Secure Sensor pattern.

2.5.2 Attack example

Fig. 5 shows the modus operandi of the attack "Intercept and tamper with the sensor output". The steps are the same as those in Fig. 3 until step 4. Step 5 shows a hacker intercepting the sensor output and modifying it before sending it to the other units. Its defense, as indicated above, is to encrypt the interface output; however, the hacker can still delete the output data, performing a DoS attack.

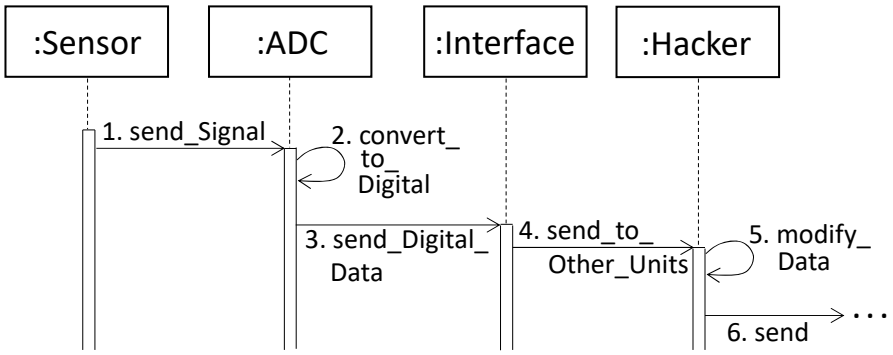

Figure 5. Attack "Intercept and tamper with the sensor output"

2.6 Implementation

For the simpler sensors, the encryptor/decryptor and the filter could be implemented as coprocessors.

2.7 Known Uses

The Secure Sensor is present in many control systems.

2.8 Consequences

This pattern has the following advantages:

- *Threat handling*. The solution is able to handle all the identified threats.
- *Diversity of sensor origins*. A secure protocol can consider these differences. We can use identity patterns to keep track of sensors (see Related patterns).
- *Sensor heterogeneity*. We are considering here only the threats to the most basic type of sensor. A derived pattern can consider additional threats.

- *Vulnerability of sensors*. The basic and inexpensive sensors can be extended with coprocessors.
- *Tradeoffs.* Sensors used in critical applications would use more security defenses at the expense of cost, functionality, or performance.

This pattern has the following disadvantages: As indicated, more elaborated sensors are more expensive. Security always incurs in some overhead. The complexity of the system also increases. For wireless devices, security also contributes to the exhaustion of their batteries.

2.9 Related Patterns

- A pattern for a Sensor Node (Sahu et al. 2010), describes the architecture and dynamics of the sensor ACP.
- A pattern for a Secure Sensor Node (Orellana et al. 2020), describes a sensor node that includes a processor, memory, and a transceiver.
- A pattern for Sensor Network Architectures describes the architecture of a wireless sensor network (Cardei et al. 2011).
- Sensor Node Design Pattern (Saida et al. 2018), is a generic design pattern intended to model the architecture of a wireless sensor node with real-time constraints; it is designed and annotated using the UML/MARTE standard.
- Identity patterns are described in (Fernandez 2013).
- Interface patterns are discussed in (M. Brambilla et al. 2017).
- Encryption patterns are shown in (Fernandez 2013).
- The Secure Radar Sensor, shown in the next section, is derived from this pattern.

3. SECURE RADAR SENSOR

3.1 Intent

A Secure Radar is a type of sensor that uses radio waves to determine the distance (range), angle, or velocity of physical objects, protected against a variety of threats.

3.2 Context

There are various types of radar that are used for many different applications, such as to monitor motion, distance, location, perception, etc. For example, Doppler radar is used to in e-healthcare systems to monitor elderly population, radars are used for traffic control In airports, to track arriving and leaving aircraft. Some may be installed in isolated places and without supervision, which makes them more vulnerable to physical attacks. Fig. 4 shows the class diagram of a typical radar system. The main components of the basic form of a radar are Transmitter, Antenna, and Receiver. The Transmitter generates electromagnetic (radio) waves with the help of the Waveform Generator. The Antenna transmits the generated waves to detect objects in the surrounding environment, and the Receiver receives the reflected waves (also known as echoes) to measure several possible variables. When to transmit and receive waves is controlled by the Transmit/Receive Switch (also known as duplexer). The Analog-to-Digital Converter (ADC) transforms the analog echoes to digital echoes (Shengjun and Shaochang 2017). The digital form of the echoes is transmitted to the Signal Processor, which uses the Pulse Compressor and the Doppler Processor for better resolution and for separating targets from clutter. The processed data is accessible in the Interface for use by external components.

3.3 Problem

Wrong readings could produce serious problems; for example, in aircraft navigation the pilot would be confused and ignore close aircraft. Other misuses include object hiding or addition. How do we protect radar measurements? All the forces of the Secure Sensor ASP apply to radar sensors, modified according to their specific context. A possible solution to this problem is guided by the following forces:

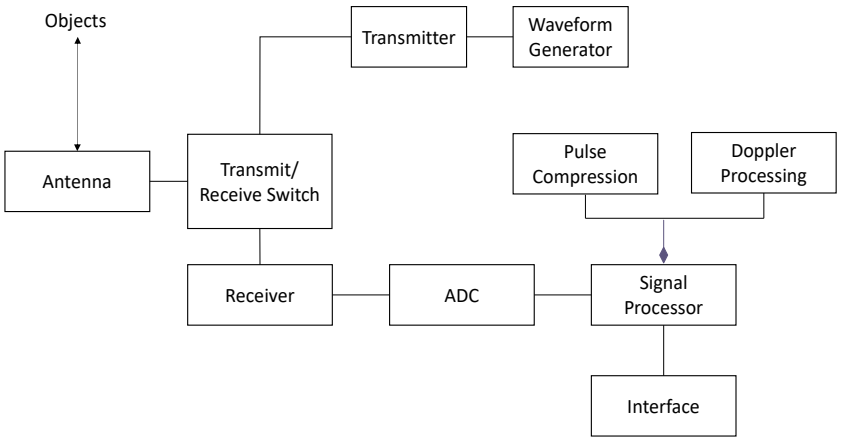

Figure 4. Class diagram of a Radar Sensor.

- *Threat handling*. The solution must be able to stop or mitigate all the identified threats of the secure radar sensor.
- *Diversity of sensor origins*. Applications may use sensors coming from different vendors and possibly with different owners. Some of these owners may be malicious. We also need an effective identification strategy to distinguish each individual sensor; this is particularly important for systems that use several radar sensors, such as autonomous vehicles.
- *Sensor heterogeneity*. There are several varieties of radar sensors such as bistatic, doppler, monopulse, etc. They may have different threats and vulnerabilities, that need to be handled appropriately.
- *Tradeoffs.* There should be tradeoffs between security and cost, functionality, or performance. This is particularly important for radars to be used in safety-critical applications, such as vehicles.

3.4 Threats

We show only some threats, more complete analyses can be found in (Cohen et al. 2019, Yan et al., 2016, and Ren et al. 2020); they may include:

- T1: Jamming: The attacker transmits radio frequency signals to interfere with the radar readings by saturating its receiver with noise. For example, jamming attacks can effectively disable the functionality of a vehicular radar, possibly causing a collision.
- T2: Flooding: Flooding of at least one of the radar network components with any type of network packet in order to create a denial-of-service (DoS) attack (Cohen et al., 2019).
- T3: Spoofing. An attacker can replicate and retransmit the transmitted signals of radars and thus providing false information to corrupt further readings or original data. An attacker can impersonate other devices and gain access to the radar network data or transmit false data to radar components. For example, the distance between an autonomous vehicle and an object can be altered, thus causing an accident.
- T4: Physical damage: This implies damaging radar components using a physical device. This could be an important threat for unattended devices.

3.5 Solution

Add the defenses indicated in 3.5.1 to the class diagram of the radar sensor pattern.

3.5.1 Countermeasures and secure system

- T1: Jamming. (Yan et al., 2016) recommends introducing randomness in the radio signal. (Yen et al. 2017) considers implementing additional authentication, physically separating networks within the same system,

switching frequencies when denied service, and communicating with legitimate devices to blacklist rogue devices. Moreover, to avoid risk/damage caused by errors (misinformation), sensor fusion technology (for example, Kalman filters) may be used, and maybe add an attack detection system.

- T2: Flooding: A general solution is to provide some redundancy in the components.
- T3: Spoofing. There has been no publication of a spoofing attack on a vehicle. This may be due to the relatively high implementation complexity of designing an effective spoofing system (Yeh at al. 2017). Mutual authentication is a possible solution.
- T4: Physical damage. Unattended radars can be enclosed with metal fences, but these can still be breached by a determined and well-prepared attacker.

Figure 5 presents a class diagram of the Secure Radar Sensor pattern. (Lu et al. 2010) suggests incorporation of an anti-jamming Filter in the system, and (Yan et al. 2016) suggests use of a Sensor Fusion Technology and an Attack Detection System. Sensor fusion is a process of combining two or more data sources (sensors) that generates a better understanding of the system behavior with respect to consistency, accuracy, and dependability. A Filter is used in a radar to remove any unwanted frequencies in signals or noise. Before a user performs any activity, both user and system must mutually authenticate. In the attack detection system, use of machine learning techniques are very common. (Riberolles et al. 2020) were able to detect spoofing attacks using a machine learning detection method, using a Long-Short Term Memory mechanism. Another method that can be used to make radar sensors secure is cryptography (Encryptor/Decryptor).

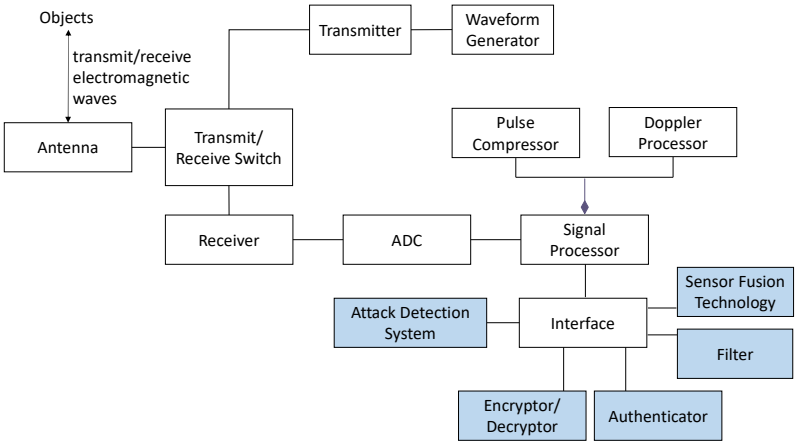

Figure 5. Class diagram of the Secure Radar Sensor pattern.

3.5.2 Dynamics

Although radar sensors are designed to resist environmental noise, intentional noise injected by attacks can hinder the performance of the radar. The injected noise is likely to lower the Signal to Noise Ratio (SNR) and makes sensing impossible (Yan et al., 2016). Fig. 6 presents a sequence diagram of the attack “Jam radar signals”. The attacker injects noise in the antenna, which would distort the measurement but could be corrected as indicated above, using sensor fusion technology or filtering if the noise frequency is outside range.

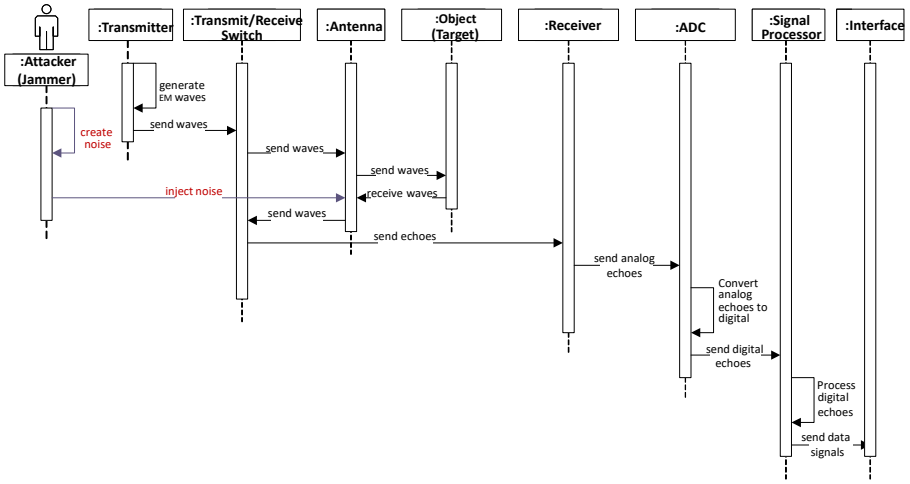

Figure 6. Sequence diagram for the attack “Jam radar signals”.

3.6 Known Uses

- Tesla Model S and Model X cars are equipped with radar sensors. A Medium Range Radar (MRR) is installed under the front bumper on Tesla Model S to support Autopilot functions like front collision avoidance and traffic-aware cruise control. Apparently, a Bosch 76-77 GHz MRR Radar sensor is installed on Tesla cars (Yan et al. 2016). There is an automotive multi-input multi-output (MIMO) radar, which transmits two different signals at the same time to protect them from eavesdroppers. The MIMO radar transmits one signal for the desired information and other signals with false information to confuse the eavesdroppers (Deligiannis et al. 2018).
- Some aircraft use radars with the Automatic Dependent Surveillance-Broadcast (ADS-B) protocol, because it provides accurate aircraft localization and efficient air traffic management (Ying et al. 2019). Also, to recognize whether other aircraft are friendly or not, radar systems use the Identifying Friend or Foe (IFF) protocol, where the radar system in the aircraft communicates with another aircraft and asks it to identify itself (Cohen et al. 2019).
- Waymo uses radar sensors in its autonomous vehicles and indicates that the vehicles contain protections to safeguard against remote attacks and threats from passengers or malicious actors in close proximity (Waymo 2020).

3.7 Consequences

This pattern has the following advantages:

- *Threat handling*. The solution is able to handle all the mentioned threats.
- *Diversity of sensor origins*. We can use identity patterns to keep track of sensors (see Related patterns). In a particular design, we can use radars from only one trusted manufacturer.
- *Sensor heterogeneity*. Again, we can restrict the products we use.
- *Vulnerability of sensors*. Radars are expensive products and can include more advanced defenses.
- *Tradeoffs.* Because of their criticality, cost and performance must be sacrified for higher security.

Possible liabilities include cost (radars are expensive), and complexity (radars are complex and the defenses increase this complexity).

3.8 Related Patterns

- A pattern for a sensor node (Sahu et al. 2010): describes the architecture and dynamics of a Sensor Node that includes a processor, memory, and a radio frequency transceiver.
- A Pattern for a Secure Sensor Node (Orellana et al. 2020): its purpose is to obtain, store and subsequently transmit data securely from a physical environment, to other nodes or Information Systems.
- A Pattern for Sensor Network Architectures (Cardei et al. 2011): describes architectures used by wireless sensor networks, which allow sensor nodes to transmit their collected data to users and users to send commands to the sensor nodes.
- Sensor Node Design Pattern (Saida et al. 2018) a design pattern to model the architecture of a wireless sensor node with real-time constraints, designed and annotated using the UML/MARTE standard.
- Identity patterns are described in (Fernandez 2013).
- Interface patterns are discussed in (M. Brambilla et al. 2017).
- Encryption patterns are shown in (Fernandez 2013).
- The Secure Sensor ASP (this paper) is the prototype for this pattern.

4. CONCLUSIONS

The sensor ASP can be used to develop concrete security patterns for other types of sensors, one of which is shown here, another is the Secure Sensor Node (Orellana et al. 2020). Other derived patterns, candidates for future work, may be lidar sensors, vision cameras, and ultrasonic sensors. A family of these patterns can become a Secure Solution Frame (Uzunov et al. 2015). Another possible future work is development of the concept of ACP for its possible use in design patterns.

ACKNOWLEDGMENTS

We thank our shepherd, Prof. Hironori Washizaki, for his precise comments that have significantly improved this pattern.

REFERENCES

S. Banerjee, D. Sethia, T. Mittal, U. Arora, and A. Chauhan. 2013. Secure sensor node with Raspberry Pi. IMPACT-2013. 26–30

M. Brambilla, et al., “Model-driven development of user interfaces for iot systems via domain-specific components and patterns,” J. of Internet Services and Applications, 8(1), 2017.

F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad and M. Stal, "Pattern-Oriented Software Architecture," in Volume 1: A System of Patterns, Wiley Publishing, 1996.

M. Cardei , E. B. Fernandez , A. Sahu and I. Cardei, "A Pattern for Sensor Network Architectures," in AsianPLoP'11, Tokyo, Japan, 2011.

A. Deligiannis, A. Daniyan, S. Lambotharan, J. A. Chambers, "Secrecy Rate Optimizations for MIMO Communication Radar," , *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2481-2492, Oct. 2018, doi: 10.1109/TAES.2018.2820370.

Shai Cohen, Tomer Gluck, Yuval Elovici, Asaf Shabtai, “Security Analysis of Radar Systems”, CPS-SPC ’19, November *11, 2019, London, UK.*

E.B. Fernandez, “A pattern for a Secure Cloud-Based IoT Architecture”, 2020. Procs. of PLoP'20, October 2020, 8 pages.

E. B. Fernandez, Nobukazu Yoshioka, Hironori Washizaki, and Joseph Yoder, "Abstract security patterns for requirements specification and analysis of secure systems'', Procs. of the WER 2014 conference, a track of the 17th Ibero-American Conf. on Soft. Eng. (CIbSE 2014), Pucon, Chile, April 2014.

E. B. Fernandez, Hironori Washizaki, Nobukazu Yoshioka, “Abstract security patterns,” 15th Conference on Pattern Languages of Programs (PLoP). PLoP ’08, October 18–20, 2008, Nashville, TN, USA

J. Kocić, N. Jovičić and V. Drndarević, "Sensors and Sensor Fusion in Autonomous Vehicles," in *26th Telecommunications forum TELFOR*, Belgrade, Serbia, November 20-21, 2018.

G. Lu, D. Zeng and B. Tang, "Anti-jamming filtering for DRFM repeat jammer based on stretch processing," *2010 2nd International Conference on Signal Processing Systems*, 2010, pp. V1-78-V1-82, doi: 10.1109/ICSPS.2010.5555517.

C. Orellana, E. B. Fernandez and H. Astudillo, "A Pattern for a Secure Sensor Node," in *Procs. 27th Conf. on Pattern Languages of Programs (PLoP)* , 2020.

Kui Ren, Qian Wang, Cong Wang, Zhan Qin, Xiaodong Lin, “The Security of Autonomous Driving: Threats, Defenses, and Future Directions”, Procs. of the IEEE**,** Vol. 108, No2, Feb. 2020. **DOI:** 10.1109/JPROC.2019.2948775

T. de Riberolles, J. Song, Y. Zou, G. Silvestre and N. Larrieu, "Characterizing Radar Network Traffic: a first step towards spoofing attack detection," *2020 IEEE Aerospace Conference*, 2020, pp. 1-8, doi: 10.1109/AERO47225.2020.9172292.

A. Sahu, E. B. Fernandez, M. Cardei and M. Vanhilst, "A Pattern for a Sensor Node," in Writers' workshop at the17th *Conference on Pattern Language pf Programs (PLoP)*, Reno, Nevada, USA, 2010.

R. Saida, Y. H. Kacem, M. S. BenSaleh and M. Abid, "A UML/MARTE Based Design Pattern for a Wireless Sensor Node," in *International Conference on Intelligent Systems Design and Applications*, Vellore, India, 2018.

D. Schreurs, M. Mercuri, P. J. Soh and G. Vandenbosch, "Radar-Based Health Monitoring," *2013 IEEE MTT-S International Microwave Workshop Series on RF and Wireless Technologies for Biomedical and Healthcare Applications (IMWS-BIO)*, 2013, pp. 1-3, doi: 10.1109/IMWS-BIO.2013.6756189.

C. S. Shih, J.-J. Chou, N. Reijers and T.-W. Kuo, "Designing CPS/IoT applications for smart //?? ," *IET Cyber-Physical Systems: Theory & Applications,* vol. 1, no. 1, pp. 3-12, 2016.

Y. Tan, S. Goddard and L. C. Perez, "A prototype architecture for cyber-physical systems," *ACM SIGBED,* vol. 5, no. 1, 2008.

Anton Uzunov, E. B Fernandez, Katrina Falkner, "Security solution frames and security patterns for authorization in distributed, collaborative systems", Computers & Security, 55, 2015, pp. 193-234, doi: 10.1016/j.cose.2015.08.003.

S. Umamaheshwari and J. N. Swaminathan. Man-In-Middle Attack/for a Free Scale Topology. 2018 International Conference on Computer Communication and Informatics (ICCCI). 1–4.

Waymo, "Waymo's Safety Methodologies and Safety Readiness Determinations". October 2020. https://storage.googleapis.com/sdc-prod/v1/safety-report/Waymo-Safety-Methodologies-and-Readiness-Determinations.pdf

Chen Yan, Wenyuan Xu, Jianhao Liu, "Can You Trust Autonomous Vehicles: Contactless Attacks against Sensors of Self-driving Vehicle", DEF_CON 24 Hacking Conference, 2016.

Enoch R. Yeh, Junil Choi, Nuria G. Prelcic, Chandra R. Bhat, and Robert W. Heath, Jr., "Security in Automotive Radar and Vehicular Networks ", 2017.

X. Ying, J. Mazer, G, Bernieri, M. Conti, L. Bushnell, and R. Poovendran, "Detecting ADS-B Spoofing Attacks using Deep Neural Networks," *2019 IEEE Conference on Communications and Network Security (CNS)*, 2019, pp. 187-195, doi: 10.1109/CNS.2019.8802732.