Sixth LACCEI International Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2008)
"Partnering to Success: Engineering, Education, Research and Development"
June 4 – June 6 2008, Tegucigalpa, Honduras.

A Course on Security for Critical Infrastructure Systems

Eduardo B. Fernandez and Maria M. Larrondo-Petrie
Florida Atlantic University, Boca Raton, Florida, USA, ed@cse.fau.edu, petrie@fau.edu

ABSTRACT

Critical infrastructures are the systems that support our everyday life and include areas such as agriculture, information and telecommunications, food, energy, water, transportation, public health, and finance. We need to protect the information necessary to control and coordinate these systems as well as to control access to the physical structures involved. This information is usually embodied in a process control system (PCS) with its corresponding information system. A PCS typically includes a supervisory, control, and data acquisition (SCADA) system, which monitors and controls switches, valves, and physical quantities (temperature, pressure) and collects and logs field data. SCADA systems are distributed systems, including workstations, wired and wireless sensors, and application software. Databases contain the necessary information. SCADA requirements include 24/7 availability, real-time operation, survivability, and remote control. Their protection includes security and reliability concerns, including authentication, authorization, intrusion detection as well as fault tolerance measures. An extra dimension is the need for safety, avoiding damage to people or costly structures. We have developed a course that provides an understanding of how to coordinate hardware and software to provide data and network protection against internal and external attacks. We study the systems involved through the use of object-oriented patterns and formal models. We analyze how to perform a systematic analysis of attacks against the infrastructure. We see which defenses are available and how to apply them. We study the effect of errors on security and safety. Another aspect considered is the effect of system architecture on security and reliability. Finally, we consider development processes to build secure and safe systems.

Keywords: security, critical infrastructure, software engineering

1. INTRODUCTION

Infrastructure systems are needed to sustain a civilized life. Infrastructure systems are found in transportation, finance and banking, government, chemical, energy, oil and gas production and distribution, health services, information management, water (drinking and irrigation), emergency services (fire, police), garbage collection, and other areas. All of these infrastructure functions are controlled by systems which are complex and becoming increasingly interdependent; each system typically depends on one or two other systems. Some are even mutually dependent, e.g. electric power generation may require oil and oil production may require electricity [Ami02].

Infrastructure control systems are often distributed and real-time, and may include embedded devices and wireless links. Many system components are remotely deployed, have unique constraints, and may be physically inaccessible for maintenance, or physically accessible for attack. More and more systems use the Internet as a communication medium, and are thus accessible in cyberspace as well. Infrastructure systems are vulnerable to a variety of unintentional errors: equipment failure, human errors, weather, and other accidents. They are also the object of intentional attacks, both external (i.e. hackers) and internal (insiders/saboteurs) [Mil02].

Security is a fundamental objective of any system where there are assets that are targets of attacks because of their economic value or potential for disruptive impact. Until now most studies of security have focused on the protection of information assets. Recent terrorist threats have brought interest in the protection of physical assets as well. Correspondingly, the studies of threats to infrastructure have focused on safety. Safety is the freedom from unacceptable risks, including threats to human lives, the environment, or to costly facilities. Safety is often

defined with assertions on how to avoid unsafe conditions, e.g. an elevator must not open its doors when moving. These assertions are related to specific states of a control system. Safety is often confused with reliability or with availability. Some perceive safety as providing "zero defect" or "failure free" software, others consider it a matter of providing uninterrupted service. Heimdahl complains of the lack of knowledge about safety by software engineers [Hei07].

Until now, security and safety concerns were developed separately. However, it is not enough to apply the same security measures used in IT systems to critical systems. While superficially there are many commonalities, delving deeper one finds many differences [Nae07]. This implies the need for a specialized use of security mechanisms and methodologies.

Hazards are states or conditions of the system, that when combined with some state of the environment, lead to mishaps. A mishap is the occurrence of an unwanted event that leads to human or environment harm. Mishaps happen because a dangerous situation was not considered in the requirements, or because there was a failure in some unit of the system, or because there was an intentional action taken against vulnerable elements of the system, as discussed below. Our point is that safety is dependent on the reliability and security of the system. Conversely, safety features or restrictions (e.g. emergency access) or reliability failures can lead to security breaches. However, high reliability and security do not guarantee safety.

Leveson categorizes hazards into three groups [Lev95]:

- The system is not available.
- The system generates an incorrect output.
- The system misses a hard deadline.

Security attacks can deny service, illegally modify system state, or introduce artificial delays. In other words, security attacks may directly affect safety through all three hazard categories. Because infrastructure systems usually have databases used to control many devices, attacks to these databases can have a large effect on safety. We need to produce systems where security and safety are built together. Safety requires reliability and methods to improve reliability are also of interest in this context.

We have developed a course on the security of safety-critical systems where we discuss security, safety, and reliability and their mutual effect as well as appropriate development methodologies. The course tries to provide a perspective of the problems involved in protecting the information related to critical infrastructure and the avoidance of hazards. Engineers need to understand how to coordinate hardware and software to provide data and network protection against internal and external attacks. We model the systems involved through the use of patterns and formal models [Ken98]. We discuss here the main ideas of this course, which we have just introduced. The course is given to upper undergraduates and to graduate students.

Section 2 discusses some background on control systems, while Section 3 describes the structure of the course. Section 4 shows some of the material we cover in the course as an illustration of its range. We end with some conclusions.

2. PROCESS CONTROL SYSTEMS

An *Industrial Control System* (ICS) or *Process Control system* (PCS) is a system to make the output of a process satisfy some requirements. They are typically based on the use of feedback loops. ICSs are implemented in several ways, including supervisory control and data acquisition (SCADA) systems, distributed control systems (DCS), and Programmable Logic Controllers (PLCs). The most complex of these configurations is the SCADA system [Sto06].

Basically, a SCADA system is composed of field units, a central controller, and communication networks that connect these components. A field unit consists of field devices and a local programmable logic controller (PLC).

Field devices, such as actuators and sensors, are monitored and controlled by local PLCs. The central controller is generally geographically separated from these field units and typically has advanced computation facilities. A typical central controller may be equipped with data servers, Human-Machine Interface (HMI) stations, and other servers with advanced computation capabilities to aid the operators in managing the entire plant. The functions of the central controller include sending settings to the field units, sending commands to the field units, and receiving status information from the field units. The functions of the field units include monitoring the environment, taking actions on the environment, and sending status information and/or alarms to the central controller if necessary. The functions of the communication networks include forwarding data and commands.

Until recently, SCADA systems were electronically isolated from all other networks and hence not likely to be accessed by outside attackers. As a result, the security issues of a SCADA system focused mostly on physical security. However, the growing demands of the industry for increased connectivity between SCADA systems and corporate networks (and/or the Internet) have resulted in an increase in security threats and vulnerabilities that are not limited to physical attacks. A recent study shows that prior to 2000, almost 70% of the reported incidents of SCADA systems were either due to accidents or to disgruntled insiders acting maliciously. Since 2001, apart from an increase in the total number of reported incidents, almost 70% of the incidents were due to attacks originating from outside attackers [Byr04].

Attacks against the central controller and the network are more harmful since they may disable the whole system whereas attacks against field units only affect specific units. However, there are more attacks at the lower levels due to the unique implementation of these systems. These include the effect of resource constraints, e.g. lack of computational power to apply cryptography, cryptography takes time and produces delays which may affect real-time deadlines, and others [Nae07]. Physical attacks are also possible because of the distribution of the controllers and sensors and any analysis must show ways to define appropriate policies that can neutralize or mitigate the identified threats.

3. STRUCTURE OF THE COURSE

The pre-requisites for the course include concepts of computer hardware and software architectures, as well as some knowledge of object-oriented concepts, in particular UML modeling. We comment below on each topic covered in the course.

OUTLINE:

1. **Context and motivation**. Importance of infrastructure in our life. Possible attacks and effect of errors. Security and reliability objectives. Complex systems. Review of UML and patterns. Security patterns. Standards and regulations.

   We have written a book on security patterns [Sch06] and we introduce some of them here, others are introduced when needed. We discuss the need to comply with regulations, e.g. HIPAA [hip].

2. **Critical Infrastructure.** Features and requirements. Standards for networked systems. Need for security, safety, and reliability (availability and fault tolerance). Process control systems, information systems, and sensors.

   Mostly material from [Sto06].

3. **Security and reliability objectives**. Systematic analysis of attacks against the infrastructure. Attack patterns. Overview of defenses (countermeasures). Effect of errors. Protection against errors.

   Following the approach of [Fer06a], we systematically enumerate the threats against a system by considering its use cases and activities, and analyzing possible ways of subverting them. A simplified version of this approach looks at possible attacks against each unit of a system if its structure is predefined. SCADA example. We illustrate the use of attack patterns [Fer07a].

4. **Countermeasures I.** Authentication. Access control models. Physical access control. Comparison and voting for fault tolerance. Safety assertions.

   Policies and mechanisms to control threats and errors. Security patterns for access control and authentication. Dependability patterns.

5. **Countermeasures II.** Cryptography. Network security. Viruses and worms. Firewalls and IDS. Web services and agents. Redundancy to protect against denial of service. Formal approaches.

   Program correctness is not enough to guarantee safety, a program can do more than its intended functions. Formal methods help for algorithmic parts. But many aspects such as interfaces with the environment, timing, implementation details, and other cannot be fully formalized. We need also semi-formal approaches. UML models complemented with OCL constraints [War03] or Petri nets are a good approach to semi-formal solutions. The integration of semi-formal and formal models is an interesting problem in itself and we study some aspects of it. Cryptography is studied from a systems (not algorithmic) viewpoint.

6. **Dependable architectures for infrastructure systems**. Effect of system architecture on security and reliability. Secure and reliable development process.

   Software for such systems must be evaluated within the context of the complete system, including the effect of the environment. There has been a good amount of work on identifying and prioritizing hazards [Lev95]. We need to do system and hazard analysis to verify safety and study how the final system controls the threats. We discuss an approach to develop secure systems [Fer06b] and extend it to consider safety and reliability aspects.

The grading of the course is based on four or five assignments (25%), and a take-home exam (75 %). This exam is really a mini-project, where the students need to design a system that satisfies security and safety constraints. The Appendix shows the exam for the last offering of this course. The next section shows a typical unit of the course, where we discuss physical access control (from Unit 4 of the outline).

4. PHYSICAL ACCESS CONTROL

Homeland security has brought an added interest in control of access to buildings and other physical structures. The need to protect assets in buildings and to control access to restricted areas such as airports, naval ports, government agencies, and nuclear plants to name a few, created a great business opportunity for the physical access control industry and a good amount of interest in the research community. One of the results of this interest was the recognition that access control to information and access control to physical locations have many common aspects. The most basic model of access control uses a tuple (s,o,t), subject, object, access type [Gol06]. If we interpret s as a person (instead of an acting executing entity), o as a physical structure (instead of a computational resource), and t as a physical access type (instead of resource access), we can make an analogy where we can apply known results or approaches from information access control. The unification of information and physical access control is just beginning but the strong requirements for infrastructure protection will make this convergence happen rapidly. Another issue is the fact that there are standard network protocols for building automation, e.g. BACnet [bac06], which are totally different from the protocols used for manufacturing automation, e.g. DNP3 [Maj05]. Both types of protocols define security standards, which means that a building intended for manufacturing would have two sets of incompatible security standards. We need some way to abstract the security requirements of the complete system without regard to specific details of each standard.

One way to achieve this unification is using a conceptual abstraction for the definition of security requirements; we have combined analysis and security patterns for this purpose. Standards and products that deal with physical units use a set of common concepts that may appear different due to a different notation; patterns make this commonality apparent. Examining existing systems, industry standards, and government regulations, we have described, in the form of patterns, the relationship and definition of a core set of features a physical access control system should have. From these patterns, it is possible to define more specific patterns that can be used to build systems in a given protocol or to define new protocols.

We have presented several patterns for access control in physical units [Fer07b]. These include:

- **Alarm Monitoring.** Defines a way to raise events in the system that might require special attention, like the tampering of a door.
- **Relays.** Defines the interactions with electronically controlled switches.

- **Access Control to Physical Structures.** Applies authentication and authorization (RBAC) to the control of access to physical units including alarm monitoring, relays, and time schedules that can control when things will happen.
- **Physical Structure**. Defines the structure and use of physical sites such as buildings, parking lots, and similar, as well as their divisions and compartments.
- **Scheduler.** Provides timing information to control access.

We combine them with access control patterns such as Role-Based Access Control to control access to physical structures. Figure 1 shows an example of this combination. This model illustrates an important aspect of patterns: their ability to be combined to make up complex applications. We can also see in this Figure the use of design patterns [Gam94], e.g. Zones are described by the Composite pattern.

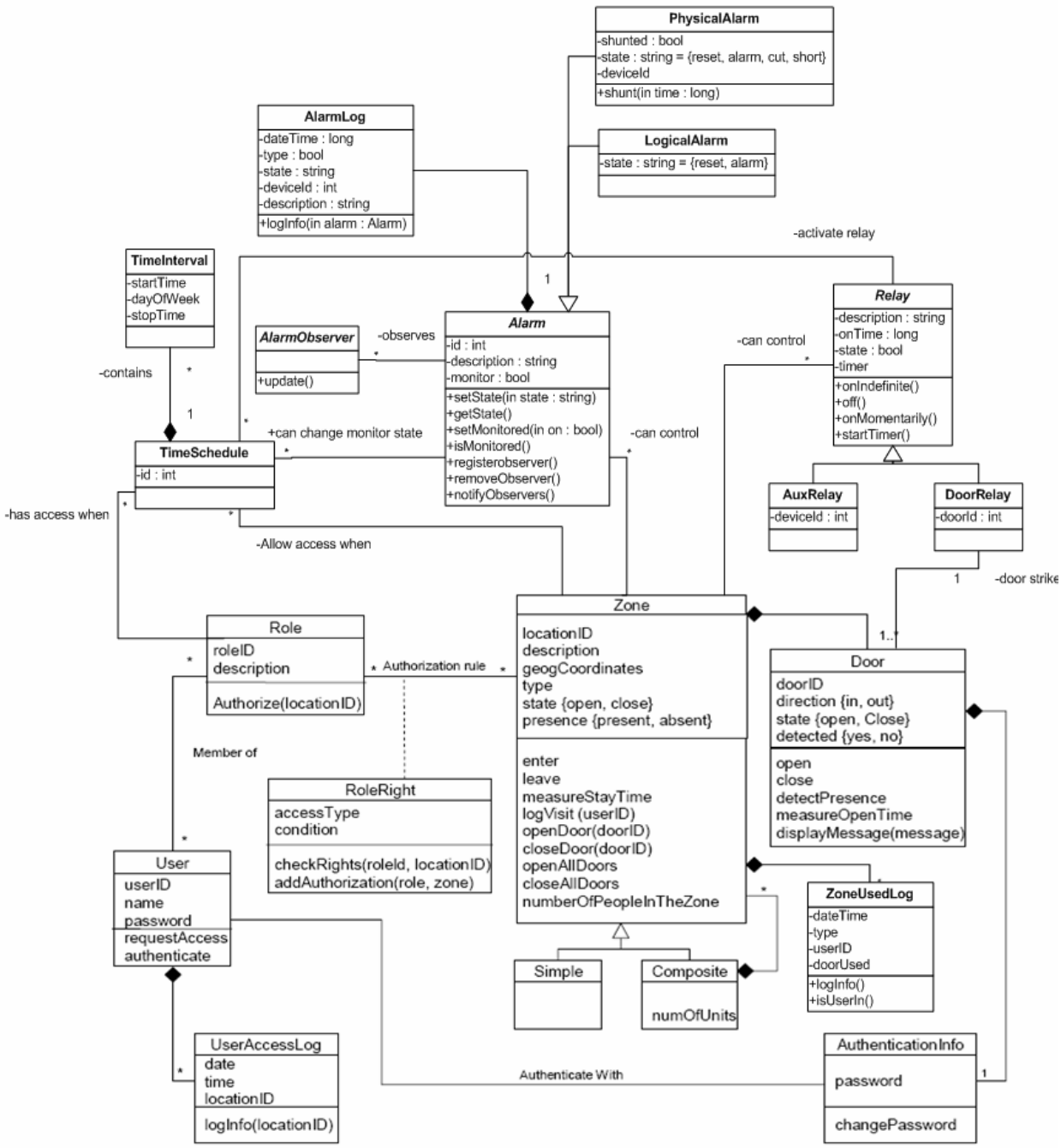

Figure 1. Class Diagram for Access Control to Physical Structures

5. CONCLUSIONS

All students make acquaintance with object concepts in their courses on data structures and introduction to programming and most of them also take a course on object-oriented design during their last years of undergraduate studies, which gives them the proper background to understand basic UML models. They see this course as a way to learn not only security but also to reinforce their knowledge of object-oriented software design. They learn clearly the difference between security, reliability, and safety, concepts which are often confused. The design-oriented approach makes the material suitable to be applied to real situations. As a result of the first offering of this course one of the students produced a paper with two patterns for SCADA systems that indicate the general structure of these systems and where different security mechanisms are required [Sha08]; another student is developing a paper and a thesis on fault tolerance patterns.

ACKNOWLEDGEMENTS

This work was possible thanks to partial support from a grant from DISA (Department of Information Security Assurance), administered by Pragmatics, Inc.

REFERENCES

[Ami02] M. Amin, "Security challenges for the electricity infrastructure", *Security and Privacy 2002* (Supplement to Computer), IEEE, 2002.

[bac06] SSPC 135/LSS-WG, October 2006. "Physical Access Control with BACnet" http://www.bacnet.org/Bibliography/BAC-10-06.pdf

[Byr04] E. Byres and J. Lowe. "The myths and facts behind cyber security risks for industrial control systems". *Proceedings.of VDE Congress*, 2004.

[Fer06a] E. B. Fernandez, M. VanHilst, M. M. Larrondo Petrie, S. Huang, "Defining Security Requirements through Misuse Actions", in *Advanced Software Engineering: Expanding the Frontiers of Software Technology*, S. F. Ochoa and G.-C. Roman (Eds.), International Federation for Information Processing, Springer, 2006, 123-137.

[Fer06b] E. B. Fernandez, M.M. Larrondo-Petrie, T. Sorgente, and M. VanHilst, "A methodology to develop secure systems using patterns", Chapter 5 in *Integrating security and software engineering: Advances and future vision*, H. Mouratidis and P. Giorgini (Eds.), IDEA Press, 2006, 107-126.

[Fer07a] E.B. Fernandez, J.C. Pelaez, and M.M. Larrondo-Petrie, "Attack patterns: A new forensic and design tool", *Procs. of the Third Annual IFIP WG 11.9 Int. Conf. on Digital Forensics*, Orlando, FL, Jan. 29-31, 2007. www.cis.utulsa.edu/ifip119. Chapter 24 in *Advances in Digital Forensics III*, P. Craiger and S. Shenoi (Eds.), Springer/IFIP, 2007, 345-357.

[Fer07b] E.B.Fernandez, J. Ballesteros, A. C. Desouza-Doucet, and M.M. Larrondo-Petrie, "Security Patterns for Physical Access Control Systems", in S. Barker and G.J. Ahn (Eds.), *Data and Applications Security XXI*, LNCS 4602, 259-274, Springer 2007. *Procs.of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, Redondo Beach, California, U.S.A, July 8-11, 2007.

[Gam94] E. Gamma, R. Helm, R. Johnson, J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, Boston, Mass., 1994.

[Gol06] D. Gollmann, *Computer security* (2nd Ed.), Wiley, 2006.

[Hei07] M. P. E. Heimdahl, "Safety and software intensive systems: Challenges old and new", *International Conference on Software Engineering, FOSE2007 Future of Software Engineering*, IEEE Computer Society, Washington, DC, USA, 2007, pp. 137-152.

[hip] HIPPA.ORG. http://www.hipaa.org/

[Ken98] E.A.Kendall, "Utilizing patterns and pattern languages in software engineering education," *Annals of Software Engineering*, vol. 6, 1998, 281-294.

[Lev95] N. G. Leveson, *Safeware: System Safety and Computers*, Addison-Wesley, 1995

[Maj05] M.Majdalawieh, F.Parisi-Presicce, D.Wijesekera, "DNPSec: A Security framework for DNP3 in SCADA Systems", *International Joint Conf. on Computer Information and Systems Sciences and Engineering*, Bridgeport, CT, USA, 10-20 December 2005.

[Mil05] A. Miller, "Trends in process control systems security", *IEEE Security and Privacy*, Sept./October 2005, vol. 3, no. 5, pp. 57-60.

[Nae07] M. Naedele, "Addressing IT security for critical control systems", *Proceedings of the 40th Hawaii International Conference on Systems Science (HICSS-40)*, January 2007.

[Sch06] M. Schumacher, E.B.Fernandez, D. Hybertson, F. Buschmann, and P. Sommerlad, *Security Patterns: Integrating security and systems engineering*, Wiley 2006.

[Sha08] Y. Shao, E.B.Fernandez, and J. Wu, "On building secure SCADA systems using security patterns", submitted for publication.

[Sto06] K. Stouffer, J. Falco, and K. Kent, *Guide to supervisory control and data acquisition (SCADA) and industrial control systems security*, Special Publication 800-82, National Institute of Standards and Technology (NIST), 2nd draft, http://csrc.nist.gov/publications/drafts/800-82/2nd-Draft-SP800-82-clean.pdf

[War03] J. Warmer and A. Kleppe, *The Object Constraint Language* (2nd Ed.), Addison-Wesley, 2003.

Appendix: Take-home exam of Spring 2008.

The ports of a country are fundamental for its economy, supplying people with food and goods, and exporting local products. Because of their open and distributed structure, they are very vulnerable. Palm Beach County has hired you as a consultant to reinforce their ports against severe disruptions or catastrophic events.

(Directions: Provide diagrams with brief annotations. Do not consider purely physical solutions, e.g. guards)

a) Consider a set of the five or six most important use cases for this system and analyze possible security and safety threats.

b) Describe roles for the actors involved in your use cases. Propose policies for stopping or mitigating the identified threats, including necessary rights for the actors.

c) Draw a UML class diagram for the port system without including any protection. Build this diagram from your use cases and from general knowledge about ports.

d) Draw a new UML class separate diagram for the port, where you have added security and safety mechanisms to implement your policies.

BIOGRAPHIC INFORMATION

EDUARDO B. FERNANDEZ (http://polaris.cse.fau.edu/~ed), is a professor in the Department of Computer Science and Eng. at Florida Atlantic University. He has published numerous papers and four books on different aspects of security, object-oriented analysis and design, and fault-tolerant systems. He holds a Ph.D. degree from UCLA. His industrial experience includes 8 years with IBM.

MARIA M. LARRONDO-PETRIE: Dr. Larrondo-Petrie is Professor and Associate Dean of Academic & International Affairs in the College of Engineering and Computer Science at Florida Atlantic University. She is the Executive Director of LACCEI, Vice President of IFEES (International Federation of Engineering Education Societies), and on the boards of the Red Cartagena de Ingenieria, the ASEE (American Society of Engineering Education) International Division Board, ASEE Women in Engineering Board and ASEE Minority Division Board. She was a past President of Upsilon Pi Epsilon International Honor Society for the Computing and Information Disciplines. Her research interests are in complex system modeling, security, and global engineering education.

Authorization and Disclaimer

Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.