Information & Security: An International Journal

Pomorova & Lysenko, vol.35, 2, 2016, 133-150 https://doi.org/10.11610/isij.3507

FORMAL AND INTELLIGENT METHODS FOR SECURITY AND RESILIENCE: EDUCATION AND TRAINING ISSUES

Oksana POMOROVA and Sergii LYSENKO

Abstract: This article presents the results of the implementation of TEMPUS SEREIN project in the Khmelnitsky National University. The main challenge was to develop curriculum and course materials for masters and PhD students. The authors introduce the module description of the course "Formal and Intelligent Methods for Security and Resilience." The module deals with the issues of development and usage of formal methods for designing secure software systems and implementation of formal methods for assuring security of computer networks. We also present some issues on the usage of intelligent systems for security and address the questions of enhancing systems' resilience. The implementation of the developed training course will improve the quality of education and will make graduates more successful on the labour market.

Keywords: formal methods, security, resilience, artificial intelligence, B-method, resilient system, software development, secure protocol, computer networks.

Introduction

Today relevance of cybersecurity is beyond any doubt.[1] Every day each of us is faced with the need to use information technology. In this situation, the actual problem is providing of the security aspects for such technologies and hence, there is a need in the specialists in security. However, the volume of specialist training in information security and cybersecurity in Ukraine is insufficient.[2,3] The peculiarity of training in this field is a dynamic development of the object, and therefore learning content. This problem is to be resolved by a timely updating of content and integration of basic and flexible training course.

The way to solve this problem is to involve and implement international experience. Such task is imposed on a Tempus − the European Union's programme that supports the modernization of higher education in the Partner Countries of Eastern Europe, Central Asia, the Western Balkans and the Mediterranean region, through university

cooperation projects.[4] There are great attempts of the bringing new knowledge about security and resilience,[5,6,7] but due to dynamic evolution of the information technologies, the new knowledge and experience in cybersecurity are to be involved in educational process.

There is no surprise, that the TEMPUS SEREIN, which is acronym from official name "Modernization of Postgraduate Studies on Security and Resilience for Human and Industry Related Domains," is appeared. The project is financed by the TEMPUS programme. It encourages higher education institutions in the EU Member States and partner countries to engage in structured cooperation through the establishment of "consortia". TEMPUS SEREIN project addresses the business and society demand on high-qualified specialists on cybersecurity assessment and management by introducing international master/doctoral/in-service programs on Cybersecurity and Resilience.[8] The main goal of the project is to produce new generation of engineering and research staff capable of performing constructive development in cybersecurity assessment and ensuring.

As a partner of the consortium, Khmelnitsky National University is responsible to develop the course CP1 "Formal (Intellectual) Methods for System Security and Resilience."[9]

The main task of this course is to create a knowledge base for formal methods for System Security and Resilience and to provide prerequisites for practical use of B-method for specifying and designing computer systems and software with formal notation. The study also expands the current research on artificial intelligence in cyber defence.

What are Formal Methods?

Formal methods are a particular kind of mathematically based techniques for the specification, development and verification of software and hardware systems. The use of formal methods for software and hardware design is motivated by the expectation that, as in other engineering disciplines, performing appropriate mathematical analysis can contribute to the reliability and robustness of a design. Formal methods are best described as the application of a fairly broad variety of theoretical computer science fundamentals, in particular logic calculi, formal languages, automata theory, and program semantics, but also type systems and algebraic data types to problems in software and hardware specification and verification.[10] To be effective, formal methods are well integrated within industrial practice.[11,12] For this reason, most formal methods are equipped with methodological guidelines for a proper use in real-size system development.

Formal methods are applied in different areas of hardware and software, including routers, Ethernet switches, routing protocols, and security applications. Nowadays there are a great amount of approaches that demonstrate the importance and usage of the formal methods for security.[12,13]

The Notion of Resilience

Resilience is the property of a system to remain trustworthy despite changes. Changes of a different nature, whether due to failures of system components or varying operational conditions, significantly increase the complexity of system development. Therefore, advanced development technologies are required to build robust and flexible system architectures capable of adapting to such changes. Moreover, powerful quantitative techniques are needed to assess the impact of these changes on various system characteristics.[14-16]

Artificial Intelligence for Security

Artificial intelligence as a field of scientific research has been employed in different data mining and machine learning classification and prediction modelling schemes.[17] Numerous useful applications already exist in the field cyber defence. They belong to applications of artificial neural nets, support vector machines, genetic algorithms, multi-agent systems. It has become obvious that many more cyber defence problems can be solved successfully only when artificial intelligence methods are used. Wide knowledge usage is necessary in decision making, and the intelligent decision support is one of the yet unsolved problems in cyber defence.[18]

All above-mentioned questions are introduced in the developed module "CP1.Formal and Intelligent Methods for Security and Resilience."

CP1. Formal and Intelligent Methods for Security and Resilience: Module Description

This course incudes three modules. The first one deals with the question about the formal methods for Architecting Secure Software Systems. The second module presents some issues about the usage of the formal methods for computer networks, and the third one introduces the notions of the resilience and the principals of the resilient systems construction.

Introduction to Formal Methods

The first question the module unit gives an answer is what are the formal methods, what is the nature of formal methods, benefits and limitation of the use of formal methods and shows a symbiotic relationship between formal methods and security.[19]

Formal Analysis and Design for Security Engineering

The second module unit is devoted to the formal analysis and design for engineering security. This approach combines the goal-oriented requirements specification with the formal design specification in order to develop secure software in a constructive, provable and cost-effective way. It demonstrates how to provide a systematic and automated bridge between semi-formal security requirements and formal design and implementation using the formal methods.

The main problem that is covered in the unit is that the formal analysis and design for engineering security demonstrates the completeness and consistency of the security requirements, which are specified with knowledge acquisition for automated specifications when transformed to B formal specifications.[20,21]

Security design specifications and implementation are achieved by using the B formal method that preserves the requisite security requirement properties.[22]

Formal Analysis and Design for Engineering Security considers security-specific elements in a systematic and constructive way while considering security early in the development lifecycle. Moreover, employing Formal Analysis and Design for Engineering Security ensures a confidence for security evaluators in the evaluation of trusted software. Nevertheless, the main side effect of employing formal methods in development is the availability of sufficient traceability information at the various phases of development and maintenance allowing for more accurate impact analysis of security changes.[23] Also, this unit presents a requirements' driven software security engineering approach with demonstrates how to integrate the semi-formal requirements methods with formal design methods in order to produce the assured software security in a cost-effective manner, and how to bridge the gap between goal-oriented requirements and formal design for security-specific elements of software.

Learning outcomes of the course unit: apply formal analysis and design for security engineering to industry-related case studies in order to demonstrate the feasibility and effectiveness of the approach in building secure computer systems and software in a provable way.

Formal Methods for Architecting Secure Software Systems

This course unit is devoted to the problem of modelling and analysing the security properties in architecture designs. This lection deals with such the semi-formal approaches, formal approaches, integrated semi-formal and formal approaches, and aspect-oriented approaches. This section demonstrates the usage of the modelling language UML to model security non-functional properties.[24] Also in this module unit, the significant need of such formal methods as automated analysis of non-functional

properties is demonstrated. For this purpose, the architecture description languages, Petri nets, temporal logic are used.

In order to achieve the modelling efficiency provided by UML and the rigorous analysis provided by formal methods, the usage of a combination of semi-formal UML and formal methods are demonstrated, and as tools in the analysis the model checking and theorem prover based approaches are used.

This course unit also demonstrates the profit of different notations, tailored notations. It presents a quantity of notations that are suitable for modelling and analysing a comprehensive collection of security properties in software architectures.

Learning outcomes of the course unit are: to model and analyse the security properties in architecture designs; model security functional and non-functional properties; to use the automated analysis of non-functional properties by formal methods; use a combination of semi-formal UML and formal methods in order to achieve the modelling efficiency provided by UML and the rigorous analysis provided by formal methods; to use the model checking and theorem prover as the tools in the analysis of non-functional properties; to use of different notations, tailored notations for modelling and analysing a comprehensive collection of security properties in software architectures.

Formal Methods for a Certifiable Secure Software System

This course unit has introduced approaches for verifying security to the extent of the source code level. This unit covers issues of a building of the minimal state machine model, which is essential to prove that the model satisfies a defined some security property.[25] In addition, course unit takes up the question of the proving, that the security model satisfies the property using a mechanical verifier.

Course unit demonstrates the procedure of the code annotation is with preconditions and postconditions. After that, it is shown, that the code should be partitioned into the concepts of Event, Trusted, and Other Code. Finally, it should be demonstrated the conformance of the Event Code and the code preconditions and postconditions with the internal events, preconditions, and postconditions of the TLS; also, it should be shown, that the Trusted Code and the Other Code are benign. Tools such as model checkers and theorem provers are already available for verifying that a formal specification satisfies a security property of interest.[26]

Learning outcomes of the course unit: to use model checkers and theorem provers for verifying that a formal specification satisfies a security property of interest; automatically generate test cases that check source code annotations; automatically construct efficient provably correct code from specifications.

Formal Methods for Assuring Security of Computer Networks

CP1 course also includes a unit which deals with the problem of formal methods for an assuring the security of the computer networks. In this course unit as an example of formal methods usage, a single session-based protocol, designed to provide mutual authentication using a public-key infrastructure is presented.[27]

Unit brings into focus a single protocol and demonstrates how multiple logical systems and formal models are able to afford insight into security issues from a different of points of view.

Unit tells that the authentication of logic and the BAN logic are considered as special-purpose modal logics, which are produced to enable the reasoning about freshness and trust. They pinpoint on the principals that have to be prepared to accept and to believe with the purpose to trust in the correctness of a protocol.

In this course unit as a general-purpose language for describing and reasoning about the behaviour of concurrent systems the process algebra CSP is presented. It is well suited for reasoning about the high-level interactions and events that may occur during a run of a protocol.[28]

Learning outcomes of this course unit: to use the BAN logic and the authentication of logic in order to verify in the correctness of a protocol; use the process algebra CSP for describing and reasoning about the behaviour of concurrent systems and for reasoning about the high-level interactions and events that may occur during a run of a protocol; take into account the security properties and build methods for assessing the security of a system; use formal methods for the detection of weaknesses and possible attacks; use and apply tools that automatically translate abstract descriptions of security protocols into process-algebraic descriptions that can be analysed with model checkers.

Formal Methods for the Analysis of Security Protocols

In this module unit, the questions of the formal methods for the analysis of security protocols are taken up. Here, an approach for the study of sound abstractions of cryptography using process algebras is presented.[29] Unit shows how to design a simple, abstract language for secure distributed communications with two forms of authentication (but no explicit cryptography). Such language is applicable to a large class of protocols, and enables simple reasoning about security properties in the presence of active attackers, using labelled traces and equivalences. Also, this chapter introduces a process algebra for specifying and reasoning about (quantum) security protocols.

Another approach placed in this unit presents a method for verifying security protocols based on an abstract representation of protocols by Horn clauses.[30] This method is the foundation of the protocol verifier ProVerif. It is fully automatic, efficient, and can handle an unbounded number of sessions and an unbounded message space. It supports various cryptographic primitives defined by rewrite rules or equations. It is shown how to prove such security properties as authentication and process equivalences.

Another chapter of this unit tells how security protocols are analysed using a variety of tools and is focusing on a variety of properties. Here, the Dolev-Yao, Rational Attacker and General Attacker threat models are discussed.[31] The general threat model for security protocols based on set-rewriting that was adopted in AVISPA [32] is leveraged so as to express the General Attacker. Also, it is shown how to use the model checker SATMC [33] in order to automatically validate a protocol under the new threats, so that retaliation and anticipation attacks can automatically be found.

Learning outcomes of the course unit are: to model and verify security protocols, analyse security protocols by using the threat models.

Formal Methods for Security of the Wireless Systems

Another course unit of the module is devoted to the formal methods for security of the wireless systems.[34] In this unit, we deal with the question of the usage the well-known technique of process calculus to the modelling of secure features of wireless systems. In addition, the question about the trust-based security to develop security mechanisms specifically tailored for wireless systems is discussed.[35] For this purpose, different process calculi are presented. The first one is a timed process calculus for wireless systems (TCWS). Here the questions about timing aspects and communication interferences are focused. This approach demonstrates the calculus dealing with time and interferences for wireless systems, because other calculi rely on the presence of some MAC-level protocol to remove interferences. However, in wireless systems collisions cannot be avoided although there are protocols to reduce their occurrences.[36]

Another formal method called timed process calculus is presented in this unit. It is intended for modelling the local broadcast communications and time passing, operating a distinction in the labelled transition system between transitions for modelling sending and receiving actions and transitions for modelling the passage of time.

Also, module unit presents a notion of well-formedness over the networks and it shows a proof that it is preserved at run-time. In this unit, it is demonstrated the usability of the calculus to model the Carrier Sense Multiple Access scheme, a widely-used MAC level protocol in which a device senses the channel before transmitting.

Here, a main behavioural equivalence a timed variant of the reduction barbed congruence is provided.[35] As usual, when working with barbed congruence, it is useful to define a labelled bisimilarity as a proof-method. Thus, it is shown a soundness result which states that our labelled bisimilarity implies our timed barbed reduction congruence. Then it is proved a number of algebraic laws on well-formed networks using the bisimulation proof-technique.

Unit presents another process calculus for MANETs called A Calculus of Trustworthy Ad hoc Networks (CTAN). Here a mobility of nodes and trust-based security are focused on, and it is shown how to implement a trust model in the calculus. Trust relations among nodes are represented by an association of a security level to them. The main objective of the model is to isolate bad nodes, i.e. nodes that do not behave as expected. In order to achieve this affect supported node revocation is involved. The process calculus for wireless networks embodying a trust model is presented here.[37]

In this module unit, it is shown that A Calculus of Trustworthy Ad hoc Networks is able to satisfy safety properties, aiming at guaranteeing that only authorised nodes receive sensible information.[38] Finally, it is presented how to use the calculus to formalise and analyse a secure version of the leader election algorithm for MANETs.[39] Then it is shown how to encode the endairA routing protocol for ad hoc networks is provided.

The last question of the unit demonstrates some timing aspects in order to express the relationship between trust and time. Using such calculus, it is possible to formalise the secure, on-demand routing protocol ARAN that uses cryptographic certificates with a time validity.

Learning outcomes of the course unit are to use different technique of process calculus to model of secure features of wireless systems.

Intellectual Methods for Security

Course unit includes a review of the artificial intelligence methods in their application to network intrusion detection system.[40] Such techniques as artificial neural networks, support vector machines, genetic algorithms and fuzzy neural networks, some aspects of the usage of the multi-agent systems for security are described in this module.[41,42]

Unit presents an overview of intrusion detection systems and the recent advances in artificial intelligence; it presents a brief account of some artificial intelligence techniques and their applications in intrusion detection system.

Also, unit discusses the concept of hybridization and ensemblage of artificial intelligence techniques and the possible hybrid architectures that researchers could explore in their design and development of intrusion detection systems.

In module unit, a new multi-agent based approach for botnet detection in a corporate area network using fuzzy logic is proposed. The detection is performed in the situations of priori uncertainty of the botnet presence in the corporate area network with taking into account the botnet demonstrations in the several computer systems available in the network.[43,44] With the usage of fuzzy logic, the analysis of the botnets' actions demonstrations in the situation of the intentionally computer system reconnection is performed. Fuzzy expert system for making conclusion about botnet presence degree in computer systems is developed. Fuzzy expert system takes into account the demonstration degree of reconnected computer system, demonstration degree of probably infected computer systems and demonstration degree of other computer systems available in the corporate area network that probably were not infected.

Learning outcomes of the course unit: to architect of an intelligent system for information security management; build the adaptive and capable systems for discovering and building new knowledge for the information security domain; use the techniques based on Artificial Intelligence for information security management and cyber defence.

Resilient Systems

In this unit, an overview of the main concepts and properties appearing in the development of resilient systems are given.[45] The notion of "resilience" is consider as an evolution of the dependability concept and it is discussed how the goal-oriented development facilitates engineering of resilient systems. Unit gives attention to the dynamic system reconfiguration as the main mechanism for achieving system resilience. Since resilience is an evolution of the notion of dependability, majority of its concepts are grounded in the classical definitions proposed for dependability that are also discussed in this unit.[46,47] So such attributes of the dependability as availability, reliability, safety, integrity, maintainability, confidentiality are explained in this chapter. Also, the notion of the goal-based development is explicated as the resilience can be seen as a property that allows the system to progress towards achieving its goals.

As a development of a resilient system is a challenging engineering task, this module unit demonstrates that it should be significantly facilitated by the use of formal model-based techniques. Also, unit shows how to build a resilient system in a rigorous way and how to verify that the system specification meets the requirements. For this purpose, the formal methods are used. In this section as a formal development framework Event-B is presented. The Event-B formalism—a variation of the B Method [48]—is a state-based formal approach that promotes the correct-by-construction development approach and verification by theorem proving.[49] The Event-B framework was influenced by the Action Systems − a formal approach to model distributed, parallel, and reactive systems.

Unit of the module says that the quantitative assessment plays an important role in the process of resilient system development because it allows the developers to predict the impact of changes on such vital aspects as, e.g., reliability and performance. Quantitative analysis helps to find suitable trade-offs between these properties as well as evaluate the impact of different architectural alternatives on system resilience. Therefore, in this unit it is shown the possibility of integration of formal development in Event-B with quantitative resilience assessment.

Formal Development and Quantitative Assessment of Resilient Distributed Systems

In this unit, as a case study a process of the development and assessment of distributed resilient systems is presented.[50-54] Such process includes steps: model construction in event-B, resilience-explicit development based on functional decomposition, modelling component interactions, execution of a resilient-explicit goal-oriented refinement process, and an implementation of modelling and assessment of resilient architectures.

Thus, unit shows that a development of a distributed system in Event-B starts from creating an abstract system specification (model). After it shows how to build an Event-B model for a centralized system that exhibits the desired externally observable behaviour and properties, and how to transform the abstract model into a detailed system specification by gradually unfolding the system architecture, precisely defining the functional behaviour as well as deriving a detailed representation of component interactions. As the quantitative assessment of different resilience characteristics is an essential part of the system design for resilience, it is shown that Event-B models can be augmented with quantitative data and, are able to be as a basis for quantitative resilience assessment.

The next step is the resilience-explicit development based on functional decomposition. Here, unit demonstrates how to define the resilience-explicit refinement process for such systems. Then the question about the generic functional decomposition as a refinement step that results in defining a high-level execution flow is discussed. It is shown how to establish a connection between a global system failure and the corresponding failures in the execution flow. Further, it is demonstrated how refinement can be used for deriving the component- based system architecture and linking component failures with those in the system execution flow. The next step is to carry out a detailed analysis of component interactions. To facilitate reasoning about such cooperative behaviour, the components are considered as agents and a resilient distributed system as a multi-agent system. The multi-agent modelling enables to define the essential properties of cooperative agent activities.

The next step to be performed is the execution of the resilient-explicit goal-oriented refinement process. Module unit introduces the detailed description of such process. First, it is shown how to define a set of specification and refinement patterns that reflect the main concepts of the goal-oriented development. It is shown that the refinement approach is employed to support the goal decomposition process, thus allowing us to define the system goals at different levels of abstraction. So, the same strategy for development of distributed goal-oriented systems by refinement is followed. Unit shows how to define the system goals, perform goal decomposition by refinement, and then how to introduce a representation of system agents, whose collaborative activities ensure goal reachability. Therefore, this resilience-explicit goal-oriented refinement approach aims at ensuring goal reachability "by construction." It allows the developers to systematically introduce the required reconfiguration mechanisms to ensure that the system progresses towards achieving its goals despite agent failures (thereby, address "negative" changes) or becomes more performant by using its agents more efficiently (thereby, address "positive" changes).

In order to achieve the system resilience a dynamic reconfiguration is used. It allows the system to adapt to changes by modifying its structure, inter-agent relationships and dependencies. Unit demonstrates how deformalize the possible interdependencies between goals and agents and to formulate the conditions for ensuring goal reachability in a reconfigurable multi-agent system. The proposed formalization gives a formal systematization of the introduced concepts and can be seen as generic guidelines for designing reconfigurable systems. Finally, in order to evaluate the impact of reconfiguration on the system performance and reliability, it is demonstrated how to augment the resulting Event-B models defining various reconfiguration scenarios with the necessary probabilistic information and how to quantitatively assess different reconfiguration strategies.

The resilience-explicit goal-oriented development approach assumes that the agents are sufficiently reliable. To validate such an assumption and to derive the constraints on agents' reliability, it is needed to employ quantitative analysis. Quantitative assessment is also required to evaluate the impact of various architectural solutions on the system performance and reliability. Integration with probabilistic model checking in PRISM allows us to achieve these objectives. Here it is shown how Event-B models are augmented with quantitative data and are transformed into input models for the PRISM model checker. As a result, quantitative assessment allows the designers to make informed design choices and develop systems with predictable resilience characteristics.

In the last part of this module unit it is demonstrated how a resilient-explicit refinement approach can be adopted to derive distributed architectures with the incorporated fault tolerance mechanisms. In this unit, it is demonstrated how to perform a re-

silience assessment of architectural alternatives by discrete event simulation in SimPy,[55] and how the quantitative analysis in SimPy allows us to evaluate the impact of a particular architectural solution on the system reliability/performance ratio.

Conclusions

The development of the training course within the SEREIN project has made it possible to eliminate knowledge gaps in the field of cybersecurity, especially in the questions of the resilient system development.

The implementation of the project SEREIN in Khmelnitsky National University will allow forming the new generation of engineering and research staff, which are able to perform the constructive development in cybersecurity assessment.

TEMPUS SEREIN project has made it possible to involve experience from the partner of the consortium into Ukrainian educational process. Communication with colleagues, study their experience have enabled us faster, better and more effectively build curricula.

We hope that masters and PhD students in the Khmelnitsky National University will study developed course successfully and become well-trained professionals in the field of cybersecurity.

We greatly appreciate all our partners for the fruitful collaboration, exchange of experience. We look forward to further cooperation.

References

1. P. W. Singer and Allan Friedman, *Cybersecurity and Cyberwar: What Everyone Needs to Know* (Oxford: Oxford University Press, 2014).
2. O.L. Holubenko, O.S. Petrov, and V.O. Horoshko, “Features of Training for Specialists in Information Security,” *Information Security* (2009), 5-17 (in Ukrainian).
3. Y.I. Hrytsyuk and T.E. Rak, “The Training Problem of Information Security Specialists for the Ministry of Emergencies of Ukraine,” in *Intelligent Decision-making Systems and Problems of Computational Intelligence* (Yevpatoria, Kherson, 2011), 272-276 (in Ukrainian).
4. Website of the national Tempus/Erasmus+ office in Ukraine, http://www.tempus.org.ua.
5. Vyacheslav Kharchenko, Chris Phillips, Peter Popov, Oksana Pomorova, Alexander Romanovsky, and Elena Troubitsyna, “MASTAC: New Curriculum for Master and Doctoral Studies in Critical Software and Computing,” in *Proceedings of*

the 2008 International workshop on Software Engineering in East and South Europe SEESE'08, Leipzig, Germany, 13 May 2008, pp. 59-64, https://doi.org/10.1145/1370868.1370879.

6. O.M. Tarasyuk and A.V. Gorbenko, "For Methods for Development of Critical Software," in *Lecture Materials*, ed. V.S. Kharchenko (Kharkiv, National Aerospace University Institute "KhAI," 2009).

7. John McDermid, Martyn Thomas, and Felix Redmill, "Professional Issues in System Safety Engineering," in *Safety-Critical Systems: Problems, Process and Practice*, ed. Chris Dale and Tom Anderson (London: Springer, 2009), 135-145.

8. TEMPUS SEREIN project website, http://serein.net.ua.

9. Website of the System Programming Department. Khmelnytskyi National University, http://spr.khnu.km.ua.

10. Jean François Monin and Michael G. Hinchey, *Understanding Formal Methods* (Springer, 2003).

11. Alexander Romanovsky, "Deployment of Formal Methods in Industry: the Legacy of the FP7 ICT DEPLOY Integrated Project," *ACM SIGSOFT Software Engineering Notes* 37, no. 5 (September 2012): 1-4. https://doi.org/10.1145/2347696.2347710.

12. José Bacelar Almeida , Maria João Frade, Jorge Sousa Pinto, Simão Melo de Sousa, "An Overview of Formal Methods Tools and Techniques," in *Rigorous Software Development: An Introduction to Program Verification* (London, 2011), 15-44. https://doi.org/10.1007/978-0-85729-018-2_2.

13. Michael G. Hinchey, Jonathan P. Bowen, and Emil Vassev, "Formal Methods," in *Encyclopedia of Software Engineering*, ed. Philip A. Laplante (Taylor & Francis, 2010), 308–320.

14. Jean-Claude Laprie, "From Dependability to Resilience," in 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Anchorage, USA, 2008.

15. Robert Neches, *Engineered Resilient Systems (ERS)*, S&T Priority Description and Roadmap; 2011.

16. Leonard Reder, John Day, Mitch Ingham, Richard Murray, and Brian Williams, *Engineering Resilient Space Systems: Introduction to Short Course* (Pasadena, CA: Keck Institute for Space Studies, 2012), http://hdl.handle.net/2014/42740.

17. Enn Tyugu, "Artificial Intelligence in Cyber Defense," in *Proceedings of the Third International Conference on Cyber Conflict*, ed. C. Czosseck, E. Tyugu, T. Wingfield (Tallinn: CCD COE, 2011), 95-105.

18. Christian Bitter, David A. Elizondo, and Tim Watson, "Application of Artificial Neural Networks and Related Techniques to Intrusion Detection," *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 18-23 July 2010, pp. 949– 954. https://doi.org/10.1109/IJCNN.2010.5596532.

19. Jeannette M. Wing, "A Symbiotic Relationship Between Formal Methods and Security," *Computer Security, Dependability and Assurance: From Needs to Solutions, Proceedings*, 7-9 July 1998 (IEEE, 1998), 26-38. https://doi.org/10.1109/CSDA.1998.798355.

20. Riham Hassan Abdel-Moneim Mansour, *Formal Analysis and Design for Engineering Security (FADES)*, PhD Dissertation (Blacksburg, VA: Virginia Polytechnic Institute and State University, 2009).

21. Hiroyuki Nakagawa, Kenji Taguchi, Shinichi Honiden, "Formal Specification Generator for KAOS: Model Transformation Approach to Generate Formal Specifications from KAOS Requirements Models," *Proceedings of the Twenty-second IEEE/ACM International Conference on Automated Software Engineering*, Atlanta, GA: 5-9 November 2007, pp. 531-532. https://doi.org/10.1145/1321631.1321729.

22. Steve Schneider, *The B-Method: An Introduction* (Basingstoke, UK: Palgrave, 2001).

23. A. van Lamsweerde, "Elaborating Security Requirements by Construction of Intentional Anti-Models," in *Proceedings of the 26th International Conference on Software Engineering*, ICSE 2004, 28-28 May 2004. https://doi.org/10.1109/ICSE.2004.1317437.

24. Lirong Dai and Kendra Cooper, "A Survey of Modelling and Analysis Approaches for Architecting Secure Software Systems," *International Journal of Network Security* 5, no. 2 (2007): 187–198.

25. Constance L. Heitmeyer, Myla M. Archer, Elizabeth I. Leonard, and John D. McLean, "Applying Formal Methods to a Certifiably Secure Software System," *IEEE Transactions on Software Engineering* 34, no. 1 (January-February 2008), 82–97. https://doi.org/10.1109/TSE.2007.70772.

26. Constance Heitmeyer, Myla Archer, Ramesh Bharadwaj, and Ralph Jeffords, "Tools for Constructing Requirements Specifications: The SCR Toolset at the Age of Ten," *Computer Systems Science & Engineering* 20, no. 1 (2005): 19-35.

27. Susan Older and Shiu-Kai Chin, "Formal Methods of Assuring Security of Protocols," *The Computer Journal* 45, no. 1 (2002): 46-54. https://doi.org/10.1093/comjnl/45.1.46.

28. Till Mossakowski and Markus Roggenbach, "Structured CSP – A Process Algebra as an Institution," in *Recent Trends in Algebraic Development Techniques*, ed.

José Luiz Fiadeiro and Pierre-Yves Schobbens (Berlin, Heidelberg: Springer, 2007), 92-110. https://doi.org/10.1007/978-3-540-71998-4_6.

29. Pedro Miguel dos Santos Alves Madeira Adão, *Formal Methods for the Analysis of Security Protocols*, PhD Dissertation (Lisbon: Instituto Superior Tècnico, Universidade Tecnica de Lisboa, 2006).

30. Bruno Blanchet, "Using Horn Clauses for Analyzing Security Protocols," in *Formal Models and Techniques for Analyzing Security Protocols*, ed. Véronique Cortier and Steve Kremer (Amsterdam: IOS Press, 2011), 86-111. https://doi.org/10.3233/978-1-60750-714-7-86.

31. Wihem Arsac, Giampaolo Bella, Xavier Chantry, and Luca Compagna, "Validating Security Protocols under the General Attacker," in Foundations and Applications of Security Analysis (Berlin, Heidelberg: Springer, 2009), 33-51. https://doi.org/10.1007/978-3-642-03459-6_3.

32. Alessandro Armando, D.A. Basin, Y. Boichut, et al. "The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications," in International Conference on Computer Aided Verification CAV 2005, ed. Kousha Etessami and Sriram K. Rajamani (Heidelberg: Springer, 2005), 281–285. https://doi.org/10.1007/11513988_27.

33. Alessandro Armando and Luca Compagna, "SAT-based Model-Checking for Security Protocols Analysis," *International Journal of Information Security* 7, no. 1 (2008): 3–32. https://doi.org/10.1007/s10207-007-0041-y.

34. Javier Lopez, Rodrigo Roman, and Cristina Alcaraz, "Analysis of Security Threats, Requirements, Technologies and Standards in Wireless Sensor Networks," in *Foundations of Security Analysis and Design V*, ed. Alessandro Aldini, Gilles Barthe, and Roberto Gorrieri, Lecture Notes in Computer Science, vol. 5705 (Berlin, Heidelberg: Springer, 2009), 289-338. https://doi.org/10.1007/978-3-642-03829-7_10.

35. Eleonora Sibilio, *Formal Methods for Wireless Systems*, PhD Thesis, TD-08-10 (Verona, Italy: Dipartimento di Informatica, Universit`a di Verona, 2011), www.univr.it/documenti/AllegatiOA/allegatooa_5570.pdf.

36. Martin Abadi and Bruno Blanchet, "Analyzing Security Protocols with Secrecy Types and Logic Programs," *Journal of the ACM* 52, no. 1 (2005): 102-146. https://doi.org/10.1145/1044731.1044735.

37. Rodrigo Roman, M.Carmen Fernandez-Zago, Javier Lopez, and Chen Hsiao-Hwa, "Trust and Reputation Systems for Wireless Sensor Networks," in *Security and Privacy in Mobile and Wireless Networking*, ed. S. Gritzalis, T. Karygiannis, and C. Skianis (Leicester, UK: Troubador, 2009), 105-127.

38. Venkat Balakrishnan, Vijay Varadharajan, and Uday Tupakula, "Trust Management in Mobile Ad Hoc Networks," in *Guide to Wireless Ad Hoc Networks*, ed. Sudip Misra, Isaac Zhang, and Subhas Chandra Misra (London: Springer, 2009), 473-502. https://doi.org/10.1007/978-1-84800-328-6_19.

39. Sudarshan Vasudevan, Jim Kurose, and Don Towsley, "Design and Analysis of a Leader Election Algorithm for Mobile Ad Hoc Networks," in *Proceedings of the 12th IEEE International Conference on Network Protocols, ICNP 2004* (IEEE Computer Society, 2004), 350-360. https://doi.org/10.1109/ICNP.2004.1348124.

40. Mariana Hentea, "Intelligent System for Information Security Management: Architecture and Design Issues," *Issues in Informing Science & Information Technology* 4 (2007): 29-40.

41. Tyugu, Artificial Intelligence in Cyber Defense, 96-102.

42. Fatai Adesina Anifowose and Safiriyu Ibiyemi Eludiora, "Application of Artificial Intelligence in Network Intrusion Detection," *World Applied Programming* 2, no. 3 (2012), 158-166.

43. Oksana Pomorova, Oleg Savenko, Sergii Lysenko, and Andrii Kryshchuk, "Multi-Agent Based Approach for Botnet Detection in a Corporate Area Network Using Fuzzy Logic," in Computer Networks Communications in Computer and Information Science, Proceedings of the 20th International Conference on Computer Networks CN 2013, vol. 370, Lwówek Śląski, Poland, 17-21 June 2013, pp. 146-156.

44. Lysenko S., Savenko O., A. Kryshchuk, Y. Klyots "Botnet detection technique for corporate area network," Proceedings of the 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (Berlin, DE, IEEE, 2013), 315-320.

45. Inna Pereverzeva, *Formal Development of Resilient Distributed Systems*, PhD Dissertation (Turku, Finland: Abo Akademi University, Faculty of Science and Engineering, 2015), http://urn.fi/URN:ISBN:978-952-12-3253-4.

46. Jean-Claude Laprie, "Resilience for the Scalability of Dependability," in *Proceedings of the Fourth IEEE International Symposium on Network Computing and Applications*, NCA'05, 27-29 July 2005, pp. 5-6. https://doi.org/10.1109/NCA.2005.44.

47. Laprie, "Resilience for the Scalability of Dependability."

48. Jean-Raymond Abrial, *The B-Book: Assigning Programs to Meanings* (Cambridge, UK: Cambridge University Press, 1996). https://doi.org/10.1017/CBO9780511624162.

49. Jean-Raymond Abrial, *Modeling in Event-B: System and Software Engineering* (Cambridge, UK: Cambridge University Press, 2010).

50. Alexei Iliasov, Elena Troubitsyna, Linas Laibinis, Alexander Romanovsky, Kimmo Varpaaniemi, Dubravka Ilic, and Timo Latvala, "Developing Mode-Rich Satellite Software by Refinement in Event-B," in *Formal Methods for Industrial Critical Systems*, FMICS 2010, ed. Stefan Kowalewski and Marco Roveri (Springer, 2010), 50-66. https://doi.org/10.1007/978-3-642-15898-8_4.

51. Anton Tarasyuk, Elena Troubitsyna, and Linas Laibinis, "Formal Modelling and Verification of Service-Oriented Systems in Probabilistic Event-B," in *Proceedings of the International Conference on Integrated Formal Methods*, IFM 2012, ed. John Derrick, Stefania Gnesi, Diego Latella, and Helen Treharne (Springer, 2012), 237-252. https://doi.org/10.1007/978-3-642-30729-4_17.

52. Yuliya Prokhorova, Linas Laibinis, Elena Troubitsyna, Kimmo Varpaaniemi, and Timo Latvala, "Derivation and Formal Verification of a Mode Logic for Layered Control Systems," in *Proceedings of the 18th Asia-Pacific Software Engineering Conference*, APSEC 2011, ed. Tran Dan Thu and Karl Leung (IEEE, 2011), pp. 49-56. https://doi.org/10.1109/APSEC.2011.38.

53. Inna Pereverzeva, Elena Troubitsyna, and Linas Laibinis, "Formal Goal-Oriented Development of Resilient MAS in Event-B," in *Proceedings of 17th International Conference on Reliable Software Technologies*, ed. Mats Brorsson and Luis Miguel Pinho (Berlin, Heidelberg: Springer-Verlag, 2012), 147-161. https://doi.org/10.1007/978-3-642-30598-6_11.

54. Paris Avgeriou, ed., *Proceedings of 4th International Workshop on Software Engineering for Resilient Systems*, SERENE 2012 (Berlin, Heidelberg: Springer-Verlag, 2012), 16-31. https://doi.org/10.1007/978-3-642-33176-3.

55. *Simulation Framework in Python*. Available at http://simpy.readthedocs.org/.

About the Authors

Oksana POMOROVA is Head of the System Programming Department of Khmelnitsky National University, Khmelnitsky, Ukraine. She graduated from the Taras Shevchenko National University of Kyiv in 1992 and received a PhD degree in Automated control systems and advanced information technologies from the Kyiv Institute of Automatics, Kyiv, in 2001. In 2007, she defended doctorate thesis in Computer systems and networks in National University "Lviv Polytechnic," Lviv. She has authored and co-authored eight books and monographs, and more than 100 papers published in Ukrainian and foreign journals. Her current research focuses on developing of methodologies and information technologies for computer systems diagnosis. She currently leads two research projects. *E-mail*: o.pomorova@gmail.com

Associate Professor Sergii LYSENKO graduated from the Khmelnitsky National University (Ukraine) in 2005, and received a PhD degree in Information technologies in Ternopil National Economic University in 2011. He focuses on the development of information technologies for malware detection. He has more than 60 publications in Ukrainian and international journals. E-mail: sirogyk@ukr.net.