Indian Journal of Science and Technology, *Vol 10(20), DOI: 10.17485/ijst/2017/v10i20/113927, May 2017*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Formal Model for Verification of ZigBee Protocol for Secure Network Authentication

Rana Muhammad Nadeem[1] and Abdul Aziz Gill[2]

[1]Department of Computer Sciences, Government Postgraduate College Burewala, Pakistan; rananadim@hotmail.com
[2]Department of Computer Sciences, Superior University, Lahore, Pakistan; aziz_gill@hotmail.com

Abstract

Background/Objectives: In the modern age, there is a need of wireless network protocols verification using suitable techniques and tools to meet the security challenges in Wireless Ad-Hoc sensor networks. **Methods/Statistical Analysis**: For verification of system like ZigBee protocol stack, formal methods are being used. The latest formal verification method called Event-B is used now a day to frame a model for verification of different wireless security protocols like IEEE 802.11 and IEEE 802.15.4. **Findings:** To describe specific properties in a suitably rich mathematical logic such as first order logic, we need to limit this expressiveness if we are to automatically verify a property. To verify any system properties, temporal logic are used for safety, correctness, reliability in wireless security protocols. In this paper we developed a model/framework in Event-B for the formal verification of ZigBee protocol and simulate it using RODIN tool. In this framework, models are specified, analyzed and verified by using formal methods. **Application/Improvements:** This framework leads toward more secure and reliable model having no inconsistencies in ZigBee.

Keywords: Drawn from Title, 5-6 Words, Word Representing the Work, Formal Model, Network Authentication, Secure Communication, Verification of ZigBee, Wireless Protocol

1. Introduction

Information technologies have passed through a rapid sequence of phases since the emergence of computers. As the era of personal computers introduced, this was propagated with the formation of computer networks[1]. Wireless networking refers to a broad topic that in essence associated with communication networks that use radio waves in the form of electromagnetic waves as a carrier and thus provides greater flexibility and convenience compared to wired networks. Major concern in the present age is to develop new methodologies based on security model developing by using formal methods. Special considerations are given on Wireless Sensor Networks (WSNs) because WSNs require extra security measures and novel characteristics[2].

Ensuring security properties in a networked system is mainly achieved by small distributed programs called security protocols. Even though security protocols rely on cryptographic primitives, they are tremendously error-prone. However, identifying the errors is not trivial even for moderate size protocols. For these reasons, formal methods are used with temporal logics to find out whether the system satisfies the requirements and to verify security protocols[3].

ZigBee standard is a new and capable WPAN protocol used for wireless Ad-Hoc sensor networks that consist of devices with very low resource requirements. ZigBee can be used in many WSN appliances such as digital electrical meters, wireless routers, control room in industries, intrusion detection purposes, medical equipment automatic sensing, home appliances and many others[4]. In reality, ZigBee offers applications framework but at the present to consider ZigBee as a technology simply for home appliances is not a fair justification.

The IEEE 802.15.4 has the capability to ensure reliable communication among available devices and its neigh-

*Author for correspondence

boring nodes, tackling critical issues such as collision avoidance and improving efficiencies in the communication. The interface is provided by this protocol to the physical communication medium, and handles assembly and decomposition of data packets.

The frequency ranges of PHY layer is 868MHz and 915MHz known as lower range and 2.4GHz know as upper range frequency. Low range is observed in Europe (868MHz) and in USA and Australia (915MHz). On the other hand upper range is observed in rest of the world (2.4GHz).

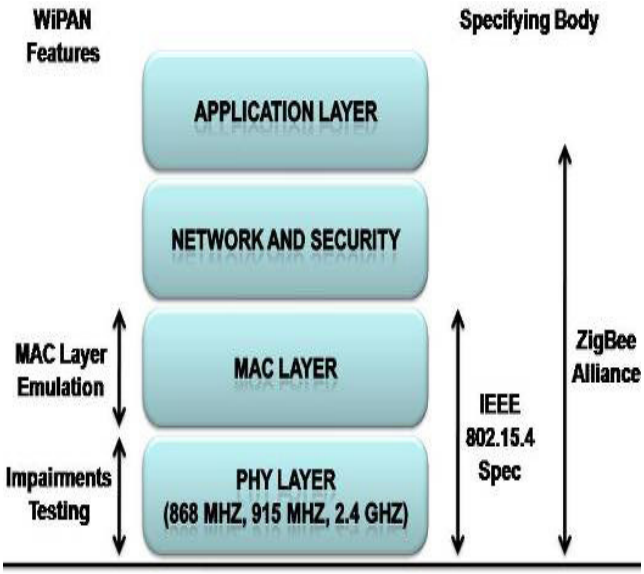

Figure 1. IEEE 802.15.4 and ZigBee Architecture[5].

For verification of system like ZigBee protocol stack in Figure 1, formal methods are used efficiently[5]. The latest formal verification method called Event-B is used now a day to frame a model for verification of different wireless security protocols like IEEE 802.11 and IEEE 802.15.4. In this paper, we developed a framework for the formal verification of ZigBee protocol. The system has been developed by using stepwise approach of Event-B. The RODIN tools have been used for specifications, analysis and verification of formal models. Our model for verification of ZigBee is verified through Event-B using RODIN tool.

2. Background

In networking area, system verification is in many cases seen as an activity that involves testing and simulation. However, these two methods cannot provide the necessary guarantees for systems, such as being flawless or functioning correctly.

In[6,7] are two automated tools that verify security protocols in formal models, and computational models, respectively. ProVerif can verify confidentiality and authentication of protocols properties. SATMC[8] is a SAT based model checker that verifies security protocols. The idea here is building a propositional formula from a description of a protocol in a multi-set rewriting formalism, thus reducing the protocol verification problem to a satisfy ability problem of a propositional formula.

Semantic analysis[9], is used for the successful verification of protocols. Similarly, different formal theory is also used for the analysis and to prove by one or more theorem[10]. Tools based on state exploration methods systematically explore the states of the model of the protocol of interest a finite state model and look for violations of certain security properties. FDR/Casper[11] and STA[12] make a not necessarily exhaustive list of examples for such tools.

One of the earliest protocol analysis approaches was developed a formal model that allowed multiple executions of a protocol running concurrently and including an attacker that could read, modify, and destroy the messages[13]. Different approach which was based on belief logic received attention of the community[14]. It consisted of modal operators describing relations between principals and data, possible beliefs such as owner of a key or sender of a message, and a set of inference rules.

Continuing with quantitative temporal information example, an early approach defined a temporal logic to reason about security protocols[15]. Later, use of time automata was suggested for modeling timeout and retransmission. Thus model checkers could be extended to support time-sensitive protocols such as in[16]. Quantitative analysis in a security context has also been used in quantifying security threats such as denial of service[17]. Probabilistic verification tools are heavily used in similar analyses.

3. Material and Methodology

In order to make accurate and reliable verification, there is need of protocols properties must be illustrated in an unambiguous and precise manner. Although we can describe specific properties in a suitably rich mathematical logic such as first order logic, we need to limit this

expressiveness if we are to automatically verify a property. To verify any system properties, temporal logic can be used for safety, correctness, reliability etc.

At present Event-B is a modern language for formal modeling and specification Event-B is an un-typed language[18]. Therefore, most event specifications have guards which define the type of input variables. Guards determine what set (type) an input variable is in. Guards can also define other constrains and properties that an event must obey in order to carry out actions defined in it. Preconditions and functions those will use in Event B for proof obligations. In Event B proof obligations are used for properties correctness and consistence. Different function and operation of the protocols are defined by events. The entire events are initialized to make it dynamic. Invariants are modeled in Event B as events. In ZigBee all the devices and nodes can be verified on the basis of these events. The ZigBee protocol verification layout is shown in Figure 2.

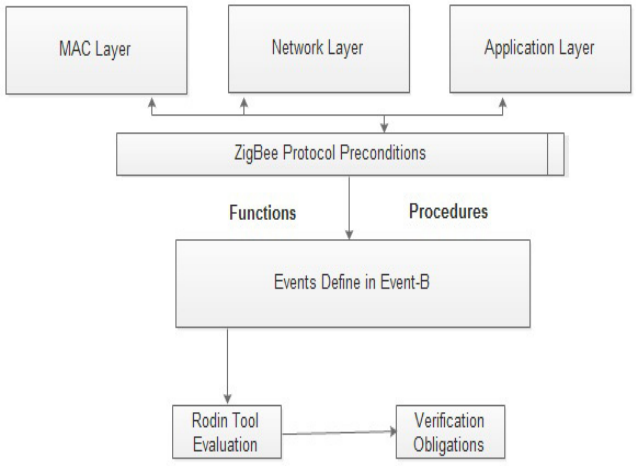

Figure 2. Event-B and ZigBee Protocols verification Layout.

3.1 Modeling Security Properties of ZigBee

Security properties of protocols defined in formal specifications have been developed in Event-B by using Camile Editor. This formal specification has been analyzed by using RODIN tools[19]. These formal specifications represent abstract formal models and more refinement is required in order to integrate these models into real world system. These abstract formal specifications of security properties consist of a static model and a dynamic model. The static aspects of the security properties are defined in the static model and dynamic properties are defined in the dynamic model. This formal specification contains details related to abstract models of security properties.

3.1.1 Static Model

The static model is defined in Neighboring-Devices-Context. This context defines three carrier sets. The carrier set NWK denotes the set of Network layer data entity primitives' types. The carrier set STATUS denotes the set of request status message used in configuration for all possible entities. The carrier set ZB_OBJ represents the set of all possible ZigBee objects (devices) in a network. The carrier set KEY represents the security key information for authentication that helpful in secure connectivity of the ZB_OBJ. Axioms in Event-B are used to define constraints on the carrier sets and constants defined in the contexts. There are three axioms defined on the carrier sets in the context Data. These axioms ensure that the carrier sets: NWK, KEY, USER, REQ_STATUS, CREDENTIALS and ZB_OBJ are all finite sets because Event-B handles only finite sets.

Carrier Sets: NWK, KEY, USER REQ_STATUS, Credentials, ZB_OBJ

axioms:
axm1: finite (NWK)
axm2: finite (REQ_STATUS)
axm3: finite (ZB_OBJ)
axm4: finite (KEY)
axm5: finite (USER)
axm6: finite (CREDENTIALS)

3.1.2 Dynamic Model

In dynamic model, behavior and system states are expressed in terms of variables, invariants and events. This machine defines variables for device in the network, resources the network may contain, allowable actions on the network resources, routers in the network, coordinators of the network, key updates, nodes authorization in the network and owner of a resource.

Initialization event is used for assigning initial values to the variables and ensures that at least one protocol state exists. This is very help full for verification of properties of the network protocols stack. In this event, all the variables are initialized to empty sets because initially no event is executed and no value is added into these variables. The initial value of these variables is necessary for verification. It ensures that at least one protocol state exists. Invariants define constraints on the network protocol and model the properties that must remain true all the time during the operation of the protocol. These invariants define the type

of variables and determine the type of values which these variables may hold in the protocol.

Variables:
NWCAP, NETWORK, NWLYRcmd, MACLYRcmd, NWstatus, Parent

Event: Initialisation begin
init1: NWCAP :=∅init2: netwrok :=∅
init3: NWLYRcmd :=∅
init4: MACLYRcmd :=∅
init5: NWstatus :=∅init6: Parent:=∅

Invariants:
inv1: NETWORK ∈ P(network)
inv2: NWCAP ∈ P(ZB_OBJ) → BOOL
inv3: NWLYRcmd ∈ NWLYR ∧ MACLYRcmd
inv4: MACLYRcmd ∈ MACLYR ∧ NWCAP
inv5: NWstatus ∈ NWLYR STATUS ∧ (Parent)
inv6: Parent ∈ (COOR ∪ SW_ROUTR = nodes)

3.1.3 Formal Specification for Adding Authenticated User

For adding a new user into the set of authenticated users in a ZigBee network all the credentials are sent as input in this event. Types of these input variables are defined for the user registration for the safety purposes. In addition to these type safety measures, other safety points ensure that this user is a registered user and must not already be added into the set of authenticated users.

Event: Adding_Authenticated_User
any user credls
where reg1: u ∈ users
regd2: credls ∈ P(CREDENTIAL)
reg3: u ∈ dom(registered_users)
reg4: u ∉ dom(authenticated_users)
reg5: credls ∉ ran(authenticated_users)
then
act1:authenticated_users ≔ authenticated_users ∪ {user ↦ credls}end

3.2 ZigBee Protocol Model in Event-B

In this model all the possible primitives are sets for modeling the ZigBee along with all the entities of networks like application layer, network layer and MAC layer. Rodin platform is used for proof obligation. For the exchange of data in Event B. Data_NWK primitive is used that can exchange data from the nearby entity of application layer. CONFRM_DATA primitives report the results for requesting the data from application layer. At the final stage the IND_DATA shows the transfer of data from network layer.

Different functions are performed by ZigBee devices like joining a network, leaving a network, rejoining a network etc. The permission of these functions (Joining or Leave) is granted by both ZigBee coordinators and routers. Routers also maintain the routing list of all neighbors as well as assign logical network addresses.

CONTEXT DATA
SETS
NWK
NWKEY
RQ_STATUS
ZB_OBJ
CONSTANTS /* 8 constants are used*/
DATA_REQ CONFRM_DATA IND_DATA
REQ_INVLD COUNT_FRM NODES
SW_ROUTR COORD
AXIOMS
axm1: DATA_REQ ∈NWK∧CONFRM_DATA ∈NWK
axm2: IND_DATA∈NWK∧REQ_INVLD∈ RQ_STATUS
axm3: NODES ∈ NWKEY ∧RQ_STATUS
axm4: COUNT_FRM ∈RQ_STATUS
axm5: NODES ⊆ZB_OBJ∧RSW_ROUTR ⊂ZB_OBJ∈COORD ⊂ZB_OBJ
axm6: NODES ∪ (SW_ROUTR ∩ COORD) =∅SW_ROUTR ∧COORD =∅
END

A machine for access control for secure user credential transfer is developed according to the protocol specification described earlier; all the events for secure access are initialized to their default values. At the time when the network access in not provided yet it keeps empty. In a network coordinators are playing main role for the secure access provision and necessary permission granted. In the following machine a ACCESS_CONTROL a function named ACCESS is defined and set its value to Boolean. The aim of this function is to describe that user in a network is capable to gain access on the network resources. At the time, when a network access is granted by the coordinator, the value of events may change. Similarly, parent user function is defined for pair node of coordinators or routers. The parent user function is also helpful for joining different users to the secure network. Events and

variables are used to refine the Event-B abstract model. Different layers primitives are modeled by using events. The Machine ACCESS_CONTROL initialization event is model as under:

MACHINE ACCESS_CONTROL
SEES USERCRD
VARIABLES
ACCESS CREDENTIALS NETWORK NWLYR NWKEY MACLYR PARENT_USER USERS STATUS COORD SW_ROUTER
INVARIANTS
inv1: NETWORK ⊆ ACCESS ∧ NWLYR ∈ MACLYR ∧ NWKEY ∈ ZB_OBJ → BOOL
inv2: NETWORK ∈ STATUS ∧ PARENT_USER ∈ {COORD ∪ SW_ROUTR = USERS}
EVENTS
INITIALISATION
begin
act1: USER1 ACCESS := TRUE
act2: COORDACCESS := FALSE
act3: SW_ROUTRACCESS := TRUE
act4: NETWORK := ∅
act5: ACCESS := {USER1 ↦ FALSE; USER2 ↦ FALSE; COORD ↦ TRUE; SW_ROUTR ↦ FALSE}
act6: PARENT_USER := {COORD ↦ USER1} ∪ SW_ROUTR ↦ USER2}
end
END

4. Results and Discussion

Formal specification of security properties have been developed in Event-B by using Camile Editor. This formal specification has been analyzed by using RODIN tool. These formal specifications represent abstract formal models and more refinement is required in order to integrate these models into real world systems. These abstract formal specifications of security properties consist of a static model and a dynamic model. The static aspects of the security properties are defined in the static model and dynamic properties are defined in the dynamic model. This formal specification contains details related to abstract models of security properties. All the specification described in the previous paragraphs has been verified and the results are shown with RODIN tool screen shots. The verification is carried out by writing the temporal logic context and then uses the relevant machines to proof obligations.

4.1 Verification Results of ZigBee Protocol Model

As the next step is ZigBee verification is done using Rodin tool. In which the network layer security and MAC layer specification is verified by proof obligations. Rodin platform is selected for this verification method, in which all the primitives are developed and request for joining the ZigBee network is check by creating the context and its machine. The authenticated user connectivity by transferring its credentials are verified. After successful connectivity the message transformation and total no of frames transferred are also verified.

Different functions are performed by ZigBee devices like joining a network, leaving a network, rejoining a network etc. The permission of these functions (Joining or Leave) is granted by both ZigBee coordinators and routers. Routers also maintain the routing list of all neighbors as well as assign logical network addresses.

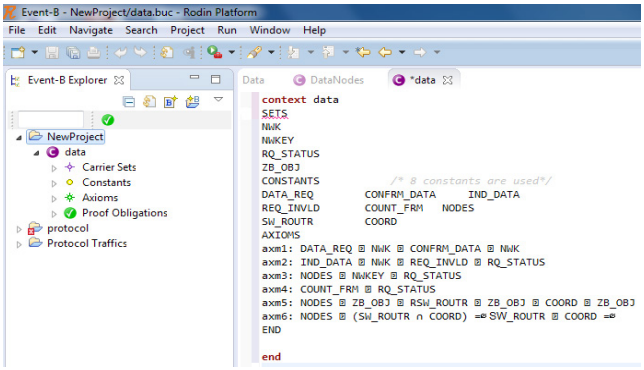

Figure 3. Proof Obligation Results of Context Data in Rodin tool.

In the Figure 3 our model is built for Data entity in which the first part defines sets. The first two sets contain the NWLYR (network layer) primitive type which is used for layer interface identification commands and for the protocol interface and REQ_STATUS is used to tell the messages status in a ZigBee network. The last and third set is used for possible devices that can join or leave named as ZB_OBJ. This set consists of COORD (coordinators), SW_ROUTR (routers) and nodes. After this model, all the axioms are defined in which network layer along with security parameters are defined. Axioms are used to follow by the components. All the restrictions are

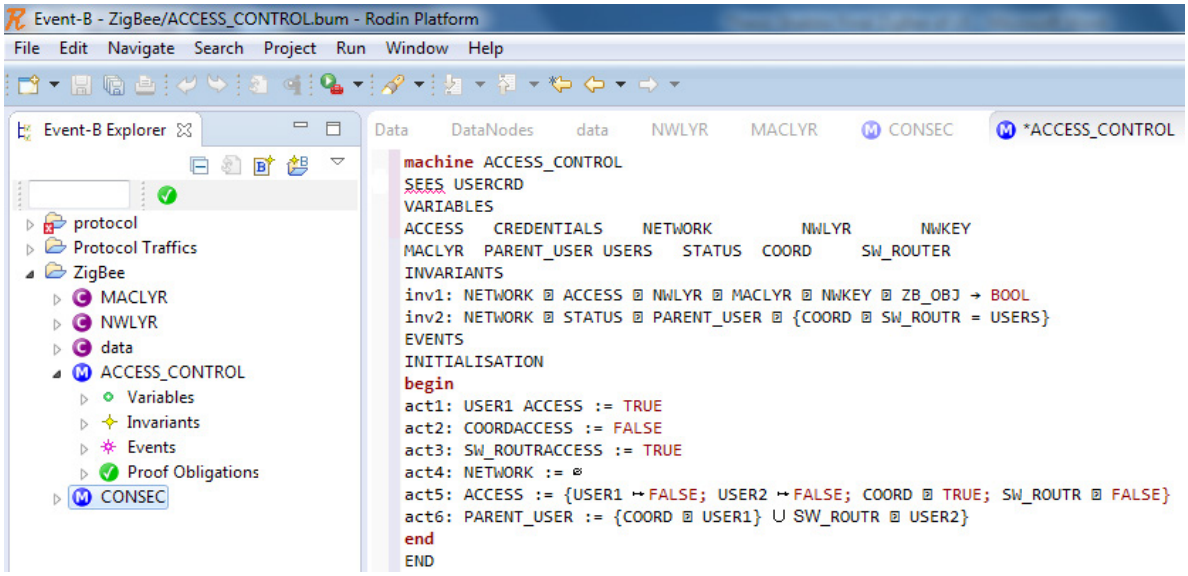

Figure 4. Proof Obligation Results of Machine ACCESS_CONTROL in Rodin tool.

set by axioms in a verification model. The verification of the above formal specification is verified using RODIN tool.

In the stage of initialization of event, a machine for access control for secure user credential transfer is developed according to the protocol specification described earlier; all the events for secure access are initialized to their default values. At the time when the network access in not provided yet it keeps empty. In a network coordinators are playing main role for the secure access provision and necessary permission granted. In the following machine a ACCESS_CONTROL a function named ACCESS is defined and set its value to Boolean. The aim of this function is to describe that user in a network is capable to gain access on the network resources. At the time, when a network access is granted by the coordinator, the value of events may change. Similarly, parent user function is defined for pair node of coordinators or routers. The parent user function is also helpful for joining different users to the secure network. The Machine ACCESS_CONTROL initialization event is model (defined) and verification results are shown in Figure 4.

5. Conclusion

The main aim of this study is to investigate the use of verification techniques powered by formal methods. Our goal was to demonstrate that current techniques in formal verification can be combined in a smart way to be useful in the verification of communication technologies especially where limited resources come in play and make things harder. The formal models properties have been developed by using Event-B and RODIN tools. In Event B proof obligations are used for properties correctness and consistence. Different function and operation of the protocols are defined by events.

These formal specifications has been analyzed and verified by using RODIN tools. We develop abstract formal specifications of security properties consist of a static model and a dynamic model. The static aspects of the security properties are defined in the static model and dynamic properties are defined in the dynamic model. In our model parent user function is also helpful for joining different users to the secure network. Verification of the security property of ZigBee protocol from a point of view that is close to the implementation and realization is carried out in this papers. We stated a new verification approach that first verifies the low level protocols rigorous manner and guarantees absolute security. We used modern tools for verification that can return probabilistic results with respect to the trade of between security and performance.

6. Future Work

In the future research using Event-B and Rodin tool a formal validation of security protocols like IEEE802.15.4 and ZigBee can be carried out. For cross check two powerful

methods program analysis and model checking can be combined in a way to benefit from the advantages of both of the methods. Thus, the state space explosion problem in model checking and lack of trace problem in program analysis can be eliminated. Developing an approach in determining the proper set of input data, and eventually getting a set of results is one of the points that need attention.

7. References

1. Zou Y. Improving physical-layer security in wireless communications using diversity techniques. IEEE Network. 2015; 29(1):42-8. Crossref.
2. Chen Z. A review of automated formal verification of ad hoc routing protocols for wireless sensor networks. Sensor Letters. 2013; 11(5):752-64. Crossref.
3. Damiano M, Merro M. A semantic analysis of key management protocols for wireless sensor networks. Science of Computer Programming. 2014; 81:53-78. Crossref.
4. Baronti P. Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards. Computer Communications. 2007; 30(7):1655-95. Crossref.
5. Wu B, Lin H, Lemmon M. Formal methods for stability analysis of networked control systems with IEEE 802.15. 4 protocol. 53rd IEEE Conference on Decision and Control. IEEE. 2014; p. 5266-71. Crossref.
6. Arapinis M. Statverif: Verification of stateful processes. Journal of Computer Security. 2014; 22(5):743-821. Crossref.
7. Meng B. Mechanized Verification of Security Properties of Transport Layer Security 1.2 Protocol with Crypto Verif in Computational Model. Information Technology Journal. 2014; 13(4):601. Crossref.
8. Armando A, Carbone R, Compagna L. SATMC: A SAT-based model checker for security-critical systems. Springer Berlin Heidelberg: International Conference on Tools and Algorithms for the Construction and Analysis of Systems. 2014; p. 31-45.
9. Macedonio D, Merro M. A semantic analysis of key management protocols for wireless sensor networks. Science of Computer Programming. 2014; 81:53-78. Crossref.
10. Prouff E, Rivain M. Masking against side-channel attacks: A formal security proof. Springer Berlin Heidelberg: Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2013; p. 142-59. Crossref.
11. Aiash M, Loo J. Introducing a novel authentication protocol for secure services in heterogeneous environments using Casper/FDR. International Journal of Communication Systems. 2014; 27(12):3600-18. Crossref.
12. Ahmad S, Ehsan B. The Cloud Computing Security Secure User Authentication Technique (Multi Level Authentication). IJSER. 2013; 4(12):2166-71.
13. Xie Q. Improvement of a three-party password-based key exchange protocol with formal verification. Information Technology and Control. 2013; 42(3):231-7. Crossref.
14. Ahmad S, Hussain S, Iqbal MFI. A formal model proposal for wireless network security protocols. Science International. 2015; 27(3):1-6.
15. Meseguer J. Twenty years of rewriting logic. The Journal of Logic and Algebraic Programming. 2012; 81(7):721-81. Crossref.
16. Grimm V. Towards better modelling and decision support: documenting model development, testing, and analysis using TRACE. Ecological Modelling. 2014; 280:129-39. Crossref.
17. Biondi F. QUAIL: A quantitative security analyzer for imperative code. Springer Berlin Heidelberg: International Conference on Computer Aided Verification. 2013; p. 702-07. Crossref.
18. Hoang, Son T, Furst A, Abrial JR. Event-B patterns and their tool support. Software and Systems Modeling. 2013; 12(2):229-44. Crossref.
19. Rodin P. Available from: http://www.event-b.org. Date accessed: 01/01/2015.