AQoPA: Automated Quality of Protection Analysis Framework for Complex Systems

Damian Rusinek, Bogdan Ksiezopolski, Adam Wierzbicki

AQoPA: Automated Quality of Protection Analysis framework for complex systems

Damian Rusinek[1], Bogdan Ksiezopolski[1,2], Adam Wierzbicki[2]

[1]Institute of Computer Science, Maria Curie-Sklodowska University, Lublin, Poland
[2]Polish-Japanese Academy of Information Technology, Warsaw, Poland

Abstract. Analysis of security economics for the IT systems is one of the important issues to be solved. The quality of protection (QoP) of IT System can be achieved on different levels. One can choose factors which have a different impact on the overall system security. Traditionally, security engineers configure IT systems with the strongest possible security mechanisms. Unfortunately, the strongest protection (especially in low resource devices) can lead to unreasoned increase of the system load and finally influence system availability. In such a situation the quality of protection models which scales the protection level depending on the specific requirements can be used. One of the most challenging issues for quality of protection models is performing quality of protection evaluation for complex and distributed systems. The manual analysis of such systems is almost impossible to perform. In the article, we proposed the Automated Quality of Protection Analysis framework (AQoPA). The AQoPA performs the automatic evaluation of complex system models which are created in the Quality of Protection Modelling Language (QoP-ML). In the article the case study of complex wireless sensor network analysis is presented. The network is deployed on a roller-coaster.

Keywords: modelling and protocol design, security protocol analysis, quality of protection, applied cryptography

1 Introduction

The security analysis of the IT systems and simultaneously that of its influence on the system performance is one of the most important topics to be solved. On one hand, the traditional approach assumes that implementation of the strongest security mechanisms makes the system as secure as possible. Unfortunately, on the other hand, such reasoning can lead to the overestimation of security measures which causes an unreasonable increase in the system load [1–4]. The system performance is especially important in the systems with limited resources such as the wireless system networks or the mobile devices. Another example where such analysis should be performed is cloud architecture. The latest research indicates three main barriers for using cloud computing which are security, performance and availability [5]. Unfortunately, when the strongest security mechanisms are used, then it will decrease system performance and further influence system

availability. This tendency is particularly noticeable in complex and distributed systems. The latest results show [6, 7] that in many cases the better way is to determine the required level of protection and adjust security measures to these security requirements [8]. Such approach is achieved by means of the Quality of Protection models where the security measures are evaluated according to their influence on the system security.

One of the most challenging issues for the QoP models is performing quality of protection evaluation for complex and distributed systems [12]. The manual analysis of such systems is almost impossible to perform. The analysis of any type of the security protocol is difficult when the experts do not use automated tools. In literature, we can indicate programs which helped the experts analyse the protocols. We can indicate the AVISPA tool [13, 14] or ProVerif [15] application, which verifies security properties for cryptographic protocols. From the Quality of Protection analysis point of view, AVISPA and ProVerif have two limitations. The first one refers to the types of the function which can be modelled. One can model only cryptographic primitives and cryptographic algorithms. The full QoP analysis must refer to all security factors which affect the overall system security. The second limitation is that these languages do not provide the structure for evaluation of the security factors performance. In the literature one can indicate the tool for QoP analysis which are modelled in the UMLsec [16]. This tool can be used for automated analysis of simple models but when we would like to analyse the scenarios when thousands of hosts take part in the protocol, then the analysis is too complex and cannot be done properly. The UMLsec is used for model-driven security as the part of model-driven development.

The major contribution of this study is introduction of Automated Quality of Protection Analysis framework which performs the automatic evaluation of QoP-ML models created in the Quality of Protection Modelling Language [9, 10]. It allows to analyse complex systems which may consist of thousands of hosts representing a wide area network which are actors in the cryptographic protocol or a complex IT system. One can balance security against performance. The full description of the AQoPA will be presented in the next sections. The AQoPA framework can be downloaded from the web page of the Quality of Protection Modelling Language Project [11].

2 Automated Quality of Protection Analysis framework (AQoPA)

In this section we present the architecture of AQoPA and data flow of model during the analysis process. The architecture of AQoPA is presented in Fig. 1. The figure presents four successive stages: stage 1 - model creation, stage 2 - security metrics definition, stage 3 - scenarios definition and stage 4 - simulation. These stages refer to the methodology of creating QoP-ML models defined in the article [9] where the details about syntax and semantics can be found.

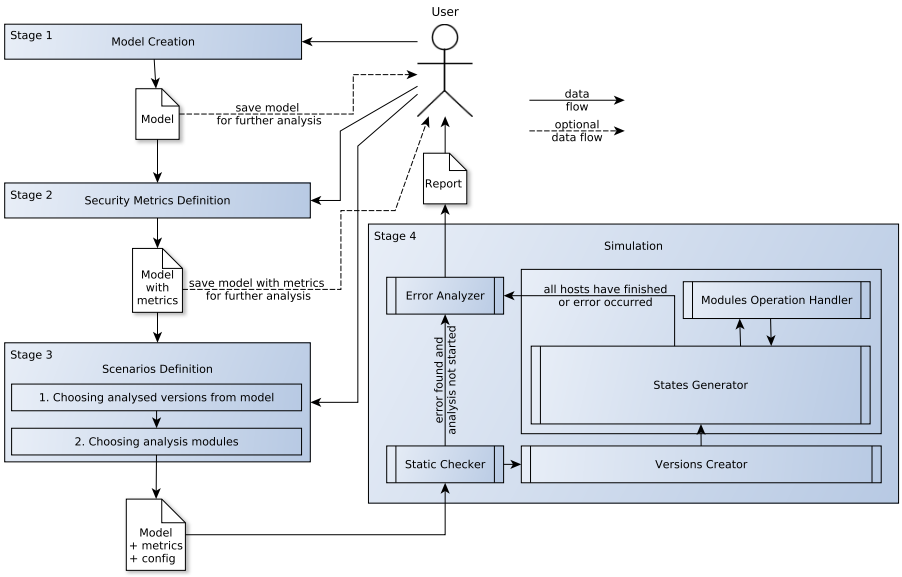


Fig. 1. AQoPA architecture and data flow

2.1 Stage 1 - Model Creation

Model creation stage is the first stage which must be performed in the AQoPA framework. The goal of this stage is to create the QoP-ML model that will be evaluated in the analysis process. The stage is divided into 4 phases. Initially the designer has to define functions (phase 1), functions equations (phase 2) and channels (phase 3). Later he can use them to create protocol flow (phase 4).

Protocol flow is defined for hosts as they are the highest level elements of the analysis. Hosts contain processes and processes can contain subprocesses. Each process and subprocess contain instructions list.

2.2 Stage 2 - Security Metrics Definition

The second stage is the security metrics definition stage which is divided into 4 phases, too. In the first phase the designer has to gather metrics and configurations of analysed devices (servers, sensors, etc.). In the second phase the designer has to select subset of metrics for the functions that are used in the protocol flow created in the first stage.

The aim of last two phases is to group selected metrics in sets and assign them to hosts. The designer can model different devices therefore the metrics for the same functions may have different values. For example, the encryption operations are many times faster on high-performance servers than on the wireless sensors

nodes. The designer has to group metrics into sets representing one device and assign these sets to hosts created in the first stage.

2.3 Stage 3 - Scenarios Definition

The aim of the third stage is to define the scenarios of the analysis process. The protocol flow is already created and the designer has to define versions that he would like to evaluate.

Versions selection Versions represent different variants of the evaluated protocol. Versions include a list of executed hosts, their processes and subprocesses that will be included in independent evaluations. The designer does not have to choose all processes from the host, but any subset that will create the target protocol flow. The differences in versions may come from using different devices or security mechanisms resulting in different metrics or from different protocol flows (i.e.. including additional processes or subprocesses in the version that implement additional security mechanism). Versions allow to evaluate complex models with a large number of hosts using repetition. The designer can repeat hosts and processes. At the end of the analysis, the designer obtains results for all evaluations and can compare them.

Modules selection Besides versions, the designer must select modules that he would like to use in the analysis process. The AQoPA is module based what means that the designer can easily add modules to the analysis process. The core of AQoPA is responsible for generating next states according to the protocol flow. The additional operations that bring results of analysis are executed in modules.

2.4 Stage 4 - Simulation

Simulation stage is the core stage of AQoPA architecture. The analysis process is realized by this stage and proceeds automatically without the user's interaction.

Static Checker Firstly, the provided model is passed to Static Checker which is responsible for syntax validation of the model. Any syntax error is passed to the Error Analyzer.

Versions creator When the model is validated by Static Checker, it is passed to the Versions creator. The task of this component is to create independent analysis process for each version selected in the Scenarios Definition stage. Creating versions involves the modification of protocol flow according to the list of executed hosts, processes and subprocesses in a particular version. As a result, AQoPA obtains as many protocol flows as many versions were selected. Each modified protocol flow is passed to the States Generator component and is analysed independently.

States Generator In the States Generator component AQoPA generates successive states of evaluated hosts executing their instructions. This process is repeated until all hosts are finished or an error occurs. Each process of next state generation includes execution of the instruction that was modelled in the QoP-ML.

This component is also responsible for detection of QoP-ML model errors. These errors may result from the designer's mistake during modelling the system (or protocol) or from the limitations due to the metrics (i.e.. using sensors - devices with limited resources). These errors may lead to the situation when a variable is used before assignment, to deadlock in communication or to ambiguity of equations. This component detects these kinds of problems and passes them to Error Analyzer which outputs the information to the user.

QoP-ML introduces equations that are used to reduce complex function calls (nested function calls in parameters). Syntax validation checks all equations according to the syntax rules, finds and reports contradictory equations and checks if one equation contains the other one. The States Generator component finds ambiguity during the reduction process. All above situations pass an error to Error Analyzer.

Modules operation handler Modules selected in the Scenarios Definition stage take part in the process of generating the next state of protocol. Execution of all instructions is passed to the modules so that they can retrieve information about the current state and prepare the results. Additionally, modules can change default instructions flow.

Error Analyzer Error Analyzer is the last component of simulation stage. It outputs the information about the error received from Static Checker or States Generator. When the analysis finishes successfully, Error Analyzer creates a result report.

3 Case Study

In this section we present the analysis of complex system performed by means of the AQoPA framework. This case study presents the features of AQoPA as the tool used to analyse the wireless sensor networks in terms of security and efficiency.

We analyse the wireless sensor network used for structural health monitoring deployed on roller-coaster. Its main aim is to collect acoustic emission data during the ride which is transmitted to the gateway and analysed for cracks. This method has been studied in [18] and is still widely used as inspection method [19]. Using wireless sensor network gives a possibility to react to detected failures in real time.

In the case study we have used the Rougarou roller-coaster from Cedar Point theme park [17]. It is interesting example because the roller-coaster was opened

in 1996 and named Mantis while now it is being rebuilt to be a floor-less coaster. Therefore the main structure remains and is going to be operated by new coaster.

The Rougarou is about 1200 meters length, 44 meters high and the speed goes up to 97 kmph. In Fig. 1ffig:rollercoaster we present an example of sensor placement on 270 degrees turn. Sensing nodes are placed every 3 meters on the track while the sink is mounted to the car and collects data from sensors during ride. During the analysis we would like to evaluate how the quality of protection of exchanging data influence the system efficiency and finally the lifetime of system.

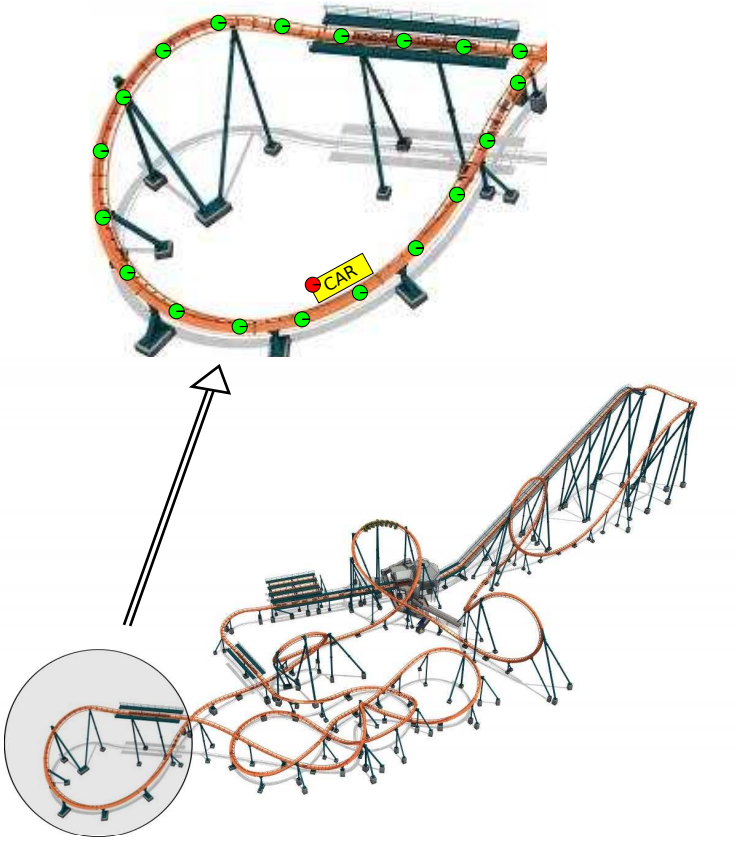


Fig. 2. WSN deployed on Rougarou roller-coaster.

3.1 Protocols

We propose three protocols to collect acoustic emission data presented in Fig. 3, 4 and 5. When data is collected the algorithms for fatigue cracks detection are used

and report any suspicious results in real time. Protocols present communication between sink, which collects data, and sensor, which senses data. The wireless sensor network consist of many sensors mounted on the track and one sink mounted to the car. The technique of sink mobility is used to collect data from sensors [20].

The first protocol is the simple one where no security is guaranteed. The second and the third protocols are modified version of protocol presented in verified in [23] which is used to authenticate new nodes in a sensor network.

In the description of protocols we use the following notation:

S	- Sensor,
$Sink$	- Sink,
R	- empty request for SHM data,
D	- acoustic emission data sampled by sensors,
n_I	- nonce generated by I,
$sk(I)$	- private key of I,
$pk(I)$	- public key of I,
$\{M\}_K$	- encryption of M with symmetric or asymmetric key K.

Protocol 1

In the first protocol we analyse the network without security mechanisms. The protocol is presented in Fig. 3.

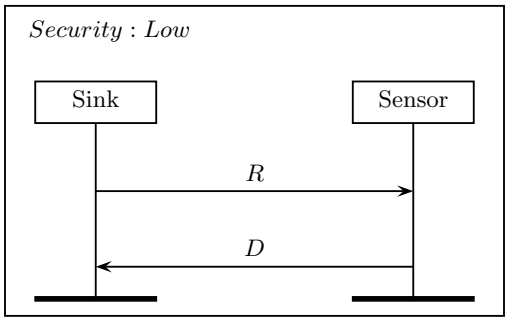


Fig. 3. Protocol 1: No security.

The sink node sends empty request for acoustic emission data to sensor when the car is near its place of mounting. In response the sensor collects acoustic emission data and returns it back unencrypted.

Protocol 2

The second protocol includes security mechanisms which provide confidentiality and authentication of sink and sensors. The communication is encrypted with

pre-deployed, symmetric key NK. The presented is shown in Fig. 4.

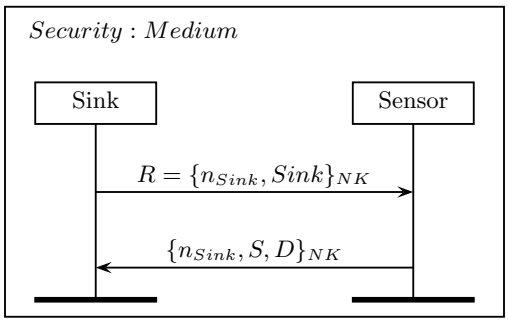


Fig. 4. Protocol 2: Symmetric cryptography

Protocol 3
The difference between second and third protocol is that the third one uses asymmetric cryptography to ensure security. We assume that the pairs of keys were pre-deployed and sink knows public keys of all sensors while sensors know the public key of sink. The protocol is presented in Fig. 5. The difference be-

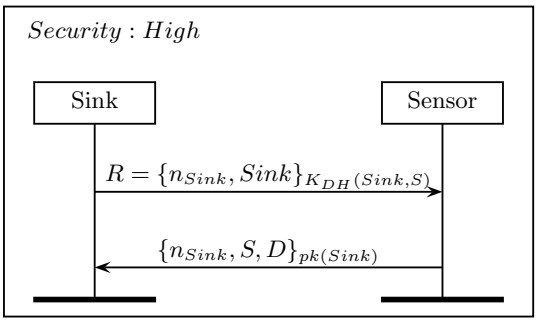


Fig. 5. Protocol 3: Asymmetric cryptography

tween original version from [23] and the third protocol is returned data. In the original version, the session key is returned while in the proposed protocol it is acoustic emission data. Additionally, in the second protocol data is encrypted with network key which has two advantages: encryption takes less time and protocol does not require asymmetric keys management. However, using network key gives possibility of reading and modifying packets to all nodes in network. Therefore, the assumption of all trusted nodes is required.

3.2 AQoPA evaluation

As the result of the case study we want to estimate the lifetime of wireless sensor network deployed on the roller-coaster depending on the protocol and type of sensor node. Firstly, we have to define the assumptions about the monitoring system.

Assumptions:

- **Number of measurements.** We assume that there are 100 rides with measurement each day.
- **Number of sensors.** We assume that the network consists of one sink mounted to the car and 400 sensors mounted to the track at equal intervals.
- **Battery.** We assume that nodes are equipped with standard AA battery with 1200 mAh capacity.

After this, we defined two type of factors which influence the lifetime of the wireless sensor network monitoring system.

Factors:

1. **Protocol**. We want to check the lifetime depending on the type of protocol selected from three previously described.
2. **Hardware**. We estimate the lifetime for two types of nodes: MicaZ with 8MHz ATmega128L CPU and Imote2 with 104MHZ PXA271 XScale CPU.

In case of page limit we do not present the QoP-ML model of the roller-coaster but the QoP-ML model of the case study can be found on the QoP-ML Project web page [11] and in the AQoPA framework.

Presented model consists of hundreds of sensors and the analysis of such complex network is impossible to be performed manually. Using AQoPA the network can be analysed automatically in short time. The lifetimes of nodes (in days) estimated with AQoPA are presented in Table 1.

Table 1. Lifetime (in days) of nodes in SHM network.

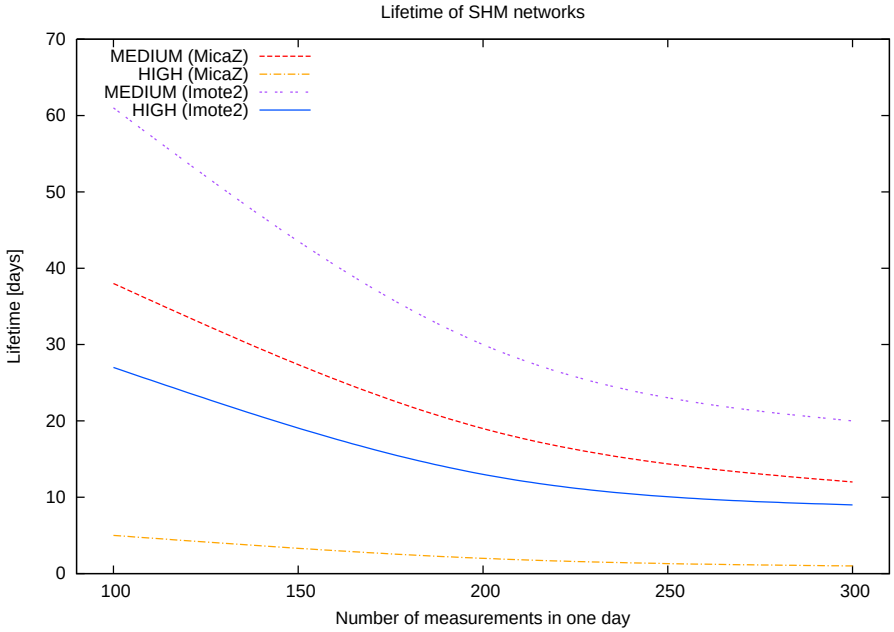
	Lifetime (in days)			
	MicaZ		Imote2	
	Sink	Sensor	Sink	Sensor
Protocol 1 - NO security	87378	5378	86677	653
Protocol 2 - MEDIUM security	38	3974	61	636
Protocol 3 - HIGH security	5	1089	27	597

One can see that lifetime of sink node for protocol 1 (*NO security*) is very long. When we compare it with other protocols one can notice significant influence of security operations on lifetime of the network. When the security mechanisms are introduced (protocol 2 and 3) the sink becomes the bottle neck

of the network. One can see that in case of MicaZ and *HIGH* security protocol the lifetime of network decreases to 5 days while for Imote2 it becomes 27 days.

One of the important factor which is worth for estimating is the wireless sensor network lifetime. We define the network lifetime as the minimum of nodes' lifetimes because we assume that each node must be operative in order to keep network working correctly. In case of *HIGH* security protocol using MicaZ sensors, lifetime of network could be extended five times only if the sink was replaced with Imote2 sensor (both nodes have the same radio chip CC2420).

Fig. 6. The lifetime of networks depending on the number of daily measurements.



Another element which can be analysis by means of AQoPA framework is the influence of measurements granularity to the network lifetime. The lifetime results in Tables 1 were estimated with assumption that there are 100 rides with measurements during one day. Figure 6 presents the decrease of lifetime depending on daily number of measurements for the protocols which guarantees the security on *MEDIUM* and *HIGH* security level. The protocol 1 (*NO security*) has not been included on Fig.6 because in this protocol the performance is obviously much more effective than other protocols.

4 Conclusions

In the article we proposed the Automatic Quality of Protection Analysis framework. One of the main contributions of AQoPA is quality of protection and efficiency analysis of complex systems which are modelled in QoP-ML (Quality of Protection Modelling Language). The complexity of models may result by e.g. a large number of hosts (actors) in the protocol flow which can be executed simultaneously or by a specifically defined order. For illustrating the capabilities of AQoPA we present an example of SHM wireless sensor network deployed on roller-coaster in order to find cracks in the track. The AQoPA has been used to estimate the lifetime of such network under different conditions depending on the level of security and hardware. The AQoPA can verify whether a proposed network which uses given protocol can be operative for questioned length of time under given, environmental circumstances. The presented in the article analysis of the wireless sensor network deployed on roller-coaster show that one can gather the results which can help designers for making the best decision about complex system parameters.

Modelling tools used for computer security applications may however be found to prospective co-existence and cooperation with modelling tools used in computational neuroscience [21, 22]. We have undertaken some steps in order to merge security aspects of computer systems with hypothesised existence of brain activity fingerprint characteristic for particular users. These research will be continue as the future work. That kind of analysis of complex system cannot be prepared manually.

5 Acknowledgements

This work is supported by Polish National Science Centre grant 2012/05/B/ST6/03364.

References

1. Stubblefield A., Rubin A.D., Wallach Dan. S.: "Managing the Performance Impact of Web Security"; Electronic Commerce Research, 5, 2005, 99116.
2. Sklavos N., Kitsos P., Papadopoulos K., Koufopavlou, O.: "Design, Architecture and Performance Evaluation of the Wireless Transport Layer Security"; The Journal of Supercomputing, 36, 1, 2006, 33-50.
3. Ksiezopolski B., Kotulski Z., Szalachowski P.: "Adaptive approach to network security"; Communications in Computer and Information Science, 158, 2009, 233-241.
4. Ksiezopolski B., Kotulski Z., Szalachowski P.: "On QoP method for ensuring availability of the goal of cryptographic protocols in the real-time systems"; European Teletraffic Seminar, 2011, 195-202.
5. Jürjens J.: "Security and Compliance in Clouds"; IT-Compliance 2011, Berlin, 4th Pan-European Conference, 2011.
6. Ksiezopolski B., Rusinek D., Wierzbicki A.: "On the modelling of Kerberos protocol in the Quality of Protection Modelling Language (QoP-ML)"; Annales UMCS Informatica AI XII, 4, 2012, 69-81.

7. Ksiezopolski B., Rusinek D., Wierzbicki A.: "On the efficiency modelling of cryptographic protocols by means of the Quality of Protection Modelling Language (QoP-ML)"; Springer: Lecture Notes in Computer Science, 7804, 2013, 261-270.
8. Lambrinoudakis C., Gritzalis S., Dridi F., Pernul G.: "Security requirements for e-government services: a methodological approach for developing a common PKI-based security policy"; Computers & Security, 2003, 26, 1873-1883.
9. Ksiezopolski B.: "QoP-ML: Quality of Protection modelling language for cryptographic protocols"; Computers & Security, 31(4), 2012, 569-596.
10. Ksiezopolski B.: "Multilevel Modeling of Secure Systems in QoP-ML"; CRC Press, pp. 1-256, 2015.
11. The official web page of the QoP-ML project: http://www.qopml.org
12. Ksiezopolski B., Zurek T., and Mokkas M.: "Quality of Protection Evaluation of Security Mechanisms"; The Scientific World Journal, vol. 2014, 18 pages, 2014.
13. Blanco V., Gonzalez P., Cabaleiro J.C.,Heras D.B., Pena T.F., Pombo J.J., Rivera F.F.: "AVISPA: visualizing the performance prediction of parallel iterative solvers"; Future Generation Computer Systems 19 (2003) 721-733.
14. Vigano L.: "Automated Security Protocol Analysis With the AVISPA Tool"; Electronic Notes in Theoretical Computer Science 115 (2006) 61-86
15. Blanchet B., Chaudhuri A.: "Automated Formal Analysis of a Protocol for Secure File Sharing on Untrusted Storage"; Proceedings of the 29th IEEE Symposium on Security and Privacy, 2008, 417-431.
16. Jürjens J.: "Tools for Secure Systems Development with UML"; International Journal on Software Tools for Technology Transfer, 9, 2007, 527-544.
17. Cedar Point web page: https://www.cedarpoint.com/
18. Yuyama S.: "Fundamental Aspects of Acoustic Emission Applications to the Problems Caused by Corrosion"; Corrosion Monitoring in Industrial Plants Using Nondestructive Testing and Electrochemical Methods, American Society for Testing and Materials, 1986, 43-74.
19. ElBatanouny M., Mangual J., Ziehl P., Matta F.: "Early Corrosion Detection in Prestressed Concrete Girders Using Acoustic Emission"; J. Mater. Civ. Eng., 26(3), 2014, 504511.
20. Rault T., Bouabdallah A., Challal Y.: "Energy efficiency in wireless sensor networks: A top-down survey"; Computer Networks, vol. 67, 2014, 104-122.
21. G. M. Wojcik, Electrical parameters influence on the dynamics of the hodgkin-huxley liquid state machine, Neurocomputing, vol. 79, pp. 6878, 2012.
22. G. M. Wojcik and J.A.Garcia-Lazaro,Analysis of the neural hypercolumnin parallel pcsim simulations, Procedia Computer Science, vol. 1, no. 1, pp. 845 854, 2010.
23. Mansour I., Rusinek D., Chalhoub G., Lafourcade P., Ksiezopolski B.: "Multihop Node Authentication Mechanisms for Wireless Sensor Networks"; Ad-hoc, Mobile, and Wireless Networks, vol. 8487, Springer International Publishing, 2014, 402418.