Computer Networks 174 (2020) 107233

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Towards formal verification of IoT protocols: A Review

Katharina Hofer-Schmitz*, Branka Stojanović

DIGITAL – Institute for Information and Communication Technologies, JOANNEUM RESEARCH Forschungsgesellschaft mbH, 17 Steyrergasse, Graz, Austria

ARTICLE INFO

MSC:
00-01
99-00

Keywords:
Formal verification
Security
Protocols
Model checkers
IoT

ABSTRACT

Formal Verification is one of the crucial methods to detect possible weaknesses and vulnerabilities at an early stage. This paper reviews formal methods for an extensive variety of protocols used in the IoT environment. It gives detailed descriptions of the considered properties and the applied methods. An in-depth literature review shows that four application fields can be distinguished, namely: (1) functional checks, (2) checks on security properties, (3) suggestions for enhanced schemes including a priori security property checks and (4) implementation checks of protocols. This paper further offers a comprehensive overview of the covered security properties and of commonly used tools for protocols in the field. Additionally, an extensive description and overview of commonly used model checkers is given and open issues and challenges in the IoT field are addressed.

Contents

* Corresponding author.
E-mail addresses: katharina.hofer-schmitz@joanneum.at (K. Hofer-Schmitz), branka.stojanovic@joanneum.at (B. Stojanović).

https://doi.org/10.1016/j.comnet.2020.107233
Received 18 December 2019; Received in revised form 13 March 2020; Accepted 23 March 2020
Available online 30 March 2020

1. Introduction

The emerging IoT technology usage in different aspects of life results in a very large number of devices connected to each other and to the Internet in a small space. These, sometimes cheap, low-resource devices often have a high number of vulnerabilities [1]. This makes each device a potential point of attack and at the same time these devices use a variety of different communication protocols [2]. One of IoT usage domains with largest reach, is the Smart Home domain. Smart Home enables the integration of smart appliances in home environments, making people's life and everyday tasks easier. IoT technology is also widely used in smart industry environments, Smart Energy, etc. In a smart environment usually a wide number of devices and services provided by different suppliers are included, based on different technologies and standards. This heterogeneity leads to an increase in security issues. Although vendors try to agree on good practices in security implementations, the use of wireless communication and setting up the system by non-security experts, provides great opportunities for attackers to intercept communication, to steal confidential information or manipulate data. Furthermore, many of the devices in usage have limited resources in processing capacity and storage, which make the execution of complex security mechanisms difficult (see [3,4]).

Therefore, it is necessary to detect possible weaknesses and vulnerabilities already at design stage to secure such systems from the very beginning. A promising technique to find weaknesses and possible vulnerabilities at design stage is Formal Verification. It ascertains the correctness of designs by using a diverse set of mathematical and logical methods. By using such methods, it is possible to check different parts of the system, as the functional correctness of implementations, programming bugs, side-channel analysis, the fulfillment of security properties and hardware Trojans, from the very beginning and provide guarantees of security [5]. Moreover, some of the tools enable a qualitative and quantitative analysis of protocols.

Protocol's formal verification procedure is presented in Fig. 1. The procedure in general consists of four basic steps, the first is studying protocol's specification in detail. The next step is manually defining a model based on a given specification. Afterwards, the defined model is translated into the model checker input language. The last step is checking the results of formal verification and potentially defining recommendation for standard changes, based on those results.

This paper focuses on the application of Formal Verification to protocols in the IoT domain. In addition to a detailed description of existing work – including a comparison of existing approaches on considered properties and used tools – this review paper provides an extensive description of the most common tools in the field. Furthermore, an explanation of relevant security properties and relations to common and detected attacks is given. To the best knowledge of the authors, there is no review paper with that focus so far. An older survey of existing tools for formal verification from 2014 is given in [6]. There, the author provides a classification and short basic description of each tool. The focus there is, however, much broader covering tools for checking abstract models, hardware description languages and correctness of software but does not consider existing work applying those tools. Further related work is given in [7], where a short review of Formal Methods on the application layer IoT protocol's MQTT and CoAP is given. Due to existing work on the latter issue, formal verification approaches applied so far on these protocols, in literature, are not considered in this paper.

The paper is structured as follows. Section 2 first addresses security in communication protocols in general, providing information on security properties and common attacks in the field. Additionally, a description of the relevant protocols, including their security mechanisms, is given. Section 3 provides a basic introduction of Formal Verification methods and details of the most commonly used tools, enabling a valuable overview of these tools. Section 4 gives details of existing tools and techniques used to detect vulnerabilities with Formal Methods for selected protocols. Based on that, a classification of application fields, details to considered security properties and an overview about usage of commonly applied tools is provided. Finally, in Section 6 a conclusion including open challenges is given.

2. Communication protocols and security

The main goal of this paper is a comprehensive review of *Formal Verification* methods for IoT protocols' security.

Communication protocols in general represent a set of rules that allow two or more entities in a communication system to transmit information via different types of connections (e.g. physical media, layer...). The protocols are defined by rules, syntax and semantics and can be implemented by hardware, software or a combination of both.

Communicating systems use well-defined formats for exchanging various messages, where each message has an exact meaning intended to elicit a response from a range of possible responses pre-determined for that particular situation [8]. Since communication protocols have to be agreed upon communicating parties, they are usually defined in technical standards.

There are two important reference models for communication protocol standardization:

- OSI – open system interconnection reference model [9],
- TCP/IP – transmission control protocol and internet protocol reference model [10,11].

Both mentioned reference models provide a standard architecture to define network communication with several layers (Fig. 2).

In the IoT field, several layered reference models are proposed in literature [12,13]. In this paper we stick to the IoT reference model ([13]), which contains three layers, see Fig. 2. In order to provide an easier comparison, also the TCP/IP and the OSI model are given in Fig. 2.

A bottom-to-top description of the reference models' layers for all three mentioned models is provided in Table 1.

According to [14] the perception layer is classified with the highest security risk level. The reasons for that are hardware limitations that prevent the implementation of robust protection methods of the data collected, stored and transmitted at this layers. Furthermore, there is a wide range of devices, which makes the establishment of security and the standardization of communication protocols more difficult.

The following section first explains general and protocol related security properties, and then lists selected IoT protocols with focus on security features.

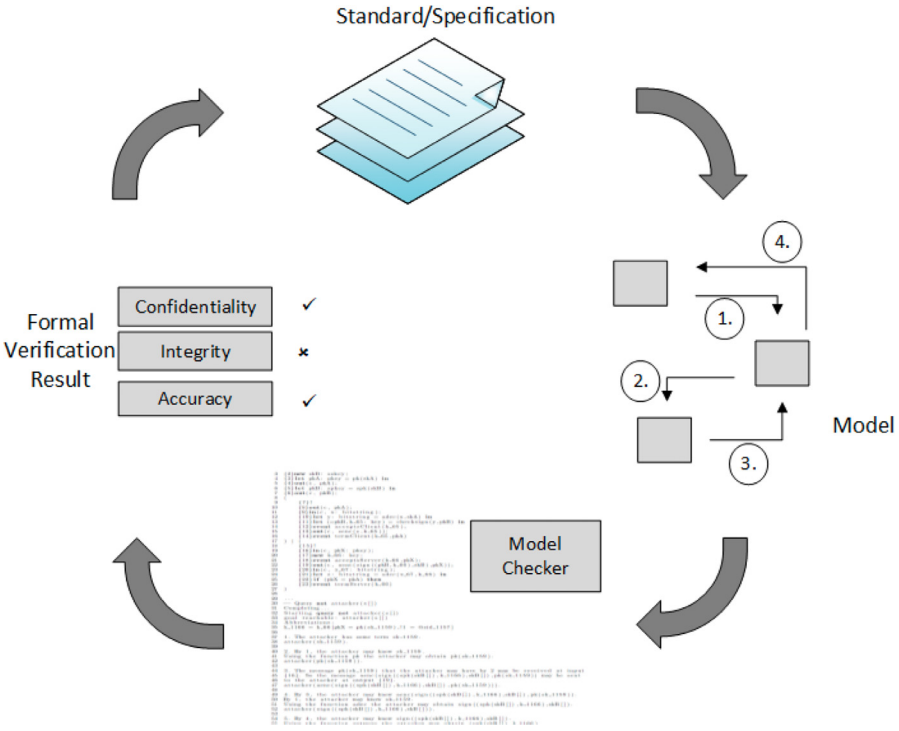

Fig. 1. Protocol's formal verification procedure overview.

Table 1
A bottom-to-top description of the layers of common reference models.

IoT [13]	TCP/IP	OSI	Description
Perception	*Network access*	*Physical*	Conveys the bit stream across the network through different physical media.
		Data link	Puts packets into network frames and sets up links across the physical network.
Network	*Internet*	*Network*	Handles addressing and routing the data – concept in which a frame is able to reach a destination beyond adjacent nodes.
	Transport	*Transport*	Manages packetization of data and delivery of the packets, including error checks and flow control.
Application	*Application*	*Session*	Sets up, coordinates and terminates conversations, including authentication and reconnection.
		Presentation	Manages the syntax processing of messages to be used in the application layer such as the encryption/decryption.
		Application	The layer where end-user applications are implemented.

2.1. Security properties

The value of information comes from the characteristics/properties it possesses – the value of information increases or, most commonly, decreases when a property of that information changes [15].

Critical properties of information that need to be protected in an information system are usually defined by the C.I.A. triangle – Confidentiality, Integrity, Availability.

Table 2 describes properties from the extended C.I.A. triangle. Accuracy and Authenticity, as additional security properties to the C.I.A. triangle, are identified as very important in modern information and communication systems [15].

When it comes to communication protocols two important security properties that are, or should be, addressed are Confidentiality (Secrecy) and Authenticity of communicating parties (Authentication).

In **Formal Verification Methods** both of these properties are usually explained under the assumption that the network is under full control of the adversary (*Dolev-Yao Model* [16]), where the security properties must be valid for all states which a protocol can reach during its execution, based on defined transition rules [17–19]. Furthermore, it is assumed, that cryptographic primitives are true, i.e. messages can only be decrypted if the adversary gets access to the key. **Secrecy**, as communication protocol's security property, is reached when a term (e.g. session key) cannot become part of adversary knowledge in all reachable states of protocol.

Authentication security property, as assurance of communication parties identities, can be described based on the following properties: Aliveness, Weak Agreement, Non-injective Agreement, and (Injective) Agreement [18,20]. Table 3 lists formal definitions of those authentication properties, based on Lowe et al. [20].

2.2. Common attacks

Security properties of a protocol can be compromised by different types of attacks. Nowadays attacks in the IoT domain can affect almost everyone due to the huge distribution of IoT devices as e.g. Smart Phones. Assets included in smart environments can operate with sensitive data and directly influence safety of people – e.g. life and health devices, access control devices. In order to protect these assets, local area and web connections have to be considered. n Requirement

Common attacks identified in IoT domain [17,21] are described in the following. Additionally, Table 4 shows which security properties of the extended C.I.A. triangle are violated within the described attacks.

- **Eavesdropping**: is a basic attack which can be applied to each protocol. During this type of attack an adversary only eavesdrops the communication channel and the messages between the communica-

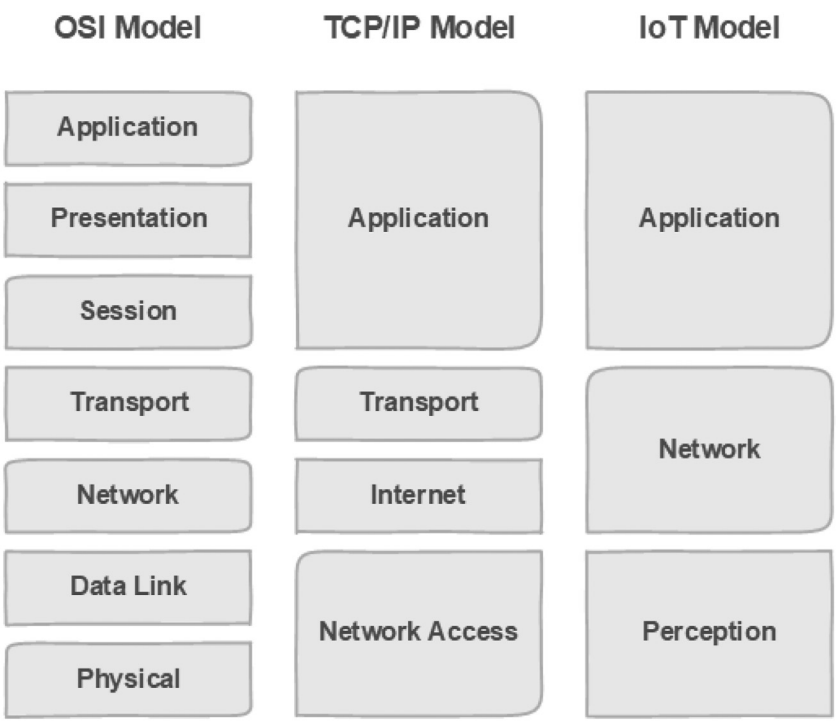

Fig. 2. OSI, TCP/IP and IoT reference models.

tion partners. Although with limited influence, eavesdropping can be a prerequisite for several other, presumably more sophisticated, attacks. Encryption can be used in order to prevent this type of attack.

- **Replay attack**: The adversary records the messages or pieces of the message and uses them at a later time. It contains two cases, one where the original message can pass as expected by the agent and another case, where the original message is prevented from arriving. In order to avoid this attack, one can use nonces or timestamps to show that the messages are generated during the time period the protocol is running.
- **Plaintext recovery**: The attacker is able to recover the plaintext.
- **Modification attack**: The content of the message is changed. In order to do so, the adversary does not necessarily need to know the exact content of the message. It is sufficient to flip some bits of a message which contain a session key. In case the messages are not integrated or their fields are not redundant, the message might yield another session key. Encryption is no sufficient prevention method for this type of attack.
- **Black Hole attack**: This attack is a frame dropping attack. A node under the influence of the attacker called Black Hole Node, silently drops application frames, when it is expected to forward them [22].
- **Impersonation attack**: In this attack an adversary obscures as a legitimate party in a system.
- **Sybil attack**: An attacker forges multiple identities.
- **Man-in-the-middle attack**: In this attack an adversary sniffs or even modifies or deletes the transmitted messages between two parties in order to make them believe, that they are directly communicating with each other.
- **Privileged Insider attack**: This attack is performed by a legal and authorized entity, that has rights to access confidential information of its organization. Then, security data is stolen or fraudulent data is injected in the system.
- **Battery Exhaustion attack**: In order to communicate with the application server field devices connect from time to time with the application server. Depending on the protocol there are different modes to do so, e.g. wakeup times. By manipulating such wakeup times frequently and thereby increasing the power consumption of the sensor, the battery gets exhausted [23].
- **Denial of Service**: DoS attacks take advantage of protocols intended for establishment and authentication of communication, by generating a lot of requests in order to disable services. In practice they usually aim at servers. It seems to be impossible to completely prevent this type of attacks, however, there are examples how to reduce the impact of such attacks, see e.g. [24].

2.3. Protocols and standards overview

Although described security properties and attacks are applicable to numerous protocols, this paper focuses on IoT protocols, since there is

Table 2
Critical information security properties [15].

Property	Description
Confidentiality	The information is protected from disclosure or exposure by unauthorised users – only authorized users have the rights and privileges to access information.
Integrity	The information is whole, complete and uncorrupted – changes shall not happen unnoticed.
Availability	Authorized users are able to access information in a required format without interference and obstruction.
Accuracy	The information is free from mistakes or errors and it has the value the user expects.
Authenticity	The information is in the same state after having been created, placed, stored or transferred.

Table 3
Authentication properties [18,20].

Property	Definition
Aliveness	A protocol guarantees to an initiator *Aaliveness* of another agent *B* if, whenever *A* (acting as initiator) completes a run of a protocol, apparently with responder *B*, then *B* has previously been running the protocol.
Weak Agreement	A protocol guarantees to an initiator *A weak agreement* with another agent *B* if, whenever *A* (acting as initiator) completes a run of a protocol apparently with responder *B*, then *B* has previously been running the protocol apparently with *A*.
Non-injective Agreement	A protocol guarantees to an initiator *A non-injective agreement* with a responder *B* on a set of data items *ds* (where *ds* is a set of variables appearing in the protocol description) if, whenever *A* (acting as initiator) completes a run of the protocol, apparently with responder *B*, then *B* has previously been running the protocol apparently with *A*, *B* was acting as responder in this run, and the two agents agreed on the data values corresponding to all the variables in *ds*.
Injective Agreement	A protocol guarantees to an initiator *A non-injective agreement* with a responder *B* on a set of data items *ds* if whenever *A* (acting as initiator) completes a run of the protocol apparently with responder *B*, then *B* has previously been running the protocol apparently with *A*, *B* was acting as responder in this run, the two agents agreed on the data values corresponding to all the variables in *ds*, and each such run of *A* corresponds to a unique run of *B*.

Table 4
Attacks violating security properties.

Attack	Violated properties
Eavesdropping/Communication sniffing	Confidentiality
Replay attack	Confidentiality, Integrity, Authenticity
Plaintext recovery	Confidentiality
Modification attack	Integrity, Accuracy, Authenticity
Black Hole attack	Integrity, Availability, Accuracy
Impersonation attack	Confidentiality, Authenticity
Sybil attack	Authenticity
Man-in-the-middle attack	Confidentiality, Integrity, Authenticity
Privileged Insider attack	Confidentiality, Integrity, Accuracy
Battery exhaustion attack	Availability
Denial of service	Availability

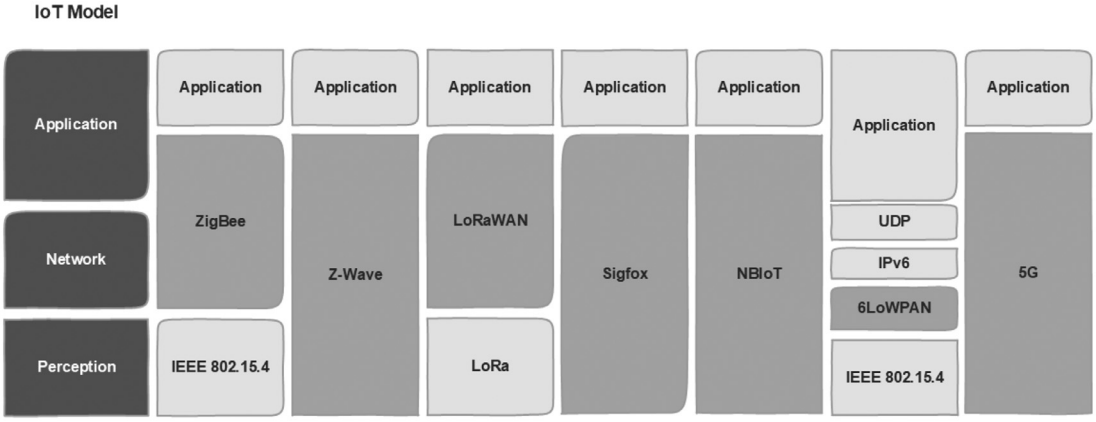

Fig. 3. IoT protocols in the protocol stack.

a great expansion of IoT devices usage in different environments and expansion of security and privacy related issues within.

Fig. 3 presents some commonly used IoT protocols in the protocol stack as an first overview. A more detailed protocol stack for the protocols is given in Fig. 4 and Fig. 5.

This paper describes in detail IoT protocols that, to the best of our knowledge, are part of Formal Verification related research in literature so far (see Table 5 for an overview of technical details).

ZigBee – ZigBee[1] was developed by ZigBee Alliance, in 2001 and first launched in 2003. It was updated in 2007 to ZigBee PRO, also known as Zigbee 2007, supporting backward compatibility. ZigBee PRO is afterward updated in 2015, having ZigBee spec. 05-3474-21 and IEEE 802.15.4–2011 as the last versions.

Zigbee is typically used in low data rate applications that require long battery life and secure networking, because it includes symmetric encryption. It is mostly used for Smart Home IoT devices, such as light bulbs, switches, locks, motion sensors and thermostats. ZigBee is an open IoT protocol based on the IEEE 802.15.4 standard[2] as a physical and data link layer. ZigBee protocol mostly corresponds to the network layer and the application layer. The application layer can be adapted, which enables the implementation of concrete specifications of different scenarios, as for example the widely used ZigBee LightLink (i.e. Philiphs Hue6) or ZigBee Green Power. The protocol uses three topologies – star, tree and mesh in which a maximum of 65k nodes are supported. The protocol supports operation in the three ISM bands, 913 MHz (North America), 868 MHz (Europe) and 2.4 GHz (world-wide). The devices range is limited to a maximum of 100 m and the maximum data rate is 250 kpbs.

Zigbee security mechanisms – Since the first version of ZigBee, many efficient and secure enhancements have been done to the standard. Although ZigBee has a few built-in security services and features, its applications are still vulnerable to network attacks, e.g. sniffing the network key [25]. ZigBee protocol utilizes an authenticated encryption algorithm designed to provide both authentication (integrity) and

[1] https://zigbeealliance.org/solution/zigbee/.

[2] https://standards.ieee.org/standard/802_15_4-2015-Cor1-2018.html.

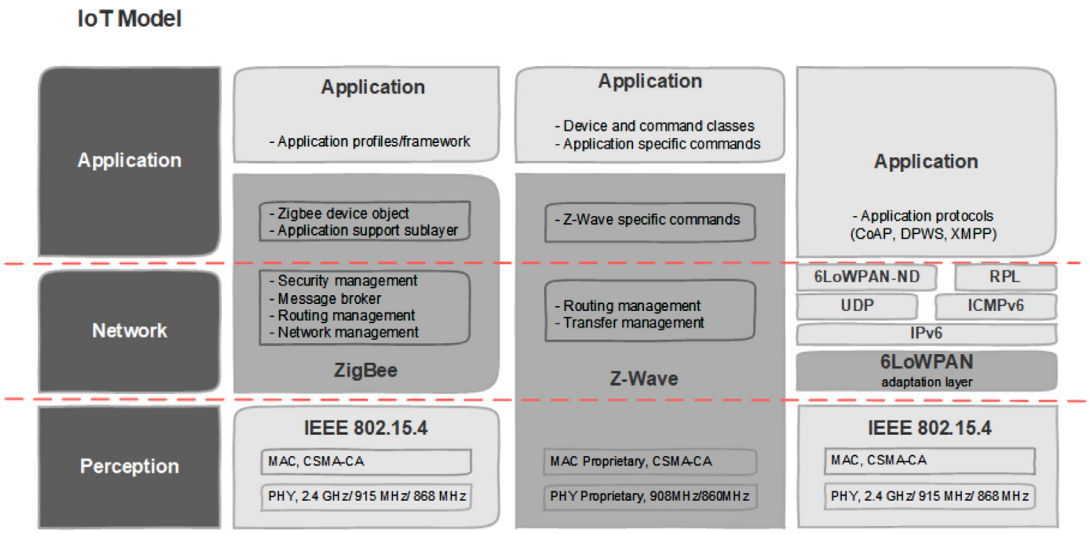

Fig. 4. Detailed protocol stack for ZigBee, Z-Wave and 6LoWPAN.

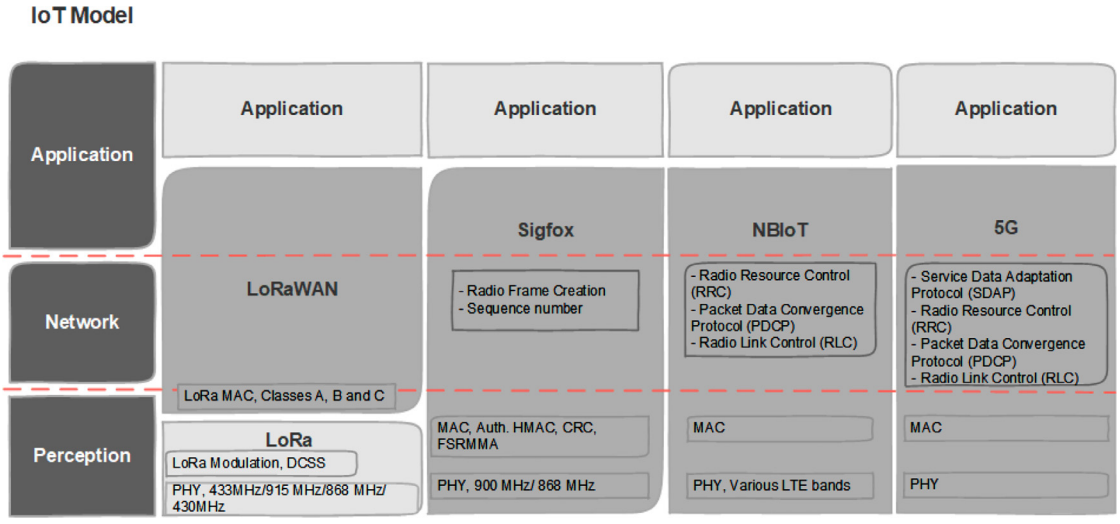

Fig. 5. Detailed protocol stack for LoRaWAN, Sigfox, NBIoT and 5G.

Table 5
IoT protocols' technical details.

Protocol	Nominal Range Limit	Typical Data Rate	Spectrum	Power usage	Standard	Alliance	Year first launched
ZigBee	Local (<100m)	250 kbps	2.4 GHz	Low (<100mW)	ZigBee spec. 05-3474-21, IEEE 802.15.4-2011	ZigBee Alliance	2003
Z-Wave	Local (<100m)	40–100 kbps	900 MHz unlicensed	Low (<100mW)	ITU-T G.9959	Z-Wave Alliance	2003
Bluetooth EDR	Local (<100m)	2 Mbps	2.4 GHz	Medium (<1W)	IEEE 802.15.1	Bluetooth Special Interest Group	1999
Bluetooth LE	Local (<100m)	1 Mbps	2.4 GHz	Low (<100mW)	IEEE 802.15.1	Bluetooth Special Interest Group	2011
LoRaWAN	Metro (>10km)	<50 kbps	900 MHz unlicensed	Low (<100mW)	Proprietary	LoRa Alliance	2015
Sigfox	Metro (>10km)	<100 bps	900 MHz unlicensed	Low (<100mW)	Proprietary	Sigfox company	2009
NB-IoT	Metro (>10km)	<250 kbps	900 MHz	Low (<100mW)	3GPP Release 13	3GPP	2016
6LoWPAN	Local (<100m)	250 kbps	2.4 GHz	Low (<100mW)	IETF/RFC 4944, IEEE 802.15.4	6LoWPAN IETF WG	2007
LTE	Metro (>30km)	>100 Mbit/s	Licensed cellular	Band dependant	3GPP Release 8 and 9	GSMA - Cellular Carriers	2010
5G	Metro (>30km)	<10 Gbps	Licensed cellular	Band dependant	3GPP 5G	3GPP ITU-R	2018

confidentiality, based on 128-bit symmetric encryption. Zigbee security architecture relies on the correct management of symmetric keys and the correct implementation of methods and security policies. The basic mechanism to ensure the mentioned security properties is the adequate protection of the used keys. In practical implementations the initial installation of the key, as well as in the processing of the security information, is based on trust. That makes the key management process the most important vulnerability of ZigBee implementations (see e.g. [3,26]).

Z-Wave – Unlike ZigBee, Z-Wave[3] was proprietory for a long time before becoming a standard. Its development is controlled by the Z-Wave Alliance which involves over 600 companies including large companies in the IoT sector like Siemens or Huawei. It was developed in 2001, and launched in 2003. It allows wireless control of residential appliances and other devices, such as lighting control, security systems, thermostats, windows, locks, swimming pools and garage door openers. Z-Wave Plus – the newest update of the protocol, was developed in 2013. It adds improvements such as better battery life as well as wireless range. It utilizes mesh architecture and supports operation in the 900 MHz unlicensed frequency band. The signal range is up to 100 m and the data rate range is $40 - 100$ kpbs.

Z-Wave security mechanisms – The newest version of the Z-Wave protocol utilizes an encryption algorithm designed to provide message integrity, confidentiality and data freshness, based on 128-bit AES symmetric encryption and message authentication code (MAC) and called Security 2 (S2). However, researchers found that, while it is difficult to break Z-Wave's S2, it is not difficult to downgrade the S2 protocol back to the original version, making any Z-Wave smart device vulnerable to attacks [3].

Bluetooth – Bluetooth protocol[4], launched in 1999, is a standard for short-range wireless communications and data exchange using short-wavelength UHF radio waves. It is managed by the Bluetooth Special Interest Group (SIG) consisting of more than 30.000 member companies in the areas of telecommunications, computing, networking and consumer electronics. Bluetooth was standardized as IEEE 802.15.1[5,6] but does not maintain the standard any more. The Bluetooth SIG developed the specification and protects the trademark. In order to mark a device as Bluetooth, the manufacturer has to meet Bluetooth SIG standards. Bluetooth is a standard wire-replacement communication protocol primarily designed for low power consumption with a short range, used, for example, for sending and receiving digital voice and data over cell phone headsets as well as wireless keyboards, mice and game controllers. It exists in numerous products such as telephones, speakers, tablets, media players, robotics systems, laptops, and console gaming equipment as well as some high definition headsets, modems, hearing aids and even watches. Bluetooth protocol utilizes master/slave architecture and supports operation in the 2.4 GHz ISM band, with maximum signal range of up to 100 m and the data rate up to 2 Mpbs (Bluetooth Enhance Data Rate - EDR).

Bluetooth Low Energy – Bluetooth Low Energy[7] is intended to provide considerably reduced power consumption and cost while maintaining a similar communication range as the classical Bluetooth. It appeared with Bluetooth version 4.0 and is not backward-compatible with the previous ones often called "classic" Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR) protocols. Bluetooth Low Energy uses the same 2.4 GHz radio frequencies as classic Bluetooth, with maximum signal range of up to 100 m and the data rate up to 1 Mpbs.

Bluetooth security mechanisms – Both Bluetooth and Bluetooth LE standards have specified basic security properties: authentication - verifying the identity of communicating devices, confidentiality (encryption) - preventing information compromise and authorization - allowing the control of resources by ensuring that a device is authorized to use a service before permitting it to do so. Bluetooth use E0/SAFER+ (secure and fast encryption routine) encryption method, while Bluetooth LE use AES encryption. Most recently discovered vulnerabilities include the Bluetooth pairing process [27] and the key negotiation [28].

LoRaWAN – LoRaWAN[8] is a Low Power, Wide Area (LPWA) networking protocol designed for wireless communication and targeting bi-directional, end-to-end security, mobility and localization services. Its specification was developed in 2015 and maintained by the LoRa Alliance, an open, nonprofit association with more than 500 members. LoRaWAN protocol provides low power consumption and supports large networks with millions of devices. It is primarily used in industrial environments, but it is also used in home automation applications. LoRaWAN defines the communication protocol and system architecture for the network and is built upon the LoRa physical layer that enables the long-range communication link. Its network architecture follows a star-of-stars topology, where gateways relay messages between end-devices and a central network server. The gateways are connected to the network server via standard IP connections. Moreover, the gateways act as transparent bridges, simply converting RF packets to IP packets and vice-versa. LoRaWAN (and LoRa) permits long-range connectivity for Internet of Things (IoT) devices in different types of industries. It uses unlicensed sub-gigahertz radio frequency bands like 433 MHz, 868 MHz (Europe) and 915 MHz (North America). It enables long-range transmissions (more than 10 km in rural areas) with low power consumption. The maximum data rate is 50 kpbs.

LoRaWAN security mechanisms – Similar to previously described communication protocols, LoRaWAN standard defines three security properties - confidentiality, authentication and integrity. LoRaWAN uses 128 bit key AES encryption, with two layers of cryptography - end device-to-network and end-to-end encryption. For providing authentication and integrity of packets to the network server and end-to-end encryption to the application server AES algorithms are used. Keys can be Activated By Personalization (ABP) on the production line or during commissioning. Another possibility is to use the Over-The-Air Activated (OTAA) in the field. The last one even allows devices to be re-keyed.

Sigfox – Sigfox[9] is a narrowband LPWAN protocol, like LoRaWAN. It was developed in 2009 in Toulouse, France, by Sigfox network operator. Originally, Sigfox was unidirectional and supported only uplink communication, the communication from the sensor network. In later versions a downlink channel has become available. Unlike LoRaWAN, in does not require proprietary hardware, but uses several hardware vendors and a single managed network infrastructure. Due to strict limitations in terms of throughput and utilization, it is intended for systems sending small and infrequent amounts of data. It is typically used in devices like alarm systems, simple power meters, and environmental sensors.

It uses the unlicensed ISM bands, like 868 MHz (Europe) and 902 MHz (North America), for a proprietary protocol, and allows a throughput of up to 100 bps uplink and 600 bps downlink.

Sigfox security mechanisms – Sigfox is one of the most secure LPWAN technologies, where security is addressed through a systematic process [29]. Sigfox devices have built-in behaviour, and predominantly operate in offline mode. Devices use broadcast radio messages to communicate, and it is not possible to access an end point through internet maliciously. During manufacturing each device is given a unique symmetrical authentication key, providing limited impact on the network if one device is compromised. For each sending or receiving message a cryptographic token is computed using the authentication key – this token is used for authentication of sender and integrity of message. Unlike LoRaWAN, it is not possible to use over the air activation (OTA) mech-

[3] https://www.itu.int/rec/T-REC-G.9959.
[4] https://www.bluetooth.com/specifications/bluetooth-core-specification/.
[5] https://standards.ieee.org/standard/802_15_1-2005.html.
[6] http://www.ieee802.org/15/pub/TG1.html.
[7] https://www.bluetooth.com/specifications/.
[8] https://lora-alliance.org/lorawan-for-developers.
[9] https://build.sigfox.com/sigfox-device-radio-specifications.

anism, because Sigfox devices are not IP addressable. Decision about using encryption is left to user.

NB-IoT – Narrowband IoT[10], another LPWAN protocol, uses cellular telecommunication band to connect wide range of devices. It is based on LTE (long term evolution) protocol standardized by 3GPP in 2016. The functionality of LTE is reduced in NB-IoT to its minimum, while enhancing the functionalities as required by IoT applications [29]. NB-IoT technology supports devices that can be used for personal use, public use, IoT applications and industrial use.

It uses 700 MHz, 800 MHz and 900 MHz frequency bands and provides 200 kbps data rate.

NB-IoT security mechanisms – NB-IoT inherits LTE's authentication and encryption, where security mechanisms aim at three different layers – perception layer, transmission layer and application layer. Since the perception layer is susceptible to both active and passive attacks, including both altering the data, and simple monitoring the network traffic, the encryption mechanisms are provided. In transmission and application layers, the main security requirements are identification and processing of massive heterogeneous data, integrity and authentication of data and access control [29].

6LoWPAN – 6LoWPAN protocol[11] was developed by 6LoWPAN IETF working group, in 2007. 6LoWPAN is an acronym of IPv6 over Low-Power Wireless Personal Area Networks. The 6LoWPAN concept originated from the idea that low-power devices with limited processing capabilities should be able to use the IP protocol and participate in the Internet of Things. It is used, for example, in automation and entertainment applications in home, office and factory environments. It can also be in use on the Smart Grid enabling Smart Meters and other devices to build a micro mesh network before sending the data back to the billing system using the IPv6 backbone. The 6LoWPAN group has defined encapsulation and header compression mechanisms that allow IPv6 packets to be sent and received over IEEE 802.15.4[12] based networks - IEEE 802.15.4 devices provide sensing communication-ability in the wireless domain. The protocol utilizes a time synchronized, self-organizing, and self-healing mesh architecture. The protocol supports operation in the 2.4 GHz ISM band world-wide, 913 MHz in North America and 868 MHz in Europe using IEEE 802.15.4 standard radios. The signal range is up to 100 m and the maximum data rate is 250 kpbs.

6LoWPAN security mechanisms – The security of 6LoWPAN widely depends on the 802.15.4 security sublayer - it uses 128 bit symmetric encryption and provides confidentiality and authentication. Most common vulnerabilities of 6LoWPAN include key distribution, DoS, jamming and physical security of nodes (see e.g. [30,31]).

LTE – In general, Long-Term Evolution (LTE) is a standard for wireless broadband communication for mobile devices and data terminals, based on the GSM/EDGE and UMTS/HSPA technologies. The standard is developed by the 3GPP and specified in Release 8 document series[13], with minor enhancements described in Release 9[14]. LTE is also known as 4G LTE, Advance 4G and 3.95G, since it does not meet the technical criteria of a 4G wireless service, as specified in the 3GPP Release 8 and 9 document series. It provides rates of 300 Mbps for downlink and 75 Mbps for uplink and a transmission range up to 100 km in the radio network.

LTE security mechanisms – LTE uses symmetric key cryptography in order to provide confidentiality and integrity. Encryption is applicable to both user traffic and signaling traffic. LTE vulnerabilities reside in both the protocol itself and vendors' implementation of LTE in their devices, and they are mostly related to key agreement process (see e.g. [32]).

5G – 5G is the fifth generation cellular network technology. The standardization requirements are defined in ITU IMT-2020. The industry association 3GPP defines any system using "5G NR" (5G New Radio) software as "5G". It follows 2G (GSM), 3G (UMTS) and 4G (LTE, LTE Advanced Pro). The ITU-R has defined three main uses for 5G:

(i) Enhanced Mobile Broadband (eMBB) - uses 5G as a progression from 4G LTE mobile broadband services, with faster connections, higher throughput and more capacity.
(ii) Ultra Reliable Low Latency Communications (URLLC) - should use the network for mission critical applications that requires uninterrupted and robust data exchange (deployment expected after 2021).
(iii) Massive Machine Type Communications (mMTC) - should be used to connect to a large number of low power, low cost devices, which have high scalability and increased battery lifetime, in a wide area (deployment expected after 2021).

Three key frequency ranges to support 5G use cases are: 1 GHz – supports widespread coverage across urban, suburban and rural areas and helps to support IoT services, 1 – 6 GHz, – a good mixture of coverage and capacity benefits, expected to operate within the 3.3 – 3.8 GHz and above 6 GHz, which is needed to meet the ultra-high broadband speeds envisioned for 5G, expected to operate in 26 GHz and/or 28 GHz bands. 5G tops out at 10 Gbps data rate, which is a hundred times faster than LTE (4G) speed of 100 Mbps.

5G security mechanisms – 5G[15] announces the beginning of a new era of network security with the introduction of IMSI encryption - all traffic data which is sent over 5G radio network should be encrypted, with protected integrity and subject to mutual authentication. Security of 5G networks is at this moment an active research topic.

3. Formal verification methods and tools

Formal Verification is a promising method to provide security guarantees by mathematically ascertaining the correctness of designs using a diverse set of mathematical and logical methods. These methods are particularly useful in order to get quantitative statements about safety and security properties of digital systems (see e.g. [5,33]). Two types of *Formal Method tools* can be distinguished:

- Model checkers
- Theorem provers.

Model checkers exhaustively and automatically verify a system's model in its model's state space with respect to a given specification. Theorem provers often require human expertise to guide a proof of correctness by providing the design and specification characteristics as algebraic constraints or theorems [5,6]. While model checkers usually are more convenient to use and targeted to a specific problem domain and verification of properties in this field, they are necessarily limited in the range of problems that they can handle.

Whereas some of the tools as AVISPA, ProVerif, Scyther and Tamarin have their focus on *security protocols* (see Fig. 6), there are also tools for statistical and/or *probabilistic checking* as UPPAAL and PRISM (see Fig. 7). While in case of verifying security protocols the main goal usually lies in the verification or falsification of security properties as secrecy and authentication, probabilistic model checking is used to analyze quantitative properties of systems which exhibit stochastic behavior, see [34,35]. This includes systems and well-known communication protocols as FireWire and Bluetooth. Moreover, probabilistic model checking is especially useful for considering communication systems, unpredictable characteristics of which, such as message delays or times to failure, are best represented in probabilistic fashion.

[10] https://www.3gpp.org/news-events/1785-nb_iot_complete.
[11] https://tools.ietf.org/html/rfc4944.
[12] https://standards.ieee.org/standard/802_15_4-2015-Cor1-2018.html.
[13] https://www.3gpp.org/specifications/releases/72-release-8.
[14] https://www.3gpp.org/specifications/releases/71-release-9.
[15] https://www.gsma.com/spectrum/wp-content/uploads/2019/07/5G-Spectrum-Positions.pdf.

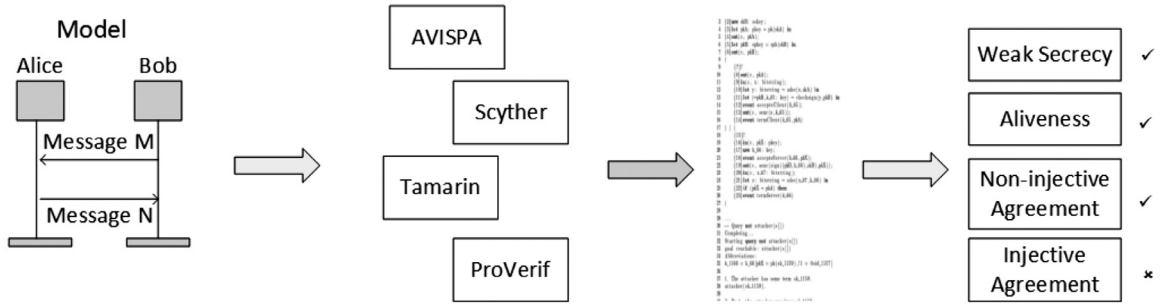

Fig. 6. Security protocol formal verification tools.

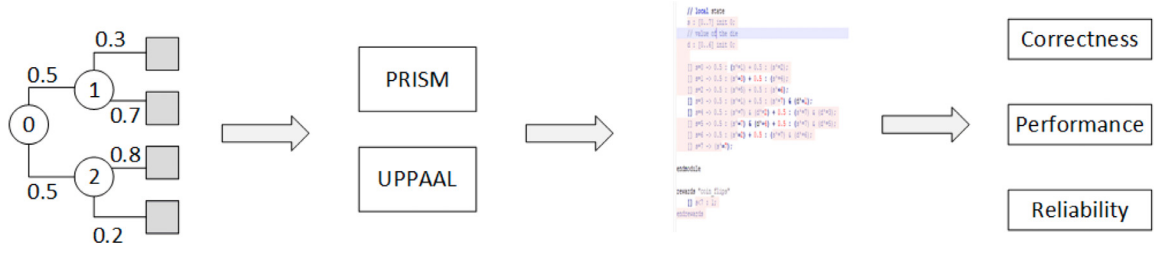

Fig. 7. Probabilistic/statistical model checkers.

Table 6
Tool for Model Checking, information from 12.02.2020, *see provided references.

Tool	Operating system			last version	Licence
	Windows	Linux	Mac		
Scyther	√	√	√	March 2016 (Windows), Mai 2014 (Linux, Mac)	GNU GPL 2
Tamarin	✗	√	√	May 2017	GNU GPL 3
AVISPA	✗	√	√	version 1.1 (June 2006)	special*
ProVerif	√	√	√	version 2.0.0 (2018)	GNU GPL 2, Windows binaries BSD
PRISM	√	√	√	version 4.5 (April 2019)	GNU GPL 2
UPPAAL	√	√	√	official release 4.0.14 (November 2019), snapshot 4.1.24 (November 2019)	free for non-commercial applications in academia only
Event-B/Rodin	√	√	√	version 3.4 (February 2018)	Eclipse Public Licence-v.1.0

While most of the tools are designed for model checking only, there are also tools as Rodin and Tamarin which additionally provide the possibility of manual guidance and therefore act as theorem prover.

In order to apply such a tool, first a model as input to the model checker has to be created according to the specification or a given implementation. Since it is in general not possible to verify the whole protocol, the most important parts according to the problem statement (e.g. functional check, check of authentication property) are modeled. Especially in the case of considering security protocols, a key feature of the model checkers comes into play. This key feature is the *number of sessions* involved. A session is a single (potentially partial) execution of a protocol. This is an important term, since attacks often interleave different sessions. However, the drawback is, that i.e. the secrecy problem is undecidable if the number of sessions and nounces is unbounded. Due to the decidability of security for bounded sessions, there are a lot of bounded-session model checkers being of practical significance, since most of the protocols require only a few sessions. These model checkers include The Constraint-Logic-based Attack Searcher (CL-Atse), the On-the-Fly Model Checker (OFMC) and the SAT-based Model Checker SAT-MC, see [33]. Furthermore, there are also unbounded-session model checkers based on abstractions as ProVerif and on heuristics, as Scyther [33].

This section provides an overview of different tools used for verifying IoT protocols considered in this paper. A general introduction to model checkers is given in e.g. [6,17,18,33]. Some basic properties for different widely used model checkers are summarized in Table 6. Moreover, additional information is provided for security protocol model checkers in Table 7 and for probabilistic/statistical model checkers in Table 8. An application of three security protocol model checkers, namely Scyther, Tamarin and ProVerif to the LTE protocol and a further comparison of these three tools is given in [36].

For a more detailed comparison between the probabilistic model checkers UPPAAL and PRISM see [37]. The authors therein investigate the tools based on a benchmark case study.

3.1. AVISPA

AVISPA[16] stands for Automated Validation of Internet Security Protocols and Application. It is a push-button tool for the automated validation of Internet *security-sensitive protocols and applications* (see [38–40]). AVISPA is available by filling in a form on the AVISPA project homepage. The main advantage of the AVISPA tool suite is, that different verification techniques can be performed on the same protocol specification. For specification of protocols in AVISPA the High-Level Protocol Specification Language (HLPSL) is used. There is even a tool called SPAN, to help protocol developers writing these specifications by interactively building Message Sequence Charts of the protocol execution. HLPSL is based on roles, that is basic roles for representing each participant role and composition of roles for representing scenarios of

[16] http://www.avispa-project.org/.

Table 7
Security protocol model checkers.

	AVISPA	ProVerif	Scyther	Tamarin
Input language	HLPSL	Applied Pi Calculus	C/Java-like syntax	based on multiset rewriting rules
Classical properties (e.g. secrecy, agreement, aliveness)	√	√	√	√
Unbounded verification	√[16]	√	√	√
Protocol specification	via roles	via Prolog rules	via roles	multiset rewriting
Attack finding and visualisation	√	√	√	√
Model Checker	√	√	√	√
Theorem Prover	✗	✗	✗	√
GUI	√	✗	√	√

Table 8
Probabilistic/Statistical model checker.

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

basic roles. The HLPSL specification is then translated into an Intermediate Format which is used by the various verification tools embedded in AVISPA as:

- On-the-Fly model checker (OFMC), which supports the specification of algebraic properties of cryptographic operators and typed and untyped protocols' models. Therefore, it performs protocol falsification and bounded verification.
- Constraint-Logic-based Attack Searcher (CL-AtSE), which supports type-flaw detection and handles associativity of message concatenation. It is also open to extensions for handling algebraic properties of cryptographic operators.
- SAT-based model checker (SATMC), which is especially useful for detecting violations of security properties.
- Tree Automate based Automatic Approximations for the Analysis of Security Protocols (TA4SP), which also can show for secrecy properties whether a protocol is flawed (by under-approximation) or is safe for any number of sessions (by over-approximation).

The main advantage of this tool suite is, its modular and provides an expressive formal language for specifying security protocols and properties. Furthermore, by integrating different back-ends, it implements a variety of automatic analysis techniques ranging from protocol falsification by finding attacks on the input protocol to the abstraction of verification for finite and infinite numbers of sessions.

3.2. Event-B and Rodin

Event-B[17] is a formal method for *system level modeling and analysis*. Is is an evolution of the B-Method developed by Jean-Raymond Abrial (see [41,42]). The key features of Event-B are:

- use of the set theory as a modeling notation
- use of refinements to represent systems at different abstraction levels
- use of mathematical proof to verify consistency between refinement levels.

With Event-B systems, whose components can be modeled as discrete transition system, can be formalized. There are two basic constructs: contexts and machines. While contexts specify the static part of a model and isolate parameters of a formal model and their properties which are assumed to hold for all instances, machines specify the dynamic part of a model, encapsulate a transition system with the state being specified by a set of variables and contain transitions modeled by a set of guarded events.

Models can be developed gradually via mechanisms such as context extension and machine refinement, which enable users to develop target systems from their abstract specifications and add more and more implementation details.

The Rodin Platform[18] is an Eclipse-based IDE for Event-B providing effective support for refinement and mathematical proof (see [43]). It is a set of tools to aid in the design and analysis of Event-B models. Special properties of the Rodin tool are, that it spans abstraction levels with a refinement methodology and that it contains both, a theorem prover and a model checker in the same tool. This is a huge advantage and enables the represention of many levels of abstraction systematically. Additionally it provides the opportunity to use both, model checkers (especially suited for straightforward verification) and theorem provers (enabling verification using higher level insights), see [6].

3.3. PRISM

PRISM[19] is a *probabilistic model checker developed* at the University of Birmingham for the *quantitative analysis of system properties exhibiting stochastic behavior* (see [34,44,45]). PRISM supports several probabilistic models as

- discrete-time Markov chains (DTMCs)
- continuous-time Markov chains (CTMCs)
- Markov decision processes (MDPs)
- probabilistic automata (PAs)
- probabilistic timed automata (PTAs)

[17] http://www.event-b.org/.

[18] http://www.event-b.org/install.html.

[19] https://www.prismmodelchecker.org/.

and extensions of these models with costs and rewards. Furthermore, it includes engines for quantitative abstraction-refinement [46] and statistical model checking [47,48]. PRISM has been used for *quantitative verification* in a wide spectrum of application domains, ranging from wireless communication protocols to quantum cryptography and to systems biology. Beside the described models, its functionalities include model checking for a wide range of properties. These properties are expressed in the PRISM language, a simple, state-based language based on the Reactive Modules formalism of Alur and Henzinger [49]. There are several model checking engines, including both, symbolic (BDD-based) and explicit-state models. Furthermore, a variety of probabilistic verification techniques, such as symmetric reduction and quantitative abstraction refinement is available. Moreover, it includes a discrete-event simulator, supporting statistical model checking methods. PRISM supports the generation of optimal adversaries/strategies, a GUI including a model editor, simulator and graphing and additionally a command-line tool. It also includes a benchmark suite with probabilistic models and associated properties.

3.4. ProvVerif

ProVerif[20] is a command line tool for automatically analyzing the *security of cryptographic protocols* based on a simple representation of the protocol by Prolog rules (see [50–53]). In ProVerif an algorithm is implemented, which determines efficiently if a fact can be proved from the Prolog rules or not. Due to the use of unification, the problem of state space explosion is avoided. Another advantage is, that the number of runs of the protocol does not need to be limited. ProVerif uses a typed variant of the pi calculus as input language and is able to proof reachability properties, correspondence assertions and observational equivalence, inducing that it is especially useful for the analysis of secrecy and authentication properties. Moreover, properties as privacy traceability and verifiability can also be considered with ProVerif. The tool can also be used for *attack reconstruction* since it tries to reconstruct an execution trace that falsifies the desired property if a property cannot be proved. Due to the used approximations, false attacks against a protocol might occur. However, according to [50] these cases are rare.

3.5. Scyther

The Scyther tool[21] was developed for the *automated verification or falsification of security protocols* (see [54]). The tool was developed under the perfect cryptography assumption, i.e. it assumes that all cryptographic functions are perfect and the adversary learns nothing from an encrypted message unless he knows the decryption key. Therefore, the tool can be used to detect problems arising from the way the protocol was constructed by using the backward search algorithm based on a symbolic representation of sets of protocol runs. Its input language is loosely based on a C/Java-like syntax. A protocol model in Scyther consists of a set of roles. Each role is a list of events, which can be communication events (send and receive) or match events. To enforce the structures and types of the arguments, pattern matching is used in all events [36]. It has to be emphasized, that there are features including the possibility of unbounded verification with guaranteed termination, analysis of infinite sets of traces in terms of patterns and also support for multi-protocol analysis. The model checking tool is based on a pattern refinement algorithm and provides concise representations of (infinite) set of traces. Therefore it assists in the analysis of classes of attacks and possible protocol behaviors. Moreover, it can be used to prove correctness for an unbounded number of protocol sessions.

3.6. Tamarin

The Tamarin tool[22] (see [55,56]) was developed for the *symbolic modeling and analysis of security protocols*. It generalizes the backwards search used by Scyther and enables:

- protocol specification by an expressive language based on multiset rewriting rules
- property specification in a guarded fragment of first-order logic allowing quantification over messages and timepoints
- reasoning modulo equational theories.

In practice that means, that the tool can handle *protocols with non-monotonic mutable global state* and *complex control flow such as loop*. Moreover, it can manage complex security models such as the eCK model [57] for key exchange protocols and equational theories such as Diffie-Hellman, bilinear pairings and user-specified subterm-convergent theories. There are two possibilities of constructing a proof in Tamarin, a fully automated mode and an interactive mode. The first one, guides the proof search by combining deduction and equational reasoning with heuristics. However, due to the undecidable nature of most properties in the setting, the tool may not terminate the verification procedure. In case of termination the tool returns either a proof of correctness or a counterexample. Verification is possible for an unbounded number of role instances and fresh values. In case of a counterexample, an attack that violates the stated properties is given. In case the tool does not terminate, it is recommended to use the interactive mode. Therewith, the user can explore the proof states, inspect attack graphs and combine manual proof guidance with automated proof search.

3.7. UPPAAL

Uppaal[23] is a toolbox for *verification of real-time systems* developed by the Department of Information Technology at Uppsala University, Sweden in cooperation with the Department of Computer Science at Aalborg University in Denmark (see [58,59]).

The toolbox is especially useful for systems that can be modeled as a collection of non-deterministic processes with finite control structure and real-valued clocks, communicating through channels or shared variables. Therefore, typical application areas include real-time controllers and communication protocols, in which timing aspects are critical. Uppaal consists of *three main parts*:

- a *description language* for modeling the system behavior as networks of automata extended with clock and data variables. Its language is a non-deterministic guarded command language with data types (e.g. bounded integers, arrays).
- a *simulator* for validation. It enables examination of possible dynamic executions of a system during early design (or modeling) stages. Therefore it provides a cheap mean of fault detection before verification by the model checker.
- a *model-checker* is based on the theory of timed automata. It covers the exhaustive dynamic behavior of the system. It can check invariant and reachability properties by exploring the state-space of a system, i.e. reachability analysis in terms of symbolic states represented by constraints.

The creators of Uppaal claim that two main design criteria are key to Uppaal's efficiency and ease of use. One key point is, that symbolic techniques are used, reducing the problems to efficient manipulation and solving of constraints. Uppaal model checker is able to automatically generate a diagnostic trace explaining why a property is or is not satisfied by a system description. Furthermore, these diagnostic traces can be loaded to the simulator, which enables the visualization and investigation of the trace.

[20] https://prosecco.gforge.inria.fr/personal/bblanche/proverif/.
[21] https://people.cispa.io/cas.cremers/scyther/.
[22] https://tamarin-prover.github.io/manual/tex/tamarin-manual.pdf.
[23] http://www.uppaal.org/.

4. Formal verification of IoT protocols

This section provides detailed information on formal method approaches to detect possible and existing vulnerabilities of the selected protocols. For the analysis common IoT domain protocols have been selected. For each protocol an extensive literature research on approaches with Formal Methods was carried out. To our best knowledge literature focuses mainly on the protocols ZigBee, Bluetooth, LoRaWAN and LTE. For 6LoWPAN the focus is on applying Formal Methods on enhanced schemes, for Z-Wave existing approaches are more a vulnerability analysis than a Formal Method. However, due to the signification of Z-Wave as Smart Home protocol, it is included in that review.

Based on the description of existing work,

- a classification of Formal Method approaches on IoT protocol's in common application fields,
- a review of used tools in common application fields,
- an analysis of considered security properties,

are given.

4.1. Formal verification of protocols overview

This subsection provides a review of tools and protocol's properties checked with Formal Methods. The work is sorted by protocol. First, a short summary of existing approaches on a specific protocol is given. Then, details for each publication including details on considered properties, tools and findings are described.

4.1.1. Zigbee

Formal Verification of ZigBee is considered in several papers. The approaches range from verifying the ZigBee protocol stack [60], formally modeling the ZigBee routing protocol and verifying it using collision avoidance and liveness properties [61], verification secure network authentication [62] to checking protocol specification for exposition of the default key to attackers [63]. Moverover, for ZigBee-2007 optimizing key confidentiality, key recovery (from a compromise case) time and the efficiency of key updates are considered in [64]. A proposed scheme is considered in [65].

Gawanmeh in [60] used `Event-B` to model and verify ZigBee. The focus is on *verifying the ZigBee protocol stack* by providing an embedding of the protocol primitives in Event-B. Concentrating on verifying certain design requirements for the protocol stack, the author models its functional operations and then verify properties related to its specifications. First, the network layer data entity and the network layer management entity are modeled. For the model, it is especially taken into account that ZigBee devices can join a network, leave a network or rejoin a network, whereas ZigBee coordinators and routers can permit devices to join or leave a network, assign logical network addresses and maintain a list of neighboring devices. Moreover, only ZigBee coordinators can establish a new network. For modeling the functional requirement a set of events is provided, where each event checks for the validity of the service for the initiating device. The following operations are modeled: the initialization event, etablishing a new network and joining a network. Then, the following four properties are checked by using the Rodin Platform:

- the devices should connect through a router or a coordinator,
- a coordinator cannot establish a network if one has already been established,
- routers and end devices cannot establish a network,
- a node (device, router or coordinator) cannot join an empty network.

All these properties are successfully discharged.

Rashid et al. in [61] considers a Smart Grid use case. However, for Formal Verification they consider *ZigBee protocol as routing protocol in the home area network*. According to Rashid in Smart Grid a reliable communication between different components is crucial. `Uppaal` model checker is used to formally model the ZigBee routing protocol and verify it using *collision avoidance* and *liveness properties*. ZigBee uses the master-slave strategy for routing purposes, that means the ZigBee coordinator acts as master whereas the smart appliances act as slaves [66]. Therefore, the ZigBee coordinator (master) communicates directly to the smart applicants (slaves). Moreover, it manages the communication between the slaves. That enables a smooth and reliable communication in the network. For the modeling and verification the authors consider the following properties assuming to have a lossless communication medium under consideration (i.e. every sent data will surely be received) and to have two slaves:

- the *collision avoidance property*, which ensures, that there will be no collision between the slaves over the wireless medium
- the *bounded liveness property*, to ensure that the delay of user-to-user communication is bounded
- the *deadlock freeness property*, which means that no collision will occur and the communication delay between users will be bounded

The stated properties are successfully verified with Uppaal model checker. This authenticates the functionality of ZigBee routing protocol.

Nadeem and Gill consider in [62] the verification of ZigBee protocol for *Secure Network Authentication*. There, the formal specifications represent an abstract formal model created in `Event-B` and consider *adding an authenticated user*. The model is successfully verified with Rodin tool. The authors state that more refinement of their models is necessary in order to integrate it into real world systems.

Melaragno et al. in [63] conduct a formal analysis of ZigBee protocol using the `Failures-Divergence Refinement (FDR)` model checker [67] and its Casper interface and the `AVISPA` model checker. Since they check *multiple authentication attack types* – although for a Smart Grid use case – their results are applicable in other use cases too. They detect the first security flaw already through manual inspection of their formal methods. That well-known security flaw is in that part of the protocol where smart-grid-enabled sensors to join the network via the local trust center. For modeling they consider a smart energy device joining the ZigBee network. Moreover, they provide the simulated attackers with knowlegde of the default key. Both, Casper and AVISPA show that the *protocol specification exposed the default key to attackers*. That allows traditional Dolev-Yao attackers to exploit and expose further information, which violates *secrecy* and *authentication* property. Moreover, they detect, that the default key's exposure during the service-discovery process is the critical point of failure.

Yuksel et al. in [64] consider ZigBee-2007. They analyze six different application profiles: Home Automation, Smart Energy, Commercial Building Automation, Personal, Home and Hospital Care, Telecom Applications and Wireless Sensor Applications. For these szenarios, a ZigBee network is modeled as continuous-time Markov chain in `PRISM` in order to *optimize key confidentiality, key recovery time* (from a compromise case) and the *efficiency of key updates*. The authors state, that their work is the first to use a stochastic model for checking ZigBee security services. In detail, they consider key update strategies, especially:

- *Time-based key update*, where the key is updated after a predefined key expire time,
- *Leave-based key update*, where the key is updated when a predefined number of devices has left the network,
- *Join-based key update*, where the key is updated when a predefined number of new devices has joined the network.

Alshahrani et al. propose in [65] a ZigBee based anonymous device-to-device mutual authentication and key exchange scheme for the Smart Home Network. The proposed scheme uses symmetric encryption and performs the agreement on a shared secret session key during communication via a trusted home controller. The authors use `BAN` logic and `AVISPA` to check their proposed scheme, where the focus on *forward secrecy* and secure *anonymous authentication*.

4.1.2. Z-Wave

As opposed to ZigBee, just very few papers consider security for Z-Wave. This is likely due to the fact that it was proprietary for a long time [3]. In [22], the focus is on identifying source and data integrity vulnerabilities of the routing protocol, whereas in [68] a vulnerability discovery on the Z-Wave door locks is performed.

In **Badenhop et al.** [22], although without applying formal verification methods, the Z-Wave routing protocol and its security implications are considered. They perform a *security analysis* on the network *to identify source and data integrity vulnerabilities of the routing protocol*. Their results show that the topology and routes may be modified by an outsider through the exploitation of the blind trust inherent to the routing nodes of the network. Moreover, a *Black Hole attack* is conducted to demonstrate a well-known routing attack that exploits the exposed vulnerabilities. Based on their discoveries, several recommendations are given to enhance the security of the routing protocol.

Fouladi and Ghanoun in [68] state to publish the first public vulnerability research on Z-Wave. In a first step, they analyze the proprietary Z-Wave protocol and uncovered the details of its encryption, authentication and key exchange. Based on that, a tool named `Z-Force` is developed for a low cost Z-Wave packet interception and injection in order to *perform vulnerability discovery on the Z-Wave door locks*. The following scenarios are analyzed:

- *Door lock inclusion into the Z-Wave network for the first time*: encryption key exchange takes place between the controller appliance and the door lock to establish a shared symmetric key.
- *Sending lock/unlock commands to the Z-Wave door lock*: the command is encrypted using the established encryption key and the authentication value is also appended to the frame.
- *Door lock inclusion after controller appliance factory rest*: The factory reset will erase the previously established key from the appliance but the door lock will still hold the old encryption key.

In their analysis they detect an implementation vulnerability in Z-Wave's *key exchange protocol* that could be exploited to take full control of a target Z-Wave door lock by only knowing the home and node IDs of the target device. Both IDs can be identified by observing the Z-Wave network traffic over a short period of time. This is easily possible due to the frequent polling of devices in a Z-Wave network, for example to get information on status or battery level of a device. The detected vulnerability is due to an implementation error in disabling the use of a temporary key after an initial network key exchange during the inclusion of a node to the network. Therewith, it has to be emphasized, that this vulnerability is not due to a flaw in the Z-Wave protocol specification.

4.1.3. Bluetooth

Formal Verification for Bluetooth is mainly applied to the classic Bluetooth but not specifically to Bluetooth Low Energy. Literature covers primarily the first stages of a communication. In [69] Formal Verification of a configuration process in the Bluetooth logical link control and adaption protocol is performed. In [70] the device discovery mechanism is considered to guarantee accurate functioning and verification of the connectivity of Bluetooth devices. In [71] a formal analysis of authentication properties of Bluetooth device pairing by checking the standard device pairing protocol and Simple Pairing protocol is done. A performance analysis of the initialization process is given in [72].

Pek and Bogunovic in [69] give results on Formal Verification of a *configuration process in the Bluetooth logical link control and adaption protocol* (L2CAP) by using `NuSVM` [73], a symbolic model checking framework which enables BDD-based model checking against CTL specifications. L2CAP is in the data link layer and provides higher protocol multiplexing, packet segmentation and reassembly and the conveying of quality of service information. The configuration process is performed before any data transfer between remote devices has happened. The authors consider two models, an ideal one which describes lossless message transfer and a real one, where they have introduced lossy channel and the concept of timer through timeout indication. Different properties of the models are checked, such as freedom of deadlock, if both peers can always finally reach end state and it is verified that either end or start state is reached. Moreover, the authors check what happens if a message is in the channel and whether it will remain in the channel or will be received by a remote device. Results show, that it remains in the channel for the ideal model, but is received by the other one. In the second model with lossy channel, the authors also model timer RTX, which generates signal timeout indication and that if a message is lost in transfer from device A to device B, an upper entity receives a timeout indication. They state that not all aspects of the protocol can be modeled using NuSVM language, but they are able to check the most interesting ones from their point of view.

Arry and Kaur in [70] consider the *device discovery mechanism* with Formal Verification by using `Uppaal`. Formal Verification is used to guarantee the accurate functioning and the verification of the connectivity of the Bluetooth devices creating short-range wireless ad hoc networks based on the frequency hopping technique. Therewith, Uppaal is used to create an ad hoc network consisting of Bluetooth transmitter and receiver. By verifying the receivers reply to the transmitter, the connectivity of the transmitter and receiver is confirmed. For checking the data connectivity, a verification is performed by checking the energy level of the receiver. The simulation shows that a high energy level of the receiver shows the acceptance of data, whereas a zero energy level depicts non-acceptance.

Chang and Shmatikov perform in [71] a formal analysis of *authentication properties of a Bluetooth device pairing* with `ProVerif`. In particular they consider *Bluetooth device pairing* which enables two devices to authenticate each other and to establish a secure wireless connection, i.e. the key secrecy for the initialization key and the authentication of session participants. In a first step, they analyze the *standard device pairing protocol* specified in the Bluetooth Core Specification based on short, low-entropy PINs for authentication. Therewith, they rediscover a known *vulnerability* that allows an attacker *to impersonate a Bluetooth device after eavesdropping* on a successful pairing session. In a second step they consider a *Simple Pairing protocol* which involves the Diffie–Hellman based key establishment. The authors perform authentication on a human visual channel, i.e. owners of mobile devices have to confirm on the screens' display the established keys by manually comparing the respective hash values of the parameters used to generate the keys. The automated, tool-supported analysis with ProVerif discovers a potential *vulnerability* caused by the *concurrent executions of authentication*. Chang and Shmatikov fix that problem by adding explicit session identifiers to the protocol. They expect that their main contribution, a formal model for a non-standard form of authentication by "human oracle", will serve as first step towards richer formal models of human authentication.

Duflot et al. apply in [72] the probabilistic model checking tool `PRISM` for an *analysis of the performance of the Bluetooth protocol*. They state to check Bluetooth version 1.1 and 1.2. The authors consider the Bluetooth *initialization process*, especially the time for the completion of the initialization process, which is a result of a non-trivial interaction between two devices. They especially compute the *best and worst-case performance* of device discovery, that means the expected time for the process to complete and the expected power consumption. In order to do so, they construct a complete model of the system that is able to compute actual performance values instead of only estimating these values from a large number of simulations. In contrast to using estimations from simulations, their approach gives them the possibility to accurately identify best- and worst-case expected times for the inquiry process. Moreover, they can even precisely establish situations that lead to these scenarios.

Sun and Sun consider in [74] the Bluetooth Standard v5.0-Part I. In home automation and entertainment systems the security depends to a huge part on the secure simple pairing (SSP). The authors state, that SSP can prevent both, passive eavesdropping and man-in-the-middle attacks. Depending on the capabilities of the device, one of the four models is

used for SSP: numeric comparison, out of band, passkey entry or just works. A formal model is developed in order to evaluate the security of the authenticated link key and the four models. The authors give a security proof for the numeric comparison and the out of band protocol to demonstrate the applicability of their formal security model. Their results show, that both protocols can resists against passive eavesdropping and man-in-the-middle attacks.

4.1.4. LoRaWAN

Literature research for LoRaWAN shows suggestions to secure LoRaWAN, e.g. [75], where a secure LoRaWAN sensor network architecture is proposed. Moreover, results consider several security features as activation methods, key management, cryptography, counter management and message acknowledgment in a proof-of-concept as implementation and testing for a fixed environment. Furthermore, Syther is used to detect a lack of synchronization between communication parties [76] and for checking encrypted key management and security requirements [77] (compare also Table 9). An enhanced version of LoRaWAN protocol is proposed in [78], which especially prevents replay attacks and ensures end-to-end security and another one in [79], where various security claims are successfully checked with the Scyther tool.

Oniga et al. proposed in [75] a novel secure LoRaWAN sensor network architecture. They claim that the standard security features provided by LoRaWAN are not sufficient for secure end-to-end communication of sensor networks and applications. Their main goal is to *prevent unauthorized access and data loss* in LoRaWAN sensor networks. Therefore, they propose a LoRaWAN network architecture and analyze potential security threats, especially concerning data protection and data privacy. They further implement different security controls to protect data transmitted over the network and to establish a strong line of defense in such networks. In detail the following scenarios are considered:

- at the end-node: *Sniffing* LoRa traffic, end-node *impersonation, replay messages*,
- at the gateway: *manipulate* communication's parameters, *Man-in-the-Middle attack* performed between Gateway and Broker,
- at the IDS: rules violation,
- at the network server: *Denial of Service*: database out-of-memory,
- at the application server: Sending *malicious payloads* using end-nodes.

Based on these findings the authors publish another paper, see [80], where they also consider the information given to an end-node during the activation process. They also perform testing, but without using Formal Methods.

Yang et al. analyze several security vulnerabilities of LoRaWAN v1.0.2 in [23]. They consider several security features in LoRaWAN, namely activation methods, key management, cryptography, counter management and message acknowledgment. Moreover, they design and describe five attacks:

- a *replay attack* leading to a selective Denial of Service on individual IoT devices
- *plaintext recovery*
- *malicious message modification*
- *falsification of delivery reports*
- *a battery exhaustion attack.*

They detect *five weaknesses* by implementing and executing the attacks in a controlled LoRaWAN environment as proof-of-concept. Moreover, they state to find five noteworthy weaknesses being able to compromise *confidentiality, integrity* and *availability* of a LoRaWAN deployment (see [23] for more details). Furthermore, they propose several countermeasures and changes to LoRaWAN protocol making these vulnerabilities less harmful.

Eldefrawy et al. give in [76] a formal security analysis of LoRaWAN with the `Scyther` tool. Two versions of LoRaWAN are considered, LoRaWAN *v*1.0 which is still the most widely deployed version of LoRaWAN, and the latest version, LoRaWAN *v*1.1. For each version a model in Scyther is built. For the first version, Eldefrawy et al. show that it is vulnerable due to a *lack of synchronization between communication parties*. That makes LoRaWAN vulnerable to *replay attacks*. Moreover, that vulnerability is related to attacks described in former research, see [81,82]. For the latest version they show with the Scyther tool that this vulnerability does not occur any longer. The latest version of LoRaWAN does not suffer from security claims based on their model. Nevertheless, they state that it is not possible to discover all the potential vulnerabilities by using tools like Scyther due to the limitations of the model.

Naoui et al. consider in [77] the *managing of encrypted key and security requirements*. They verify with the `Scyther` tool that LoRaWAN is vulnerable to *network server attack, DoS attack, replay attack* and *compromised key attack*. In a second step, an enhanced version of LoRaWAN architecture is proposed and evaluated with Scyther. Result show that the countermeasures fix the possible entry points properly, since non of the mentioned attacks is possible. Then, the enhanced version is compared to the basic LoRaWAN and to existing work like [83]. Based on the comparison, all three solutions guarantee the main security objectives including confidentiality, integrity and authentication. Moreover, the enhanced version of Naoui also avoids replay attacks and DoS attacks, which neither the original LoRaWAN, nor the version in [83] does.

Naoui et al. propose in [79] a novel enhanced LoRaWAN solution, which is adapted to secure the Smart Home remote control. The goal of the enhanced version is to prevent vulnerabilities detected in Activation by Personalization (ABP) and Over The Air Activation (OTAA) methods. The authors prove the security of their proposed scheme by an informal security analysis and a formal security verification with the `Scyther` tool. There, the authors consider *secrecy, aliveness and weak agreement, non-injective synchronization* and *non-injective agreement* and verify successfully all their claims. Furthermore, a performance comparison between their proposed scheme and other schemes in literature is given.

You et al. propose in [78] an enhanced LoRaWAN protocol and formally analyze their version with `Burrow-Abadi-Needham` (BAN) logic and the `AVISPA` tool. They state that LoRaWAN protocol fulfills the basic security properties, but suffers from the following vulnerabilities:

- The join procedure causes a vulnerability leading to an exploitation by *replay attacks*.
- No *end-to-end security* is given because the application session key between each device and its application server is established with the help of the core network.
- The network and application session keys cannot provide perfect *forward secrecy*. Since they are established based on a long-term shared key, every device can be easily broken and compromised.

Based on that, they propose a protocol supporting mutual authentication, secret key exchange, perfect forward secrecy, end-to-end security and defense against replay attacks. Then, they analyze their enhanced LoRaWAN version with BAN logic. Furthermore, they verify their claims by using the AVISPA tool.

4.1.5. Sigfox

For Sigfox there are just a few papers considering its security in detail. In [84] the Tamarin tool is used to check several properties as authentication, key establishment and message secrecy. In [85] a vulnerability analysis, especially addressing replay attacks is performed.

Kim et al. in [84] applied `Tamarin` to consider several properties under the Dolev-Yao model and under Perfect Forward Secrecy (PFS) [86]. In the Dolev-Yao model an insecure wireless channel is assumed, where an attacker intercepts the communication and is able to catch all messages, but only can read non encrypted messages or those where the attacker has the key for. PFS protects past sessions against future compromise of secret keys, i.e. even if the attacker gets access to a session

key, secrets of previous sessions are not revealed. The authors in [84] especially check the properties *authentication, key establishment* and *message secrecy*. All those properties are verified under the Dolev-Yao model. However, for PFS an attack on the key establishment has been detected. Furthermore, the authors considered cryptographic DoS attacks in IoT applications. The authors state, that those attacks are extremely effective since they effectively bind the constrained resources by causing intensive computations. In a first step, DoS attacks are modeled by using the Tamarin prover by using a Sigfox public key cryptography example. Their result shows, that Sigfox is vulnerable to an DoS attack on the signature. The authors model that by packing Sigfox packets – which all carry a signature – with a cryptographically heavy signature. If an adversary can force this step an unlimited number of times, the successful cryptographic DoS attack happen. In order to protect IoT protocols from such attacks, the authors propose a server puzzle, which can be used in any IoT protocol.

Coman et al. give in [85] a vulnerability analysis on Sigfox, including Proof-of-Concept attacks. The authors state that Sigfox is vulnerable to replay attacks. The reason is, that the 12-bit Sequence Number which is transmitted with every uplink frame and protected Message Authentication code, has a too small size. Therefore attackers can either inject previously send messages into the system or perform DoS-ing on the end-device. In their conclusion the authors state, that Sigfox in its current state should not be used in any critical applications, before having a better replay protection.

4.1.6. Narrowband IoT

As for Sigfox for Narrowband-IoT there are also just a few papers considering security in detail. In [85] a vulnerability analysis of the protocol is given. In [87] a scheme is proposed and checked against several security properties with the model checker Scyther.

Coman et al. perform in [85] a vulnerability analysis on Narrowband-IoT, including a Proof-of-Concept considering scans using malicious User Equipments (UEs). UEs can send IP data, therefore classical attack techniques over IP as e.g. port scanning, ARP spoofing or DNS spoofing need to be prevented inside Narrowband-IoT networks. Their results show, that an attacker can infiltrate private Narrowband-IoT networks and scan ports on the found Narrowband IoT devices. In a second stage, forged messages can be send to those open ports. The authors conclude, that Narrowband-IoT offers sufficient security guarantees. However, before deploying critical applications the user has to ensure that best security practices are enforced on the network, due to poor protection from UEs private network.

Cao et al. propose in [87] a fast mutual authentication and data transfer scheme for massive Narrowband IoT devices. The authors state that their scheme integrates the access authentication and secure data transmission process. Moreover, the authentication and data transmission of a group of Narrowband IoT devices can be achieved at the same time. The authors check their proposed scheme with the Canetti-Krawczyk model and additionally with the `Scyther` tool in order to show its security against attacks as replay attack, reflection attack, man-in-the-middle attack. Therefore, the authors especially check *secrecy, aliveness, weak agreement, non-injective agreement* and *non-injective synchronization*, which are all successfully verified by the Scyther tool.

4.1.7. 6LoWPAN

For 6LoWPAN the application of formal methods concentrate on applying it for proposed schemes.

Qiu and Ma propose in [88] an enhanced authentication and key establishment scheme for 6LoWPAN networks in machine to machine communication. They are using `Protocol Composition Logic` (PCL) and formally verify the *logical correctness* of their scheme by using the Simple Promela Interpreter `(SPIN)`. Furthermore, they state that their scheme can prevent multiple attacks as *replay attack, man-in-the-middle attack, impersonation attack, Sybil attack* and *compromised attack*.

Qui and Ma propose in [21] a secure Proxy Mobile IPv6 (MPIPv6) scheme enabling 6LoWPAN devices to roam efficiently and securely in 6LoWPAN networks. By using `AVISPA` and a Java simulation, they show that their proposed scheme in 6LoWPAN could efficiently enhance the security functionalities to prevent various attacks as *replay attack, Man-in-the-middle attack, privileged insider attack* and *Sybil attack* with less computational cost.

Qui and Ma propose in [89] a scheme to support secure and seamless handovers for a group of resource constrained 6LoWPAN devices. The authors use `BAN` logic and the tool `Scyther` for proving that it is resistant against a *replay attack*, a *Man-in-the-middle attack*, an *impersonation attack*, a *privileged insider attack* and a *Sybil attack*. Additionally the authors give a performance evaluation of their proposed scheme.

4.1.8. LTE

For LTE research focuses on security properties as in [36], where three different model checkers are used to prove secrecy, agreement and key freshness. Several security properties are also considered in [90]. Moreover, literature proposes new schemes as in [91] and [92], where enhanced schemes to secure authentication are suggested.

Henda et al. consider in [36] and [18] Dual Connectivity, a new feature in LTE. The authors present a model for a *key establishment* of this feature. Moreover, they provide a detailed formal analysis by using three popular academic tools, namely `Scyther`, `Tamarin` and `ProVerif` automatically prove security properties of secrecy, agreement and key freshness. The authors state that their main purpose is to evaluate the suitability of different tools for modeling a rapidly changing system. They further note that their verification with Scyther and Tamarin tool provides weaker results, because all the properties could only be proven in the bounded models. For the Scyther tool the user has to specify a bound which acts as limit on the number of runs. In Tamarin the limit is on the proof depth, where the relation to the protocol parameters is not clear to the authors. Furthermore, the authors evaluate the tools for usability, expressiveness and performance. For usability Scyther is considered to be the most-user friendly one due to its simple input language and graphical interface. In performance ProVerif has a clear advantage because Scyther and Tamarin only converge for bounded models. On the expressiveness, Tamarin is considered to be the best, since it provides support for arbitrary control flow structures such as loops and conditionals. As final remark the authors state, that none of the tools would provide full support of all features on their own according to their case study. However, in combinations good results can be obtained.

Vassilakis et al. perform in [90] a formal security verification of the key establishment protocol for the dual connectivity (DC) in small cell LTE networks. Therefore, the authors model the key establishment protocol and used the model checker `Scyther` in order to verify properties as *secrecy, agreement* and *key freshness*. All the properties are successfully verified for the key establishment protocol, however, some properties fail for simulating the key leakage with Scyther.

Cao et al. consider in [91] two types of base stations in LTE wireless networks, namely Home eNodeB and eNodeB. To achieve seamless handovers between these two base stations, third the Generation Partnership Project requires distinct procedures with a complex key management mechanism. A drawback of this is that it cannot give backward security. The authors propose *a fast and secure handover authentication scheme*. They state that their scheme is simple, gives a desirable efficiency and also provides several security features including Perfect Forward/Backward Secrecy. For proving that `AVISPA` is used.

Lai et al. propose in [92] a secure and efficient authentication and key agreement (AKA) protocol called SE-AKA, which can fit in with all of the group authentication scenarios in the LTE network. The authors state that the Evolved Packet System (EPS) in the Long Term Evolution (LTE) networks, the EPS-AKA protocol, which is an emerging standard for fourth-generation wireless communication, cannot resist several frequent attacks, i.e. *redirection attack, man-in-the-middle attack* and *DoS at-*

tack. The proposed SE-AKA provides strong security including privacy-preservation. Furthermore, it provides a group authentication mechanism which can effectively authenticate group devices. The authors perform security analysis and Formal Verification by using `ProVerif` and verified *entity mutual authentication, confidentiality, privacy-preservation (anonymity), key forward/backward secrecy*. Moreover, SE-AKA resists against redirection attack, man-in-the-middle attack and DoS attack.

4.1.9. 5G

5G is a protocol used in many domains, as also mentioned in [93], where a short review on Formal Methods for connected vehicle protocols – including 5G – is given. Important work in the domain of Formal Verification for 5G include the work of Basin et al. in [94] and Cremers and Dehnel-Wild in [95]. For completeness a short summary of boths papers is given. Recent approaches in [96,97] also consider the 5G-EAP-TLS.

Basin et al. perform in [94] an exhaustive formal analysis of 5G authenticated key exchange (AKA) protocol with the `Tamarin` prover. Their contribution is very broad, starting with a formalization of the 5G standard, giving a formal model of 5G AKA and also providing a security analysis of 5G AKA, addressing *confidentiality, authentication* and *privacy*. The findings of the authors show that some security goals and assumptions in the 5G standard are underspecified or even missing, including central goals like the agreement on the session key. Furthermore, their research shows a possible privacy attack. Based on that, the authors give recommendations and propose a provable fix to the security issues.

Cremers and Dehnel-Wild consider in [95] also 5G AKA with `Tamarin` but focus on *a fine-grained formal analysis* by considering all parties defined by the protocol specification. The findings of the authors show that 5G AKA relies on the unstated assumption on the inner workings of underlying channels, which results in an attack exploiting a potential race condition. Therefore, it is easily possible to ”correctly” implement the standard in an insecure manner. Based on the detected vulnerabilities in the standard, the authors propose a possible fix.

Zhang et al. consider in [96] the 5G-EAP-TLS protocol. The main security goal for that protocol – defined for subscriber authentication in limited use cases as private networks or IoT envirnoments – is to provide mutual authentication. The authors developed a formal model for 5G-EAP-TLS and used the model checker `Scyther` for checking the *secrecy* of the Subscription Permanent Identifier, the *secrecy* of the session key, the *non-injective agreement* on the data and the *non-injective synchronization* of events. The results of Scyther show, that only secrecy can be successfully proven, the other statements do not hold.

Zhang et al. focus in [97] also on the 5G-EAP-TLS protocol. There, the authors extended their previously work in [96] and used the tool `ProVerif`. Therewith, they are able to model the behavior of the protocol more precisely. For the Formal Verification the authors consider two *authentication* and three *secrecy* statements. The authentication statements, namely that the home network and the subscriber should agree on the identity of each other after a successful termination of the protocol and that both parties agree on the pre-master key are falsified. The secrecy statements can be verified, i.e. the adversary cannot obtain the Subscription Permanent Identifier, the pre-master key nor the session key from an honest subscriber. Additionally the authors suggest a fix and prove that the fix satisfies the stated security properties.

Publications on suggestions to fix 5G are manifold and ongoing, including the last four papers. Therefore, just a short overview of these approaches fitting to the IoT domain is given here. **El Idrissi et al.** propose in [98] an enhancement of the existing AKA protocol to improve authentication and authorization in 5G networks by using Elliptic Curve Cryptosystem and `AVISPA` to check against potential security attacks. **Koutsos** propose in [99] a provably fixed version of 5G AKA that prevents privacy attacks. For formal verification the `Bana-Comon` indistinguishably logic is used. Another suggestions is given by **Conceicao et al.** in [100]. In their paper, a new protocol for key establishment without any prior trust relationship between User Equipment and a Machine Type Communication device is suggested. For verifying the authors use `ProVerif`.

Table 9
Attacks found for LoRaWAN.

Source	[77]	[23]	[76]-v1.0	[76]-v1.1
Replay attacks	√	√	√	✗
DoS attacks	√	✗	–	–
network server attack	√	–	–	–
compromised key attack	√	–	–	–
Plaintext recovery	–	√	–	–
Malicious message modification	–	√	–	–
Falsification of delivery reports	–	√	–	–
Battery exhaustion attack	–	√	–	–

Table 10
Main application fields for formal methods, * just vulnerability analysis.

Protocol	functional	propose extension	security properties	implementation
ZigBee	√	√	√	✗
Z-Wave	✗	✗	√*	√*
Bluetooth	√	✗	√	✗
LoRaWAN	✗	√	√	✗
Sigfox	✗	✗	√	✗
Narrowband-IoT	✗	√	√	✗
6LoWPAN	✗	√	√	✗
LTE	✗	√	√	✗
5G	√	√	√	✗

5. Discussion

This section summarizes and classifies existing approaches in the literature. Furthermore, pros and cons, including weaknesses of the investigated literature are pointed out. Based on that, challenges and open issues in the field are stated.

5.1. Existing approaches

Considering the presented papers and the work therein, one can conclude that there are four main areas in order to apply Formal Verification Methods to protocols used in IoT environment, see Table 10. First, there are functional tests performed by Formal Verification containing e.g. checks of the protocol stack and time for key update and key recovery. Second, there are proposed schemes or extended protocols which are provably fixed by using Formal Methods. A third class applies Formal Verification Methods and tries to verify security properties, including proofs that some special attacks are impossible. Depending on the tool, there is even the option that it reconstructs an attack if the considered security property is falsified. The last class concentrates on checking implementations of protocols. It has to be emphasized, that both, verification of protocol's specifications and of its implementations are important. Due to the translation of the protocol into code, misunderstandings might happen, necessary functions or methods in the code can be missing or programming bugs can appear.

Table 10 shows, that for all protocols, security properties are checked. Some of them even go along with checking security properties on enhanced versions of the protocol. Functional checks and a checking of implementations are less common in literature for protocols under consideration. However, that strongly depends on the type of protocol, as e.g. for MQTT and CoAP the situation is different (see [7]) and checking functional correctness and implementations are more common in literature than considering explicitly security properties.

5.1.1. Focus on security properties

Table 11 provides an overview of protocols checked against different security properties. The table shows that the most commonly checked

Table 11
Considered security properties for protocols (independent of verification/falsification result), + indicates a check, - no check.

Protocol	availability	confidentiality	integrity	authenticity	accuracy
ZigBee	+	+	–	+	–
Z-Wave	–	+	+	+	–
Bluetooth	–	+	+	+	–
LoRaWAN	+	+	+	+	+
Sigfox	+	+	–	+	–
Narrowband-IoT	–	+	–	+	–
6LoWPAN	–	+	+	+	+
LTE	–	+	–	+	–
5G	–	+	–	+	–

security property is confidentiality. The reason for that lies in the use cases the protocols are usually used in, as e.g. Smart Home. There privacy concerns are of course in the focus, which are especially addresses in security properties as confidentiality and authenticity. However, depending on the use the situation can be very different, e.g. in Smart Grid where availability is much more important.

A detailed analysis of the investigated approaches shows, that there is a huge difference between the depth of checking security properties within the different protocols. While there are several publications addressing 5G and LoRaWAN, there are just a view approaches for other protocols as ZigBee, Z-Wave, 6LoWPAN, Sigfox and Narrowband-IoT.

For Z-Wave literature focus on performing vulnerability analysis, without applying formal verification tools. For 6LoWPAN and Narrowband-IoT the investigated literature consider the application of model checkers to enhanced schemes only. For ZigBee existing approaches [63] check multiple authentication attack types with AVISPA, especially the case where the attacker has knowledge of the default key. For Sigfox the Tamarin tool is applied in [84] for an in-depth check of several properties as authentication, key establishment and message secrecy. The authors verified the properties under the Dolev-Yao model and detected an attack on key establishment under Perfect Forward Secrecy.

In Bluetooth, literature covers the authentication properties of a Bluetooth device pariing with ProVerif [71], especially the key secrecy for the initialization key and the authentication of session participants.

For 5G, beside the number of publication, the depth of existing approaches to investigate the topic is remarkable too. There are two very valuable approaches on 5G-AKA, the exhaustive formal analysis in [94] addressing confidentiality, authentication and privacy and [95], a fine-grained formal analysis considering all parties defined in the protocol specification. The 5G-EAP-TLS protocol is considered in [96,97], where in the first publication the Scyther tool is used and in the second one the authors apply the model checker ProVerif. They state, that they are able to model the behavior of the protocol more precisely with the ProVerif tool.

For LoRaWAN it has to be noted, that different version and different tools are applied in literature. While [76] considers v1.0 and v1.1 and detects vulnerabilities, especially a lack of synchronization between communication parties and a vulnerability to replay attacks with Scyther for v1.0, those vulnerabilities can be proven to be absent for v1.1. Further approaches consider approaches with Scyther to check the managing of encrypted key and security requirements, detecting vulnerabilities to network server attack, DoS attack, replay attack and compromised key attack. A replay attack is also detected by applying AVISPA in [78] stating also a lack of end-to-end security and a lack of forward secrecy.

For LTE [36] the authors models the key establishment in three different model checkers, namely Scyther, Tamarin and ProVerif. With those tools the authors investigate secrecy, agreement and key freshness. The authors state that results with the tools Scyther and Tamarin are weaker, since the properties could only be proven in the bounded model, while that was different for ProVerif. Additionally the different tools have been evaluated in terms of usability, expressiveness and performance. Another approach considering security properties for LTE in [90] gives a formal security verification of the key establishment protocol for the dual connectivity.

5.1.2. Focus on probabilistic model checking

Overall it can be stated, that – excluding enhanced schemes – there are less publications applying probabilistic model checking then applying security protocol model checkers to the considered protocols.

Probabilistic model checking is done for ZigBee and Bluetooth. For both protocols there are two publications each.

For ZigBee UPPAAL and PRISM have been applied, but for checking different properties. With UPPAAL in [61] the collision avoidance property, the bounded liveness property and the deadlock freeness property are sucessfully verified. PRISM is used to model a ZigBee network as continuous-time Markov chain in order to optimise key confidentiality, the key recovery time (from a compromise case) and the efficiency of key updates.

For Bluetooth also both tools, UPPAAL and PRISM are applied. While [70] considers the device discovery mechanism, especially for checking the data connectivity, PRISM is used in [72] for an analysis of the performance of the Bluetooth protocol for version 1.1. and version 1.2. The authors especially give a best and worst-case performance of the device discovery as the expected time for the process to complete and the expected power consumption.

5.1.3. Overview of tools in literature

An overview on model checkers and methods used for different protocols is given in Table 12. Moreover, an overview of tools used in different application fields is given in Fig. 8.

Table 12 shows that the tool most often used in literature for considered protocols is Scyther followed by AVISPA, ProVerif and Tamarin.

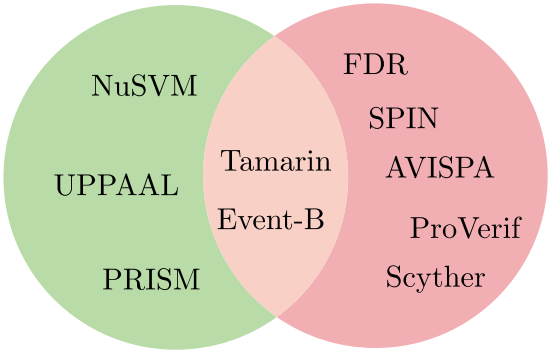

Fig. 8. Formal Verification Tools for protocols under consideration with respect to application fields (left: functional, right: security properties).

Table 12
Overview of tools used for different protocols in literature.

	ZigBee	Bluetooth	LoRaWAN	6LoWPAN	LTE	5G	Sigfox	NB-IoT
AVISPA	[63,65]		[78]	[21]	[91]	[98]		
ProVerif		[71]			[36,92]	[97,100]		
Event-B	[60,62]							
PRISM	[64]	[72]						
UPPAAL	[61]	[70]						
NuSMV		[69]						
Scyther			[76,77]	[89]	[36,90]	[96]		[87]
Tamarin					[36]	[94,95]	[84]	
SPIN				[88]				
FDR	[63]							

Fig. 8 shows that most of the tools are mainly used for one of the application fields. Only Event-B is used for two, that is functional correctness and security properties. The reason why most of the tools are mainly used in one application field is likely due to different focuses of the model checker, which also raises the challenge to use them in different application fields.

According to Table 12 it shall be noted, that most of the protocols – beside ZigBee and Bluetooth – are mainly investigated by security protocol model checkers. However, for those two protocols it has to be stated, that they are both verified with PRISM and UPPAAL. Although it is obvious, it shall be stated that nearly all publications only apply one of the different model checkers. This is remarkable, since several publications state limitations of their model, e.g. [36,76]. While [76] only states, that due to the limitations of the model it was not possible to discover all potential vulnerabilities, in [36] three security protocol model checkers have been applied. The authors there state, that none of the tools provide full support of all features according to their case study. However, in combination they have been able to obtain good results. Moreover, most of the publications focus on the widely used Dolev-Yao model. Nevertheless, approaches as in [84], where the Dolev-Yao model and the PFS are taken under consideration and yield different results on possible attacks for the key establishment property might be worth for a better overall understanding and in-depth analysis.

5.2. Challenges & open issues

In general it is not possible to verify the whole protocol. Therefore, the first challenge one has to solve when doing protocol verification, is to identify critical parts for a detailed consideration of the specification or the standard. These critical parts need then to be modeled and later on transferred to the input language of the selected model checker, which possess the second challenge. Due to limitations of the selected tool, adjustments on the protocol model might be necessary, which can lead to a weaker result. Although there are approaches to overcome that issue, see [36], where three security protocol model checkers have been applied, a combination of several tools to improve the overall result are rare in literature yet. The drawback with that approach is, that three different input models – due to the different input languages of the tools – have to be created. In order to overcome that drawback, translators or tools for assisted translation in order to make the process faster, might be helpful.

An unexpected point during the research was, that there is a lack of investigated implementations of the protocols under consideration. While there are several approaches to check protocol's implementations for application layer protocols as MQTT and CoAP [7], for the investigated protocols there are no formal methods applied to implementations, beside a detailed vulnerability analysis of a Z-wave's implementation. However, as shown in [7] and the references therein, checking of protocol's implementations is crucial. In [101], where the authors used formal verification to check if the MQTT implementations of three open source implementations adhere to the standard and therefore checked 13 normative statements, selected to cover many different aspects of the protocol, all three implementations failed for different normative statements. Depending on the statement, even two or three of the implementations failed for the same. Another publication [102] considering the security of applications implementing MQTT detected several unknown vulnerabilities in widely used applications implementing the MQTT protocol.

Depending on the protocol, additional consideration is possible for forced protocol downgrading, which is especially important for 5G. There, due to unavailable technology, downgrading from e.g. 5G to LTE/3G is quite likely and it's impact on security has to be taken into consideration [93].

As summary, several research perspectives – some general one and some addressing especially the investigated protocols – can be stated. First, the general ones shall be summarized:

- Translators in order to transfer input for one model checker to input for another model checker.
- Combination of tools for better overall results.
- Detailed case study for different protocols and properties to investigate the pros and cons of different tools in depth and give a more detailed comparison of those tools.
- Influence on the security due to forced protocol downgrading (e.g. for 5G to LTE/3G).

Second, research perspectives for the investigated protocols are addressed explicitly:

- Model checking for different versions of a protocol (e.g. for protocols as Bluetooth).
- Formal verification of protocols not considered yet as e.g. Z-Wave, Narrowband IoT, KNX or Thread
- Investigation of implementations:
  - In order to verify if the follow protocol's specification.
  - In order to verify that they do not pose additional security issues.

6. Conclusion

This paper considers the application of Formal Verification to protocols in the IoT environment. A broad range of considered protocols are described in detail, with a focus on the security mechanisms. Additionally, a general description of common tools and the basic technology behind is given as introduction to application of Formal Verification. The main focus of this paper, applications of Formal Verification for IoT protocols can be separated into four classes. First, functional checks including functional correctness of the protocol. Second, verification or falsification of security properties. A falsification of some security properties open the window for the third approach: the proposal of an enhanced scheme or protocol to prevent the vulnerability. Finally, there are tests on implementations of the protocols. Therein, the goal is usually to check, if a certain implementation sticks to the standard and does not open any security issues e.g. by containing implementation bugs.

Details of the checked properties, including a short description of the approach, with focus on the considered property and used tool, is given. Based on that, application fields and security properties covered for each protocols are shown. Furthermore, insights on abilities of different commonly tools and suitable applications fields are provided. In an overall consideration, approaches using security protocol model checkers and probabilistic model checkers are summarized, including consideration of weaknesses and strength of the already covered topics. Based on the investigated literature, challenges and open research perspectives are stated.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was funded in part by the Austrian Federal Ministry of Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK) and in part by European Union's Horizon 2020 research and innovation programme under grant agreement No 830892, project SPARTA.

References

[1] S. Radomirovic, Towards a Model for Security and Privacy in the Internet of Things, in: Proc. First Intl Workshop on Security of the Internet of Things, 2010.
[2] M. Ge, Graphical Security Modelling and Assessment for the Internet of Things, University of Canterbury, 2018 Ph.D. thesis.
[3] S. Marksteiner, V.J.E. Jimenez, H. Valiant, H. Zeiner, An Overview of Wireless IoT Protocol Security in the Smart Home Domain, in: 2017 Internet of Things Business Models, Users, and Networks, 2017, pp. 1–8, doi:10.1109/CTTE.2017.8260940.
[4] J. Batalla, A. Vasilakos, M. Gajewski, Secure Smart Homes: Opportunities and Challenges, ACM Computing Surveys 50 (2017) 1–32, doi:10.1145/3122816.
[5] K. Keerthi, I. Roy, A. Hazra, C. Rebeiro, Formal Verification for Security in IoT Devices, in: Security and Fault Tolerance in Internet of Things, Springer, 2019, pp. 179–200.
[6] R.J. Punnoose, R.C. Armstrong, M.H. Wong, M. Jackson, Survey of Existing Tools for Formal Verification, Technical Report, Sandia National Laboratories, Livermore, CA (United States), 2014.
[7] K. Hofer-Schmitz, B. Stojanović, Towards Formal Methods of IoT Application Layer Protocols, in: 2019 12th CMI Conference on Cybersecurity and Privacy (CMI), 2019, pp. 1–6, doi:10.1109/CMI48017.2019.8962139.
[8] P. Merlin, A Methodology for the Design and Implementation of Communication Protocols, IEEE Transactions on Communications 24 (6) (1976) 614–621, doi:10.1109/TCOM.1976.1093347.
[9] ISO/IEC, Information Technology–Open Systems Interconnection–Basic Reference Model: The Basic Model, International Organization for Standardization/International Electrotechnical Commission and others, 1994.
[10] R. Braden, Requirements for Internet hosts-communication layers, Network Working Group, 1989a.
[11] R. Braden, Requirements for Internet hosts - application and support, Network Working Group, 1989b.
[12] E. Fernandes, A. Rahmati, K. Eykholt, A. Prakash, Internet of Things Security Research: A Rehash of Old Ideas or New Intellectual Challenges? IEEE Security & Privacy 15 (4) (2017) 79–84.
[13] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun, H.-Y. Du, Research on the architecture of Internet of Things, in: 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), 5, 2010, doi:10.1109/ICACTE.2010.5579493.
[14] I. Cvitić, M. Vujić, S. Husnjak, Classification of Security Risks in the IoT Environment, in: 26th International DAAAM Symposium on Intelligent Manufacturing and Automation, 2016, pp. 731–740.
[15] M.E. Whitman, H.J. Mattord, Principles of Information Security, Cengage Learning, 2011.
[16] D. Dolev, A. Yao, On the Security of Public Key Protocols, IEEE Transactions on Information Theory 29 (2) (1983) 198–208, doi:10.1109/TIT.1983.1056650.
[17] M. Pourpouneh, R. Ramezanian, A Short Introduction to Two Approaches in Formal Verification of Security Protocols: Model Checking and Theorem Proving, The ISC International Journal of Information Security 8 (1) (2016) 3–24.
[18] K. Pfeffer, Formal Verification of a LTE Security Protocol for Dual-Connectivity: An Evaluation of Automatic Model Checking Tools, KTH Royal Institute of Technology, School of Information and Communication Technology, 2014 Master's thesis.
[19] C.J.F. Cremers, Scyther: Semantics and Verification of Security Protocols, Eindhoven University of Technology Eindhoven, Netherlands, 2006.
[20] G. Lowe, A hierarchy of authentication specifications, in: Proceedings 10th Computer Security Foundations Workshop, IEEE, 1997, pp. 31–43.
[21] Y. Qiu, M. Ma, A PMIPv6-Based Secured Mobility Scheme for 6LoWPAN, in: 2016 IEEE Global Communications Conference (GLOBECOM), 2016, pp. 1–6, doi:10.1109/GLOCOM.2016.7841534.
[22] C.W. Badenhop, S.R. Graham, B.W. Ramsey, B.E. Mullins, L.O. Mailloux, The Z-Wave routing protocol and its security implications, Computers & Security 68 (2017) pp.112–129.
[23] X. Yang, E. Karampatzakis, C. Doerr, F. Kuipers, Security Vulnerabilities in LoRaWAN, in: 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI), 2018, pp. 129–140, doi:10.1109/IoTDI.2018.00022.
[24] C. Meadows, A Formal Framework and Evaluation Method for Network Denial of Service, in: Proceedings of the 12th IEEE Computer Security Foundations Workshop, IEEE, 1999, pp. 4–13.
[25] S. Khanji, F. Iqbal, P. Hung, ZigBee Security Vulnerabilities: Exploration and Evaluating, in: 2019 10th International Conference on Information and Communication Systems (ICICS), IEEE, 2019, pp. 52–57.
[26] T. Zillner, S. Strobl, ZigBee Exploited - the good, the bad and the ugly, Magdeburger Journal zur Sicherheitsforschung (2015).
[27] E. Biham, L. Neumann, Breaking the Bluetooth Pairing–The Fixed Coordinate Invalid Curve Attack, in: International Conference on Selected Areas in Cryptography, Springer, 2019, pp. 250–273.
[28] D. Antonioli, N.O. Tippenhauer, K.B. Rasmussen, The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation Of Bluetooth BR/EDR, in: 28th USENIX Security Symposium (USENIX Security 19), USENIX Association, Santa Clara, CA, 2019, pp. 1047–1061.
[29] S. Chacko, M.D. Job, Security mechanisms and Vulnerabilities in LPWAN, in: IOP Conference Series: Materials Science and Engineering, 396, IOP Publishing, 2018, p. 012027.
[30] P. Kasinathan, C. Pastrone, M.A. Spirito, M. Vinkovits, Denial-of-Service detection in 6LoWPAN based Internet of Things, in: 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2013, pp. 600–607.
[31] P. Pongle, G. Chavan, A survey: Attacks on RPL and 6LoWPAN in IoT, in: 2015 International Conference on Pervasive Computing (ICPC), 2015, pp. 1–6, doi:10.1109/PERVASIVE.2015.7087034.
[32] J. Cao, M. Ma, H. Li, Y. Zhang, Z. Luo, A Survey on Security Aspects for LTE and LTE-A Networks, IEEE Communications Surveys Tutorials 16 (1) (2014) 283–302, doi:10.1109/SURV.2013.041513.00174.
[33] D. Basin, C. Cremers, C. Meadows, Model Checking Security Protocols, in: Handbook of Model Checking, Springer, 2018, pp. 727–762.
[34] A. Hinton, M. Kwiatkowska, G. Norman, D. Parker, PRISM: A Tool for Automatic Verification of Probabilistic Systems, in: H. Hermanns, J. Palsberg (Eds.), Tools and Algorithms for the Construction and Analysis of Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 441–444.
[35] C. Baier, L. de Alfaro, V. Forejt, M. Kwiatkowska, Model Checking Probabilistic Systems, Springer International Publishing, Cham, pp. 963–999. 10.1007/978-3-319-10575-8_28
[36] N.B. Henda, K. Norrman, K. Pfeffer, Formal Verification of the Security for Dual Connectivity in LTE, in: 2015 IEEE/ACM 3rd FME Workshop on Formal Methods in Software Engineering, 2015, pp. 13–19, doi:10.1109/FormaliSE.2015.10.
[37] A. Naeem, F. Azam, A. Amjad, M.W. Anwar, Comparison of Model Checking Tools Using Timed Automata - PRISM and UPPAAL, in: 2018 IEEE International Conference on Computer and Communication Engineering Technology (CCET), 2018, pp. 248–253, doi:10.1109/CCET.2018.8542231.
[38] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P.H. Drielsma, P.C. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, L. Vigneron, The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications, in: International Conference on Computer Aided Verification, Springer, 2005, pp. 281–285.
[39] L. Viganò, Automated Security Protocol Analysis With the AVISPA Tool, Electronic Notes in Theoretical Computer Science 155 (2006) 61–86, doi:10.1016/j.entcs.2005.11.052. Proceedings of the 21st Annual Conference on Mathematical Foundations of Programming Semantics (MFPS XXI)
[40] AVISPA v1.0 User Manual, 2006. http://avispa-project.org.
[41] J.-R. Abrial, Modeling in Event-B: System and Software Engineering, Cambridge University Press, 2010.
[42] T.S. Hoang, Industrial Deployment of System Engineering Methods, Springer, pp. 211–236.
[43] 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) (2010) 447–466.
[44] M. Kwiatkowska, G. Norman, D. Parker, PRISM 4.0: Verification of Probabilistic Real-Time Systems, in: International Conference on Computer Aided Verification, Springer, 2011, pp. 585–591.
[45] M. Kwiatkowska, G. Norman, D. Parker, PRISM: Probabilistic Symbolic Model Checker, in: International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, Springer, 2002, pp. 200–204.
[46] M. Kattenbelt, M. Kwiatkowska, G. Norman, D. Parker, A game-based abstraction-refinement framework for Markov decision processes, Formal Methods in System Design 36 (3) (2010) 246–280.
[47] T. Hérault, R. Lassaigne, F. Magniette, S. Peyronnet, Approximate Probabilistic Model Checking, in: International Workshop on Verification, Model Checking, and Abstract Interpretation, Springer, 2004, pp. 73–84.

[48] H.L. Younes, R.G. Simmons, Probabilistic Verification of Discrete Event Systems Using Acceptance Sampling, in: International Conference on Computer Aided Verification, Springer, 2002, pp. 223–235.

[49] R. Alur, T.A. Henzinger, Reactive Modules, Formal Methods in System Design 15 (1) (1999) 7–48.

[50] B. Blanchet, An Efficient Cryptographic Protocol Verifier Based on Prolog Rules, in: Proceedings of the 14th IEEE Workshop on Computer Security Foundations, CSFW '01, IEEE Computer Society, USA, 2001.

[51] B. Blanchet, Automatic Verification of Security Protocols in the Symbolic Model: The Verifier ProVerif, Springer International Publishing, Cham, pp. 54–87. 10.1007/978-3-319-10082-1_3

[52] B. Blanchet, Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif, Foundations and Trends®in Privacy and Security 1 (1-2) (2016) 1–135, doi:10.1561/3300000004.

[53] B. Blanchet, B. Smyth, V. Cheval, M. Sylvestre, ProVerif 2.00: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial, 2018.

[54] C.J.F. Cremers, The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols, in: A. Gupta, S. Malik (Eds.), Computer Aided Verification, Springer Berlin Heidelberg, 2008, pp. 414–418.

[55] S. Meier, B. Schmidt, C. Cremers, D. Basin, The TAMARIN Prover for the Symbolic Analysis of Security Protocols, in: International Conference on Computer Aided Verification, Springer, 2013, pp. 696–701.

[56] D. Basin, C. Cremers, J. Dreier, R. Sasse, Symbolically Analyzing Security Protocols Using Tamarin, ACM SIGLOG News 4 (4) (2017) 19–30.

[57] B. LaMacchia, K. Lauter, A. Mityagin, Stronger Security of Authenticated Key Exchange, in: W. Susilo, J.K. Liu, Y. Mu (Eds.), Provable Security, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 1–16.

[58] J. Bengtsson, K. Larsen, F. Larsson, P. Pettersson, W. Yi, UPPAAL — a Tool Suite for Automatic Verification of Real-Time Systems, in: R. Alur, T.A. Henzinger, E.D. Sontag (Eds.), Hybrid Systems III, Springer Berlin Heidelberg, Berlin, Heidelberg, 1996, pp. 232–243.

[59] G. Behrmann, A. David, K.G. Larsen, A Tutorial on UPPAAL 4.0, Department of Computer Science, Aalborg University, 2006.

[60] A. Gawanmeh, Embedding and Verification of ZigBee Protocol Stack in Event-B, Procedia Computer Science 5 (2011) 736–741, doi:10.1016/j.procs.2011.07.097. The 2nd International Conference on Ambient Systems, Networks and Technologies (ANT-2011) / The 8th International Conference on Mobile Web Information Systems (MobiWIS 2011)

[61] A. Rashid, O. Hasan, K. Saghar, Formal Analysis of a ZigBee-based Routing Protocol for Smart Grids using UPPAAL, in: 2015 12th International Conference on High-capacity Optical Networks and Enabling/Emerging Technologies (HONET), 2015, pp. 1–5, doi:10.1109/HONET.2015.7395420.

[62] R.M. Nadeem, A.A. Gill, A Formal Model for Verification of ZigBee Protocol for Secure Network Authentication, Indian Journal of Science and Technology 10 (20) (2017).

[63] A.P. Melaragno, D. Bandara, D. Wijesekera, J.B. Michael, Securing the ZigBee Protocol in the Smart Grid, Computer 45 (4) (2012) 92–94, doi:10.1109/MC.2012.146.

[64] E. Yüksel, H.R. Nielson, F. Nielson, M. Fruth, M. Kwiatkowska, Optimizing ZigBee Security using Stochastic Model Checking, Technical Report, Technical University of Denmark, Informatics and Mathematical Modelling and Oxford University, Computing Laboratory, 2012.

[65] M. Alshahrani, I. Traore, I. Woungang, Anonymous mutual IoT interdevice authentication and key agreement scheme based on the ZigBee technique, Internet of Things 7 (2019) 100061, doi:10.1016/j.iot.2019.100061.

[66] A. Hafeez, N.H. Kandil, B. Al-Omar, T. Landolsi, A. Al-Ali, Smart Home Area Networks Protocols within the Smart Grid Context, Journal of Communications 9 (9) (2014) 665–671.

[67] Failures-Divergence Refinement, Formal Systems (Europe) Ltd and Oxford University Computing Laboratory, 2010.

[68] B. Fouladi, S. Ghanoun, Security Evaluation of the Z-Wave Wireless Protocol, 2013. Black Hat USA.

[69] E. Pek, N. Bogunovic, Formal Verification of Logical Link Control and Adaptation Protocol, in: Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference (IEEE Cat. No.04CH37521), 2, 2004, pp. 583–586Vol.2, doi:10.1109/MELCON.2004.1346997.

[70] S. Arry, A. Kaur, Formal verification of device discovery mechanism using UPPAAL, International Journal of Computer Applications 58 (19) (2012).

[71] R. Chang, V. Shmatikov, Formal Analysis of Authentication in Bluetooth Device Pairing, 2007. FCS-ARSPA07.

[72] M. Duflot, M. Kwiatkowska, G. Norman, D. Parker, A Formal Analysis of Bluetooth Device Discovery, International Journal on Software Tools for Technology Transfer 8 (6) (2006) 621–632.

[73] A. Cimatti, E. Clarke, F. Giunchiglia, M. Roveri, NUSMV: a new symbolic model checker, International Journal on Software Tools for Technology Transfer 2 (4) (2000) 410–425, doi:10.1007/s100090050046.

[74] D.-Z. Sun, L. Sun, On Secure Simple Pairing in Bluetooth Standard v5. 0-Part I: Authenticated Link Key Security and Its Home Automation and Entertainment Applications, Sensors 19 (5) (2019) 1158.

[75] B. Oniga, V. Dadarlat, E. De Poorter, A. Munteanu, A secure LoRaWAN sensor network architecture, in: 2017 IEEE SENSORS, 2017, pp. 1–3, doi:10.1109/ICSENS.2017.8233990.

[76] M. Eldefrawy, I. Butun, N. Pereira, M. Gidlund, Formal Security Analysis of LoRaWAN, Computer Networks 148 (2019) 328–339, doi:10.1016/j.comnet.2018.11.017.

[77] S. Naoui, M.E. Elhdhili, L.A. Saidane, Trusted Third Party Based Key Management for Enhancing LoRaWAN Security, in: 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), 2017, pp. 1306–1313, doi:10.1109/AICCSA.2017.73.

[78] I. You, S. Kwon, G. Choudhary, V. Sharma, J. Seo, An Enhanced LoRaWAN Security Protocol for Privacy Preservation in IoT with a case study on a smart factory-enabled parking system, Sensors 18 (6) (2018) 1888.

[79] S. Naoui, M. Elhdhili, L. Saidane, Novel Enhanced LoRaWAN Framework for Smart Home Remote Control Security, Wireless Personal Communications 110 (4) (2020) 2109–2130, doi:10.1007/s11277-019-06832-x.

[80] B. Oniga, V. Dadarlat, E. De Poorter, A. Munteanu, Analysis, design and implementation of secure LoRaWAN sensor networks, in: 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), 2017, pp. 421–428, doi:10.1109/ICCP.2017.8117042.

[81] S. Tomasin, S. Zulian, L. Vangelista, Security Analysis of LoRaWAN Join Procedure for Internet of Things Networks, in: 2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), 2017, pp. 1–6, doi:10.1109/WCNCW.2017.7919091.

[82] S.J. Na, D.Y. Hwang, W.S. Shin, K.-H. Kim, Scenario and countermeasure for replay attack using join request messages in LoRaWAN, in: 2017 International Conference on Information Networking (ICOIN), 2017, pp. 718–720, doi:10.1109/ICOIN.2017.7899580.

[83] S. Naoui, M.E. Elhdhili, L.A. Saidane, Enhancing the security of the IoT LoraWAN architecture, in: 2016 International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN), IEEE, 2016, pp. 1–7.

[84] J.Y. Kim, R. Holz, W. Hu, S. Jha, Automated Analysis of Secure Internet of Things Protocols, in: Proceedings of the 33rd Annual Computer Security Applications Conference, ACM, 2017, pp. 238–249.

[85] F.L. Coman, K.M. Malarski, M.N. Petersen, S. Ruepp, Security Issues in Internet of Things: Vulnerability Analysis of LoRaWAN, Sigfox and NB-IoT, in: 2019 Global IoT Summit (GIoTS), 2019, pp. 1–6, doi:10.1109/GIOTS.2019.8766430.

[86] J.Y. Kim, W. Hu, D. Sarkar, S. Jha, ESIoT: Enabling Secure Management of the Internet of Things, in: Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 219–229, doi:10.1145/3098243.3098252.

[87] J. Cao, P. Yu, M. Ma, W. Gao, Fast Authentication and Data Transfer Scheme for Massive NB-IoT Devices in 3GPP 5G Network, IEEE Internet of Things Journal 6 (2) (2019) 1561–1575, doi:10.1109/JIOT.2018.2846803.

[88] Y. Qiu, M. Ma, An Authentication and Key Establishment Scheme to Enhance Security for M2M in 6LoWPANs, in: 2015 IEEE International Conference on Communication Workshop (ICCW), 2015, pp. 2671–2676, doi:10.1109/ICCW.2015.7247582.

[89] Y. Qiu, M. Ma, Secure Group Mobility Support for 6LoWPAN Networks, IEEE Internet of Things Journal 5 (2) (2018) 1131–1141, doi:10.1109/JIOT.2018.2805696.

[90] V.G. Vassilakis, I.D. Moscholios, M.D. Logothetis, M.N. Koukias, Formal Verification of Key Establishment for Dual Connectivity in Small Cell LTE Networks, 2017. The sixth International Conference on Communications, Computation, Networks and Technologies INNOV2017.

[91] J. Cao, H. Li, M. Ma, Y. Zhang, C. Lai, A simple and robust handover authentication between HeNB and eNB in LTE networks, Computer Networks 56 (8) (2012) 2119–2131.

[92] C. Lai, H. Li, R. Lu, X.S. Shen, SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks, Computer Networks 57 (17) (2013) 3492–3510, doi:10.1016/j.comnet.2013.08.003.

[93] B. Stojanović, K. Hofer-Schmitz, Formal Methods for Connected Vehicle Protocols, in: 2019 27th Telecommunications Forum (TELFOR), 2019, pp. 1–4, doi:10.1109/TELFOR48224.2019.8971034.

[94] D. Basin, J. Dreier, L. Hirschi, S. Radomirovic, R. Sasse, V. Stettler, A Formal Analysis of 5G Authentication, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS '18, ACM, New York, NY, USA, 2018, pp. 1383–1396, doi:10.1145/3243734.3243846.

[95] C. Cremers, M. Dehnel-Wild, Component-Based Formal Analysis of 5G-AKA: Channel Assumptions and Session Confusion, Network and Distributed Systems Security (NDSS) Symposium 2019, Internet Society, 2019.

[96] J. Zhang, Q. Wang, L. Yang, T. Feng, Formal Verification of 5G-EAP-TLS Authentication Protocol, in: 2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC), 2019, pp. 503–509, doi:10.1109/DSC.2019.00082.

[97] J. Zhang, L. Yang, W. Cao, Q. Wang, Formal Analysis of 5G EAP-TLS Authentication Protocol Using Proverif, IEEE Access 8 (2020) 23674–23688, doi:10.1109/ACCESS.2020.2969474.

[98] Y.E.H. El Idrissi, N. Zahid, M. Jedra, An Efficient Authentication Protocol for 5G Heterogeneous Networks, in: International Symposium on Ubiquitous Networking, Springer, 2017, pp. 496–508.

[99] A. Koutsos, The 5G-AKA Authentication Protocol Privacy, in: 2019 IEEE European Symposium on Security and Privacy (EuroS P), 2019, pp. 464–479, doi:10.1109/EuroSP.2019.00041.

[100] F. Conceicao, N. Oualha, D. Zeghlache, Security Establishment for IoT Environments in 5G: Direct MTC-UE Communications, in: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017, pp. 1–5, doi:10.1109/PIMRC.2017.8292693.

[101] K. Mladenov, S. van Winsen, C. Mavrakis, Formal verification of the implementation of the MQTT protocol in IoT devices, 2017, SNE Master Research Projects 2016-2017.

[102] S. Hernández Ramos, M.T. Villalba, R. Lacuesta, MQTT Security: A Novel Fuzzing Approach, Wireless Communications and Mobile Computing 2018 (2018) 1–11, doi:10.1155/2018/8261746.

Dr. Katharina Hofer-Schmitz is a senior researcher of the Competence Group Cyber Security and Defence at DIGITAL - Institute for Information and Communications Technologies at JOANNEUM RESARCH. She holds a doctoral degree (PhD) in Mathematics. Her current research focus is cyber security, especially intrusion detection, advanced persistent threats, security by design and the application of formal verification methods and machine learning to problems in that field.

Dr. Branka Stojanović is a senior researcher of the Competence Group Cyber Security and Defence at DIGITAL - Institute for Information and Communications Technologies at JOANNEUM RESEARCH. She graduated in Telecommunication Engineering and obtained doctoral degree (PhD) in Electrical Engineering and Computer Science, from the School of Electrical Engineering, University of Belgrade, Serbia. She is CISSP certified. Her field of research activities covers signal processing algorithms, network security, forensics, biometrics and machine learning. The most recent focus of her research interest is the application of machine learning technologies for cyber security problems in security at runtime and security by design applications.