Multi-Agent System and Digital Twin Models for Security Study of Cyber-Physical System

Pre-study

October 2022

Student	Zoé Lagache[1]
KTH Examiner	Roberto Guanciale
KTH Supervisor	Musard Balliu
LCIS supervisor	Oum-El-Kheir Aktouf[2]
	Annabelle Mercier[3]

[1] lagache@kth.com

[2] oum-el-kheir.aktouf@lcis.grenoble-inp.fr

[3] annabelle.mercier@lcis.grenoble-inp.fr

Keywords: Multi-Agent System, Digital Twin, Cyber-Physical system, Security

Glossary

BDI Belief–Desire–Intention. 9

CCLCBM Center for Connected Learning and Computer Based Modeling. 9

CIA Confidentiality, Integrity, Availability. 4, 6, 7

CPS Cyber-Physical System. 1–5, 7

DT Digital Twin. 2, 10, 11

IoT Internet of Things. 2

MAS Multi-Agent System. 2, 7–9

MITM Man-In-The-Middle. 5

TMS Trust Management System. 8

Acronyms

DDoS Distributed Denial of Service. 7

DoS Denial of Service. 7

FIPA Foundation for Intelligent, Physical Agent. 9, 10

JADE Java Agent DEvelopment. 9

MaDKit Multiagent Development Kit. 9

PADE Python Agent DEvelopment. 9

SPADE Smart Python Agent Development Environment. 10

SPOF Single Point Of Failure. 8

Contents

Chapter 1

Introduction

1.1 Background

In a world in an increasing need of control over physical processes, monitoring became mandatory. In order to be accurate enough, more an more systems use electronic systems for monitoring, or even in order to create a digital interface to manipulate physical processes more easily. As an example, cars are since several years now equipped with many sensors to ensure their proper functioning and the safety of the passengers [1]. This is what we call Cyber-Physical System (CPS). These systems are becoming increasingly common and play a crucial role in many aspects of modern life.

1.2 Problem

According to the CPS Steering Group [2], CPS are defined as the interaction between physical systems and processes using computations and communication abilities. However, the integration of computing and communication technologies also introduces new security risks [3] [4] which could have a huge impact. We can imagine the Cyber-Physical System in an airport monitoring each planes entering or leaving the airport. If such a system undergoes, for instance, a DoS attack, the consequences could be devastating. This is why there is a growing need to study the security of CPS and develop effective strategies for protecting them.

1.3 Purpose

Cyber-Physical Systems can be found in a large range of fields, going from healthcare to the electrical power grid management. Numerous applications of it exist and could benefit from a stronger degree of security.

The project, motivated by this need of security, aim to contribute to the field by using innovative approaches such as Multi-Agent System (MAS) and Digital Twin (DT) models. This could help advance the state of the art in Cyber-Physical System security and provide new insights and solutions for addressing the challenges in this area.

By combining the MAS (see 2.2) and the DT (see 2.3) approaches, the project could help at identifying potential security vulnerabilities in cyber-physical systems and develop strategies for addressing them. This could help improve the security of these systems and protect them from cyber threats.

1.4 Goals

In this master thesis, we propose to study how Multi-Agent System (MAS) and DT models can help with this aforementioned problem, and how they complete each other. A Multi-Agent System is a system composed of agents collaborating with each other in order to achieve a common goal. The agents are able to communicate with their local neighbours and, most of the time, only have a local view of the system. MAS are now a trend in the Internet of Things (IoT) field thanks to its decentralization aspect. In the other hand Digital Twins are often associated to a way to track and analyse a system in real time, often in order to predict its behaviour. Both models have the potential to be very useful tools for protecting and preventing cyber attacks on CPS.

Our goals with this project are:

- Developing multi-agent system and digital twin model for studying the security of CPS
- Identifying potential security vulnerabilities in CPS
- Developing strategies for addressing these vulnerabilities and improving the security of CPS using this model

- Evaluating the effectiveness of these strategies through simulation and experimentation

1.5 Research methodology

In order to establish the state of the art, we are going to use mainly Google Scholar and Scopus to look for papers or articles and Zotero to save them.

The results from the state of the art will lead us to our model creation. Then we will simulate this model with the chosen simulator to check if it works as we want and if we can extract some security claims in relation with the chosen attack model.

From a reproducibility perspective, we will try to provide every sources of paper and tool we use when possible, and will state when it is not possible, which is unlikely in our case. Indeed, most of the tool we will use should be free. However, some papers may require institutional access or a fee to access the full document.

1.6 Delimitations

Definitions of both Multi-Agent systems and Digital Twins do not have a consensus. We will have to determine one. This could limit the set of possible interpretations of these two concepts. In addition to this, this work will only focus on one specific vulnerability model and will not be generalized to more than the chosen model. As a limitation, we can also take the duration of the project into consideration. Indeed this is a 5 months project which does not leave much room to study all the possible aspect of the subject.

1.6.1 Ethics and sustainability

In computer science, simulation is often something that is energy consuming. We will try to tend to optimized solutions but this is not the in the scope of this project. We will not focus on a sustainable solution for security of CPS analysis.

Moreover, As an ethical and sustainability perspective, we have to say that this work will discuss attacks on CPS and may present examples that could be used outside of this work context. However, since the project is primarily concerned with theoretical concepts, this is unlikely.

Chapter 2

Background

In this chapter, we are going to introduce the main areas of the subject by providing the basic background on them. The related work will also be discussed.

2.1 Cyber-physical Systems

In this section, we are going to explain how CPS are defined for this thesis and focus on what kind of vulnerability this kind of system is facing an which parameter of the Confidentiality, Integrity, Availability (CIA) model is affected for each vulnerability.

2.1.1 What are Cyber-physical Systems

As indicated in 1.2, CPS are the combination of the physical world and the cyber space interacting with one another. Most of the time, the cooperation consists in the cyber system monitoring the physical one as seen in Chen et al. [5], Lei et al. [6] and Tsang et al. [7] papers.

2.1.2 Cyber-physical System vulnerabilities

In figure 2.1, we represented a CPS by two parts: the physical process(es) part and the cyber system. Both of these parts are interacting with each other through sensors and actuators which compose the interface between the two worlds. The cyber system is composed of computing devices receiving data from the sensors, processing it, and sending the result to the actuators. The green arrows are indicating the monitoring

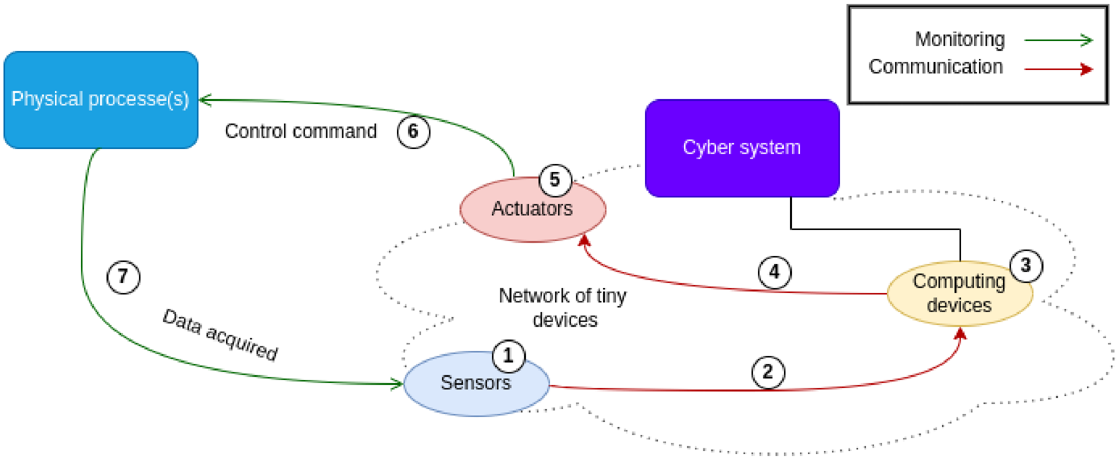

Figure 2.1: CPS main vulnerabilities

interaction, while the red ones represent the communication within the cyber system. The numbers point to the parts of the CPS that are subject to vulnerabilities.

By summarizing the classifications done by Wang et al. [3], Wazid et al. [8] and Singh et al. [4], we obtained the resulting vulnerabilities mapping:

Communication attacks

These attacks have the potential to be operated on all communication links, i. e. at 2, 4, 6 and 7 in figure 2.1. Communication attacks can include:

Eavesdropping: a passive attack where the attacker is listening to a communication between the two or more nodes of the system. This harms the *confidentiality* of the communication.

Man-In-The-Middle (MITM): the base concept is the same as the eavesdropping attack except that the attacker is able to intercept the communication packets and thus to modify them. This attack can impact the *confidentiality* and the *integrity* of the communication.

Network or routing attacks

We put in this category the attacks that are the result of a changing of behaviour from one element of the system that can impact changes in the rest of the network. All parts of the system from 1 to 5 in figure 2.1 could be impacted by such attacks.

Blackhole attack: the attacker is able to corrupt one or several node in the networked system so they advertise their neighboring nodes. This way, they become more attractive in the path-finding algorithm of the nodes. Nevertheless, once the blackhole nodes receive a packet, they drop it. This attack may be called Sinkhole attack in literature while designing Blackhole attacks as several nodes dropping packets, without making themselves more attractive. This attack may perturb the *availability* of some part of the networked system.

Greyhole attack: this attack is the same as the Blackhole attack but not all the packets are dropped. A filter is used to select which packets to drop. As well as for the Blackhole attack, the availability is impacted.

Wormhole attack: at least two nodes are required to carry out this attack. These nodes are normally not able to communicate with each other but the attacker will upgrade them so they can. This can be done by several ways from simply adding the route in their table to modify the nodes to increase their emission and reception ranges, in the case of they are too far away from each other. This attack does not affect any of the CIA parameters by itself but it can help multiple follow-up attacks which damage one or several of them.

Physical attacks

Physical attacks can be done on the devices that are the closest to the real world, thus on the actuators and on the sensors (1 and 5 in figure 2.1). They can be:

Side channel attack: the attacker analyze physical parameters that may vary depending on algorithms implementations and their inputs and is then able to find secret information. This impact the *confidentiality* of the system.

Fault injection: the attacker injects a physical fault in the device to change its behaviour in order for it to be malicious. This can be done by injecting quick voltage faults or electromagnetic faults, etc. Such attack have the potential to extract secret information or bypass system security, for example. As well as for the Side Channel Attack, *confidentiality* is compromised here.

Jamming attack: the attacker jams the communication hence no packets can be sent and received between the sensor and the signal transforming device. Jamming attacks impact the *availability* of part of the system.

Application attacks

Malware spreading attack: the attacker spread a piece of malicious code into one or more devices of the system. What the code does vary a lot from revealing secret keys to making a service unavailable, and thus create a Denial of Service (DoS), for example. This kind of attack is mostly observable on the computing devices (3 in figure 2.1). As a result of its diversity, any of the three parameters of the CIA model can be harmed.

Other kind of attacks

Here are attacks that can be part of several of the previous sections.

DoS attack: the attacker put down a device so it cannot work anymore. This device is often chosen strategically. Ways to achieve this attack are numerous. For example, it can be by sending an extremely high number of messages so a server receiving them is flooded and cannot work properly. If the attack is carried out by several device instead of one, we call that a Distributed Denial of Service (DDoS) attack. Depending on whether one considers the vulnerable part to be the one that is exploitable or the one that suffers the damage, DoS is located either on 2 or on 3 in figure 2.1. It affects the *availability* of the system.

2.1.3 Cyber-physical Systems simulation

In computer science, simulation is the act of digitally representing a phenomenon using a model. It is often used when it is not possible or too expensive to produce the phenomenon in the real world, or when one wants to represent abstract concepts. Simulation is also a way to test our work without having to impact the real world. Critical systems failures can put lives in danger.

The value of CPS simulation resides in that CPSs already have information about the real world in a digital environment. They are studied since several years now and are still confronted to simulation questions as shown by Thule et al. work [9], a 2019 paper presenting a framework for CPS simulation.

2.2 Multi-Agent Systems

A MAS is a system with a collection of, at least, two agents, cooperating with each other. These agents are entities containing an algorithm defining their behavior and making them autonomous in their decision making to achieve a common goal. If we add capabilities constraints like computing or memory limitations, then we are talking about embedded agents. As specified in section 1.4, MAS have been discussed a lot in the computer science field. It is a model that can be used at different level and in different fields. For example, a MAS could be a software working on multiple autonomous threads, representing the agents, on a computer. In this case, the separation of the agents is at an application level. Another use case could be a fleet of drones interacting with each other with one drone corresponding to one agent. This diversity of application areas comes from the high customisability and scalability of MAS.

2.2.1 Multi-Agent System and security

MAS are not the first choice for security or safety concerns, due to their autonomy and decentralization that makes them complicated, or even impossible, to monitor. However, the approach can solve problems coming from centralization like Single Point Of Failure (SPOF). For example, Cai et al. [10] propose an Intrusion Detection System based on an Agents approach where different kind of agents are responsible for monitoring different kind of resources in the system.

Moreover, a subject that is often associated to MAS is Trust Management System (TMS). The TMS define how an agent can decide to trust another agent or not, depending on specified conditions. It also define the behaviour of the agent in both cases.

Another interesting addition that can be done is a way to protect the communications between the agents. This can be done through message signature and encryption.

Building a MAS integrating a TMS with secured communications make a system easier to protect in the way that securing one agent is simpler than to monitor an entire system.

2.2.2 Multi-Agent System tools

This section aims at comparing several tool for simulating MAS. To gather the tools we are going to talk about, we based our research on previous works done at the LCIS laboratory [11][12] in addition to some personal ones as specified in 1.5. Among all these simulators, our first criteria of selection was on the version number. We did not select any tools under version 1.x, in other words that are still at a beta step of development. The second one was on the running platform. We must be able to run the simulator on Debian 11 (Bullseye).

First selection

GAMA [13] is an open source tool for agent based simulation allowing to incorporate agent locations thanks to the GAML language. It was created in 2019.

JACK Intelligent Agents [14] is a multi-agent system development framework written in Java made in 1999. It has the particularity to use the Belief–Desire–Intention (BDI) model.

Java Agent DEvelopment (JADE) [15] is also a Java framework for agent-based development but with the difference that JADE is Foundation for Intelligent, Physical Agent (FIPA) compliant. JADE is open source and was developed in 2001.

SARL [16] is an open source general-purpose agent-oriented programming language that comes with the Janus execution platform[1]. SARL was developed in 2014.

Multiagent Development Kit (MaDKit) [17] is an open source library, written in 2000 in Java, for simulating MAS.

Mesa [18] is an open-source agent-based modeling framework. It was written in python and first released in 2021. Mesa comes with a web-browser visualization and aims at offering an easy and customisable way to create agent-based models.

NetLogo [2] is a simulator with the goal to simulate natural or social phenomena. It is continuously improved by the Center for Connected Learning and Computer Based Modeling (CCLCBM). NetLogo is open source and comes with its own language and a web-browser interface.

Python Agent DEvelopment (PADE) [19] is another MAS framework in Python that is FIPA compliant. It provides an open source tool for developing, executing and managing a MAS. PADE was created in 2019 for application on power grids.

[1] http://www.sarl.io/runtime/janus/

[2] https://ccl.northwestern.edu/netlogo/

Smart Python Agent Development Environment (SPADE) [20] is a development platform in python using the Jabber communication framework for agents communications. SPADE is also open source and FIPA compliant. It was developed in 2006.

The chosen one

From the pre-selection we made, Mesa stands out. It is one of the most recent work and aims at being easy to use which is needed in our project since we don't have much time to spend to learn new tools. Python is also a language with a big and active community which makes support on this language more accessible. On top of that, the tool was used in previous and current projects done at the laboratory [11] thus we can get help on how to handle Mesa and the possible difficulties we could encounter.

2.3 Digital Twins

A Digital Twin is a virtual representation of a physical object or system. It is a digital replica of a physical entity, which can be used to model and simulate the real-world characteristics and behaviors. DT are typically created by using data from sensors and are updated in real-time as new data becomes available. They can be used for a wide range of applications, including design and engineering, manufacturing and production, operations and maintenance, and more.

The concept is arduous to frame because most of the subjects working with real-time simulation can, directly or not, to DT. For example, the work done by Mrissa et al. [21] presents a concept of avatars as a runtime environment and a virtual abstraction of physical objects.

2.3.1 Digital Twins for security

Since DT tend to have the same behaviour as the physical entity they are a twin of, they are an interesting tool to monitor complex systems. For example, they can be used to predict malfunctioning or malicious behaviour, like proposed by the Automatic Network Guardian for Electrical (ANGEL) [22].

They can also be used to simulate and then test these complex systems. Thus we can identify and address potential vulnerabilities existing on the system. These vulnerabilities can then be prevented or a recovery system can be set up, mitigating the effects of cyber attacks.

Additionally, DT can be used to evaluate how effective are different security measures and strategies, by simulating how a system would behave under different scenarios and with different security measures in place. This can help organizations choose the most effective ways to protect their systems from cyber attacks.

Bibliography

[1] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert, “Trends in automotive communication systems,” *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1204–1223, 2005.

[2] CPS Steering Group, “Cyber-Physical Systems Executive Summary,” Chicago, IL, USA., Mar. 2008.

[3] E. K. Wang, Y. Ye, X. Xu, S. M. Yiu, L. C. K. Hui, and K. P. Chow, “Security Issues and Challenges for Cyber Physical System,” in *2010 IEEE/ACM Int’l Conference on Green Computing and Communications & Int’l Conference on Cyber, Physical and Social Computing*, Dec. 2010, pp. 733–738.

[4] S. Singh, N. Yadav, and P. K. Chuarasia, “A Review on Cyber Physical System Attacks: Issues and Challenges,” in *2020 International Conference on Communication and Signal Processing (ICCSP)*, Jul. 2020, pp. 1133–1138.

[5] L. Chen, F. Hu, S. Wang, and J. Chen, “Cyber-physical system fusion modeling and robustness evaluation,” *Electric Power Systems Research*, vol. 213, p. 108654, Dec. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779622007246

[6] C.-U. Lei, K. Wan, and K. L. Man, “Developing a Smart Learning Environment in Universities Via Cyber-Physical Systems,” *Procedia Computer Science*, vol. 17, pp. 583–585, Jan. 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050913002081

[7] Y. P. Tsang, T. Yang, Z. S. Chen, C. H. Wu, and K. H. Tan, “How is extended reality bridging human and cyber-physical systems in the IoT-empowered logistics and supply chain management?” *Internet of Things*, vol. 20, p. 100623, Nov. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660522001044

[8] M. Wazid, A. K. Das, S. Kumari, and M. K. Khan, "Design of sinkhole node detection mechanism for hierarchical wireless sensor networks," *Security and Communication Networks*, vol. 9, no. 17, pp. 4596–4614, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1652

[9] C. Thule, K. Lausdahl, C. Gomes, G. Meisl, and P. G. Larsen, "Maestro: The into-cps co-simulation framework," *Simulation Modelling Practice and Theory*, vol. 92, pp. 45–61, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X1830193X

[10] Y. J. Cai, X. Y. Cheng, and Y. Pan, "Solutions of Single Point of Failure in Intrusion Detection System," *Applied Mechanics and Materials*, vol. 128-129, pp. 285–288, 2012. [Online]. Available: https://www.scientific.net/AMM.128-129.285

[11] Bonnet Adrian, "Sécurisation des communications dans un système multi-agents embarqués," LCIS, Tech. Rep., 2021.

[12] Derdour Najoua, "Conception et réalisation d'un protocole de gestion de confiance dans les systèmes multi agents," LCIS, Tech. Rep., 2020.

[13] P. Taillandier, B. Gaudou, A. Grignard, Q.-N. Huynh, N. Marilleau, P. Caillou, D. Philippon, and A. Drogoul, "Building, composing and experimenting complex spatial models with the GAMA platform," *GeoInformatica*, vol. 23, no. 2, pp. 299–322, Apr. 2019. [Online]. Available: https://doi.org/10.1007/s10707-018-00339-6

[14] P. Busetta, R. Rönnquist, A. Hodgson, and A. Lucas, "JACK Intelligent Agents - components for Intelligent Agents in Java," *www.agentlink.org*, Jan. 1999. [Online]. Available: http://www.agent-software.com.au/media/research/jack/busetta99jack.pdf

[15] F. Bellifemine, A. Poggi, and G. Rimassa, "Developing multi-agent systems with a fipa-compliant agent framework," *Software: Practice and Experience*, vol. 31, no. 2, pp. 103–128, 2001. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/1097-024X%28200102%2931%3A2%3C103%3A%3AAID-SPE358%3E3.0.CO%3B2-O

[16] S. Rodriguez, N. Gaud, and S. Galland, "SARL: a general-purpose agent-oriented programming language," in *the 2014 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*. Warsaw, Poland: IEEE Computer Society Press, 2014.

[17] O. Gutknecht and J. Ferber, “MadKit: a generic multi-agent platform,” in *Proceedings of the fourth international conference on Autonomous agents*, ser. AGENTS ’00. New York, NY, USA: Association for Computing Machinery, 2000, pp. 78–79. [Online]. Available: https://doi.org/10.1145/336595.337048

[18] J. Kazil, D. Masad, and A. Crooks, “Utilizing python for agent-based modeling: The mesa framework,” in *Social, Cultural, and Behavioral Modeling*, R. Thomson, H. Bisgin, C. Dancy, A. Hyder, and M. Hussain, Eds. Cham: Springer International Publishing, 2020, pp. 308–317.

[19] L. S. Melo, R. F. Sampaio, R. P. S. Leão, G. C. Barroso, and J. R. Bezerra, “Python-based multi-agent platform for application on power grids,” *International Transactions on Electrical Energy Systems*, vol. 29, no. 6, p. e12012, 2019, e12012 ITEES-18-0867.R2. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/2050-7038.12012

[20] M. E. Gregori, J. P. Cámara, and G. A. Bada, “A jabber-based multi-agent system platform,” in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS ’06. New York, NY, USA: Association for Computing Machinery, 2006, p. 1282–1284. [Online]. Available: https://doi.org/10.1145/1160633.1160866

[21] M. Mrissa, L. Médini, J.-P. Jamont, N. Le Sommer, and J. Laplace, “An Avatar Architecture for the Web of Things,” *IEEE Internet Computing*, vol. 19, no. 2, pp. 30–38, Mar. 2015, conference Name: IEEE Internet Computing.

[22] W. Danilczyk, Y. Sun, and H. He, “Angel: An intelligent digital twin framework for microgrid security,” in *2019 North American Power Symposium (NAPS)*, 2019, pp. 1–6.