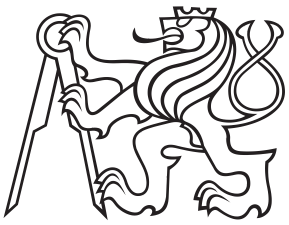
Dissertation Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

Context-aware security of Internet of Things applications

Michal Trnka

Supervisor: doc. Ing. Karel Richta, CSc.
Supervisor–specialist: Ing. Tomáš Černý, MSc., Ph.D.
Field of study: Electrical Engineering and Information Technology
Subfield: Information Science and Computer Engineering
June 2022

Acknowledgements

This thesis would not have been possible without the help of many people and institutions. Two individuals provided me crucial support during my research. My supervisor, doc. Richta, supported me and led me through the challenging process of doctoral studies. Co-supervisor dr. Černý was always available for advice, feedback, or consultation.

My sincere thanks go to my family. My parents have supported me throughout my education process. I would also like to mention my wife, who had enough patience and understanding to move with me to the USA for nine months with a newborn baby. Finally, my thoughts go to my two daughters, who have often had to spend evenings and even weekends without their beloved father.

I want to express my sincere gratitude to the Fulbright Commission for sponsoring my research visit and to Baylor University for allowing me to do a crucial part of my research in their facilities in Waco, TX.

Declaration

I declare that this thesis has been composed solely by myself. I have used ideas and content of my own properly published articles. Except where states otherwise by reference or acknowledgment, the work presented is entirely my own.

Prague, June 3, 2022

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou literaturu na které můj výzkum staví. V práci jsem vycházel z myšlenek a obsahu svých řádně publikovaných prací.

V Praze, 3. června 2022

Abstract

Security is a major research topic within the Internet of Things landscape. In recent years, the development of Internet of Things solutions has moved forward significantly. However, security has lagged behind all the notable progress that has been accomplished.

In my research, I focus on authentication, authorization, and some aspects of identity management of Internet of Things network participants. I consider the challenges from the software engineering perspective – the architecture and the high-level design of authentication and authorization solutions. This implies that I operate on the application layer of the networking stack. Specifically, I concentrate on three main areas - context retrieval, context-aware authorization, and identity and security rules sharing within the scope of the Internet of Things.

I begin by evaluating the current state of the art in order to show where my research stands in relation to the work of other researchers.

Initially, I propose a method for determining context from the network neighborhood. The method evaluates available devices on the network and tracks their temporal changes. The changes in the composition of the devices on the network are quantified and are used as additional contextual information.

The core part of the research is focused on authorization. I describe a context-aware extension of Role-Based Access Control using security levels. A level is a linear single value representation of the context. The user's level is determined during logging into the application via various configurable context resolvers.

The last part of the thesis covers identity management in the Internet of Things. I utilize a centralized element to store the identities of devices and users, and to provide authentication in the form of a token. In addition, the central server provides additional attributes, such as the roles in the token.

Keywords: Internet of things, dissertation, software security, software engineering, authentication, authorization

Supervisor: doc. Ing. Karel Richta, CSc.
Department of Computer Science
Resslova 9, E-430
Prague

Abstrakt

Zabezpečení je jedna z klíčových oblastí výzkumu v oblasti Internetu věcí. V nedávné době se vývoj Internetu věcí významně posunul kupředu, nicméně zabezpečení stále zůstává pozadu za pokrokem, který byl dosažen ve zbylých oblastech.

Ve své práci se zaměřuji především na autentizaci, autorizaci a částečně na správu identit účastníků komunikace Internetu věcí. A na tuto problematiku nahlížím z perspektivy softwarového inženýrství, tj. zajímám se o architekturu a obecnou strukturu řešení. Znamená to tedy, že operuji na aplikační vrstvě síťového modelu. V popředí mého zájmu stojí tři oblasti, kterými jsou získání kontextu, autorizace s ohledem na kontext, a sdílení identit a zabezpečovacích pravidel v rámci Internetu věcí. V této práci shrnuji rešerši současného stavu poznání a popisuji kam patří mé bádání v rámci stávajícího širšího výzkumu.

Zaměřil jsem se nejprve na metodu zjišťování kontextu ze síťového okolí, která rozpoznává dostupná zařízení v síti, a vyhodnocuje jejich vývoj v průběhu času. Změny ve složení těchto zařízení jsou poté kvantifikovány a požity jako další kontextová informace.

Nejdůležitější část mého výzkumu zaujímá autorizace nebo-li ověření přístupových oprávnění. V této části rozšiřuji zabezpečení pomocí rolí o kontextový element v podobě úrovně zabezpečení. Tzn., že daná úroveň je lineární hodnota reprezentující stávající kontext, a úroveň zabezpečení uživatele je vyhodnocena během jeho přihlášení do aplikace pomocí různých nastavitelných rozhodovacích mechanismů, které vyhodnocují specifické aspekty kontextu.

V poslední části mé práce se zabývám správou identity na Internetu věcí. K tomu využívám centrální prvek pro ukládání identity zařízení a uživatelů. Tento prvek vydává token, který je používán k přihlašování do síťového prostředí. Může však obsahovat i další atributy pro autorizaci.

Klíčová slova: Internet věcí, dizertace, aplikační zabezpečení, softwarové inženýrství, autentikace, autorizace

Překlad názvu: Zabezpeční systémů pro Internet věcí s ohledem na kontext

Contents

Chapter 1

Introduction

The Internet of Things (IoT) is an environment in which numerous heterogeneous devices, possibly including small devices, interact and cooperate. Each device may have a specialized function where the overall ecosystem provides various features that may be more complex. IoT solutions are currently deployed in diverse domains that range from agriculture through transportation, retail, physical security, industrial automation, home solutions and healthcare, all the way up to defense systems and space exploration.

As ubiquitous networks of mutually-connected devices surround us, it is crucial to understand their security and privacy. The IoT has extensive access to data and a remarkable ability to influence our lives. A security issue can have a severe impact on privacy or can be very costly, and it can also affect human health or even be life-threatening. The large number of cooperating devices makes security all the more complicated. It raises numerous problems: Which participants can we share data with? Which participants may we generally interact with? How to authenticate participants, how to detect a malicious participant, how to introduce a new device into the network, how and when to retire the device, and much more. The situation is further complicated by the heterogeneity of the environment. The devices in a network have different software versions, operating systems, manufacturers, and often also different owners. Typically, not all users and stakeholders in the network show significant concern for security [1]. This implies that security needs to be enforced by the system, and must not be left for users to decide. In addition, security is often ignored during the early adoption phase, in order to go to market as soon as possible [2]. It is therefore hardly surprising that security is considered as one of the most crucial challenges [3], [4] of the IoT ecosystem.

Generally, the security challenges for IoT are similar to the security challenges

for traditional applications. However, conventional security architectures were not designed to fully include communication of machines with each other, typically with limited computational resources, and the security architectures were not set up with the heterogeneous and distributed environment of IoT in mind. Standard solutions therefore tend to struggle or even fail, and IoT security solutions need to better reflect the specific requirements of the IoT environment. Another notable contrast between IoT and traditional applications is the altering nature and the fluctuating environment of IoT. Devices connect and disconnect dynamically from the network, and devices and applications are deployed in an environment that is not completely under the control of the operator.

One of the properties of the IoT environment is its broad access to context [5]. The context provides an explanation for the data provided by any participant and allows us to understand the participant's situation better. Moreover, the context can be leveraged to enhance existing security methods with a context that enables additional security and that improves or enables personalization. Attempts to leverage context information to enhance "traditional application security" have been around for more than 15 years and are backed up by solid research in this domain [6]–[8]. My work is therefore able to utilize existing knowledge, extend it and transfer it to the IoT environment.

This dissertation focuses on authentication, authorization, and some aspects of identity management of IoT devices and users. From the perspective of the standard ISO OSI model, it provides an answer to issues on the highest application layer. My solution in this dissertation aims to provide an easy-to-use context-aware authentication method (or methods) for IoT solutions and a context-aware authorization architecture tailored to the IoT domain, mitigating current challenges and capitalizing on the many advantages and strong points of IoT.

The specific goals of this thesis are:

1. **Develop a method for determining context in the IoT environment.** Leverage the extended access to the context and take into consideration specific properties of IoT devices. The proposed method must be simple to adopt and must be optimized for constrained devices.

2. **Develop a context-aware security architecture usable for IoT applications.** Explore existing security architectures and, based on their strengths and weaknesses, propose either a new architecture or the evolution of an existing security architecture. The proposed solution must be scalable and easy to adopt.

3. **Enable security rules to be shared across participants in the IoT environment.** Utilize existing tools and protocols and enable quick rule update propagation. Create a mechanism with a single focal point for security administration of the IoT deployment.

The goals mentioned above form a detailed security design that is constructed specifically for the IoT environment. It provides a complete solution from context retrieval, through security architecture, to security rules synchronization across the network. It allows using only selected parts of the solutions that fit particular needs and replace the other parts with some alternative options. The approach bases on current, existing, and proven solutions. Special emphasis was placed on easy adoption by the system designers, architects, maintainers, and developers. The main advantage of the proposed design is that it will leverage the natural advantage of the IoT environment – access to the context.

Significance: Accomplishing the goals stated above enables security concerns in IoT solutions to be addressed. It will reduce the work efforts of developers, architects, quality engineers, and system maintainers. Currently, either traditional security architectures and approaches are used or a custom solution is developed (or, in the worst case, security is completely ignored). The results presented in this dissertation provide a complete solution tailored specifically for the IoT environment. The solution aims to leverage the advantages of IoT and to mitigate its security issues.

Scientific merit: The dissertation describes a novel and unique method for context retrieval for IoT devices. It is developed with constrained devices in mind, and is tailored to the computational constraints that they have. The data can be stored on a master device that is controlling the end devices. Further, the dissertation defines an extension of traditional security architectures with context-aware elements. The extension is specific, with simple implementation as one of the main characteristics. It can therefore be adopted in distinct parts of the system, allowing an architect or a developer to decide what is wanted, and the extension can theoretically be applied with various traditional architectures. Finally, the dissertation proposes a method for sharing security rules across the devices in the IoT network.

Broader impact: The results presented in the dissertation will contribute to faster adoption of various IoT solutions (often called smart solutions) by allowing developers to concentrate on relevant business objectives, rather than spending time developing a security architecture. The findings will also help to reduce the

number of security incidents. A further benefit of this work is that it will allow developers from other domains of Information Technology (IT) to migrate easily to IoT development. In this way, the findings will contribute to the further spread of IoT solutions.

Organization of the dissertation: The last part of this section is devoted to a general technology overview. Related work is detailed in Chapter 2. Chapter 3 describes the context retrieval method. Context-aware authorization research is presented in Chapter 4. The rule sharing method is elaborated in Chapter 5. Conclusions, a summary of the contribution of the work, and future work opportunities are presented in Chapter 6.

1.1 Overview of the current state-of-the-art

This section provides an overview of the relevant technologies and principles for the dissertation. The aim of this brief introduction is to present the state of current knowledge and to put my research into its context. I give a brief overview of the Internet of Things and of context-awareness. Then I go through selected prominent authorization architectures and principles related to authorization, present selected context-aware security architectures, and discuss the relationship between the Internet of Things and context-awareness.

1.1.1 Internet of Things

The origins of the Internet [9] can be traced back to the 1980s, when computers first began to be connected together on a major scale. Ever since that time, more and more new types of devices have been plugged into networks. The earliest devices to be extensively connected with computers were printers and data projectors, but since the first decade of the 21st century the connection of other devices has ramped up [10]. Present-day networks include an enormous number of types of "smart objects" [11]. The environment where these devices cooperate together to reach common goals is called the IoT [3].

The number of IoT devices is expected to continue to grow. It is impossible to calculate the exact number of connected devices, but various industry reports have shown an increasing trend and predict continuing growth. The numbers of IoT devices that are reported vary (according to the way the devices are defined), but the trend is clear. Reports from tech companies illustrate the rate of growth. Cisco expects growth from 3.9 billion devices in 2018 to 5.3 billion in 2023 [12]. Intel

estimated the number of IoT devices in 2020 to be 200 billion [13], and a recent business report predicts growth of the IoT market from USD 139.3 billion in 2019 to USD 278.9 billion by 2024, an average yearly growth of 14.9% [14].

IoT solution deployments are becoming increasingly popular. They span across various domains and vary in size, and their production readiness varies from academic or experimental systems through local adoption to large companies or a countrywide solution. IoT applications that are attracting tremendous attention include:

1. Smart power grids [15]–[19]. This enables energy delivery, consumption and asset optimization of the grid. It enables the demand for electricity to be matched with electricity supply, and it therefore prevents a blackout if the demand exceeds the supply. Smart power grids reduce waste, costs, and pollution due to oversupply.

2. Smart healthcare [20]–[24], which focuses on easing overloaded healthcare systems, and therefore saves time, costs, and lives. The predominant approach is home monitoring of patients using smart devices (e.g., wearables). Smart healthcare allows patients to visit the hospital or get specialized treatment at the right time. Alternatively, it can be utilized for rehabilitation, where smart devices can adjust the plan according to the personalized needs and the progress of the patient.

3. The smart city [25]–[28] includes IoT applications such as smart mobility [29], [30], smart city governance [31]–[33], smart homes [34], [35] and a smart power grid. In a smart city, the technology can adapt to the flow of city life or can even optimize the flow.

1.1.2 Selected security principles and their evolution

Initially, computers were used as advanced machines to process various calculations and other processes, without storing input or output data. The systems supported multiple users. However, no data were stored, so security issues were not prevalent. However, when computers began to be used for data management and storage with multiple users accessing the system, the problem of access control emerged.

From the 1970s on, two predominant access control models were used — Mandatory Access Control (MAC) and Discretionary Access Control (DAC)) [36]. MAC is predominantly used in applications with strict, centralized access control. Access rules are set by the administrators and are enforced by the system; users are not

allowed to set or modify access policies for system resources. DAC is the opposite; no central element is needed, and each user determines the access policy for the resources that he owns.

As the complexity of applications increased and evolved into complex information systems with hundreds or thousands of users, a conceptual framework for easier access management was needed. Role-based Access Control Role-based Access Control (RBAC) [37] allows users to be grouped together into groups, known as roles; each user may be assigned multiple roles. Access rules are further defined for roles, and not for individual users. The roles often follow the organizational structure of the institution using the information system. They are therefore easy for business owners of the application to understand. RBAC was introduced in the early 1990s and quickly became the predominant access control model.

The access control methods described above deal predominantly with authorizing users to access specific resources or to take specific actions, rather than describing how the user should be authenticated; authentication is considered a prerequisite for authorization. Authentication may be accomplished using three basic credential categories. The first category, "Something I am", represents properties about the user, including the user's location or biometric characteristics. "Something I have" involves credentials that have been given to a user; the user possesses the credential. This category includes all types of keys, tokens, cards, or even personal devices like phones. The last and most familiar category is "Something I know", most often represented by passwords, but not limited to them – it also includes the user's knowledge of security questions, their interaction history, and other information.

Identity management is closely related to authentication and authorization. The virtual identity of a user (or of a device) stores attributes such as references to credentials, RBAC roles, and other required information. During authentication, a user claims ownership of this virtual identity, and its attributes can be further used for authorization.

At the most basic level, each application manages identity independently, using very little information -– generally, identity includes both a principal (an identity-unique identifier) and credentials used for authentication. As applications became more complex, the information required for user authorization grew to include roles or identity attributes. As the number of applications per user and the number of users per service increase, it becomes difficult both for the user and for service administrators to manage the growing amount of identity information that is required. These developments led to the need for federated identity management — a way of providing identity services for multiple applications, often tied to authentication

mechanisms. Currently, several implementations of federated identity management exist, including the use of LDAP [38] for identity management and the use of OpenID [39] as an identity service.

1.1.3 Context-awareness

A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task. This is the definition of context-awareness by Dey [40]. To understand it, we need to explain what context actually is. The definition of context offered by Abowd [5] is the most prevalent and the most cited. He defines context as: *context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the applications themselves.*

Abowd published his definition of context in 1999. This is the time when the first context-aware applications were developed. Chen and Kotz, in the article "A Survey of Context-Aware Mobile Computing Research" [41] from 2000, presented more than ten context-aware applications and summarized multiple methods for sensing the context, and for modeling the context and architecture proposals. Harter et al. in 2001 proposed in their architecture to use data from external sensors [42]. A massive increase in the usage of context-awareness occurred when Web 2.0 [43] became popular and mobile applications started to gain attention [44]. With Web 2.0, users started to participate in the content of websites, and this led to the possibility to retrieve more information about the users. The use of mobile applications leads to the ability to extract extra contextual information, e.g. exact position, and also to the demand of users for personalized applications.

We can categorize the usage of context into three categories [40]:

1. Presentation of information and services to a user — the system uses context to provide more accurate information for the user. An example of this is a personalized search on a search engine.

2. Contextual tagging of information to support later retrieval – the system automatically adds to the data some contextual information. For example, the location of the user and the time when the user created the data.

3. Automatic execution of a service for a user — the system determines when it is appropriate to launch some service based on the context. This may be recalculating a route when the user leaves the recommended path, switching

on a light when the user enters a room, or, in the computer security domain, launching additional authentication based on the user's location history.

One of the main challenges for context-aware applications is context retrieval [45]. Some information is automatically available for the system (e.g., time, frequency of log-ins, the history of user interaction with the application), while other information can be guessed but not guaranteed (e.g., geographical location determined from the IP address). However, much information is difficult to obtain (e.g., biometric information about the user). All the information about the user's context may significantly increase the security of the system, and at the same time may improve the user's experience of the application.

The idea of context-aware systems having access to information from sensors is not novel; in fact, it is more than a decade old [45]. For the IoT solutions, solutions make little sense without the use of contextual information. Contextual information is used to present the information to the user, to execute services, to control IoT devices, or to tag the data for later auditing, statistical or other use. In IoT solutions, there is no standardized or "best" approach for most of the main challenges: how to model context, what principle/architecture offers the best access to it, what is the best reasoning model. The acquisition of context can range from direct access to sensors, through the use of various middleware solutions for aggregating contextual information from a specific part, to a big contextual data lake [46]. Reasoning models [47], [48] can be rules-based, built on top of supervised or unsupervised learning, ontology-driven or probabilistic reasoning. A survey article that offers much insight is [49]. This article specifically describes IoT context-aware computing, and provides an in-depth overview of the topic. It covers network architectures, open challenges, context types and categorization, levels of context-awareness for various systems, context management principles, context acquisition techniques and their lifecycle, context models, and already existing contextual systems.

1.1.4 Context-aware security architectures

Security architectures can benefit from context-awareness. Context-aware elements bring benefits for both the user and the owners/maintainers of the application. The application user is presented with a better user experience, and the application owners achieve higher security of their system if context information is considered for security.

Usually, users are assigned various roles or permissions for resources in ap-

plications, and security rules are independent of context. We can expect that users and application owners would benefit from application security based on context to provide specific access to resources based on context. Applications using context-aware security can be much less obtrusive for users. Users can be asked for different authentication methods based on context; they can be authorized for the same resource in various ways, depending on their context. For example, access from City A can have different access rights than access from City B. Users can even sometimes omit authentication because their context is trustworthy by itself (e.g., access from the company workplace). Similar to users, application operators, too, can profit from context-based authentication and authorization. They might define stricter security rules for suspicious user behavior (e.g., Internet access to confidential resources at night). Using context allows system administrators to impose more fine-grained security rules, which would otherwise be tangled through multiple business domains and would make the security rules unsustainable for maintenance.

When adopting context-aware security architectures, two basic approaches are possible. Either extend and adapt some existing security architecture for context-awareness, or develop an entirely new architecture. Solutions from both categories have been explored and described in the literature, though the adaptation of some proven architecture is more common.

Extending RBAC requires more effort and is not straightforward. Various paths have therefore been developed for achieving context-aware RBAC. One approach is to add another set of roles to RBAC. Moyer et al. [50] propose creating two additional sets of object roles and environmental roles and tying permissions to a trio of roles. Further research has simplified the requirement to just one additional set of environmental roles [51]. These rules are hierarchically composed and represent the current state of the system. Similarly to this approach, it is possible to have an additional set of context roles [6]. A slightly different method is to introduce the concept of trust, and to extend simple RBAC with this concept [7].

A different method is to grant roles to the user during authentication based on her context [52]. In this way, the user can obtain new roles that reflect her context. The idea has been further developed into Context-Aware RBAC [53]. There is an additional layer of authorization architecture, which is responsible for granting and revoking roles when the context changes. The roles therefore dynamically reflect the context.

It is also possible to extend RBAC by adding another element not based on roles. An example is adding context constraints to security policies [54]. When the

permission is checked, the user needs not only to possess permission for the resource (based on her role) but also to fulfill context constraints. Similar approaches involve introducing other system participants into the system. The approach either determines the access rights on the basis of four elements: permission, role, context, and authentication method [55] or, alternatively, it can use four different context actors: the context owner, the context provider, context broken, and context-aware service [8].

Most security architectures are strictly based on a pre-existing solution. One method that might work with every security architecture is to add another context dimension to the current security rules [56]. Another remarkable idea is to assign permissions directly to contexts [57].

Chapter 2

Literature review

This literature review provides an overview of the progress of research in the domain of IoT security. This is a broad discipline, and I therefore focus mainly on papers dealing with authorization, authentication, and identity management, specifically at the highest layer of the network stack, typically the application layer. A "network stack" is not the precise model used for the IoT. However, I use the term because no more standard vocabulary is available to describe the IoT technology and communication architecture; there does not yet appear to be a widely-agreed term. I am interested in architectures, projects, solutions, proposals, identity-management of IoT devices and frameworks dealing with user-to-machine and machine-to-machine authentication and authorization, as these topics largely overlap with my dissertation research. The papers selected for presentation in this literature review have been identified by a systematic search [58] through major indexing sites and portals. The selected papers are analyzed to provide a comprehensive overview and classification of existing work.

This Chapter aims to achieve the following goals:

- Categorize and provide a taxonomy of security solutions.
- Identify context-aware security solutions and go through their methods.
- Examine whether IoT solutions are already existing solutions adapted for the IoT environment, or whether novel methods are proposed.
- Explore the architecture of the security solution in terms of whether the security solution is centralized or distributed.
- Enumerate the existing solutions to find out whether they are focused on User to Machine (U2M) interactions or on Machine to Machine (M2M) interactions.

I carried out a literature overview in late 2017, and it led to an article [A.3] that was published in early 2018. The article had a considerable impact on the scientific community, as it obtained 51 citations, 15 of them from journals indexed in Web of Science (WoS) Science Citation Index Expanded (SCIE). This chapter is based on the article, but it has been greatly extended by research conducted and published before 2021. The extension is part of [A.2].

For a reader who wants to get more familiar with the whole broad topic of IoT security, or who wants to read additional materials providing an overview of the research problem, I introduce here some authoritative surveys and systematic study papers from recent years. Noor et al. published a broad IoT security survey [59]. I consider this study to be excellent, although it is limited to years 2016 – 2018. The most recent overview is provided in [60], which was published in July 2020. Milovlaskaya et al. summarized information security research in [61]. A survey of continuous authentication methods [62] provides a comprehensive overview of this specialized issue. Another focused study [63] goes through industrial IoT security issues.

2.1 Search

This chapter primarily uses data from my survey article [A.3], which contains data from 2017 and earlier. However, during the time between submitting the survey article and writing this thesis, there have been a significant number of newly-published papers, and I have therefore updated the initial set of papers with the newest scientific results.

2.1.1 Initial Search

In order to make a systematic review of all existing research, and to provide answers to our research questions, I performed searches at the following indexing sites and portals: IEEE Xplore, ACM Digital Library (ACM DL), WoS (Core), SpringerLink, and ScienceDirect.

To show that my search queries provide results relevant for this dissertation, I evaluated the search query results against a control set of papers identified as matching the scope through a manual search before I performed the search queries. When a search query returned papers from the control set, this provided evidence of the usefulness of the search query.

The search query consists of two parts. The first part targets terms and keywords to be included in the paper, and the second part removes papers that contain

Indexer	Query
General query	("Internet of Things" OR "IoT") AND "Security" AND ("Authentication" OR "Authorization" OR "Identity" OR "Access control") AND NOT ("Network" OR "Hardware" OR "RFID" OR "Protocol" OR "Cryptography" OR "Survey" OR "Study")
IEEE Xplore	("Abstract": "Internet of Things" OR "Abstract": "IoT") AND ("Abstract": "Authentication" OR "Abstract": "Authorization" OR "Abstract": "Identity" OR "Abstract": "Access Control") AND "Index Terms": "Security" AND NOT("Index Terms": "Network" OR "Abstract": "Hardware" OR "Abstract": "Cryptography" OR "Abstract": "Protocol" OR "Document Title": "Survey" OR "Abstract": "RFID" OR "Document Title": "Study")
ACM DL	Abstract:(IoT "Internet of Things") AND Abstract:(Authentication Authorization Identity "Access Control") AND Title:(-study -Survey) AND Abstract:(-Hardware -rfid -Cryptography) AND Keyword:(-Hardware -Physical -Network)
WoS	TI=(Internet of Things OR IoT) AND TS=(Authentication OR Authorization OR Identity OR Access Control) NOT TS=(Hardware OR Cryptography OR Protocol OR RFID OR Physical OR Network) NOT TS=(Survey OR Study) AND TS=Security
SpringerLink	(Authentication OR Authorization OR Identity OR "Access Control") + title ("Internet of Things" OR IoT)
ScienceDirect	TITLE-ABSTR-KEY("Internet of Things" OR "IoT") AND TITLE-ABSTR-KEY(Authentication OR Authorization OR Identity OR "Access Control") AND KEY(Security) AND NOT (TITLE-ABSTR-KEY(Hardware OR Cryptography OR Protocol OR RFID) OR title(study OR survey) OR key(Physical OR Network))

Table 2.1: Queries used for the search

terms we are not interested in. Naturally, I am interested in research about the IoT, so I include "Internet of Things" or "IoT" as one of the main groups. Another important term is "Security", and I have targeted only those papers that deal with security. Further restriction terms refine the results to include only papers with "Authentication", "Authorization", "Access Control" or identity management, which is shortened to "Identity". The second portion of the query is to limit the number of articles in the result set. I removed papers that deal with the lower levels of the network stack. This translates to the terms "Network", "Hardware", "RFID", and "Protocol". Cryptography is not a particular focus of this survey, so I have also removed research with this keyword. Finally, I have removed papers that are surveys themselves, containing "Survey" or "Study" in their title.

The query syntax differs for each indexing site, but I aimed to search through abstracts or keywords/topics, where applicable. The queries are constructed as

Indexer	Results	Prefiltered	Relevant
IEEE Xplore	120	29	14
ACM DL	84	9	7
WoS	67	31	13
SpringerLink	33	8	6
ScienceDirect	27	9	2
Total	331	86	42

Table 2.2: Number of articles processed in the survey

similarly as possible. The exact queries used, including the general query that I used as a template, are listed in Table 2.1.

I encountered an issue with the search function in SpringerLink. The search system is not able to process an advanced query, such as the query that I designed. I used a more straightforward query that returned 383 papers. I then processed the results by constructing a short script that opens the particular page for each exported paper, extracts the abstract, and performs the advanced query locally on our machine.

Running the query across all five indexing services gives us a set of 387 papers, from which I exclude those less than four pages in length. Since WoS indexes papers that appear at other sites, it contains 16 duplicate papers, which I also removed. As a final filter, I read the abstract of each article and removed those papers not within the designed scope; this gave me 86 prefiltered candidate papers. I also excluded my own article [A.6], as it is discussed in a separate chapter.

I read the remaining papers one by one, with some exceptions. The full text of one paper could not be downloaded; this was removed from the results set. Three of the papers were highly similar extensions of another paper in the results set. In this case, I used the extended paper and discarded the shorter versions. I also removed papers that did not fit into the scope of the literature review – those where the abstract had initially indicated a connection to our research questions, but the full text did not show a connection. The complete statistics of the papers that were found, prefiltered, and included for every indexing site are shown in Table 2.2.

2.1.2 Update search

The initial idea was to update the research with the same approach, just for years 2018 – 2020. However, this turned out to be unrealistic; the amount of research in the area of IoT security has multiplied. There are currently five times more research publications on the topic than three years ago. Table 2.3 illustrates the growth of

Indexer	2017	2020	Growth
IEEE Xplore	120	507	387
ACM DL	84	511	427
WoS	67	349	282
WoS SCIE	21	155	134
ScienceDirect	27	171*	144*
Total	298	1537	1241

Table 2.3: Growth in the number of publications

Primary source	count
IEEE Xplore	7
ACM DL	0
WoS	8
Springer	4
ScienceDirect	2

Table 2.4: Primary sources of publications

the research. I intentionally skipped SpringerLink, as it requires post-processing on the computer. ScienceDirect has changed the search to allow a maximum of 8 Boolean operators, and for this reason its results contain a larger set of articles and * marks the numbers. Finally, I included both the WoS Core collection and, separately, the SCIE index. In the Total row, I use only the larger WoS Core collection, as it is a superset of SCIE.

I decided to go through only the WoS SCIE articles to extend the initial set. The reason is that the vast majority of useful articles are indexed in WoS SCIE. In addition, these articles typically have the highest impact on the scientific community (measured by citations).

The statistics are as follows - out of 155 articles, I filtered out 67 based on abstracts that I read. I found that 21 articles were related to the topic of the dissertation. This is a significant growth, as the initial search contained only 11 articles that were indexed in WoS SCIE. Table 2.4 shows the distribution of the primary sources and suggests that 13 of the articles that were found would duplicate other indexing services. In addition, a further 18 papers were variations and adaptations of Ciphertext-Policy Attribute-Based Encryption (CP-ABE) for IoT. Although this is not precisely the topic of the dissertation, CP-ABE can be understood as a means of authorization. It is therefore quite closely related to our topic.

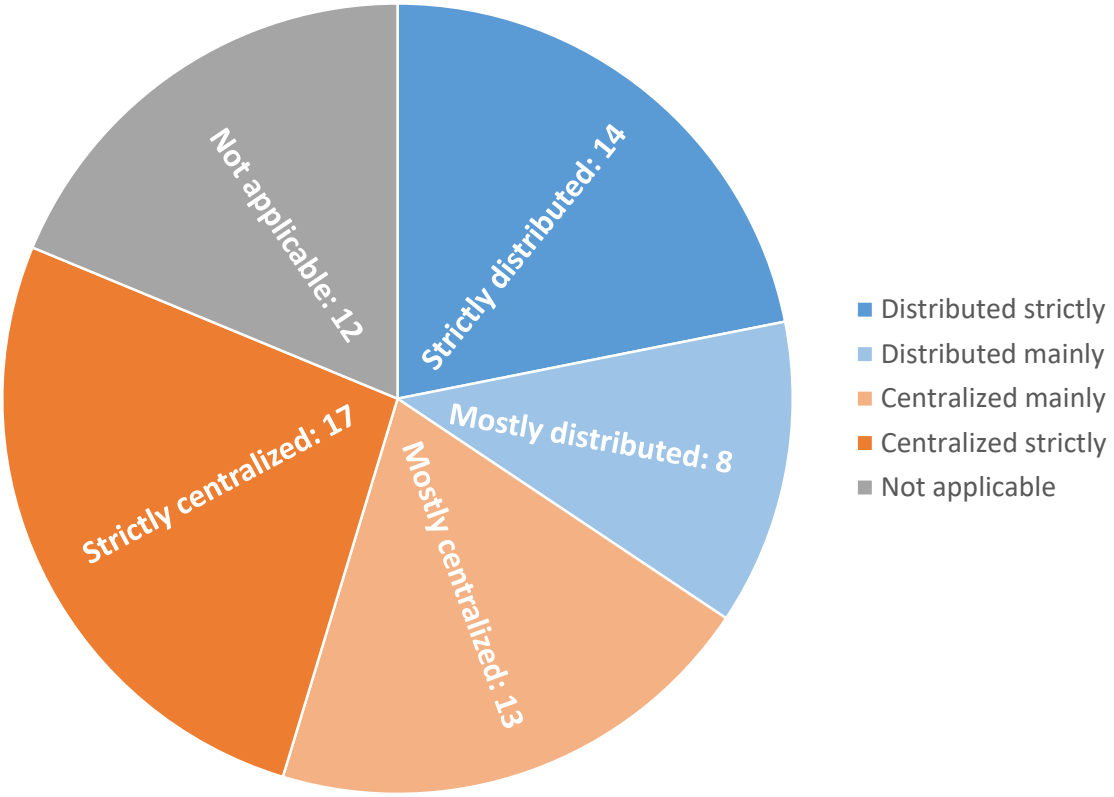


Figure 2.1: Number of keywords found across all articles

2.1.3 Final result set

In the final set that will be used for the statistics and for the overview, the two searches are combined. A total of 63 related publications are categorized and described in the following sections. For a better comparison, I use only the 11 articles from WoS SCIE in the first result set.

2.2 Taxonomy

To find candidate categories based on the most prevalent keywords, I employ the RAKE [64] algorithm for keyword extraction. First, I transform the PDF documents using pdftotxt [65] and strip references and appendices. Then, I apply the RAKE algorithm with the following parameters for keyword extraction: at least five characters, a maximum of two words for the keyword, and at least four occurrences in the text. For each keyword, I then find matching articles. Only keywords present in at least two papers are taken into consideration. I then group synonymous keywords into categories. As a consequence of this approach, a paper

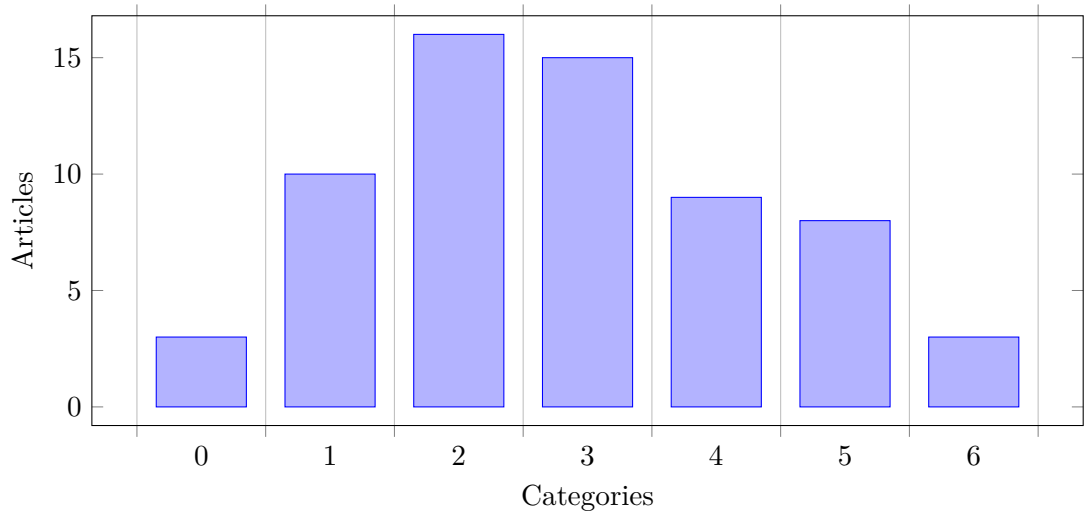


Figure 2.2: Number of categories suggested by RAKE per article

may fall into multiple categories.

The results (excluding general terms) suggest the following categories of papers. They are also illustrated in Figure 2.1

- **authentication:** papers that address authentication [66]–[100]
- **authorization:** articles dealing with authorization [70], [72], [73], [75], [78], [80], [81], [83]–[85], [87], [90]–[92], [95], [97], [99]–[114]
- **service:** solutions that can be used both in IoT and Service Oriented Architecture (SOA) [67]–[69], [73]–[75], [77], [79], [81], [83], [86], [88], [90], [92], [96], [98], [101], [111], [115]–[119]
- **token:** articles that use any form of token as an information bearer in their proposal [71], [73], [75], [86], [90], [93], [96], [99]–[102], [105], [106], [108], [111], [112], [115], [120]
- **cloud:** research addressing security issues of cloud-based IoT devices [66], [68], [72], [87], [88], [94], [97], [99], [106], [110], [119], [121]–[123]
- **context:** papers using or proposing context-aware methods [66], [75], [84], [86], [92], [93], [100], [101], [103], [114], [115], [121], [122], [124]
- **identity management:** solutions discussing identity management [67], [70], [74], [85], [86], [89], [91]–[93], [97], [98], [100], [112], [115], [116], [119], [120], [125]

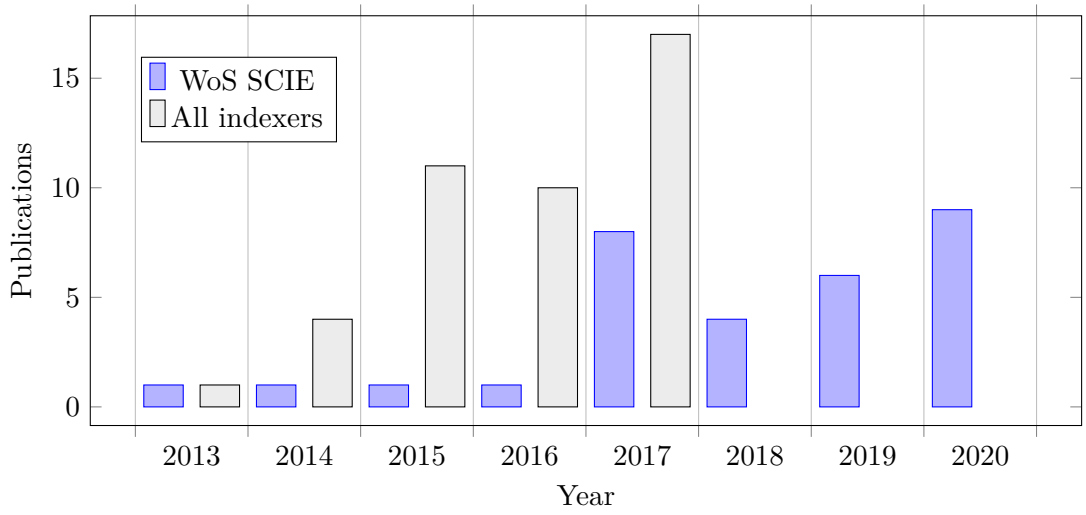


Figure 2.3: Number of publications per year

- **attribute-based:** the subset of authorization proposals that involve Attribute-base Access Control (ABAC) [67], [80], [83], [85], [90], [93], [100], [103], [111], [112], [114], [122], [125]
- **blockchain:** research that utilizes blockchain [84], [95], [97]–[99], [112]–[114]
- **health care:** projects that specifically address the health care domain [70], [74], [80], [83], [95], [106], [114], [122]
- **roles:** the subset of authorization proposals that involve RBAC [70], [80], [90], [113], [114], [126]

Two of the papers do not fit into any of the above categories [127], [128]. One article [127] is likely too short for RAKE to perform any meaningful analysis; I have not identified any apparent reason why [128] is not categorized by the algorithm. Nevertheless, both of the papers address authentication, and I have included them in this category.

In total, slightly over 50% of the articles get two or three keywords. A significant number of research papers fit into one of four categories. Two papers did not fit any category, and another three fit into five categories. This statistic is shown in Figure 2.2. As illustrated in Figure 2.3, the research covered by this survey shows an evident increase in interest in IoT security based on the number of articles published. The grey bars represent the initial data set; the blue bars are articles indexed only in WoS SCIE (manually extracted from the first set and combined with the update search). The chart illustrates steady growth, except for 2017, when the number publications increased sharply.

The authentication, authorization and services categories are described in the subsequent subsections, as they are the most populous categories. Identity management is also described in a separate section, as it is closely related to authentication and also to my research. Articles with context awareness elements are further described in a section of their own.

2.2.1 Authentication

Authentication is addressed by 36 papers from our pool – more than half of the articles in the survey. Authentication is the process of confirming an identity claimed by an entity. In the vast majority of cases, it is confirmation of an identity that the entity claims with the use of credentials.

Traditional authentication methods, enhanced with multifactor authentication based on a location, are described in [66]. The system considers user location, and they develop an additional factor for multifactor authentication, which ascertains the physical possibility of a user being in a particular location, e.g., a user cannot possibly be in Los Angeles if she just logged in from New York. This adds extra security without requiring the user to perform an extra action.

In [67], the authors suggest enhancing privacy during authentication by basing authentication on attributes rather than on identities. A trusted authority issues certificates which prove that an entity possesses a particular attribute; these certificates are used for authentication when communicating with other services. This scheme preserves both entity privacy and the advantages of centralized identity management.

The authentication model for cloud-based IoT is elaborated by Barreto et al. [68]. Their solution supports two authentication stages: one for basic access and a second stage for advanced access, e.g., for administrative purposes. Barreto et al. do not describe specifically how the authentication should be done; instead, they specify methods that cloud services should provide for authentication.

To achieve efficient and smart authentication of IoT devices, Cagnazzo et al. [69] suggest using Quick Response (QR) codes; specifically, XignQR [129]. Every device has a printed QR code that contains important information about it, e.g., an ID representing its service provider, authentication server address, and digital signature. Scanning the QR code and sending it to the authentication manager allows the manager to decide which authentication method it should enforce on the user. This approach can be useful when physically managing large numbers of devices simultaneously, e.g., in a medical environment or in a factory.

A security framework following the Architecture Reference Model (ARM) [130] is

described in [73]. It bases authentication on the Extensible Authentication Protocol (EAP) over LAN [131]. EAP is widely used and recognized as a mechanism to provide flexible authentication through various EAP methods. These methods allow an EAP peer to be authenticated by an EAP server through EAP authentication for network access. While their work proposes interesting solutions, the developers of ARM do not provide any case study or usability study

Kumar et al. [74] assume that the best authentication method for wearables and nearables (devices that are not worn but are generally close to the user) are biometric information of their owner. The proposed solution requires her to register their biometric characteristic(s) in person with the authentication provider. Later, access points close to the user – wearables or nearables – capture the user's biometric information and authenticate it by comparing those characteristics with the registered characteristics. However, there is an issue with privacy, as many users are reluctant to share their personal information. A slightly different method is to measure the user's gait and authenticate the user based on it [89], [94]. The initial gait is trained on a 1-minute walk. The innovative feature of the method is that it improves accuracy by speed adaptive methods and smart threshold calculation for gain template matching. Another very different method of authenticating with the use of biometric information is presented in [88]. It proposes the use of brainwaves for authentication. Users are shown various images that they are either familiar or unfamiliar with, and their reactions are measured through brainwaves.

Three almost identical works proposed the OpenID [39] protocol as an authentication method in the IoT environment [75], [90], [96]. They describe a central service issuing tokens and communicating through RESTful API [132] over the HTTPS [133] protocol, allowing rapid development and acceptance among IoT devices, as all technologies used are proven, well-documented, and widely supported. A downside is that the OpenID protocol was not designed with IoT usage in mind, and can be more demanding of computation and network resources than specialized protocols.

Another framework [76] for authentication is formally described using process algebra, specifically[134]. The framework contains three authentication forms. An *entity* authentication is the capability of verifying the identity that the entity claims. An *action* authentication refers to the authentication of the actions of devices and whether they are allowed. A *claim* authentication verifies the authenticity of devices' claims about previous actions. It also has three strength levels for each form: weak, non-injective, and injective. The paper does not provide any proof of concept or any other kind of demonstration of their solution.

A mechanism of HTTPS-based authentication for IoT devices using a hash-chain generated between the server and the client is explained in [77]. This hash-chain is generated during the login process, and serves as a One Time Password for the client to authenticate against services. If a device does not have the required capabilities (e.g., battery lifetime, computational power, a network connection) to generate the hash-chain, or if those capabilities are in use for other functions, another device acting as a proxy may be used to generate the hash-chain.

Continuous authentication of personal IoT devices is addressed by Shazad et al. [78]. Current practice is to authenticate an entity just once when a session is established, and to keep the devices authenticated until some timeout occurs or the session is otherwise closed. This session persistence presents a potential security risk. The authors divide devices into two categories – those which maintain physical contact with the user and those which do not. Devices that keep contact can be authenticated using various biometric information, both direct (blood flow rhythm) and indirect (using an inertia measurement unit to check the user's gait). The authors propose using radio frequency signals for devices that are not in physical contact with the user. For example, Wi-Fi signals are reflected by the human body, and the resulting distortions can be measured and used to determine users' walking speed, gait cycle, and other physical properties. A different approach for continuous authentication of users is presented in [92]. It describes users' context-aware authentication (and authorization) based on their behavioral patterns observed through IoT devices. The confidence manager does the authentication, and then the results are used both in the authentication process and in the authorization process.

Advanced authentication methods better than the current approaches are suggested in [79]. Most of the traditional methods have flaws or were not designed to be used frequently (e.g., passwords — almost no one can memorize strong and unique passwords for every service or device they use, so users reuse their passwords). The proposal is based on users' digitized memories. Users would authenticate themselves against their digitized memories based on date and time, place, people or pets, devices, habits, audio, or ownership recognition. They map different suitable methods, including choice selection, alphanumeric input, image part selection, or interactive categorization.

Article	Centralized	Decentralized	U2M	M2M	Context-aware	Specifics
[66]	Yes	Yes	Yes	No	Yes	Service answering whether user can be in the given location
[67]	Yes	Yes	Yes	Yes	No	Use of attributes for authentication
[68]	Yes	No	Yes	Yes	No	Authentication through cloud
[69]	Yes	Yes	Yes	No	No	Reading QR codes physically present on a device
[70]	Yes	Yes	N/A	N/A	No	Framework designed to preserve patient privacy
[71]	Yes	No	Yes	Yes	No	Adjustment of Web API management; OpenID Connect
[72]	No	Yes	Yes	Yes	No	Authentication for devices with constrained computational power
[73]	Yes	No	No	Yes	No	ARM compliant; EAPoL; RADIAL
[74]	No	Yes	Yes	No	No	Biometric from wearable and nearables
[75]	Yes	No	Yes	Yes	No	OpenID Connect
[76]	Yes	No	Yes	Yes	No	Authentication framework mathematical description using CSP algebra
[77]	Yes	No	Yes	Yes	No	HTTPS-based device authentication using hash chain as One Time Password
[78]	N/A	N/A	Yes	No	Yes	Biometric; continuous authentication
[79]	Yes	No	Yes	No	Yes	User's electronical history
[80]	Yes	No	Yes	Yes	No	Authentication based on attributes
[81]	N/A	N/A	No	Yes	No	WS-Security adaptation for IoT
[82]	N/A	N/A	Yes	No	No	One time passwords using words chosen by a user
[83]	Yes	No	Yes	Yes	No	Full security framework
[84]	No	Yes	Yes	Yes	No	Blockchain access control framework
[85]	N/A	N/A	No	Yes	No	Authentication on perception level
[86]	No	Yes	Yes	Yes	Yes	Privacy preserving based on partial identities
[87]	Yes	No	Yes	No	No	Smart home, FIDO
[88]	Yes	No	Yes	No	No	Privacy preserving based on partial identities
[89]	Yes	No	Yes	No	Yes	Authentication based on gait
[90]	Yes	No	Yes	Yes	No	OpenID Connect
[91]	Yes	No	Yes	Yes	No	Authenticaiton based on functional right
[92]	No	Yes	Yes	Yes	Yes	Continuous context-aware authentication
[93]	Yes	No	Yes	No	Yes	Confidence score calculated from context
[94]	Yes	No	Yes	No	Yes	Gait analysis; ECG like signal processing
[95]	No	Yes	Yes	Yes	No	Blockchain
[96]	No	Yes	Yes	Yes	No	OpenID Connect
[97]	No	Yes	Yes	Yes	No	Blockchain
[98]	No	Yes	Yes	Yes	No	Blockchain
[99]	No	Yes	Yes	Yes	No	Blockchain
[100]	Yes	No	Yes	Yes	Yes	XACML

Table 2.5: Summary of authentication articles

Wiseman et al. [82] present an interesting niche problem, along with a solution. They address the issue of pairing an IoT device with its "master" account. Connecting from devices using a password can be difficult, or even impossible because of the lack of a proper input method. One method to avoid this is to let the device display an access code and add the access code to the master account. Wiseman et al. examine this process from a user experience perspective, and compare convenience between alphanumeric codes and codes generated from human-readable words.

A privacy-preserving, decentralized identity management framework for the IoT is presented in [86]. Identity in the IoT is extended not only to users but also to IoT devices themselves, using an ARM-compliant, claims-based approach built on top of Identity Mixer technology [135]. The authors define partial identities as subsets of a user or device's virtual identities that preserve privacy while being sufficient to provide identity confirmation. They show the use of their framework with Distributed Capability-Based Access Control [73]. Identity attributes are disclosed by specific proof, and are employed during authorization based on XACML rules to obtain the capability tokens that are used to access a service.

Khalid et al. [97] decentralize authentication using blockchain technology. There is a fog layer for every domain or application to allow authentication (and possible authorization rules storing) of the devices. When the device connects to a network, it finds a close fog authentication server and authenticates through it. It receives a private key, and a public key is stored in the blockchain. Devices can communicate only with devices that are authenticated, and their identity is propagated in the blockchain. A similar blockchain-based approach is used in [98]. It uses multiple blockchains for communication of IoT devices, where there are multiple local blockchains and a single global blockchain. This categorizes devices into simple devices, proxy nodes, and manager nodes. Proxy nodes authenticate (and authorize) near constrained devices and use a local blockchain for it. The local blockchain is restricted to a specific application or deployment. If a device wants to communicate with a device outside of its network, it uses a manager node that is part of the global blockchain. Another similar blockchain method, which also uses a fog layer, is proposed by Pallavi et al. [99].

Finally, there is a group of papers [70]–[72], [80], [81], [83]–[85], [87], [91], [93], [95], [100] that address authentication tangentially either as part of a broader and more complex framework or project, or to solve authentication issues as a side effect of dealing with another problem.

Table 2.5 presents an overview of authentication research, reflecting information that I extracted from the papers. It shows which solutions support centralized and

decentralized architectures, which are for U2M or M2M communication, and which possess at least some elements of context-awareness.

2.2.2 Authorization

Authorization is the process of granting permission to execute specific actions to given entities – in our scenario, specifically to users, devices, or applications. There are a total of 32 articles in the identified pool addressing this topic. Authorization ties with Services as the second-busiest category.

Access control based on trust in an ARM-compliant model is proposed by [101]. This paper describes various levels of trust, a multidimensional attribute that describes various concerns in the network. The authors specify the following dimensions: quality of service (including network availability and throughput), security (e.g., authentication and authorization protocols, encryption), reputation (recommendations from other devices), and social relationship (the group or groups of IoT devices to which an individual device belongs, e.g., those made by a specific manufacturer or those currently in a particular location). This trust is used for final authorization within the environment.

The authors of [70] describe a complex framework for use in the healthcare field. They employ a version of RBAC where a user, specifically a patient, grants permission to access his data based on a particular role – a group of doctors and nurses. A centralized authentication server enforces the resulting security rules.

Another paper [102] develops an authorization architecture based on IoT-OAS [136], authenticating users using tokens similar to those used in OpenID. Each device has a designated owner and a set of actions or permissions. Users may request and share permissions with one another; multiple operational cases are described in the paper.

Gerdes et al. [72] tackle the problem of authorization and authentication for devices with constrained computational power. The authors divide IoT devices into a "constrained" category and a "less constrained" category, based on resource availability, and they allow less-constrained devices to perform some authorization functions on behalf of the constrained devices. The paper includes basic methods for these authorization management tasks. "Principal actors", representing the person or the company that owns the specific device or the data on the device, must set appropriate policies for each situation about which tasks can or cannot be offloaded.

A solution to the problem of data access control across a shared network is developed in [103]. The authors use Ciphertext-Policy Attribute-based Encryption

[137], and enhance it with a set of policy descriptions in an eXtensible Markup Language (XML) file. The access policies are based on entity attributes and are structured as a binary tree with "And" and "Or" operations available. Entities present a keyserver with a list of their attributes, and the keyserver generates a key that can only decrypt data to which the listed attributes allow access. A similar approach that is discussed in [91] is aimed at reducing privileged access. It suggests giving access to functionalities rather than assigning roles or attributes. In this approach, functionality consists of two elements – data type and the actions that are allowed to them. The rules are enforced by identity-based encryption [138] performed on a cloud server.

The framework introduced in [73] supports not only authentication but also authorization, which is enabled by creating an suthorization server that issues access tokens according to security rules stored in XACML [139], an XML schema for representing authorization and entitlement policies. Entities request authorization tokens based on their attributes, and then use the tokens to access services provided by, or data stored on, another server or device.

Kurniawan et al. find classic security strategies unsuitable, because they are centralized and scale poorly in the IoT environment. They propose a trust-based model [104] based on Bayesian decision theory. The authors compute Bayesian trust values based on three inputs: experience (the history of interactions between the actors), knowledge (what is already known about the entity and the context), and recommendation (how much trusted peers trust the entity in question). These trust values are used as input to a loss function that determines the cost of an action. Access control decisions are made based on the loss function's output, given a particular trust value.

Numerous proposals based on the existing OAuth protocol [140] use tokens that encode the access rights (e.g., roles or attributes) of the token owner and a configurable lifespan. Some methods [90], [106] use JSON Web Token (JWT) [141]; another proposal [75] use a special token format which allows for additional features. All the proposals communicate through a RESTful API.

Another framework for securing API-enabled IoT devices in smart building [108] is also inspired by OAuth and uses JWT. The proposed security manager is split into two services to enable better scalability. The first service is an authentication manager that authenticates users or services with a process similar to, but not identical to OAuth, and then issues a JWT. The second service is an access control manager that verifies whether the access is allowed, based on XACML rules set by the system administrator and the identity of the requesting side (which is provided

by the token).

Alkhresheh et al. also build their framework [100] around XACML policies. Their framework eases maintenance and increases security by generating XACML policies based on attributes, context, and predication. The policies are then continuously enforced. Administrators of the system describe the policies in the elementary format, consisting of simple policies that together form more complex policies and are used to generate XACML policy dynamically.

Blockchain technology is used in [105] to store, distribute, and verify authorization rules. Each node in the network has a full database of all access control policies for each resource-requester pair in the form of transactions. Access is granted by giving a token to the requester entity and propagating it in the blockchain. The blockchain also serves as an auditing and logging tool. Trust in the network is based on the distributed nature and the large size of the network; it is challenging to gain unauthorized access or to disable the network by attacking a central element. A slightly different approach using blockchain is presented in [84]. Rules based on OrBAC [142] are distributed through a blockchain, and, based on the history of the communication, the rules are updated with reinforced learning algorithms. Another blockchain utilization is shown by [95]. The article describes cross-domain permission sharing and access control, which is currently done by a trusted third party or resource owner. The article introduces authentication and authorization sharing based on the blockchain, which mitigates the risk of a single point of failure. The security rules are enforced by "smart contracts", which are either local, e.g., per domain or per deployment, or single global, which store global security policies. As the blockchain principle is currently a trending research topic for the IoT, there are also other authorization framework proposals based on it [99], [112].

Tasali et al. [80] discuss current standards for healthcare devices, including Integrated Clinical Environment [143] and Medical Application Platform [144]. The conclusion is that they barely address authorization and authentication (if they address it at all). Their solution is based on ABAC, enhanced with attribute inheritance inspired by RBAC. Attribute inheritance allows the "plug-and-play" configuration of new devices based on device types represented as attributes pre-set on the devices.

Another option is to isolate each function of the device and provide access just to that functionality [80]. Functionalities have some similarities with the concept of microservices. The proposed functionality-centric access control framework mainly reduces application-level attacks on "Misused functionality" or "Reduced functionality".

Article	Centralized	Decentralized	U2M	M2M	Context-aware	Specifics
[70]	Yes	Yes	N/A	N/A	No	Rules tied to the data
[72]	No	Yes	Yes	Yes	No	Constrained devices
[73]	Yes	No	No	Yes	No	ARM compliant; describes access control generally
[75]	Yes	No	Yes	Yes	No	OAuth; tokens
[78]	N/A	N/A	Yes	No	Yes	Biometric information used
[80]	Yes	No	Yes	Yes	Yes	Supports with attribute inheritance
[81]	N/A	N/A	No	Yes	No	WS-Security adaptation for IoT
[83]	Yes	No	Yes	Yes	Yes	Full security framework
[84]	No	Yes	Yes	Yes	No	Reinforced learning to update rules
[85]	N/A	N/A	Yes	Yes	Yes	Perception layer framework
[87]	Yes	No	Yes	No	No	Smart home
[90]	Yes	No	Yes	Yes	No	OAuth
[91]	Yes	No	Yes	Yes	No	Functionality based
[92]	No	Yes	Yes	Yes	Yes	Continuous context-aware authorization
[95]	Yes	Yes	Yes	Yes	No	Blockchain; policies sharing
[97]	No	Yes	Yes	Yes	No	Blockchain
[99]	No	Yes	Yes	Yes	No	Blockchain
[100]	Yes	No	Yes	Yes	Yes	XACML
[101]	Yes	Yes	Yes	Yes	No	ARM compliant; ABAC; trust based
[102]	No	Yes	Yes	No	No	Tokens; Possible to share permissions
[103]	No	Yes	N/A	N/A	Yes	Data decryption only with correct attributes
[104]	No	Yes	N/A	Yes	Yes	Bayesian decision theory for authorization
[105]	No	Yes	Yes	Yes	No	Propagation through blockchain
[106]	Yes	No	Yes	Yes	No	OAuth; tokens
[107]	Yes	Yes	Yes	Yes	No	Access control specified for functionalities
[108]	Yes	Yes	No	Yes	No	OAuth; XACML; tokens
[109]	N/A	N/A	No	Yes	No	Constrained devices
[110]	Yes	Yes	No	Yes	No	Gateway, device and cloud share data encryption
[111]	No	Yes	Yes	Yes	Yes	User centric; smart power grid
[112]	No	Yes	Yes	Yes	No	Blockchain; capabilities
[113]	No	Yes	Yes	Yes	Yes	Continuous trust verification
[114]	Yes	No	Yes	Yes	Yes	RBAC; teams

Table 2.6: Summary of authorization articles

Djilali et al. [114] build on top of an RBAC authorization system that assigns users and devices into teams. The teams are one-off collaboration units and are created ad-hoc and last only as long as the collaboration is needed. The security rules are enforced by a central server that has access to global and team context.

A proposal for energy-constrained devices called Time Division Multiple Access is described in [109]. The schema is well suited for sensors with known communication patterns, such as a repeating communication schedule in which sensors periodically report data. The proposed communication scheme optimizes the trade-off between device lifetime and distortion of the data that is transmitted. A different application of ABAC, focused on reducing the storage and communication overhead, is described in [85].

Sicari et al. provide a full specification for a security framework for smart healthcare [83]. They describe three main points (locations) for policy enforcement

– a policy administration point, a policy enforcement point, and a policy decision point. The access roles are described using XML in a format inspired by ABAC. Another domain-specific article [111] describes a user-centric IoT platform to empower users to manage security and privacy concerns in the Internet of Energy. There is also a specific solution for a smart home environment [87], which extends FIDO [145]. A user on his phone needs to authorize all the device's actions. When the user acquires a new device, he needs to register it and provide authorization attributes. For this, the user uses the registration token issued by the manufacturer and provided with the device.

Another access control model for IoT running in the cloud [110] secures data using hierarchical attribute-based encryption. The encryption is done in two steps. The first part of the encryption is done on the device; the secondary encryption is done on the gateway. This reduces the load on the device. Decryption is likewise split between the cloud and the device in order to save application resources. The encryption scheme's hierarchical nature allows security policies to be updated using an update key based on information from the data source, without the device itself needing to re-encrypt the data.

Four of the reviewed papers [78], [81], [97], [113] discuss authorization only tangentially. The complete overview of authorization research is presented in Table 2.6.

2.2.3 Services

This section presents an overview of solutions that either support IoT-as-a-service or provide security-as-a-service. This means that, at a minimum, the security client (an entity) or the security provider follows the principles of SOA [A.14]. Frequently, both of the actors can be viewed as services. In this research review, I have 16 research publications that include SOA compatibility, although not every paper in this category uses the term SOA or "service"; instead, they are frequently called by synonyms, e.g., "central entity", "authorization or authentication server". Most centralized security approaches can be viewed as a service.

Most of the surveyed proposals contain an identity management, authentication, or authorization service. An application in the IoT environment may offload the authentication process to such a security service [68], [69], [73], [75], [79], [83], [86], [88], [90], [96], [98], [116], [119]. A few proposals also allow the distribution of access rights or other properties used for authorization from the service to its clients [67], [73], [111], [115]. Some of the services also provide additional features such as enhanced user privacy [67], [77], [92], [115]. They anonymize entity identities by

hiding identity details from the service provider, and guarantee the entity's identity by the trustworthiness of the identity management service itself.

Two of the papers in this category stand out. The first adapts the Web Service (WS) Security specification [146], which is intended for loosely-coupled distributed systems, to the IoT environment by extending it to allow identity management functions to be offloaded from computationally "weak" devices to "strong" devices [81]. The proposed method, termed DPWSec, also simplifies the original WS-Security specification by removing unneeded portions: multi-hop security, statelessness, hosting and hosted devices, and the device profile communication model. The second paper describes a security framework within the scope of the Device Profile for Web Services using the XACML standard for rule description [118]. Three parts of the framework are described – the policy enforcement point (where the policies are enforced), the policy decision points (where the policies are evaluated), and the policy information points (where the audit logs are kept).

2.2.4 Identity Management

Identity can be viewed as a set of user attributes, either virtual or real. Identity management is the mechanism for storing and retrieving user identities. Typically, users are forced to have several unconnected identities for various services. In the IoT environment, the identity should be available for the whole IoT network (or at least for some significant part) while preserving the user's privacy, although this does not mean that each user must have a single identity. The identity concept is also extended from users to include sensor identities in the IoT. Identity management is closely connected to authentication, which verifies that a user (or a device) is the owner of that identity, and authorization, which is the process of granting access to a resource based on user attributes (i.e., identity). Eight of the articles in this category address identity or identity management at least partially.

Traditionally, user identity contains the principal along with the credentials used for authentication. This entails a privacy risk, especially if the identity is shared with multiple services whose operators are not known in advance. These operators may appear on and disappear from the network at any time in the dynamic IoT concept. Many of the articles tackle the problem of privacy by limiting a user's identity to only their attributes, without any unique information that could lead to the disclosure of their identity. One proposal is for a trusted party to issue cryptographic containers containing user attributes [67]. It is not specified that the trusted party must be a single entity in a network, so we can assume that

multiple trusted parties can exist simultaneously. The use of attributes instead of identity for authorization is also proposed in [120]. Gusmeroli et al. propose a slightly different approach using capabilities instead of attributes [115]. This proposal also supports anonymous capabilities that allow authentication without disclosing identity. Fremantle et al. describe the federated identity model [119], based on OAuth 2.

The problem of assigning an identity to devices is described in [116]. An IoT device inherits its user's identity through various methods based on a relationship between the user and the device. The authors formulate methods for devices strictly connected to a single user, and provide identity extensions from users to devices that frequently change users.

A complete framework for decentralized identity management to enhance user privacy is introduced in [86]. The framework defines partial identities as the least sufficient subsets of full identities for a requesting service that does not disclose any unnecessary information about a user. A different decentralized identity management framework [125] takes the device's trust into the context. The trust is dynamically calculated based on the history of interactions and the trust of the other participants. There is also a very similar concept [93] of confidence, which is calculated from contextual information.

The principle of storing a user's biometric information in access points, serving as identity servers, and thus linking a real user's identity with his virtual identity through wearables, is presented in [74]. Some articles suggest moving the identity management part into a blockchain network [97], [98], [112]. The rest of the articles [70], [85], [89], [91], [92], [100] deal with identity management only partially, and the main contribution of their work lies in other areas.

2.3 Context awareness

One trend in present-day application development is towards context-awareness. Context is defined by Abowd [5] as any information that can be used to characterize an entity's situation. An entity is a person, a place, or an object that is considered relevant to the interaction between a user and an application, including the user and the application themselves. In the context of the IoT, the concept of the entity can be extended not only to an interaction between a user and an application but also to an interaction between two applications.

Solutions using context-aware security can provide a much better user experience as well as increased security [49]; often, both can be achieved at the same time.

However, context-awareness has attracted less interest in from the viewpoint of security than from the viewpoint of user experience. This is probably because computer security is traditionally a more conservative domain of computer science. In this section, I focus on research that speaks to an interest in context-aware security.

The most common approach to achieve context-aware security is with the use of ABAC. ABAC differs from RBAC in that an entity (a user or a device) performing an action is not authorized based on matching the roles assigned to it to roles that allow specific actions. In ABAC, every action is mapped to a specific set of attributes that an entity must possess in order to take that action. An example of such a rule for reading a document is that the entity must be from the same department as the creator of the document, must be employed in a management position, and must be located in the same building or complex.

One option is to specify access rules using ABAC for each piece of data at creation time and to join those rules with the data so that during network transportation, updates, or copying, the rules stay consistent. In order to manipulate the data, an entity must possess the specified attributes [103]. Another method is to use a three-module architecture. The first module, the policy enforcement point, is responsible for invoking checks on access rules. The second module, the policy information point, gathers information about an entity's attributes, including their context. Finally, the policy decision point compares security rules with the information gathered about the entity, and decides whether the action is allowed or declined [80], [122]. Security rules can be written in XML using XACML [80], [100] or using the Ontology Web Language [122]. While [83], [85], [111], [117], [120] and [114] do not mention context information specifically, the ABAC implementations in these papers could also utilize context-aware attributes.

Instead of extending ABAC, another option is to adapt the well-described Capability-based Access Control [147] architecture. A capability (known in some systems as a key) is a communicable, unforgeable token of authority. It refers to a value that references an object along with an associated set of access rights. This token may contain additional contextual rules, defined in XACML format, which must be satisfied for the token to be valid [115]. A variation on this is to use Distributed Capability-Based Access Control [73], as described in [86].

A novel authorization architecture based on Bayesian decision theory [104] also considers context. The trust parameters of history, knowledge, and reputation (described in the Authorization section) may include contextual elements that are either acquired directly by the device itself or are provided indirectly by a peer

device. Machine learning techniques used to enhance access rights [84] also consider the context in terms of the history of the previous interaction.

Biometric information may be considered "contextual" by definition, so biometric authorization is context-aware [74], [78], [89], [94]. Many devices, especially wearables, directly measure the user's physical traits (e.g., heart rhythm or body temperature). Other "nearable" devices can provide additional information such as weight or gait, both of which can be measured by video sensors. All of this information can be compared to a user's known physical or kinesiological properties.

Beyond simple biometric data, a user's digital life may be considered as a context for identity management. A user's photos, videos, blog posts, and browsing history can be used to authenticate that user [79]. Given sufficient digital history, security questions can be devised which no-one but the authentic user can answer. This benefits the user by not needing to memorize passwords or carry other credential material; the user's own memories are sufficient. Another similar proposal, which restricts context to information from network traffic, authenticates the use of contextual information provided by a smart home [121].

A different approach is to evaluate the history of actions. This can include communication patterns, actions performed, or even a typical context at the given time for a user or for a device. For sensors, the values they produce can be observed, and some patterns or limits can be determined [113]. Then the information is used as an additional factor for authentication. A similar approach can be used on users. IoT devices can monitor the user's activity, and the usual patterns can be evaluated for authentication [92]. Alternatively, communication history can be evaluated – the device's current trust can be calculated based on the participants and their trust at a given moment [93], [126].

2.4 Existing vs. novel approaches

Existing research projects in IoT security that propose an actual solution or method can be roughly aligned to two categories: those which extend or adjust existing architectures or programs to better suit the IoT environment, and those which propose entirely new ideas for solving environment-specific problems. However, the classification is not strictly binary, and it is often difficult to judge the novelty of any particular proposal. The reader will point out that all research is meant to be "novel"; I use the word here in a narrower sense to mean an entirely new approach that does not make use of existing technologies or standards.

The works considered here that apply or adapt existing technologies and methods

from other security domains to the IoT environment are often based on OAuth 2 technology [71], [76], [90], [96], [106], [108], [119]. Two proposals also adapt the WS-Security specification to IoT devices and to communication between them [81], [118].

The most innovative solutions share some common properties. Most of them are suitable for distributed use, and none require administrator interaction. They can handle device connection and disconnection, as well as security rule distribution and validation. In many cases, the responsibility for the creation of access rules is moved from administrators to data owners. Some papers show operation with trust between devices and dynamic calculation of trust among various communication partners[93], [101], [104], [120], [125]. One proposal adjusts ABAC to be more dynamic and to allow a device to pick its own attributes; other devices must subsequently confirm that the device really does possess the claimed attribute. Security rules are set during data creation using ABAC, and are then connected to those data for its whole life-cycle [117]. Other innovative approaches suggest propagating all security rules through a blockchain in the network [84], [95], [97]–[99], [105], [112]–[114]. One study proposes access control based not on roles or attributes, but rather on functionalities of the IoT node [107]. Access control for cloud applications based on attributes [110], using the computational power of sensor gateways of the cloud itself, is suitable for constrained devices.

In summary, there are various novel proposals [84], [93], [101], [104], [105], [107], [110], [117], [120], [125], especially focusing on distributed solutions [84], [95], [97]–[99], [104], [105], [110], [112]–[114], [117], [120], that potentially suit the IoT environment well in terms of scalability, maintainability, and flexibility. However, due to their novelty it is difficult or impossible to predict which ideas might be adopted or might come into wide use. A significant amount of research within [71], [75], [81], [85], [90], [96], [106], [108], [118], [119] is focused on the adoption of existing technologies. All of these papers have presented promising results.

2.5 Distribution vs. centralization

The IoT is a diverse, complex, and fast-changing environment. It comprises a large number of devices that interact autonomously. Objects also appear and disappear autonomously and with high frequency. Given these differences from a more standard network environment, I focus in this section on the paradigms that are used in security solutions.

A conventional centralized approach is straightforward for system administrators

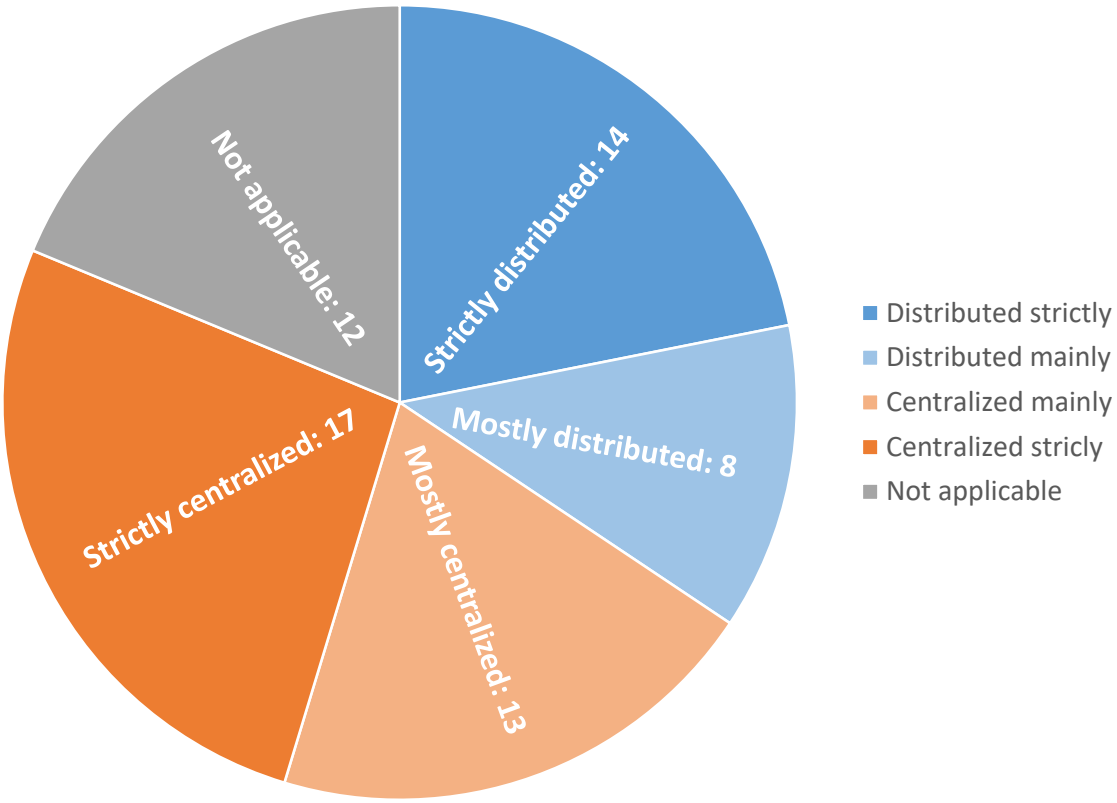


Figure 2.4: Categorization of distributed and centralized solutions

to set up, maintain, and audit. It also forms a stable point in the network from which users and applications can build trust. Implementing centralized solutions is simpler for the central server, and also for applications using it. Many of the existing centralized solutions for networks and applications can be extended to operate in the IoT environment without overly costly adjustments. However, using a centralized architecture in the IoT presents several drawbacks, including limited flexibility and scalability.

By contrast, the attributes of distributed architectures are entirely opposite. They scale well and are built with flexibility as the primary goal. However, synchronization, maintenance, and auditing present serious difficulties. There is also the issue that business users, legal entities, and others, may require a single trusted central entity to stand behind their security solution. A distributed architecture does not meet this requirement.

To further complicate matters, the line between a distributed solution and a centralized solution is often not clear. While some solutions can be considered exclusively in one category, a significant number of proposals may work under both paradigms. Figure 2.4 shows a chart of distributed and centralized solutions.

Requiring a central server for identity management prevents distributed operation, for obvious reasons. This limitation is sometimes imposed for domain-specific reasons (e.g. in the healthcare domain [70], [74], [83], [106], [111], [114]), at other times it arises simply as a function of the technologies or methods that are employed [71], [75], [87]–[90], [94], [119], [121], [124]. In one proposal, the authentication method requires having as much historical data about an entity as possible, to the point that the authentication data storage requirements make it impractical to host the data at multiple locations [79].

At the other end of the spectrum, the technologies used in some proposals specifically preclude centralization. For instance, methods which rely on the creators of data to specify security rules, or which grant access selectively, do not operate with a central server [72], [86], [96], [102], [104], [111], [113], [117], [127]. Blockchain-based access rule verification [84], [97]–[99], [105], [112] also cannot be centralized, and the same applies to extensions of the ABAC system which rely on peer devices to confirm an entity's attributes over the network [120].

Most of the ideas in the papers surveyed can be used in both centralized and decentralized architectures. Centralized solutions can often be decentralized by multiplying the central elements [66]–[68], [73], [77], [80], [91]–[93], [95], [101], [107], [108], and decentralized proposals can be centralized by limiting the number of security control elements to a single node [69], [103], [110], [115], [120], [126]. Similarly, some of the research that I reviewed [78], [81], [82], [85], [100], [101], [109], [118], [123]–[125], [128] cannot be categorized in either category. They work equally well for either architecture without modification, and can be seen as complementary extensions for complex security solutions, helping with particular issues (e.g. authentication, auditing, context awareness).

2.6 User vs. device-centrism

In IoT, two basic communication patterns exist: either a user interacts with devices, or devices interact among themselves. The first type is designated as the U2M category. The other scheme of communication is designated M2M. Some of the proposals fit both patterns; this section explains how security models support particular communication models, and the limitations that are encountered.

One important restrictive factor is the need for human input into the interaction. In some cases, various items of information about the actual user are required for security reasons: biometric information [74], [78], [88], [89], [94], [128], a user's digital history [79] or real world history [92], or a user's location [66]. Other

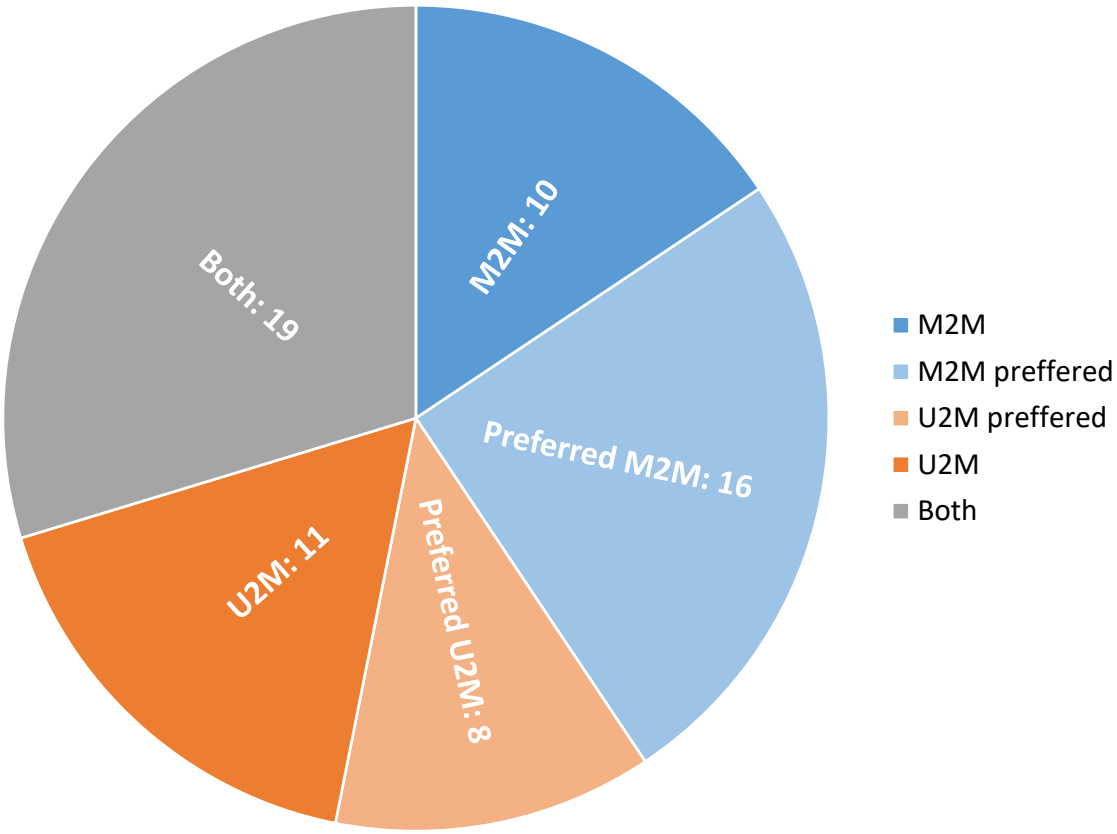


Figure 2.5: Categorization of U2M and M2M solutions

approaches require direct user interaction, e.g. scanning a QR code, providing input on a phone device, or generating a password using words [69], [82], [87]. Any of these cases requires U2M communication.

Generally a device is capable of performing constant and repetitive tasks, but its decision capabilities are limited: goals or objectives can only be set by a user. Users, on the other hand, may find monotonous or continual-load requirements onerous at best and impossible at worst. Given these differences in capability, the adaptation of existing M2M security technologies [73], [81], [85], [108], [125], [127] works well for IoT scenarios where a user is not required. Proposals exist for M2M authentication even with low-resource devices [72], [109], [110], [113].

Finally, many of the solutions described in U2M research can be used for M2M identity management with little to no modification [66], [70], [76], [83], [86], [93], [102], [122] and vice versa [67], [80], [84], [95], [97], [101], [104], [105], [107], [111], [112], [118], [120]. Some of the research even includes existing U2M technologies being used for M2M purposes [75], [106], and many of the papers surveyed are useful for either communication model [68], [71], [77], [90], [91], [96], [98]–[100],

[103], [114]–[117], [119], [121], [123], [124], [126].

Figure 2.5 shows that research contributions in the U2M communication model occur with similar frequency to those in the M2M model. The vast majority of projects can be used for either communication scheme, which demonstrates the versatility of the security solutions and proposals.

2.7 Threats to validity

A literature overview is a highly subjective type of research and therefore suffers from threats to validity. I have identified several threats that need to be addressed or at least mentioned. In order to eliminate most of them, I have followed recommended guidelines for conducting systematic studies [58].

Limiting the search by automatically eliminating articles that contain the keywords "Network", "Hardware", "RFID", "Protocol" and "Cryptography" potentially discards interesting articles that are within the scope of our investigation. Especially those that deal with "Wireless sensor networks", which are also usable in the IoT environment. These keywords were excluded solely to narrow down the scope of the results for a manual search, and it is possible that interesting and relevant research has been missed.

The evidence selection is based on professional indexing sites. I may have missed some articles published in other sources (e.g., journals not indexed in WoS). In addition, the queries that I used to search for articles explored only abstracts. This means that some articles that should have been included may have been omitted because they contained some of the excluded words or did not contain any of the included words. I tried to eliminate this issue by testing our queries against the manually-selected control set.

Data extraction bias is another possible threat to validity. I addressed this primarily by ensuring that each paper received several individual reviews focused on each research question. Using the RAKE algorithm to extract keywords from the papers also to some extent mitigated data extraction bias, because the same extraction method was applied to each paper. Data were acquired at two different points in the time, in 2017 and 2020. They were also processed with a gap of three years. In that period of time, my subjective view on the articles could have changed, and therefore the selection by reading an abstract (or by reading the whole article) may have evolved slightly. Also, as noted, the second time the review was done, only WoS SCIE was examined, and this may have left a significant paper unnoticed. However, it is probably safe to assume that every significant research

paper is published in a journal indexed by WoS.

Exclusion and inclusion of papers due to their scope is also a potential threat. To mitigate this threat, I followed the methods for establishing selection criteria suggested in [58]. I read numerous related works and spent a considerable amount of time reading the selected papers to ensure they fit within the scope of our study. I removed papers that focus specifically on cryptography, networking, and low-level device security. I also excluded papers that did not provide specific results, and that listed only suggestions or opinions without proposing a solution.

All of the papers were treated equally in the survey, although not all published research has the same quality or the same impact on the community. I have provided some overview of the impact of each article in Table 2.7 and Table 2.8, including metadata about the impact of the paper and the probable quality of the publication source. To measure the impact on the community, I chose two sources: data from publishers and from Google Scholar [148]. Publishers generally provide their own list of citations. One disadvantage of using this publisher-provided data is that it may tend to miss citations from sources unknown to it. Google Scholar was therefore chosen as a universal and fully populated article aggregator. Google Scholar provides its own citations list, but these citations also include self-citations, and it may take a few months for articles or citations to appear in Google Scholar. To quantify the quality of the publishing media, I chose two methods. For journals, I used the Impact Factor [149] from WoS SCIE (2019), which is widely recognized as the most prominent and possibly the oldest journal indexing tool. It proved more difficult to rank conferences. The most appropriate measure for our needs seems to be the latest 2020 Computing Research Education (CORE) Association of Australasia conference ranking [150] , which presents independent rankings of sponsored conferences. CORE ranks conferences with letters C, B, A, and A* for their quality (A* is the highest ranking, C is the lowest). A disadvantage is that not all conferences are included in the ranking list, and the ranking itself is managed by a small group of scientists from a particular geographic area. The citation numbers were updated on February 7th, 2021.

Article	Published in	IF or CORE	Year	Source citations	Google citations	Views
[66]	Conference	N/A	2016	6	19	437
[67]	Conference	N/A	2016	16	23	526
[68]	Conference	N/A	2015	7	43	586
[69]	Conference	N/A	2016	3	6	898
[70]	Journal	2.892	2017	9	22	997
[71]	Conference	A	2015	7	22	1400
[72]	Book chapter	N/A	2015	2	2	13
[73]	Journal	11.42	2015	82	117	2400
[74]	Conference	N/A	2017	10	32	629
[75]	Journal	1.151	2017	2	6	1826
[76]	Conference	B	2014	0	1	1400
[77]	Conference	N/A	2016	4	3	621
[78]	Journal	4.231	2017	32	47	3489
[79]	Conference	C	2015	6	13	275
[80]	Conference	C	2017	2	9	365
[81]	Conference	N/A	2015	2	5	221
[82]	Conference	A*	2016	1	3	337
[83]	Journal	N/A	2017	11	30	85
[84]	Journal	N/A	2017	32	94	N/A
[85]	Journal	N/A	2014	97	88	N/A
[86]	Journal	1.508	2017	25	38	886
[87]	Journal	13.727	2018	26	76	216
[88]	Journal	2.645	2018	3	4	895
[89]	Journal	9.936	2018	11	27	1066
[90]	Journal	2.645	2019	5	7	1281
[91]	Journal	13.727	2019	5	7	48
[92]	Journal	2.645	2019	5	6	1327
[93]	Journal	9.936	2019	1	4	363
[94]	Journal	3.745	2020	1	1	480
[95]	Journal	3.745	2020	2	1	1000
[96]	Journal	1.151	2020	0	0	852
[97]	Journal	3.458	2020	15	28	1722

Table 2.7: Community impact of articles (Part 1/2)

Article	Published in	IF or CORE	Year	Source citations	Google citations	Views
[98]	Journal	0.648	2020	0	0	N/A
[99]	Journal	1.061	2020	0	0	121
[100]	Journal	9.936	2020	0	1	217
[101]	Journal	3.05	2016	67	112	1829
[102]	Conference	N/A	2015	6	8	297
[103]	Conference	C	2015	6	11	235
[104]	Conference	N/A	2015	6	9	413
[105]	Conference	N/A	2017	92	235	5100
[106]	Conference	N/A	2016	10	23	986
[107]	Conference	C	2017	13	30	351
[108]	Conference	B	2016	8	17	458
[109]	Journal	6.779	2017	7	14	652
[110]	Journal	2.892	2017	19	28	855
[111]	Journal	1.151	2017	3	8	1776
[112]	Journal	9.112	2020	2	5	436
[113]	Journal	3.275	2020	1	1	1180
[114]	Journal	1.594	2020	0	0	N/A
[115]	Journal	1.366	2013	185	291	240
[116]	Conference	N/A	2017	2	4	1000
[117]	Conference	N/A	2016	1	4	1600
[118]	Conference	N/A	2014	11	19	129
[119]	Journal	1.546	2018	3	11	1938
[120]	Journal	11.051	2017	20	37	1141
[121]	Conference	B	2015	88	241	5886
[122]	Conference	N/A	2014	3	8	395
[123]	Conference	B	2017	11	31	643
[124]	Conference	C	2015	3	20	406
[125]	Journal	2.024	2018	1	11	199
[126]	Journal	2.602	2018	6	7	244
[127]	Conference	N/A	2017	0	1	790
[128]	Conference	N/A	2016	4	9	464

Table 2.8: Community impact of articles (Part 2/2)

Chapter 3

Context retrieval and Authentication

Obtaining the context is a crucial aspect of context-aware applications, and it is the first step in context-aware computing. Contextual information is used for context-aware security, and without coherent, relevant, and up-to-date data, context-aware security may fail to provide valid results.

Traditional use cases require a limited number of context sources (e.g. sensors), which are made available to interested participants. However, for security usage we need to get contextual information from as many participants as possible, preferably from all participants. This data also needs to be in a unified format, so that the security rules can be reused across devices and applications.

The main goal of this chapter is to design a solution capable of detecting a physical attack on the IoT network or on an IoT device. Attack methods may involve introducing new devices or replacing devices (either to monitor the communication or to provide malicious data), relocating devices or disabling devices. I assume that the attacker is not aware of the network topology, and that she does not know about the counter-measures that are being employed. It is assumed that the attacker does not have any advanced knowledge or training.

A limiting factor for the method is that it requires a large enough network (with at least tens or low hundreds of devices). It requires the devices (or their middleware layer) to have sufficient computation resources to determine their network neighborhood. An additional limitation is that the method is not able to determine small network changes or changes that happen gradually over time.

In this chapter, I present the part of my research that focuses on finding novel contextual data that could be accessible to all IoT devices. It explains the proposed method, describes the algorithm, and then it evaluates the solution both in a real-world scenario and in a simulation.

To demonstrate the value of context usage, I use context as an additional authentication factor. There is no modification of traditional authentication,

only the addition of an extra factor: something I am. Authentication is based purely on contextual information. The contextual information can also be used for context-aware authorization, as described in the following chapter.

In Chapter 3, I will:

- Present an additional authentication factor for usage in the IoT environment.
- Illustrate how to set up the method in a network.
- Discuss how various settings affect the proposed method and how to determine the ideal values.
- Demonstrate the feasibility of the method based on a network with hundreds of unique devices, and provide experimental data allowing better insight into applicability of the method.

The research described in this chapter was initially explored in a conference paper [A.7], followed by a progress report presented in another conference article [A.8], and a final journal article [A.1].

3.1 Proposed method

The idea of the proposed method is based on the regular network context reports provided by every IoT device. Devices retrieve a list of all devices discoverable in the network and send it to the server regularly. Ideally, this information is passed along during every server request. Due to the bandwidth of the network, storage considerations, the computation capabilities of the server, and other limiting factors, information passing can be restricted to a specific reasonable timeframe (e.g., every 15 minutes) to reduce the communication overhead. The server subsequently stores the data for further use, evaluates the received data, and eventually proceeds with further actions. These actions may include an additional authentication request to a suspicious device (which may or may not be the device that triggered the action), a notification to a network administrator, or even a limitation to or removal of network access for a suspicious device. A network context scan is performed on end devices, and the server performs only a context evaluation, which results in great scalability.

The utilization of our approach and its full range of possibilities requires a significant amount of contextual data gathered over extended periods of time, preferably in various distinct physical locations, across multiple different networks,

and mainly with knowledge of security incidents that have occurred. An extensive data set of this kind can be analyzed using standard algorithms based on decision tree induction [151] or on advanced adaptive fuzzy rule-based classification [152]. Once the patterns are recognized, they can be searched for in real-time, and appropriate control mechanisms can be activated as needed. Unfortunately, I do not possess a data set of this kind. Therefore, in this chapter, I describe a method for analyzing the network context of a particular device.

The method utilizes "recurring" devices for analyzing network context. A recurring device is a device that has been in the network for several consecutive days. For example, such a device is typically present in the network at a particular time. The Internet follows the standard OSI networking model [153], possibly with MAC layer protocols adapted for IoT devices [154]. Therefore MAC addresses are used as device identifiers because, by definition, they are unique. In most scenarios, the risk of counterfeiting is not significant, as multiple devices with spoofed MAC addresses would need to be introduced into the network. The possibility of the attacking device changing its MAC address does not affect our method more than any other device with a spoofed MAC address, as this MAC address is treated as one of the addresses on the network. A potential successful attack targeting our method would lead to a higher ratio of false positive authentications, which would not affect the user experience or the overall security (in comparison with not using our method at all as an additional factor).

The recurring device list is created specifically for a given network. A recurring device may be a recurring device in more than one network, but this is rarely the case.

An example of such a situation is a personal device carried by a user; the device is in a network during the day when the user is at work, but is in a different "home" network at night.

3.1.1 Illustration of The Proposed Approach

Recurring devices are determined on the basis of historical values that are stored by the server. If a device appears in the network at the same time over multiple consecutive days, it is marked as a recurring device. Recurring devices are determined from a limited historical timeframe (e.g., the last five days) and, therefore, the set of recurring devices varies from one day to the next. When the process is started, recurring devices cannot be determined, as there is no reference point. A list of recurring devices can be made when the timeframe passes (e.g., after five days). Recurring devices are calculated every day given the historical values. The

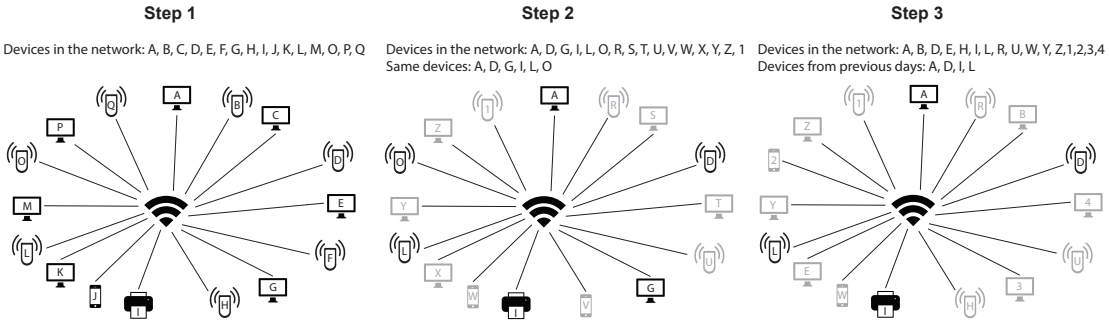


Figure 3.1: Creation of a set of recurring devices in three steps

algorithm takes all devices from the first day and marks them as candidates. Every subsequent day, it removes from the list of candidates devices that are not present during the day. After all the steps are completed, the candidate list forms the final list of recurring devices.

Figure 3.1 illustrates the three-step determination of recurring devices. The steps illustrate a network at the same time over three consecutive days. The sample network consists of 16 various devices – so it can easily be visualized. Real networks often contain hundreds of network elements. During step 1, all devices are considered to be recurring device candidates. In step 2, there are the same six devices as in step 1. These are new recurring device candidates. In the final step, four devices from the candidate list are present. This list forms a new list of recurring devices, and can be used on the following day.

During communication, a device sends the list of all reachable devices in the network. The same rules described above are applied to determine recurring devices; thus, the device does not need to obtain the list for each request. The server compares the sent list with the list of recurring devices (which I call the benchmark) for the given network for a roughly similar time frame. The server also uses the provided list to modify and verify the benchmark for the following days. Our preliminary implementation of the approach can configure the desired recurring device match with the devices in the network. Figure 3.2 illustrates a network with 16 devices and a set of recurring devices consisting of four devices from the previous figure – A, D, I, and L. In this example, the match is 75% (device A is missing). If the threshold is not met (e.g., if there is a 70% match), then the network context of the device is marked as suspicious, and further steps can be taken – the administrator is notified, an additional authentication factor can be invoked, or a more sophisticated network search for malicious devices can be triggered.

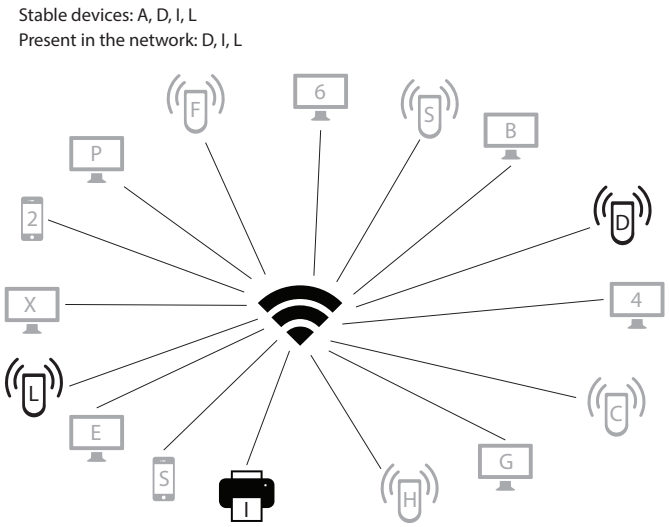


Figure 3.2: Using network context to determine changes in the network

3.1.2 Problem Model and Algorithm

We model the analyzed network as a set of devices $N = \{n_1, n_2, \ldots, n_n\}$, where device n is every network element with an MAC address. Timeframe $t =< t_{start}, t_{end} >, t_{end} - t_{start} < 1\ day$, is a time period during a single day. Times t_{start} and t_{end} can be equal; in this case, timeframe t is a time point, and not a time interval. Age (denoted as age) is the number of consecutive days for which the benchmark is created. I denote the day in which the analysis is performed as d.

Benchmark is $B(t, d, age) = \bigcap_{x=d-age-1}^{d-1} devices(N, t, x)$ where $devices(N, t, x)$ denotes the set of devices present in the network in a randomly selected time from timeframe t during day x.

We define $match(t, d, age) = \frac{|B(t,d,age) \cap N(t,d)|}{|B(t,d,age)|}$ as the proportion between the number of devices from the benchmark and the number of devices in the benchmark present on the network, where t is the timeframe and d is the day in which the analysis is performed.

Then, $threshold$ is the value of $match(t, d, age)$ such that if $threshold > match(t, d, age)$ the authentication check (as introduced in subsection 3.1.1) is passed.

In Algorithm 1, I describe the process for determining the threshold and age. The algorithm accepts the following inputs:

1. The set of all analyzed timeframes T
2. Analyzed network N

Algorithm 1: getAgeAndThreshold(T, N)

Input : Timeframes T, Network N, ε, lim
Output : age_{opt}, $threshold$

$devices(N,t,d)$ = set of present devices in N for t and d, $t \in T$, d is day
$B(t,d,age) = \bigcap_{x=d-age-1}^{d-1} devices(N,t,x)$
$match(t,d,age) = \frac{|B(t,d,age) \cap N(t,d)|}{|B(t,d,age)|}$
$age_{opt} \leftarrow 0$
for $age = 2 \dots lim - 1$ **do**
    $match_{sum} \leftarrow 0$
    $match_{prevSum} \leftarrow 0$
    for *each* $t \in T$ **do**
        $match_{sum} = match_{sum} + match(t, age+1, age)$
        $match_{prevSum} = match_{prevSum} + match(t, age, age-1)$
    end
    $match_{avg} = \frac{match_{sum}}{|T|}$
    $match_{prev} = \frac{match_{prevSum}}{|T|}$
    $\Delta match = match_{avg} - match_{prev}$
    if *(* $\Delta match < \varepsilon$ *)* **then**
        goto 20
    end
    $age_{opt} = age$
end
$threshold \leftarrow 1$
for $d = age_{opt} + 1 \dots lim$ **do**
    for *each* $t \in T$ **do**
        $curMatch = match(t, d, age_{opt})$
        if *(* $curMatch < threshold$ *)* **then**
            $threshold = curMatch$
        end
    end
end
return $age_{opt}, threshold$

3. Constant ε defining when to stop the algorithm
4. Constant lim, which is the number of days for which I run the algorithm

The outputs of the algorithm are:

1. age_{opt}, which denotes the optimal age
2. $threshold$, which denotes the maximal possible threshold for the given lim

The principle of Algorithm 1 is described by the following steps:

1. For network N try iterate ages (2 to lim) over all benchmark periods. Use the last day suitable for the given age and save the average match. Similarly, compute the match for age-1 for the same day. Compare the current match to the match one day shorter, and if the match function starts converging to meet the algorithm's stopping criteria defined by ε, use the previously found age, denoted as age_{opt}. This is represented by lines 4 through 18.
2. For the timespan from $age_{opt} + 1$ to lim, determine $threshold$ such that every $match$ for each of the analyzed timespans is equal or higher than $threshold$. The respective lines of the algorithm are 19 to 27.

3.2 Experimental Verification

To verify the proposed method using a real network, I conducted the case study that will be described in this section. To demonstrate the validity of the proposed approach, I performed: (1) an evaluation, using a real network, and (2) a simulation of the network with various possible events that could happen (e.g., recurring device disappearance or MAC address spoofing). Details are presented in the following subsections.

3.2.1 Real-network evaluation

Initially, I determine relevant timeframes for a benchmark. Then, I determine whether exactly the same time of the day needs to be used for the measurements on various days, or whether an approximate interval can be used. Once I have such values, I proceed to determine a threshold for the percentage of recurring devices in the network based on historical network data.

Five weeks of measurements are performed in the same network, and six control measurements are conducted. I performed the case study on the Baylor University

Age	8:00	12:00	16:00	Avg	Pr. Avg	Morning	Noon	Afternoon	Avg	Pr. Avg
2	65%	62%	57%	61.46%	33.80%	58%	65%	58%	60.19%	29.55%
3	67%	58%	70%	65.09%	52.69%	60%	54%	74%	62.39%	51.57%
4	72%	80%	75%	75.62%	66.35%	46%	62%	73%	60.43%	50.74%
5	93%	81%	76%	83.31%	76.59%	92%	77%	78%	82.21%	80.48%
6	96%	90%	76%	87.49%	78.99%	100%	88%	76%	87.94%	81.56%
7	84%	89%	86%	86.37%	82.13%	73%	67%	83%	74.05%	71.90%
8	100%	83%	88%	90.44%	85.14%	100%	86%	88%	91.07%	88.69%
9	90%	95%	91%	92.13%	91.38%	94%	100%	71%	88.39%	82.04%

Table 3.1: Benchmark age determination with limit 10 (Algorithm 1 lines 5-19)

Wi-Fi network in the Department of Computer Science with hundreds of unique devices. I chose this network for the experiment because it provides a considerable number of devices in which users periodically connect and disconnect (e.g., students' devices) with various schedules and devices that are always present (e.g., printers). I conducted the research during my visit to Baylor University, in cooperation with other researchers. I performed six analyses per day, evaluating the network only on weekdays. Three analyses were conducted at fixed times – 08:00, 12:00, and 16:00 – and three were conducted at random times within specific time intervals representing morning (07:30 - 10:00), noon (11:00 - 13:00), and afternoon (14:00 - 17:00).

First, I aimed to determine how many days are needed for the benchmark. I ran the algorithm limited to 10 days and with epsilon of 5%. Let me explain why I chose these values and what implications this choice carries. I chose 10 days as the limit, as I considered 10 days to be a reasonably large number of days in comparison with the whole dataset. I expected the benchmark age to be lower, leaving us a few days to determine the threshold. Theoretically, one day is enough to determine the threshold, but a single day could lead to some anomalies that would surface as false positives. At the same time, the upper limit still leaves twenty days, which is a reasonable number of days to run the algorithm with the benchmark created. In a real world application, we could extend the limit if the number of days turned out to be too small – e.g. after other parameters have been determined, or after some experience of running the algorithm live against the targeted network for some time. Epsilon of 5% was chosen more or less arbitrarily, as I had no previous experience. The lower the epsilon that we choose, the more days we need to run the initial algorithm to discover the optimal benchmark age

	Devices count	Benchmark size	Recurring devices count	Benchmark match
8:00	272	31	26	84%
12:00	620	31	26	84%
16:00	931	35	31	89%
Morning	309	26	21	81%
Noon	581	29	20	69%
Afternoon	560	30	25	83%

Table 3.2: Day 11 measurement: devices in the network at specific times and in time intervals with a 6-day benchmark age of

and threshold. As in the case of the days, however, if my choice turned out to be wrong, the number of days could be adjusted after observing the results and the algorithm could be rerun.

I ran the algorithm twice – once for the fixed timeframes and once for the intervals. I show combined examples of benchmark matches for ages 2 to 9 in Table 3.1, with different benchmark periods. The table also lists the average matches and the average matches of the previous benchmark age (please note that the previous average value differs from the average value in the row above it, as the two values were calculated for different days). The percentage shown in each line is for the next day after the benchmark was calculated; e.g. for age 2, the table shows the match on day 3. For age 6, the table shows the match on day 7. The gradual increase in the match ratio is caused by the decreasing number of recurring devices.

With the chosen epsilon of 5%, we observe that the optimal age is 6. This is because age 7 shows a difference of only 4.24% (2.15% for the intervals). This is where the algorithm would stop calculating, and further values are shown only for illustration. These findings are consistent across all measurement times, even for those values taken randomly within an interval, as illustrated by the right side of the table.

I illustrate the percentage of recurring devices found for the two benchmark period types for all days in Figure 3.3, where the 12:00 and noon measurements are used. The randomized interval measurements are illustrated in the chart on the right, and they fluctuate significantly more than the measurements taken every day at the same time, which are depicted in the chart on the left. Note that only weekdays are used; i.e., day 6 (11, 16, ...) corresponds to a Monday.

The next question concerns the difference between measurements taken at strictly the same time and measurements taken during the same time interval. Randomized

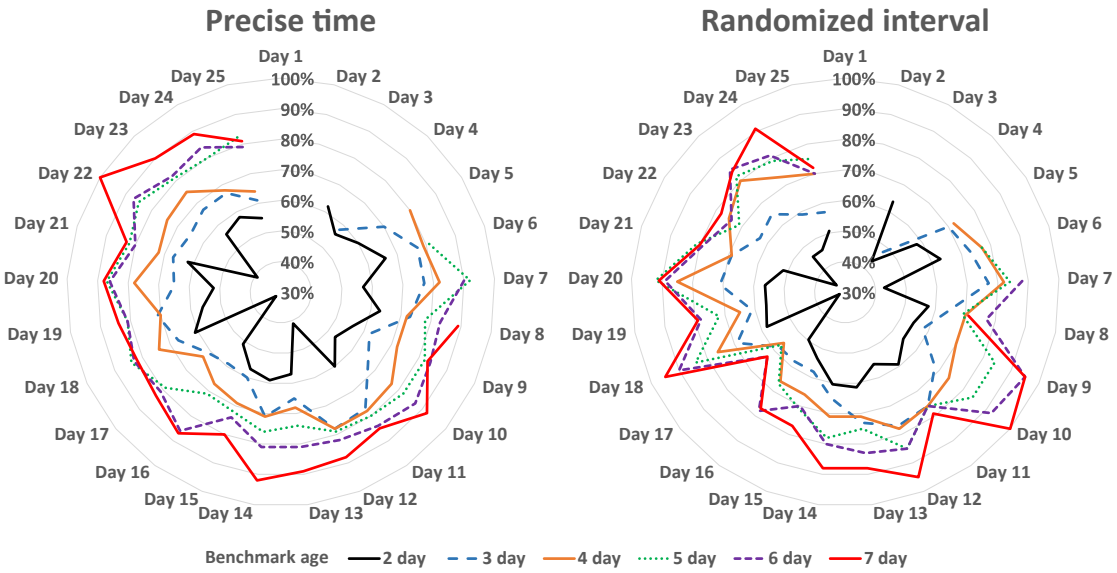


Figure 3.3: Percentage of recurring devices for each day, using different benchmark ages

measurements reduce the possibility of intentionally spoofing the network and providing fictitious MAC addresses to inflate the set of recurring devices. I use day 11 to demonstrate our findings. I choose specifically a 6-day benchmark and illustrate the number of devices in the network, the benchmark size, the recurring device count, and the benchmark match for each time in Table 3.2. For each time or time interval, I use the corresponding times or intervals on the previous days to determine the benchmark. The table shows that there is a decrease in the match percentage between the time interval measurement and the corresponding fixed time measurement ranging from 3% to 15%. In addition, the interval measurements show higher variation between their matches. I chose to continue the case study with fixed time measurements, because they provide higher consistency. The measurements also confirm this higher consistency in Table 3.3, where the recurring devices for a specific time never drop below a 73% match, while the interval measurements can drop as low as a 65% match.

With the benchmark period set using fixed times for the measurement strategy and a benchmark age of 6 days, the last missing piece is the threshold. Algorithm 1 (with inputs of a 10-day limit and epsilon 5%) gave us output of 76% as the maximum threshold. This is the minimum value over the set time measurements up to day 10, inclusive.

	8:00		12:00		16:00		Morning (07:30-10:00)		Noon (11:00-13:00)		Afternoon (14:00-17:00)	
Day	Devices	Recurring devices	Devices	Recurring devices	Devices	Recurring devices	Devices	Recurring devices	Devices	Recurring devices	Devices	Recurring devices
Day 1	343	N/A	769	N/A	729	N/A	568	N/A	599	N/A	568	N/A
Day 2	447	N/A	615	N/A	628	N/A	606	N/A	585	N/A	722	N/A
Day 3	349	N/A	645	N/A	753	N/A	337	N/A	629	N/A	781	N/A
Day 4	365	N/A	546	N/A	521	N/A	389	N/A	715	N/A	534	N/A
Day 5	245	N/A	456	N/A	557	N/A	191	N/A	537	N/A	580	N/A
Day 6	271	N/A	546	N/A	696	N/A	314	N/A	703	N/A	651	N/A
Day 7	429	96%	566	90%	573	76%	449	100%	518	88%	653	76%
Day 8	261	82%	691	83%	715	86%	182	69%	627	73%	705	82%
Day 9	416	100%	656	84%	514	82%	491	100%	578	89%	532	82%
Day 10	252	86%	540	87%	520	81%	280	96%	562	96%	446	76%
Day 11	272	84%	620	84%	931	89%	309	81%	581	69%	560	83%
Day 12	510	79%	772	82%	701	86%	198	79%	611	92%	577	83%
Day 13	316	79%	830	81%	981	82%	317	86%	765	83%	632	76%
Day 14	538	81%	689	81%	728	86%	162	79%	600	87%	746	88%
Day 15	320	77%	726	74%	709	78%	175	76%	677	73%	451	83%
Day 16	436	88%	811	86%	1166	76%	352	86%	600	84%	877	82%
Day 17	571	85%	765	81%	1004	79%	626	79%	720	66%	563	76%
Day 18	360	83%	1304	83%	1247	80%	351	88%	1206	83%	703	76%
Day 19	611	81%	938	81%	823	79%	549	81%	885	85%	664	75%
Day 20	304	82%	676	86%	739	81%	430	79%	966	89%	683	65%
Day 21	405	87%	1168	80%	928	73%	480	75%	1180	82%	829	77%
Day 22	564	79%	968	87%	723	77%	689	79%	994	81%	664	95%
Day 23	400	91%	818	82%	1147	78%	496	91%	1252	88%	986	72%
Day 24	602	92%	687	84%	708	75%	581	77%	740	81%	603	79%
Day 25	309	92%	576	79%	548	84%	378	78%	519	74%	386	75%

Table 3.3: Day overview of the 6-day benchmark: number of devices on the network and recurring devices for each measurement during every day

The output of the algorithm can be tweaked manually. Lowering the threshold effectively decreases the number of false negative authentications, but increases the risk of false positive authentication. Increasing the threshold has the opposite effect. For this evaluation I chose to lower the threshold to 70%, as I wanted to give us some safety margin as we determined the threshold only across four days (days 7 through 10) out of the 25 days in the evaluation.

Table 3.3 presents the network evaluation for each day and time or time interval during our study, using the six-day benchmark. Day 1 in the table corresponds to the Monday of the first week of the case study, with days 6, 11, 16, and 21 also being Mondays. The number of devices in the network varied from 250 to over 1000, with Fridays and some Monday periods being the days with the fewest devices, and mornings were the time with the lowest number of active devices. However, the percentage of recurring devices was reasonably consistent, never dropping below 73% across all days and times (for the fixed intervals).

Four control measurements were conducted to verify the ability to detect changes in the context. The measurements were compared with the six-day benchmark from previous days, based on the base network at Baylor University. All measurements were taken at 12:00 to allow an exact match with the benchmark, which should provide the greatest similarity. One measurement was retaken in an entirely distinct environment, but with some devices from the base network regularly appearing there. The place that was chosen is an apartment complex in which a considerable number of Baylor students were living. However, there were zero matches, most likely because the network is segmented into smaller subnetworks that I could not scan. Two other measurements were taken in a similar environment, where it can be assumed that similar devices were in use. The chosen places were locations within Baylor University but outside of the base network, with many devices flowing between the networks. They provided a match of 5% and 0%, confirming that places with high fluctuation in the same devices are not matched. For the last control measurement, I chose our base network but during the weekend, in order to check whether I could also detect a change in the main network context. There were fewer than one-fifth of the usual number of devices during the analysis, and the match was only 29%. All of the control measurements produced values significantly lower than the threshold of 70% set in the previous paragraph. An overview of the results is presented in Table 3.4, including the benchmark size and the number of recurring devices found for each day of the measurements. I can also reveal that a fifth control measurement was carried out at a grocery store on day six. This measurement have not been included in the table, as a six-day benchmark

Place	Devices	Bench.	Recurring devices	Percentage	Day
Apartment	8	31	0	0%	9
Dinning hall	1095	30	0	0%	7
Commons	42	31	2	6%	9
Saturday	118	31	11	35%	10

Table 3.4: Control Measurements

cannot be used on this day. I have used a five-day benchmark to illustrate how the algorithm works when presented with an unrelated large network. The match was 3% in the network of 595 devices and with a benchmark size of 31 matching single devices.

This verification shows that I can detect anomalies in the network. It provides data that illustrate this ability in a network with hundreds of users active at the same time. It demonstrates how the 6-day benchmark was chosen as the ideal benchmark age, it explains when measurements taken at random times in an interval are better for analyzing networks than measurements taken at the same fixed times, and it describes the process for determining the optimal threshold value for this particular scenario. The control measurements demonstrate the ability to detect an unfamiliar context in numerous networks with different characteristics or at different times in the base network.

We evaluated the performance of this method in a network. ARP scans were used to determine which devices were available. With our method, therefore, each device receives an ARP request. I evaluated the performance in a network with 254 addresses. With six devices scanning simultaneously, the scanning increases the latency of the network (measured between two other devices) from 2 ms to between 13 and 20 ms. A full scan of the network with 254 addresses takes slightly under 3 seconds.

3.2.2 Simulation

In this section, I simulate the behavior of the network in potential situations that did not occur during our five-week real-world evaluation, but that are of significant concern. For the simulation, I used the real-world network measurements, and I adjusted them to particular scenarios by removing or adding the devices into the measured data. I explored cases that could potentially lead either to a false negative classification or to a false positive classification. For the initial simulations, I chose day 11, time 12:00, from our measurements. For scenarios that could lead

Simulation	Simulation match	Original match	Threshold	Simulation classification
Failure same day	80.65%	83.87%	70%	True positive
Failure day before	83.33%	83.87%	70%	True positive
Adverse device	84.38%	83.87%	70%	True positive
Attack with 15 devs.	56.52%	35.48%	70%	True negative
Spoof attack	38.71%	35.48%	70%	True negative

Table 3.5: Simulation with threshold 70%

to false positives, I chose the Saturday following day 10 and again 12:00 time, as I already had data for this day and time in the control measurements. The results are summarized in Table 3.5, and are described below.

The first simulated case is the failure of a stable device. This can be divided into two events. The device can fail before the measurement is taken, which means that it is not included in the current benchmark. Alternatively, it can fail on the same day, and it is therefore included in the benchmark. Failure on the same day reduces the number of recurring devices from 26 to 25, and therefore the match decreases from 83.87% to 80.65%, which is well above the threshold. Failure of the device in the preceding days reduces the benchmark size from 31 to 30 and also reduces the number of recurring devices to 25, which leads to an 83.33% match. This is again above the threshold that I had set. The only measurement where failure on the same day would lead to a false negative is day 21 at 16:00, as it would reduce the match to 69.23% (failure on the day before would only reduce the match to 72.00%).

The second scenario is when an adversary is present on the network from the beginning. This leads to an increase in the number of devices in the benchmark and in the number of stable devices found. In our simulation, this increases the match from 83.87% to 84.38%. The benchmark increases by one to 32, and the number of recurring devices increases to 27.

The third case simulates a broad attack on the network, with 15 malicious devices present on the network. This increases the number of devices in the network to 133, increases the benchmark size to 46, and the increases number of recurring devices from 11 to 26. This leads to a match of 56.52%, while the match without the attack was 35.48%. Given our network and the specific day, an attack would need to consist of 22 devices to reach our threshold and thus lead to a false positive.

The fourth case simulates an attack where the malicious devices spoof the MAC

address to one of the benchmark addresses not present in the network. The presence of the device increases the number of recurring devices to 12 and increases the match from 35.48% to 38.71%. Eleven devices in a coordinated attack would be needed to lead to a false positive. I therefore identify this as the weakest part of our method, as 11 devices is a considerably smaller number than the 22 devices from the previous scenario. In addition, these 11 devices may be present on the network only during the attack, and therefore they are more likely to remain unnoticed by the network administrators.

3.3 Threats to validity

The experimental verification presented in this study is based on an experiment with one selected network and a simulation of various situations that can occur during network operation. This can be considered as a threat to validity. Although the network used in the experiment was sufficiently extensive, it cannot be assumed that other large networks will have a similar topology and similar characteristics.

However, this issue can be mitigated by adjusting the parameters of the proposed methods. In networks where devices do not fluctuate as much as they do in university networks, or in networks where there are many newcomers or irregularities, the values for the threshold, the optimal benchmark size, or the measurement times may vary significantly.

Another concern may be raised regarding the fact that I used MAC addresses as a device identifier in the proposed method and in the experiments. Generally, MAC addresses are easy to spoof, and if attackers determine the set of recurring devices, they can spoof them in the network, which would lead to a false positive result. To mitigate this issue, alternative device identification can be used. With an alternative identification of a device, the principle of the method does not change.

3.4 Discussion

The *threshold* given by Algorithm 1 can be further adjusted to modify the behavior of the method. Lowering the threshold reduces the number of false positives and increases the number of false negatives. Increasing the threshold has the opposite effect. Each percentage point that I remove from the threshold increases the number of devices that are allowed to pass the authentication. For example, this could provide a safety margin while reducing the accuracy of the method.

The *number of benchmark days* determines the adaptability to network changes.

Networks with a higher number of fluctuating devices will have a lower value than networks where the same devices are present all the time. These values can be modified to suit a particular network.

The *definition of timeframes* affects the behavior characteristics of the method. Basically, the longer the timeframe, the more the devices fluctuate. While this can offer some extra protection against MAC spoofing, it lowers the threshold and can therefore lead to false positive authentications.

The proposed method is dependent on the size of the network. At least tens of overall devices are needed to provide meaningful results, and hundreds of devices are needed to achieve a consistent output.

The approach described here provides an *additional authentication factor*, and it is therefore not sufficient as a standalone authentication method.

In addition, the method detects changes not in the behavior of the devices themselves but in the network neighborhood of the devices. Therefore, the proposed method cannot detect remote device hijacking.

3.4.1 Alternative approaches

The solution presented in this chapter relies on comparing a benchmark with the current state of the network. I have presented a single approach for building a benchmark and determining the match threshold. There are other options that might be explored. One option is to use same days of the week for benchmark creation (e.g. not the last six days but the last six Mondays). This approach would require a much longer history of the network to construct the benchmark, but we could reuse the approach described in this chapter.

Another option is to use a fixed number of random days from a fixed set of last days. For example, we would use six days out of the last ten days to determine the threshold and later to construct the benchmark. In order to make this option work, we would need to come up with an approach to determine the appropriate number of days to select and the length of the period (in days) from which the days will be selected. Once this is known, we would face another choice – how and when to select the days from the period to create the benchmark. One approach would be to select the days when the initial parameters are determined, and to use these parameters when each benchmark is established. Another option is to randomly select days from the set when each benchmark is established. With this approach, the threshold would need to be calculated based on all possible combinations of the days in the set.

I performed a limited experiment to provide some initial insight into the method.

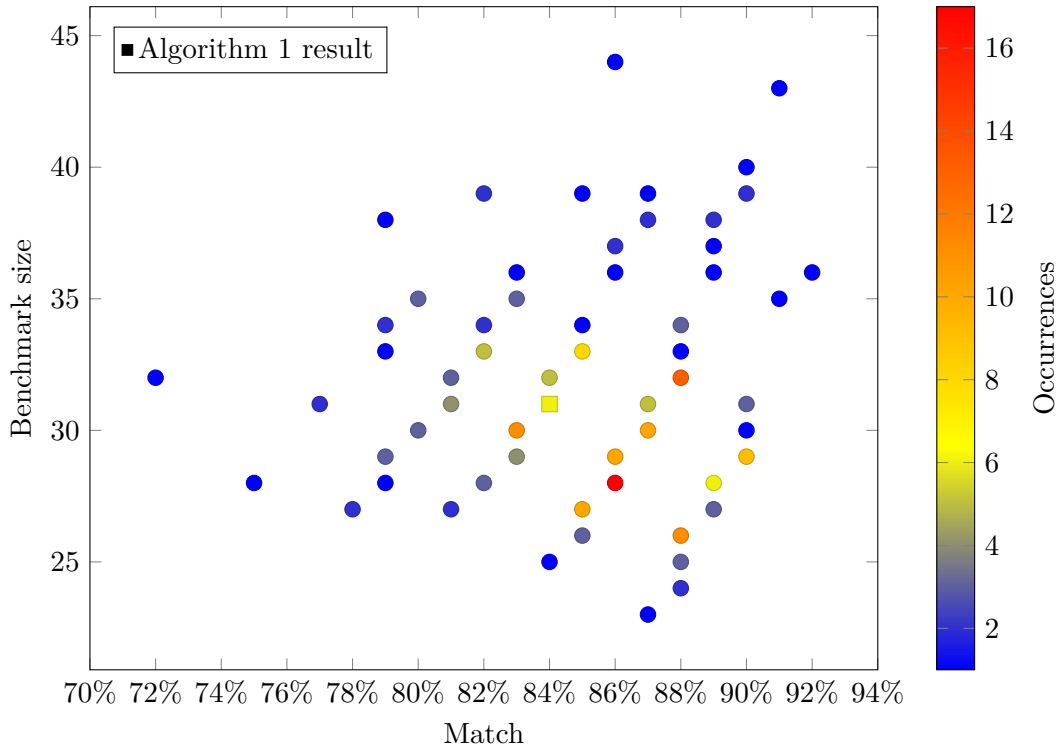


Figure 3.4: Six of ten days benchmark result on day 11 with 12:00 time.

I tried constructing a benchmark with six out of ten days using 12:00 time, and applying it on day 11. Six days were chosen to allow me to compare this algorithm with the previous algorithm as a baseline. This also means that one of the possible combinations will correspond to results that were shown before when using a period of days 5 to 10, inclusive. I selected ten as the number of days to choose from because it is the maximum allowing comparison on day 11. This leads to 210 possible combinations, which should provide a good overview, and can be displayed or verified manually.

I have performed a limited experiment to provide some initial insight into the method. I have tried constructing benchmark with six of ten days and use it on the day 11. Six days were chosen to allow me to compare it with the previous algorithm as a baseline. Also, it means that one of the possible combinations will correspond to the one described above (using days 5 to 10, inclusive). As the number of the days to choose from I chose ten. This leads to 210 possible combinations, which should give good overview while being possible to easily display or manually verify.

Figure 3.4 shows the results that I obtained. The chart shows the matches and the respective benchmark sizes on day 11. Color coding is used to mark the numbers of occurrences. The square symbol indicates the match and the benchmark size

corresponding to day 11 at 12:00 shown in Table 3.2. The minimum match is 72%, with a benchmark size of 32. Most the matches were between 82% and 90%, with a benchmark size between 26 and 35.

This fairly limited experiment proves the feasibility and the potential of this alternative method. However, further research is needed to determine how many days should be chosen from how big a set; how the days for the benchmark should be selected; whether only one benchmark, or multiple benchmarks, or all the benchmarks, should be chosen; and what benefits and what drawbacks will arise.

3.5 Summary

The proposed solution allows the context for all types of IoT devices to be determined. The case study proves its feasibility and usability. The solution makes decisions based on changes in the context in the network around devices, and it can therefore detect suspicious or even malicious behavior. It is a simple mechanism in terms of device resources, and it can be deployed on any IoT device capable of communication over TCP/IP, allowing system operators to inspect the network and, if needed, to take appropriate actions to resolve an issue. The context information can be used as an additional security factor in conjunction with existing security architectures.

The real-world experiments have demonstrated the feasibility of the approach in a network with a significant number of devices. The results indicate that the concept can provide valid results and can increase the security of the devices and of the entire network. This sort of approach is particularly suitable for secure locations, such as laboratories, energy sources, or military bases, where the aim is to limit the access of external devices. However, the solution may not be ideal for locations where devices have a high churn rate, e.g. for shopping centers.

This method alone cannot be used for device authentication, but it can serve as an additional factor during the authentication process. With an unfamiliar or suspicious network context, actions can be taken such as further authentication or time- or resource-intensive network analysis. An example of a suspicious network is one involving the sudden appearance of a significant number of unknown devices.

Chapter 4

Context-aware authorization

While having contextual information is a crucial prerequisite for context-aware security, the sole fact of having access to context does not make an application context-aware. Currently, the most prevalent authorization architecture, RBAC, does not support context-awareness. RBAC is a pure abstraction in the form of roles over permission assignment to the users. The situation with MAC and DAC is similar – only permissions are assigned to the user, without any contextual conditions.

Application owners and operators, as well as software developers, recognize the added value of context-aware authorization. However, none of the numerous proposals for context-aware authorization has become widely used [50], [54], [55], [57]. Context-aware authorization has not come into more frequent use because it is considered too complicated for practical use, or too innovative. It requires the whole authorization system to be redesigned, as it is challenging, from the engineering perspective and also from the security auditing perspective, to incorporate context-aware authorization into an existing solution.

The main goal is to increase the security of the data in the systems. My aim is to add an extra level of data protection that will make it harder for an attacker to retrieve, change or remove confidential data. It is anticipated that the attacker will use an existing application and will gain unauthorized access to the credentials of some user privileged enough to see (or edit or remove) data. The attacker is not expected to be a highly-skilled person or organization.

The solution presented in this chapter is aimed at application developers who have an existing access control system in place, preferably RBAC. One of the main requirements is therefore ease of adoption and potential gradual introduction into all parts of the system. In this chapter, I present my research on extending RBAC with context-aware elements. The extension is based on user security levels, which quantify the user's context. To access resources, the user is required to possess

a particular security level, in addition to her usual access rights. This proposal allows an existing RBAC solution and existing RBAC architectures to be extended with context-aware elements.

In this chapter, I will:

- Extend traditional RBAC by adding a context-aware element.
- Implement the proposal into an open-source Identity Management (IDM) and security management solution.
- Demonstrate the approach on in a case study, and make comparisons with traditional approaches.

The research reported on in this chapter has been presented at two conferences. The initial idea was presented in a submission [A.11] to RACS'15, and an extended version [A.10] was presented at SAC'16. The second of these papers has been well received and has been quite frequently cited (32 citations, ten of them in articles in impacted journals). The ideas expressed in this chapter have therefore already had an impact on the scientific community.

4.1 Proposed Solution

Authorization policies in organizations tend to be very consistent, and change only slightly over time, if at all. Most organizations do not want, or even do not need, to apply radical changes. Context-aware authorization must offer a way for organizations to evolve their current security without resorting to extreme and radical change. New authorization rules can be built on to existing and well-proven solutions, and the solution will continue to be well accessible for people who are familiar with current solutions.

I propose the creation of a security level that is based on context. This serves as an addition to the traditional roles in RBAC. The level can be understood as a quantification of the trustworthiness of the user, and it is dynamically tied to the user and to the user's context. This security level creates a second authorization constraint in addition to traditional security permission. The resources in an application can therefore subsequently have two different kinds of authorization rules – classical policies tied with roles, and a security level. The two approaches are independent and are complementary to each other. It is possible to have one approach without the other, but the consequences of using solely context-aware security without other security policies are difficult to predict.

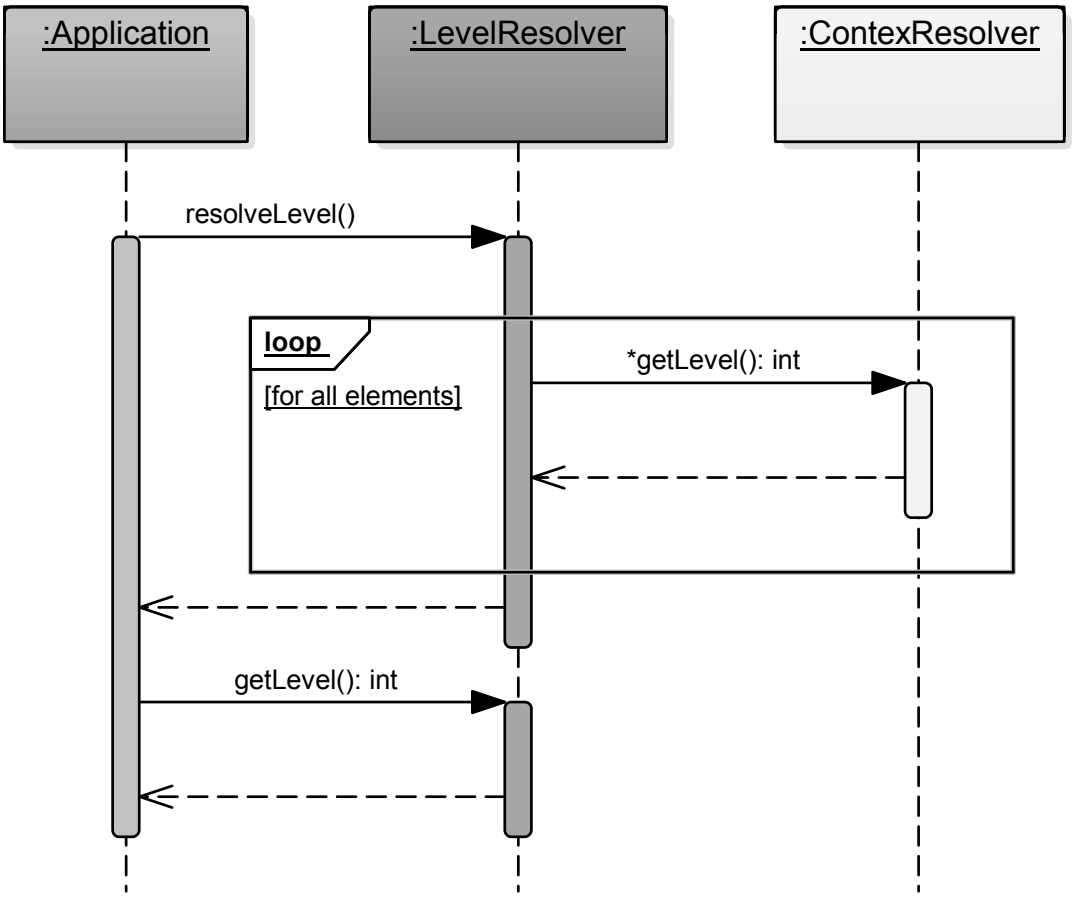


Figure 4.1: Process of determining the security level

As the user's context and the application changes, the level needs to reflect the dynamic nature of the context. There are several moments when the level can be calculated. The first moment is to calculate the level while the user's account is being created. However, this does not reflect the dynamic nature of the context, and is therefore unsuitable for our needs. The opposite extreme is to determine the level with each request for authorization. This would reflect the changing context most reliably, but it is very demanding in terms of computational resources, and it is also time-consuming, as the context check may not be trivial. The best compromise seems to be to determine the level during the user's login into the application. Figure 4.1 shows a system sequence diagram in which the user level is determined and is stored for further use. This approach reduces the number of context checks by several orders and, at the same time, it provides a very accurate snapshot of the user's context. In cases when the context changes rapidly, the user can perform relogin, or the application can even enforce a new level calculation

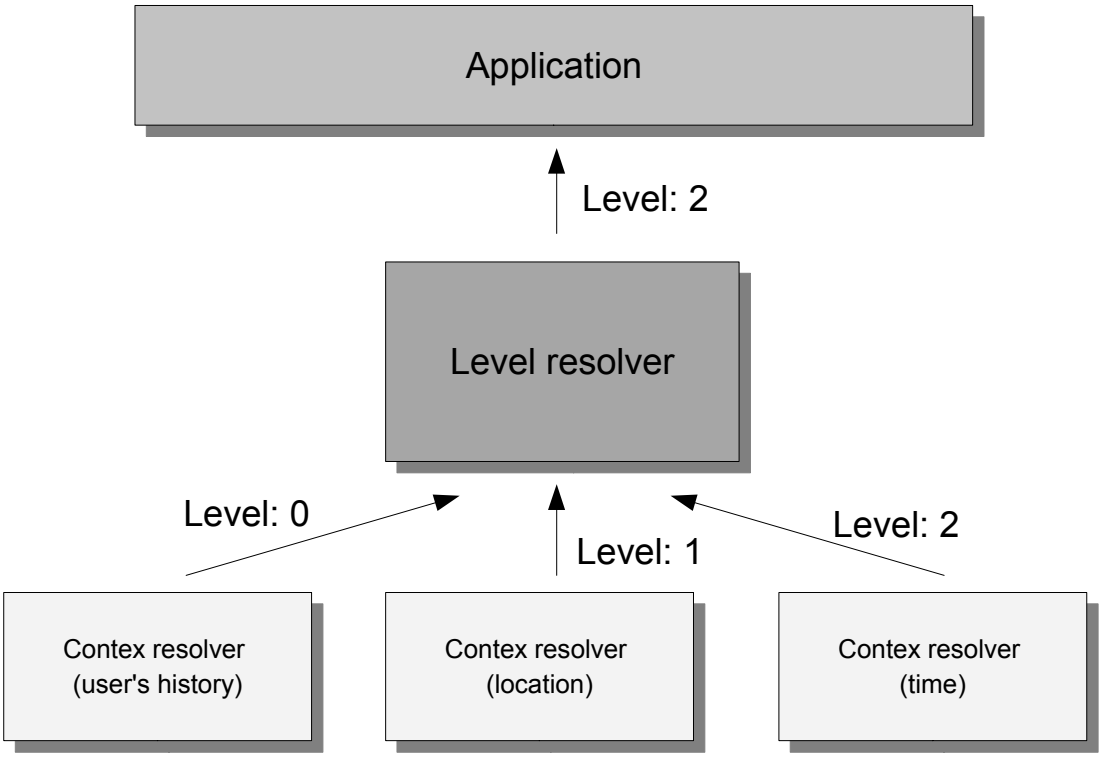


Figure 4.2: Level determination from given context resolvers

manually.

Context resolvers achieve level resolution as shown in Figure 4.2. Each resolver takes responsibility for checking one particular part of the context. For example, one resolver determines the network context from which the user comes. Another resolver checks the time of the day, and so on. Each resolver returns the level that it grants to the user. As the security resolver is written within the application, it has access to the user's information (e.g., her request, information about her stored in a database), and it can also use information about the application (e.g., number of requests, number of users).

The security resolver can even consider the machine that the application is running on (e.g., the load of the machine, resource usage, the location of the server). The final level is not set in the resolver; the resolver itself decides which level to grant, based on its own knowledge and logic. After every resolver has performed its inner logic and has determined the level on its own, the highest level is used as the final user security level.

The level representation itself is very abstract. All that is necessary is for the level to be comparable with other levels. Then it is possible to determine whether

the level is higher or lower than the required level, and to determine the highest level. It is therefore not crucial whether a number, a string or even some more complex structure represents the level. This leaves much space for customization for a given application.

```
@AllowedRoles('admin','manager')
@RequiresLevel(3)
public Resource getResource(int Id) { ... }
```

Listing 4.1: Sample use of security levels for resources

Listing 4.1 shows the usage of levels in the code. The definition of allowed roles to access the method can be seen, as is common in RBAC. The required security level that the user needs to possess in order to be able to invoke the method can also be seen. The proposed solution has many advantages. The most important advantages are that it is:

- Lightweight – it does not require any complex structures in the application, nor does it not consume significant system resources.
- Easy-to-use – it just requires the addition of another type of constraint on resources that need to possess context-aware authorization.
- Voluntary – if an administrator wants to use plain RBAC she can, and she can add level restrictions only for selected resources.
- Scalable – there is no predefined set of levels, nor is there a limit to the number of application levels.
- Universal – the solution can be modified and used with other authorization architectures, not only with RBAC.

However, the solution has some limitations, which need to be worked on further. The most significant limitations are:

- It is hard to determine the exact context – it can sometimes happen that a resource should be accessible just from a given context. For example, some resources are accessible only during the day, while others are accessible just at night. Such a situation cannot be handled with the proposed solution.
- The levels are linear – the structure of the levels is strictly linear, and it is therefore impossible to build some tree or even more complex structure of levels. There are often multiple context rules, which are granted different sets

of rights. The levels cannot, for example, model a geographical situation when users from one state have certain rights, but people in different locations in the state have different specialized rights.

A level resolver can be used together with the network context described in section Chapter 3. Listing 4.2 shows an example of a resolver that uses `NetworkContextService` to determine a perceptual match for a given number of previous days. If the match is above 50%, it returns level 2, if not, zero is returned.

```
public class NetworkContextResolver implements ContextResolver{
  ...

  public int resolveLevel(){
      int match = this.networkContextService.getMatchByDays(3);
      if(match) >= 50){
          return 2;
      }
      return 0;
  }
}
```

Listing 4.2: Sample use of security levels for resources

4.2 Experimental verification

The solution described above was implemented into the open-source PicketLink project [155], and it is a part of PicketLink's released codebase. PicketLink is an identity management and security framework focused on compatibility with Java EE specifications. During my doctoral studies, the PicketLink project merged with the KeyCloak project[156].

To demonstrate the value of my approach, I created two prototypes of a simple e-shop: the first prototype uses the proposed solution, and the second prototype relies on traditional security methods. Then I compared the implementations. I will point out the differences and also the increased effectiveness of my proposal. Both variations of the application have been developed using Java EE 7 [157].

The security functionality of the two approaches is the same from the perspective of the user or administrator, and contains multiple actions and different authorization rules. Users without any form of authentication can browse the items in this shop and can add them to a cart. Users who have logged in using their login

User's status	Actions	Obtained
none	Browse e-shop	default
logged in	View order history	username/pwd
verified	Pay for purchase Change delivery adddress Set trusted IP	SMS code verification Access from set IP

Table 4.1: User's status and allowed actions

name and password can view their order history and their delivery address. Finally, there is a third level of authentication of the user called "verified user". This status allows the user to change her delivery address and to pay for the purchases. This security level can be obtained by additional authentication performed in one of two ways. The first option is to use a specially generated verification code delivered to the phone by a text message. A second option is that the system allows a user to set a trusted IP address (it can be set only if the user has already been verified). When the user logs in from that IP address, he/she is automatically considered verified.

The application is very simplified and contains only a small number of actions (represented by secured service layer methods). Table 4.1 summarizes the actions for each user status, and also shows how the security status is obtained. It is evident that the authorization rights are simple for the use of this application discussed here; however, the rights will most likely be very complicated for real applications.

```
@HasRole('customer')
public void makeOrder(Order o) throws NotTrustedUserException {
  if(!ipCheck.isIpTrusted()&&!smsCheck.isSmsVerified()){
    throw new NotTrustedUserException();
  }
  ...
}
```

Listing 4.3: Method secured traditional way

In implementations without levels, every secured method needs a code for determining a user's context. As Listing 4.3 shows, this adds some lines of unrelated code into the methods, and also a new declaration of thrown exception. The code exhibits obvious concern tangling [158], represented by classes `IpCheck` and `SmsCheck`.

Implementation of the same logic using the proposed levels is displayed in

Listing 4.4. It is clear that the method using security levels is significantly shorter and has no unrelated code inside. Concern separation [158] increases the cohesion [159] of the method and at the same time reduces coupling [159]. The class `IpCheck` has been changed to a level resolver, which reduces the dependencies, as all the resolvers are invoked automatically during login. The class `SmsCheck` has been deleted completely because the framework allows the level to be set up in the authenticator, as is shown in Listing 4.5. Listing 4.4 demonstrates that the approach with levels adds just a single line with annotation to the code of secured methods. In addition, it keeps the code for determining the level in a separate package from the application's business logic. All of this contributes to faster development once the levels are set up, and also to easier maintenance and testing of the code. If levels are not used, there needs to be a condition for each contextual check inside the given method. This unnecessarily increases the complexity and reduces the readability of the code. Even if the authorization rules were extracted to another class, it would add one more dependence for the given class. The proposed solution can also reduce the number of total classes in the application, because some levels are determined automatically by annotations (e.g., over authenticators).

```
@HasRole('customer')
@RequiresLevel('2')
public void makeOrder(Order o){
  ...
}
```

Listing 4.4: Method secured with levels

Implementing the same logic using proposed levels is displayed in Listing 4.4. It is clear that the method using security levels is significantly shorter and does not have any unrelated code inside. Concern separation [158] increases cohesion [159] of method and at the same time reduces coupling [159]. The class `IpCheck` has been changed to a level resolver, which reduces dependencies as all the resolvers are invoked automatically during login. The class `SmsCheck` was deleted completely because the framework allows setting up the level in authenticator as is shown in Listing 4.5. The Listing 4.4 demonstrates that the approach with levels adds to code of secured methods just a single line with annotation. Besides, it keeps the code for determining level separated from the application's business logic in a separate package. All of this contributes to faster development once the levels are set up as well as easier maintenance and testing of the code. Without using levels, there needs to be a condition for every contextual check inside the given method. Therefore, the complexity of the code is unnecessarily increased, and

readability decreased. Even if the authorization rules were extracted to another class, it would add one more dependence for the given class. The proposed solution can also decrease the number of total classes in application because some levels are determined automatically by annotations (e.g., over authenticators).

```
@SecurityLevel("2")
public class SmsAuthenticator extends
BaseAuthenticator {
  ...
}
```

Listing 4.5: Authenticator for SMS verification

In the example given here, the implementation with levels removes three lines of codes and an exception declaration, while adding one annotation, in one half of the secured methods. It also deletes one class (while adding one annotation to the authenticator). The second class is changed, and there are no dependencies to it. It is very likely that, with more complicated applications, the benefits will be even more significant. The result of the case study can be summarized as follows: better re-use, lower coupling, higher cohesion, less code (about three lines of code per rule usage, and about ten lines per rule declaration). There can be significant code savings in large projects. For example, a project with 100 authorization rules, each used 300 times, saves almost 2000 lines of code.

4.3 Threats to validity

The research results have been validated only in a single case study of limited size. The results are part of an open-source library, but it is unclear whether they have ever been deployed in production usage.

Having the levels linear can be a limitation for the production usage of the code. In the time since this research was published, other promising methods for context-aware security have appeared. For example, ABAC might provide similar outcomes with greater flexibility.

4.4 Summary

My study presents a convenient way to enhance RBAC architecture with a context-aware element. The context-aware architecture aspect is represented by the security level, which is a linear abstraction of trust based on the user's context. To access a

resource in the application, the user must possess not only the required role but also the required security level (or a higher security level). This solution retains the advantages of the RBAC architecture while enhancing it with context awareness. Though no research has been carried out to support this claim, the approach seems to be easily portable to various other security architectures.

The theoretical results of this research have led to an open-source contribution that has validated our approach from an engineering perspective and has also enabled us to implement a high-speed case study. The case study has demonstrated that our approach is feasible and offers significant and apparent benefits in comparison with plain RBAC with manually-added contextual functionality.

Chapter 5

Security rules sharing

The IoT is built on the idea that multiple devices cooperate together to reach a common goal. In an IoT network an individual device may be cheap, single-focused and expandable, but the value of the whole ecosystem increases dramatically when there is good coordination with other devices. Therefore, an individual device needs to be able to trust other devices – to communicate safely with them, to provide reliable information, and ultimately to deliver value for the user. Confidence must be established not only among devices but also between users and devices.

However, device management and the creation of reliable and secure environment is a major open issue in IoT [160]. IoT can be divided into three layers – perception, transportation, and application. Device identity management must be implemented at least on the application layer. This layer is responsible for all communication with the end user and for a significant part of the communication with devices, because the layer gathers all relevant data for the user. However, implementing identity management on other levels too can contribute additional benefits.

This chapter presents my research conducted on trust and confidentiality in IoT. I propose a framework for device authentication and essential identity management. The framework consists of a centralized identity store, and it uses already existing security standards and technologies. The centralized solution allows a response that is fast enough to prevent any further damage in the case of an attack targeting devices [161], while at the same time the re-use of existing technologies allows smoother and faster adoption.

To illustrate the need for a solution of this type, let us imagine the following situations with a smart car. Initially, there is a car equipped with a location sensor and a connection to the Internet. This vehicle could provide its location on request. In the base configuration, the location could be used only in emergencies. However, later we may want to change the settings. For example, the car operator may decide to participate in some form of smart transportation. Alternatively,

an insurance company may offer a car owner a lower rate based on small annual mileage. Having a central identity store, in this case, would make everything easier. The car operator would allow devices with the role of an "insurance locator" or "London smart traffic" to communicate with the vehicle.

The goal of this chapter is to develop a proof of concept for an authentication and identity management solution that is usable in the IoT environment. The critical requirement is to support authentication of applications, IoT devices and also users. Another critical concern is the possibility to adopt the solution quickly. It should therefore utilize communication on a TCP/IP layer (preferably using HTTPS), and it must be compatible with existing technologies. The solution will be verified and tested in a centralized environment with a limited number of devices In this stage, the administrator may need to carry out manual work when setting up and operating the solution.

My research presented in this chapter was based on three tasks:

- Describe centralized IoT authentication and the IDM system.
- Implement the proposal in a case study, and verify the results.
- Evaluate the performance overhead of the proposal.

The research presented in this chapter was initially published as a conference paper [A.6], and was later extended into a journal article [A.5]. Both papers have received decent recognition from the scientific community and have been cited in impact factor journals and elsewhere.

Other journal articles [75], [90], [96], which have presented very similar results, are by authors that I have never had any interaction with. The reports on their research were published a few months, or longer, after my conference paper. This suggests that we came to the same results independently. Obtaining multiple identical results from multiple separate research efforts serves to validate the results, and can be used as evidence for the validity of the results.

5.1 Proposed solution

The research led to a central identity store solution, which would keep a record for each device connected to the network. The central element contains unique identifiers for devices, together with their credentials, and it also supports RBAC by storing the roles internally. All machines and applications in the network can use these roles for their authorization rules. The trusted central identity provider

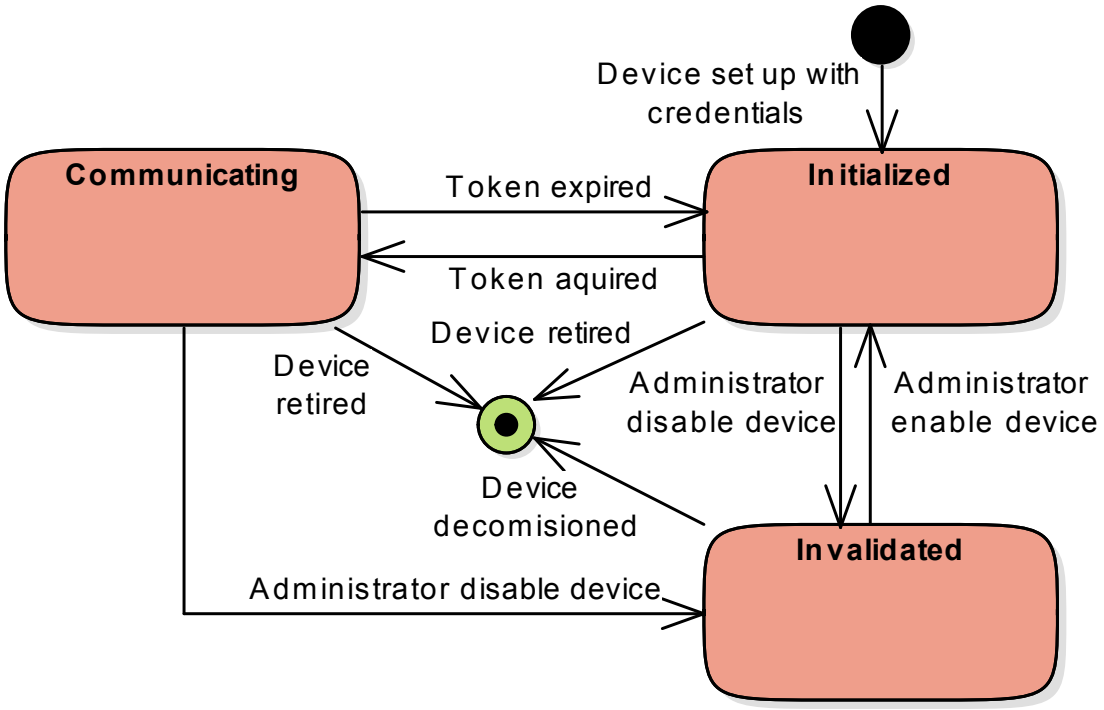


Figure 5.1: Diagram of possible states of the device

creates an environment in which both participants can verify the other partner's identity, and they can also determine whether the partner is allowed to perform a given action.

Using the central identity element in IoT promotes a trusted environment. Devices do not deal with machine-to-machine trust; it is enough to establish confidence in the identity store. Whenever there is a suspicion about a hostile takeover of any device, the device can be disabled with a single action. This action ensures immediate propagation through the whole network. This kind of approach also applies to less severe situations, such as a device malfunction that results in transmitting incorrect data. However, using any central element in a network architecture poses known security threats, for example, a Denial-of-Service attack. Sufficient protection is therefore needed.

Let us consider two communication roles (not only) in IoT. The first role, called the provider, exposes services to others. In the proposed method, the provider must register at the identity store as an identity client when the provider decides that services it provides are confidential. The second role, the consumer, uses the provider's functionality and initiates the communication.

The consumer needs to have a registered identity in the central identity store. To

initiate communication with a secured service, the consumer authenticates using an identity store and retrieves a token representing her identity (and possibly other information, e.g. roles), signed by the identity store. Later, the consumer uses the token for communication with the service provider, which validates the token using the signature that is provided. This enables authentication of the consumer with a trusted element, and it therefore prevents misuse by malicious service providers.

The identity store does not need to serve solely as an authentication service; it also may provide additional functionality. For example, it can provide additional data that will be used for authorization. I focus on roles for RBAC but, generally, any information can be provided by the central store, e.g., attributes for ABAC. Then the service provider can specify the roles required for a given action. When the consumer tries to use the service, her roles are verified with the identity store. This also means that the roles are global for the specific IoT environment, which reduces efforts related to administration and the number of repeating configurations across all systems.

Additional information about the device is stored in the token returned by the central store. The token is signed, and the receiving application verifies the token using the central identity. This is especially useful when communicating with stateless services, as the request contains all required information about the service caller for authentication and authorization.

Figure 5.2 demonstrates the authentication and the authorization workflow of the devices. The following steps describe the procedure:

- The administrator creates an account for a device and sets up its roles.
- The device is configured with credentials provided by the administrator, and requests a token from the store.
- For any confidential communication, the device uses the token to authenticate itself.
- The application or device receiving the communication verifies the identity and roles from the token at the central store.
- The administrator can disable or remove a device from the identity store and therefore effectively disable it for any cooperation.

Configuring a device in such a network does not require significant effort. First, the device is provided with credentials when it is set up. Before it initiates communication with its partner, it requests a token with credentials. The credentials

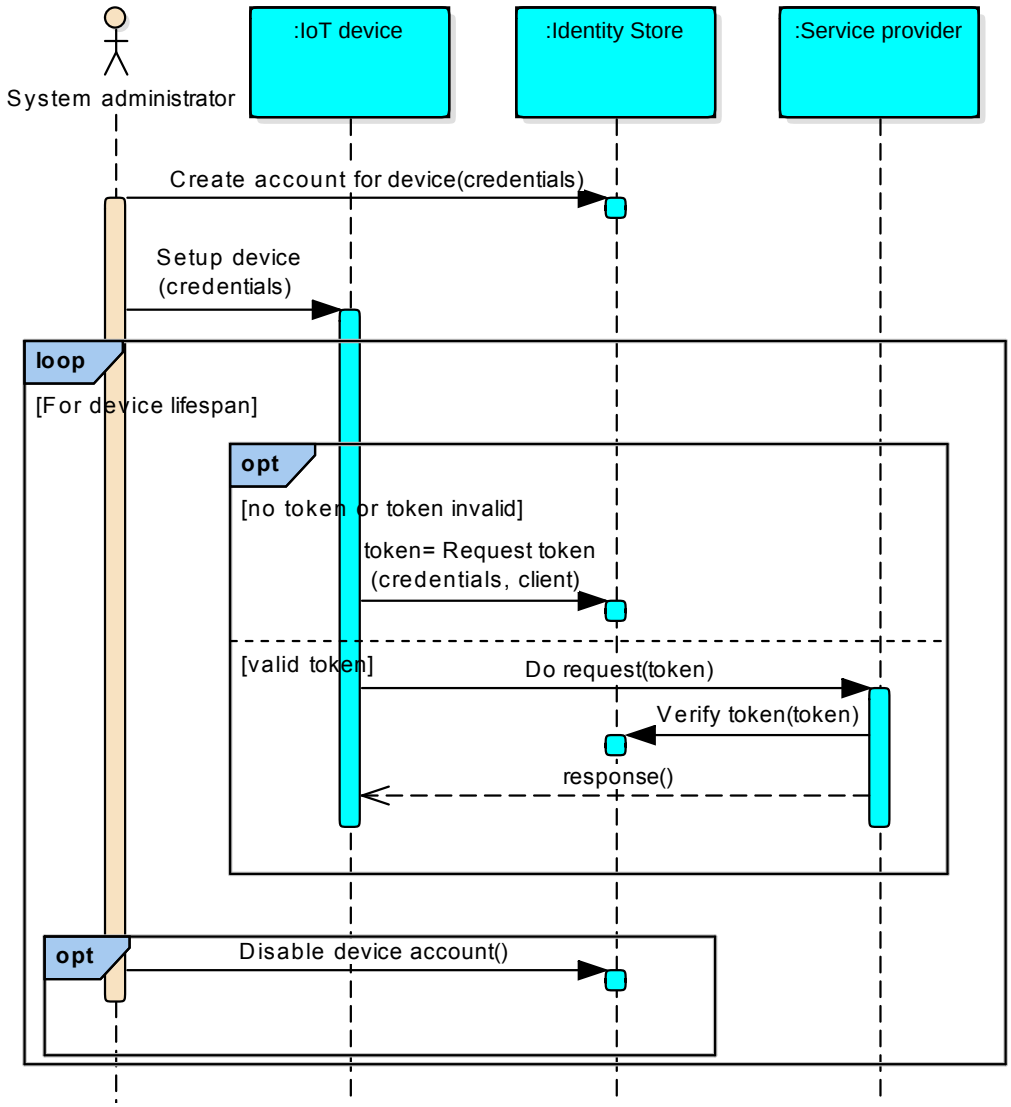


Figure 5.2: Diagram of communication in the proposed solution

are valid solely for the given service provider, and are restricted to a certain period of time. In some instances, a time-unlimited token is viable; in others, a token with short-time validity is preferred. However, once the device obtains a token, it can communicate freely with the partner. The partner can verify the identity of the device and also its roles, based on the presented token.

The solution itself is composed of two parts: the administration application, consisting of the user interface and the IDM server itself, and then the library for IoT devices, consisting mainly of the communication module. Communication between the modules is carried out via the Internet over the family of HTTP Internet protocols [162]. However, support for additional protocols such as MQTT

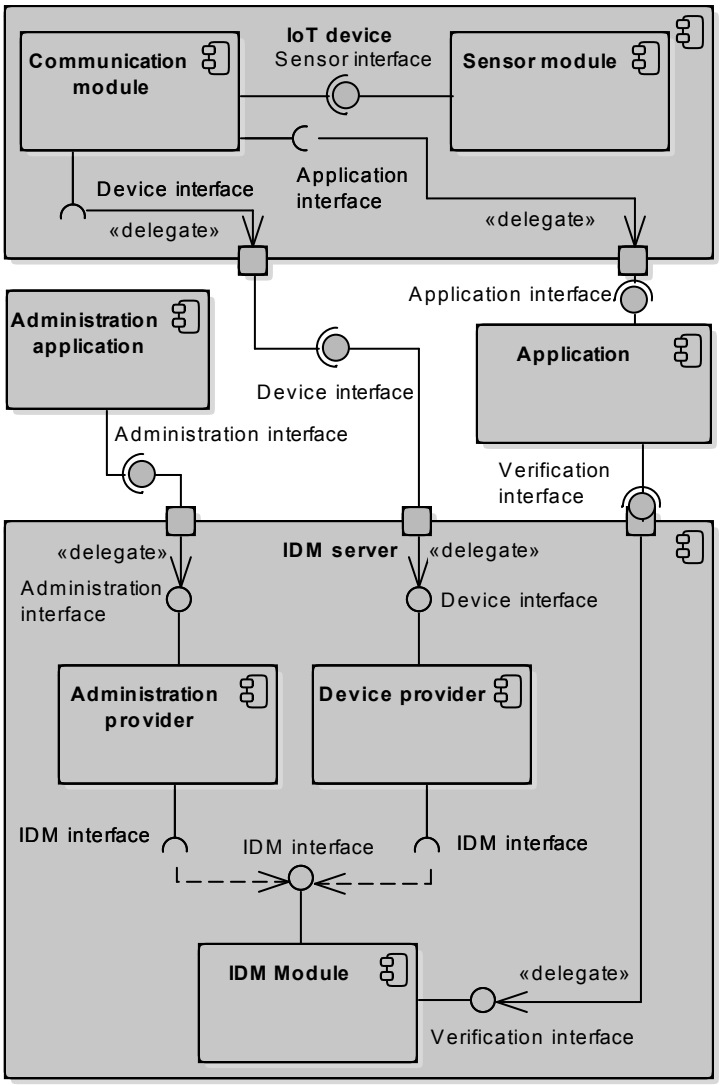


Figure 5.3: Component schema of the proposal

[163] can easily be added. Communication inside the modules is carried out through the native Application Programming Interface (API) of the given programming language. Figure 5.3 presents a component diagram of the proposed architecture.

- IDM module – a module that administers devices and their roles. It also verifies tokens.
- Device provider – allows devices to log in and refresh the token.
- Administration provider – a module that enables an administrator to add, remove or disable devices.

- Administration application – an application that provides the User Interface (UI) for the administrator.
- Communication module – a module which is embedded into the IoT device and takes care of every communication. It authenticates the device, retrieves the token, and uses it for further communication.
- Sensor module – This module contains the business logic of the IoT device.
- Application – An application that that uses data from the sensor. It needs to verify the device's token against the IDM server.

The framework supports communication over REST API [132]. This allows utilization of all the technologies and properties of the HTTP protocol [162]. At first, the TLS [164] protocol (as well as its predecessor SSL protocol [165]) is tightly integrated with the HTTP protocol (called HTTPS [162]). HTTPS provides communication security over the Internet and prevents eavesdropping, tampering or message forgery. Another advantage provided by this approach is that firewalls rarely block communication on ports 80/443. HTTP(S) is tied to REST architecture, and there may be a more suitable protocol. However, no other protocol is so widely used and adapted as HTTP(S).

5.2 Case study

Based on the framework proposal described in the previous section, I created a prototype that builds on existing solutions (as suggested by Finkelstein [166]), integrated together to provide the expected functionality. Building on top of existing solutions allowed me to leverage existing experience and to simplify the transition to possible real usage. Furthermore, using an existing infrastructure allowed me to focus on novel approaches, rather than on re-engineering already solved challenges. And, mainly, because of the existing infrastructure, the applicability and the integrability of the proposal can be verified with existing production-level tools. This ensures that the current state-of-the-art can be extended, and that no other crucial technologies need to be developed as a replacement.

Roman [160] states that traditional Web 2.0 Single Sign-On (SSO), such as OpenID [39] or Shibboleth [167], could also be used in this situation. However, it should be noted that these technologies were not designed to fulfill certain IoT requirements, such as identity disclosure. I therefore opted to try out existing

technologies, in order to determine whether (and to what extent) they are adequate for usage in the IoT ecosystem.

A small-scale simulation of IoT was created for the purposes of the investigation. It consists of the following major elements:

- Central identity store – Keycloak [156] was chosen, as it provides SSO and IDM for web applications and mainly for RESTful web services. For the purposes of this case study, I leveraged mainly the support of the Oauth 2 [140], OpenID Connect [39] and JWT [141] standards.
- Two sensors were used -– movement sensor HC-SR501, and temperature sensor DS18B20. Both of the sensors provide digital output, and can therefore be used without any analog-to-digital converter. However, the sensors still need to be connected to some device with computational capabilities to transmit the data over the Internet. In this case, Raspberry Pi is used to host sensor services.
- An application using data from sensors – simple application with RESTful interface. It gathers data from sensors and exposes them to users via the JavaScript web front end.

The central identity store is deployed as a standalone application. It contains two roles – `temperatureSensor` and `movementSensor`. Next, an account was created for every device and was assigned appropriate roles. The password and username of the given account must be provided to the particular sensor. Authentication token expiry needs to be handled, but to simplify the case study I chose to never let the authentication expire. This allows a device to use the token as long as needed without the necessity to refresh it periodically. The OAuth 2 protocol is used for bearer token [168] acquisition, and the token that is issued follows the JWT standard. The advantage of using JWT tokens is that they contain additional information, such as user roles. Communication with the central identity store can therefore be reduced to a single call aggregating multiple information and thus improving the performance.

The sensors themselves do not possess any computational power, and therefore they need a device to control and observe them. In this case study, they are directly wired onto the bus of a Raspberry Pi computer. A small script written in JavaScript on top of the Node.js framework performs all the sensors' logic. The script differs for various types of sensors, and needs to be initialized with credentials for the particular sensor. The communication process is as follows::

- Acquire a token with the username/password after start-up.
- Every second, information is sent to the central application, with the token for authentication and authorization.
- If the token becomes invalid, attempt to re-authenticate.

All the communication between sensors, the central identity store, and the central application is made through RESTful interfaces. As the sensors are managed by the JavaScript service, additional mocked sensors were deployed into the case study environment, with no impact on the scalability of the infrastructure.

The central application receives data from the sensors and displays them on a web page. In order to do this, the application consists of two parts – the backend and the frontend. The backend part uses Java EE, and it leverages the Keycloak adapter to facilitate integration with the central identity server. This provides a RESTful interface for gathering data from sensors and also for exposing the gathered information. The frontend part of the application is also connected to the RESTful interface.

The case study demonstrates that it is possible to use the proposed scheme, which provides anticipated advantages, such as broad machine-to-machine trust and rapid incident reaction. However, the case study also indicated some limitations that should be addressed. First, there is a need to distribute credentials for each sensor, store them in the device and use them to obtain a token. Second, an administrator needs to manually create an account for each sensor, set up its roles, and propagate identification and a password to the sensor.

5.2.1 Performance evaluation

The performance overhead of our case study is very low. Our measurements show that it takes from 115ms to 130ms (with a mean time of 123ms) to retrieve or refresh the token. This was measured in the Node.js program controlling the sensor. The validity of the token can be determined by the system administrator on the basis of anything between a single request and an unlimited time period. An illustration of the amount of time that a device spends managing the security token is shown in Table 5.1. It can be seen that the overhead would become significant only if data were sent from the sensor every second, using a single use token.

Another consideration is network usage. At first sight, there may seem to be a significant increase in the volume of data that is sent. The data are transferred from the sensor using the HTTP GET request. The request does not contain any

Token validity.	Percentage of device time in the worst scenario
1 second	13 %
1 minute	0.21667%
5 minutes	0.04333%
1 hour	0.00361%
1 day	0.00015%

Table 5.1: Overhead of sensor communication

request body and therefore the length of the request is small. For example, the requests of the temperature sensor are only 189 bytes in length. There are 152 bytes for the URL address, 6 bytes for the data itself, and 31 for various symbols needed in the HTTP request, such as headers. With security added, the HTTP request changes in size to 1398 bytes. The token itself is 1185 bytes long. This may seem to be a colossal overhead; however, even 1kB of added data would be an insignificant increase in the context of current Internet technologies.

5.3 Threats to validity

The measurements for the case study were performed in a small environment. The application ran on the same computer as was used for the user's connection and validation. The sensor network consisted of two real sensors – one motion sensor, and one temperature meter. These sensors were connected to Raspberry Pi, which administered both of them. I did not have sufficient resources to simulate a large-scale IoT environment. The performance is therefore open to question. Based on the performance of Keycloak, there is good reason to believe that thousands of sensors would be manageable. However, I am not able to estimate where the limits of the solution are, whether it is of the order of tens of thousands, hundreds of thousands, or even millions of sensors.

Because of the small testing network, I did not try more than two roles for authorization. There is no doubt that the devices and central stores can manage significantly more roles than any device would ever need. Nevertheless, it is still necessary to administer them. I used the RBAC system, which can become hard to maintain as the number of roles increases. I assume that more than 100 roles would be hard to manage. However, no research has been carried out to estimate how many roles would be needed for the IoT environment.

5.4 Summary

The suggested solution addresses IoT device management with the main focus on centralized authentication, while some attention has also been given to the centralized policy definition point. The proposal presented here is built around a centralized OAuth 2 [140] server that administers all the devices and enables their roles to be defined. The chosen authentication protocol then allows all the devices in the network to communicate securely in the environment. The centralized nature of the authentication enables a fast reaction in case of an adventitious event.

I implemented the solution using Keycloak [156], with a small number of device clients to prove its correctness. The results indicate that the approach is feasible, reasonably simple to implement and, mainly, does not involve a considerable overhead.

Chapter 6

Conclusion

Conventional security architectures and approaches are unsuitable for IoT security for multiple reasons. They do not scale well, they are not prepared for a heterogeneous environment, and they were not built with constrained devices in mind. In addition, they do not leverage the advantages of the IoT, such as broad access to context. With the increasing popularity and prevalence of IoT solutions in recent years, IoT security has become a prominent issue.

During my research, I have focused on extending traditional security approaches with context-aware elements, and transferring them into the IoT environment. I have also proposed a method for context retrieval for IoT devices. Overall, I have developed a solution containing context resolving, I have adapted existing RBAC security architecture to consider contextual information, I have provided a method for sharing and propagating new or updated security rules across IoT devices without an additional overhead, and I have participated in testing the IoT solutions.

The specific contributions of my Ph.D. research can be summarized as follows:

1. A survey with a broad overview of existing security research in the IoT domain. The survey not only lists recent IoT research but also categorizes the research into multiple categories, provides an overview of what research has had the most impact, and analyzes trends.

2. A method for determining the context of IoT devices from their network neighborhood. Devices use a snapshot of the state of the network containing all available devices. This snapshot is then examined and compared, and a significant deviation from the normal state is used for the subsequent authentication (or authorization) rules. This method is largely customizable with various parameters, and it is applicable to any device communicating over the Internet network.

3. Enhancement of (mainly) RBAC with elements of context-awareness. I added another dimension to the architecture that describes the context. The context is expressed through a single-dimensional property called "security level". The results of this effort formed part of the PicketLink open-source security and identity management project [155]. With minor changes, the solution would work with other security architectures. This research [A.10] has had the most impact on the scientific community of all my scientific work. This is the most cited of all my articles until now.

4. A system for sharing authentication and authorization rules in the IoT environment. This system uses existing solutions, namely OAuth 2 [140], OpenID Connect [39] and JWT [141] to propagate security rules. The rules are stored in a centralized authority that acts as a single source of truth for the security policies.

All of this work has been presented at reputable conferences and in peer-reviewed journals. All of the code that I have created is publicly available, either as part of the open-source project, the public git repository, or as an attachment to the articles.

6.1 Future work

The results of the research conducted during my doctoral studies open up multiple opportunities for future work. Initially, a possible research project would be to use some form of an AI algorithm to evaluate the contextual information from the devices. The project can be aimed at determining the parameter values of the algorithm, or it may even remove the need for the parameters, and the AI algorithm will evaluate the security threats.

The context retrieval that I have presented uses the network neighborhood, which is the only subset of context available to devices. In the IoT environment, various devices can recognize various contextual information through their sensors. It would be beneficial to explore the possibility of retrieving the context from all accessible devices and correlating it with the given device (or user). This would require mechanisms for context sharing from devices, and also methods for approximating the relevance of this context to other network participants. I conducted some initial experiments on this topic [A.12] in cooperation with an undergraduate student, but the idea was left unfinished.

The algorithm for determining the benchmark in context retrieval is fairly simple,

and further options can be explored. I have highlighted some possible directions in subsection 3.4.1. However, there are more options that might be feasible. As an example, we could employ an artificial intelligence algorithm to determine the best benchmark creation method and comparison method.

For the proposed security architecture, I enhanced RBAC with security levels. While this allows for basic context-awareness, it cannot express a more complex state of the context. It would be interesting to adapt the solution for usage with ABAC, and to represent the context as attributes.

The sharing of security rules that I have described is done with the use of existing conventional methods - OAuth 2. It would be valuable to explore methods for propagating security rules without the need for a centralized element. This would require the creation of a method for describing the rules in a standard format, a mechanism for discovering and sharing the rules in the network, including verifying them to prevent malicious rules and attacks, and then developing an engine that could apply the rules.

Finally, it would be interesting to explore the possibility of describing the IoT participants using a directed graph. The graph would capture the ownership relationship, the types of devices, the required protection, connections with other participants, and other relevant information. Based on this topology, we could develop a mechanism for determining correct security rules and the places where the security rules should be enforced.

Bibliography

[1] F. Loi, A. Sivanathan, H. H. Gharakheili, A. Radford, and V. Sivaraman, "Systematically evaluating security and privacy for consumer iot devices", in *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, ser. IoTS&P '17, Dallas, Texas, USA: Association for Computing Machinery, 2017, pp. 1–6, ISBN: 9781450353960. DOI: 10.1145/3139937.3139938. [Online]. Available: https://doi.org/10.1145/3139937.3139938.

[2] R. Anderson and T. Moore, "The economics of information security", *Science*, vol. 314, no. 5799, pp. 610–613, 2006, ISSN: 0036-8075. DOI: 10.1126/science.1130992. eprint: https://science.sciencemag.org/content/314/5799/610.full.pdf. [Online]. Available: https://science.sciencemag.org/content/314/5799/610.

[3] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey", *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2010.05.010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128610001568.

[4] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of things security: A survey", *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 2017, ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2017.04.002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517301455.

[5] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness", in *Handheld and Ubiquitous Computing: First International Symposium, HUC'99 Karlsruhe, Germany, September 27–29, 1999 Proceedings*, H.-W. Gellersen, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 304–307, ISBN: 978-3-540-48157-7. DOI: 10.1007/3-540-48157-5_29. [Online]. Available: https://doi.org/10.1007/3-540-48157-5_29.

[6] S.-H. Park, Y.-J. Han, and T.-M. Chung, "Context-role based access control for context-aware application", in *High Performance Computing and Communications*, M. Gerndt and D. Kranzlmüller, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 572–580, ISBN: 978-3-540-39372-6.

[7] R. Bhatti, E. Bertino, and A. Ghafoor, "A trust-based context-aware access control model for web-services", in *Proceedings of the IEEE International Conference on Web Services*, ser. ICWS '04, Washington, DC, USA: IEEE Computer Society, 2004, pp. 184–, ISBN: 0-7695-2167-3. DOI: 10.1109/ICWS.2004.15. [Online]. Available: https://doi.org/10.1109/ICWS.2004.15.

[8] R. J. Hulsebosch, A. H. Salden, M. S. Bargh, P. W. G. Ebben, and J. Reitsma, "Context sensitive access control", in *Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies*, ser. SACMAT '05, Stockholm, Sweden: ACM, 2005, pp. 111–119, ISBN: 1-59593-045-0. DOI: 10.1145/1063979.1064000. [Online]. Available: http://doi.acm.org/10.1145/1063979.1064000.

[9] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff, "A brief history of the internet", *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 5, pp. 22–31, Oct. 2009, ISSN: 0146-4833. DOI: 10.1145/1629607.1629613. [Online]. Available: http://doi.acm.org/10.1145/1629607.1629613.

[10] C.-L. Hsu, H.-P. Lu, and H.-H. Hsu, "Adoption of the mobile internet: An empirical study of multimedia message service (mms)", *Omega*, vol. 35, no. 6, pp. 715–726, 2007, Special Issue on Telecommunications Applications, ISSN: 0305-0483. DOI: https://doi.org/10.1016/j.omega.2006.03.005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305048306000594.

[11] G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. D. Johnson, "M2m: From mobile to embedded internet", *IEEE Communications Magazine*, vol. 49, no. 4, pp. 36–43, Apr. 2011, ISSN: 0163-6804. DOI: 10.1109/MCOM.2011.5741144.

[12] Cisco Systems, Inc., "Cisco annual internet report (2018–2023)", Tech. Rep., 2020, Accessed on 20.9.2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[13] Intel Corporation, "Guide to internet of things", Tech. Rep., 2020, Accessed on 20.9.2020. [Online]. Available: https://www.intel.com/content/www/us/en/internet-of-things/infographics/guide-to-iot.html.

[14] MarketsandMarkets Research Private Ltd., "Iot solutions and services market by component (platform, solution and services), service (consulting, and integration and deplpyment), vertical (smart manufacturing, smart energy and smart transportation), and region - global forecast to 2024", Tech. Rep., 2020, Accessed on 25.10.2020. [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/iot-solutions-and-services-market-120466720.html.

[15] D. S. Markovic, D. Zivkovic, I. Branovic, R. Popovic, and D. Cvetkovic, "Smart power grid and cloud computing", *Renewable and Sustainable Energy Reviews*, vol. 24, pp. 566–577, 2013, ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2013.03.068. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S136403211300227X.

[16] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "Smart Grid Technologies: Communication Technologies and Standards", *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, vol. 7, no. 4, 529–539, Nov. 2011, ISSN: 1551-3203. DOI: {10.1109/TII.2011.2166794}.

[17] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart Grid - The New and Improved Power Grid: A Survey", *IEEE COMMUNICATIONS SURVEYS AND TUTORIALS*, vol. 14, no. 4, 944–980, 2012. DOI: {10.1109/SURV.2011.101911.00087}.

[18] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "A Survey on Smart Grid Potential Applications and Communication Requirements", *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, vol. 9, no. 1, 28–42, Feb. 2013, ISSN: 1551-3203. DOI: {10.1109/TII.2012.2218253}.

[19] R. Morello, C. De Capua, G. Fulco, and S. C. Mukhopadhyay, "A smart power meter to monitor energy flow in smart grids: The role of advanced sensing and iot in the electric grid of the future", *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7828–7837, 2017.

[20] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of things for smart healthcare: Technologies, challenges, and opportunities", *IEEE Access*, vol. 5, pp. 26 521–26 544, 2017, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2775180.

[21] A. Pantelopoulos and N. G. Bourbakis, "A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis", *IEEE TRANSACTIONS ON SYSTEMS MAN AND CYBERNETICS PART C-APPLICATIONS AND REVIEWS*, vol. 40, no. 1, 1–12, Jan. 2010, ISSN: 1094-6977. DOI: {10.1109/TSMCC.2009.2032660}.

[22] Y.-L. Zheng, X.-R. Ding, C. C. Y. Poon, B. P. L. Lo, H. Zhang, X.-L. Zhou, G.-Z. Yang, N. Zhao, and Y.-T. Zhang, "Unobtrusive Sensing and Wearable Devices for Health Informatics", *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, vol. 61, no. 5, SI, 1538–1554, May 2014, ISSN: 0018-9294. DOI: {10.1109/TBME.2014.2309951}.

[23] M. M. E. Mahmoud, J. J. P. C. Rodrigues, S. H. Ahmed, S. C. Shah, J. F. Al-Muhtadi, V. V. Korotaev, and V. H. C. De Albuquerque, "Enabling technologies on cloud of things for smart healthcare", *IEEE Access*, vol. 6, pp. 31 950–31 967, 2018.

[24] A. Solanas, C. Patsakis, M. Conti, *et al.*, "Smart health: A context-aware health paradigm within smart cities", *IEEE Communications Magazine*, vol. 52, no. 8, pp. 74–81, Aug. 2014, ISSN: 0163-6804. DOI: 10.1109/MCOM.2014.6871673.

[25] T. Nam and T. A. Pardo, "Conceptualizing smart city with dimensions of technology, people, and institutions", in *Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times*, ser. dg.o '11, College Park, Maryland, USA: Association for Computing Machinery, 2011, pp. 282–291, ISBN: 9781450307628. DOI: 10.1145/2037556.2037602. [Online]. Available: https://doi.org/10.1145/2037556.2037602.

[26] V. Albino, U. Berardi, and R. M. Dangelico, "Smart Cities: Definitions, Dimensions, Performance, and Initiatives", *JOURNAL OF URBAN TECHNOLOGY*, vol. 22, no. 1, 3–21, Jan. 2015, ISSN: 1063-0732. DOI: {10.1080/10630732.2014.942092}.

[27] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities", *IEEE INTERNET OF THINGS JOURNAL*, vol. 1, no. 1, 22–32, Feb. 2014, ISSN: 2327-4662. DOI: {10.1109/JIOT.2014.2306328}.

[28] M. Batty, K. W. Axhausen, F. Giannotti, A. Pozdnoukhov, A. Bazzani, M. Wachowicz, G. Ouzounis, and Y. Portugali, "Smart cities of the future", *EUROPEAN PHYSICAL JOURNAL-SPECIAL TOPICS*, vol. 214, no. 1, 481–518, Nov. 2012, ISSN: 1951-6355. DOI: {10.1140/epjst/e2012-01703-3}.

[29] C. Benevolo, R. P. Dameri, and B. D'Auria, "Smart mobility in smart city", in *Empowering Organizations*, T. Torre, A. M. Braccini, and R. Spinelli, Eds., Cham: Springer International Publishing, 2016, pp. 13–28, ISBN: 978-3-319-23784-8.

[30] Z. Ning, F. Xia, N. Ullah, X. Kong, and X. Hu, "Vehicular Social Networks: Enabling Smart Mobility", *IEEE COMMUNICATIONS MAGAZINE*, vol. 55, no. 5, 49–55, May 2017, ISSN: 0163-6804. DOI: {10.1109/MCOM.2017.1600263}.

[31] A. Meijer, "Smart city governance: A local emergent perspective", in *Smarter as the New Urban Agenda: A Comprehensive View of the 21st Century City*, J. R. Gil-Garcia, T. A. Pardo, and T. Nam, Eds. Cham: Springer International Publishing, 2016, pp. 73–85, ISBN: 978-3-319-17620-8. DOI: 10.1007/978-3-319-17620-8_4. [Online]. Available: https://doi.org/10.1007/978-3-319-17620-8_4.

[32] G. V. Pereira, P. Parycek, E. Falco, and R. Kleinhans, "Smart governance in the context of smart cities: A literature review", *INFORMATION POLITY*, vol. 23, no. 2, 143–162, 2018, ISSN: 1570-1255. DOI: {10.3233/IP-170067}.

[33] A. Meijer and M. P. Rodriguez Bolivar, "Governing the smart city: a review of the literature on smart urban governance", *INTERNATIONAL REVIEW OF ADMINISTRATIVE SCIENCES*, vol. 82, no. 2, SI, 392–408, Jun. 2016, ISSN: 0020-8523. DOI: {10.1177/0020852314564308}.

[34] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A review of smart homes—past, present, and future", *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1190–1203, Nov. 2012, ISSN: 1558-2442. DOI: 10.1109/TSMCC.2012.2189204.

[35] M. Chan, D. Esteve, C. Escriba, and E. Campo, "A review of smart homes - Present state and future challenges", *COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE*, vol. 91, no. 1, 55–81, Jun. 2008, ISSN: 0169-2607. DOI: {10.1016/j.cmpb.2008.02.001}.

[36] R. Sandhu, "Access control: The neglected frontier", in *Information Security and Privacy*, J. Pieprzyk and J. Seberry, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 219–227, ISBN: 978-3-540-49583-3.

[37] D. Ferraiolo, J. Cugini, and D. R. Kuhn, "Role-based access control (rbac): Features and motivations", in *Proceedings of 11th annual computer security application conference*, 1995, pp. 241–48.

[38] K. Zeilenga, "Lightweight directory access protocol (ldap): Technical specification road map", RFC Editor, RFC 4510, Jun. 2006, http://www.rfc-editor.org/rfc/rfc4510.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4510.txt.

[39] N. Sakimura, J. Bradley, M. Jones, B. De Medeiros, and C. Mortimore, *Openid connect core 1.0 incorporating errata set 1*, 2014. [Online]. Available: https://openid.net/connect/.

[40] A. K. Dey, "Understanding and using context", *Personal Ubiquitous Comput.*, vol. 5, no. 1, pp. 4–7, Jan. 2001, ISSN: 1617-4909. DOI: 10.1007/s007790170019. [Online]. Available: http://dx.doi.org/10.1007/s007790170019.

[41] G. Chen and D. Kotz, "A survey of context-aware mobile computing research", Hanover, NH, USA, Tech. Rep., 2000.

[42] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application", *Wirel. Netw.*, vol. 8, no. 2/3, pp. 187–197, Mar. 2002, ISSN: 1022-0038. DOI: 10.1023/A:1013767926256. [Online]. Available: http://dx.doi.org/10.1023/A:1013767926256.

[43] T. O'Reilly, "What is web 2.0? design patterns and business models for the next generation of software.", 2005. [Online]. Available: http://oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.

[44] N. Mallat, M. Rossi, V. K. Tuunainen, and A. Öörni, "The impact of use context on mobile services acceptance: The case of mobile ticketing", *Information & Management*, vol. 46, no. 3, pp. 190–195, 2009, ISSN: 0378-7206. DOI: https://doi.org/10.1016/j.im.2008.11.008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378720609000202.

[45] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems", *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 2, no. 4, pp. 263–277, Jun. 2007, ISSN: 1743-8225. DOI: 10.1504/IJAHUC.2007.014070. [Online]. Available: http://dx.doi.org/10.1504/IJAHUC.2007.014070.

[46] H. Chen, T. Finin, Anupam Joshi, L. Kagal, F. Perich, and Dipanjan Chakraborty, "Intelligent agents meet the semantic web in smart spaces", *IEEE Internet Computing*, vol. 8, no. 6, pp. 69–79, 2004.

[47] M. Perttunen, J. Riekki, and O. Lassila, "Context representation and reasoning in pervasive computing: A review", *International Journal of Multimedia and Ubiquitous Engineering*, pp. 1–28,

[48] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques", *Pervasive and Mobile Computing*, vol. 6, no. 2, pp. 161–180, 2010, Context Modelling, Reasoning and Management, ISSN: 1574-1192. DOI: https://doi.org/10.1016/j.pmcj.2009.06.002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119209000510.

[49] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey", *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 414–454, First 2014, ISSN: 1553-877X. DOI: 10.1109/SURV.2013.042313.00197.

[50] M. J. Moyer and M. Abamad, "Generalized role-based access control", in *Proceedings 21st International Conference on Distributed Computing Systems*, 2001, pp. 391–398.

[51] M. J. Covington, W. Long, S. Srinivasan, A. K. Dev, M. Ahamad, and G. D. Abowd, "Securing context-aware applications using environment roles", in *Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies*, ser. SACMAT '01, Chantilly, Virginia, USA: ACM, 2001, pp. 10–20, ISBN: 1-58113-350-2. DOI: 10.1145/373256.373258. [Online]. Available: http://doi.acm.org/10.1145/373256.373258.

[52] G. Sladic, B. Milosavljević, and Z. Konjovic, "Context-sensitive access control model for business processes", vol. 10, pp. 939–972, Jun. 2013.

[53] D. Kulkarni and A. Tripathi, "Context-aware role-based access control in pervasive computing systems", in *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies*, ser. SACMAT '08, Estes Park, CO, USA: Association for Computing Machinery, 2008, pp. 113–122, ISBN: 9781605581293. DOI: 10.1145/1377836.1377854. [Online]. Available: https://doi.org/10.1145/1377836.1377854.

[54] G. Neumann and M. Strembeck, "An approach to engineer and enforce context constraints in an rbac environment", in *Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies*, ser. SACMAT '03, Como, Italy: Association for Computing Machinery, 2003, pp. 65–79, ISBN: 1581136811. DOI: 10.1145/775412.775421. [Online]. Available: https://doi.org/10.1145/775412.775421.

[55] G. K. Mostéfaoui and P. Brézillon, "A generic framework for context-based distributed authorizations", in *Modeling and Using Context*, P. Blackburn, C. Ghidini, R. M. Turner, and F. Giunchiglia, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 204–217, ISBN: 978-3-540-44958-4.

[56] J. C. D. Lima, C. C. Rocha, I. Augustin, and M. A. R. Dantas, "A context-aware recommendation system to behavioral based authentication in mobile and pervasive environments", in *2011 IFIP 9th International Conference on Embedded and Ubiquitous Computing*, 2011, pp. 312–319.

[57] A. Corrad, R. Montanari, and D. Tibaldi, "Context-based access control management in ubiquitous environments", in *Third IEEE International Symposium on Network Computing and Applications, 2004. (NCA 2004). Proceedings.*, 2004, pp. 253–260.

[58] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update", *Information and Software Technology*, vol. 64, no. Supplement C, pp. 1–18, 2015, ISSN: 0950-5849.

DOI: https://doi.org/10.1016/j.infsof.2015.03.007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584915000646.

[59] M. binti Mohamad Noor and W. H. Hassan, "Current research on internet of things (iot) security: A survey", *Computer Networks*, vol. 148, pp. 283–294, 2019, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2018.11.025. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128618307035.

[60] P. M. Chanal and M. S. Kakkasageri, "Security and privacy in iot: A survey", *Wireless Personal Communications*, vol. 115, no. 2, pp. 1667–1693, Nov. 2020, ISSN: 1572-834X. DOI: 10.1007/s11277-020-07649-9. [Online]. Available: https://doi.org/10.1007/s11277-020-07649-9.

[61] N. Miloslavskaya and A. Tolstoy, "Internet of things: Information security challenges and solutions", *Cluster Computing*, vol. 22, no. 1, pp. 103–119, Mar. 2019, ISSN: 1573-7543. DOI: 10.1007/s10586-018-2823-6. [Online]. Available: https://doi.org/10.1007/s10586-018-2823-6.

[62] F. H. Al-Naji and R. Zagrouba, "A survey on continuous authentication methods in internet of things environment", *Computer Communications*, vol. 163, pp. 109–133, 2020, ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2020.09.006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366420319204.

[63] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, "A systematic survey of industrial internet of things security: Requirements and fog computing opportunities", *IEEE Communications Surveys Tutorials*, vol. 22, no. 4, pp. 2489–2520, 2020. DOI: 10.1109/COMST.2020.3011208.

[64] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic keyword extraction from individual documents", *Text Mining: Applications and Theory*, pp. 1–20, 2010. DOI: 10.1002/9780470689646.ch1.

[65] *Pdftotext*. [Online]. Available: http://www.xpdfreader.com.

[66] I. Agadakos, P. Hallgren, D. Damopoulos, A. Sabelfeld, and G. Portokalidis, "Location-enhanced authentication using the iot: Because you cannot be in two places at once", in *Proceedings of the 32Nd Annual Conference on Computer Security Applications*, ser. ACSAC '16, Los Angeles, California, USA: ACM, 2016, pp. 251–264, ISBN: 978-1-4503-4771-6. DOI: 10.1145/2991079.2991090. [Online]. Available: http://doi.acm.org/10.1145/2991079.2991090.

[67] G. Alpár, L. Batina, L. Batten, V. Moonsamy, A. Krasnova, A. Guellier, and I. Natgunanathan, "New directions in iot privacy using attribute-based authentication", in *Proceedings of the ACM International Conference on Computing Frontiers*, ser. CF '16, Como, Italy: ACM, 2016, pp. 461–466, ISBN: 978-1-4503-4128-8. DOI: 10.1145/2903150.2911710. [Online]. Available: http://doi.acm.org/10.1145/2903150.2911710.

[68] L. Barreto, A. Celesti, M. Villari, M. Fazio, and A. Puliafito, "An authentication model for iot clouds", in *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Aug. 2015, pp. 1032–1035. DOI: 10.1145/2808797.2809361.

[69] M. Cagnazzo, M. Hertlein, and N. Pohlmann, "An usable application for authentication, communication and access management in the internet of things", in *Information and Software Technologies: 22nd International Conference, ICIST 2016, Druskininkai, Lithuania, October 13-15, 2016, Proceedings*. Cham: Springer International Publishing, 2016, pp. 722–731, ISBN: 978-3-319-46254-7. DOI: 10.1007/978-3-319-46254-7_58. [Online]. Available: https://doi.org/10.1007/978-3-319-46254-7_58.

[70] F. Chen, Y. Luo, J. Zhang, J. Zhu, Z. Zhang, C. Zhao, and T. Wang, "An infrastructure framework for privacy protection of community medical internet of things", *World Wide Web*, Apr. 2017, ISSN: 1573-1413. DOI: 10.1007/s11280-017-0455-z. [Online]. Available: https://doi.org/10.1007/s11280-017-0455-z.

[71] P. Fremantle, J. Kopecký, and B. Aziz, "Web api management meets the internet of things", in *The Semantic Web: ESWC 2015 Satellite Events: ESWC 2015 Satellite Events, Portorož, Slovenia, May 31 – June 4, 2015, Revised Selected Papers*. Cham: Springer International Publishing, 2015, pp. 367–375, ISBN: 978-3-319-25639-9. DOI: 10.1007/978-3-319-25639-9_49. [Online]. Available: https://doi.org/10.1007/978-3-319-25639-9_49.

[72] S. Gerdes, C. Bormann, and O. Bergmann, "Chapter 11 - keeping users empowered in a cloudy internet of things", in *The Cloud Security Ecosystem*, R. Ko and K.-K. R. Choo, Eds., Boston: Syngress, 2015, pp. 231–247, ISBN: 978-0-12-801595-7. DOI: https://doi.org/10.1016/B978-0-12-801595-7.00011-2. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128015957000112.

[73] J. L. Hernández-Ramos, M. P. Pawlowski, A. J. Jara, A. F. Skarmeta, and L. Ladid, "Toward a lightweight authentication and authorization framework for smart objects", *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 4, pp. 690–702, Apr. 2015, ISSN: 0733-8716. DOI: 10.1109/JSAC.2015.2393436.

[74] T. Kumar, A. Braeken, M. Liyanage, and M. Ylianttila, "Identity privacy preserving biometric based authentication scheme for naked healthcare environment", in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–7. DOI: 10.1109/ICC.2017.7996966.

[75] S.-H. Lee, K.-W. Huang, and C.-S. Yang, "Tbas: Token-based authorization service architecture in internet of things scenarios", *International Journal of Distributed Sensor Networks*, vol. 13, no. 7, p. 1 550 147 717 718 496, 2017. DOI: 10.1177/1550147717718496. eprint: https://doi.org/10.1177/1550147717718496. [Online]. Available: https://doi.org/10.1177/1550147717718496.

[76] L. Liu, B. Fang, and B. Yi, "A general framework of nonleakage-based authentication using csp for the internet of things", in *Web Technologies and Applications: APWeb 2014 Workshops, SNA, NIS, and IoTS, Changsha, China, September*

5, 2014. Proceedings. Cham: Springer International Publishing, 2014, pp. 312–324, ISBN: 978-3-319-11119-3. DOI: 10.1007/978-3-319-11119-3_29. [Online]. Available: https://doi.org/10.1007/978-3-319-11119-3_29.

[77] A. Pinto and R. Costa, "Hash-chain based authentication for iot devices and rest web-services", in *Ambient Intelligence- Software and Applications – 7th International Symposium on Ambient Intelligence (ISAmI 2016)*, H. Lindgren, J. F. De Paz, P. Novais, A. Fernández-Caballero, H. Yoe, A. Jiménez Ramírez, and G. Villarrubia, Eds. Cham: Springer International Publishing, 2016, pp. 189–196, ISBN: 978-3-319-40114-0. DOI: 10.1007/978-3-319-40114-0_21. [Online]. Available: https://doi.org/10.1007/978-3-319-40114-0_21.

[78] M. Shahzad and M. P. Singh, "Continuous authentication and authorization for the internet of things", *IEEE Internet Computing*, vol. 21, no. 2, pp. 86–90, Mar. 2017, ISSN: 1089-7801. DOI: 10.1109/MIC.2017.33.

[79] N. Shone, C. Dobbins, W. Hurst, and Q. Shi, "Digital memories based mobile user authentication for iot", in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, Oct. 2015, pp. 1796–1802. DOI: 10.1109/CIT/IUCC/DASC/PICOM.2015.270.

[80] Q. Tasali, C. Chowdhury, and E. Y. Vasserman, "A flexible authorization architecture for systems of interoperable medical devices", in *Proceedings of the 22Nd ACM on Symposium on Access Control Models and Technologies*, ser. SACMAT '17 Abstracts, Indianapolis, Indiana, USA: ACM, 2017, pp. 9–20, ISBN: 978-1-4503-4702-0. DOI: 10.1145/3078861.3078862. [Online]. Available: http://doi.acm.org/10.1145/3078861.3078862.

[81] S. Unger and D. Timmermann, "Dpwsec: Devices profile for web services security", in *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Apr. 2015, pp. 1–6. DOI: 10.1109/ISSNIP.2015.7106961.

[82] S. Wiseman, G. Soto Mino, A. L. Cox, S. J. Gould, J. Moore, and C. Needham, "Use your words: Designing one-time pairing codes to improve user experience", in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16, San Jose, California, USA: ACM, 2016, pp. 1385–1389, ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858377. [Online]. Available: http://doi.acm.org/10.1145/2858036.2858377.

[83] S. Sicari, A. Rizzardi, L. Grieco, G. Piro, and A. Coen-Porisini, "A policy enforcement framework for internet of things applications in the smart health", *Smart Health*, vol. 3-4, pp. 39–74, 2017, ISSN: 2352-6483. DOI: https://doi.org/10.1016/j.smhl.2017.06.001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352648316300435.

[84] A. Outchakoucht, H. ES-SAMAALI, and J. Philippe, "Dynamic access control policy based on blockchain and machine learning for the internet of things", *International Journal of Advanced Computer Science and Applications*, vol. 8, Jan. 2017. DOI: 10.14569/IJACSA.2017.080757. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2017.080757.

[85] N. YE, Y. Zhu, R.-c. WANG, R. Malekian, and L. Qiao-min, "An efficient authentication and access control scheme for perception layer of internet of things", *Applied Mathematics & Information Sciences*, vol. 8, Jul. 2014.

[86] J. Bernal Bernabe, J. L. Hernandez-Ramos, and A. F. Skarmeta Gomez, "Holistic privacy-preserving identity management system for the internet of things", *Mobile Information Systems*, vol. 2017, p. 20, 2017. DOI: 10.1155/2017/6384186.

[87] B.-C. Chifor, I. Bica, V.-V. Patriciu, and F. Pop, "A security authorization scheme for smart home internet of things devices", *Future Generation Computer Systems*, vol. 86, pp. 740–749, 2018, ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2017.05.048. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17311020.

[88] W. Chiu, C. Su, C.-Y. Fan, C.-M. Chen, and K.-H. Yeh, "Authentication with what you see and remember in the internet of things", *Symmetry*, vol. 10, no. 11, p. 537, Oct. 2018, ISSN: 2073-8994. DOI: 10.3390/sym10110537. [Online]. Available: http://dx.doi.org/10.3390/sym10110537.

[89] F. Sun, C. Mao, X. Fan, and Y. Li, "Accelerometer-based speed-adaptive gait authentication method for wearable iot devices", *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 820–830, 2019. DOI: 10.1109/JIOT.2018.2860592.

[90] S.-R. Oh, Y.-G. Kim, and S. Cho, "An interoperable access control framework for diverse iot platforms based on oauth and role", *Sensors*, vol. 19, no. 8, p. 1884, Apr. 2019, ISSN: 1424-8220. DOI: 10.3390/s19081884. [Online]. Available: http://dx.doi.org/10.3390/s19081884.

[91] H. Yan, Y. Wang, C. Jia, J. Li, Y. Xiang, and W. Pedrycz, "Iot-fbac: Function-based access control scheme using identity-based encryption in iot", *Future Generation Computer Systems*, vol. 95, pp. 344–353, 2019, ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2018.12.061. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X1830997X.

[92] P. Nespoli, M. Zago, A. Huertas Celdrán, M. Gil Pérez, F. Gómez Mármol, and F. J. García Clemente, "Palot: Profiling and authenticating users leveraging internet of things", *Sensors*, vol. 19, no. 12, p. 2832, Jun. 2019, ISSN: 1424-8220. DOI: 10.3390/s19122832. [Online]. Available: http://dx.doi.org/10.3390/s19122832.

[93] N. Ghosh, S. Chandra, V. Sachidananda, and Y. Elovici, "Softauthz: A context-aware, behavior-based authorization framework for home iot", *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 773–10 785, 2019. DOI: 10.1109/JIOT.2019.2941767.

[94] S. Batool, A. Hassan, N. A. Saqib, and M. A. K. Khattak, "Authentication of remote iot users based on deeper gait analysis of sensor data", *IEEE Access*, vol. 8, pp. 101 784–101 796, 2020. DOI: 10.1109/ACCESS.2020.2998412.

[95] G. Ali, N. Ahmad, Y. Cao, S. Khan, H. Cruickshank, E. A. Qazi, and A. Ali, "Xdbauth: Blockchain based cross domain authentication and authorization framework for internet of things", *IEEE Access*, vol. 8, pp. 58 800–58 816, 2020. DOI: 10.1109/ACCESS.2020.2982542.

[96] S.-R. Oh and Y.-G. Kim, "Afaas: Authorization framework as a service for internet of things based on interoperable oauth", *International Journal of Distributed Sensor Networks*, vol. 16, no. 2, p. 1 550 147 720 906 388, 2020. DOI: 10.1177/1550147720906388. eprint: https://doi.org/10.1177/1550147720906388. [Online]. Available: https://doi.org/10.1177/1550147720906388.

[97] U. Khalid, M. Asim, T. Baker, P. C. K. Hung, M. A. Tariq, and L. Rafferty, "A decentralized lightweight blockchain-based authentication mechanism for iot systems", *Cluster Computing*, vol. 23, no. 3, pp. 2067–2087, Sep. 2020, ISSN: 1573-7543. DOI: 10.1007/s10586-020-03058-6.

[98] S. Zhang, Y. Cao, Z. Ning, F. Xue, D. Cao, and Y. Yang, "A Heterogeneous IoT Node Authentication Scheme Based on Hybrid Blockchain and Trust Value", *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS*, vol. 14, no. 9, 3615–3638, Sep. 2020, ISSN: 1976-7277. DOI: 10.3837/tiis.2020.09.003.

[99] K. N. Pallavi and V. Ravi Kumar, "Authentication-based access control and data exchanging mechanism of iot devices in fog computing environment", *Wireless Personal Communications*, Oct. 2020, ISSN: 1572-834X. DOI: 10.1007/s11277-020-07834-w. [Online]. Available: https://doi.org/10.1007/s11277-020-07834-w.

[100] A. Alkhresheh, K. Elgazzar, and H. S. Hassanein, "Daciot: Dynamic access control framework for iot deployments", *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11 401–11 419, 2020. DOI: 10.1109/JIOT.2020.3002709.

[101] J. Bernal Bernabe, J. L. Hernandez Ramos, and A. F. Skarmeta Gomez, "Taciot: Multidimensional trust-aware access control system for the internet of things", *Soft Computing*, vol. 20, no. 5, pp. 1763–1779, May 2016, ISSN: 1433-7479. DOI: 10.1007/s00500-015-1705-6. [Online]. Available: https://doi.org/10.1007/s00500-015-1705-6.

[102] S. Cirani and M. Picone, "Effective authorization for the web of things", in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Dec. 2015, pp. 316–320. DOI: 10.1109/WF-IoT.2015.7389073.

[103] W. Han, Y. Zhang, Z. Guo, and E. Bertino, "Fine-grained business data confidentiality control in cross-organizational tracking", in *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies*, ser. SACMAT '15, Vienna, Austria: ACM, 2015, pp. 135–145, ISBN: 978-1-4503-3556-0. DOI: 10.1145/2752952.2752973. [Online]. Available: http://doi.acm.org/10.1145/2752952.2752973.

[104] A. Kurniawan and M. Kyas, "A trust model-based bayesian decision theory in large scale internet of things", in *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Apr. 2015, pp. 1–5. DOI: 10.1109/ISSNIP.2015.7106964.

[105] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman, "Towards a novel privacy-preserving access control model based on blockchain technology in iot", in *Europe and MENA Cooperation Advances in Information and Communication Technologies*, Á. Rocha, M. Serrhini, and C. Felgueiras, Eds. Cham: Springer International

Publishing, 2017, pp. 523–533, ISBN: 978-3-319-46568-5. DOI: 10.1007/978-3-319-46568-5_53. [Online]. Available: https://doi.org/10.1007/978-3-319-46568-5_53.

[106] P. Solapurkar, "Building secure healthcare services using oauth 2.0 and json web token in iot cloud scenario", in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, Dec. 2016, pp. 99–104. DOI: 10.1109/IC3I.2016.7917942.

[107] S. Lee, J. Choi, J. Kim, B. Cho, S. Lee, H. Kim, and J. Kim, "Fact: Functionality-centric access control system for iot programming frameworks", in *Proceedings of the 22Nd ACM on Symposium on Access Control Models and Technologies*, ser. SACMAT '17 Abstracts, Indianapolis, Indiana, USA: ACM, 2017, pp. 43–54, ISBN: 978-1-4503-4702-0. DOI: 10.1145/3078861.3078864. [Online]. Available: http://doi.acm.org/10.1145/3078861.3078864.

[108] S. Bandara, T. Yashiro, N. Koshizuka, and K. Sakamura, "Access control framework for api-enabled devices in smart buildings", in *2016 22nd Asia-Pacific Conference on Communications (APCC)*, Aug. 2016, pp. 210–217. DOI: 10.1109/APCC.2016.7581479.

[109] A. Biason, C. Pielli, A. Zanella, and M. Zorzi, "Access control for iot nodes with energy and fidelity constraints", *IEEE Transactions on Wireless Communications*, pp. 1–1, 2018, ISSN: 1536-1276. DOI: 10.1109/TWC.2018.2808520.

[110] Q. Huang, L. Wang, and Y. Yang, "Decent: Secure and fine-grained data access control with policy updating for constrained iot devices", *World Wide Web*, vol. 21, no. 1, pp. 151–167, Jan. 2018, ISSN: 1573-1413. DOI: 10.1007/s11280-017-0462-0. [Online]. Available: https://doi.org/10.1007/s11280-017-0462-0.

[111] J. A. Martínez, J. L. Hernández-Ramos, V. Beltrán, A. Skarmeta, and P. M. Ruiz, "A user-centric internet of things platform to empower users for managing security and privacy concerns in the internet of energy", *International Journal of Distributed Sensor Networks*, vol. 13, no. 8, p. 1 550 147 717 727 974, 2017. DOI: 10.1177/1550147717727974. [Online]. Available: https://doi.org/10.1177/1550147717727974.

[112] S. Pal, T. Rabehaja, M. Hitchens, V. Varadharajan, and A. Hill, "On the design of a flexible delegation model for the internet of things using blockchain", *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3521–3530, 2020. DOI: 10.1109/TII.2019.2925898.

[113] C. Lupascu, A. Lupascu, and I. Bica, "Dlt based authentication framework for industrial iot devices", *Sensors*, vol. 20, no. 9, p. 2621, May 2020, ISSN: 1424-8220. DOI: 10.3390/s20092621. [Online]. Available: http://dx.doi.org/10.3390/s20092621.

[114] H. B. Djilali, D. Tandjaoui, and H. Khemissa, "Enhanced dynamic team access control for collaborative internet of things using context", *Transactions on Emerging Telecommunications Technologies*, vol. n/a, no. n/a, e4083, DOI: https://doi.org/10.1002/ett.4083. eprint: https://onlinelibrary.wiley.

com/doi/pdf/10.1002/ett.4083. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4083.

[115] S. Gusmeroli, S. Piccione, and D. Rotondi, "A capability-based security approach to manage access control in the internet of things", *Mathematical and Computer Modelling*, vol. 58, no. 5, pp. 1189–1205, 2013, The Measurement of Undesirable Outputs: Models Development and Empirical Analyses and Advances in mobile, ubiquitous and cognitive computing, ISSN: 0895-7177. DOI: https://doi.org/10.1016/j.mcm.2013.02.006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S089571771300054X.

[116] A. Majeed and A. Al-Yasiri, "Formulating a global identifier based on actor relationship for the internet of things", in *Interoperability, Safety and Security in IoT: Second International Conference, InterIoT 2016 and Third International Conference, SaSeIoT 2016, Paris, France, October 26-27, 2016, Revised Selected Papers*. Cham: Springer International Publishing, 2017, pp. 79–91, ISBN: 978-3-319-52727-7. DOI: 10.1007/978-3-319-52727-7_10. [Online]. Available: https://doi.org/10.1007/978-3-319-52727-7_10.

[117] D. Schreckling, J. D. Parra, C. Doukas, and J. Posegga, "Data-centric security for the iot", in *Internet of Things. IoT Infrastructures: Second International Summit, IoT 360 2015, Rome, Italy, October 27-29, 2015, Revised Selected Papers, Part II*. Cham: Springer International Publishing, 2016, pp. 77–86, ISBN: 978-3-319-47075-7. DOI: 10.1007/978-3-319-47075-7_10. [Online]. Available: https://doi.org/10.1007/978-3-319-47075-7_10.

[118] K. Fysarakis, I. Papaefstathiou, C. Manifavas, K. Rantos, and O. Sultatos, "Policy-based access control for dpws-enabled ubiquitous devices", in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, Sep. 2014, pp. 1–8. DOI: 10.1109/ETFA.2014.7005233.

[119] P. Fremantle and B. Aziz, "Cloud-based federated identity for the internet of things", *Annals of Telecommunications*, vol. 73, no. 7, pp. 415–427, Aug. 2018, ISSN: 1958-9395. DOI: 10.1007/s12243-018-0641-8. [Online]. Available: https://doi.org/10.1007/s12243-018-0641-8.

[120] D. Hussein, E. Bertin, and V. Frey, "A community-driven access control approach in distributed iot environments", *IEEE Communications Magazine*, vol. 55, no. 3, pp. 146–153, Mar. 2017, ISSN: 0163-6804. DOI: 10.1109/MCOM.2017.1600611CM.

[121] V. Sivaraman, H. H. Gharakheili, A. Vishwanath, R. Boreli, and O. Mehani, "Network-level security and privacy control for smart-home iot devices", in *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct. 2015, pp. 163–167. DOI: 10.1109/WiMOB.2015.7347956.

[122] M. Poulymenopoulou, F. Malamateniou, and G. Vassilacopoulos, "A virtual phr authorization system", in *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, Jun. 2014, pp. 73–76. DOI: 10.1109/BHI.2014.6864307.

[123] J. Wilson, R. S. Wahby, H. Corrigan-Gibbs, D. Boneh, P. Levis, and K. Winstein, "Trust but verify: Auditing the secure internet of things", in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '17, Niagara Falls, New York, USA: ACM, 2017, pp. 464–474, ISBN: 978-1-4503-4928-4. DOI: 10.1145/3081333.3081342. [Online]. Available: http://doi.acm.org/10.1145/3081333.3081342.

[124] I. B. .-. Pasquier, A. A. Ouahman, A. A. E. Kalam, and M. O. de Montfort, "Smartorbac security and privacy in the internet of things", in *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, Nov. 2015, pp. 1–8. DOI: 10.1109/AICCSA.2015.7507098.

[125] B. Gong, Y. Wang, X. Liu, F. Qi, and Z. Sun, "A trusted attestation mechanism for the sensing nodes of internet of things based on dynamic trusted measurement", *China Communications*, vol. 15, no. 2, pp. 100–121, 2018. DOI: 10.1109/CC.2018.8300276.

[126] H.-C. Chen, "Collaboration iot-based rbac with trust evaluation algorithm model for massive iot integrated application", *Mobile Networks and Applications*, vol. 24, no. 3, pp. 839–852, Jun. 2019, ISSN: 1572-8153. DOI: 10.1007/s11036-018-1085-0. [Online]. Available: https://doi.org/10.1007/s11036-018-1085-0.

[127] C.-Y. Chen, "Efficient authentication for tiered internet of things networks", in *Quality, Reliability, Security and Robustness in Heterogeneous Networks: 12th International Conference, QShine 2016, Seoul, Korea, July 7–8, 2016, Proceedings*. Cham: Springer International Publishing, 2017, pp. 469–472, ISBN: 978-3-319-60717-7. DOI: 10.1007/978-3-319-60717-7_46. [Online]. Available: https://doi.org/10.1007/978-3-319-60717-7_46.

[128] H. Ren, Y. Song, S. Yang, and F. Situ, "Secure smart home: A voiceprint and internet based authentication system for remote accessing", in *2016 11th International Conference on Computer Science Education (ICCSE)*, Aug. 2016, pp. 247–251. DOI: 10.1109/ICCSE.2016.7581588.

[129] N. Pohlmann, M. Hertlein, and P. Manaras, "Bring your own device for authentication (byod4a) – the xign–system", in *ISSE 2015: Highlights of the Information Security Solutions Europe 2015 Conference*, H. Reimer, N. Pohlmann, and W. Schneider, Eds. Wiesbaden: Springer Fachmedien Wiesbaden, 2015, pp. 240–250, ISBN: 978-3-658-10934-9. DOI: 10.1007/978-3-658-10934-9_20. [Online]. Available: https://doi.org/10.1007/978-3-658-10934-9_20.

[130] A. Bassi, M. Bauer, M. Fiedler, T. Kramp, R. van Kranenburg, S. Lange, and S. Meissner, *Enabling Things to Talk: Designing IoT Solutions with the IoT Architectural Reference Model*, 1st. Springer Publishing Company, Incorporated, 2016, ISBN: 3662524945, 9783662524947.

[131] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "Extensible authentication protocol (eap)", RFC Editor, RFC 3748, Jun. 2004, http://www.rfc-editor.org/rfc/rfc3748.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3748.txt.

[132] R. T. Fielding, "Architectural styles and the design of network-based software architectures", in University of California, 2000, ch. Representational State Transfer (REST).

[133] E. Rescorla, "Http over tls", RFC Editor, RFC 2818, May 2000, http://www.rfc-editor.org/rfc/rfc2818.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2818.txt.

[134] A. W. Roscoe, *The Theory and Practice of Concurrency*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1997, ISBN: 0136744095.

[135] J. Camenisch and E. Van Herreweghen, "Design and implementation of the idemix anonymous credential system", in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, ser. CCS '02, Washington, DC, USA: ACM, 2002, pp. 21–30, ISBN: 1-58113-612-9. DOI: 10.1145/586110.586114. [Online]. Available: http://doi.acm.org/10.1145/586110.586114.

[136] S. Cirani, M. Picone, P. Gonizzi, L. Veltri, and G. Ferrari, "Iot-oas: An oauth-based authorization service architecture for secure services in iot scenarios", *IEEE Sensors Journal*, vol. 15, no. 2, pp. 1224–1234, Feb. 2015, ISSN: 1530-437X.

[137] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption", in *2007 IEEE Symposium on Security and Privacy (SP '07)*, 2007, pp. 321–334. DOI: 10.1109/SP.2007.11.

[138] D. Boneh and X. Boyen, "Efficient selective-id secure identity-based encryption without random oracles", in *Advances in Cryptology - EUROCRYPT 2004*, C. Cachin and J. L. Camenisch, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 223–238, ISBN: 978-3-540-24676-3.

[139] *The extensible access control markup language (xacml) version 3.0*, OASIS Standard, 2017. [Online]. Available: http://openid.net/developers/specs/.

[140] D. Hardt, *The OAuth 2.0 Authorization Framework*, RFC 6749 (Proposed Standard), Internet Engineering Task Force, Oct. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6749.txt.

[141] M. Jones, J. Bradley, and N. Sakimura, *JSON Web Token (JWT)*, RFC 7519 (Proposed Standard), Updated by RFC 7797, Internet Engineering Task Force, May 2015. [Online]. Available: http://www.ietf.org/rfc/rfc7519.txt.

[142] A. A. E. Kalam, R. E. Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miege, C. Saurel, and G. Trouessin, "Organization based access control", in *Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, Jun. 2003, pp. 120–131. DOI: 10.1109/POLICY.2003.1206966.

[143] *Astm f2761-09(2013), medical devices and medical systems - essential safety requirements for equipment comprising the patient-centric integrated clinical environment (ice) - part 1: General requirements and conceptual model*, ASTM International, West Conshohocken, PA, 2009. [Online]. Available: www.astm.org.

[144] J. Hatcliff, A. King, I. Lee, A. Macdonald, A. Fernando, M. Robkin, E. Vasserman, S. Weininger, and J. M. Goldman, "Rationale and architecture principles for medical application platforms", in *2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, Apr. 2012, pp. 3–12. DOI: 10.1109/ICCPS.2012.9.

[145] R. Lindemann, D. Baghdasaryan, and E. Tiffany, *Fido universal authentication framework protocol, version v1. 0-rd-20140209*, FIDO Alliance, Feb. 2014.

[146] *Web services security: Soap message security 1.1*, OASIS Standard, 2006. [Online]. Available: https://www.oasis-open.org/committees/wss/.

[147] B. C. Neuman, "Proxy-based authorization and accounting for distributed systems", in *[1993] Proceedings. The 13th International Conference on Distributed Computing Systems*, May 1993, pp. 283–291. DOI: 10.1109/ICDCS.1993.287698.

[148] *Google scholar*, https://scholar.google.com/.

[149] G. E, "The history and meaning of the journal impact factor", *JAMA*, vol. 295, no. 1, pp. 90–93, 2006. DOI: 10.1001/jama.295.1.90. eprint: /data/journals/jama/5006/jco50055.pdf. [Online]. Available: +%20http://dx.doi.org/10.1001/jama.295.1.90.

[150] *Core conference ranking*, http://portal.core.edu.au/conf-ranks/.

[151] R. L. Lawrence and A. Wright, "Rule-based classification systems using classification and regression tree (cart) analysis", *Photogrammetric engineering and remote sensing*, vol. 67, no. 10, pp. 1137–1142, 2001.

[152] K. Nozaki, H. Ishibuchi, and H. Tanaka, "Adaptive fuzzy rule-based classification systems", *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 238–250, Aug. 1996, ISSN: 1063-6706. DOI: 10.1109/91.531768.

[153] "Information technology – open systems interconnection – basic reference model: Naming and addressing", International Organization for Standardization and the International Electrotechnical Commission, Geneva, Switzerland, ISO/EIC 7498-1:1997, 1997.

[154] L. Oliveira, J. Rodrigues, S. Kozlov, R. Rabêlo, and V. Albuquerque, "Mac layer protocols for internet of things: A survey", *Future Internet*, vol. 11, no. 1, p. 16, Jan. 2019, ISSN: 1999-5903. DOI: 10.3390/fi11010016. [Online]. Available: http://dx.doi.org/10.3390/fi11010016.

[155] *Picketlink*. [Online]. Available: http://picketlink.org/.

[156] *Keycloak*. [Online]. Available: https://www.keycloak.org/.

[157] *Java platform, enterprise edition (java ee) 7*. [Online]. Available: https://docs.oracle.com/javaee/7/index.html.

[158] P. Tarr, H. Ossher, W. Harrison, and S. M. Sutton, "n degrees of separation: Multi-dimensional separation of concerns", in *Proceedings of the 21st International Conference on Software Engineering*, ser. ICSE '99, Los Angeles, California, USA: Association for Computing Machinery, 1999, pp. 107–119, ISBN: 1581130740. DOI: 10.1145/302405.302457. [Online]. Available: https://doi.org/10.1145/302405.302457.

[159] W. P. Stevens, G. J. Myers, and L. L. Constantine, "Structured design", *IBM Systems Journal*, vol. 13, no. 2, pp. 115–139, 1974. DOI: 10.1147/sj.132.0115.

[160] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things", *Computer Networks*, vol. 57, no. 10, pp. 2266–2279, 2013, Towards a Science of Cyber SecuritySecurity and Identity Architecture for the Future Internet, ISSN: 1389-1286. DOI: http://dx.doi.org/10.1016/j.comnet.2012.12.018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128613000054.

[161] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the internet of things: Perspectives and challenges", *Wireless Networks*, vol. 20, no. 8, pp. 2481–2501, 2014, ISSN: 1572-8196. DOI: 10.1007/s11276-014-0761-7. [Online]. Available: http://dx.doi.org/10.1007/s11276-014-0761-7.

[162] R. Fielding and J. Reschke, *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*, RFC 7231 (Proposed Standard), Internet Engineering Task Force, Jun. 2014. [Online]. Available: http://www.ietf.org/rfc/rfc7231.txt.

[163] A. Banks, E. Briggs, K. Borgendale, and R. Gupta, "MQTT Version 5.0", OASIS, Standard, Mar. 2019. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html.

[164] E. Rescorla, "The transport layer security (tls) protocol version 1.3", RFC Editor, RFC 8446, Aug. 2018.

[165] A. Freier, P. Karlton, and P. Kocher, *The Secure Sockets Layer (SSL) Protocol Version 3.0*, RFC 6101 (Historic), Internet Engineering Task Force, Aug. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6101.txt.

[166] A. Finkelsteiin and J. Kramer, "Software engineering: A roadmap", in *Proceedings of the Conference on The Future of Software Engineering*, ser. ICSE '00, Limerick, Ireland: ACM, 2000, pp. 3–22, ISBN: 1-58113-253-0. DOI: 10.1145/336512.336519. [Online]. Available: http://doi.acm.org/10.1145/336512.336519.

[167] *Shibboleth*. [Online]. Available: https://www.shibboleth.net/.

[168] M. Jones and D. Hardt, "The oauth 2.0 authorization framework: Bearer token usage", RFC Editor, RFC 6750, Oct. 2012, http://www.rfc-editor.org/rfc/rfc6750.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6750.txt.

Scientific results of author

This section shows publications and a list of selected citations of the author of this dissertation. Where applicable, I list the Impact Factor (2020) or CORE2020 ranking and citation count. All the citations were obtained from the Google Scholar database on May 8th, 2022, and exclude auto citations

Awards

During my studies, I received a Fulbright scholarship, which allowed me to spend a fruitful year full of exciting research at Baylor University in Waco, Texas, USA.

Related Publications

Journals with Impact Factor

[A.1] M. Trnka, J. Svacina, T. Cerny, E. Song, J. Hong, and M. Bures, "Securing internet of things devices using the network context", *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4017–4027, 2020, ISSN: 1551-3203. DOI: 10.1109/TII.2019.2954100 (IF: 10.22, Citations: 6, Contribution: 50%)
Selected citation: [B.9]

[A.2] M. Trnka, A. S. Abdelfattah, A. Shrestha, M. Coffey, and T. Cerny, "Systematic review of authentication and authorization advancements for the internet of things", *Sensors*, vol. 22, no. 4, 2022, ISSN: 1424-8220. DOI: 10.3390/s22041361 (IF: 3.58, Citations: 1)
Selected citation: [B.21]

[A.3] M. Trnka, T. Cerny, and N. Stickney, "Survey of authentication and authorization for the internet of things", *Security and Communication Networks*, vol. 2018, pp. 1–17, 2018, ISSN: 1939-0114. DOI: 10.1155/2018/4351603 (IF: 1.79, Citations: 51, Contribution: 90%)
Selected citations: [B.2], [B.4], [B.7], [B.8], [B.12], [B.13], [B.17], [B.20], [B.22], [B.24], [B.25], [B.29], [B.30], [B.34], [B.35]

[A.4] M. Klima, M. Bures, K. Frajtak, V. Rechtberger, M. Trnka, X. Bellekens, T. Cerny, and B. S. Ahmed, "Selected code-quality characteristics and metrics for internet of things systems", *IEEE Access*, vol. 10, pp. 46 144–46 161, 2022. DOI: 10.1109/ACCESS.2022.3170475 (IF: 3.37)

Other peer reviewed journals

[A.5] M. Trnka and T. Cerny, "Authentication and authorization rules sharing for internet of things", *Software Networking*, vol. 2018, no. 1, pp. 35–52, 2018, ISSN: 2445-9739. DOI: 10.13052/jsn2445-9739.2017.003 (Citations: 6)
Selected citation: [B.28]

In proceedings indexed in ISI

[A.6] M. Trnka and T. Cerny, "Identity management of devices in internet of things environment", in *2016 6th International Conference on IT Convergence and Security (ICITCS)*, 2016, pp. 1–4. DOI: 10.1109/ICITCS.2016.7740343 (Citations: 23)
Selected citation: [B.5], [B.10], [B.11], [B.14], [B.23], [B.26]

[A.7] M. Trnka, M. Tomasek, and T. Cerny, "Context-aware security using internet of things devices", in *Information Science and Applications 2017*, 2017, pp. 706–713, ISBN: 978-981-10-4154-9. DOI: 10.1007/978-981-10-4154-9_81 (Citations: 5)

[A.8] M. Trnka, F. Rysavy, T. Cerny, and N. Stickney, "Using wi-fi enabled internet of things devices for context-aware authentication", in *Information Science and Applications 2018*, 2019, pp. 635–642, ISBN: 978-981-13-1056-0. DOI: 10.1007/978-981-13-1056-0_62 (Citations: 3)
Selected citation: [B.27]

[A.9] T. Cerny, M. Trnka, and M. J. Donahoo, "Towards shared security through distributed separation of concerns", in *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, 2016, pp. 169–172, ISBN: 9781450344555. DOI: 10.1145/2987386.2987394

Other proceedings

[A.10] M. Trnka and T. Cerny, "On security level usage in context-aware role-based access control", in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, 2016, pp. 1192–1195, ISBN: 9781450337397. DOI: 10.1145/2851613.2851664 (CORE ranking: B, Citations: 32)
Selected citations: [B.1], [B.3], [B.6] [B.15], [B.16], [B.18], [B.19], [B.31], [B.32], [B.33]

[A.11] M. Trnka and T. Cerny, "Context-aware role-based access control using security levels", in *Proceedings of the 2015 Conference on Research in Adaptive and*

Convergent Systems, 2015, pp. 280–284, ISBN: 9781450337380. DOI: 10.1145/2811411.2811498 (Citations: 3)

[A.12] M. Trnka, J. Svacina, T. Cerny, and E. Song, "Aspect oriented context-aware and event-driven data processing for internet of things", in *Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems*, 2018, pp. 319–323, ISBN: 9781450358859. DOI: 10.1145/3264746.3264761

[A.13] M. Bures, B. S. Ahmed, V. Rechtberger, M. Klima, M. Trnka, M. Jaros, X. Bellekens, D. Almog, and P. Herout, "Patriot: Iot automated interoperability and integration testing framework", in *2021 IEEE 14th International Conference on Software Testing, Validation and Verification (ICST)*, 2021, pp. 454–459. DOI: 10.1109/ICST49551.2021.00059 (CORE ranking: A, Citation: 1)

Unrelated Publications

Other peer reviewed journals

[A.14] T. Cerny, M. J. Donahoo, and M. Trnka, "Contextual understanding of microservice architecture: Current and future directions", *SIGAPP Appl. Comput. Rev.*, vol. 17, no. 4, pp. 29–45, 2018, ISSN: 1559-6915. DOI: 10.1145/3183628.3183631 (Citations: 131, Citations with IF: 21)

In proceedings indexed in ISI

[A.15] J. Sebek, M. Trnka, and T. Cerny, "On aspect-oriented programming in adaptive user interfaces", in *2015 2nd International Conference on Information Science and Security (ICISS)*, 2015, pp. 1–5. DOI: 10.1109/ICISSEC.2015.7371024 (Citations: 3, Citations with IF: 1)

Selected Citations

Below I list selected citations from articles in journals with Impact Factor, referencing only publications related to the dissertation topic..

[B.1] A. Kesarwani and P. M. Khilar, "Development of trust based access control models using fuzzy logic in cloud computing", *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 5, pp. 1958–1967, 2022, ISSN: 1319-1578. DOI: https://doi.org/10.1016/j.jksuci.2019.11.001 (IF: 13.43)

[B.2] M. Barbareschi, A. D. Benedictis, E. L. Montagna, A. Mazzeo, and N. Mazzocca, "A puf-based mutual authentication scheme for cloud-edges iot systems", *Future Generation Computer Systems*, vol. 101, pp. 246–261, 2019, ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2019.06.012 (IF: 7.19)

[B.3] A. Kayes, W. Rahayu, T. Dillon, E. Chang, and J. Han, "Context-aware access control with imprecise context characterization for cloud-based data resources", *Future Generation Computer Systems*, vol. 93, pp. 237–255, 2019, ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2018.10.036 (IF: 7.19)

[B.4] J.-P. A. Yaacoub, M. Noura, H. N. Noura, O. Salman, E. Yaacoub, R. Couturier, and A. Chehab, "Securing internet of medical things systems: Limitations, issues and recommendations", *Future Generation Computer Systems*, vol. 105, pp. 581–606, 2020, ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2019.12.028 (IF: 7.19)

[B.5] A. K. Das, S. Zeadally, and D. He, "Taxonomy and analysis of security protocols for internet of things", *Future Generation Computer Systems*, vol. 89, pp. 110–125, 2018, ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2018.06.027 (IF: 7.19)

[B.6] A. Kayes, W. Rahayu, P. Watters, M. Alazab, T. Dillon, and E. Chang, "Achieving security scalability and flexibility using fog-based context-aware access control", *Future Generation Computer Systems*, vol. 107, pp. 307–323, 2020, ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2020.02.001 (IF: 7.19)

[B.7] R. H. Aswathy and N. Malarvizhi, "A design of lightweight ecc based cryptographic algorithm coupled with linear congruential method for resource constraint area in iot", *Journal of Ambient Intelligence and Humanized Computing*, Jan. 2021, ISSN: 1868-5145. DOI: 10.1007/s12652-020-02788-0 (IF: 7.1)

[B.8] M. Mahbub, "Progressive researches on iot security: An exhaustive analysis from the perspective of protocols, vulnerabilities, and preemptive architectonics", *Journal of Network and Computer Applications*, p. 102 761, 2020, ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2020.102761 (IF: 6.28)

[B.9] G. Fortino, F. Messina, D. Rosaci, and G. M. L. Sarne, "Resiot: An iot social framework resilient to malicious activities", *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 5, pp. 1263–1278, 2020. DOI: 10.1109/JAS.2020.1003330 (IF: 6.17)

[B.10] P. R. Sousa, J. S. Resende, R. Martins, and L. Antunes, "The case for blockchain in IoT identity management", *Journal of Enterprise Information Management*, ISSN: 1741-0398. DOI: {10.1108/JEIM-07-2018-0148} (IF: 5.4)

[B.11] A. Čolaković and M. Hadžialić, "Internet of things (iot): A review of enabling technologies, challenges, and open research issues", *Computer Networks*, vol. 144, pp. 17–39, 2018, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2018.07.017 (IF: 4.47)

[B.12] W. Long, C. H. Wu, Y. P. Tsang, and Q. Chen, "An end-to-end bidirectional authentication system for pallet pooling management through blockchain internet of things (biot)", *Journal of Organizational and End User Computing*, vol. 33, no. 6, pp. 1–25, Nov. 2021, ISSN: 1546-2234. DOI: 10.4018/JOEUC.290349 (IF: 4.35)

[B.13] D. Das, S. C. Sethuraman, and S. C. Satapathy, "A decentralized open web cryptographic standard", *Computers and Electrical Engineering*, vol. 99, p. 107 751, 2022, ISSN: 0045-7906. DOI: https://doi.org/10.1016/j.compeleceng.2022.107751 (IF: 3.82)

[B.14] S. Wang, H. Li, J. Chen, J. Wang, and Y. Deng, "Dag blockchain-based lightweight authentication and authorization scheme for iot devices", *Journal of Information Security and Applications*, vol. 66, p. 103 134, 2022, ISSN: 2214-2126. DOI: https://doi.org/10.1016/j.jisa.2022.103134 (IF: 3.82)

[B.15] A. S. M. Kayes, R. Kalaria, I. H. Sarker, M. S. Islam, P. A. Watters, A. Ng, M. Hammoudeh, S. Badsha, and I. Kumara, "A survey of context-aware access control mechanisms for cloud and fog networks: Taxonomy and open research issues", *Sensors*, vol. 20, no. 9, p. 2464, Apr. 2020, ISSN: 1424-8220. DOI: 10.3390/s20092464 (IF: 3.58)

[B.16] J. Ji, G. Wu, J. Shuai, Z. Zhang, Z. Wang, and Y. Ren, "Heuristic approaches for enhancing the privacy of the leader in iot networks", *Sensors*, vol. 19, no. 18, p. 3886, Sep. 2019, ISSN: 1424-8220. DOI: 10.3390/s19183886 (IF: 3.58)

[B.17] P. Nespoli, M. Zago, A. Huertas Celdrán, M. Gil Pérez, F. Gómez Mármol, and F. J. García Clemente, "Palot: Profiling and authenticating users leveraging internet of things", *Sensors*, vol. 19, no. 12, p. 2832, Jun. 2019, ISSN: 1424-8220. DOI: 10.3390/s19122832 (IF: 3.58)

[B.18] Z.-Y. Wu, "A secure and efficient digital-data-sharing system for cloud environments", *Sensors*, vol. 19, no. 12, p. 2817, Jun. 2019, ISSN: 1424-8220. DOI: 10.3390/s19122817 (IF: 3.58)

[B.19] X. C. Yin, Z. G. Liu, B. Ndibanje, L. Nkenyereye, and S. M. Riazul Islam, "An iot-based anonymous function for security and privacy in healthcare sensor networks", *Sensors*, vol. 19, no. 14, p. 3146, Jul. 2019, ISSN: 1424-8220. DOI: 10.3390/s19143146 (IF: 3.58)

[B.20] S. P. Singh, N. B. Ali, and L. Lundberg, "Smart and adaptive architecture for a dedicated internet of things network comprised of diverse entities: A proposal and evaluation", *Sensors*, vol. 22, no. 8, 2022, ISSN: 1424-8220. DOI: 10.3390/s22083017 (IF: 3.58)

[B.21] C. Gupta, I. Johri, K. Srinivasan, Y.-C. Hu, S. M. Qaisar, and K.-Y. Huang, "A systematic review on machine learning and deep learning models for electronic information security in mobile networks", *Sensors*, vol. 22, no. 5, 2022, ISSN: 1424-8220. DOI: 10.3390/s22052017 (IF: 3.58)

[B.22] M. Akil, L. Islami, S. Fischer-Hübner, L. A. Martucci, and A. Zuccato, "Privacy-preserving identifiers for iot: A systematic literature review", *IEEE Access*, vol. 8, pp. 168 470–168 485, 2020 (IF: 3.37)

[B.23] R. Sardar and T. Anees, "Web of things: Security challenges and mechanisms", *IEEE Access*, pp. 1–1, 2021. DOI: 10.1109/ACCESS.2021.3057655 (IF: 3.37)

[B.24] R. F. Olanrewaju, B. U. I. Khan, M. A. Morshidi, F. Anwar, and M. L. B. M. Kiah, "A frictionless and secure user authentication in web-based premium applications", *IEEE Access*, vol. 9, pp. 129 240–129 255, 2021. DOI: 10.1109/ACCESS.2021.3110310 (IF: 3.37)

[B.25] S. V. Sudarsan, O. Schelén, and U. Bodin, "Survey on delegated and self-contained authorization techniques in cps and iot", *IEEE Access*, vol. 9, pp. 98 169–98 184, 2021. DOI: 10.1109/ACCESS.2021.3093327 (IF: 3.37)

[B.26] P. Ahmadi, K. Islam, T. Maco, and M. Katam, "A survey on internet of things security issues and applications", in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2018, pp. 925–934. DOI: 10.1109/CSCI46756.2018.00182 (IF: 3.37)

[B.27] D. M. A. D. Silva and R. C. Sofia, "A discussion on context-awareness to better support the iot cloud/edge continuum", *IEEE Access*, vol. 8, pp. 193 686–193 694, 2020. DOI: 10.1109/ACCESS.2020.3032388 (IF: 3.37)

[B.28] S. Pešić, M. Ivanović, M. Radovanović, and C. Bădică, "Caavi-rics model for observing the security of distributed iot and edge computing systems", *Simulation Modelling Practice and Theory*, p. 102 125, 2020, ISSN: 1569-190X. DOI: https://doi.org/10.1016/j.simpat.2020.102125 (IF: 3.27)

[B.29] M. Tahir, M. Sardaraz, S. Muhammad, and M. Saud Khan, "A lightweight authentication and authorization framework for blockchain-enabled iot network in health-informatics", *Sustainability*, vol. 12, no. 17, p. 6960, Aug. 2020, ISSN: 2071-1050. DOI: 10.3390/su12176960 (IF: 3.25)

[B.30] L. Guo, J. Wang, and W.-C. Yau, "Efficient hierarchical identity-based encryption system for internet of things infrastructure", *Symmetry*, vol. 11, no. 7, p. 913, Jul. 2019, ISSN: 2073-8994. DOI: 10.3390/sym11070913 (IF: 2.71)

[B.31] S. Magomedov, A. Gusev, D. Ilin, and E. Nikulchev, "Users' reaction time for improvement of security and access control in web services", *Applied Sciences*, vol. 11, no. 6, 2021, ISSN: 2076-3417. DOI: 10.3390/app11062561 (IF: 2.67)

[B.32] A. S. M. Kayes, W. Rahayu, and T. Dillon, "Critical situation management utilizing IoT-based data resources through dynamic contextual role modeling and activation", *Computing*, vol. 101, no. 7, SI, 743–772, Jul. 2019, ISSN: 0010-485X. DOI: 10.1007/s00607-018-0654-1 (IF: 2.22)

[B.33] A. S. M. Kayes, J. Han, W. Rahayu, T. Dillon, M. S. Islam, and A. Colman, "A Policy Model and Framework for Context-Aware Access Control to Information Resources", *The Computer Journal*, vol. 62, no. 5, pp. 670–705, Jul. 2018, ISSN: 0010-4620. DOI: 10.1093/comjnl/bxy065 (IF: 1.49)

[B.34] D. Berardi, S. Giallorenzo, J. Mauro, A. Melis, F. Montesi, and M. Prandini, "Microservice security: A systematic literature review", *PeerJ Computer Science*, vol. 8, e779, Jan. 2022, ISSN: 2376-5992. DOI: 10.7717/peerj-cs.779 (IF: 1.39)

[B.35] Z. Houhamdi and B. Athamena, "Identity identification and management in the internet of things", *The International Arab Journal of Information Technology*, vol. 17, pp. 645–654, Jul. 2020. DOI: 10.34028/iajit/17/4A/9 (IF: 0.67)