SOME SECURITY FEATURES OF SELECTED IOT PLATFORMS

ROBERT KAŁASKA[1] AND PAWEŁ CZARNUL[2]

[1]*robert.kalaska@gmail.com*

[2]*Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology*
Gabriela Narutowicza 11/12, 80–233 Gdansk, Poland

(received: 21 November 2019; revised: 17 December 2019;
accepted: 23 December 2019; published online: 8 January 2020)

Abstract: IoT (Internet of Things) is certainly one of the leading current and future trends for processing in the current distributed world. It is changing our life and society. IoT allows new ubiquitous applications and processing, but, on the other hand, it introduces potentially serious security threats. Nowadays researchers in IoT areas should, without a doubt, consider and focus on security aspects. This paper is aimed at a high-level review of the existing IoT enabling standalone middleware solutions and frameworks in terms of potential application areas, architecture and components, communication APIs as well as support for key security features including access control, support against attacks on service, device authorization and data filtering. On the one hand, it allows the developer to choose the middleware best matching their needs. On the other hand, it can serve as a starting point for further research on middleware security features based on the provid ed security related open areas and challenges.

Keywords: IoT security features, IoT architecture, IoT platforms, IoT middleware
DOI: https://doi.org/10.17466/tq2020/24.1/c

1. Introduction

We have been able to observe stages of Internet evolution in computer science and technology. We can divide the phase of growth into 4 stages. Web 1.0 which started in the 1990s had a very poor graphical environment and was designed mainly as text enabling resources. Its main feature was to deliver information to the reader. The reader had a passive state there, so they could consume information without interaction. The next phase, Web 2.0, gave an opportunity to exchange information through web forums and other tools which allowed the user to give feedback on the presented information. Web 3.0, still in use, allowed full interaction between the user and the web content. Web 3.0 allows processing information passed from users or gathered on other sites in an

intelligent way. It is possible with the use of server applications and currently very advanced front-end technologies. Currently, we are on the verge of Web 4.0, which could be also named as IoT. It has various descriptions assumed by researchers over the last years. These can be focused on interconnected objects concept, interoperability of things having identities in a smart space within a given context [1], allowing interconnection of things at any place, with anything and anyone using any network and any service [2]. Finally, another definition focuses on sharing information among devices using a unified framework allowing ubiquitous sensing and data analytics [3].

For the following consideration and survey we focus on the last of the aforementioned definitions as it is more abstract than the others. It does not describe each context precisely but it defines that the context of IoT is composed by time, place, thing or person and a path to access it as a service resource. The other definitions are good descriptions of the IoT concept but as IoT is abstract itself so we believe that its definition should not be very concrete and should not limit future technological solutions.

The era of IoT started with WSN (Wireless Sensor Networks). The concept of WSN was to create a system of interconnected objects which could help in peoples' lives and business [4], [5], [6]. The main components for WSN are RFID (Radio-frequency identification) components and equipment. There have been some successful implementations of WSN in warehouses, agriculture or whole supply chain systems such as [7], [8] or [9]. The important difference between WSN and IoT is that WSN are focused on incorporation of RFID hardware and create systems for specific goals, while IoT is more abstract and in its concept it will not be a concrete system but rather a platform to build on heterogeneous devices and subsystems, which everyone will be able to use according to their own needs and to achieve their goals.

In the research area we can find information which mix IoT and WSN as one concept such as [10]. As we have stated before there is a need to distinguish these two concepts. WSN are closer system solutions while IoT is going to be a heterogeneous network, which is not aimed at a specific goal. As stated in [11] WSN may be a part of IoT but not vice versa.

This paper aims at a review of selected IoT middleware solutions in terms of security. We have prepared a brief description of middleware systems developed over the last eighteen years and compared them in terms of security features. The papers starts with a short description of an IoT architecture. Then, security aspects in the layers of the discussed architecture are considered. Subsequently, selected middleware solutions are compared by key security features and a short discussion regarding the latter in IoT middleware systems is included.

2. IoT Architecture

The architecture of IoT is described in [12], [13] and [3]. A short summary will serve as a good starting point for the following sections. A holistic view of

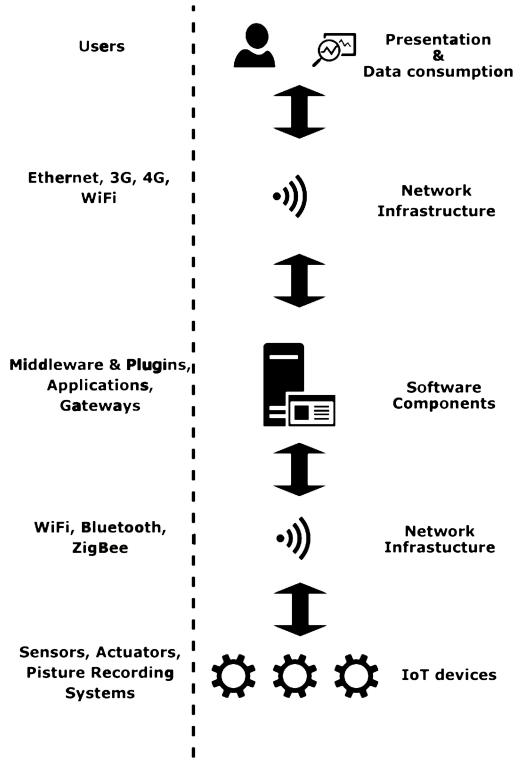

Figure 1. Holistic view of IoT architecture

an IoT architecture should take into account several components such as users, devices and software as depicted in Fig. 1.

For the sake of our survey it is crucial to consider the following key IoT layers (Fig. 2):

1. Perception layer;
2. Network layer;
3. Middleware layer;
4. Application layer;
5. Business layer.

In the next section we discuss security issues for all these 5 layers. This partitioning is natural for IoT. The first layer describes devices. IoT is built on thousands of interconnected components. These devices could be passive like RFID tags, a simple active device equipped with a battery or an external power source but needing an external gateway to communicate with the Internet or even more complex devices with their own communication stack. The main target for these components is to gain raw data and execute commands (e.g. open a window, switch the light, etc.), hence, it can be called the Perception layer. Then, all of the available data needs to be accessible to other devices, thus, it is necessary for devices to be interconnected. This part is accomplished by the

network layer. Having access to all the data and devices, the architectural solution should provide a layer for managing and filtering data, properly connect raw data with its context to get the information, provide access and authorization, handle security issues and give access to the gathered data through various interfaces. All of these functions are assigned to the middleware layer. On such a stack, companies may build their own applications which consume information from the middleware layer. This is done within the application layer and finally applications achieve business goals at the business layer located at the top of the stack. Fig. 2 presents the data flow and interactions between layers.

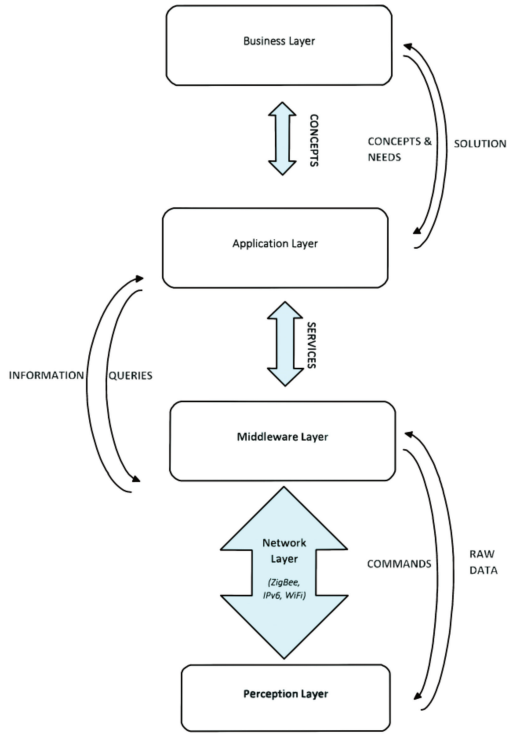

Figure 2. Layers and information flow

From the implementation point of view, we need to provide a concept for interfaces between layers. The perception layer and the middleware layer are connected through the network layer. These may be achieved through various protocols like ZigBee, WSN, IPv6 or WiFi. As we have pointed out before, the connection between the middleware layer and the application layer should be achieved through interfaces. It has been stated [14] that the best way to handle queries on this layer is to implement the SOA (Service Oriented Architecture). We are not going to focus on this issue in our article. Advanced surveys covering this aspect can be found in articles [15], [16] or [17]. The data flow between the application layer and the business layer has raised its own technical issues

which are not the target of this paper. In the next section we will focus on the middleware layer with the SOA approach. In general, the SOA approach can be used to integrate distributed processing at several levels i.e. to gather data from IoT devices and process it in parallel using high performance computing systems or clouds [18]. Such processing can be arranged into workflows with proper synchronization and parallel processing of workflow nodes that can denote data acquisition, partitioning, processing and merging [19, 20].

3. Security

Information security [21] in the context of a secure computer system means that each of the aspects such as confidentiality, integrity of information and availability needs to be covered. Confidentiality means that sensitive information is protected against unauthorized access, integrity means that all data stored in a computer system must not be corrupted (even by getting unauthorized access, device failure or accidental changes) and availability denotes that information may be accessed at any time. A broader discussion can be found in [22].

Security is one of the most important aspects in the IoT. As a global network capable of reaching every device in peoples' lives it has many security issues to address. Some may be critical for our health, other for our property and our privacy. It could even be dangerous for the security of a country or a society. As an example, we can imagine an e-health [23] system which doses a drug. An attacker may change the assigned dose and damage the condition or even the life of a patient. An example concerning property could be getting inside a home control system [24] and unlocking the door or turning off the surveillance system. In terms of privacy, somebody could take control over our CCTV (closed circuit television) cameras or gain access to our shared information, which should be accessible only to a group of trusted individuals. On the country level we can imagine a situation where an attacker sends fake notifications to a power station (as a consumer) so it reduces the power while the real power consumption by customers is very large, as a result the power station turns off.

The number of attacks in solutions which may be a part of IoT in the future has increased significantly over last years. In [25] we can find out that in 2013 there were 2 crucial incidents, in 2015 there were 4, in 2016 there were 7, while the first quarter of 2017 began with 3 serious issues. Article [26] comments on another report prepared by Symantec, pointing out that the number of attacks in IoT increased by 600% between 2016 and 2017 and that many users continued to use older operating systems. It is important to inform people how much these can affect their IoT systems. As commented in [27], based on a report of Zingbox, 41% of security issues in medical IoT systems were due to bad user practices and we can expect misuses of devices connected to the Internet everywhere.

IoT as a concept faces many difficulties concerning security resulting from its nature. The European Union Agency for Network and Information Security

prepared a report [25] based on information gained from consultations with many security experts. This report points out 12 main difficulties in IoT security:

1. Very large attack surface — in terms of data security (IoT can process peta bytes of data associated with different kinds of human activities) and in terms of a geographically distributed architecture. It potentially makes it possible or easier to find out new, and potentially not yet secured paths for an attacker.
2. Limited device resources — sufficient cryptography operations for establishing a secure environment is currently very hard on such devices as sensors which have limited hardware capabilities.
3. Complex ecosystem — to protect IoT solutions it is important to recognize vulnerabilities in the whole environment as an ecosystem, this can require specific knowledge which may be difficult to achieve while securing a selected IoT component or layer only.
4. Fragmentation of standards and regulations — technical standards and legal regulations do not always catch up as fast as new solutions and technologies are provided to the market, so it is hard to derive common standards.
5. Widespread deployment — currently deployed solutions can be extended to work with IoT, this connection can result in new security threats.
6. Security integration — many device and software vendors may use different security approaches which leads to differences not only in implementations but even with security levels as well.
7. Safety aspects — in terms of cooperation with the physical world and threats such as invalid movements of robo arms or hacking the traffic lights.
8. Low cost — to achieve higher revenues companies may cut costs associated with security developments.
9. Lack of expertise — IoT is still in at an early stage so we have no sufficient surveys on its deployment and still not many researchers have suitable knowledge. It is important if new threats are recognized and fixed before an attacker exploits them.
10. Security updates — a growing IoT market will find out new security issues. Providing all security updates should work automatically with no human intervention.
11. Insecure programming — cutting development costs and desire for short time to market may lead to many security vulnerabilities. Developing IoT solutions needs many security tests and deep analysis of possible security issues.
12. Unclear liabilities — as IoT is a very complex and decentralized network it is hard to define liabilities for security issues.

Detailed descriptions of each difficulty can be found at [25]. We can extend this list with 3 additional points. Any device in IoT can be deployed in any place. Getting physical access to a device would be very easy, so we propose to add *"Unauthorized physical access"*. Another aspect which comes with getting

unauthorized physical access to devices is *"Data uncertainty"*. We should take into account the fact that a sensor can be cheated by adding physical disturbances to its surrounding. One more issue for the whole system stability is *"Testing difficulties"*. It is very hard to test the whole IoT solution and check all possible software and execution paths. It is a big challenge for QA (quality assurance) teams. These 15 points are, as we have called them, difficulties. While there is no way to fully address them (at the time of writing), we should take them into consideration while developing IoT solutions and, where possible, try to minimize the corresponding risk.

Data is transferred and processed in each layer, so if an attacker has access to low level data, he or she can have strong influence on the whole solution built on IoT. Security measures in IoT need to be considered in each layer to be reliable. A similar concept can be found in [28].

In the perception layer we can consider the following security issues [29], [30]:

- firmware modification;
- misrepresentations of sensor data;
- lack of power or damage;
- unauthorized devices.

Firmware modification lets attackers use a device for their own goals and get inside a network. It can be avoided by using code signing and locking the write operation to the device memory. Unlock could be available only after getting authorized access (e.g. lock with a code). Misrepresentation of sensor data may occur due to the sensor failure or a physical operation on a device (e.g. falsely heating up a temperature sensor). There is no way to cope with this issue to a full extent, but there are some methods to manage it. Data from a sensor should be filtered by comparing it with the expected values and history data. Simple operations like comparison with expected values can be done on the device side, while computations requiring more resources can be done on the middleware side or by the first device available to perform such an operation. Lack of power or damage may occur at any time. In those situations we should use redundant solutions for critical devices to give enough time for maintenance. Unauthorized devices may potentially connect to an IoT network and disturb the network or send invalid information. In such situations there is a need for creation of a trust chain of producers. It could be a solution similar to the public key infrastructure. Devices should be able to get information about other devices without human interaction.

The network layer has different security issues [29], [31]:

- interference in packages;
- eavesdropping;
- jamming.

The aforementioned issues are well known in the network security area so nowadays we have good solutions to deal with them. Interference in packages (e.g.

man-in-the-middle attack) or eavesdropping could be solved by using encrypted connections (e.g. SSL (Secure Socket Layer)). The problem is that many IoT devices do not have enough resources to perform cryptography operations. Therefore, there is a field for future research. Moreover, every kind of network can be disturbed. Jamming is an important issue for wireless networks, because everybody who has sufficient hardware may disturb the network stability. There is no simple and common solution to counteract. Radio frequencies should be controlled by an independent institution which should be prepared for such a situation and be able to react fast.

The middleware layer is the brain of IoT. We can distinguish the following security issues [32], [29], [30]:

- unauthorized access;
- attacks on service (e.g. DoS, DDoS);
- unauthorized devices;
- misrepresentation of sensor data.

Unauthorized access has a strong influence on the whole system. It applies to access to services delivered by the middleware as well as access to administrative options or the machine(s) where the middleware has been deployed. Well established access management policies should be used in middleware applications and in the company which delivers and hosts such solutions. Attacks on a service should be handled by applications and devices for network security like WAF (Web Application Firewall) and IPS (Intrusion Prevention System). Similarly to the perception layer, in the middleware layer we should be able to recognize and trust devices in the IoT network. Mechanisms similar to those in the perception layer can be implemented. Misrepresentation of sensor data may be well served in the middleware layer. In this layer historical data can be collected and compared with the read values and computations can be performed to ensure that such data is valid and can be used by applications.

In the application layer we can outline the following security issues [32]:

- unauthorized access;
- program changes;
- data corruption;
- endpoint changes.

Similarly to the middleware layer unauthorized access is the most important issue and could be resolved in same way as in the lower layer. Program changes apply to similar issues as firmware modification in the perception layer. There should be efficient ways for protecting the application against adding unknown components and overriding the existing libraries. Avoiding data corruption can be achieved through backup solutions. There is also another important issue. The attacker may change the endpoint destination to the service which we are going to use. We should use cryptography solutions while using services, for example using WSS (Web Services Security) in WSDL (Web Services Description Language)

communication or rely on an SSL connection using REST (Representational state transfer) solutions. Moreover, these technologies protect us against network security issues from the middleware to the application layer.

In the business layer there are security issues connected with the business value such as:

- concept stealing;
- customer data security.

In this article we are not going to focus on business security.

To sum up the above description we organized possible security functionalities (in reference to Section 4) in each layer. These are presented in Table 1.

Table 1. Map of security functionalities and layers

		Layers			
		Perception layer	Network layer	Middleware layer	Application layer
Functionalities	Access Control and authorization	✓		✓	✓
	Support for preventing network attacks		✓		
	Device authorization and secure connection	✓	✓	✓	✓
	Data filtering and integrity	✓	✓	✓	✓

4. Description of analysis method

We start with a review of middleware solutions which were objects of past research, and then we go through middleware systems which are open source and their functionalities are still improved by the community. For our survey we chose 37 middleware platforms. As we have stated before our goal is to focus on the security aspects of the delivered solutions. In line with the previous section we will distinguish middleware solutions by their functionalities in security:

- Access control and authorization;
- Support for preventing network attacks (eq. Denial of Service (DoS));
- Device authorization and secure connection;
- Data filtering and integrity.

The above functionalities should cover trust challenges in IoT such as data security, user privacy, distributed trust models, attestation, privacy and policy preservation, encrypted search and policy management. We could associate these functionalities with addressing relevant challenges as presented in Table 2.

Additionally, in the comparison we include the year of publication of the article concerning the given solution to check if there is a correlation between security features in the solutions in terms of release dates.

Table 2. Map of challenges and functionalities

		Challenges					
		Data security	User privacy	Distributed trust models	Attestation	Encrypted search	Policy management
Functionalities	Access Control and authorization	✓	✓	✓			✓
	Support for preventing network attacks	✓					
	Device authorization and secure connection	✓			✓	✓	
	Data filtering and integrity	✓					

The selected platforms were developed with various purposes in terms of the intent of use. In Tab. 3 we sum up some of the key information about the described projects. In the following section we shortly describe the features of each of these platforms focusing on those related to security, wherever possible.

5. Review of selected IoT platforms

In this section we review several current and past IoT enabling solutions. The survey contains selected standalone middleware solutions as well as frameworks for IoT. From the point of view of this paper all of these can be considered middleware solutions. We focus on the security components. In the research area we can find information about over 50 middleware solutions, mainly based on research papers. There are some open source solutions, mostly developed for business purposes but some of those like [33] or [34] were developed in cooperation of scientific community and business communities.

5.1. Overview of selected solutions

Solutions chosen for our survey can be divided into 2 groups depending on the architecture and usage:

- Framework — integration requires additional libraries or writing one's own wrapper components.
- Standalone middleware — offers well-known and commonly used APIs (application programming interfaces).

5.1.1. Framework solutions

1. Smart Messages [35] is a component for user defined applications. It runs a user program on nodes of interest. Its architecture is based on a network of virtual machines. It works in the whole network, passing the state and commands to neighboring cells. From the security point of view, it covers a very good access control model between cells in the network. It is built on the

protection domain model. There are five domains: Owner, Family, Origin, Code and Others. The protection domains are stored in an access control list for each SM (Smart Message). This mechanism allows the system to decide which SM is able to operate on each tag. The authors of the solution have deployed an application called EZCab which is designed for locating and booking free cabs.

2. MiLAN [36] is a solution designed for best energy and QoS management. It is based on a network of devices running a scaled down version of MiLAN. Its design focuses on getting the longest sensors battery lifetime while meeting the required QoS parameters. The QoS (Quality of Service) parameters are defined in user applications which are built on top of MiLAN. There is no direct information about security solutions in this middleware that we can take into consideration for comparison. The authors of MiLAN have deployed a personal health monitor application as a showcase solution.
3. MoCA [37] is a middleware system developed in a publisher-subscriber model. It focuses on developing and deploying context-aware applications on mobile devices. It is built on 4 main services which are responsible for configuration, context information management, discovering and location information management. We could not find any security related information implemented in the middleware layer. Two applications have been developed as use cases for the middleware. W-Chat is a chat program which is able to catch messages for the user which temporarily disconnects. Another application is NITA. NITA is an application for posting messages for predefined regions. It allows users who enter the region and have valid authorization parameters to read information.
4. MidFusion [38] is a middleware architecture very similar to MiLAN. The goal of the authors was to develop a solution which would support the best sensor usage while ensuring user QoS parameters. The sensor selection is based on a Bayesian decision problem. The main difference is that it gives a possibility to discover new sensors by running an application, while in MiLAN, the application should know *a priori* what sensors are available. We could not find any information related to the security features. An application for detection of an intruder in the building has been deployed using this middleware.
5. Mires [39] is a solution based on the publish-subscriber architecture. It has no predefined security solutions, but we can treat one of its functionalities as a security feature in terms of data uncertainty. Mires has an aggregation policy, where the user can define the stop criteria. It allows the user to predefine a data integrity filter. No more information about security can be found. To illustrate the middleware features, the authors have proposed an application for visualization of a network response for subscriber requests in selected topics. The application user may set the topics of interest and rules for requesting data (e.g. every 5 minutes).

6. Sensation [40] is a database based solution. The main idea for Sensation is to abstract a data model from an application in a similar way as JDBC (Java Database Connectivity) works in Java. It is designed to work with different WSNs through a USL (Unified Sensor Language). An application developer can use its resources through a delivered API. Sensation does not include any security solutions.
7. SwissQM [41] is based on a database approach with sensors running as virtual machines. Its main target is to offer better support for data management and to decrease traffic in a network using adaptive sampling. This technique may lead to extended battery lifetime, same as with other approaches. Unfortunately, the system has no security solutions proposed by the authors.
8. TS-Mid [42] is middleware running on top of TinyOS. It is based on the JavaSpaces technology. It lets the whole wireless sensor networks be divided into regions. Middleware may then operate on regions which are built out of predefined sensors. The main goal which can be achieved is decreasing the traffic in the network. This solution has no security features. The authors have proposed a showcase application for monitoring sensor states. The user may query a chosen region for interesting data through a graphical interface.
9. HYDRA [43] is middleware which combines a service oriented architecture and a model driven architecture. It is divided into application and device components. The HYDRA architecture was developed with focus on security. It has a semantic resolution of security which was proposed in [44]. It allows HYDRA to combine all the security features such as access control, data integrity and privacy. No detailed information could be found due to closed project websites.
10. COPAL [45] is a middleware solution which focuses on context provisioning. It works along the publisher-listener architecture. Each context component contains the required attributes and optional attributes which allow context processing to assign the user to the selected group. The main feature of COPAL is the COPAL-DSL component which helps developers to build an application. Engineers may create a model with COPAL-DSL. This model allows generating code skeletons and deployment artifacts. COPAL has an Authorization feature. Context information can be available for chosen services only. No more security features were found in the literature.
11. Leshan [46] is a framework built as an implementation of the LwM2M (Light-weight Machine to Machine) protocol. It focuses on providing device management and service enablement. Its architecture is divided into an LwM2M client which operates on an end device and LwM2M servers. Its main functionalities are device configuration, bootstrapping, firmware update, diagnostics, connection management, control, data reporting, lock and wipe. Security features are divided into credential related procedures where available solutions are certificates, preshared keys, public raw keys

and PKI deployments and a second feature group with security paths where the possible options are DTLS (Datagram Transport Layer Security), SMS (Short Message Service), DTLS over SMS, OSCORE (Object Security for Constrained RESTful Environments). The description of the LwM2M protocol also defines its security abilities which replace initial keys during the bootstrap procedure, multiple server deployment with different credentials and also providing security for every path.

5.1.2. Standalone middleware solutions

1. Cougar [47] is a middleware solution which uses a database approach for managing sensor data. As in the database approach, the main goal was to manage a sensor network with declarative queries. Researchers delivered a complete solution but they did not focus on security issues. The only aspect that has been taken into account is data uncertainty. Cougar allows collecting raw data from many sensors and compute an average value in the logical leader node, so it can avoid measurement errors. The architecture of Cougar is a loosely-coupled distributed architecture. Each device has an abstract query proxy layer which interacts with the routing layer and the application layer. Two demo applications QueryProxy Demo and WaveGUI Demo [48] were deployed using Cougar.
2. DSWare [49] is a solution which focuses on data integrity and minimizing the traffic in the network. The application layer may work with middleware through services. We could not find any information related directly to security, but the middleware has many algorithms to protect data. The most important one from the point of view of security of data and avoiding data uncertainty is grouping several sensors into one logical sensor, which lets the software take the decision which information is correct. Its architecture is organized into services which abstract lower layers of the IoT network for the application layer. It is divided as follows: data subscription, event detection, data storage, group management, data caching and scheduling. The middleware was tested on a demo application for real-time vent detection.
3. IrisNet [50] is a database approach solution. The solution is targeted at efficient querying. It is based on XML files using an XPATH resolver mechanism. Queries to the system are sent with senselets — short instructions to operate with data through a network. It covers security issues in several ways. Senslets can be digitally signed or not. This solution allows determining the query author and allows using full data or a limited set according to the security policy (trusted or not). Moreover, it has a mechanism to back up cached data in two ways. The same data is kept by a nearby sensor and in a far away one. When the main sensor fails, the data (historical) can be gathered from another replica. Researchers built three test solutions based on IrisNet: a parking-space finder for tracking parking spaces, a co-

astal imaging service allowing tracking nearshore phenomena, IrisLog — a service for network and host monitoring.

4. CoBrA [51] is a context broker middleware solution. It focuses on how to provide the best architecture to allow system components to work together and understand each other in different contexts. One of the four key features of this solution is to protect user privacy. It has been achieved with a very advanced privacy policy solution. It is based on a SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications) policy ontology. In this system the developer can define privacy on each context level such as location, friends, time and many more. Moreover, the user can adjust the granularity of information which can be accessed. Even if somebody joins the network, they will not get access to information which is not shared by the user. During the CoBrA project several applications have been deployed including EasyMeeting, CoBrA Demo Toolkit and CoBrA Text Messaging Commands.
5. Global Sensor Network (GSN) [52] is a middleware platform which is based on a virtual sensor solution. Each node has its virtual sensor representation which abstracts from implementation details. It is based on the publish-subscriber model which performs data streaming through SQL-based queries. It contains two important security functionalities implemented in two architecture components: integrity service and access control. The integrity service provides data consistency through electronic signatures and encryption while access control enables entitled parties to use the system. For demonstration purposes the authors prepared a configuration with four sensor networks which could be queried through a Web interface. The platform is available under the GNU GPL license and can be downloaded from [53].
6. e-SENSE [54] [55] is a middleware solution which is based on the subscriber-consumer model. It allows sharing data from WSN to B3G (Platforms Beyond 3G) networks. It has a cluster formation algorithm which forms sensor groups into clusters. The solution is quite similar to TS-Mid [42]. In [54] the authors present a Security Manager but there is only information about the available policy management (which is used for configuration issues) and Access Control between sensors and higher layer gateways. In [55] the authors tested an application for recognizing human activities by using wireless sensor nodes worn on the body and integrated into working tools.
7. DAViM [56] is built with an architecture based on a virtual machine approach. It allows users to define their own application using services. Services can be defined to operate on different sensors performing an operation defined by the network owner. Its sole feature, from the security point of view, is isolation of multiple applications. All applications run on their own virtual machines which limits interference between them. For demonstration purposes, the authors have deployed a surge application which sends sensor readings to a base station periodically.

8. CroCo [57] is a middleware which focuses on context management. It uses an ontology through the XML description. Privacy is supported through the Privacy Enforcer. This component checks whether the chosen resource may have access to the queried information. No more security features are described. In [57], we could find information about two example applications deployed on CroCo. One is Personal Document Management for managing documents and the other is the Adaptive Co-Browsing Application which can be defined as an extension for traditional web browsing.
9. MUSIC [58] is a middleware platform which aims at service oriented environments. It has a service oriented architecture built on OSGi (Open Services Gateway initiative). It manages the available services and provides self-adaptation for service management for best achievement of a given application goal. No information about security is provided. For a demonstration scenario, the authors have proposed a composition of services for location, map, route in a travel assistant working with an InstantSocial application.
10. SPBCA [59] is an architecture based on Tuple Space and Ontology. The solution focuses on unifying the interaction process in a heterogeneous network. It runs on each device (agent based middleware). From the security point of view it has several solutions that improve security. Agents are assigned to spaces (similar to TS-Mid). In this way access control can be improved. A component called TS Manager manages the privacy of each space and handles data consistency. No more security features are described. The authors of the solution have proposed a GUI for run-time service customization as a showcase application.
11. Feel@Home [60] is a middleware solution which aims at resource sharing. It focuses on context management across a domain-specific environment. It has access control which is provided on two layers. It starts from the domain level and checks if the requested data is accessible, checking the target object privacy policy. It has no more security solutions.
12. UbiROAD [61] is a middleware architecture designed especially for driving purposes. It is an agent based solution which aims to interact with cars, its embedded devices, humans as well as the road infrastructure, and traffic systems. The authors of the system envisage a semantic ontology-based solution to build a trust management system. There is no detailed information how it could be implemented and the architecture does not provide any other security solutions.
13. KASOM [62] is a middleware solution focused on knowledge awareness. It is service oriented middleware built on three main components: framework services, knowledge management services and communication services. For security reasons it has a component called Security Service which manages major security issues. No detailed information about this component could be found. An application for testing has been deployed in an e-health scenario. The authors built a healthcare telemonitoring application on

KASOM and deployed it on real WSAN (wireless sensor and actuator network). An experiment showed that it had positive influence on the quality of service of a Sanatorium.

14. UbiSOAP [63] is a middleware tool which focuses on implementing a better way of communication using SOAP (Simple Object Access Protocol). It allows many network interfaces to work together and to keep track of interconnected devices even when they change their connection points. It has no security features. UbiSOAP has been released under an open source license. Three showcase applications have been deployed on UbiSOAP — *Pocket doctor*, *Field service management*, *Crisis management system*.
15. MOSDEN [64] is a middleware solution based on Android. It provides sensing as a service model. The main goal is to provide sustainable data collections on IoT devices. It allows keeping information about sensor capabilities and health instead of collecting whole sensor data. There is no information about security features for this architecture.
16. IoTSyS [65] is a middleware solution based on a service oriented architecture. It is designed mainly for home automation purposes. Security features are granted with access control delivered with a public key infrastructure and security policies (XACML (eXtensible Access Control Markup Language) — based authorization). Safe connections are established with SSL. Data integrity between nodes has been achieved using the AES (Advanced Encryption Standard) encryption. IoTSyS is distributed with an open source 3-Clause BSD License. The project archive is available under [33]. Applications for home automation including HVAC (Heating, Ventilation, Air Conditioning), alarms or light control systems have been deployed using this infrastructure.
17. PRISMA [66] is a service oriented middleware solution. Its main feature is the REST based architecture so the data from devices is accessible like other web resources through HTTP requests. It has one similar feature to MiLAN, which is an algorithm for extending the lifetime of batteries of devices. Similarly to MiLAN it could be considered as an additional feature for security. No other security functions could be found.
18. EMMA [67] is an agent based, resource oriented architecture solution. The goal of the authors was to develop a system for reducing data transmission in an IoT network through services running on each node over Contiki OS. EMMA uses CoAP (Constrained Application Protocol) to communicate with nodes and publishes as REST resources. There is no information provided about security features.
19. RERUM [68] is a middleware product focused on security. It is based on another IoT middleware solution called OpenIOT which is a cloud platform. RERUM provides many security features. It has a solution for access control. The device security is provided with a mechanism which maps a physical device address to a generated key associated with a virtual device. All

communication packets are signed with electronic signatures based on ECC (Elliptic Curve Cryptography) cryptography. A data granularity mechanism is used to protect a piece of information, so that the end user may define general actions like 'when too cold, turn on heating' while not getting information about the whole sensor data and network architecture.

20. EMMA [69] is a publisher-subscriber solution which is based on MQTT (MQ Telemetry Transport). It focuses on minimizing the network traffic. The authors have proposed a solution for dynamic network changes due to a reconfiguration of connections. It is based on buffering and tunneling the MQTT traffic for background middleware and reconnects to the node when it is needed. There are no security functionalities in the currently available solution. The authors have prepared a simulation where energy tokens are shared in a concurrent environment like a balanced energy consumption in a smart home.
21. Xively [70] is a part of Google Cloud IoT. Its architecture is divided into 3 layers: *edge devices* which is a connector between devices and upper layers, *data analytics* in the cloud for storing and filtering data and the last one — *data usage* for subsequent data processing. It has a very strong authentication design including a secure device on every node which could perform cryptography operations. Connecting JWT (JSON Web Token) with a signing operation it gives a lightweight way to open a two-way secure connection. It has been deployed in agriculture, home automation or even remote support solutions.
22. Kaa [71] is an open source platform for IoT. It is a message based solution supporting MQTT, CoAP and other protocols using a gateway architecture. Communication channels can be secure or open. Secure channels are encrypted. It also has a solution for device management. Each device has to present valid credentials. Credentials may be pre-shared keys, tokens or a login and password combination. There is also a solution for data consistency which is based on simple response codes, but it allows devices to ensure that the sent data has been properly received. On the project page we could find many possible use cases for Kaa like agriculture, automotive, healthcare, IIoT, smart building and cities or even sport & fitness.
23. SiteWhere [72] is another open source platform. The current release in version 2.0 is based on a microservices architecture and all communication is carried out with REST APIs. The JWT is used to grant access control and each REST call has a security header containing a node id and an authentication token. The whole solution is based on Spring Boot deployed on Docker, so it can be easily deployed with redundant services which is a good solution against DoS attacks.
24. DeviceHive [73] is an IoT middleware solution. It communicates via MQTT, WebSocket and REST. Similar as SiteWhere, it has a microservices architecture. It uses TLS (Transport Layer Security) encryption for secure

connection. Access control is granted with a role-based security model and authentication with JWT tokens. DeviceHive can be used across many industrial areas like Automotive, Energy & Utilities, Smart Environments (cities, building or agriculture), IIoT, Insurance and more [74] with JWT tokens. DeviceHive can be used across many industrial areas like Automotive, Energy & Utilities, Smart Environments (cities, building or agriculture), IIoT, Insurance and more [74].

25. FIWARE [75] is context broker middleware which is built on different components called Generic Enablers. Information can be gathered using FIWARE NGSI API. It is an URI (Uniform Resource Identifier) based API using HTTP requests. Access control is granted with the Access Control Generic Enabler (e.g. Keyrock using tokens or AuthZForce using XACML). Communication with device layers is established using IoT Agents or with external services via System Adapters. There are subsystems like STH Comet, Cygnus or Draco for data persistence. Security in FIWARE is well-established with the provided Generic Enablers, moreover, the user can adjust the needed security level choosing the best Generic Enablers for his or her own goals. FIWARE has been used in many fields like e-Health [76], agriculture [77] or transportation [78] and [79].
26. SgxIoTGuard [80] is a middleware solution which aims at providing data security using the Intel SGX technology. It allows executing selected operations secured by the hardware. Its architecture is divided into the SGX Trusted IoT Gateway and the IoT Middleware platform. The idea is to collect data from devices through an IoT Gateway which performs data encryption, then encrypted data can be further processed by higher level components. Its decryption is performed within the middleware platform using the same hardware mechanism. Both the Gateway and the Middleware platform consist of two modules: an untrusted module and a trusted module. The untrusted module communicates with other components while the trusted module exchanges data only with the untrusted module. This solution in its current state provides data integrity only. Additionally, using data access policy enforcement it may provide access to data for selected users or components. For test purposes, researchers built an application which established communication and data exchange with a Philip Hue Hub with a ZigBee light bulb, Samsung SmartThings Hub with a Motion/Prox-imity sensor, Dlink IoT camera, Belkin Wemo Switch, Wemo Wall socket and a heart rate monitor mobile application on Android.

6. Analysis of selected IoT platforms in terms of selected security features

Based on the reviewed solutions we created a comparison table in order to better summarize the security features. Table 4 shows which of the security features are covered by each solution. Where a security feature was covered we

Table 3. Key information regarding considered platforms

PROJECT	OBJECTIVES	MAIN COMPONENTS	API/COMMUNICATION	SAMPLE APPLICATIONS
Cougar [47]	Manage sensor network with declarative queries		XML queries	QueryProxy Demo and WaveGUI Demo
DSWare [49]	Provide data service abstraction for application layer	Data subscription, event detection, data storage, group management, data caching and scheduling.	SQL-like statements	Real-time event detection
IrisNet [50]	Efficiently query globally distributed sensors	Sensing Agents, Organizing Agents	XPATH queries	Parking-space finder, coastal imaging service, IrisLog
Smart Messages [35]	Develop component for user-defined applications to execute on nodes of interest	Tag space, virtual machine, code cache	NS - user defined application with SM as component	EZCab
CoBrA [51]	Provide context-aware middleware platform	Context knowledge base, Context Reasoning Engine, Context Acquisition Module, Privacy Management Module	Jena API	EasyMeeting, CoBrA Demo Toolkit and CoBrA Text Messaging Commands
MoCA [37]	Develop and deploy context-aware applications on mobile devices	Configuration Service, Context Information Service, Discovery Service, Location Information Service	Server API, Client API, ProxyFramework	W-Chat, NITA

MiLAN [36]	Minimize energy usage while best fitting QoS parameters	Data channels, Remote network control, Network specific local network control	Predefined set of interfaces	Personal health monitor
MidFusion [38]	Use best set of sensors while meeting QoS parameters and cost of information acquisition		NS - applications must be modeled using Bayesian networks	Detection of an intruder in the building
Mires [39]	Develop middleware in message oriented architecture	Publish/subscribe service, routing component, data aggregation service + additional services (may be implemented by application developers)		Visualization of network response for subscriber in chosen topic
Sensation [40]	Abstract data model to application using database model similar to JDBC	Profile registry, offline data storage, sensor abstraction layer	Java library	
GlobalSensor Networks [52]	Provide middleware for heterogeneity sensor environment, enabled for fast deployment and data filtering	Integrity service, Access Control, GSN/Web - Services Interfaces, Query Manager, Storage, Virtual Sensor Manager	Use wrapper classes, web services	
e-SENSE [54] [55]	Capture environmental data for beyond 3G communication with wireless sensor networks	Connectivity,middleware, management, and application		Recognition of human activities with network of body sensors

SwissQM [41]	Improve limitation in data independence and integration with higher layers	Gateway node and sensor nodes	NS - write own QM programs	
TS-Mid [42]	Decrease network traffic	LRTSpaceM, AggregateM, TSpaceM	TSpace interface	Application for monitoring sensor states
DAViM [56]	Enable dynamic service management and isolation between running applications	Application store, Operation store, VM store, VM controller, Coordinator		Application sending periodically data to a base station
CroCo [57]	Context management	Consistency Check and Reasoning, Data Management, Provision and Update		Personal Document Management, Adaptive Co-Browsing Application
MUSIC [58]	Service management	Context Manager, QoS Manager, Adaptation Manager, Kernel	Web Service, CORBA, RMI, or UPnP	Travel assistant
HYDRA [43]	Develop middleware to support networked embedded system	Context Manager, Data Acquisition Component	NS - develop own application using Context Aware Framework interfaces	
SPBCA [59]	Unify interaction process in heterogeneous network	TSManager, Reaction Manager, Reaction Listeners and Spaces,	Queries through SWRL	GUI for run-time service customization

Feel@Home [60]	Context management framework for different domain context management	Global administration server, Domain Context Manager	SOAP	
COPAL[45]	Middleware for context provision	Query Factory, Processor Registry, Context Type Registry, Publisher Registry	COPAL DSL	Cozy&Green Service
UbiROAD [61]	Middleware target to road and drivers environment	Semantic Adapters, RgbDF, RpiDF	S-APL	
KASOM [62]	Knowledge management	Framework services, knowledge management services, communication services	SOAP	E-health scenario
UbiSOAP [63]	Improve communication in IoT with SOAP	Network-agnostic connectivity layer, ubiSOAP communication layer	Web Services	Pocket doctor, Field service management, Crisis management system
MOSDEN [64]	Provide sustainable data collections on IoT device	Virtual Sensor Manager, Storage, Plugin Manager, Query Manager	Same as GSN, additionally REST	Mobile Sensor Data Engine
IoTSyS [65]	Middleware which focuses on home and building automation technologies	Browser-based Interaction, Contrl & Monitoring System, SaaS Interaction, Discovery Service, Mobile Computing	HTTP, CoAP, SOAP	HVAC, alarms or light control systems

PRISMA [66]	Provide REST abstraction, manage network resource and applications and support QoS mechanism	Application Layer (Proxy Publish and Subscribe, Web Server, Application Control Component), Service Layer (Event component, Decision component, Publish and Discovery component), Access Layer (Communication component, Topology Control Component, Data Acquisition Component, Context Monitor Component)	REST	
EMMA [67]	Develop a system for reducing data transmission on IoT network based on local services running on each node over Contiki OS	Erbium CoAP Engine, EMMA Engine	REST	
RERUM [68]	Develop middleware with strong security features	Service Manager, Federation Manager, Data & context Manager, GVO registry, GVO manager, Security Server	REST	
EMMA [69]	Minimize network traffic	Controller, Broker, Gateway, Network monitoring protocol	MQTT	Sharing energy tokens in concurrent environment
Xively [70]	Solution focus on customer needs	Edge devices, data analytics, data usage	MQTT and REST	Agriculture, home automation and others
Kaa [71]	Open source middleware platform	Control Service, Operation Service, Bootstrap Service	REST	Agriculture, automotive, healthcare and otherd

SiteWhere [72]	Open source middleware platform	Global Microservices, Multitenant Microservices	REST	
DeviceHive [73]	Open source middleware platform	Auth Service, MQTT Connector, Frontend Service, Plugin management Service, Backend Service, Internal WS Proxy, Plugin WS Proxy	MQTT, WebSocket and REST	Automotive, Energy Utilities, Smart Environments
FIWARE [75]	Build an IoT platform which is accessible with unified API and may use interchangeable components	Context Broker, Context Processing, Analysis and Visualization, Access Control, IoT Agents	With upper layers FIREWARE NGSI API, with lower layers HTTP,MQTT, LWM2M and more depending on IoT Agent	Automotive, Energy & Utilities, Smart Environments
Leshan [46]	Provide libraries to make implementation of LwM2M easier	Leshan server, Leshan client	CoAP	Transport, Industry, Smart Cities, Connected Car
SgxIoTGuard [80]	Ensure data security using Intel SGX technology	SGX Trusted IoT Gateway, IoT Middleware Platform		

put ✓, otherwise we put ×. Additional information about the covered features (especially implementation characteristics) can be found in Section 5.1.

Comparing entries in Table 4 by support for selected security features we can conclude that middlewares have become more secure over the last 17 years. Beginning with middlewares that have no support for security we end with entries which cover a minimum 2 out 5 security features. Same conclusions were also proved by [81] where authors compared search results for terms (”IoT” OR ”Internet of Things”) AND ”Middleware” performed at points with an interval of 18 months (06.2015 – 12.2016). They found that there were about 40% more articles than within the first search.

Taking into account division for frameworks and standalone middlewares in Section 5 and security functionalities mentioned in Section 4 we can conclude that framework solutions should assure at least data filtering and integration and device authorization since others can be developed at integration time while a standalone middleware should provide all security functionalities since users may treat them as a standalone, complete platform. After comparing provided features we can conclude that most of framework solutions (6 of 11) have no provided security features. Only one of the listed has full support for security and it is the *Leshan* solution. This may be due to the fact, that most of framework solutions are old ones, where main efforts were put into key functionalities. In standalone middleware solutions there is also only one platform which supports all security features. It is called *RERUM* but in this group there are fewer solutions with no security features (9 of 26) and also as in the previous group, these are mostly older platforms. Moreover, from the listed security features ”Access Control and Authorization” is the most common feature (14 of 26), the second one is ”Device filtering and data integrity” (10 of 26), while the last two features are ”Device authorization and secure connection” (4 of 26) and ”Support for preventing network attacks” (3 of 26).

Table 4. Security features in middleware solutions

PROJECT	ACCESS CONTROL AND AUTHORIZATION	SUPPORT FOR PREVENTING NETWORK ATTACKS	DEVICE AUTHORIZATION AND SECURE CONNECTION	DATA FILTERING AND INTEGRITY	ADDITIONAL SECURITY FEATURES	YEAR
Cougar	×	×	×	✓	×	2002
DSWare	×	×	×	✓	×	2003
IrisNet	✓	×	×	✓	×	2003
Smart Message	✓	×	×	×	×	2003
CoBrA	✓	×	×	×	×	2004
MiLAN	×	×	×	×	✓	2004
MidFusion	×	×	×	×	✓	2004
Mires	×	×	×	✓	×	2005
Sensation	×	×	×	×	×	2005
GlobalSensor Networks	✓	×	×	✓	×	2006
e-SENSE	✓	×	×	×	×	2006
SwissQM	×	×	×	×	✓	2007
TS-Mid	×	×	×	×	✓	2008
DAViM	×	×	×	×	✓	2008
CroCo	✓	×	×	×	×	2008
MUSIC	×	×	×	×	×	2009
HYDRA	✓	×	×	✓	×	2009
SPBCA	✓	×	×	✓	×	2009
Feel@Home	✓	×	×	×	×	2010
COPAL	✓	×	×	×	×	2010

UbiROAD	×	×	×	×	✓	✓	2010
KASOM	×	×	×	×	✓	✓	2011
UbiSOAP	×	×	×	×	×	×	2012
MOSDEN	×	×	×	×	×	×	2013
IoTSyS	✓	×	×	✓	×	×	2013
PRISMA	×	×	×	×	×	×	2014
EMMA	×	×	×	×	×	×	2015
RERUM	✓	×	✓	✓	✓	✓	2016
EMMA	×	×	×	×	×	✓	2018
Xively	✓	×	✓	×	×	×	2018
Kaa	✓	×	✓	✓	✓	×	2018
SiteWhere	✓	✓	×	×	×	×	2018
DeviceHive	✓	✓	×	✓	✓	×	2018
FIWARE	✓	✓	✓	✓	✓	✓	2015
SgxIoTGuard	×	×	×	✓	✓	✓	2018
Leshan	✓	✓	✓	✓	✓	✓	2019

7. Summary

In this paper we presented support for security features in various IoT middlewares, following discussing layers of IoT as well as representatives of IoT enabling standalone middlewares and frameworks and their objectives and properties. We divided security features into 4 functionalities:

- Access Control and authorization.
- Support for preventing network attacks.
- Device authorization and secure connection.
- Data filtering and integrity.

Additionally, we addressed additional features, which contain improvements not directly connected with provided security features but having positive influence on security measures. All information was collected into several tables to present which solutions cover which security aspects. The most common functionality is access control and authentication mechanism, the second one is ensuring data integrity. Few of the solutions discussed in the paper support device authorization. Limited support is provided against network attacks like DoS, but these kinds of attacks should be prevented within the deployment environment. Features which were not directly connected with security, but can be assumed as additional positive properties in security terms are, for example, extending battery lifetime in network devices or adapting data granularity. As we can see, many security features had been implemented but there are still open areas for future research. We can conclude that authentication and access control are quite well-established in new solutions. As these form an important part of current web applications, they can be successfully adjusted for IoT. Especially related to granting access for users or authenticating external services. Securing against attacks on services is mainly the responsibility of the deployment environment. Data filtering and consistency are well covered in terms of redundancy and error corrections. Using encrypted connections protects us against data leakage and data corruption but these mechanisms are as strong as the provided cryptography solutions. According to that we need to provide a secure way for device authorization. Well established protection from this side will increase the security level for the whole IoT network. Some research was performed [82] on cryptography for resource limited devices. We could make use of technologies connected with blockchain which may be useful at the device level and in M2M (Machine to Machine) communication. A state-of-the-art review in this area was performed in [83]. Another article which presents usage of the blockchain technology in data acquisition can be found in [84]. Taking a high-level view on this features we can point out a few important directions for future research:

- Trust management and identification of IoT devices.
- Cryptography algorithms for resource limited devices.
- External secure devices dedicated for cryptography operations.
- Providing a trusted supply chain for IoT devices.

Trust management and identification of IoT devices is the key feature to provide a reliable and secure IoT environment, but it is not possible until fast and secure cryptography is available in each device. We could say that trust management and identification of IoT devices is the main goal while the next three aforementioned points should provide a state-of-the-art tools for future development.

It is important to assign difficulties defined in Section 3 as future challenges in the IoT area. We can propose a solution which does not fully address all of them but can be the very first step against many attacks on IoT. Getting information from an audit log in cooperation with AI (artificial intelligence) algorithms may give promising results in investigating dangerous behaviors in IoT. Similar solutions are implemented in IoT platforms which are available in the PaaS (Platform as a Service) model [85].

Apart from the directions pointed out above more effort should be put into finding reliable solutions for investigation of security issues in the available IoT architectures. Such a solution was proposed in [86]. It is based on the SMT (Satisfiability Modulo Theories) framework which allows finding paths to attack an IoT system. Another solution was proposed in [87]. It extends the Hierarchical Attack Representation Model and is able to obtain a graphical security model representing system vulnerabilities, calculate cost of attack, probability of attack and possible improvements after deployment of chosen patches. A similar solution was presented in [88]. It is called IoTChecker and is an ontology based framework to verify and fix system weaknesses such as bugs in firmware, insufficient network security or default system configuration.

As the security in IoT has become a very important aspect in the research area we can expect more and more innovative security features in the near future.

References

[1] *INFSO D.4 Networked Enterprise & RFID, INFSO G.2 Micro & Nanosystems, RFID Working Group of the European Technology Platform on Smart Systems Integration (EPoSS). Internet of things in 2020, a roadmap for the future* 2008, *Technical report*, European Research Cluster on the Internet of Things

[2] Dr. Vermesan O, Dr. Friess P, Guillemin P, Gusmeroli S, Sundmaeker H, Dr. Bassi A, Jubert I S, Dr.Mazura M, Dr.Harrison M, Dr.Eisenhauer M and Dr.Doody P 2011 *Internet of things strategic research roadmap.*, IERC, European Research Cluster on the Internet of Things

[3] Gubbi J, Buyya R, Marusic S and Palaniswami M 2013 *Internet of things (iot): A vision, architectural elements, and future directions*, Future Generation Computer Systems, **29** (7) 1645

[4] Karpischek S, Michahelles F, Resatsch F and Fleisch E 2009 *Mobile sales assistant - an nfc-based product information system for retailers* 20

[5] Reilly D F, Welsman-Dinelle M, Bate C and Quinn K I 2005 *Just point and click?: using handhelds to interact with paper maps*, In Mobile HCI

[6] Niyato D, Hossain E and Camorlinga S 2009 *Remote patient monitoring service using heterogeneous wireless access networks: architecture and optimization*, IEEE Journal on Selected Areas in Communications, **27** (4) 412

[7] Yuksel A and Yüksel E 2012 *Rfid technology in business systems and supply chain management*, Journal of Economic and Social Studies, **1** 53
[8] Burrell J, Brooke T and Beckwith R 2004 *Vineyard computing: Sensor networks in agricultural production*, *Pervasive Computing*, IEEE, **3** 38
[9] Martínez-Sala A S, Egea-López E, Garcia-Sanchez F and García-Haro J 2009 *Tracking of returnable packaging and transport units with active rfid in the grocery supply chain*, Computers in Industry, **60** (3) 161
[10] Vashi S, Ram J, Modi J, Verma S and Prakash C 2017 *Internet of things (iot): A vision, architectural elements, and security issues*, *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)* 492
[11] Perera C, Zaslavsky A, Christen P and Georgakopoulos D 2013 *Context aware computing for the internet of things: A survey*, CoRR
[12] Tan L and Wang N 2010 *Future internet: The internet of things* **5** 376
[13] Khan R, Khan S, Zaheer R and Khan S. 2012, Future internet: The internet of things architecture, possible applications and key challenges 257
[14] Delicato C F, Pires P F and Zomaya A Y 2014 *Service-Oriented Middleware: Overview and Illustrative Example* 675 Springer Berlin Heidelberg, Berlin, Heidelberg
[15] Spiess P, Karnouskos S, Guinard D, Savio D, Baecker O, Sá de Souza L M and Trifa V *Soa-based integration of the internet of things in enterprise services*, *2009 IEEE International Conference on Web Services* 968
[16] Razzaque M A, Milojevic M, Palade A and Clarke S 2015 *Middleware for internet of things: a survey*, IEEE Internet of Things Journal, **3** 1
[17] Issarny V, Georgantas N, Hachem S, Zarras A, Vassiliadist P, Autili M, Gerosa M A and Hamida A B 2011 *Service-oriented middleware for the future internet: state of the art and research directions*, Journal of Internet Services and Applications, **2** (1) 23
[18] Czarnul P 2015 *Integration of Services into Workflow Applications*, Chapman and Hall/CRC
[19] Czarnul P 2013 *Modeling, run-time optimization and execution of distributed workflow applications in the jee-based beesycluster environment*, The Journal of Supercomputing, **63** (1) 46
[20] Czarnul P 2013 *A model, design, and implementation of an efficient multithreaded workflow execution engine with data streaming, caching, and storage constraints*, The Journal of Supercomputing, **63** (3) 919
[21] 2018 *Information security management systems - Overview and vocabulary*, Standard, International Organization for Standardization
[22] Russell D and Gangemi G T 1991 *Computer Security Basics*, O'Reilly & Associates, Inc. USA
[23] Zao J K, Wang M, Tsai P and Liu J W S 2010 *Smart phone based medicine in-take scheduler, reminder and monitor*, *In The 12th IEEE International Conference on e-Health Networking, Applications and Services* 162
[24] Czarnul P 2013 *Design of a distributed system using mobile devices and workflow management for measurement and control of a smart home and health*, *In 2013 6th International Conference on Human System Interactions (HSI)* 184
[25] *Baseline security recommendations for iot in the context of critical information infrastructures* 2017, *European Union Agency for Network and Information Security*, Technical report, ENISA - European Union Agency for Network and Information Security
[26] Rayome A D 2018 *As iot attacks increase 600% in one year, businesses need to up their security*
[27] Vigliarolo B 2018 *Bad user practices caused 41% of medical iot security issues in 2017*
[28] Krawczyk H 2019 *Service-oriented cyberspace for improving cybersecurity*, TASK Quarterly: scientific bulletin of Academic Computer Centre in Gdansk, **23** (2) 141

[29] Mayuri A and Sudhir T 2015 *Internet of things: Architecture, security issues and countermeasures*, International Journal of Computer Applications, **125** 1
[30] Farooq M U, Waseem M, Khairi A and Sadia Mazhar P S 2015 *A critical analysis on the security concerns of internet of things (iot)*, *International Journal of Computer Applications* **111** 1
[31] Xiaohui X 2013 *Study on security problems and key technologies of the internet of things* 407
[32] Sicari S, Rizzardi A, Grieco L and Coen-Porisini A 2015 *Security, privacy and trust in internet of things: The road ahead*, Computer Networks, **76** (01) 146
[33] Jung M 2013 *Iotsys - internet of things integration middleware*, https://code.google.com/archive/p/iotsys/
[34] OpenIoT Consortium *Openiot project*, http://www.openiot.eu
[35] Kang P, Borcea C, Xu G, Saxena A, Kremer U and Iftode L 2004 *Smart messages: A distributed computing platform for networks of embedded systems*, The Computer Journal, **47** (4) 475
[36] Heinzelman W B, Murphy A L, Carvalho H S and Perillo M A 2004 *Middleware to support sensor network applications*, Netwrk. Mag. of Global Internetwkg., **18** (1) 6
[37] Sacramento V, Endler M, Rubinsztejn H K, Lima L S, Goncalves K, Nascimento F N and Bueno G A 2004 *Moca: A middleware for developing collaborative applications for mobile users*, IEEE Distributed Systems Online, **5** (10) 2
[38] Alex H, Kumar M and Shirazi B 2004 *Midfusion: middleware for information fusion in sensor network applications*, *In Proceedings of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference* 617
[39] Souto E, Guimarães G, Vasconcelos G, Vieira M, Rosa N, Ferraz C and Kelner J 2006 *Mires: a publish/subscribe middleware for sensor networks*, Personal and Ubiquitous Computing, **10** (1) 37
[40] Hasiotis T, Alyfantis G, Tsetsos V, Sekkas O and Hadjiefthymiades S 2005 *Sensation: a middleware integration platform for pervasive applications in wireless sensor networks. In Proceeedings of the Second European Workshop on Wireless Sensor* 366 Networks
[41] Mueller R, Alonso G and Kossmann D 2007 *Swissqm: Next generation data processing in sensor networks*, In CIDR'07
[42] Lima R d C A, Rosa S N and Marques I R L 2008 *Ts-mid: Middleware for wireless sensor networks based on tuple space*, *In 22nd International Conference on Advanced Information Networking and Applications - Workshops (aina workshops 2008)* 886
[43] Badii A, Crouch M and Lallah CH 2010 *A context-awareness framework for intelligent networked embedded systems* 105
[44] Hoffmann M, Badii A, Engberg S, Nair R, Thiemert D, Matthess M and Schütte J 2019 *Towards semantic resolution of security in ambient environments*, Fraunhofer SIT
[45] L F, Sehic S and Dustdar S 2010 *Copal: An adaptive approach to context provisioning. 2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications* 286
[46] Eclipse *Eclipse leshan*, https://projects.eclipse.org/projects/iot.leshan
[47] Yao Y and Gehrke J 2002 *The cougar approach to in-network query processing in sensor networks*, ACM SIGMOD Record, **31** (09) 9
[48] *Cornell University Cougar project*, http://www.cs.cornell.edu/database/cougar/index.php
[49] Li S, Lin Y, Son S H, Stankovic J A and Wei Y 2004 *Event detection services using data service middleware in distributed sensor networks*, Telecommunication Systems, **26** (2) 351
[50] Gibbons P B, Karp B, Ke Y, Nath S and Seshan S 2003 *Irisnet: An architecture for a worldwide sensor web*, IEEE Pervasive Computing, **02** (4) 22

[51] Chen H, Finin W T and Joshi A 2003 *An intelligent broker for context-aware systems*
[52] Aberer K, Hauswirth M and Salehi A 2006 *A middleware for fast and flexible sensor network deployment*, *In Proceedings of the 32Nd International Conference on Very Large Data Bases. VLDB Endowment*, VLDB, **6** 1199
[53] Aberer K, Hauswirth M and Salehi A 2006 *Gsn global sensor networks* https://github.com/LSIR/gsn
[54] Gluhak A, Presser M, Shelby Z, Scotton P, Schott W and Chevillat P 2019 *e-sense reference model for sensor networks in b3g mobile communication systems*
[55] Lombriser C, Perianu M M, Perianu R M, Roggen D, Havinga P and Troster G 2007 *Organizing context information processing in dynamic wireless sensor networks*, *In 2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information* 67
[56] Horré W, Michiels S, Joosen W and Verbaeten P 2008 *Davim: Adaptable miaddleware for sensor networks*, IEEE Distributed Systems Online, **9** (1) 1
[57] Pietschmann S, Mitschick A, Winkler R and Meißner K 2008 *Croco: Ontology-based, cross-application context management*, *In 2008 Third International Workshop on Semantic Media Adaptation and Personalization* 88
[58] Rouvoy R, Barone P, Ding Y, Eliassen F, Hallsteinsen S, Lorenzo J, Mamelli A and Scholz U 2009 *Software engineering for self-adaptive systems*, *chapter MUSIC: Middleware Support for Self-Adaptation in Ubiquitous and Service-Oriented Environments* 164 Springer-Verlag, Berlin, Heidelberg
[59] Zhou X, Tang X, Yuan X and Chen D 2009 *Spbca: Semantic pattern-based context-aware middleware*, *In 2009 15th International Conference on Parallel and Distributed Systems* 891
[60] Guo B, Sun L and Zhang D 2010 *The architecture design of a cross-domain context management system* 499
[61] Terziyan V, Kaykova O and Zhovtobryukh D 2010 *Ubiroad: Semantic middleware for context-aware smart road environments*, *In Proceedings of the 2010 Fifth International Conference on Internet and Web Applications and Services, ICIW* **10** 295 Washington, DC, USA
[62] Corredor I, Martínez M J, Familiar S M and López L 2012 *Knowledge-aware and service-oriented middleware for deploying pervasive services*, J. Netw. Comput. Appl., **35** (2) 562
[63] Raverdy P, Issarny V and Caporuscio M 2010 *ubisoap: A service-oriented middleware for ubiquitous networking*, IEEE Transactions on Services Computing, **5** 86
[64] Perera C, Jayaraman P P, Zaslavsky A, Georgakopoulos D and Christen P 2014 *Mosden: An internet of things middleware for resource constrained mobile devices*, *In 2014 47th Hawaii International Conference on System Sciences(HICSS)* **00** 1053
[65] Jung M 2014 *An integration middleware for the internet of things*
[66] Delicato F, Silva J R, Pirmez L, Pires P, Portocarrero J, Batista T and Rodrigues T 2014 *Prisma: Publish/subscribe and resource oriented middleware for wireless sensor networks* **2014**
[67] Duhart C, Sauvage P and Bertelle C 2015 *EMMA:A resource oriented framework for service choreography over wireless sensor and actor networks*, CoRR
[68] Moldovan G, Tragos E Z, Fragkiadakis A, Pohls H C and Calvo D 2016 *An iot middleware for enhanced security and privacy: The rerum approach*, *In 2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)* 1
[69] Rausch T, Nastic S and Dustdar S 2018 *Emma: Distributed qos-aware mqtt middleware for edge computing applications*, *In 2018 IEEE International Conference on Cloud Engineering (IC2E)* 191
[70] *Google. Xively*, https://xively.com/
[71] *KaaIoT Technologies. Kaa*, https://www.kaaproject.org/overview/

[72] *SiteWhere LLC Sitewhere*, http://www.sitewhere.org/
[73] *DataArt Solutions, Devicehive*, https://devicehive.com/
[74] *DataArt Solutions, Devicehive - usage*, https://devicehive.com/industries/
[75] *FIWARE FOUNDATION Fiware*, https://www.fiware.org/
[76] Celesti A, Fazio M, Márquez G F, Glikson A, Mauwa H, Bagula A, Celesti F and Villari M 2019 *How to develop iot cloud e-health systems based on fiware: A lesson learnt*, Journal of Sensor and Actuator Networks, **8** (1) 1 https://www.mdpi.com/2224-2708/8/1/7
[77] López-Riquelme J A, Pavón-Pulido N, Navarro-Hellín H, Soto-Valles F and Torres-Sánchez R 2017 *A software architecture based on FIWARE cloud for Precision Agriculture*, Agricultural Water Management, **183** (C) 123
[78] Fernández P, Santana J M, Ortega s, Trujillo A, Suárez J P, Domínguez C, Santana J and Sánchez A 2016 *Smartport: A platform for sensor data monitoring in a seaport based on fiware*, Sensors, **16** (3) 1 https://www.mdpi.com/1424-8220/16/3/417
[79] Carnevale L, Galletta A, Fazio M, Celesti A and Villari M 2018 *Designing a fiware cloud solution for making your travel smoother: The fliware experience*, *In 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)* 392
[80] Ayoade G, El-Ghamry A, Karande V, Khan L, Alrahmawy M and Rashad Z M 2019 *Secure data processing for iot middleware systems*, The Journal of Supercomputing, **75** (8) 4684
[81] Fremantle P and Scott P 2017 *A survey of secure middleware for the internet of things*, PeerJ Computer Science, **3** (e114)
[82] Vulpe A, Arseni S C, Marcu I, Voicu C and Fratu O 2017 *Building a unified middleware architecture for security in iot.*, *In Álvaro Rocha, Ana Maria Correia, Hojjat Adeli, Luís Paulo Reis, and Sandra Costanzo, editors*, Recent Advances in Information Systems and Technologies 105 Cham
[83] Panarello A, Tapas N, Merlino G, Longo F and Puliafito A 2018 *Blockchain and iot integration: A systematic survey*, Sensors, **18** 2575
[84] Anus A and Buchwald P 2019 *Safe locker - a secure cargo transport control support it system*, TASK Quarterly: scientific bulletin of Academic Computer Centre in Gdansk, **23** (1) 19
[85] López D, Uribe M, Santiago C, Murgueitio D, Garcia E, Nespoli P and Marmol F G 2018 *Developing secure iot services: A security-oriented review of iot platforms*, Symmetry, **10** 669
[86] Mohsin M, Anwar Z, Husari G, Al-Shaer E and Rahman M 2016 *Iotsat: A formal framework for security analysis of the internet of things (iot)* 180
[87] Ge M, Hong J B, Guttmann W and Kim D S 2017 *A framework for automating security analysis of the internet of things*, J. Netw. Comput. Appl., **83** (C) 12
[88] Mohsin M, Anwar Z, Zaman F and Al-Shaer E 2017 *Iotchecker: A data-driven framework for security analytics of internet of things configurations*, Computers & Security, **70** 199