Formal Verification of an Efficient Architecture to Enhance the Security in IoT

Eman K. Elsayed[1], L. S. Diab[2], Asmaa. A. Ibrahim[3]
Mathematical and Computer Science
Al-Azhar University
Cairo, Egypt

***Abstract*—The Internet of Things (IoT) is one of the world's newest intelligent communication technologies. There are several kinds of novels about IoT architectures, but they still suffer from security and privacy challenges. Formal verification is a vital method for detecting potential weaknesses and vulnerabilities at an early stage. During this paper, a framework in the Event-B formal method will be used to design a formal description of the secure IoT architecture to cover the security properties of the IoT architecture. As well as using various Event-B properties like formal verification, functional checks, and model checkers to design different formal spoofing attacks for the IoT environment. Additionally, the Accuracy of the IoT architecture can be obtained by executing different Event-B runs like simulations, proof obligation, and invariant checking. By applied formal verification, functional checks and model checkers verified models of IoT-EAA architecture have automatically discharged 82.35% of proof obligations through different Event-B provers. Finally, this paper will focus on introducing a well-defined IoT security infrastructure to address and reduce the security challenges.**

***Keywords*—Internet of things (IoT); IoT architecture; IoT security; formal modeling and verification; Event-B**

I. Introduction

The Internet of Things (IoT) is one of the most recent research topics these days. IoT [1] allows various devices to communicate with one another over the Internet. As a result, it ensures that the device is intelligent and sends information to a central system, which will check and take necessary measures by the task at hand. To make the IoT paradigm a reality, things or objects must be identified, as well as sensing, networking, and processing capabilities.

Recent advances [2] in wireless technology, advanced communications, and intelligent systems have demonstrated a strong potential and a strong attempt to enhance human life in every way possible. Depending on the different application domains of IoT, the heterogeneity of the devices, and the ubiquitous communication, IoT is primarily composed of several sensors (wireless and automatic). It requires a deep understanding of IoT architecture. Its architecture is made up of four main layers [3, 4]; Perception, Network, Middleware, and Application layer. The interconnection of massive heterogeneous frameworks and networks of systems is referred to as IoT technology.

Because IoT devices [5] have different designs, implementations, and maintenance, they have a variety of problems and weaknesses in their software and hardware.

When all the security requirements are met successfully, a system is considered secure [6]. Confidentiality, integrity, authentication, availability, authorization, non-repudiation, and privacy are all essential requirements. For each one-off them and must ensure security in all different layers from different threats. As a result, the entire deployment architecture must be secured from attacks that could hinder IoT services or jeopardize data privacy, integrity, or confidentiality.

Since this Internet of Things is composed up of interconnected networks and heterogeneous devices, it inherits the security problems facing computer networks. Since small devices or items with sensors have limited power and memory, IoT protection is further complicated by resource constraints. Consequently, security solutions need to be adapted to the constrained architectures.

Recently, a lot of effort has gone into dealing with security issues in the IoT paradigm. Some of these approaches focus on a particular layer of security, while others aim to provide end-to-end security for IoT. According to [7], the author proposed a new efficient and secure architecture model for the Internet of things called IoT-EAA, which tends to provide end-to-end security for IoT through the top one of the IoT applications, as well as resolving various security issues that exists at various bottom layers. “Fig. 1” shows the IoT-EAA security architecture model, which contains five layers, (Hardware Layer, Network Communication Layer, Service Application Layer, Connectivity Management Layer, and Security Layer).

Fig. 1. IoT-EAA Architecture.

Event-B is a formal method for formal specification and development of systems [8] that an extension of method B. In formal Event-B methods, step-by-step models of systems can be created starting with an abstract model and enrich the abstract models with more details to form concrete models.

To ensure that a refined model conforms to abstract models [9], a series of test loads are generated to show that the refinements are correct. Event models B contain two parts called context and machine. The static part of the model, such as sets and constants, is contained in the context, while the dynamic part, such as variables and events, is contained in the machine. The main character in Event-B is refinement, which allows for the system's gradual development.

Rodin platform tool [10, 11] introduces support for Event-B modeling, automatic creation, and proving rules. Rodin is an Eclipse-based application that implements Event-B. An environment includes advanced automated provers such as PP, ML, and SMT, which generate proofs for refinements, feasibility, invariants, and well-definedness of expressions within guards, acts, and invariants. When the automatic proof discharge fails, a manual proof discharge is used. Event-B also includes an interactive proving method for manual proof. Rodin platform has a critical feature, which is the proof obligation generator. It generates proof obligations.

IoT network security [12] is divided into two categories: technological challenges and security challenges. The technological challenges are those that arise as a result of the heterogeneity and ubiquity of devices, while the security challenges are primarily related to the system's basic functions. The technological challenges mainly include [13] scalability, performance, computing, wireless technologies, and the distributed paradigm while security challenges include ensuring confidentiality, integrity, end-to-end security, and permanent availability of services.

There is a different security threat to IoT such as Denial of Service, Brute Force, Man in the middle attacks and many other attacks are visualized in the interconnected network. There are several reasons [14] for occurs these attacks like weak passwords, no encryption, personal information leakage, etc., if such security attacks are not solved to some safe level the market of IoT will be harmful because of the week service of security. It involves not only these security issues but also have other issues of access control, authentication of different networks, and some problems of the information store. This problem requires having a well-defined security infrastructure to address these problems and reduce security Threats [15].

This paper introduces a contribution by using one of the most important formal methods called Event-B, to enhance the security of IoT technology. This involves model checking and theorem proving for IoT architecture discharge in the Rodin platform. Hence, this paper will provide structured verification for IoT architecture that focuses on security checking for each IoT architecture layer, which considers the early stages of building the IoT systems.

The rest of the paper is organized as follows. Section II discusses related works about using formal methods in the IoT area, including previous studies for the IoT and formal methods. Section III proposed some mathematical definitions for the configurations of the IoT-EAA architecture as well as our methodology for proofing IoT-EAA architecture mathematically. Section IV discusses the verification method. Finally, the last part presents the concluding remarks and future work in Section V.

II. Related Work

Formal Verification is a promising method for ensuring security by using a variety of mathematical and logical methods to mathematically verify the accuracy of designs. Formal methods are used to implement several approaches in the IoT domain.

In [16] authors review formal methods for various protocols used in the IoT environment. They concern with the security mechanisms for communication protocols in the IoT communication layer only, but in this paper, we used formal verification methods to check the security mechanisms for each layer in the IoT architecture.

In [17] Authors improve the security and detecting various security issues at an early stage for the IoT application layer by introducing formal methods on different protocols in this layer. However, the authors concentrate on the security mechanisms for protocols in the IoT application layer only.

In [18] authors suggest a unified approach for verifying the communication protocols over a framework using machine-decomposition within Event-B. However, this approach does not introduce security properties in the IoT area.

The authors of [19] presented a comprehensive study of the most used formal verification methods and approaches for verifying and analyzing the correctness of cryptographic protocols and algorithms' security properties.

Authors in [20] introduced an automated alternative approach for supporting the early stages of the security verification process in chains. The proposed strategy analyzed the control and data planes, which included various security algorithms established in chains as security functions.

The SAT-based Model-Checker (SATMC) was suggested by the authors in [21] as a systematic verification method for verifying the correctness of critical security systems. Security protocols, business processes, and application programming interfaces for security were all included (APIs).

III. Methodology

This section introduces the proposed method to verify the correctness of the IoT-EAA mathematically through two phases which can be classified into two sub-sections. The first subsection introduces the mathematical description of the IoT-EAA architecture model, and the second subsection will illustrate Modelling and Verifying IoT-EAA Architecture using Event-B.

A. Mathematical Description for the IoT-EAA Architecture

This section describes the IoT-EAA architecture's mathematical description, including its composite entities and operational functions. The key physical and virtual components of the IoT-EAA architecture are also explained below.

- Definition1: (service Application layer) that is defined as a three-tuple.

$A = [A_{id}, A_{type}, A_{sp}]$

Where A_{id} denotes the application ID and A_{type} denotes the purpose for which the application is used (such as medicine, education, finance, entertainment, utility, and gaming). A_{sp} specifies the minimum system requirements for running the application, such as the Processor, primary memory, and secondary storage requirements, as well as the operating system version.

- Definition2: (Network communication layer) is defined as a six-tuple.

$NC = [ND, S, T, T_s, R, D]$

Where ND is Network devices that called routers are used to direct packets between networks. Also, S denotes the Source, which generates data to be transmitted (sensors or actuators), and T is the Transmitter that Converts data into transmittable signals through Ts, which the Transmission System that Carries data to the R Receiver to Convert the received signal into data and received it to the D the Destination that Takes the incoming data.

- Definition3: hardware layer denoted by HW and defined as a three-tuple.

$HW = [HW_{id}, HW_{st}, HW_{type}]$

where, HW_{id} is an integer that represents the hardware's unique ID.

HW_{st} represents whether the hardware is in an active or inactive state, and is represented as a Boolean, $HW_{st} = \{0, 1\}$, where the values 0 and 1 symbolize the inactive and active states, respectively.

- Definition4: The specifications of the Hardware denoted by (HWtype) are represented as a six-tuple.

$HW_{type} = [P, M, B, S, c, f]$

where, P stands for the hardware processor specifications, which include information such as processor core speed, bus specifications, and internal register (cache memory) size. The memory size, memory clock, and data rate requirements for primary memory (RAM) are stored in M.

Tuple B contains information about the battery, such as voltage, size (AA or AAA), type (Ni or C electrodes), and the number of batteries needed is the symbolic representation of the various kinds of sensors that make up the node's sub-modules. The hardware used for wireless communication for the node, such as Bluetooth and ZigBee, is represented by the tuple c. f denotes the frequency range in which the HW runs.

- Definition5: connectivity management layer is defined as a two-tuple.

$CM = [HM, NM]$

Where the HM dented the management of IoT hardware and NM dented the management of Network communication.

Property1: The function of connectivity management, which mange the connection between *HW* and *NC* as represented in Equation (1).

$$F(CM): HM \longleftrightarrow NM \quad (1)$$

The operator $\longleftrightarrow$ dented the management of the connectivity between HW and NC layers.

Now all components of the IoT architecture will define in Equation (2).

$$\text{IoT AR} = \sum ((HW \succ NC) / CM) \succ A \quad (2)$$

The operator $\succ$ denotes the existence of a successor relationship between two operands. For example, $X \succ Y$ denotes that Y is a successor of X.

To satisfy the security in wholly the IoT architecture as represented in Equation (3).

$$\text{IoT AR} = \sum ((HW \succ NC) / CM) \succ A \Leftrightarrow S \quad (3)$$

The proposed theory for IoT security: the IoT application service satisfies a high degree of security if and only if secure the connection of hardware devices and network by managing the connection between them.

B. Modeling and Verifying IoT-EAA Architecture using Event-B

Formal methods consider an important tool for providing quantitative statements about safety and security properties for the digital systems [22]. These methods are usually used to formally verify a model. Therefore, Model checkers and Theorem provers are two different types of Formal Method tools. In model checkers, a system's model verifies its state space exhaustively and automatically according to a given specification. Human expertise is often required by theorem provers to guide the proof of correctness by providing design and specification characteristics as algebraic constraints or theorem [23].

Some tools, such as AVISPA [24], Scyther [25], and Tamarin, concentrate on security protocols, while others, such as UPPAAL [26], PRISM [27], and Rodin platform [11], focus on Event-B modeling for statistical and probabilistic verification. When it comes to security design verification, the primary objective is usually to verify or falsify security properties such as secrecy and authentication.

Table I shows the various tools for verifying IoT protocols as well as the architecture for probabilistic/statistical model checkers.

According to Table I, the Event-B formal method will be used, which has the simulations and proof obligations that include both model checker the theorem prover that tends to verify the correction of the IoT-EAA architecture model.

TABLE I. PROBABILISTIC / STATISTICAL MODEL CHECKER

	UPPAAL	PRISM	Rodin
Input language	XTA and XML	PRISM language	Event-B language
typical applications	real-time controllers and communication protocols with critical timing aspects	verification of probabilistic real-time systems	Validation and verification of probabilistic real-time systems
statistical model checking	√	√	√
probabilistic model checking	✗	√	√
Model Checker	√	√	√
Theorem Prover	✗	✗	√
Simulator	√	√	√
GUI	√	√	√
Case Studies	√	√	√

The IoT-EAA is established in Event-B. To get a better overview of the IoT-EAA architecture, the Event-B refinement technique will be used to build the IoT-EAA Event-B model gradually and follow, down – top layers, at the initial model the down layer called the Hardware layer that defines the properties for different devices, which are used for data collection. Then go to the top layer in the contract model through two refinements called machine for network communication layer, which refines the machine for the hardware layer and sees the context for the network communication layer. As well as a machine for the service application layer that refines the machine for the network communication layer and sees the context for the service application layer.

To present the IoT-EAA architecture, additionally, introduce three incremental refinements of the IoT-EAA architecture model. These refinements implemented by Event-B modeling language to formalize the given architecture refinements implemented by Event-B modeling language to formalize the given architecture.

As shown in Fig. 2 the relationship between context and machine for IoT-EAA architecture is described. Machines and Event-B contexts are included in the model. The contexts contain all the required data structures and axioms to set up a machine.

The IoT architecture layer is implemented as events on the machine, and the properties that must be verified are written as invariants.

The Initial Model (Hardware layer): it contains several devices, practically; by using the Rodin platform in the preparation phase consisted of the device state on/off, An Event-B context declares a device state-defined using axiom3 for device state. An abstract model declares a list of variables defined by invariants (inv3 – inv8); as well as different events for the network communication layer and security as shown in Fig. 3.

Three events were introduced to one event to specify the desired functional behavior for the hardware layer of an IoT-EAA, As well as an event for the connectivity management layer and the security layer of an IoT-EAA. These events include guard(s) for enabling the given action(s) and the actions that define the changes to the states of the hardware layer. Here, we provide all events related to the hardware layer (data collection, manage the connection, and security), the hardware layer machine component will be described.

The first refinement (Network Communication layer): this refinement refined the initial model behavior into two phases; one focus on the network communication and the other phase refined the connectivity management layer and security layer into several sub-events. Practically, in this refinement, which includes (manage connection and security) events.

As well as Two new events to specifying the desired functional behavior (send data and receive data) are introduced in the network communication layer. In this refinement, we define an enumerated set and a list of variables to formalize the network communication operations defined by invariants (inv1 – inv11) that will be described in Fig. 4.

The second refinement (Service Application layer): this refinement can refine the Network Communication layer by introducing detailed events for the Service Application layer such as an interface with end-users that able to be linked for the major gap between users and applications; as well as security events for the security layer. In this refinement, an enumerated set and a list of variables were defined to formalize the service application operations by invariants (inv1 – inv5) context and machine for these refinements will be described in Fig. 5.

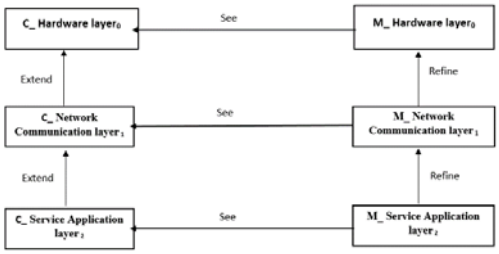

Fig. 2. Machine and Context Relationships for IoT- EAA Architecture.

Fig. 3. Variables and Invariants for the Hardware Layer.

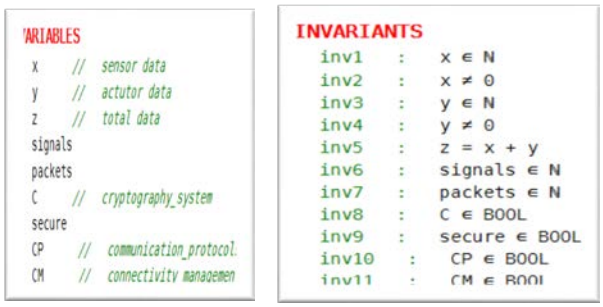

Fig. 4. Machine for Network Communication Layer.

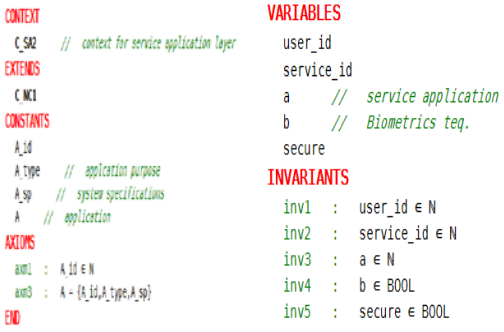

Fig. 5. Context and Machine for the Service Application Layer.

IV. Verification Methods

In this section, we present the backbone of Event-B called a proof obligation generator [10]. This step runs after the static checker that checks the texts of the contexts and machines.

The validated models of IoT-EAA have together discharged 140 proof obligations, of which 82.35% proof obligations were automatically discharged through different Event-B provers. Well-definedness of predicates and expressions in invariants, guards, actions, variants, and witnesses for all events, feasibility checks, variable reuse check, guard reinforcing, and witness feasibility in refinements are all part of the proof obligations.

Variant checks for natural numbers and decreasing variants for convergent and predicted occurrences, theorems in axioms and invariant preservation for refinements and invariants used for verification of required security properties, theorems in axioms and invariant preservation for refinements and invariants used for verification of required security properties, theorems in axioms and invariant preservation for refinements and invariants used for verification of required security properties.

- Detecting some IoT security attacks using Event-B formal method.

IoT vision has been suffered from unprecedented attacks, which have resulted in the loss of privacy, organized crime, mental anguish, and the potential for human life to be jeopardized [28]. IoT has different attacks that occur in different IoT layers, one of these attacks called spoofing attack [29] is introduced, which considers a more dangerous attack for IoT applications.

Spoofing is the act of misrepresenting a communication from an unknown source as coming from a reliable source. Spoofing attacks can target a variety of domains, including emails, phone calls, and websites, or they can be more technical, like a computer spoofing an IP address; spoofing an email, Address Resolution Protocol (ARP), or Domain Name System (DNS) server.

On IoT nodes, a dynamic IP address attachment can be expanded from IPV4 to IPV6 when IPV4 addresses are insufficient for future requirements. Simple changes such as the IP stack are updated to support message exchange and avoid the use of complex cryptographic schemes for authentication.

To verify that the proposed method is useful for securing the IoT applications, various types of spoofing attacks are detected using the Rodin platform. We applied two types of spoofing attacks called “ip_address _spoofing” and “ARP_spoofing.” Executing various runs and observing the sequence of events and variable values in each of these events will provide accuracy in securing the model.

By establishing a new event in the machine of the hardware layer for IoT_EAA architecture detected the security error because the secure action must be “FALSE ” (if the IP address for the hardware layer does not equal the IP address for the attacker device this considers conflict as well as event guard that is (if the security protocol sp is true then the security must be false) as illustrated in Fig. 6 with representing the mechanism of Event-B for detecting different types of spoofing attacks.

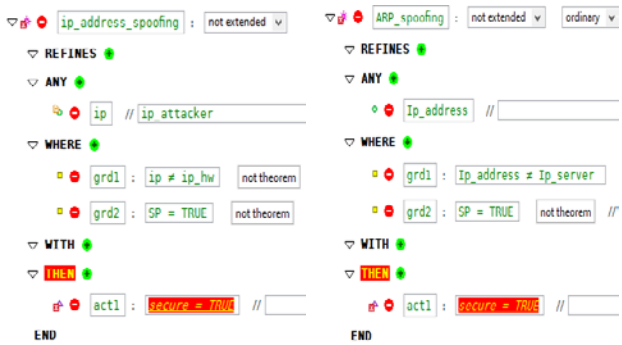

Fig. 6. Different Types of Spoofing Attack effect in Security.

V. Conclusion and Future Works

This paper looks at a vital application for the Formal Verification of IoT Architectures, focusing on security mechanisms. That is, different Event–B properties such as simulations, proof obligation, and invariant checking are used to verify the accuracy of the IoT-EAA architecture, which are then discharged in the Rodin platform to enhance security and detect security concerns at an early stage. Also, each IoT-EAA architecture layer's security issues will be discussed. Using the proposed method, various types of spoofing attacks were introduced in the Roin platform. We verified that various security properties are discovered, as well as the proposed IoT Architecture (IoT-EAA) in general.

In future work, IoT-EAA architecture will be enhanced to cover all semantic IoT security properties. As well as using different verification methods to verify various types of IoT protocols.

REFERENCES

[1] M. U. Farooq, M. Waseem, S. Mazhar, A. Khairi and T. Kamal, "A Review on Internet of Things (IoT)", International Journal of Computer Applications 113(1):1-7, 2015.

[2] O. Mariya, and A. Rhattoy. "A secure model for machine to machine device domain based group in a smart city architecture." International Journal of Intelligent Engineering and Systems 12.1, 151-164 , 2019.

[3] M. U. Farooq, M. Waseem, S. Mazhar, A. Khairi and T. Kamal "A Critical Analysis on the Security Concerns of Internet of Things (IoT)," International Journal of Computer Applications (0975 8887) Volume 111-No. 7, 2015.

[4] S. Vashi; J. Ram; J. Modi; S. Verma; C. Prakash "Internet of Things (IoT): A vision, architectural elements, and security issues", In 2017 international conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) (pp. 492-496). IEEE, 2017.

[5] Soumyalatha, S. G. Hegde, "Study of IoT: understanding IoT architecture, applications, issues and challenges", In 1st International Conference on Innovations in Computing & Net-working (ICICN16), CSE, RRCE. International Journal of Advanced Networking & Applications, May 2016.

[6] M. Imdad, D. W. Jacob, H. Mahdin, Z. Baharum, S. M. Shaharudin, & M. S. Azmi, "Internet of things (IoT); security requirements, attacks and counter measures". Indonesian Journal of Electrical Engineering and Computer Science, 18(3), 1520-1530, 2020.

[7] A. A. Elngar, E. K. Elsayed, & A. A. Ibrahim, "A New Efficient and Secure Architecture Model for Internet of Things". In International Conference on Innovative Computing and Communications (pp. 401-416). Springer, Singapore, 2020.

[8] J.-R. Abrial, Modeling in Event-B: System and Software Engineering. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[9] A. S. A. Hadad, Ma, C., & A. A. O. Ahmed, "Formal Verification of AADL Models by Event-B". IEEE Access, 8, 72814-72834, 2020.

[10] J. R. Abrial, M. Butler, S. Hallerstede, T. S. Hoang, F. Mehta, & L. Voisin, "Rodin: an open toolset for modelling and reasoning in Event-B", International journal on software tools for technology transfer, 12(6), 447-466, 2010.

[11] Rodin: "A Tool for Event-B formal method". [Online]. Available: http://wiki.Event-B.org/index.php/Rodin Platform.

[12] R. Mahmoud, T. Yousuf, F. Aloul, & I. Zualkernan, "Internet of things (IoT) security: Current status, challenges and prospective measures. In 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST) (pp. 336-341). IEEE, 2015.

[13] Khan, M. A., & Salah, K. "IoT security: Review, blockchain solutions, and open challenges. Future Generation Computer Systems, 82, 395-411, 2018.

[14] A. W. Ahmed, M. M. Ahmed, O. A. Khan, & M. A. Shah, "A comprehensive analysis on the security threats and their countermeasures of IoT. International Journal of Advanced Computer Science and Applications, 8(7), 489-501, 2017.

[15] I. Andrea, C. Chrysostomou, & G. Hadjichristofi, "Internet of Things: Security vulnerabilities and challenges". In 2015 IEEE symposium on computers and communication (ISCC) (pp. 180-187). IEEE, 2015.

[16] K. Hofer-Schmitz, & B. Stojanović, "Towards formal verification of IoT protocols: A review". Computer Networks, 174, 107233, 2020.

[17] K. Hofer-Schmitz, & B. Stojanović, "Towards formal methods of IoT application layer protocols". In 2019 12th CMI Conference on Cybersecurity and Privacy (CMI) (pp. 1-6). IEEE, 2019.

[18] M. Diwan, & M. D'Souza, "A framework for modeling and verifying IoT communication protocols", In International Symposium on Dependable Software Engineering: Theories, Tools, and Applications (pp. 266-280). Springer, Cham, 2017.

[19] M. A. Al-humaikani, L. B. A. Rahim, A Review on the Verification Approaches and Tools used to Verify the Correctness of Security Algorithms and Protocols, International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 10, No. 6, pages 146-152, 2019.

[20] N. Schnepf, R. Badonnel, A. Lahmadi, S. Merz, "Automated verification of security chains in software-defined networks with synaptic", Proc. IEEE Conference on Network Softwarization (NetSoft), 2017.

[21] A. Armando, R. Carbone, L. Compagna, "SATMC: a SAT-based model checker for security protocols, business processes, and security APIs", Int. J. Softw. Tools Technol. Transf., Vol. 18, No. 2, 2016.

[22] K. Keerthi, I. Roy, A. Hazra, and C. Rebeiro, "Formal verification for security in iot devices," in Security and Fault Tolerance in Internet of Things. Springer, pp. 179–200, 2019.

[23] R. C. Armstrong, R. J. Punnoose, M. H. Wong, & J. R. Mayo, "Survey of existing tools for formal verification". SANDIA REPORT SAND2014-20533, 2014.

[24] AVISPA. "A tool for Validation of Internet Security Protocols." [Online]. Available: http://www.avispa-project.org/.

[25] Scyther: "A tool for the automatic verification of security protocols." [Online]. Available: https://people.cispa.io/cas.cremers/scyther/.

[26] UPAAL: "A toolbox for modeling, simulation and verification of real time systems." [Online]. Available: https://uppaal.org/.

[27] PRISM: "A tool for formal modelling and analysis." [Online]. Available: https://www.prismmodelchecker.org/.

[28] H. A. Abdul-Ghani, D. Konstantas, & M. Mahyoub, "A comprehensive IoT attacks survey based on a building-blocked reference model", International Journal of Advanced Computer Science and Applications, 9(3), 355-373, 2018.

[29] N. Wang, L. Jiao, P. Wang, M. Dabaghchian, & K. Zeng, "Efficient identity spoofing attack detection for iot in mm-wave and massive mimo 5g communication", In 2018 IEEE Global Communications Conference (GLOBECOM) (pp. 1-6). IEEE, 2018.