A Pattern for a Secure Sensor Node

CRISTIAN ORELLANA, Universidad Técnica Federico Santa María
EDUARDO B. FERNANDEZ, Florida Atlantic University
HERNÁN ASTUDILLO, Universidad Técnica Federico Santa María

Today, with the proliferation of the Internet of Things (IoT), the use of sensors is widely extended to domains as diverse as health, support and automation of industrial processes under the paradigm of Industry 4.0, traffic control, smart cities, monitoring of transport fleets, among many other applications. A problem that emerges from these scenarios is that both the data that circulates through sensor ecosystems, as well as that which is temporarily stored by these sensors, may eventually be subject to intervention by malicious users, generating: (1) a negative impact on the privacy, integrity and confidentiality of user and system data; and (2) compromise the security of the sensor nodes to execute malicious actions on a physical or digital environment, impacting on the safety of people, and economically on companies and communities. In this article, we introduce a pattern for the architectural design of a secure sensor node that characterizes the problem and helps architects solve it using a reusable, adaptable, and extensible solution. This pattern describes a Secure Sensor Node, a Cyber-Physical System (CPS) whose purpose is to obtain, store and subsequently transmit data securely from a physical environment, to other nodes or Information Systems.

Categories and Subject Descriptors: D.2.11 [**Software Engineering**]: User Interfaces—*Software Architectures—Patterns*; B.4.1 [**Input/Output and Data Communications**]: User/Machine Systems—*Data Communication Devices*; D.1.5 [**Programming Techniques**]: Models—*Object-oriented Programming*

General Terms: Design

Additional Key Words and Phrases: Architecture, Pattern, Sensors, Wireless Communication

ACM Reference Format:

Orellana, C. Fernandez, E.B. and Astudillo, H. 2020. A Pattern for Secure Sensor Node Proc. of the 27th Conf. on Pattern Langs. of Progs. (PLoP'20), 10 pages.

1. INTRODUCTION

Currently, the ubiquity of the internet has allowed a wide spectrum of devices to be connected and interact with other devices of a different nature. For example, there are devices that collect information from the environment and transmit it over the Internet so that other systems can analyze that data, or interact with the environment using this data as parameters for a specific process.

These ecosystems gradually allowed the interaction and convergence of two orthogonal worlds: the physical environment and the digital one. These new ecosystems have given rise to concepts such as Cyber-Physical System (CPS) and Internet of Things (IoT). A Cyber-Physical System (CPS) is a system that combines physical and computer or cyber components. The physical component consists of systems that exist in nature, such as biological entities as well as those developed by humans, such as transportation and energy-producing

Author's address: C. Orellana, UTFSM, Av. España 1680, Valparaíso, Chile; email: cristian.orellanae@usm.cl; Eduardo B Fernandez, Dept. of CEECS, FAU, 777 Glades Road, Boca Raton, FL 33431, USA; email: fernande@fau.edu; Hernán Astudillo, UTFSM, Av. España 1680, Valparaíso, Chile; email: hernan@inf.utfsm.cl

systems. This component exists, operates, and interacts with its environment in continuous or ordinary time. The computational component consists of systems and entities involved in processing, communicating, and controlling information via computational means [Rawat et al. 2015]. The Internet of Things (IoT) is a concept in which the virtual world of information technology integrates seamlessly with the real world of things. “Things” are any identifiable physical object independent of the technology that is used for identification or providing status information of the objects and its surroundings [Uckelmann et al. 2011].

A sensor is a device that measures a physical quantity, e.g. light, temperature, or pressure, and converts it into a signal which can be read by a human or by an instrument. Sensors are typically small, self-contained, battery-powered, low cost devices. A Wireless Sensor Network (WSN) is a network composed of a large number of tiny sensor nodes that have the ability to sensing, communicate and perform processing using parameters collected from a physical environment [Fahmy 2016]. Sensors are generally low power and distributed in an ad hoc, decentralized fashion [Vacca 2009]. Sensors are increasingly ubiquitous elements that can be found in industrial ecosystems, healthcare, smart cities, and everyday appliances. They are also used to diagnose health problems through instruments and wearables, to assist vehicle driving and to allow early identification of domestic security incidents and environmental risk factors.

A sensor node is a node in a wireless sensor network able to perform some processing, gathering sensory information, and communicating with other connected nodes in the network [Sahu et al. 2010]. Since the sensor nodes are typically designed to interoperate with other devices through protocols and standards where security is not the main architectural driver, and considering also the little processing capacity that they usually have due to their hardware characteristics, it becomes necessary have a design reference to build a secure sensor node to mitigate potential security threats [Modares et al. 2011].

This work introduces the Secure Sensor Node pattern, which is part of a set of patterns that characterize a secure IoT architecture. Fig. 1 illustrates the pattern diagram that supports a secure IoT ecosystem. Our intended audience are system designers and security researchers.

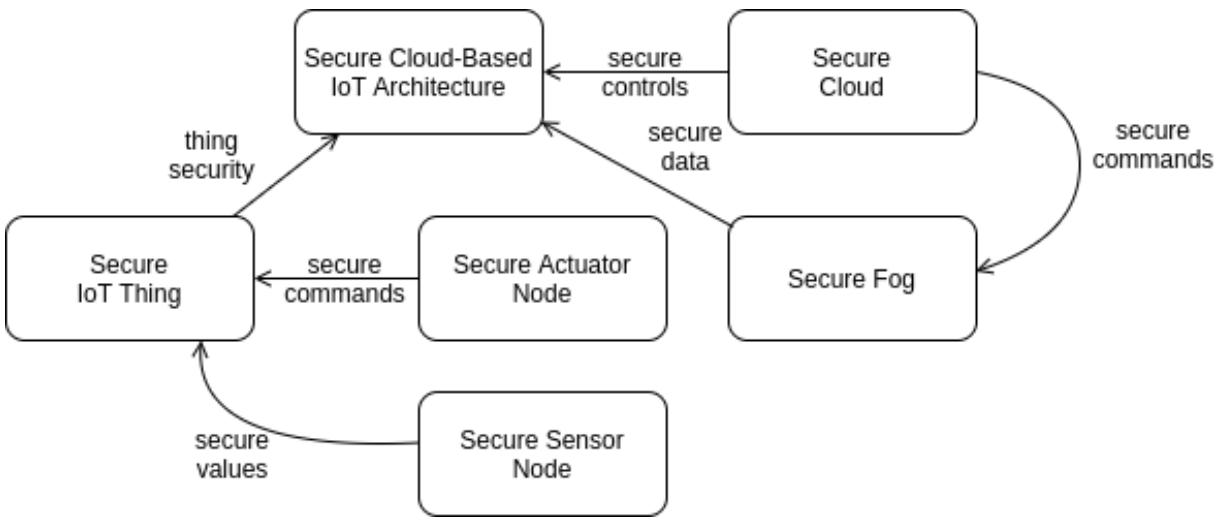

Fig. 1. Pattern diagram of IoT architecture ecosystem

2. SECURE SENSOR NODE

INTENT

A Secure Sensor Node is part of a Cyber-Physical System (CPS) [Rawat et al. 2015] whose purpose is to obtain, store and subsequently transmit data securely from a physical environment, to other nodes or Information Systems.

EXAMPLE

In the oil industry ecosystem, pipelines that transport several types of oil and gas use SCADA systems [Thames and Schaefer 2017] composed of electronic sensors to keep track of the continuous flow of hydrocarbons. These sensors measure and transmit data to allow real-time reporting and monitoring of historical accuracy and consistency. The data from these sensors are sent to an industrial Flow Computer, which can store only a few reports and also can send a report to a local or remote printer. These ecosystems were designed a few decades ago to work in trusted environments, so in principle, they do not have any security mechanism to guarantee that the transmitted data is trustworthy with respect to the measurement, which makes it possible for an attacker to intercept such communication for malicious purposes [Orellana et al. 2019]. For example, an attacker could intercept the communication channel between the node and the flow computer, altering the data to simulate a lower level of oil volume, which allows stealing a part of the product that comes through the pipeline.

CONTEXT

Sensor nodes are used to collect values from a physical environment and transmit them to other Sensor Node or some external system that is responsible for the storage, availability and processing of this data. Typically, this data can constitute sensitive information or information that, if adulterated, implies a significant negative impact on the company. Sensor nodes are part of a WSN [Tubaishat et al. 2004], which can support: (1) Ad-hoc topologies (also known as P2P or mesh), (2) Star topologies (also known as infrastructure mode networks), and (3) Tree topologies, in which there is a hierarchical structure of nodes to establish communication [Gutiérrez et al. 2013]. Some of the protocols used by sensor nodes include MQTT, HTTP, CoAP, DDS, AMQP, DNP3, and Modbus [Thames and Schaefer 2017]. Among the most widely used standards for IoT are Bluetooth Low Energy (BLE) [Fahmy 2016], IEEE 802.11ah and IEEE 802.15.4 [Ahmed et al. 2016].

PROBLEM

A sensor node seamlessly performs data collection and transmission tasks in a sensor network. However, if the Sensor Node does not have mechanisms to guarantee security in its operation, it could be vulnerable to multiple security threats that can compromise the data stored, transmitted and eventually the environment with which it interacts. The following list corresponds to potential threats that can be executed by an attacker:

—**T1. Tampering:** An attacker intercepts sensor outputs to gain access to sensitive data. The attacker can leak this data or use it to compromise other systems using lateral movement techniques [Sood and Enbody 2014].

—**T2. Flooding:** From the WSN, an attacker repeatedly generates requests in an uncontrolled manner to generate log rotation and thus eliminate part of the log that is useful for forensic analysis [Alani 2018]. If the logs are incomplete, it is impossible to reconstruct an incident that occurred in the system.

—**T3. Spoofing:** An attacker performs a spoofing attack by compromising the access credentials of a sensor node and its respective network configuration.

—**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 [Rayes and Salam 2018] which interact with a physical environment to perform unwanted actions.

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

—**Integrity.** The sensor node must guarantee that both transmitted and stored data cannot be altered by a third party who is not authorized to manipulate this data [Wang et al. 2011].

—**Confidentiality.** Only authorized users should have read access to data sent or stored by the device. The communication channels must guarantee that the transmitted data cannot be read during the sending.

—**Availability.** The sensor node should send data continuosly.

—**Auditability.** The system should allow the reconstruction and supervision of both historical and ongoing events to those who are granted access for these purposes.

—**Overhead.** Sensor data processing should not result in a large overhead.

—**Power consumption.** RFID devices [Chen and Chen 2016] are usually power constrained. Our solution must be economic on the use of power.

3. SOLUTION

The Secure Sensor Node pattern establishes the guidelines for a sensor node to transmit and store data securely in a WSN, regardless of its topology. A secure sensor node is capable of encrypting data in communications and storage while keeping a record of events for auditing purposes. This pattern can be described as an extension of the Sensor Node pattern [Sahu et al. 2010] with security capabilities. To provide security features to a sensor node, it is required to implement additional components to encrypt data, log events, and perform authentication and authorization mechanisms with other nodes of the WSN. This pattern's structure and dynamics will be illustrated using UML, a widely adopted approach to representing security patterns [Washizaki et al. 2021].

3.1 STRUCTURE

Fig. 2 presents a structural view of the Sensor Node without security.

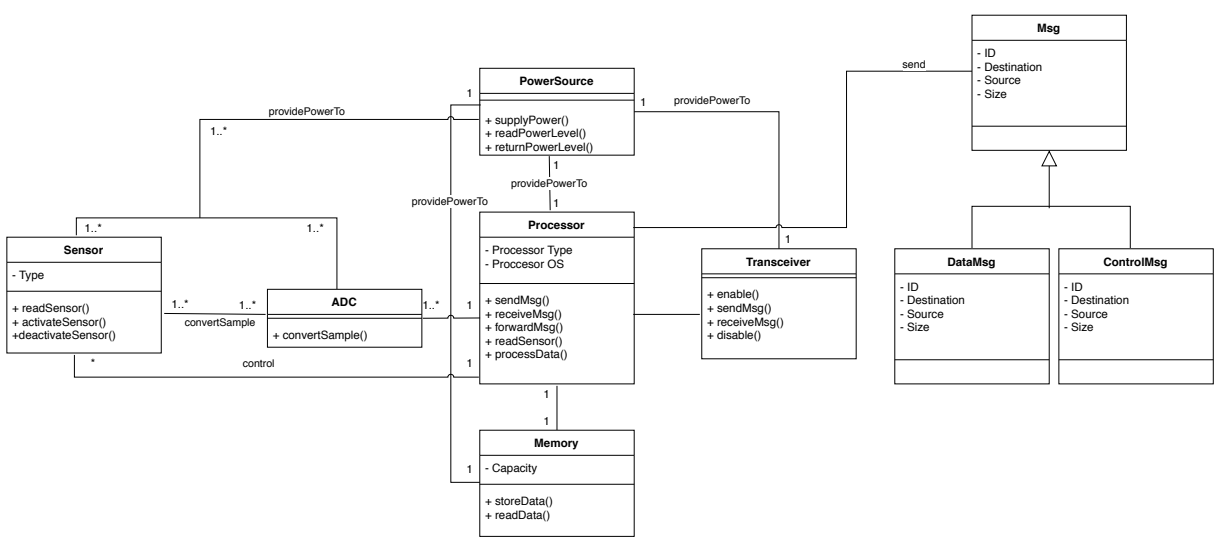

Fig. 2. Class diagram of the Sensor Node

Fig. 3 illustrates the entities involved that allow extending a Sensor Node to a Secure Sensor Node. The structure proposed by the Secure Sensor Node pattern can be used partially or totally. It can even be extended to cover specific scenarios.

—Encryptor is responsible for encrypting and decrypting the data sent and stored by the Secure Sensor Node. It also securely manages access tokens and credentials that are used in authentication processes on a WSN. Security credentials can be managed with a low-cost Hardware Security Module (HSM) [Parrinha and Chaves 2017]. Encryptor stores the encryption settings for WSN nodes and services.

—Authenticator is responsible for retrieving, configuring, and transmitting access data and credentials securely using a pre-defined authentication flow. The Authenticator is used to authenticating over a network [El-hajj et al. 2017], to access other Secure Sensor Nodes in the WSN, and eventually to access API endpoints that are outside the network, either on-premise or in the Cloud.

—Authorizer is used to providing access to the Secure Sensor Node by other nodes or clients. This applies in scenarios where the Secure Sensor Node provides services for other nodes on the network, and also to update and configure the device. The authorizer is responsible for establishing an access control model that ensures that only previously authorized devices or users can access specific services or resources of the Secure Sensor Node.

—Security Logger/Auditor has the purpose of keeping a record of all the accesses and events that occur in a Secure Sensor Node for auditing purposes. The Security Logger/Auditor can be configured to support multiple levels of logging.

—Secure Processor must provide two properties for any off-chip data, confidentiality, and integrity [Lee et al. 2016]. The Secure Processor invokes Encryptor to execute cryptographic methods to encrypt and decrypt data; and also to establish a secure channel in the communications managed by the Transceiver.

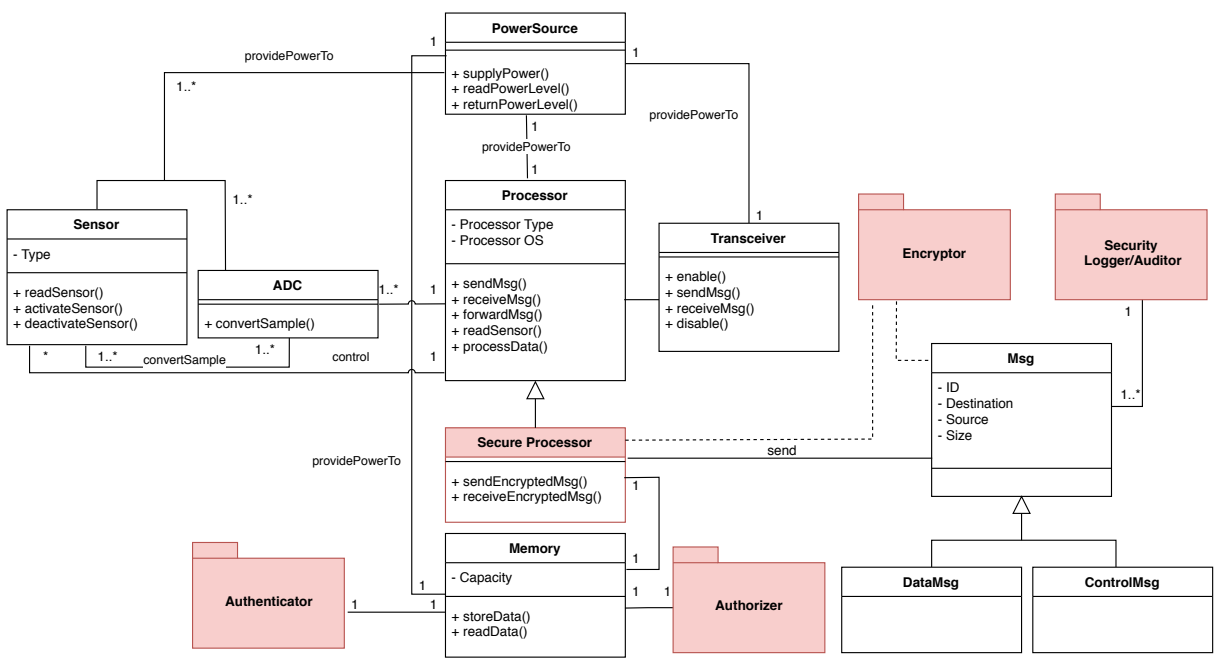

Fig. 3. Class diagram of the Secure Sensor Node

3.2 DYNAMICS

Fig. 4 shows the use case "Send encrypted message from Node1 to Node2".

(1) The Secure Processor sends a message to the Memory requesting access to data.
(2) The Memory performs a data read operation and sends the data to the Secure Processor.
(3) The Secure Processor executes a processing operation on the received data to build a message.
(4) The Secure Processor encrypts the message using an encryption operation available in Encryptor.
(5) The Encryptor returns the encrypted message to the Secure Processor
(6) The Secure Processor sends the encrypted message to the Transceiver.
(7) The Node1 Transceiver sends the encrypted message to the Node2 Transceiver.

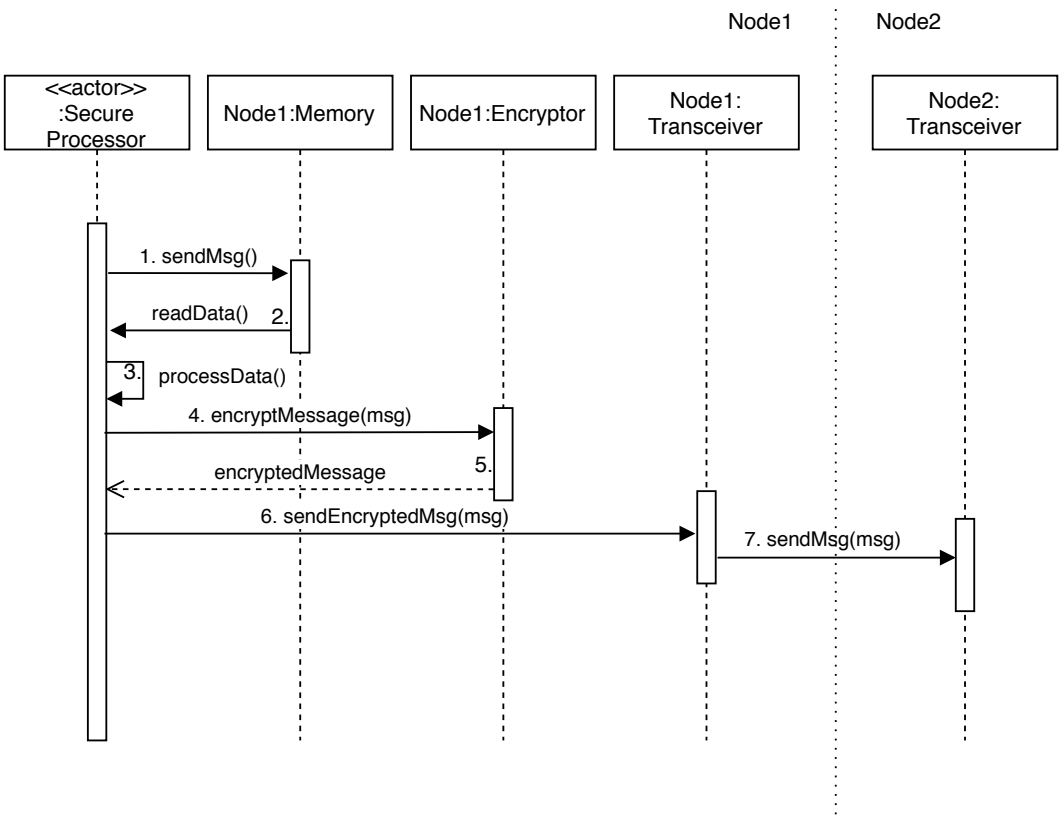

Fig. 4. Sequence diagram of use case "Send encrypted message from Node1 to Node2"

4. IMPLEMENTATION

A Secure Sensor Node can be designed from hardware and software solutions that implement data encryption while providing authentication and access control. Since these devices are small in terms of memory and processing capacity, there can generally be a trade-off in the encryption algorithm regarding eventual performance issues

[Maitra et al. 2019]. Encryption can be used at different levels: network-level encryption or application-level encryption. If encryption is implemented in communications, network nodes must be able to securely associate with a router or other node under a specific topology. If encryption is at the application level, the nodes must ensure the correct implementation of a protocol with encryption capabilities.

Implementing encryption on a Secure Sensor Node involves some technical challenges: (1) secure storage of credentials (2) the decrease in overhead of the network payload due to the use of encryption.

The implementation should also consider the incorporation of audit components in the device that allow the monitoring and review of events and actions in the secure sensor node.

Finally, a secure sensor node must incorporate authorization and access control mechanisms to allow that the services offered by the device can only be consumed by clients or machines that are previously authorized. For example, in WSNs that implement microservices-based architectures [Lu et al. 2017], authorization and access control mechanisms must exist in each protected resource on the network. In a practical context, it is possible to find a wide variety of solutions for the implementation of access control mechanisms in IoT ecosystems, which explore the applicability of access control models such as Role-based Access Control (RBAC) and Cryptography-Based Access Control (CBAC) [Maw et al. 2014]. Currently, there are standards such as OAuth 2.0 and OpenID Connect that can be used over HTTP or MQTT protocols for authorization and authentication tasks, respectively [Siriwardena 2014]. To implement these elements, there are also services deployed in the Cloud, such as AWS IoT[1] or Azure IoT[2] that provide authentication, authorization, and audit trails securely through APIs.

5. KNOWN USES

(1) Zolertia Zoul [Kurniawan 2018] is based on TI's CC2538 system on chip (SoC), featuring an ARM Cortex-M3 with 512KB flash, 32Kb RAM, double RF interface. The Zoul is a core module developed by Zolertia[3] to target most Internet of Things (IoT) applications and products, providing a flexible and affordable module solution to integrate to most existing products and solutions.
Specifications:
—Cryptoprocessor (AES-ECB/CBC/CTR/CBC-MAC/GCM/CCM-128/192/256, SHA-256)
—ISM 2.4-GHz IEEE 802.15.4 & Zigbee compliant. ISM 868-, 915-, 920-, 950-MHz ISM/SRD Band.
—AES-128/256, SHA2 Hardware Encryption Engine.
—ECC-128/256, RSA Hardware Acceleration Engine for Secure Key Exchange.

(2) Raspberry Pi based Secure Sensor Node [Banerjee et al. 2013] This secure sensor node prototype can be used to develop a body sensor module which involves processing, encryption and sending of large amount of data gathered from multiple sensors.
Specifications:
—Raspbian Wheezy
—Low power single-core 700 MHz, 512 MB RAM
—RC4 encryption
—ADXL345 3-axis accelerometer (13 bits).
—DELL Bluetooth Dongle.

Both devices provide the elements described by the Secure Sensor Node pattern in their implementation. These elements can be implemented through software, using operating system libraries for RIOT OS[4] and Contiki-NG[5] or through hardware modules.

[1] https://aws.amazon.com/iot/
[2] https://azure.microsoft.com/en-us/overview/iot/
[3] https://zolertia.io
[4] https://www.riot-os.org/
[5] https://github.com/contiki-ng/contiki-ng

6. CONSEQUENCES

The Secure Sensor Node pattern has the following advantages:

—Secure Sensor Node allows secure communication with other WSN nodes.

—Secure Sensor Node supports auditability by allowing a record of events or incidents that occurred on the device.

—Secure Sensor Node implements access control mechanisms to allow access to private APIs and other services.

The Secure Sensor Node pattern has the following disadvantages:

—Due to data encryption, there may be a negative impact on CPU performance, and battery life.

—Complexity in initial setup.

The Secure Sensor Node pattern has the following limitations:

—There is a threat that is not covered by the pattern since this requires physical attacks on the secure sensor node, for which the pattern presents no solution: *An attacker with physical access to the sensor node disconnects it from the power supply, causing an alteration in the logical structure of the partitions or the operating system resulting from an incomplete sequence of operations.* In this scenario the only defense is physical protection.

—This pattern does not protect against DoS attacks [Daud et al. 2018] and other network attacks because this requires solutions and techniques that are beyond the scope of it. These countermeasures could impact the entire WSN.

7. RELATED PATTERNS

—Secure Cloud-Based IoT Architecture [Fernandez 2020]. This pattern defines security protection for data assets and for the communication channels in a hierarchy of layers in order to neutralize the system threats and reduce the complexity of managing security.

—Sensor Node [Sahu et al. 2010]. This pattern corresponds to the insecure version of the pattern proposed in this article. A pattern for sensor network architectures that describes the structure and dynamics of a WSN.

—Sensor Network Architecture Pattern [Cardei et al. 2011]. This pattern describes an abstract view of the structure and general architecture of a wireless sensor network. Using a sensor network pattern makes the design of such a network simpler and more convenient, and can facilitate their integration with the rest of the IT system when applicable.

—Sensor Node Design Pattern [Saida et al. 2018]. A generic design pattern intended to support the modeling of the architecture of a wireless sensor node with real time constraint.

—Authenticator [Fernandez-Buglioni 2013]. After identification the authenticator grants access to the system if the requester is a registered subject.

—Authorizer [Fernandez-Buglioni 2013]. Control of who can access a network entity and in what way. An important variety is Role-Based Access Control.

—Security Logger and Auditor [Fernandez-Buglioni 2013]. How can we keep track of user's actions in order to determine who did what and when? Log all security-sensitive actions performed by users and provide controlled access to records for Audit purposes.

—Encryptor. It is part of the Secure Channel pattern [Braga et al. 1998].

8. CONCLUSIONS

The extensive attack surface to which IoT systems are exposed makes it necessary for a Sensor Node to have security mechanisms that provide protection against threats to nodes, sensor networks and the technological infrastructure where they are deployed. The Secure Sensor Node pattern is designed to cover security aspects that

are not addressed by the Sensor Node pattern. In this way, it implements design decisions to cover mechanisms for authentication, authorization, node monitoring, data encryption and event logs.

9. ACKNOWLEDGMENTS

We thank our shepherd Christian Kohls for his permanent support, and for his exhaustive review of our work, which allowed us to significantly improve the article. This work has been partially supported by ANID PCHA/Doctorado Nacional under grant 2017-21171506, ANID PIA/APOYO AFB180002 (CCTVal) and by UTFSM/DPP.

REFERENCES

N. Ahmed, H. Rahman, and Md.I. Hussain. 2016. A comparison of 802.11ah and 802.15.4 for IoT. *ICT Express* 2, 3 (2016), 100 – 102. DOI:http://dx.doi.org/https://doi.org/10.1016/j.icte.2016.07.003 Special Issue on ICT Convergence in the Internet of Things (IoT).

Mohammed M. Alani. 2018. IoT Lotto: Utilizing IoT Devices in Brute-Force Attacks. In *Proceedings of the 6th International Conference on Information Technology: IoT and Smart City (ICIT 2018)*. Association for Computing Machinery, New York, NY, USA, 140–144. DOI:http://dx.doi.org/10.1145/3301551.3301606

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

A Braga, C Rubira, and Ricardo Dahab. 1998. Tropyc: A pattern language for cryptographic object-oriented software. *Pattern Languages of Program Design* 4 (1998).

Mihaela Cardei, Eduardo Fernández, Anupama Sahu, and Ionut Cardei. 2011. A pattern for sensor network architectures, Vol. 2011. 1–8. DOI:http://dx.doi.org/10.1145/2524629.2524641

Min Chen and Shigang Chen. 2016. *RFID Technologies for Internet of Things* (1st ed.). Springer Publishing Company, Incorporated.

M. Daud, R. Rasiah, M. George, D. Asirvatham, A. F. A. Rahman, and A. A. Halim. 2018. Denial of service: (DoS) Impact on sensors. In *2018 4th International Conference on Information Management (ICIM)*. 270–274.

M. El-hajj, M. Chamoun, A. Fadlallah, and A. Serhrouchni. 2017. Analysis of authentication techniques in Internet of Things (IoT). In *2017 1st Cyber Security in Networking Conference (CSNet)*. 1–3.

Hossam Mahmoud Ahmad Fahmy. 2016. *Wireless Sensor Networks: Concepts, Applications, Experimentation and Analysis* (1st ed.). Springer Publishing Company, Incorporated.

Eduardo B. Fernandez. 2020. A pattern for a Secure Cloud-Based IoT Architecture. In *Proceedings of the 27th Conference on Pattern Languages of Programs (PLOP '20)*. Association for Computing Machinery, USA.

Eduardo Fernandez-Buglioni. 2013. *Security Patterns in Practice: Designing Secure Architectures Using Software Patterns* (1st ed.). Wiley Publishing.

Daniel Gutiérrez, S.L. Toral, Federico Barrero, Nik Bessis, and Eleana Asimakopoulou. 2013. *The Role of Ad Hoc Networks in the Internet of Things: A Case Scenario for Smart Environments*. Vol. 460. 89–113. DOI:http://dx.doi.org/10.1007/978-3-642-34952-2_4

Agus Kurniawan. 2018. *Introduction to Wireless Sensor Networks*. Apress, Berkeley, CA, 1–46. DOI:http://dx.doi.org/10.1007/978-1-4842-3408-2_1

J. Lee, T. Kim, and J. Huh. 2016. Reducing the Memory Bandwidth Overheads of Hardware Security Support for Multi-Core Processors. *IEEE Trans. Comput.* 65, 11 (2016), 3384–3397.

D. Lu, D. Huang, A. Walenstein, and D. Medhi. 2017. A Secure Microservice Framework for IoT. In *2017 IEEE Symposium on Service-Oriented System Engineering (SOSE)*. 9–18.

S. Maitra, D. Richards, A. Abdelgawad, and K. Yelamarthi. 2019. Performance Evaluation of IoT Encryption Algorithms: Memory, Timing, and Energy. In *2019 IEEE Sensors Applications Symposium (SAS)*. 1–6.

Htoo Maw, Hannan Xiao, Bruce Christianson, and James Malcolm. 2014. A Survey of Access Control Models in Wireless Sensor Networks. *Journal of Sensor and Actuator Networks* 3, 2 (Jun 2014), 150–180. DOI:http://dx.doi.org/10.3390/jsan3020150

H. Modares, R. Salleh, and A. Moravejosharieh. 2011. Overview of Security Issues in Wireless Sensor Networks. In *2011 Third International Conference on Computational Intelligence, Modelling Simulation*. 308–311.

Cristian Orellana, Mónica M. Villegas, and Hernán Astudillo. 2019. Mitigating Security Threats through the Use of Security Tactics to Design Secure Cyber-Physical Systems (CPS). In *Proceedings of the 13th European Conference on Software Architecture - Volume 2 (ECSA '19)*. Association for Computing Machinery, New York, NY, USA, 109–115. DOI:http://dx.doi.org/10.1145/3344948.3344994

D. Parrinha and R. Chaves. 2017. Flexible and low-cost HSM based on non-volatile FPGAs. In *2017 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*. 1–8.

Danda B. Rawat, Joel J. P. C. Rodrigues, and Ivan Stojmenovic. 2015. *Cyber-Physical Systems: From Theory to Practice*. CRC Press, Inc., USA.

Ammar Rayes and Samer Salam. 2018. *Internet of Things From Hype to Reality: The Road to Digitization* (2nd ed.). Springer Publishing Company, Incorporated.

Anupama Sahu, Eduardo B. Fernandez, Mihaela Cardei, and Michael Vanhilst. 2010. A Pattern for a Sensor Node. In *Proceedings of the 17th Conference on Pattern Languages of Programs (PLOP '10)*. Association for Computing Machinery, New York, NY, USA, Article 7, 7 pages. DOI:http://dx.doi.org/10.1145/2493288.2493295

Raoudha Saida, Yessine Hadj Kacem, Mohammed S. BenSaleh, and Mohamed Abid. 2018. A UML/MARTE Based Design Pattern for a Wireless Sensor Node. In *ISDA*.

Prabath Siriwardena. 2014. *Advanced API Security: Securing APIs with OAuth 2.0, OpenID Connect, JWS, and JWE* (1st ed.). Apress, USA.

Aditya Sood and Richard Enbody. 2014. *Targeted Cyber Attacks: Multi-Staged Attacks Driven by Exploits and Malware* (1st ed.). Syngress Publishing.

Lane Thames and Dirk Schaefer. 2017. *Cybersecurity for Industry 4.0: Analysis for Design and Manufacturing* (1st ed.). Springer Publishing Company, Incorporated.

Malik Tubaishat, Jian Yin, Biswajit Panja, and Sanjay Madria. 2004. A Secure Hierarchical Model for Sensor Network. *SIGMOD Rec.* 33, 1 (March 2004), 7–13. DOI:http://dx.doi.org/10.1145/974121.974123

Dieter Uckelmann, Mark Harrison, and Florian Michahelles. 2011. *Architecting the Internet of Things* (1st ed.). Springer Publishing Company, Incorporated.

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

John R. Vacca. 2009. *Computer and Information Security Handbook*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Qian Wang, Kui Ren, Shucheng Yu, and Wenjing Lou. 2011. Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance. *ACM Trans. Sen. Netw.* 8, 1, Article 9 (Aug. 2011), 24 pages. DOI:http://dx.doi.org/10.1145/1993042.1993051

Hironori Washizaki, Tian Xia, Natsumi Kamata, Yoshiaki Fukazawa, Hideyuki Kanuka, Takehisa Kato, Masayuki Yoshino, Takao Okubo, Shinpei Ogata, Haruhiko Kaiya, Atsuo Hazeyama, Takafumi Tanaka, Nobukazu Yoshioka, and G. Priyalakshmi. 2021. Systematic Literature Review of Security Pattern Research. *Information* 12 (01 2021), 36. DOI:http://dx.doi.org/10.3390/info12010036