Fakultät für Informatik
Technische Universität München

Edge-Driven Proximity Service Platform for the Internet of Things in Indoor Environments

Michael Andreas Haus

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende:

Prof. Dr. Claudia Eckert

Prüfende der Dissertation:

1. Prof. Dr.-Ing. Jörg Ott
2. Associate Professor Dr. Hamed Haddadi,
   Imperial College London
3. Assistant Professor Dr. Aaron Yi Ding,
   Delft University of Technology

Die Dissertation wurde am 08.07.2020 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 17.03.2021 angenommen.

Acknowledgment

My PhD has been a challenging journey full of highs and lows. Especially in 2018 were most of my papers got rejected and I had to do a major readjustment, neglecting certain parts and focus on most important parts based on the paper reviews. Besides that, I had to improve my writing skills and learn how to present complicated technical approaches in a reduced way including crucial design decisions that somebody else is able to understand and assess the system solution.

I would like to express my sincere gratitude to my supervisor, Prof. Dr.-Ing. Jörg Ott, for his continuous support, patience, motivation, time and guidance. A special thanks to my co-advisor Prof. Dr. Aaron Yi Ding, without his support and joint brainstorming, that work would never been possible. Moreover, I want to thank the co-authors of my papers and the anonymous reviewers of my papers for their valuable and constructive feedback to improve my work. Furthermore, I appreciate discussion with other peers in the community during scientific events and my colleagues providing an international working environment at the chair of Connected Mobility which also helped me a lot to improve my English skills. Finally, I would like to thank my parents for their understanding and much-needed encouragement throughout my PhD.

Abstract

The evolution of the Internet started in the late 1960s up to the Internet of Things (IoT) connecting not only people but also objects around us with the goal that the things know what we like, what we want, and what we need. What will be the platform that supports such a vision? In this dissertation, towards a future IoT platform, we propose our EdgeProx platform which consists of two key building blocks for indoor IoT environments. Firstly, the goal of the iPresence system module is that objects become networked based on ubiquitous data exchange through proximity wireless technologies. We use visible light for communication/signaling and introduce a 3D printed custom light bulb with a dedicated modulation scheme for visible light communication (VLC) inspired by Morse coding. Our lighting configuration framework integrates the custom light bulbs into a network for communication and signaling. We can flexibly define different groups of light bulb(s) to form semantic subnetworks covering different areas and controlling the spatial granularity of users and devices. Besides that, future IoT environments will reach such a scale that every human interaction for the device management process will become untenable. The aim of iConfig as our second system module is to enable fully self-maintained IoT environments by the automated management of heterogeneous devices. iConfig provides a global device map for administration and multiple edge modules intended to run on user devices, e.g., wearables, to interact with physically nearby IoT devices. We use an edge module dedicated for drones being completely independent from users to further reduce the operational costs of IoT device management. On this basis, iService takes advantage of our EdgeProx platform and demonstrates its usefulness by realizing several proximity services: fine-granular device association, private service discovery, automated authentication for wireless networks, and automated authorization for smart homes. Based on VLC embedded in everyday environments from iPresence, the aim of iService is to improve the user experience by automating tedious and reoccurring user tasks, e.g., authentication and authorization, and integrate social applications, e.g., data sharing, in the surrounding user environment. iConfig supports the well-functioning of the proximity services regarding a high service reliability by identifying and localizing broken IoT devices to replace them as soon as possible.

Zusammenfassung

Die Entwicklung des Internets begann in den späten 1960er Jahren bis hin zum Internet der Dinge (Internet of Things, IoT), das nicht nur Menschen, sondern auch Objekte miteinander verbindet, mit dem Ziel, dass die Dinge wissen, was wir mögen, was wir wollen und was wir brauchen. Wie wird die Plattform aussehen, die solch eine Zukunft unterstützt? In dieser Dissertation, hinsichtlich einer zukünftigen IoT-Plattform, führen wir als Ansatz unsere EdgeProx-Plattform ein, bestehend aus zwei wichtigen Bausteinen für IoT-Umgebungen in Innenräumen. Das Ziel des iPresence-Systemmoduls besteht darin, Objekte mittels drahtlosen Nahbereichstechnologien für den einfachen Datenaustausch zu vernetzen. Wir verwenden sichtbares Licht für die Kommunikation/-Signale und nutzen eine maßgeschneiderte 3D-gedruckte Glühbirne mit einem speziellen Daten-Modulationsschema inspiriert durch Morse-Kodierung für die Kommunikation mit sichtbarem Licht (Visible Light Communication, VLC). Zudem integriert unser Rahmenwerk die maßgeschneiderten Glühbirnen in ein drahtloses Netz zur Konfiguration von Beleuchtungsquellen hinsichtlich Kommunikation und Signalmustern. Wir können verschiedene Gruppen von Glühbirnen flexibel definieren, um semantische Teilnetze zu bilden, die verschiedene geografische Bereiche abdecken und die Granularität der räumlichen Nähe von Benutzern und Geräten steuert. Darüber hinaus werden zukünftige IoT-Umgebungen eine solche Größe erreichen, dass menschliche Interaktionen für den Geräteverwaltungsprozess zu aufwändig wird. Das Ziel von iConfig als unser zweites Systemmodul ist die vollständig selbstverwaltete IoT-Umgebung durch das automatisierte Management einer großen Zahl heterogener Geräte. iConfig bietet zur Verwaltung eine globale Karte von Geräten in Innenräumen und mehrere Softwaremodule, die auf Benutzergeräten, z.B. Wearables, laufen, um mit physisch nahe gelegenen IoT-Geräten zu interagieren. Wir verwenden Drohnen für eine Geräteverwaltung, die völlig unabhängig von Benutzern funktioniert, um die Betriebskosten für die Verwaltung von IoT-Geräten weiter zu senken. Auf dieser Basis nutzt iService die Vorteile unserer EdgeProx-Plattform und demonstriert ihre Nützlichkeit durch die Realisierung mehrerer nähebasierender Dienste. Die Dienste umfassen feingranulare Gerätegruppierung, private Diensterkennung, automatische Authentifizierung für drahtlose Netze und die Autorisierung für in-

telligente Häuser. Durch die VLC-Kommunikation eingebettet in alltäglichen Umgebungen von iPresence, ist es unser Ziel, die Benutzererfahrung durch die Automatisierung langwieriger und wiederkehrender Benutzeraufgaben, z.B. Authentifizierung und Autorisierung, zu verbessern und soziale Dienste, z.B. Datenaustausch, in die Umgebung zu integrieren. Um eine hohe Servicezuverlässigkeit zu erreichen unterstützt iConfig die reibungsfreie Funktionsweise der nähebasierten Dienste durch die Identifizierung und Lokalisierung defekter IoT-Geräte und deren rascher Austausch.

Contents

List of Abbreviations

AMPPM	Adaptive Multiple Pulse Position Modulation
API	Application Programming Interface
AUC	Area Under the Curve
BLE	Bluetooth Low Energy
BYOIoT	Bring Your Own Internet of Things
COTS	Commercial Off-The-Shelf
D2D	Device-to-Device
DAC	Discretionary Access Control
DTW	Dynamic Time Warping
FoV	Field of View
FPGA	Field Programmable Gate Array
GPIO	General-Purpose Input/Output
IaaS	Infrastructure as a Service
IoT	Internet of Things
ISI	Inter-Symbol Interference
LBS	Location-Based Service
LED	Light-Emitting Diode
M2M	Machine-to-Machine
MAC	Mandatory Access Control
MANET	Mobile Ad hoc Network
MEMS	Micro-Electro-Mechanical Systems

MPTCP	Multipath TCP
MTC	Machine Type Communication
NFC	Near-Field Communication
NUC	Next Unit of Computing
OFDM-OOK	Orthogonal Frequency-Division Multiplexed On-Off Keying
OOK	On-Off-Keying
PaaS	Platform as a Service
PBS	Proximity-Based Service
PET	Private Equality Testing
PPT	Private Proximity Testing
PSI	Private Set Intersection
RBAC	Role-Based Access Control
RFID	Radio-Frequency Identification
RSSI	Received Signal Strength Indication
SaaS	Software as a Service
SMC	Secure Multiparty Computation
STFT	Short Time Fourier Transform
SVM	Support Vector Machine
TOTP	Time-Based One-Time Password
TP	Trusted Party
USC	Ultrasound Communication
UWB	Ultra-Wide Band
VLC	Visible Light Communication
VLS	Visible Light Signaling
WSN	Wireless Sensor Network

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their numeral. All publications are subject to a full peer-review process.

1. **M. Haus**, M. Waqas, A. Y. Ding, Y. Li, S. Tarkoma, and J. Ott. Security and Privacy in Device-to-Device (D2D) Communication: A Review. *IEEE Communications Surveys & Tutorials*, 19(2):1054–1079, 2017. doi:10.1109/COMST.2017.2649687

2. **M. Haus**, A. Y. Ding, and J. Ott. LocalVLC: Augmenting Smart IoT Services with Practical Visible Light Communication. In *Proceedings of the 20th IEEE International Symposium on 'A World of Wireless, Mobile and Multimedia Networks' (WoWMoM)*, pages 1–9, 2019. doi:10.1109/wowmom.2019.8793022

3. **M. Haus**, J. Ott, and A. Y. Ding. DevLoc: Seamless Device Association using Light Bulb Networks for Indoor IoT Environments. In *Proceedings of the Fifth IEEE/ACM International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 231–237, 2020. doi:10.1109/IoTDI49375.2020.00030

4. **M. Haus**, A. Y. Ding, and J. Ott. Multimodal Co-Presence Detection with Varying Spatio-Temporal Granularity. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 1–7, 2020. doi:10.1109/PerComWorkshops48775.2020.9156105

5. **M. Haus**, A. Y. Ding, Q. Wang, J. Toivonen, L. Tonetto, S. Tarkoma, and J. Ott. Enhancing Indoor IoT Communication with Visible Light and Ultrasound. In *Proceedings of the 53rd IEEE International Conference on Communications (ICC)*, pages 1–6, 2019. doi:10.1109/icc.2019.8762001

6. **M. Haus**, A. Y. Ding, and J. Ott. Managing IoT at the Edge: The Case for BLE Beacons. In *Proceedings of the 3rd ACM MobiCom Workshop on Experiences with the Design and Implementation of Smart Objects (SmartObjects)*, pages 41–46, 2017. doi:10.1145/3127502.3127510

7. **M. Haus**, J. Krol, A. Y. Ding, and J. Ott. Feasibility Study of Autonomous Drone-based IoT Device Management in Indoor Environments. In *Proceedings of the 1st ACM SIGCOMM Workshop on Mobile AirGround Edge Computing, Systems, Networks, and Applications (MAGESys)*, pages 1–7, 2019. doi:10.1145/3341568.3342105

Author's Contribution

Publication 1: 'Security and Privacy in Device-to-Device (D2D) Communication: A Review'

The paper idea originated from me, Aaron Yi Ding, and Yong Li. I have provided the introduction identifying the unique properties of D2D communication, and more details about security and privacy in D2D including their requirements, relations, and the attack and threat model. As one of the two main parts, I analyzed and classified privacy-preserving approaches for D2D communication. Our co-author Muhammad Waqas provided the other main part, the security solutions for D2D communication. The discussion section was a shared work between me and Aaron Yi Ding to identify best practices, lessons learned, and open issues. I co-edited the paper with Aaron Yi Ding and Jörg Ott.

Publication 2: 'LocalVLC: Augmenting Smart IoT Services with Practical Visible Light Communication'

I came up with the idea for the paper as foundation for proximity-based services. I have designed, implemented, and evaluated the entire system including the two exemplary use cases. Finally, I co-edited the paper with Aaron Yi Ding and Jörg Ott.

Publication 3: 'DevLoc: Seamless Device Association using Light Bulb Networks for Indoor IoT Environments'

I came up with the idea for the paper, and designed and realized the custom light bulb. Thereby, we used the hardware platform from the openVLC project (`http://www.openvlc.org` [last checked 16.06.2020]) for visible light communication and Oleksii Moroz designed and created the 3D case for our custom light bulb. On this basis, I have implemented the entire DevLoc system, performed the evaluation, and interpreted the results. I co-edited the paper with Jörg Ott and Aaron Yi Ding.

Publication 4: 'Multimodal Co-Presence Detection with Varying Spatio-Temporal Granularity'

I came up with the idea for the paper and setup the data collection framework to gather the required sensor data from mobile devices. Moreover, I have implemented the entire data analysis and interpreted the results. I co-edited the paper with Aaron Yi Ding and Jörg Ott.

Publication 5: 'Enhancing Indoor IoT Communication with Visible Light and Ultrasound'

The original idea for the paper came from me and Qing Wang. I have further specified and refined the paper content. The analysis of user's mobility pattern was a joint effort, the idea came from me and Leonardo Tonetto performed the analysis. Qing Wang provided the visible light communication module including all related measurements and Juhani Toivonen provided the ultrasound communication module including all related measurements. Regarding secure IoT group communication, I have provided the idea and specification for the mobile device grouping, the implementation and evaluation of the prototype came from Jimmy Abu Al Denien, and I proposed the platform for infrastructure-supported device grouping. Finally, I co-edited the paper with Aaron Yi Ding, Qing Wang, Juhani Toivonen, and Jörg Ott.

Publication 6: 'Managing IoT at the Edge: The Case for BLE Beacons'

I came up with the idea for the paper, inspired by the usage of an existing IoT platform and involved shortcomings. I have designed and implemented the entire iConfig system which resulted in a working prototype. I co-edited the paper with Aaron Yi Ding and Jörg Ott.

Publication 7: 'Feasibility Study of Autonomous Drone-based IoT Device Management in Indoor Environments'

I came up with the idea for the paper to extend our existing iConfig platform regarding a fully autonomous operation. I have designed the drone system, implemented the

integration for the iConfig framework, and interpreted the evaluation results. Jan Krol implemented and evaluated the edge module dedicated for drones. I co-edited the paper with Aaron Yi Ding and Jörg Ott.

1 Introduction

The evolution of the Internet begins in the late 1960s with connecting two computers together and then moved towards creating the World Wide Web, available in 1991, connecting a large number of computers together [8]. Later, mobile devices connected to the Internet and formed the mobile Internet. With the rise of social networks, users started to become connected together over the Internet. Finally, it is moving towards the Internet of Things (IoT), a term first coined in 1999, a network that not only connects people, but also the objects around them [9]. The ultimate goal of the IoT is to create a better world for human beings, where objects around us know what we like, what we want, and what we need and act accordingly without explicit instructions [8, 10]. The IoT covers various aspects related to the extension of the Internet and Web into the physical realm, where people and things can be connected anytime, anyplace, with anything and anyone, ideally using any path/network and any service [11, 8]. The three main system-level characteristics of the IoT are [11]: 1) anything communicates wirelessly among themselves and forms ad hoc networks of interconnected objects, 2) anything is identifiable with a digital name being able to specify relationships among things, and 3) anything interacts with the local environment through sensing and actuation. IoT is a technology architecture bundling different technologies together to do something new with a high impact on several aspects of everyday life [8]. Hence, content and services will be always around us and available enabling new applications to support new ways of working, interaction, entertainment, and living [11]. Nowadays, sensing and actuation coverage is still in early stages of development, we have a fragmented environment with IoT islands, e.g., buildings have sensors to save energy, all of these are just the tip of the iceberg [12]. The steady increase of sensing devices will lead to a significant qualitative change in how we work and live caused by systems-of-systems that synergistically interact to form totally new and unpredictable services. What will be the platform or platforms that support such a vision?

A large number of IoT applications will arise across many areas of life due to the enormous amount and variety of data generated by IoT devices [13]. For instance, the data volume of IoT connections worldwide will increase from 13.6 zettabytes in 2018 up

to 79.4 zettabytes in 2025, a compound annual growth rate of 28.7 % [14]. Moreover, it is forecasted that the industrial IoT market will grow from a value of 77.3 billion U.S. dollars in 2020 by 43 % up to 110.6 billion U.S. dollars in 2025 [15]. The IoT application domain is mainly divided into three categories: industry, environment, and society [16]. For example, *industry-focused IoT applications*: supply chain management, transportation and logistics, aerospace, aviation, and automotive, *society-focused IoT applications*: telecommunication, medical technology, healthcare, smart building, home and office, media, entertainment, and ticketing, and *environment-focused IoT applications*: agriculture and breeding, recycling, disaster alerting, and environmental monitoring.

The vision of 'anytime, anywhere, anymedia' communications has been pushing the advances in wireless technologies for a long time. Adding 'anything' to the above vision leads to the IoT concept, which is supported by the advancements of micro-electromechanical systems (MEMS) technology, wireless communications, and digital electronics [16]. This trend enabled powerful sensors with reduced size, weight, energy consumption, and cost to equip objects of everyday life with micro-controllers and transceivers for communication, sensing, and computation [17, 13]. Emphasized due to their self-configuring capabilities enabling large scale IoT deployments, the three most important foundational technologies for the IoT are radio-frequency identification (RFID), wireless sensor networks (WSNs), and mobile ad hoc networks (MANETs) [18]. Those enable ubiquitous sensing across many areas of modern day living to measure, infer, and understand environmental indicators, combined with a communicating–actuating network leading to the IoT vision [17].

Due to the heterogeneous field of applications enabled by the IoT, it is a challenge to build a platform capable to satisfy the requirements of all possible application scenarios [13]. A special challenge is that most of the IoT services are to be delivered to mobile users, i.e., connect them with their desired services continuously while on the move [10]. To support new IoT applications in an easy manner, it is important that middleware, frameworks, and application programming interfaces (APIs) provide generic and reusable functionalities, such as intelligence, semantic interoperability, context-awareness [8]. The middleware is fundamental to hide details of different technologies and integrate legacy technologies into new ones, so that developers can simply use the functionality to implement an application, without knowing details about the underlying technologies [16]. Towards a practical platform for the IoT, the following eight key points need to be considered [11]:

- management of a large heterogeneity of devices taking part in the system with different computation and communication capabilities,

- scalability issues arise at different levels as everyday objects get connected to a global information infrastructure, including: naming and addressing, data communication and networking, information and knowledge management, and service provisioning and management,
- ubiquitous data exchange through proximity wireless technologies to enable objects to become networked, which may pose issues in terms of spectrum availability,
- energy-optimized solutions are a primary goal for a variety of IoT entities to minimize the consumed energy for communication and computing purposes,
- object localization and tracking so that entities can be identified and are provided with short-range wireless communication capabilities to finally track object locations in the physical realm,
- self-organization capabilities to minimize human intervention by distributing intelligence in the system, i.e., IoT nodes are able to autonomously organize themselves into ad hoc networks, including device and service discovery, providing the basic means for data sharing and coordinated tasks,
- semantic interoperability and data management to exchange and analyze massive amounts of data turning them into meaningful information for automated reasoning, and
- embedded security and privacy-preserving mechanisms are a key requirement for the acceptance by users, due to the close integration with the physical environment, IoT technology should be secure and privacy-preserving by design.

We envision an incremental IoT development along which IoT technologies will gradually be deployed to equip existing systems and applications with additional information and communication capabilities. In this thesis, we explore proximity, being near to somebody or something [19], in different facets to enrich the IoT ecosystem and enable practical services in the IoT domain. We argue that proximity is a natural context among users and things which can be beneficial to enable self-organization of networks and ubiquitous data exchange. For instance, seamless device association among physically nearby devices for easy data sharing. Besides that, we explore mechanisms and tools towards automated IoT device management reducing operational costs within the device management process. We claim that future IoT environments will reach such a scale, that every human interaction within the management process will become untenable and hence we need mechanisms for fully self-maintained IoT environments.

1.1 Problem Statement

In this dissertation, we mainly deal with two different domains of future IoT platforms: 1) physical co-presence, i.e., proximity, among distributed entities for context-awareness and 2) management of spatially distributed, heterogeneous, wireless IoT devices. On this basis, we explore different platform services taking advantage of IoT devices which are physically nearby and autonomously managed.

We approach the current missing context-awareness of IoT middleware as an issue that limits the availability and usability of IoT services for individuals and industries. It is beneficial to integrate some sort of intelligence into the IoT platform based on context to provide relevant information and/or services where relevancy depends on the current situation. We argue that the current efforts for a practical solution of proximity wireless technologies for ubiquitous data exchange, as a key system-level feature of IoT platforms, are unilaterally focused on radio-based communication. To realize communication restricted to devices in proximity using electromagnetic waves, which easily penetrates physical barriers, we have to add an additional complexity layer to attain the distance limit. This means continuous time measurements of signal propagation to infer distances between sender and receiver whether somebody or something is nearby. There are several drawbacks: 1) precise time measurements require a resource-demanding time synchronization among distributed entities, and 2) due to the fast propagation of electromagnetic signals, the distance-bounding reasoning has to use narrow time windows to infer whether entities are nearby leading to wrong system estimates and higher user dissatisfaction. To overcome these problems, we take advantage of proximity, being near to somebody and/or something, to enable context-awareness based on an distance-limited ubiquitous data exchange for a future IoT platform. We argue that the non-radio based spectrum, such as visible light communication (VLC) and ultrasound communication (USC), can satisfy our requirements regarding a suitable proximity wireless technology for context-awareness. Moreover, we avoid interference with existing radio-based communication, like Wi-Fi, which usually causes a decreased communication performance. By using the non-radio based spectrum, if a communication is possible, we know implicitly that the involved entities are physically close-by which provides a beneficial context information for IoT services, e.g., data sharing. The natural distance-bounding feature of non-radio based communication is beneficial in many ways and not limited to IoT environments. For example, to improve user privacy and system security due to a physically restricted attack space, or using proximity as metric to self-organize networks, such as local social networks. Our aim is to enrich existing radio-based communication, as

primary communication means, with non-radio based communication in a smart way to benefit from the unique distance-bounding feature.

Besides that, we encounter a rapidly growing number of connected IoT devices, in 2025, around 37 billion connected devices are predicted, whereas about 25 billion (68 %) are related to the IoT [20]. Another challenge is the large heterogeneity of IoT devices as IoT environments are based on a multitude of devices, e.g., smartphones, sensors, embedded systems, smart meters, point-of-sale terminals, consumer electronics, and wearables. Each of these devices have their own purpose, only together they are able to satisfy all service requirements of an IoT environment. For instance, IoT boards serve as local gateways, collecting sensor data, and provide backend connectivity, whereas standalone Bluetooth low energy (BLE) beacons are cheap and simple to attach to many objects serving for object localization and tracking. In spite of the growing demand, we argue there is a lack of instruments to seamlessly manage large IoT deployments in which ad-hoc management is becoming untenable, especially for IoT devices without Internet connectivity and missing backend integration. It is important to recognize and integrate new IoT devices into the management backend as smooth as possible with a minimum of human intervention. The inconvenient manual registration of IoT devices at the management backend increases the risk that not all IoT devices will be registered or other errors occur in this process. As a result, we have an incomplete map of surrounding IoT devices, the lack of awareness about spatially distributed IoT assets limits the services that can be provided. Consequently, we believe that a streamlined management process is a key step for an appropriate IoT device management. There is a lack of a flexible system architecture using a centralized management backend, e.g., for device monitoring, and localized device actions for easy device integration. We have a special focus on the usability of edge devices for device registration with respect to an enhanced user interaction, more fluent and natural, with surrounding IoT devices.

The above considerations yield to three fundamental questions that form our research problem in this thesis:

- **RQ1: How to efficiently determine the spatial proximity of users without disclosing their locations?** (Related publications: 1, 2, 3, 4, and 5) We believe that the ability to recognize physically nearby users and/or objects is essential to enable different IoT services. In this thesis, we take advantage of non-radio based communication for context-awareness with a special emphasis on how to integrate the communication into everyday objects.

- **RQ2: How to effectively manage IoT deployments consisting of spatially distributed, heterogeneous, wireless IoT devices?** (Related publications: 6 and 7) The users and industry have recognized the potential of IoT initiated by a piecemeal implementation of the IoT vision based on the growing number of distributed IoT devices. Thereby, we explore how to manage the rapid and uncoordinated growth of the IoT. We need an appropriate system architecture to cope with the dynamic nature of IoT environments to limit the manual user interaction for device registration and maintenance.

- **RQ3: How to enable user-oriented and privacy-aware IoT services by utilizing spatial proximity awareness in the managed indoor IoT areas?** (Related publications: 2 and 3) We look at possibilities how to benefit from the proximity knowledge of the surrounding environment. For example, a data sharing service among physically close-by users to improve user privacy by restricted data access. Moreover, we pinpoint how the automated management of IoT devices supports high quality services.

1.2 Research Methodology

I have performed an extensive literature review in terms of the current state-of-the-art and open issues in the research field which needs to be addressed. Afterwards, I performed smaller literature reviews for certain subtopics, identified specific open issues, designed and realized system prototypes to highlight a possible solution addressing the real-world problem. I systematically evaluated the prototypes in multiple ways to show their usefulness and drawn our conclusions and findings based on measurements results. In this regard, I evaluated the system prototypes: 1) offline in smaller and controllable real-world testbeds to gain system measurements and I performed trace-driven simulations to evaluate the systems in a larger scale, which is hardly feasible in the real environment due to the required extensive efforts, and 2) online via user studies to gain insights with respect to human-computer-interaction and system improvements.

1.3 Approach

We provide two building blocks of key system-level features for a potential future IoT platform instrumenting indoor environments with IoT technologies to improve the satisfaction level of humans populating it.

Owing to the growing popularity of IoT and thereby the increasing number of IoT devices, we propose an edge-driven proximity service platform named EdgeProx: a) to take advantage of these IoT devices for user-oriented services and b) to address the challenges to manage these IoT devices. Our platform is targeted at indoor spaces because people spend their time mostly indoors [21] and further supported by statistics, in 2025, smart buildings and smart homes are key growth domains according to the number of new IoT connections [22]. EdgeProx follows two principles to enable user-oriented services: 1) deployable at low-cost by using off-the-shelf components, i.e., basic programmable boards with general-purpose input/output (GPIO) support and 2) a practical platform that imposes as little constraints as possible on typical indoor IoT usage, e.g., open-design with APIs for developers. EdgeProx takes advantage of device-to-device (D2D) communication as a new paradigm in mobile networking to facilitate data exchange between physically proximate devices. We identified two major models of D2D communication networks [1]: 1) standalone D2D without traversing fixed network infrastructures and 2) network-assisted D2D requiring infrastructure, such as base stations or access points, for communication organization and resource utilization. EdgeProx uses emerging network-assisted D2D communication to create a platform for proximity-based services [23] and addresses user privacy in D2D communication to protect user data as one of the six vulnerability categories of the security and privacy domain [1].

We evaluated visible light and ultrasound for indoor IoT communication to complement radio-based communication with proximity wireless technologies. Based on our insights, we have chosen visible light as medium for ubiquitous data exchange to achieve a reasonable data rate and energy trade-off (µJ/Byte), whereas ultrasound prototypes provide low bit rates and poor energy efficiency which greatly limits possible use cases [5]. In addition, visible light can effectively ward off the interference with existing radio-based communication, e.g., Bluetooth and Wi-Fi, and can serve as a visual feedback channel for users, such as changing the light color in a smart home to indicate a successful operation. EdgeProx takes advantage of ubiquitous light sources, e.g., widely used light-emitting diode (LED) lamps in residential and office settings, around us to achieve a minimal deployment overhead. We benefit from the unique distance bounding attribute of visible light in two different ways, in the usual manner as VLC and as visible light signaling (VLS) for specific tasks, such as device association among users and IoT devices based on the similarity of light signals.

In the following, we give a brief overview of our EdgeProx platform summarized in Fig. 1.1 including our two realized system modules named iPresence for co-presence

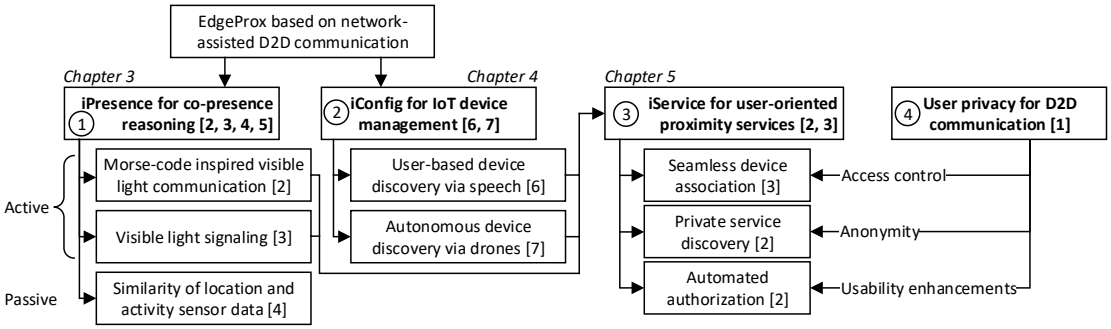


Figure 1.1: EdgeProx platform and corresponding publication(s) in brackets

reasoning and iConfig for IoT device management, and iService taking advantage of the two system modules.

Related to context-awareness based on ubiquitous data exchange, iPresence ① uses visible light for active co-presence reasoning including communication and signaling. Moreover, iPresence explores activity and location-related sensor data, e.g., magnetometer, for passive co-presence reasoning, to complement the active co-presence reasoning in certain situations. We mainly focus on the more reliable active co-presence reasoning where we implemented a Morse-code inspired data modulation for VLC and a custom light bulb integrated in a network of light bulbs. In combination with a configuration backend we can specify which information to be broadcasted for a certain proximity service. For instance, light advertisements for private service discovery, light tokens and credentials for automated authentication and authorization, and location identifiers for device mapping related to iConfig. Thereby, we use multiple communication channels combining mid-range radio-based communication, such as Wi-Fi, with short-range non-radio-based communication, like visible light from iPresence. VLC is only used as downlink to broadcast certain information which requires the property to be distance-limited enabling useful proximity services. The well-known radio-based communication acts as basic means for information exchange using up and downlink channel. Moreover, we realized visible light signaling (VLS) to broadcast light patterns to define different proximity regions for device association, not usable for data exchange in contrast to VLC. The passive method for co-presence reasoning is only a feasibility study as an extension of iPresence, not a part of the EdgeProx platform, using context information, such as location and user activity related sensor data, whether it is possible to infer user proximity from sensor data.

iConfig for IoT device management ② provides a backend for device administration including a global device map with device status, e.g., battery power, and location

information, such as a room number or relative indoor coordinates. We reduce the main barrier of IoT device management by an easy registration of IoT devices at the iConfig backend. Therefore, we emphasized on speech control for user-based device discovery. The user wears a smart glass with integrated speech control to encounter nearby Wi-Fi and Bluetooth devices, such as IoT boards and Bluetooth beacons, and synchronizes the device data, e.g., location, energy system state, to the iConfig backend for device monitoring, debugging, and localization. To achieve an autonomous operation and further reducing the operational costs, we use COTS drones in indoor environments for device detection and registration being independent of any ground control station, the drone controller and device detection platform flies with the drone. The user-oriented proximity services based on iPresence also benefit from iConfig in the following two ways: 1) maintain service reliability by identifying and localizing broken IoT devices for replacement and 2) location information of IoT devices.

To demonstrate the usefulness of our EdgeProx platform, iService ③ provides a set of user-oriented and privacy-aware proximity services: seamless device association, private service discovery, automated authentication and authorization. Our aim is to improve the user experience by finding novel ways how to apply visible light: a) automate tedious and reoccurring user tasks and b) integrate social applications in everyday environments. In the IoT domain, embedded security and privacy-preserving mechanisms are a key requirement for the user acceptance. Our proximity services based on EdgeProx address some open issues regarding user privacy ④ which we identified for D2D communication (basis for IoT platform architecture). To be specific, we achieved user anonymity based on distance-limited service advertisements using VLC, and we improved the usability of security solutions, e.g., two-factor authentication, by a fully automated operation avoiding manual interaction.

1.4 Contributions

The main contributions of this dissertation are:

- The investigation of D2D communication for data exchange between physically proximate devices and identifying the unique D2D characteristics compared to machine-to-machine (M2M) communication and mobile ad hoc networks (MANETs). In particular, the in-depth analysis and discussion to identify open issues of user privacy in D2D communication.

- The design and implementation of a 3D printed custom light bulb with a dedicated modulation scheme for VLC inspired by Morse coding to eliminate the light flickering effect. The light bulb can be embedded as a regular light source into the infrastructure and addresses the crucial challenge faced by conventional VLC designs: usability in practical deployment.

- The design and implementation of a lighting configuration framework to create a network of custom light bulb(s) and flexibly define different proximity regions and thereby control the spatial granularity of user's proximity. Moreover, an in-depth analysis to use VLS for co-presence reasoning.

- The collection of a multimodal sensor dataset for co-presence reasoning, sensor data from mobile devices related to location or user activity, e.g., barometer, accelerometer. The dataset includes multiple proximity verification sets to be able to associate sensor's data with a spatial granularity.

- The identification of suitable sensor modalities for co-presence reasoning of users covering different geographic granularity. Moreover, the quantified impact of device heterogeneity and user mobility on co-presence reasoning and a precise energy analysis on mobile devices to assess the energy demand of different sensors for co-presence reasoning.

- The design and implementation of a private service discovery to protect the user identity. Furthermore, to enhance usability, the automated scheme for authorization of home control functions.

- The design and implementation of an IoT device management with a global device map for administration and multiple edge modules intended to run on user devices, e.g., wearables, and drones to interact with physically nearby IoT devices. A usability study and testbed experiments to highlight the difference between manual and automated IoT device management.

- The design and implementation of a drone control independent of any ground control station, including two area exploration strategies tested in multiple indoor testbeds. In addition, the optimization of the flight path using hot spots, i.e., cover multiple IoT devices at once without flying to each device individually, to find an optimum in the trade-off between limited flight time and maximum of managed IoT devices.

1.5 Thesis Structure

To better understand the EdgeProx platform, in Chapter 2, we explain essential IoT technologies to drive the vision and we introduce cloud computing, edge computing, and D2D communication regarding the architecture of the EdgeProx platform. Moreover, we highlight multiple approaches for proximity-based services (PBS) to infer the physical presence of users and how to protect sensitive user data. For the main part of this thesis, Chapter 3 (RQ1) explains iPresence including basics of our light modulation and signaling scheme to send and receive information via visible light and introduces light bulb networks composed of our custom light bulb. In Chapter 4 (RQ2), we present the semi-centralized iConfig platform for IoT device management and focus on the device discovery and device mapping using wearables and drones. Chapter 5 (RQ3) highlights iService which consists of multiple proximity services based on the EdgeProx platform for a better integration into IoT environments, e.g., seamless device association using VLS for data sharing. We conclude this thesis with a discussion in Chapter 6.

2 Background

This chapter serves as background to be able to understand the design decisions for our EdgeProx platform. We first approach the IoT from different networking and communication technologies in terms of how to connect devices. Second, we introduce cloud computing, edge computing, and D2D communication regarding our edge-driven platform architecture for the IoT. Finally, we analyze proximity to infer physical presence among users and how to protect their privacy. To clarify, we do not explain VLC basics because we use VLC only as a tool to fulfill user-oriented services, which require proximity as enabling factor either for integrated services, e.g., data sharing, or to automate tedious user tasks, e.g., two-factor authentication.

2.1 Essential IoT Technologies

The IoT is recognized as one of the most important areas of future technology and is gaining wide attention from several industries. We describe the IoT building blocks to gain a better insight into the real meaning and functionality of the IoT.

From a high level perspective, we describe in the following which functionality is needed as foundation for the IoT. Identification and addressing of objects is crucial to match services with their demand and communication technologies connect heterogeneous objects together. Moreover, the IoT includes performance-limited computation units, e.g., micro-controllers, micro-processors, field programmable gate arrays (FPGAs). On this basis, sensing gathers data from objects and sends it to the cloud for analysis and take specific actions for IoT services [10]. Thereby, the semantic ability is important to extract knowledge from data to provide powerful services. The IoT involves four different service classes building on top of each other. Identity-related services are the most basic to identify and transform real world objects into a virtual representation being usable for applications [10]. Information aggregation services collect and summarize raw sensor measurements for further data analysis and collaborative-aware services use the obtained data to make decisions and react accordingly [10]. Finally, ubiquitous services aim to provide collaborative-aware services anytime they are needed by anyone who

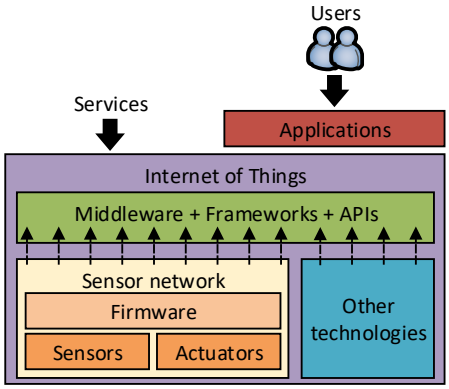


Figure 2.1: Most important IoT components [8]

needs them anywhere. To pinpoint, being able to identify things is critical for the success of the IoT, the addressing scheme must fulfill the following requirements: uniqueness, reliability, persistence, and scalability [17]. Moreover, due to the large heterogeneity of participating objects and wide variety of applications, the middleware between things and application layer plays a key role to abstract functionalities and communication capabilities of the devices [24]. Hiding the details of different technologies is fundamental for an easy adoption by developers to quickly realize novel services [25]. Fig. 2.1 shows the most important IoT components to process collected data from sensors to enable decision-making where actuators perform the decided actions.

As our EdgeProx platform has a special emphasis on novel communication technologies, such as visible light and ultrasound, we explain several communication technologies which pave the way for the IoT development to support a wide range of applications in different domains [18]. Radio frequency identification (RFID) was a breakthrough in the embedded communication paradigm to enable the design of microchips for wireless data communication [17]. The key feature of RFID is the automatic identification of objects using attached RFID tags, i.e., a microchip with an antenna [17, 24]. A typical RFID system consists of a tag (or responder) and a reader [26]. Passive RFID tags are not battery powered and use the power of the reader's interrogation signal for communication, e.g., in supply chain management [17]. Active RFID readers with own battery supply can instantiate the communication. The main advantage of RFID is the automated identification and data capture of distributed objects [26]. Besides that, wireless sensor networks (WSNs) are the most essential IoT component, which can cooperate with RFID systems to better track the status of things, e.g., getting information about position, movement, temperature, etc. [24]. A WSN is a network of things, namely tiny inexpensive autonomous devices equipped with sensors, which can take measurements,

locally store, handle sensed data, and can communicate to each other in a multi-hop fashion [18, 24]. Data is generated by sensor nodes and collected by special nodes, mobile and static sink nodes, which send the data to low-end computational devices [8, 24]. These devices perform a certain amount of processing on the sensor data and send the data further to high-end computational devices and reaches finally the cloud, where it is shared, stored, and processed significantly [8]. WSNs face many issues: 1) communication, such as communication range, reliability, and 2) resources, like restricted power, storage capacity, bandwidth availability [24].

Ad hoc wireless technologies have enabled mobile ad-hoc networks (MANETs) that can complement the sensing and communication infrastructure of the IoT domain. However, they are clearly different from sensor networks and have several weaknesses, e.g., not scalable [8]. MANETs are networks of mobile nodes (of the people) with impromptu connections without relying on fixed infrastructures. The wireless mobile nodes can freely and dynamically self-organize into arbitrary and temporary ad hoc network topologies [27]. It allows people in a restricted area to send, receive, and share data, without the need of either infrastructure or centralized support [18]. In this regard, opportunistic networks are one of the most interesting evolutions of MANETs, mobile nodes are able to communicate with each other even if a route connecting them never exists [28]. While messages are en route between sender and destination, any possible node can opportunistically be used as next hop as long as it is likely that the message gets closer to the destination [28]. In contrast to traditional MANETs, the nodes are not supposed to possess or acquire extensive knowledge about the network topology.

2.2 Edge-Driven Platform Architecture for the IoT

We explain the basics of cloud computing and how IoT and the cloud can benefit from each other. Moreover, we introduce edge computing as a meaningful extension of cloud computing to fulfill different needs of IoT. Taken together, this helps to better understand the architecture of our EdgeProx platform in the IoT context.

2.2.1 Cloud Computing

In the classical enterprise paradigm, an internal service provider runs its application on a physical server located on their premises, compared to the cloud computing paradigm, where the service provider rents computing resources from the cloud provider. A definition of the cloud states [29]: ‘Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing re-

sources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.' The architecture of a cloud can be split into four layers: hardware, infrastructure, platform, and application [30]. Each layer acts as a service for the layer above and as a consumer for the layer below resulting in a Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) [30]. SaaS means the provisioning of applications running in the cloud, PaaS refers to platform-layer resources, e.g., software development frameworks, and IaaS provides processing, storage, and network resources. Cloud computing acts as a base technology for the IoT to provide virtually unlimited capabilities and resources to store and process the enormous amounts of data generated by IoT devices [17, 24, 31]. For example, the growth of IoT applications requires scalable computing and storage solutions as largely driven by the IoT, the total amount of data created by any device will reach 847 zettabytes per year by 2021, up from 218 zettabytes per year in 2016 [32]. There are different kinds of clouds; in this thesis we focus on the local cloud deployed in a local network, which provides low communication delay but has computation limitations due to its sparse resources [33].

IoT environments deal with a wide heterogeneity of devices, technologies, and protocols where cloud computing can provide beneficial properties, such as scalability, interoperability, flexibility, reliability, efficiency, availability, and security [24]. In many cases, the cloud can provide the intermediate layer to implement applications and services that exploit things or data produced by them [24]. In the other direction, cloud computing can benefit from the IoT by extending its scope, including spatially distributed real world things in dynamic environments, to deliver new services in a large number of real life scenarios [24]. When critical IoT applications move to the cloud, concerns arise in terms of service availability, trust in the service provider, data ownership, e.g., physical location of data, and security [24]. Some integration challenges of cloud computing and the IoT are [31]: 1) heterogeneity of devices, operating systems, platforms, and services for applications, 2) applications with specific performance requirements for communication, computation, and storage, and 3) reliability for mission-critical applications, such as vehicles on the move and the vehicular communication is often unreliable. Finally, 4) big data requires special attention to transport, storage, access, and processing of the huge amount of data produced by networked devices.

2.2.2 Edge Computing

We introduce and analyze the concept of edge computing motivated by the fact that our EdgeProx platform relies on edge computing. The local cloud of EdgeProx provides

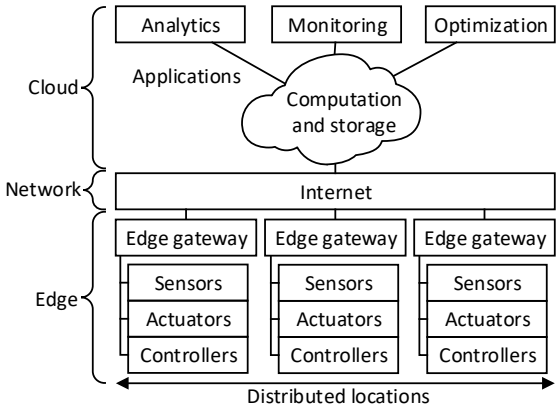


Figure 2.2: Three-layer IoT architecture composed of cloud and edge computing [33]

visualization, monitoring, and service provisioning. Our custom light bulbs act as edge nodes to provide different user-oriented services, such as device association, data sharing, where the physical vicinity of users is crucial.

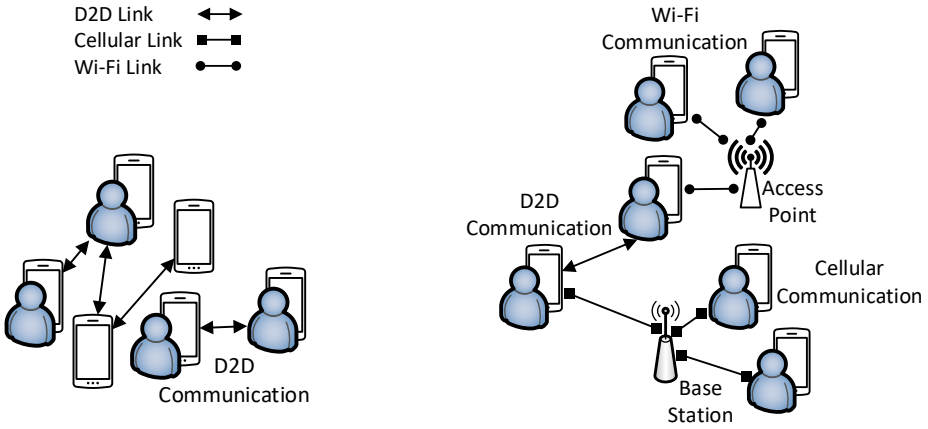
Traditional cloud solutions are great for large-scale general purpose computations with benefits regarding flexibility, efficiency, and the ability to store and use data [34, 35]. In the common IoT architecture, the data streams generated from distributed IoT devices are transmitted to the cloud for data processing, which is efficient as the computing power exceeds the capabilities of the network edge by several orders of magnitude [36, 37, 38]. However, the network bandwidth constraints are becoming the bottleneck for the cloud-based computing paradigm due to the vast and rapidly growing number of connected things, which generate data at Exabyte order [39, 40, 38]. In addition, cloud computing is problematic for IoT applications using private data to share with third-party service providers. Furthermore, cloud computing fails to meet new requirements of some IoT applications, such as a short response time (stringent latency) as many industrial control systems demand that end-to-end latencies between sensor and control node stay within a few milliseconds [38, 35].

To address the previous challenges, edge computing is able to extend cloud computing to be closer to the users and things it supports, i.e., move computing and storage service from the cloud to devices (nodes) at the network edge in the proximity of data sources [38, 35]. We define 'edge' as any computing and network resource along the path between data sources and cloud [38]. For example, a smartphone serves as edge between body things and the cloud or a gateway as edge in a smart home between home things and the cloud. Fig. 2.2 shows a three-layer IoT architecture where the edge layer resides between cloud and sensors. Due to the distributed architecture and closeness to things

and end-users, IoT applications and services benefit from edge computing in multiple ways [38, 34, 10]: 1) shorter response time (reduced latency) for real-time services, e.g., edge resources perform data aggregation to send partially processed data instead of larger raw data to the cloud for further processing, 2) better scalability as the number of low-cost edge nodes can be flexibly increased to cope with the number of increasing end-users and things, and 3) the density of edge devices helps to achieve resilient and replicated services. To benefit from edge computing, the challenge to integrate edge computing with the IoT must be overcome, i.e., how to manage edge computing infrastructure so that all edge nodes provide the requested services efficiently to IoT devices [35]. To clarify, rather than cannibalizing cloud computing, it should be a fruitful interplay between cloud and edge computing, especially for data management and analytics, where edge computing enables new types of applications and services [41].

2.2.3 Device-to-Device (D2D) Communication

Current technologies for mobile communication, like cellular networks, are infrastructure-dependent. The data traffic is always routed through a core network, even if source and destination are in close proximity to one another. However, new communication technologies are required that can scale network capacity and enable data exchange on-demand over the right network connections [1]. The aim of device-to-device (D2D) communication is to leverage the physical proximity of communicating devices to extend the cellular coverage to achieve high data rates and low latency [42]. We identified two major models of D2D communication networks: 1) standalone D2D and 2) network-assisted D2D [1]. According to [43, 44] and shown in Fig. 2.3(a), standalone D2D enables devices to communicate directly with each other without crossing fixed network infrastructures, such as access points or base stations. The standalone D2D communication relies on local hardware capabilities and D2D devices must be able to organize communications by themselves. The local connectivity of D2D communication is motivated by geographic validity, i.e., local events are not interesting for the rest of the world, and temporal validity means that information is only valid for a limited amount of time. In contrast and highlighted in Fig. 2.3(b), network-assisted D2D communication requires infrastructure, such as base stations or access points, for communication organization and resource utilization [45]. Network-assisted D2D communications have emerged as a means to improve energy efficiency and reduce latency [23]. Common D2D applications and services include data offloading, gaming, content distribution, and group communication [44, 46, 43]. D2D communication offers strong anonymity and content privacy because the shared information is not stored at a central storage. Furthermore, D2D

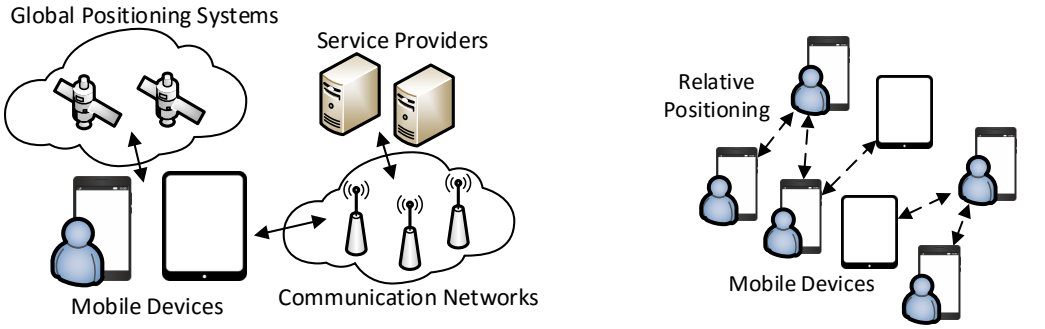


(a) Standalone D2D communication **(b)** Network-assisted D2D communication

Figure 2.3: System models of D2D communication [1]

communication offers enhanced system throughput, low end-to-end transmission delay, and energy savings due to the direct routing of D2D traffic [44, 42, 47]. On the other hand, D2D communication also entails some drawbacks, standalone D2D communication uses only device-managed links which makes centralized relay or channel management impossible [42]. The network-assisted D2D communication can partially manage relay and channel selections.

To clarify, machine-to-machine (M2M) communication, also known as machine type communication (MTC) [43], and mobile ad hoc networks (MANETs) are similar to D2D communication. In the following, we show the distinct properties between D2D communication, M2M communication, and MANETs. M2M communication occurs among machines or devices without human mediation and is routed through core networks, even if source and destination are proximate to one another [43, 48, 44, 49]. In addition, M2M communication is application-oriented and technology-independent, whereas D2D communication is technology-dependent and focuses on proximity services with opportunistic connectivity [43]. M2M communication is targeted at applications to automatically collect and deliver measurement information. Some unique features of M2M communication: communication between a massive number of devices, small and infrequent data transmission, and reduced need to recharge mobile devices [49]. The communication spectrum is a clear difference between D2D communication and MANETs, D2D communication uses the licensed and unlicensed spectrum and MANETs mainly use the unlicensed spectrum where interference is a major issue [44]. To pinpoint, D2D communication can rely on assistance from the network infrastructure for control functions, like synchronization, resource allocation, and routing, that are costly in a MANET [50].



(a) Location-based Services (LBS) **(b)** Proximity-based Services (PBS)

Figure 2.4: Common architectures of position aware systems [51]

Hence, the routing patterns vary, D2D communication uses mainly single-hop transmission whereas MANETs commonly use multi-hop routing leading to a decreased network performance [50, 44].

2.3 Proximity-based Services (PBS)

We realized multiple proximity services based on our EdgeProx platform. Hence, we provide a definition of Proximity-based Services (PBS) in comparison to Location-based Services (LBS). Moreover, we present multiple approaches for PBS, i.e., proximity detection methods provided as a service for easy usage by third-parties. A service is a collection of data and associated behaviors to accomplish a particular function or feature of a device or portions of a device [26].

The widespread use of LBS is based on the mainstream popularity of mobile devices, such as smartphones and tablets. LBS are based upon the absolute position of an user to answer the question 'where we are?'. Most of the LBS use global positioning systems, like GPS, to provide location-specific information. Thus, the functionality is mainly restricted to outdoor environments and the localization operation is energy-consuming, which is an issue especially for resource constrained mobile devices. In contrast, PBS are a subclass of the well-known LBS and aim to find co-location with other points of interest to answer the question 'who are we with?'. The popularity of PBS is largely driven by social networking applications, in which the direct communication between nearby mobile devices is particularly interesting. We can identify two essential phases of PBS: 1) users detect other users in the vicinity and 2) users intuitively share information and services among nearby devices. The goal of LBS and PBS is to improve the users' daily lives by providing a personalized service to enable sharing of location

information and location-aware information retrieval. Therefore, the LBS focus on a centralized architecture, the location server acts as Trusted Party (TP) which receives coordinates from the users to provide location-specific information, e.g., nearby friends. Fig. 2.4(a) shows the common LBS architecture consisting of four major entities: mobile devices, global positioning systems, communication networks, and service providers [52]. The assumption of a TP is risky regarding user privacy [53] and most of the LBS use global positioning systems limiting their functionality to outdoor environments, although people spend the majority of their time indoors [21]. In comparison to LBS with a global positioning, the PBS are trying to solve the issues of LBS by focusing on an infrastructure-less environment without a TP as highlighted in Fig. 2.4(b). Nevertheless, numerous PBS use absolute positioning and hence a TP to detect user proximity such as LBS. We can always build a PBS on top of an LBS. However, to solve the privacy issues, we need to build PBS without them and use instead context information, e.g., sensor data, to infer spatial closeness among users. The proximity of mobile devices is typically inferred in two ways: 1) by calculating distances between entities using coordinates from a positioning system and 2) by computing similarities of context information that is sensitive to the user's activity or location. Existing work [54, 55, 56] mainly focuses on strict geographic definitions of proximity based on relative distances, i.e., user A is 200 m away from user B. Our research efforts are directed along the semantic proximity [57]: 'Information about a location, its environmental attributes (e.g., noise level, light intensity, temperature, and motion) and the people, devices, objects, and software agents that it contains.' Use cases of PBS include public safety, localized social networking, home automation, local data transfer (offloading), mobile advertisements, and group recommendation.

2.3.1 Co-Presence Reasoning among Users and Devices

We introduce a proximity taxonomy related to our EdgeProx platform, the iPresence system module for co-presence reasoning: 1) communication technology for active co-presence reasoning and 2) context information, e.g., sensor data, for passive co-presence reasoning.

Active co-presence reasoning using radio-based communication Conventional radio-based communication, such as Wi-Fi, Bluetooth, ZigBee, is less sensitive to spatial barriers. The data signal penetrates wider areas which hinder a meaningful proximity reasoning. Hence, we need an additional complexity layer, i.e., a distance-bounding protocol, to achieve a distance limit between two entities. Distance-bounding protocols run a fast-exchange phase where one party would like to determine a practical upper-

bound on the physical distance to the other party [58]. For instance, to secure car entrance systems which are vulnerable to relay attacks between car and key [59]. The protocol is based on timing the delay between sending and receiving bits. The delay time of responses allows to compute an upper-bound on the distance, i.e., round trip delay time divided into twice the speed of light. This is based on the fact that electromagnetic waves travel nearly at the speed of light. A broad range of distance bounding protocols have been proposed for different technologies: RFID [60], ultra-wideband (UWB) [61], wireless ad hoc networks [62], and sensor networks [63]. The rigid distance threshold limits the accuracy of co-presence reasoning because varying situations require different thresholds, such as people walking side by side or people sit together at a table. Other approaches exchange multiple messages among communicating devices and measure the received signal strength indication (RSSI) to calculate relative distances among them [21]. However, the RSSI of the user's wireless connection can change unexpectedly and yields excessive false positives and false negatives. Many approaches use Bluetooth and Wi-Fi for co-presence reasoning based on time measurements of directly exchanged messages due to ease of implementation and wide availability in mobile devices [64]. To avoid additional complexity, such as distance-bounding protocols, we can use short-range communication like near-field communication (NFC), which provides a sniff-proof and distant secure channel with a maximum of 20 cm. From a usability perspective, the NFC technique is less flexible to support mobile scenarios, e.g., user moves at home in a range of 20 m. NFC requires to 'touch' the reader and is not scalable in crowded environments as it demands users to gather at a specific place.

Active co-presence reasoning using non-radio based communication In contrast, non-radio based communication, such as VLC and USC, has a high sensitivity on physical obstacles. The physical medium enables data exchange and the signal propagation (transmission range) is naturally restricted by the spatial surroundings, such as walls, doors, windows. VLC and USC can complement radio-based communication as it does not interfere with saturated radio spectrum, and it provides enhanced system security and user privacy because adversaries need line-of-sight access. Radio-based communication transmits data by using electromagnetic waves where radio waves are limited to a frequency range of 3 Hz to 300 GHz. The frequency range of visible light with 430 to 790 THz is 1200 times greater in comparison to the radio waves. This can help to solve to some extent the network capacity problem of wireless radio-based data transmissions in IoT environments with a vast number of devices. The VLC range of 4 to 30 m and performance between 100 kb/s up to 15 Gb/s results from different flavors of VLC platforms using a diverse set of hardware: from off-the-shelf IoT board with

low cost LED transmitters, over FPGA, to laser diodes better utilizing the visible light spectrum [65, 5]. However, the perceived throughput is mainly impacted by the intensity of ambient light. Compared to USC, VLC received more attention by the research community, the openVLC project [66] provides an open source platform for VLC including a software-defined MAC layer to support different experiments. Different modulation schemes were adopted for VLC at the transmitter, such as simple On-Off-Keying (OOK) modulation or Adaptive Multiple Pulse Position Modulation (AMPPM) to support dimming [67]. Supported by VLC, passive communication with ambient light [68, 69, 70] is a newly available communication channel. The signal emitter only sends a light signal without data modulation to activate nearby reflective surfaces. The environment includes mobile elements with distinctive reflecting surfaces that allows modulation of incoming light signals. The photodiode at the receiver's side decodes modulated light signals to read passive information. Passive communication provides several benefits: no power source required, enabled on demand by presence of light emitter, and improves communication privacy as only devices in the nearby environment receive the response for a limited amount of time. The drawbacks of visible light as communication medium include a high energy overhead and unpleasant visual experiences due to shining light beams. VLC has been enabling many applications related to the IoT, such as indoor localization [71, 72, 73], human identification [74, 72], gesture recognition [75], activity detection [76], occupancy detection [77], screen-to-camera communication, vehicle-to-X communication, object identification, and so on.

Besides VLC, we can use sound waves between 20 and 24 kHz to transmit information which is inaudible for humans and further increases the network capacity. Prototypes achieve bit rates between 8 to 1280 bit/s with a communication range from 5 cm to 25 m [78, 79, 5]. There are multiple modulation schemes for USC, e.g., the Orthogonal Frequency-Division Multiplexed On-Off Keying (OFDM-OOK) scheme [5] using off-the-shelf audio hardware like smartphones. To construct an ultrasound message, eight frequencies are used to define eight bits in a byte and one frequency for a parity check, encoding each bit in the byte to the same symbol. Each symbol uses a fixed duration of 46.4 ms (2048 samples at 44.1 kHz) and a guard interval of the same length between the symbols to prevent inter-symbol interference (ISI). A preamble and postamble with all bits on and thrice the regular pulse length defines the start and end of the message. Demodulating an ultrasound message includes: 1) message synchronization based on the preamble and postamble, 2) performing a short time Fourier transform (STFT) with a sample size matching the symbol length used for modulation, and 3) extracting the modulated byte sequence using a computed signal threshold to distinguish

bit one and zero. Ultrasound supports a range of use cases including device grouping [80, 81] and co-presence reasoning, i.e., speaker emits inaudible tones which are captured only by physically nearby devices, and proximity marketing [82] with location-tailored advertisements in casinos, museums, retail, and airports.

Passive co-presence reasoning using context information Location or user activity related context information, such as noise level, lighting, magnetic field, is beneficial to infer whether entities are physically nearby. Three important aspects of context are: where you are, who you are with, and what resources are nearby [83]. The context helps a system to provide relevant information and/or services to the user where relevancy depends on the user's current situation [84]. A number of approaches are using visual, acoustic, tactile or vibrational sensors for co-presence reasoning based on the similarity of Bluetooth signals [85], Wi-Fi and LTE signals [86, 87, 88], ambient sound [89, 90], magnetometer readings [91], ambient noise and luminosity [92], accelerometer data affected by hand shaking [93, 94], images where users see a similar scenery at roughly the same time [95], gait cycle detection of moving users [96], and Doppler profiling for ultrasounds combined with voice profiling to detect close-by devices [97]. Other systems [98, 99] are using sound as distance estimator between mobile devices using propagation delays of audio beacons that are transmitted by each phone. Thereby, some approaches for co-presence reasoning require infrastructure support, such as BLE beacons [100, 101] or ultrasound beacons [102] to emit messages to recognize user's co-presence.

2.3.2 Protection of Sensitive User Data

In the following, we give a brief overview about mechanisms to protect user data: access control and anonymity, which we applied to improve the user privacy of our proximity services. To clarify, 'Privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others' [103].

Access control is a mechanism that allows the user to grant or deny a service provider or other users the right to perform a certain action on the user's sensitive information [1]. The user should decide whether to share this information or not during the communication. Most systems for access control use semantic web technologies, such as OWL ontologies, RDF or SWRL, to model privacy policies, user context, and roles. We identified three different access control techniques [104]: 1) discretionary access control (DAC) uses the identity of the subject for authorization of certain actions, 2) role-based access control (RBAC) takes advantage of the subject role within a structured organization, such as a company or hospital, to define access control policies, and 3) mandatory access

control (MAC) uses a sensitivity level assigned to each object and a policy defines which sensitivity level is allowed to access the private information.

Obfuscation hides the user identity by reducing the data accuracy which may result in a negative impact on the service quality. Anonymity-based techniques overcome this problem by protecting the user identity without sacrificing the information accuracy. We can combine an anonymization technique with a reputation mechanism to create trust among anonymous communication partners. Hence, mobile users feel more comfortable and are willing to share more sensitive content, even sharing content with strangers. For example, the approach in [105] anonymously verifies the reputation score of users by using periodically changing pseudonyms associated with a reputation level. Cryptographic blind signatures are used to prove the source reputation without revealing individual user identity. Pseudonyms are another idea to achieve anonymity: a pseudonym is a subject identifier other than one of the subject's real names [106]. Two pseudonym requirements are essential to ensure privacy [107]: 1) a new pseudonym should always be available in case of a pseudonym change and 2) a pseudonym must have a validity period to avoid tracking. Since each pseudonym is unique, all corresponding user messages are linkable, which requires additional techniques to exchange pseudonyms between mobile users achieving non-linkability. These mechanisms can be categorized into three groups. First, periodical change randomizes the period to change pseudonyms, e.g., a time-slotted pseudonym pool with swapping functionality. Every mobile user has a pseudonym pool and uses each pseudonym for a specific time slot [108]. Second, context-based mix zones take advantage of social spots, i.e., crowded environments, where users don't exchange any messages and each user receives a new pseudonym when leaving the mix zone [109, 110]. Third, collaboration among nearby users to communicate with each other to synchronize their pseudonyms and confuse the adversary. The mobile device monitors the neighbors within a certain radius and the pseudonym exchange occurs only when the predefined threshold of nearby users is reached [111].

3 iPresence: Co-Presence Reasoning using Visible Light and Sensor Data

Recall from Chapter 1, the currently missing context-awareness of IoT middleware limits the availability and usability of IoT services. To integrate some sort of intelligence into the IoT platform is beneficial to provide relevant information and/or services depending on the current situation. As a key system-level feature of IoT platforms to enable context-awareness, the current efforts for a practical proximity communication are unilaterally focused on radio-based communication. We have chosen the emerging visible light for distance-limited ubiquitous data exchange among users and IoT devices based on light sources around us. In this regard, we designed and developed iPresence including active co-presence reasoning using visible light and passive co-presence reasoning using activity and location-related sensor data. We focused on the active co-presence reasoning for a reliable proximity service, the passive method for co-presence reasoning is a feasibility study as a possible extension of iPresence, not a part of the EdgeProx platform. Visible light provides appealing benefits not present in existing radio-based communication where distance boundary is becoming a highly desired attribute for various IoT services. In theory, such a distance boundary could be achieved by combining dedicated system design and communication technologies. In practice, we rarely find IoT services that can fully benefit from this feature due to the complexity of exploiting various radio-based communication technologies to attain the distance limit in different IoT environments. Since visible light does not pass through opaque objects, it is a good candidate to reinforce the spatial barrier control over different IoT services that demand for distance-bounding wireless communication. To take advantage of the unique distance-bounding feature of visible light, active iPresence combines mid-range radio-based communication, such as Wi-Fi or Bluetooth as primary communication means, with visible light for communication (VLC) and signaling (VLS). In this chapter, we aim to contribute to the first question of the research problem that we have stated in the introduction: **How to efficiently determine the spatial proximity of users without disclosing their locations?** The entire code of iPresence is available at `https://github.com/TUM-cm/iPresence` [last checked 16.06.2020].

We begin addressing this question in Publication 2 by building LocalVLC, a system for VLC including a 3D-printed custom light bulb with a dedicated modulation scheme for VLC inspired by Morse coding to eliminate the light flickering effect. The light bulb can be embedded as a regular light source into the infrastructure and addresses the crucial challenge faced by conventional VLC designs: usability in practical deployment.

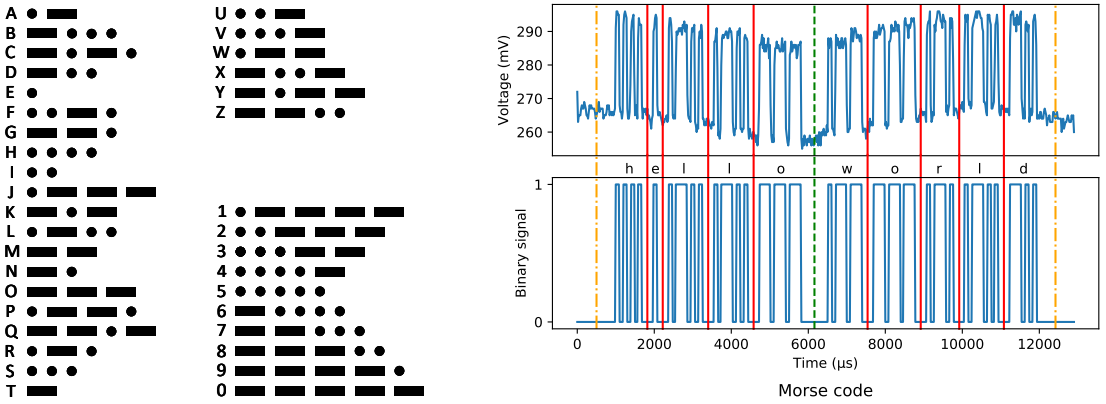
In Publication 3, we present DevLoc including the design and implementation of a lighting configuration framework to integrate our custom light bulbs into a network for communication and signaling. We can flexibly define different groups of light bulb(s) where we adopted a master-slave principle to form semantic subnetworks to cover larger rooms or across multiple rooms. As a result, we can control the spatial granularity of user proximity to overcome the main disadvantage of location tags, where the neighborhood is entirely dependent on the type of location tag without explicit control.

As a possible extension of iPresence, in Publication 4, we explore the opportunity to utilize built-in sensors from mobile devices for passive co-presence reasoning. The main goal is to support developers building better context-aware applications by a deepened understanding of which set of context information is appropriate for co-presence reasoning. Therefore, we use a multimodal sensor dataset gathered from mobile devices related to location or user activity, e.g., barometer, accelerometer.

3.1 Data Modulation for VLC and Visible Light Signaling

In spite of the aforementioned distinct advantages of visible light, many VLC designs focus mainly on communication performance and novel functionality [112, 113, 65, 66, 70, 114, 115, 116]. Therefore, plenty current designs for VLC commonly require dedicated LED lights to emit modulated light beams which entail high energy overhead and unpleasant visual experiences due to perceptible light blinking effects. This greatly limits the deployment and applicable scenarios of visible light and leads to the fundamental question of adopting visible light: how to achieve a practically deployable VLC and VLS with low cost and power footprint? We tackle the usability challenge for VLC by proposing a system solution named LocalVLC in Publication 2. For our VLC transmissions we seek a robust and yet simple encoding scheme for creating a low-rate signaling channel. We propose Morse encoding as lightweight data encoding for VLC to achieve a broadcast with low processing overhead. Common VLC modulation schemes can hardly keep light pulses imperceptible and hence causing light flickering effects [75].

To clarify, we do not aim for bulk data transmission or to replace well-established, de facto radio-based communication, such as Wi-Fi. Our aim is to add the distance-



(a) International Morse code defined by two signs to model Latin alphabet and Arabic numerals. Dot as smallest time base unit and dash with three time units.

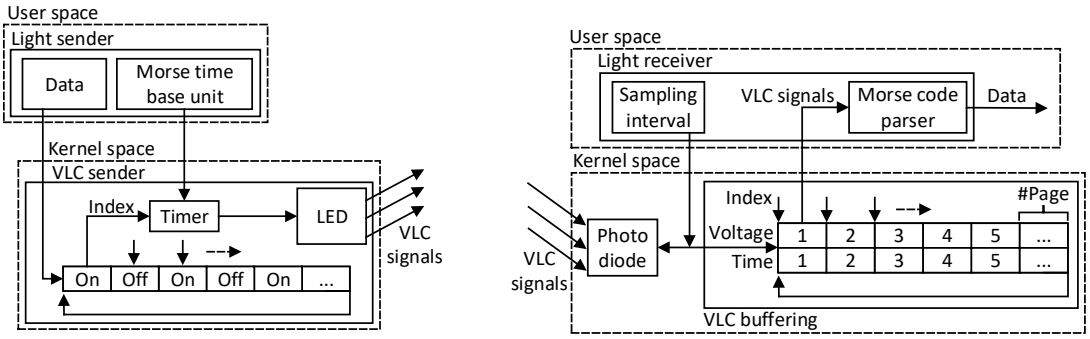
(b) Processing of Morse-code modulated light signal to decode information. We detect the change points from light on and off phases and calculate the duration of each phase to identify letters, words, and messages.

Figure 3.1: Basics of Morse code and example of data processing [2]

bounding feature to radio-based communication, which is otherwise hard to achieve, requiring additional complexity via distance-bounding protocols. Hence, we only need a satisfying throughput to transmit a limited amount of information in a reoccurring, broadcasting way, to enrich existing wireless communication with a distance-bounding information attribute that was missing before. Our distance-bounding visible light communication and signaling runs on any low-cost IoT board without special hardware requirements.

3.1.1 Working Principle of LocalVLC Morse Encoding

For data encoding we use the Morse code defined by the International Telecommunication Union [117]. The Morse code is based on two signs, a dot as the smallest time base unit and a dash is about three time units. Fig. 3.1(a) presents the ISO basic Latin alphabet and Arabic numerals encoded in Morse code. For example, the letter 'B' consists of one dash followed by three dots. We transmit data as a series of light on and off periods, each light on phase is followed by a light off phase. We use light off phases of different lengths to distinguish letters, words, and messages. Fig. 3.1(b) shows the processing of 'hello world' as raw light signal where LocalVLC uses the following algorithm for signal decoding. To get a binary sequence of light on and off phases, we quantize the raw voltage signal with a mean threshold. In the next step, we detect change points of light



(a) Processing chain for VLC sender **(b)** Processing chain for VLC receiver

Figure 3.2: LocalVLC sender and receiver [2]

on and off phases and compute the duration of each phase. To enhance the decoding robustness, we dynamically calculate the letter threshold by the mean over all light off phases. Given by the Morse code, the duration thresholds for words and messages are the multiple of the letter threshold. Based on the corresponding duration, we categorize all light on phases into dot or dash signals. This signal series of letters is split into single letters and decoded using a letter dictionary. Finally, we detect word or message stops for the correct formatting sign, either a white space or line break. Regarding the LocalVLC sender and receiver, we implemented two Linux kernel modules to send and receive data encoded in Morse code. The VLC sender in Fig. 3.2(a) requires as most important parameter the Morse time base unit to specify the period for a dot, the smallest time unit for Morse encoding. On this basis, all signs: a-z and 0-9 can be encoded into different light on and off phases to trigger two real-time kernel timers switching the LED between on and off state. For a limited period of time, the VLC sender periodically transmits the same information, e.g., service identifier or password token. With respect to the VLC receiver in Fig. 3.2(b), we implemented another Linux kernel module to sample the raw light signal from the photodiode as voltage in mV. A higher voltage value indicates a light on phase and a lower voltage value indicates a light off phase. After parsing the VLC signals, we apply error correction based on a majority rule, i.e., the VLC receiver selects the most frequently received information via a sliding time window of a few milliseconds.

3.1.2 System Parameters and Performance of LocalVLC Morse Encoding

The evaluation of our Morse-code inspired data modulation is divided into two parts. First, we identify the best working system parameters with respect to throughput, latency, and error rate. On this basis, we analyze the communication field of view (FoV)

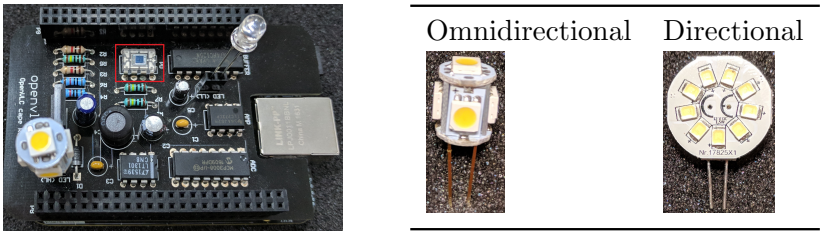


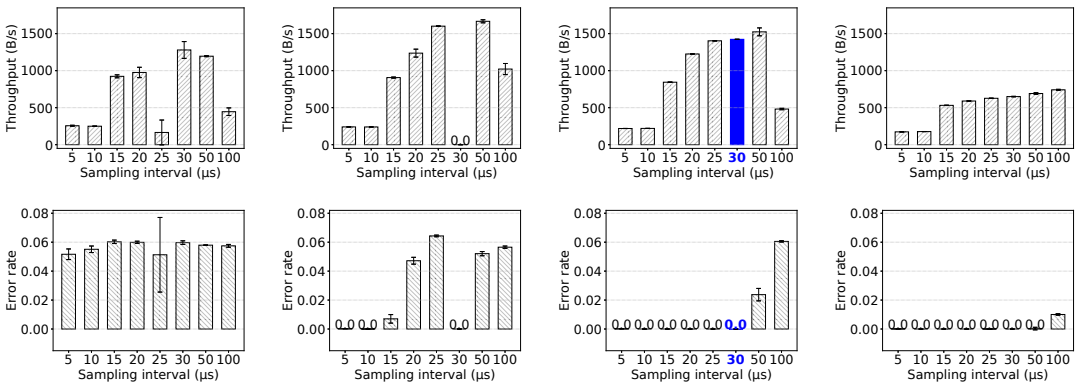
Figure 3.3: VLC evaluation platform (BeagleBone Black) with two LED types as transmitter where the photodiode (red frame) acts as a receiver [2]

and transmission distance. Second, we compare our LocalVLC data encoding based on Morse code with Manchester encoding regarding throughput and energy consumption.

Test Environment We use the VLC platform in Fig. 3.3 from the openVLC project [66] with two different LED types as transmitter and a photodiode as VLC receiver which is widely used in mobile devices, e.g., as ambient light sensor. Our testbed consists of a sender and receiver in a distance of 50 cm. The sender continuously transmits a test string: 'abcdefghijklmnopqrstuvwxyz1234567890'. The evaluation consists of 100 rounds where the VLC receiver gathers VLC transmitted data for a duration of ten seconds to calculate throughput, error rate, and energy consumption.

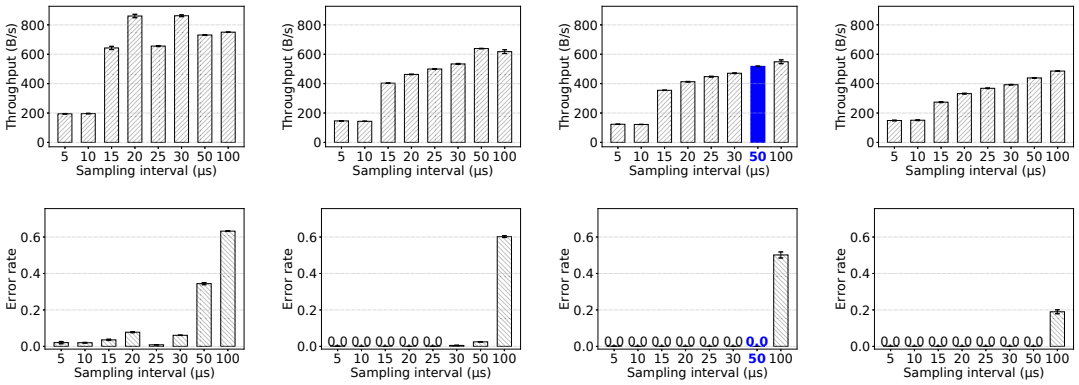
Best operational parameters for LocalVLC Morse encoding Using directional and omnidirectional LED as VLC transmitter, we determine the best working VLC parameters, the sampling interval at the receiver and the Morse time base unit at the sender, both of which affect the throughput and error rate. At the receiver, we use a sampling interval (µs) in the range of [5, 10, 15, 20, 25, 30, 50, 100, 150], i.e., how often the photodiode is sampled to receive data. At the sender, we analyze the Morse time base unit, the smallest time base unit (one dot) to encode the to be transmitted data. Using the omnidirectional LED, Fig. 3.4 shows the throughput and error rate for different sampling intervals and Morse time base units. The best working VLC parameters are: sampling interval of 30 µs and Morse time base unit of 50 µs, achieve a throughput of 1.4 kB/s without errors. Moreover, Fig. 3.5 presents the throughput and error rate using the directional LED. The best working VLC parameters are: sampling interval of 50 µs and Morse time base unit of 130 µs result in a throughput of 517.68 Bytes/s without errors. The directional LED has some hardware limitations, which leads to a lower sampling interval and decreased throughput. We recognize a flickering effect at the VLC sender with a Morse time base unit of 150 µs, the frequency range of our VLC sender is above this range.

VLC communication characteristics: maximum transmission distance and field of view (FoV) In terms of transmission range, Fig. 3.6 shows the maximum



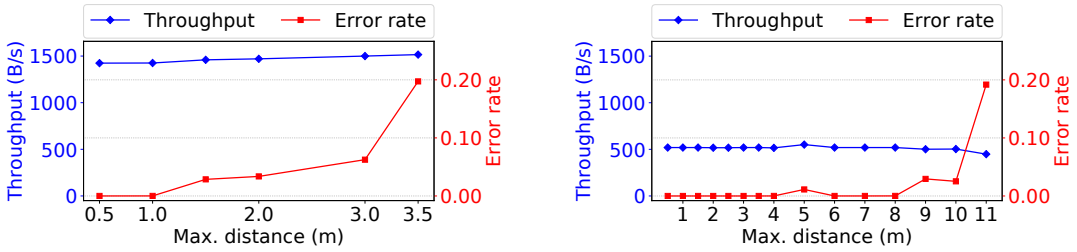
$\lambda = 10\,\mu s$ $\lambda = 20\,\mu s$ **$\lambda = 50\,\mu s$** $\lambda = 100\,\mu s$

Figure 3.4: Evaluation of LocalVLC encoding parameters using *omnidirectional LED*. The best working parameters are highlighted in blue: sampling interval of 30 µs and Morse time base unit of 50 µs (λ) $\approx$ 20 kHz [2]



$\lambda = 100\,\mu s$ $\lambda = 120\,\mu s$ **$\lambda = 130\,\mu s$** $\lambda = 150\,\mu s$

Figure 3.5: Evaluation of LocalVLC encoding parameters using *directional LED*. The best working parameters are highlighted in blue: sampling interval of 50 µs and Morse time base unit of 130 µs (λ) $\approx$ 7.69 kHz [2]

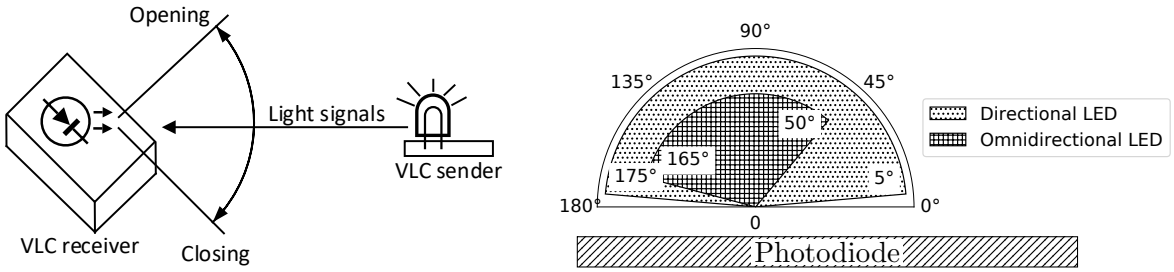


(a) Omnidirectional LED **(b)** Directional LED

Figure 3.6: Maximum range of VLC communication with different LED transmitters regarding throughput (B/s) and error rate [2]

achievable transmission range with regard to throughput and error rate. The omnidirectional LED is able to cover a distance of 3 m and the directional LED reaches a maximum distance of 10 m. The error rate increases with a larger distance between sender and receiver up to a level at which the communication is no longer possible. The throughput refers to the received number of characters within a time slot regardless of whether the data is correct, this is indicated by the error rate. In the next step, using the omnidirectional and directional LED, we measured the opening and closing FoV at the photodiode, receiver side. Fig. 3.7(a) highlights the testbed, in a distance of 0.5 m between sender and receiver, we changed the orientation of the receiver, alignment to the transmitter, and continuously monitored the VLC error rate to identify the opening and closing FoV. We measure the FoV in Fig. 3.7(b), the omnidirectional LED obtains a range of 165°–50° and the directional LED achieves a FoV of 175°–5°. The directional LED achieves a larger communication angle due to a better saturation, stronger light signal into one direction, the photodiode still receives data at more extreme angles. The omnidirectional LED provides the benefit of a surrounding signal propagation, several receivers around the LED at different orientations are able to receive the signal. However, the weaker light signal results in a smaller communication angle compared to directional light targeted to the receiver. In practice, at the given situation, we can utilize mirrors to steer the light signal to dynamically adapt the communication distance and FoV.

Comparison of LocalVLC Encoding with Manchester Code To highlight performance differences in terms of throughput and energy consumption, we compare our LocalVLC encoding based on Morse code with a baseline using On-Off keying modulation with Manchester code and Reed-Solomon error correction code. The data encoding with Manchester code utilizes a 50 kHz sampling frequency. We use the high voltage Monsoon power device to measure the energy consumption by powering our hardware platform (BeagleBone Black) with 5 V as shown in Fig. 3.3. To compute the required



(a) Testbed to evaluate opening and closing field of view at the photodiode in a distance of 0.5 m between sender and receiver.

(b) Opening and closing field of view at the photodiode, receiver side, with directional and omnidirectional LED at the sender side.

Figure 3.7: VLC communication angle: opening and closing field of view at the photodiode

energy in Joule per Byte (J/B), we measure the current (mA) and voltage (V) during the data transmission only at the receiver. The energy consumption at the sender is mainly influenced by the LED power, the data encoding scheme has only a minor influence on the system's energy. On average, our LocalVLC encoding achieves a 8.75 times higher throughput of 1010.16 B/s compared to Manchester encoding achieving 115.51 B/s. For the energy consumption we obtain a similar result because the energy $E = U \cdot I \cdot t$ is mainly dependent on the transmission time. LocalVLC encoding consumes 8.27 times less energy as Manchester encoding, 1.88 mJ/B compared to 15.58 mJ/B. Our LocalVLC encoding achieves a several orders of magnitude better performance with respect to throughput and energy consumption in comparison to the usual Manchester encoding. The superior performance is caused by a more efficient LocalVLC encoding compared to Manchester code. Since Morse code takes advantage of encoding pulses and pauses with different length and pauses are more frequent, while the Manchester code requires a high and a low pulse for sending each bit. In addition, our error correction does not decrease the possible payload, selecting the most frequently received information over a moving time window is computation-wise faster as the Reed-Solomon error correction code.

To sum up, under practical settings, LocalVLC can support up to 10 meters of range, and attain a reasonable throughput of up to 1.4 kB/s with low error rate and energy consumption. Compared to the widely adopted Manchester encoding, we can achieve 8x improvement on both throughput and energy consumption.

3.1.3 Generation and Recognition of Visible Light Signals

As supplement to VLC, we use random light patterns to define different proximity regions which is beneficial for certain use cases, e.g., device association. Our generated light patterns are unpredictable nonces associated with a location, like a shared pool of entropy between all users at a given location at a given time. Two key properties of light patterns are [118]: 1) reproducibility meaning that two measurements at the same place and time match with high probability and 2) unpredictability so that an adversary at another location is unable to produce a location tag that matches the tag measured at the actual location at that time. To generate a light pattern, we independently create a random series of light on and off periods and combine them. As we explain in the following, the duration of each light on and off period is in the range of [1, 5] ms. The hardware of our light receiver determines the minimum duration of light on and off phases, i.e., how fast the photodiode can be sampled. To avoid an unpleasant visual experience, the maximum duration of each light on and off period is limited to overcome light flickering effects which are visible by human eyes. We introduce a 10 % duration difference among light on and off signals to ensure that the time periods are sufficiently distinct improving the detection rate of light patterns at the light receiver. We sample the raw light signal as voltage in mV by using a photodiode at the light receiver. A higher voltage indicates a light on period and a lower voltage indicates a light off period.

The cycle detection algorithm from [96] is able to find reoccurring light patterns by signal matching of arbitrary co-aligned light signals and reaches a reliable signal segmentation using normalization. The algorithm gets as input a vector of voltage amplitudes $z = (z_1, ..., z_n)$ and the output is a sequence of consecutive light signal patterns. We take advantage of auto-correlation and distance calculation to find repeating signal parts. The auto-correlation can be efficiently calculated via the Wiener-Khinchin theorem [119] with complexity $n \cdot log(n)$

$$
\begin{aligned}
F_R(f) &= FFT[z] \\
S(f) &= F_R(f) \cdot F_R^*(f) \qquad * \hat{=} \text{conjugate} \\
R(\tau) &= IFFT[S(f)]
\end{aligned}
$$

where z are the voltage amplitudes. The auto-correlation $R(\tau)$ results in m non-ambiguous local maxima $\zeta = \arg\max(R(\tau)) = \{\zeta_1, ..., \zeta_i ..., \zeta_m\}$. We calculate the dis-

tances among all local maxima and a mean distance

$$\delta_{\text{mean}} = \left\lceil \frac{\sum_{i=1}^{m-1} \zeta_{i+1} - \zeta_i}{m-1} \right\rceil$$

where δ_{mean} can be used to choose minima indices from z which refer to signal patterns. Every local maximum specifies a start point and δ_{mean} a search range to find the local minima μ

$$\mu = \{\mu_1, ...\mu_i, ..., \mu_{m-1}\}$$
$$\mu_i = \arg\min(z_{\zeta_i}, z_{\zeta_i+1}, ..., z_{\zeta_i+\delta_{\text{mean}}})$$

Every μ_j refers to an index of a minimum in z limited by the range of δ_{mean}. The voltage amplitude z can be split by the indices in μ into light cycles

$$Z = \{Z_1, ..., Z_i, ..., Z_q\}$$
$$Z_i = (z_{\mu_{\frac{i}{2}}}, ..., z_{\mu_i}, ..., z_{\mu_{\frac{i+1}{2}}-1}) \text{ with } i = \{2, 4, ..., q\}$$

Identified by our experiments, due to sudden changes of light patterns caused by light interference, the number of successfully recognized light patterns decreases. Therefore, we implement our own method to detect light patterns based on the period of each light on and off phase. We define the light signal as a list of periods

$$\hat{z} = \{(s_1, d_1), ..., (s_n, d_n)\}$$

where $s_n \in \{0, 1\}$ specifies whether the light phase is on or off and $d_i \in \mathbb{Z}$ is the duration of each period. To improve the robustness of signal pattern detection, the light sender introduced a 10 % signal margin between the light on and off periods and hence we merge similar signal parts with a time difference smaller than 10 %. The resulting unique signal parts determine the signal pattern with the period of each phase. To extract the light pattern, we overlay the light signal with a time window of the pattern length which is manually specified for the system.

3.2 Configurable Light Bulb Network with Proximity Regions

The previous section introduced the basics using visible light for communication (VLC) and signaling (VLS). In the next step, we integrate the light evaluation platform from Fig. 3.3 into our custom light bulb to realize a practically deployable light bulb network.

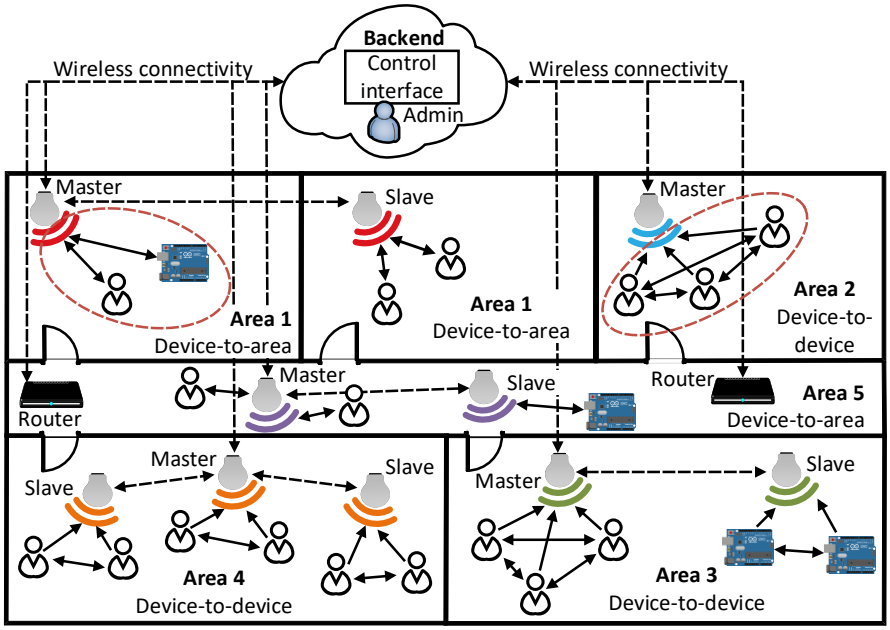


Figure 3.8: Light bulb network to enrich existing Wi-Fi infrastructure for fine-grained proximity services including mobile users and IoT devices [3]

In Publication 3, we take advantage of ubiquitous light sources around us for the DevLoc framework as shown in Fig. 3.8 ensuring practicality for proximity services in indoor IoT environments. DevLoc combines the spatially more fine-grained visible light with radio-based communication like Wi-Fi as the primary communication means covering larger areas. We enrich the lighting infrastructure by adding light communication (VLC) and signaling (VLS) to the widely used Light-Emitting Diode (LED) lamps in residential and office settings. We provide a complete framework to manage the lighting infrastructure where the administrator at the DevLoc backend can configure the data to be broadcasted via our custom light bulbs which require the unique distance-bounding feature of visible light. The different colors in Fig. 3.8 illustrate varying data for light communication or signaling at the light bulbs. Moreover, we are able to control spatial granularity of user proximity to support IoT applications with varying spatial expansion of proximity. Thereby, we overcome the main disadvantage of location tags where users have no control over the spatial granularity of proximity [118].

A central task of DevLoc is to make the light bulb network configurable to specify which data is broadcasted via visible light. At the DevLoc backend with the registered light bulbs the following actions are available: 1) for VLC the administrator defines the to be broadcasted light information and 2) for VLS the DevLoc backend randomly

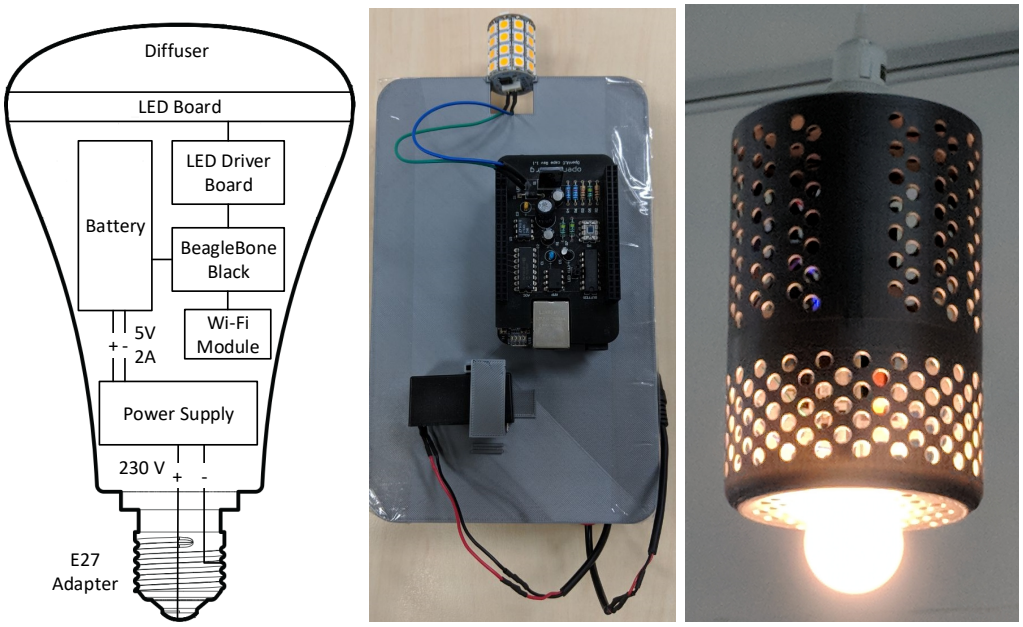


Figure 3.9: Hardware architecture of our custom light bulb for DevLoc; installed light bulb components; our deployed 3D-printed DevLoc light bulb [3]

creates different light patterns for different proximity regions defined by semantically linked light bulbs broadcasting the same pattern.

As a central part of DevLoc and inspired by [120, 114, 121], our custom light bulb in Fig. 3.9 offers an API for communication/signaling and controls the LED transmitter and wireless modules, such as Wi-Fi and Bluetooth. During normal operation the battery is loaded and provides the power for the BeagleBone Black. The battery improves the service availability and maintains the lighting in case of a power blackout. Based on our custom light bulb, we are able to replace existing illumination units to limit the problem of light pollution, where different visible lights would be overlapping for illumination and communication.

3.2.1 Adaptable Spatial Granularity

DevLoc allows to define the geographic structure of user proximity based on selected proximity areas. By using a master-slave principle for the light bulbs, we are able to semantically link multiple rooms or regions and thereby flexibly control the user proximity. Initially, each light bulb and Wi-Fi router registers itself at the DevLoc backend running the lighting configuration framework. Hence, DevLoc knows all distributed light bulbs and their specific areas and randomly chooses for each region one of the light bulbs as

master light bulb, the remaining ones act as slaves. We use a publish-subscribe mechanism to realize the master-slave principle. Via the subscription to the central backend of DevLoc, each master light bulb receives the configured light information to be broadcasted which is then further published to the slave light bulb(s). We can dynamically configure the spatial granularity of user proximity by adapting the groups of light bulbs covering different areas. We broadcast the same light information over different rooms which are semantically the same area, e.g., area one in Fig. 3.8 to link two rooms. The achievable spatial granularity of user proximity is defined by the size of the rooms and regions like corridors, and the number and geographical distribution of light bulbs. To achieve the most fine-granular user proximity, each light bulb works on its own as master.

For a better integration between visible light and Wi-Fi, our master light bulbs continuously monitor the wireless connections of the Wi-Fi router being able to trigger proximity actions. This is required for certain use cases, such as device associations, being able to combine visible light and Wi-Fi communication to achieve a practically usable system. In case of a new Wi-Fi client, the master light bulb is able to run signal matching of light signals to infer which devices are in the same light communication range instead of being only in the same Wi-Fi coverage. Due to the larger Wi-Fi coverage, one router can be combined with multiple master light bulbs. We do not use signal strength changes of the user's Wi-Fi connection, for example, to estimate how long the Wi-Fi connection further exists, because it can change unexpectedly and produces excessive false positives and false negatives.

Furthermore, the master-slave mechanism provides the basis to keep the technical changes on existing illumination to a minimum. Only the master light bulb(s) need computing power to perform certain actions for proximity services, e.g., signal matching for device associations. The slave light bulbs require only a wireless connection to receive the commands from the corresponding master light bulb. On the other hand, this limits the flexibility of spatial user proximity if some light bulbs are not able to fulfill both roles: master and/or slave due to missing hardware capabilities.

3.2.2 Two Modes of Co-Presence Reasoning: Device and Area

To support either location-based services (LBS) or proximity-based services (PBS), we are able to specify the mode of proximity for each master light bulb: device-to-area proximity for LBS and device-to-device proximity for PBS. The dotted red circles in Fig. 3.8 highlight the proximity among different entities using different light information as input for signal comparison: a) device-to-area using the device's light signal and an area's reference light signal or b) device-to-device using only the device's light signals. We

encounter three main differences between device-to-area and device-to-device proximity: 1) the trigger point in time of co-presence reasoning, 2) the required number of devices for proximity computation, and 3) the signal comparison between different entities which affect the resulting binding either device-to-area or device-to-device. For device-to-area co-presence reasoning, after the device is connected to the Wi-Fi router, the corresponding master light bulb(s) immediately start the co-presence reasoning and compare the device's signal to the area's reference signal. There is no restriction regarding number of connected devices, e.g., at least two connected devices for device-to-device co-presence reasoning. We establish a direct binding between the device and area. We know which device is in which area and at the same time which other devices are nearby. On the other hand, for device-to-device co-presence reasoning we need at least two connected devices at the Wi-Fi router. To associate a new Wi-Fi client to a specific device group, the master light bulb randomly chooses one client from each existing device group for signal matching. The participating devices do not know at which indoor region they are located, they only know which other clients are nearby. Hence, we can only realize PBS among close-by users and LBS are not feasible because location-related information is missing using the device-to-device co-presence reasoning.

3.3 Extension: Location and User Activity Related Sensor Data

In Publication 4 we address the question whether multimodal sensor data are suitable to realize a co-presence reasoning with varying spatio-temporal granularity? The main problem is a practical dataset including several proximity verification sets and sensor data of mobile devices, where we ensure the physical co-presence of user devices during data collection. Co-presence reasoning enables context-aware applications, such as social networking among nearby users, and the modeling of human behavior. We aim to identify which set of context information is appropriate for co-presence reasoning helping developers to build better context-aware applications. Co-presence is defined as: two individuals are 'close' when their similarity of context information is large [122]. Context describes any information that can be used to characterize the situation of a person, place, or object that is considered relevant to the interaction between a user and an application [84].

3.3.1 Overview of Proximity Data Collection

We gathered a multimodal dataset using the AWARE framework [123] from 126 students over three months for co-presence reasoning. This dataset is released as an anonymized

Table 3.1: Overview of sensor dataset [4]

Data characteristic	Sampling rate	Study data
User activity	5 s	(Linear) accelerometer
User position	5 min	GPS, network
User environment	5 min	Barometer, magnetometer, temperature, light, gravity, gyroscope, rotation, GSM towers, Bluetooth and Wi-Fi devices

subset named 'Proximityness' [124] at CRAWDAD for evaluation of co-presence reasoning. Table 3.1 shows the sensor data to be collected from the student's mobile devices. Our proximity verification sets include GPS and network locations, and Bluetooth, Wi-Fi, and GSM encounters. For a sufficiently fine-grained Bluetooth verification of user's proximity, we distributed 50 BLE beacons at different parts of the campus covering main entrances, lecture halls, library, and cafeteria. Initially, we analyzed whether the data quality is sufficient as our co-presence reasoning is dependent on sensor data sensed from multiple devices at the same time and place. The parameters to assess the dataset quality to infer user's proximity: 1) enough active users contributed sensor data over a longer period and 2) sensor data is evenly distributed over time. Based on this, we select the following wireless and sensor data targeted for co-presence reasoning: Wi-Fi access points (APs) connected, Wi-Fi neighbors, Bluetooth neighbors, GSM towers connected, GSM tower neighbors, GPS and network location, accelerometer, barometer, and magnetometer.

3.3.2 Co-Presence Reasoning with User Mobility / Device Heterogeneity

By using different proximity verification sets with a variety of spatial granularity, e.g., Bluetooth, Wi-Fi, GSM, our aim is to analyze if sensor modalities are appropriate for co-presence reasoning. Our gathered and selected sensor data for evaluation: barometer, magnetometer, and accelerometer is dependent on the user's location or activity where we assume that people share the same context similarity. We use multiple proximity periods $\in$ [5, 10, 15, 20, 25, 30] min to evaluate the time granularity of user's co-presence. To verify the user's proximity, two user devices have to encounter the same wireless device, e.g., Bluetooth device, Wi-Fi access point, or GSM cell tower, within the proximity time window. As we aim for recent proximity encounters, we set the proximity time window in the range of 5 to 30 min. Moreover, within the proximity period, we take the difference among all devices and nearby devices to recognize remote devices which are not close

Table 3.2: For each proximity verification set we identify the most effective sensor modality to detect co-presence fulfilling different spatio-temporal granularity [4]

Verification set	Sensor data	Spatial granularity	Proximity period	Proximity signal distance $\bar{\delta}_p$	Non-proximity signal distance $\bar{\delta}_{np}$	Signal distance ratio $\bar{\delta}_{np}/\bar{\delta}_p$
Bluetooth neighbors	Accelerometer	26–124 m	25 min	992.9	1888.5	1.9
Wi-Fi APs connected	Magnetometer	98–819 m	20 min	652.9	1455.8	2.2
Wi-Fi neighbors	Barometer	165 m–1.2 km	30 min	0.7	13.2	18.3
GSM towers connected	Magnetometer	15–243 km	15 min	772.5	2181.8	2.8
GSM tower neighbors	Magnetometer	73–266 km	30 min	2906.3	9353.8	3.2

to each other. For signal comparison, we require timely aligned sensor's data across user devices. Hence, our data collection framework executes the clock drift correction of user devices during the daily data upload to the data collection server. For co-presence reasoning, we use the dynamic time warping distance named δ to compute the signal similarity within each group of devices in proximity, given by our proximity verification set and between each device in proximity and all distant devices. We take the mean of signal distances among nearby devices $\bar{\delta}_p$ as well as between remote devices $\bar{\delta}_{np}$. We assume that the signal similarity among nearby devices is higher compared to that of distant devices. We use the raw sensor signal to evaluate the basic performance of our co-presence reasoning. Table 3.2 presents the most effective sensor modality and proximity period for each proximity verification set with a varying spatial granularity using the maximum signal distance ratio $\bar{\delta}_{np}/\bar{\delta}_p$ between the proximity $\bar{\delta}_p$ and non-proximity signal distance $\bar{\delta}_{np}$. Co-presence reasoning is not possible if the signal distance ratio is one, i.e., no difference in the signal similarity among nearby and remote devices. Hence, the larger the signal distance ratio the better for co-presence reasoning. The proximity period specifies how much time elapses before we can identify the most effective co-presence reasoning. Moreover, given by the proximity verification set, we are able to associate sensor's data with a spatial granularity. There is a clear distinction of signal similarity among nearby and remote devices which enables co-presence reasoning with a

varying spatio-temporal granularity. Based on our results, developers of context-aware applications can select the suitable wireless or sensor data for co-presence reasoning. For example, the magnetometer data offers the most diverse spatial granularity of user's co-presence from a few hundred meters up to kilometers, compared to the accelerometer data with a working range between 30–100 m.

To analyze the impact of user mobility on co-presence reasoning, we use the entropy of random device encounters at different locations to specify the user mobility covering both movement patterns and variability. The randomness of user mobility is crucial for co-presence reasoning, e.g., a user moves several hundred meters each day but only between two positions resulting in a higher moving distance but low entropy. We calculate the user entropy for every proximity verification set based on users' encounters of location-tagged Wi-Fi APs (covering 88.2 % of all users) and apply x-means clustering to identify two user groups: 34 users with high mobility and 63 users with low mobility and for comparison we treat all users as a third user group with medium mobility. Based on our evaluation results, we see that the users' mobility has only a minor impact on the co-presence reasoning.

Moreover, we quantify the impact of device heterogeneity where the co-presence accuracy decreases by 47 % due to varying hardware of mobile devices within the proximity group and their different sensing ranges and sensitivities. For our analysis we enrich nearby devices defined by our proximity verification set with sensor names or device models. Different device models are treated as one device because they use the same sensor hardware, e.g., iPhone 6, iPhone 6s, and iPhone 6s Plus. We split nearby devices into subgroups according to their sensor hardware or device model to obtain a potentially higher signal similarity among devices in proximity using only the same sensor hardware neglecting different sensing ranges and sensitivities. Remote devices are treated as one device group regardless of their sensor hardware. If we are only using devices with the same sensor hardware for each proximity group, our results show that, as expected, the signal distance ratio increases: accelerometer by 1.25x, magnetometer by 1.6x, and barometer by 5.5x compared to device groups with mixed sensor hardware for co-presence reasoning.

To summarize, due to the long sensing time for a reliable co-presence result, this type of proximity estimation is actually not usable in everyday applications, i.e., the mechanism must work in a reliable fashion within a few seconds. Hence, sensor data can only further enrich the co-presence reasoning in certain situations.

3.3.3 Energy Analysis of Sensors for Co-Presence Reasoning

For a satisfying usability and motivated by the limited battery capacity of mobile devices, we perform an energy analysis on mobile devices to assess the energy demand of different sensors for co-presence reasoning. Our proximity dataset contains five Samsung Galaxy S5 devices (model: SM-G900F). Hence, as sampling device for our sensor energy measurements we use the Samsung Galaxy S5 where we replaced the detachable battery with a Monsoon high-voltage power monitor. The Monsoon device powers the smartphone with 3.85 V being able to take the energy measurements: time, voltage, and current to compute the energy $E(mJ) = U(V) \cdot I(mA) \cdot t(s) = \sum_{i=1}^{n} U_{t_i} \cdot I_{t_i} \cdot (t_{i+1} - t_i)$. The sensor measurements are taken continuously in the background for co-presence reasoning so that they are immediately available. As a result, different sensors have only a minor effect on the smartphone's runtime. For instance, the standalone sensor's energy consumption to receive GPS locations is 3193x higher compared to barometer readings, whereas the smartphone's runtime only decreases by 25 minutes. In general, the smartphone in idle state dominates the total energy consumption with 98 % compared to the sensor readings with 2 %. Hence, the co-presence reasoning running on mobile devices has no negative impact on the runtime of mobile devices with limited battery capacity. Besides that, the CPU overhead to gather the sensor measurements is negligible, no study participant reported performance problems of their mobile device that it is continuously utilized and the smartphone runtime reduces only by of 1.6 %, i.e., no higher battery drain due to the CPU utilization, when we compare the accelerometer readings with a high sampling rate of 5 s and the magnetometer readings with a low sampling rate of 5 min.

3.4 Summary

In this chapter of the thesis, we describe mechanisms to achieve fine-grained co-presence reasoning for the IoT ecosystem. We identify the currently missing context-awareness of IoT middleware as a main limiting factor in terms of availability and usability of IoT services. It is beneficial to integrate some intelligence into the IoT platform to provide relevant information and/or services depending on the current situation. Hence, we make proposals to use visible light and sensor data for co-presence reasoning among users and IoT devices regarding a practical context-awareness.

Publication 2 explores a dedicated modulation scheme for VLC inspired by Morse coding to eliminate the light flickering effect and achieve a broadcast with low processing overhead. Compared to common VLC modulation schemes using Manchester encoding,

we can achieve 8x improvement on both throughput and energy consumption. In addition, we introduce a 3D-printed custom light bulb which can be embedded as a regular light source into the infrastructure and addresses the crucial challenge faced by conventional VLC designs: usability in practical deployment. We extend our previous work in Publication 3 by a lighting configuration framework to integrate our custom light bulbs into a network for communication and signaling. Thereby, can control the spatial granularity of user proximity flexibly defined through different groups of light bulb(s) to form semantic subnetworks for different spatial regions. Publication 4 analyzes sensor data from mobile devices to support developers building better context-aware applications by a deepened understanding of which set of context information is appropriate for co-presence reasoning. We prove the assumption that sensor data of physically nearby users is similar using activity (accelerometer) and location (magnetometer, barometer).

4 iConfig: Edge-Driven IoT Device Management using Wearables and Drones

Future computing environments are embedded with many sensors and much of the deployed IoT technology is designed to be invisible. By 2025, around 68 % of all connected devices (37 billion) will be related to the IoT with a wide heterogeneity of devices, such as smartphones, sensors, embedded systems, smart meters, point-of-sale terminals, consumer electronics, and wearables [20]. The lack of awareness about spatially distributed IoT assets both limits the services that can be provided and raises concerns with respect to user privacy and system security. The aim of IoT device management is an enhanced use of public resources, improving quality of services for citizens while reducing operational costs of (public) administration. Security and governance issues are among the challenges IoT poses to organizations stemming from the widespread adoption of IoT devices, their diversity, standardization obstacles, and inherent mobility. Organizations might not know exactly which IoT devices are connected to their network, particularly caused by the trend that employees bringing their own IoT devices (BYOIoT) to the workplace. This situation threatens the security and integrity of the network and the devices. In spite of the growing demand, there is a lack of tools to seamlessly manage large IoT deployments in which ad hoc management is becoming untenable, especially for IoT devices without Internet connectivity and missing backend integration. A streamlined management process for different IoT devices is a key step for a suitable IoT device management.

In this chapter, we aim to contribute to the second question of the research problem that we stated in the introduction: **How to effectively manage IoT deployments consisting of spatially distributed, heterogeneous, wireless IoT devices?** The entire code of iConfig is available at `https://github.com/TUM-cm/iConfig` [last checked 16.06.2020].

We begin addressing this question in Publication 6 by presenting the design and implementation of an IoT device management called iConfig with a global device map for

administration and multiple edge modules intended to run on user devices, e.g., wearables, to interact with physically nearby IoT devices. Moreover, our usability study and testbed experiments highlight the difference between manual and automated IoT device configuration. The video `https://www.youtube.com/watch?v=PuqgJLW1Llw` [last checked 16.06.2020] shows our working prototype.

The edge modules in Publication 6 are designed for users interested in managing and interacting with IoT environments. In Publication 7, we extend this work by introducing an additional edge module targeted at drones being completely independent from users to further reduce the operational costs of IoT device management. In addition, we optimize the flight path using hot spots, i.e., cover multiple IoT devices at once without flying to each device individually, to find an optimum in the trade-off between limited flight time and maximum number of discovered IoT devices.

4.1 Semi-Centralized Platform for Device Management

We identified three main challenges for IoT device management. First, there is no well-defined IoT management procedure including all necessary device operations corresponding to each phase of the IoT device life cycle: device registration, configuration, monitoring, and debugging. Second, the operational costs are growing due to time consuming and error prone manual configuration of distributed IoT infrastructures. Third, we need a unified management process for IoT devices installed at various locations, especially difficult to handle in large-scale deployments. To tackle these challenges, we propose iConfig in Publication 6 that takes care of all deployed IoT devices and covers the entire aforementioned IoT device life cycle. The iConfig framework follows the principles: 1) open platform for developers to enable add-on services, e.g., software and firmware updates, 2) streamlined device management process with a minimum amount of manual tasks, and 3) automatic configuration of IoT devices avoiding misconfigurations as one of the dominant causes of system failures [125]. Our design aims at controlling the full spectrum of IoT devices from high end IoT boards to low budget BLE beacons. We focus on the management of standalone BLE beacons that represent one of the most challenging IoT devices due to a missing backend connection and require edge modules for device management. BLE beacons are small-size, battery powered wireless devices that broadcast short-range BLE messages to nearby mobile devices, e.g., smartphones [126]. The BLE messages contain information about indoor surroundings for location aware actions. Major use cases for BLE beacons are indoor localization and proxim-

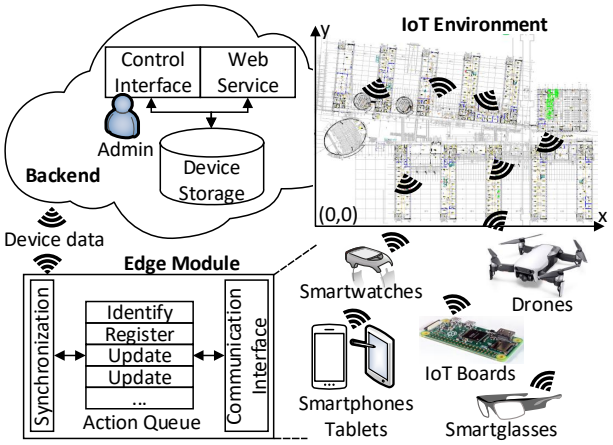


Figure 4.1: iConfig platform overview for IoT device management [7]. The edge modules run at multiple end-devices: static user-independent (IoT boards), mobile user-dependent (smartwatches, smartglasses, smartphones), and mobile user-independent end-devices (drones).

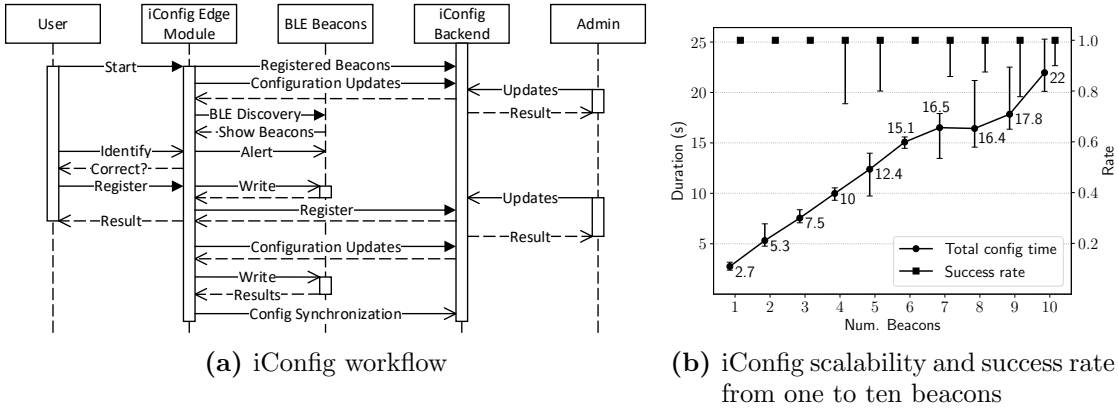
ity detection of devices and users, e.g., universities take advantage of BLE beacons to navigate students through the library, guiding them to study spaces and services [127].

4.1.1 iConfig System Architecture

The iConfig system architecture in Fig. 4.1 consists of two major modules: the mobile edge module and the backend module. Our system can identify, register, and update IoT devices (in our case BLE beacons). The edge modules allow iConfig to connect various IoT devices to its backend enforcing a unified management procedure. The iConfig backend runs at a centralized infrastructure, such as a local server or in the cloud, stores all device data and provides a REST API for communication with the iConfig edge modules. The control interface for the administrator includes status information about device functionality (e.g., broken BLE beacon), device health (e.g., battery state), and offers localization via an indoor device map. The administrator can change several device configurations at once by linking devices to groups and groups to configurations.

4.1.2 iConfig Workflow

Regarding the iConfig workflow in Fig. 4.2(a), the user's main task is to register IoT devices (in our case BLE beacons). The user holds a smartphone running the iConfig edge module and walks around to detect nearby BLE beacons. As the application starts, the edge module automatically receives the registered beacons from the iConfig backend and shows only unregistered beacons when they are discovered by the user. The registered

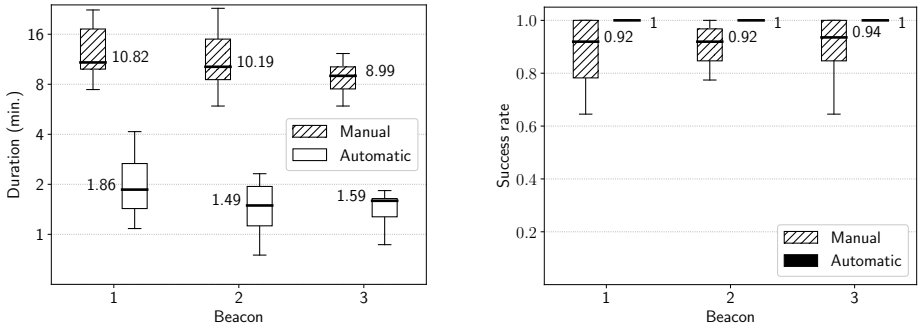


(a) iConfig workflow

(b) iConfig scalability and success rate from one to ten beacons

Figure 4.2: iConfig system details and scalability evaluation [6]

are beacons are previously discovered and registered once at the iConfig backend by the user. After discovering unregistered beacons, the user can identify one beacon at a time. A beacon shows a red light as feedback for successful identification. The user is able to register the beacon at the iConfig backend including additional information for device localization, e.g., nearest room, picture of device place. For all beacon operations, the iConfig edge module uses an action queue to synchronize automated updates of beacon settings and user actions, such as identify and register beacons. For a satisfying usability the user interactions are prioritized over automatic beacon updates. During device registration, to ensure that the device is ready to use, the edge module automatically configures the BLE beacon with a default configuration. Moreover, while registering and updating of BLE beacons, the edge module collects device maintenance data, such as battery voltage, to enable device monitoring, e.g., to detect broken beacons. Finally, the edge module synchronizes all configuration data to the iConfig backend. The device registration has to be done only once per beacon, all other actions, e.g., device updates, occur in the background without user interaction. To update settings of BLE beacons, we accumulate received signal strength indication (RSSI) for every beacon and update beacons according to descending RSSI, i.e., physically nearest beacons first. The update of a BLE beacon is independent of user actions and automatically started if two conditions are fulfilled: 1) adapted BLE configuration available from iConfig backend and 2) the beacon is currently discovered by the user by the iConfig edge module. The update process runs in the background of the iConfig edge module only noticeable by a blinking red light from the BLE beacon as a feedback of successful configuration.



(a) Configuration time per BLE beacon

(b) Configuration success rate per BLE beacon

Figure 4.3: Results of iConfig user study to highlight the difference between manual and automated device configuration [6]

4.1.3 Evaluation of System Scalability

In terms of system performance of iConfig, we analyze the configuration scalability over multiple beacons. For a dense device deployment, the scalability testbed consists of ten beacons placed in a circle with a diameter of 1 m around the smartphone running the iConfig edge module. We evaluate ten cases including one to ten beacons each over 20 rounds with BLE beacons using the maximum transmission power (5 dBm - 80 m) and maximum advertisement rate (10 Hz - 10 times a second) reflecting the worst case in terms of interference among BLE beacons. Fig. 4.2(b) shows that the configuration time grows linearly over all beacons, on average an increase of 2.2 s per beacon. Moreover, we analyze the success rate, in most cases iConfig achieved a success rate of 100 %, so that all device parameters were correctly set to the pre-defined value. With a dense device deployment and strong interference among devices, iConfig can achieve a satisfying configuration time per IoT device with high success rates.

4.1.4 User Study for Automated Device Management

We conduct a user study to explore the difference between manual and automatic configuration of IoT devices. In total, ten persons took part in the study, all PhD students or Postdocs with a strong background in computer science. First, the participants performed the manual configuration of three BLE beacons with a predefined configuration using the vendor application [128]. The following 14 parameters can be configured: a) sBeacon (proprietary format of the beacon manufacturer) including transmission power and advertisement rate of hard-coded beacon identifier, b) iBeacon with beacon identifier, transmission power, and advertisement rate, c) Eddystone beacon including beacon

identifier, broadcasted URL, and transmission power and advertisement rate for telemetry (e.g., battery voltage, device temperature), identifier, and URL packets, and d) access password. Afterwards, the participants use iConfig to register the same beacons at the backend where the default configuration was automatically written to every BLE beacon. We compute the success rate by comparing the predefined configuration with actual beacon settings. Fig. 4.3(a) shows the configuration time per beacon. The manual configuration of IoT devices using the vendor app took, on average, six times longer compared to the iConfig automated configuration with a minimal user interaction saving 83 % of the configuration time. We see a slight decrease of configuration time when the user gets more familiar with the configuration system. Fig. 4.3(b) shows the configuration success rate, i.e., how many parameters were correctly set at the BLE beacon. In case of the manual device configuration using the vendor app, the lowest success rate over all beacons was 58 % and the median is around 92 %. Only 1/3 of all manual configurations were completely correct. The manual configuration of IoT devices is time consuming and error prone. In contrast, our iConfig automated device configuration achieved for all BLE devices the success rate of 100 %. By the automatic configuration of IoT devices, we avoid misconfigurations which become one of the dominant causes of system failures. In the following survey of the study participants, the manual configuration using the vendor application was mostly rated as difficult, whereas the automatic device configuration by iConfig was rated as easy. The configuration process requires less manual interaction and is thereby faster and easier to fulfill.

The long term benefits of iConfig come from serving as platform for developers to enable add-on services: 1) monitoring of distributed IoT devices via a global device map including configuration status or localization of broken devices for replacement, 2) debugging of IoT devices via collected maintenance data, such as up-time or battery voltage, to identify malfunction devices, 3) easy distribution of software and firmware updates via a backend service to ensure up-to-date software versions and improve IoT security, and 4) parameter updates for a set of devices.

4.2 Autonomous Device Discovery and Mapping using Drones

To overcome the limitation of previous iConfig edge modules that are mobile user-dependent running on a smartphone (Publication 6) and smartglass, we extend iConfig with edge modules dedicated for drone-based IoT device management in Publication 7. This reduces the operational costs by being independent of users. Moreover, we are able to autonomously gather data for indoor device maps ensuring a reasonable service

(a) Assembly view **(b)** Schematic view

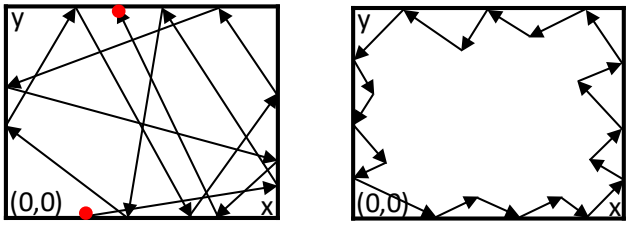
Figure 4.4: Drone platform for autonomous device management [7]

quality. In the domain of IoT device management, these comprehensive device maps of Wi-Fi and BLE devices serve as basis for device localization and monitoring. For the intended purpose, a drone is most suitable to move more freely within indoor regions at which robots move on the ground with obstacles on the way, such as tables, chairs, and staircases. Moreover, we are able to perform device presence detection based on repeated drone flights to monitor changes in the environment.

4.2.1 Platform for Autonomous Device Management

For a fully autonomous area exploration, to avoid any dependency to a ground-control station, the drone controller, e.g., smartphone, flies with the drone just like the device detection platform as shown in Fig. 4.4(a). We use the commercial off-the-shelf DJI Mavic Air drone because it can be easily controlled autonomously, is small enough to fly indoors, and is powerful enough to carry a smartphone for on-board control and an IoT board (Raspberry Pi Zero W) for device detection using wireless traces. For an easier region exploration, the drone flies at the height of 1.8 m to avoid most obstacles on the flight route, e.g., chairs, tables. The schematic view in Fig. 4.4(b) presents the smartphone controlling the drone and powering the Raspberry Pi Zero W which performs the device detection by receiving Wi-Fi probe requests during channel hopping and BLE messages from beacons.

Regarding the drone control, we use the drone's API and navigate the drone via virtual sticks which are translated into movement. Inspired by the reactive control of [129], we actively control the flight direction of the drone only when detecting spatial barriers. Our drone control supports two different strategies for area exploration including the detection and localization of wireless devices. First, we describe the random direction strategy in Fig. 4.5(a) (red dots mark the start and end position), the drone flies up to



(a) Random direction **(b)** Follow boundary

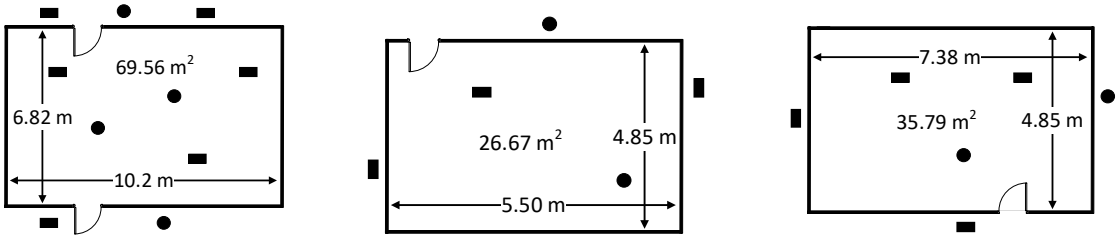
Figure 4.5: Strategies for area exploration [7]

a barrier, such as a wall, and randomly selects another direction until no barrier blocks the drone. Second, the boundary following strategy in Fig. 4.5(b) shows that the drone flies in its heading until it detects a barrier, then it turns right and left to closely follow the boundary. Besides that, we define a relative coordinate system for our university building to be able to generate device maps with positions of recognized devices. The drone starts to discover each room at a predefined origin of the room's reference frame. Afterwards, the smartphone as drone controller estimates its own position using time, flight direction, and velocity. To be able to link the encountered devices with a room position using matched log timestamps, we synchronize the time between the smartphone providing the drone's position and the Raspberry Pi identifying wireless devices. While the drone's flight, the smartphone logs the current relative position and time and the Raspberry Pi logs the encountered device's MAC address and time. After the drone's flight, we process the collected data and, for each device, we choose encounters with strongest RSSI, i.e., at this time the drone was nearest to this wireless device. Finally, we create a device list including MAC address, relative position, time, and signal strength.

4.2.2 Device Management Platform in Testbed

We evaluate our iConfig edge module dedicated for drones in different testbeds. Due to our control design of the drone, we are independent of any ground control station, the control unit, e.g., smartphone, and the device detection platform, e.g., IoT board, flies with the drone. The major drawback is the reduced flight time of the drone caused by the additional weight of the drone controller and device detection platform. Without our iConfig drone edge module the maximum flight time is 14.45 min with one battery load, with our iConfig platform the flight time decreases by 41.1 % to 8.51 min.

To analyze the two strategies for area exploration in terms of explored area over time and device localization error, our three real-world testbeds in Fig. 4.6 consist of a varying number of Raspberry PIs acting as Wi-Fi or BLE device inside and outside



(a) Show room: 6 x Wi-Fi, 4 x BLE

(b) Meeting room: 3 x Wi-Fi, 2 x BLE

(c) Office room: 4 x Wi-Fi, 2 x BLE

Figure 4.6: Test environments for autonomous area exploration (●: BLE devices and ▬: Wi-Fi devices) [7]

of each room to represent printers, BLE beacons, IoT sensor boards, etc. The random direction strategy for area exploration covers $0.94\,\mathrm{m}^2$/min and the boundary following strategy obtains $0.66\,\mathrm{m}^2$/min. This shows that our area exploration is in general too slow to discover a sufficient area within the limited flight time of one battery load. On average, to detect and localize distributed devices in our testbeds, the flight control using boundary following takes on average 1.33 min with a localization error of $1.59\,\mathrm{m} \pm 0.19\,\mathrm{m}$ compared to a 3.5 times increase of 4.67 min using random direction with a localization error of $1.67\,\mathrm{m} \pm 0.35\,\mathrm{m}$. In this regard, boundary following is faster and gains a smaller localization error. However, boundary following will only work if the boundaries lead you into the reach of all devices. With respect to limitations of our device detection, we can only recognize active wireless devices which broadcast Wi-Fi probes or BLE messages, passive devices cannot be detected. Moreover, by our experiments we recognized that the drone keeps a safety distance of 1.8 m to obstacles which restricts the indoor area exploration to medium- and large-sized regions. It is impossible to fly the drone in small rooms, e.g., $\leq 12\,\mathrm{m}^2$, or corridors with a width of two or three meters.

4.2.3 Optimizing Device Coverage in Simulation

We aim to optimize the flight route of the drone, as identified by our experiments, the drone's battery capacity is the main limiting factor to explore a larger area. We know the device positions after the initial area exploration and we can adapt the flight route to reach more IoT devices within the limited flight time using one battery load. Therefore, based on a detailed building map, we model and simulate two larger areas including a varying number of IoT devices: university lab ($564\,\mathrm{m}^2$) with 23 smaller rooms of an average room size of $25\,\mathrm{m}^2$ and a university hall ($3038\,\mathrm{m}^2$) as one large room. For the simulation, we measured the drone's velocity and energy drain rate during the

flights in our real-world testbeds. In the university lab with more spatial barriers the drone achieves a velocity of 0.9 m/s compared to the university hall with fewer obstacles achieving a speed of 1.5 m/s. The energy limit of the drone is set to $\geq 35\,\%$, hence the drone can fly back from any position in the room to a predefined position to change or recharge the battery. We randomly distribute a different number of IoT devices ranging from five devices (dense), over two devices (medium), and to one device (sparse) for an area of $25\,m^2$ similar to a single room, which is then scaled to larger area sizes. Each IoT device is either a Wi-Fi or BLE device and we randomly select a wireless range of [5, 15] m for Wi-Fi devices and [1, 7] m for BLE devices. For a simulated maintenance task the drone randomly stays at each IoT device evenly distributed between 5 and 10 s. To compute the flight path of the drone, we construct two different graphs of IoT devices: sampling-based graph and a visibility graph using the map data from our simulations. For the sampling-based graph, we generate points equidistantly in a grid pattern over the entire workspace of the drone, which is modeled based on the findings from the real-world testbeds. We connect every sampled point to all other points through an edge if the edge is not intersecting with an obstacle like a wall. In the visibility graph, the connected nodes, can see each other, this means no edge intersects with an obstacle. We apply common Dijkstra and A* path planning on these graphs representing the simulation environment to find the best flight path for each environment: a) university lab and hall, and b) device distribution: dense, medium, and sparse. In total, we compare four different path generation approaches: sampling-based graph using Dijkstra and A* path planning, and visibility graph with Dijkstra and A* path planning. Over all simulation runs, the sampling-based graph applying Dijkstra path planning performs best in terms of most discovered IoT devices and least explored area, even without utilizing a search heuristic as used by A* path planning. Moreover, to further optimize the flight path, we take advantage of hot spots to save drone's energy and reach more IoT devices with one battery load. By hovering at hot spots the drone can reach multiple IoT devices at once which is more efficient than to fly to each device individually. To find the positions of hot spots, we compute the intersection points among the wireless ranges of all IoT devices. Our simulations reveal that hot spots significantly increase the number of reachable IoT devices by 39.76 %. However, the hot spot effect decreases from a dense, over medium to a sparse number of distributed IoT devices. In addition, we see that only in the university lab with a sparse and medium number of distributed IoT devices we are able to discover all IoT devices. The university hall is simply too large to be completely explored with one drone's battery load. Hence, even in case of hot spots,

over all simulation environments it is still impossible to reach all IoT devices distributed over the entire space with one drone's battery load.

4.3 Summary

This chapter of the thesis describes results related to automated IoT device management using wearables and drones. Moreover, we enhance the human-computer interaction for IoT devices using speech control. The aim of IoT device management is an improved use of public resources, enhancing quality of services for citizens while reducing operational costs of (public) administration. For large IoT deployments ad hoc management is becoming untenable and we need a tool including a streamlined process for effective device management.

Publication 6 presents our IoT device management called iConfig with a global device map for administration and multiple edge modules intended to run on user devices, e.g., wearables, to interact with physically nearby IoT devices. Our usability study and testbed experiments reveal that the manual configuration of IoT devices took, on average, six times longer compared to iConfig automated configuration with minimal user interaction, which reflects a time saving of 83 %. Only 1/3 of all manual configurations were entirely correct. By automatic configuration of IoT devices, we avoid misconfigurations which become one of the dominant causes of system failures. To extend our work, previous edge modules are dedicated for users interested in managing and interacting with IoT environments, we introduce an additional edge module in Publication 7 targeted at drones being completely independent from users to further reduce the operational costs of IoT device management. We have implemented two different area explorations: a) random direction, the drone flies up to an obstacle like a wall and randomly chooses another direction until no obstacle blocks the drone, and b) boundary following, the drone flies in its heading until it recognizes an obstacle, then it turns right and left to follow the boundary as closely as possible. In general, our area exploration is too slow to discover a reasonable area size within the limited flight time of one battery load. Moreover, we performed simulations of indoor environments to analyze different flight path generation methods. Over all simulation runs, the sampling-based graph applying Dijkstra path planning performs best in terms of most discovered IoT devices. In addition, at hot spots the drone is able to reach multiple IoT devices at once which significantly increase the number of reachable IoT devices by 39.76 %. However, it is still not possible to reach all IoT devices distributed over the entire space with one drone's battery load.

5 iService: User-Oriented and Privacy-Aware Proximity Services

We demonstrate the practicality of our EdgeProx platform via a set of user-oriented proximity services using visible light for communication and signaling from iPresence. Our aim is to improve the user experience by automating tedious and reoccurring user tasks, e.g., authentication and authorization, and integrating social applications in everyday environments based on embedded VLC. A special role plays user privacy, which is currently underrepresented compared to more dominant security solutions, that end users are actually accepting and using the provided IoT services. Moreover, our iConfig device management from Chapter 4 supports the well-functioning of the proximity services with respect to a high service reliability by identifying and localizing broken IoT devices to replace them as soon as possible.

In this chapter, we aim to contribute to the third question of the research problem that we have stated in the introduction: **How to enable user-oriented and privacy-aware IoT services by utilizing spatial proximity awareness in the managed indoor IoT areas?** The entire code of iService is available at `https://github.com/TUM-cm/iService` [last checked 16.06.2020].

We address this question by realizing a set of different proximity services. To enable social applications, our fine-granular seamless device association from Publication 3 uses the similarity of visible light signaling in different spatial areas, which are impossible to differentiate with propagating Wi-Fi signals. Besides that, in Publication 2, we improved the user privacy in public spaces by turning around the action cycle for service discovery to be entirely initiated from the IoT environment. The users remain passive and cannot be tracked by collecting service advertisements via distance-limited VLC when approaching a service area. Furthermore, in Publication 2, we fully automated the authorization for the remote control of smart homes to improve the usability of the system's security based on out-of-band VLC to exchange secret keys.

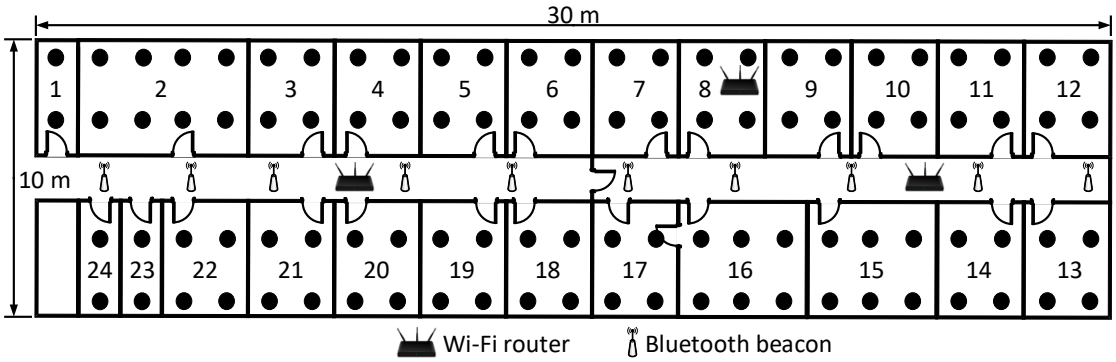


Figure 5.1: We model our university lab as simulation environment for device associations. For comparison with device localization, we take real traces of the Wi-Fi and Bluetooth environment at different positions (●) [3].

5.1 Fine-Granular Seamless Device Associations

For indoor IoT environments spontaneous device associations are particularly interesting where users establish a connection in an ad-hoc manner to enable serendipitous interaction. In Publication 3, we take advantage of the similarity of visible light signaling (VLS) provided by iPresence for device associations. The main motivation for our system comes from the fact that visible light enables spatially fine-granular device associations that are impossible to achieve with propagating Wi-Fi. In addition, our device association service is seamlessly available through our custom light bulbs integrated in the surrounding environment. The device association runs directly on our edge-driven custom light bulbs forming subnetworks which can be flexibly defined via our lighting configuration framework. Each proximity region broadcasts another random light pattern. We provide an in-depth analysis of the most crucial system parameters, including statistical and time-series tailored features of light signals and machine learning classifiers, distance metrics, and correlation metrics to classify devices into different device groups.

We use a dedicated simulator running two different simulations: 1) static device associations where no user moves and stays in the same room and 2) dynamic device associations where users are moving between different rooms receiving varying light patterns. We determine for each case the best working device association in terms of high detection accuracy and low runtime. We perform a trace-driven simulation where each association client uses three different real traces from our university lab (Fig. 5.1): Wi-Fi and Bluetooth scans, and random light patterns. We achieve a realistic simulation by emulating the network latency between the association server and the clients by a ran-

Table 5.1: Settings from parameter estimation (italic) and simulation parameters (bold) for device association [3]

Simulation	Parameters	
Static & Dynamic	**Localization classifiers**	Content-based filtering, random forest, SVM
	Sampling period localization	5 s
	Similarity metrics	Pearson, Spearman
	Similarity threshold	0.7
	Similarity equalize method	DTW
	Similarity classifiers	Random forest, extra trees, gradient boosting
	Sampling period to train similarity classifiers	50 ms
Static	**Length of light patterns**	[2, 4, 6, 8, 10]
	Number of users	[2, 3, 4, 5, 6, 7, 8, 9, 10]
Dynamic	**Association frequency**	[10, 20, 30] s
	Number of users	[3, 5, 10]
	Number of rooms	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

dom wait time before the clients send the requested environment data to the association server.

Parameter Estimation for the Simulations of Device Associations To enable our simulations, we have identified the best working parameters for device associations, which are summarized and italicized in Table 5.1 and used for static and dynamic device associations.

Regarding localization classifiers, we include a common indoor localization based on the similarity of Wi-Fi and Bluetooth signals as reference to compare the results of our device association based on light patterns. We gather a list of Wi-Fi routers and Bluetooth beacons containing MAC addresses and signal strengths (RSSI) for each measurement point (●) at our university lab as shown in Fig. 5.1. We evaluate the sampling period, i.e., how many traces are required to achieve a reasonable localization accuracy. We use supervised machine learning: support vector machine (SVM) and random forest with 10-fold cross validation. Therefore, to compare the list of Wi-Fi routers or Bluetooth beacons seen at two different measurement points, we have taken a feature subset from the work in [87], e.g., number of overlap, size of the union, Manhattan distance of

RSSI. On average, the sampling period with 5 s achieves the highest accuracy for Wi-Fi and Bluetooth localization.

With respect to similarity metrics, for distance, e.g., Euclidean, and correlation, e.g., Spearman, we need a similarity threshold from which we can infer that the devices are physically close to each other. If the result of the distance or correlation metric is above the similarity threshold, e.g., 0.8, we assume that the devices are nearby. Moreover, as many distance and correlation metrics require input data with equal length, we need an equalize method to unify the signal lengths of light patterns to compare them. Therefore, we setup a testbed comparing raw light signals with the same increasingly distorted light signal to identify the best working similarity threshold $\in [0, 1]$ and equalize method $\in$ {fill, cut, dynamic time warping (DTW)}, either to fill vectors with their average value to the longest of all vectors, cut to the shortest of all vectors, or use the DTW mapping to unify the vector lengths.

In terms of similarity classifiers, meaningful features are important for device association based on machine learning to achieve a good performance. Our feature selection identifies the features with highest entropy, i.e., information content, and lowest runtime. To find the most robust features in terms of distorted light patterns, we include light patterns with increasing white noise from 0 to 100 %. For our light patterns consisting of integer voltage values, we compute statistical features (min, max, median, var, std, mean, sum, length) and time-series tailored features via tsfresh [130]. To identify the most important statistical features we use three different machine learning models: extra trees, gradient boosting, and random forest. In contrast, tsfresh performs a time series feature extraction using scalable hypothesis tests combining 63 time series characterization methods to identify the most meaningful features from a total of 794 time series features. On this basis, we perform an offline evaluation of our device association to identify the best working sampling period for light patterns to train different classifiers. The device association uses up to ten association clients and applies 10-fold cross validation for the classifiers: extra trees, gradient boosting, and random forest. These are trained via sampling periods $\in [30, 120]$ ms for different light patterns where the sampling period of 50 ms achieves the best result, average over accuracy, precision, recall, and F1, with 0.91 over all classifiers.

Static Device Simulation of Device Associations In the static simulation no user is moving and each user remains in the same room. The association server waits until all devices are connected and starts the device association. Table 5.1 shows the parameters for the static simulation. We perform the device association using random light patterns with a varying length $\in \{2, 4, 6, 8, 10\}$, e.g., a length of four means two random light on

Table 5.2: Best working classifiers and features for device association using simulations with static and moving users [3]

Simulation	Association technique	Feature type	Runtime	Accuracy	Precision	Recall	F1
Static	SVM	Wi-Fi	2.61 s	.32	.2	.44	.26
	Random forest	Bluetooth	2.64 s	.34	.48	.52	.38
	Gradient boosting	Statistical	0.45 s	1	.75	.75	.75
	Gradient boosting	Tsfresh	4.02 s	1	.94	.94	.94
Dynamic	Content-based filtering	Wi-Fi	0.61 s	1	.78	.78	.78
	Content-based filtering	Bluetooth	0.59 s	.95	.81	.81	.81
	Extra trees	Statistical	0.26 s	.75	1	.75	.83
	Gradient boosting	Tsfresh	1.58 s	.9	1	.9	.93

and two random light off phases, and for at least two users up to ten users. By using 10-fold cross validation, Table 5.2 presents the best working device association techniques in terms of a fast reasoning and a high average over accuracy, precision, recall, and F1-score. In general, the machine learning based device association performs similarly or slightly better than the signal similarity metrics, like Spearman and Pearson. In contrast, the device localization using Bluetooth and Wi-Fi features works far worse.

Dynamic Device Simulation of Device Associations In comparison to the static device association simulation, the users are moving between different rooms in the dynamic simulation. Fig. 5.1 shows our simulation environment for device associations, where the rooms are positioned in a rectangular grid with an intra room distance of 2 m and inter room distance of 3 m and we calculate the distances among all room combinations. For each user we calculate a random path between the rooms using the duration of one simulation iteration of 20 min and distribute the time as duration of stay over the rooms using a multinomial distribution. As a result, the user's random path is a list of tuples with the duration of stay for each room where the user stays and moves to the next room if the duration of stay is expired. For example, user A has the random path: [(1, 120), (3, 300), ...] which specifies that the start position is in room 1 and after 120 s

the user moves to room 3 and stays there for 5 min, and so forth. Thereby, we randomly create user groups for each room, i.e., at which simulation time how many users are in the same room. During the simulation each user chooses a random movement speed in the range of 1.25 to 1.53 m/s (4.5–5.5 km/h) [131] for each movement between rooms. If the users are in motion they are in the corridor and not associated with any room. For device association, each room acts independently of other rooms and is associated with unique location-dependent environment data including Wi-Fi and Bluetooth scans, and light patterns. Table 5.1 shows the parameters for dynamic device simulation: association frequency, number of users, and number of rooms. Via 10-fold cross validation, Table 5.2 shows the best working device association techniques with respect to a fast runtime and a high average across accuracy, precision, recall, and F1-score. In contrast to the static simulation, the device association based on similarity metrics works in general slightly better compared to machine learning based device association. The device localization using Wi-Fi and Bluetooth features achieves a similar result.

5.2 Private Service Discovery for Smart Buildings

We take advantage of iPresence to improve the user privacy in public spaces by turning around the action cycle for the discovery of wireless networks and services. Regarding user privacy and Wi-Fi connectivity, mobile devices are not actively scanning for networks, but the devices passively listen for beacon management frames from access points which broadcast that they are there and available. To shorten the time until the wireless connectivity is available, the other direction is that whenever our mobile devices have their Wi-Fi radio interface on, they periodically try to connect to known (connected in the past) wireless access points through Wi-Fi probe-requests, including MAC address of sending device and network name of the known access point [132]. If such a wireless network is in range, the relevant access points respond with a probe-response so that the network is available without waiting for a beacon frame. The semantic information of Wi-Fi probe requests are sent completely unencrypted and makes Wi-Fi easy to use without exchanging a key or password in the beginning. However, if the information is sniffed it can help to harm the user privacy by user tracking revealing personal habits. As countermeasure, the vendors of operating systems for mobile devices apply MAC randomization of the Wi-Fi interface to prevent user tracking. Besides that, by turning around the action cycle for the discovery of wireless networks and services, we have shown a new approach where users remain entirely passive (no need of GPS, Wi-Fi or BLE discovery) collecting advertisements via VLC when approaching the service area,

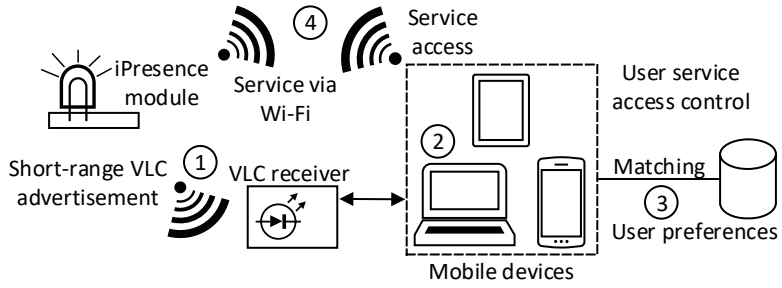


Figure 5.2: System overview for private service discovery [2]

without revealing users' private information, e.g., location tags, wireless interface meta data. Moreover, we can refine the range of service discovery based on VLC, compared to service announcements over Wi-Fi or Bluetooth, by reducing the 'visibility' of service advertisements only to what is immediately relevant in the vicinity. For example, we can realize a spatially fine-grained service advertisement for commercial coupons in a shopping mall with a dense distribution of shops.

Our private service discovery scheme is illustrated in Fig. 5.2. The principle is to use short-range VLC advertisements ① dedicated for proximity services. The service advertisement is encrypted using the lightweight Speck cipher [133] and includes a location identifier, the password to access the service and a description for the user interface to explain which functionality is provided by the service. To be specific, our VLC using LocalVLC data encoding lacks support for binary data. Therefore, we apply Base16 encoding to transmit non-alphabetical data, such as encrypted service advertisements. The VLC receiver obtains and processes the VLC transmitted service advertisements which are accumulated over time and shown to users ②. Users are able to define allowed services for specific times and locations. Based on this, the user preferences are matched with the collected VLC service advertisements ③ to decide whether the service is used and carry out operations, e.g., turning on wireless interfaces or GPS, to access certain services. If the user allows the advertised service, the device use the service through, for example, a standard radio-based communication, such as Wi-Fi ④. For instance, in the domain of smart buildings, our scheme for private service discovery can be used to proactively advertise services of the smart building, such as indoor navigation to the next printer (office environment) or the control of the sun blind.

Verifier		**Proofer**
Secret key $k \leftarrow X$		
Interval $i = 30\,s$		
	$\xleftrightarrow{\text{sync. clock}}$	
	$\xrightarrow{k, i}$	
		$x \leftarrow \text{TOTP}\,(k, i)$
		$\text{token}_1 \leftarrow x\,(t_1)$
	$\xleftarrow{\text{token}_1}$	
$x \leftarrow \text{TOTP}\,(k, i)$		
$\text{token}_2 \leftarrow x\,(t_2)$		
$y \leftarrow \text{verify}\,(\text{token}_1, \text{token}_2)$		
	$\xrightarrow{y}$	

(a) Standard TOTP used for two-factor authentication to share a secret and generate a timely restricted authentication token

Verifier		**Proofer**
Secret key $k \leftarrow X$		
Interval $i = 30\,s$		
$x \leftarrow \text{TOTP}\,(k, i)$		
$\text{token} \leftarrow x\,(t_1)$		
	$\xrightarrow{\text{token } \textbf{via VLC}}$	
$\text{nonce} \leftarrow X$		
	$\xrightarrow{\text{nonce}}$	
		$c_1 \leftarrow \text{Speck}\,(\text{key=token})$
		$n_1 \leftarrow \text{increment}\,(\text{nonce})$
	$\xleftarrow{e_1 \leftarrow \text{encrypt}\ (n_1, c_1)}$	
$c_2 \leftarrow \text{Speck}\,(\text{key=token})$		
$n_2 \leftarrow \text{increment}\,(\text{nonce})$		
$e_2 \leftarrow \text{encrypt}\,(n_2, c_2)$		
$y \leftarrow \text{verify}\,(e_1, e_2)$		
	$\xrightarrow{y}$	

(b) Adapted authorization scheme which utilizes time-based one-time passwords as encryption keys and integrates a challenge and response mechanism for automated authorization

Figure 5.3: Scheme comparison: standard scheme for two-factor authentication using TOTP and our adapted scheme which automates the user authorization by taking advantage of challenge and response

5.3 Automated Authorization for Smart Homes

The idea of a smart home involves a communication network that connects the key electrical appliances and services, and allows to be remotely controlled, monitored, or accessed. Typically, the web interface of the home control is accessible through the Internet. To secure the access, some systems use only one authentication factor, user name and password, more secure home control gateways utilize two-factor authentication. The second authentication uses an additional secret, something the user has, e.g., token [134], ambient sound [135], or something a user is, e.g., biometric characteristics. Widely used are time-based one-time passwords (TOTPs) wherefore the user previously exchanged a shared secret. The user generates a TOTP based on the shared secret and current time which is verified at the home control hub. The user has to manually generate new passwords for each authorization attempt. In contrast, we fully automate the key management for the remote control of smart homes to avoid time-consuming user interactions and improve the usability of system's security. The differences between our adapted TOTP scheme and a standard TOTP scheme are highlighted in Fig. 5.3. In the standard TOTP scheme, the verifier generates a random secret key and shares the secret key and the validity period with the proofer. On this basis, the proofer generates a TOTP and sends it to the verifier. The verifier performs the same action and compares the two tokens. The user is successfully authenticated if the two tokens are equal. It is important that the clock between verifier and proofer is synchronized because the generated tokens are based on the current time and restricted to a validity period. In practice, the standard TOTP scheme requires ongoing manual interaction between the user and access control, the secret key has to be exchanged, e.g., via a QR code, and the user has to manually generate a new authentication token. In comparison, for our adapted TOTP scheme, we automated the exchange of the secret key by using an out-of-band channel, in our case VLC. For an adversary it is harder to intercept the VLC due to a spatially restricted communication range. The VLC transmitted token is used as encryption key for an additionally exchanged nonce. The integrated challenge-response mechanism enables the automated authorization without manual user interaction.

From a system perspective, we categorize the functionality of the home control into sensitive, e.g., open the entrance door, and standard control functions, such as light on and off. The standard control functions can be used remotely and at home. In contrast, the sensitive control functions can only be used at home using an automated authorization scheme to enhance the usability of the system's security. The home control gateway consists of two independent action flows, the distribution of light tokens and

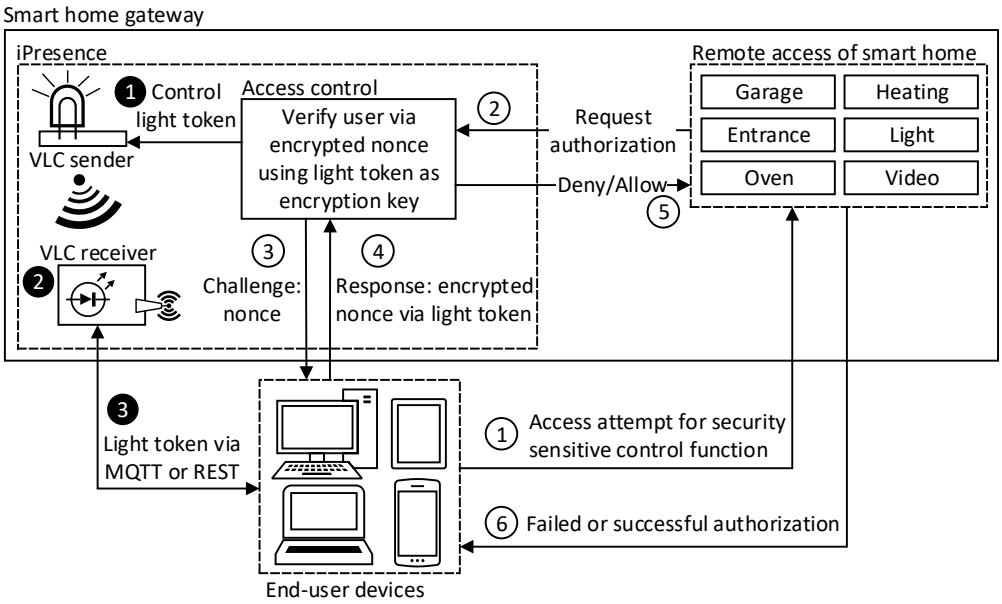


Figure 5.4: Key management for the remote control of smart homes [2]

the user authorization at the remote access of the smart home. For user authorization, the access control generates encryption keys using TOTP and broadcasts it via the VLC sender ❶. The VLC receiver ❷ continuously receives light transmitted data and selects the up-to-date encryption key based on a majority rule, most frequently received message within one message buffer. The user's device is cable-bound to the light receiver ❸ and receives light tokens. On this basis, the user can attempt to access a security sensitive control function via our Android app for smart home control ①. That triggers the remote control of the smart home to request an authorization from the access control ②. The access control sends a nonce to the end user device ③, the Android app for smart homes increments the received nonce and encrypts it via the Speck cipher using the out-of-band light token as encryption key ④. The access control of the smart home gateway performs the same action and compares the encrypted nonce from the end user device with the own incremented and encrypted nonce. In case the encrypted nonces are equal ⑤, the access control grants access to the security sensitive home control function, otherwise it denies the access. Finally, the Android app for smart home control ⑥ displays the authorization result to the end user.

We evaluate our automated authorization for smart homes in terms of robustness and efficiency. With respect to robustness, we measure the success rate of user authorizations with a varying token refresh period in the range of 5 to 60 s, i.e., the duration after the home control generates and broadcasts new light tokens for authorization. The success

rate of 5,000 user authorizations with changing token refresh period: 86.68 % (5 s), 93.3 % (10 s), 97.62 % (30 s), and 98.84 % (60 s). The authorization fails due to the latency to receive a light token, each time we change the light token, for a short period of time the users' client has not the up-to-date valid light token. The less frequently we change the broadcasted light token the more successful are the user authorizations. Regarding efficiency of our solution, in case of a successful user authorization, the duration ranges between 0.15 to 0.2 s until the home control functionality is available for the user.

5.4 Summary

This chapter serves to demonstrate the practicality of our IoT platform via a set of user-oriented proximity services using visible light for communication and signaling. Our aim is to improve the user experience by automating tedious and reoccurring user tasks, e.g., authorization, and integrating social applications in everyday environments based on embedded VLC.

Our fine-granular seamless device association from Publication 3 uses the similarity of visible light signaling in different spatial areas, which are impossible to differentiate with propagating Wi-Fi signals. We generate random light patterns for each proximity region to allow device association. Based on statistical features and time-series tailored features, we used machine learning classifiers and distance and correlation metrics to classify devices into different devices groups for data sharing. Therefore, we performed two trace-driven simulations for a) static device-to-device association, no user moves and stays in the same room, and b) dynamic device-to-area association, users are moving between different rooms receiving varying light patterns. In general, machine learning-based signal similarity performs best compared to distance and correlation metrics. Besides that, in Publication 2, we improved the user privacy in public spaces where users remain passive by collecting service advertisements via distance-limited VLC when approaching a service area. Users remain entirely passive without any need to associate to a certain network and no need of Wi-Fi or BLE discovery. Users can define preferred services for specific times and locations. On this basis, the user preferences are matched with the collected service advertisements for carrying out operations, such as turning on wireless interfaces, to access certain services of smart buildings, e.g., indoor navigation. Furthermore, in Publication 2, we fully automated the authorization for the remote control of smart homes to improve the usability of the system's security. We categorize the functionality of the home control into sensitive, e.g., open the entrance door, and standard control functions, such as light on and off. The sensitive control functions can

only be used at home via an adapted authorization scheme using VLC as an out-of-band channel to exchange secret keys and integrating a challenge-response mechanism for on-demand access requests. Thereby, we avoid the need for ongoing manual interaction between the user and home control for authorization. Our system evaluation with 5,000 authorization attempts showed that the success rate ranges from 84 % to 99 % and it takes between 0.15 s to 0.2 s until the home control functionality is available for the user.

6 Conclusion and Future Work

In this dissertation, we proposed our ready-to-deploy EdgeProx platform to be fully integrated in everyday living environments, such as home, public transport, workplace, acting as open platform to enrich IoT ecosystems. On this basis, we improve the quality of life of users through novel proximity services, like seamless device association, and facilitate the life of authorities to easily manage spatially deployed IoT devices and maintain a high service quality. In the following, we revisit the research questions posed at the beginning of this thesis in the introduction, and discuss the achievements as well as the shortcomings of the presented solutions and highlight potential future work.

RQ1: How to efficiently determine the spatial proximity of users without disclosing their locations? (Related publications: 1, 2, 3, 4, and 5) For active co-presence reasoning of iPresence, as a system module of EdgeProx, LocalVLC [2] introduced a custom light bulb that allows us to replace existing illumination infrastructure to simultaneously use visible light for illumination and communication/signaling. Moreover, our novel Morse code-inspired modulation scheme can operate on off-the-shelf LEDs with low energy overhead. It can effectively overcome the light flickering effect by encoding data into high frequency light pulses without requiring extra processing hardware, such as FPGAs or micro-controllers. As next step, DevLoc [3] integrated the custom light bulbs into a semi-centralized lighting configuration framework being able to manage the broadcasted light information. In addition, we can flexibly define different groups of light bulb(s), i.e., proximity regions, where we adopted a master-slave principle to form semantic subnetworks to cover larger rooms or across multiple rooms. Hence, we can control the spatial granularity of user proximity to overcome the main disadvantage of location tags, where the user proximity is entirely dependent on the type of location tag. As a possible extension of iPresence, we showed that location and activity related sensor data from mobile devices is suitable for passive co-presence reasoning [4]. Moreover, we highlighted the negative impact of device heterogeneity caused by sensor data obtained from device sensors with varying sensing ranges and sensitivities, and we found that the user mobility has only a negligible effect on the co-presence reasoning. Finally,

our energy analysis outlined that the co-presence reasoning running on mobile devices has no negative impact on the runtime of mobile devices with limited battery capacity.

Visible light enables fine-granular co-presence reasoning, e.g., per room, which is impossible to achieve by using radio-based communication, like Wi-Fi. On the other hand, VLC and VLS are negatively influenced by environmental conditions, such as ambient light, e.g., direct sun light, leading to a reduced data rate or impossible communication. Moreover, visible light as communication medium requires line-of-sight between the entities and suffers from limited end-device support. Most user devices and IoT environments do not have sufficient hardware capabilities for real-time processing of light signals and require add-on hardware, like a USB dongle, to process light signals. Thereby, the everyday usage is limited due to a restricted user mobility and visible light is mainly used in research testbeds. As future work, to enhance the end-device integration for VLC, our proposed MEC^2-Hub [5] extends multipath protocols, such as multipath TCP (MPTCP), to support multiple communication paths via different communication media combining visible light with radio-based communication, like Wi-Fi or Bluetooth, on the network stack level. Each network subflow in MEC^2-Hub can use a combination of physical transmission media, such as visible light and Wi-Fi, with varying properties regarding transmission range and data rate. Ultrasound communication is another transmission medium with similar sensitivity on spatial barriers like visible light and less influenced by environmental conditions. However, in practice the use is severely limited due to a low bit rate (64 bit/s) and bad energy efficiency. Based on our own energy measurements, compared to Bluetooth, VLC consumes 124x more and ultrasound 7343x more energy to transmit the same amount of information. Hence, we need novel data encoding schemes and specialized audio hardware, e.g., microphone and speaker, for a practical use. Regarding passive co-presence reasoning, the main limitation is the long time needed to gather enough sensor data to ensure a reliable proximity result whether somebody or something is nearby. Moreover, the large hardware diversity of mobile device sensors is a major challenge for a well functioning co-presence reasoning. This leads to a negative impact on user experience: Passive co-presence reasoning can only be a supplement, to active co-presence reasoning using visible light, to overcome problems in certain situations achieving a more reliable co-presence reasoning. In the long term future, we envision that iPresence will enable further novel applications based on the unique feature of a fine-grained co-presence reasoning and our design choice to make the co-presence reasoning available as service to be easily usable by developers.

RQ2: How to effectively manage IoT deployments consisting of spatially distributed, heterogeneous, wireless IoT devices? (Related publications: 6 and 7)

As second system module for EdgeProx, iConfig [6] provides a framework for automated IoT device management. The backend module of iConfig runs at a centralized infrastructure, such as a local server or in the cloud, and stores all device data. The control interface with a global indoor map of devices includes information about device updates, device functionality (e.g., broken), device health (e.g., battery capacity), and device location. We use programmable edge modules to connect various heterogeneous IoT devices to the management backend enforcing a unified configuration procedure. The edge modules are designed to run on multiple end-devices that are static user-independent (IoT boards) for continuous access to IoT devices, mobile user-dependent (smartphones, smartwatches, smartglasses) combined with speech control for fluent interaction with surrounding IoT devices, and mobile user-independent (drones) for autonomous IoT device management [7]. Our evaluation revealed that the manual configuration of IoT devices takes several times longer compared to the iConfig automated configuration based on a minimal user interaction. Moreover, automatic configuration of IoT devices avoids misconfigurations which can become one of the dominant causes of system failures; most manual device configurations were faulty. We placed a special emphasis on an easy and mostly automated detection and integration of new IoT devices at the management backend. Our user study showed that speech control on wearables improves the situation, besides the conventional screen-keyboard setup, to fluently and naturally interact with surrounding IoT devices. Moreover, we performed a feasibility study over different testbeds to use drones in indoor environments to achieve an entirely autonomous device management in terms of minimal operational costs. We were able to detect and localize most IoT devices, i.e., Wi-Fi and Bluetooth devices, from wireless traces within a few minutes obtaining a reasonable small localization error. To optimize the drone's flight path reaching more IoT devices within a limited flight time, our simulation of indoor environments revealed that to visit only hot spots, which cover multiple IoT devices at once, significantly increases the number of discovered IoT devices.

The architecture of iConfig is flexible enough to address the heterogeneity of IoT environments. Based on our experiments and a user study, we see that speech control on wearables is a step towards a better human-computer interaction for IoT devices to improve the user experience. However, it is still not an optimal solution, due to a small screen of the smart glass or loud ambient noise which may make speech recognition impossible. Hence, a future research direction is to explore additional hardware devices, e.g., body implants, targeted for users to easier interact with IoT devices. With respect to a fully automated device management, using drones to discover indoor environments looks like a promising direction, but we need more specialized, smaller drones which are

able to cope with narrow spaces. Otherwise, the obstacle detection of the drone will prevent it from exploring the entire space. The main limiting factor of space exploration is the small battery capacity of the drone with a flight time of about 20 minutes. For future work, to explore more space, with one drone we need more powerful batteries or reduce the weight of our iConfig platform for drone control and device detection and fully integrate it into a specialized configuration drone for a longer flight time. Otherwise, we can simultaneously fly multiple drones in different areas and merge the gathered data to generate an overall device map. Furthermore, an additional strategy for area exploration, the so-called hybrid area exploration, can be beneficial to gain further insights about the drone's flight path and precision of estimated IoT device locations. The hybrid control mode applies by default the random direction strategy, the drone flies up to an obstacle, like a wall, and randomly chooses another direction until no obstacle blocks the drone. When during movement the drone detects an increasing strength of a wireless signal, we allow the drone to follow this signal. At the peak of the signal strength the drone falls back to the random direction control mode. In the long term future, the global device map of distributed IoT devices from iConfig serves as platform for developers to enable add-on services: 1) monitoring of distributed IoT devices to localize devices for replacement, 2) debugging of IoT devices using collected maintenance data, such as uptime or battery voltage, to identify malfunctioning devices, 3) software and firmware updates distributed via backend service to ensure up-to-date software versions and improve IoT security, and 4) efficient parameter updates for a set of devices. Especially useful for IoT environments, with the location information of the global device map, we can associate sensor streams from IoT boards with a location and facilitate data merging and filtering, in case we have the same information from multiple IoT boards placed in a similar region. Hence, we are able to manage IoT data streams by reducing the massive amounts of data generated from IoT devices to a meaningful subset for useful decisions.

RQ3: How to enable user-oriented and privacy-aware IoT services by utilizing spatial proximity awareness in the managed indoor IoT areas? (Related publications: 2 and 3) To realize multiple practical proximity services, iService combines traditional radio-based communication, like Wi-Fi, with active co-presence reasoning using visible light from iPresence. The services benefit from the sensitivity on physical barriers of visible light as transmission medium. We especially address user privacy in the IoT ecosystem, which is currently underrepresented, but important so that end users are accepting and using the provided IoT systems. For example, we used VLC as a distance-bounding out-of-band channel to improve the usability of security solutions,

i.e., to automate tedious and reoccurring user tasks for authorization [2], to achieve a higher user satisfaction. Moreover, we improved the user privacy in public spaces by turning around the action cycle for service discovery to be entirely initiated from the IoT environment [2]. The user cannot be tracked because the wireless interfaces of the user device are disabled. The user decides based on distance-limited service advertisements and predefined user preferences to enable the wireless interface to use the service. Besides that, based on seamless device association [3], we can enable a large number of social applications to fulfill ad hoc information needs. For instance, Alice is a tourist, rides on the subway and wants to ask locals for the best way to the museum.

The current technologies make the IoT concept feasible by combining edge and cloud computing from more centralization to more immersive distribution, from bigger and farther away clouds to computation and control closer to sensors, actuators, and users [16, 40]. Cloud computing provides massive storage, heavy duty computation, global coordination, and wide-area connectivity, complemented by edge computing with real time processing, rapid innovation, and user-centric services [40]. As trillions of things and sensors will be connected, it is necessary to have an appropriate architecture that permits easy connectivity, control, communications, and useful applications [12]. Scalability issues often arise at different levels with respect to data transfer and networking, data processing and management as more and more physical objects are connected to the network [26]. To address these issues, it is important to embed smart controls into the IoT architectures: a) context-aware computing techniques to better understand sensor data and help decide what data needs to be processed and b) bringing artificial intelligence into things and communication networks for self-configuration, self-optimization, self-protection, and self-healing [26]. From the viewpoint of communication and networking, the IoT is a complicated heterogeneous network to connect various types of networks through various communication technologies [26]. To limit the effort to integrate IoT with existing IT systems or legacy systems, we need a widely accepted platform hiding the heterogeneity of networks and communication technologies and provides a naming service to various applications for a unified information infrastructure [26]. To self-organize networks and data flows, there is a strong interest to use social networking to enhance the communications among different IoT devices [26]. Many IoT applications will be based on a deployed sensing, actuation, and communication platform building a network of things [12]. In these deployments, a crucial challenge is to maintain a long-lived and dynamic system, where local interactions can lead to inconsistency of global system states [12, 40]. One possible architectural approach is to borrow the app store concept from the smartphone world for an unbounded development of novel applications

[12]. This makes it easy to install and run new apps on the infrastructure, for example, to automatically order food, monitor and control heart rate, or predict an impending medical problem for early treatment [12]. Please note that an app can be part of the system state, but unlikely all of it. Humans will often be the integral parts of the IoT system so that humans and things will operate synergistically [12]. This requires more research regarding human-in-the-loop control as modeling human behaviors is challenging due to the complex physiological, psychological, and behavioral aspects of human beings [12].

The ubiquity and interactions involved in the IoT enables many useful services for individuals and bears the risk of violating user privacy [12]. Hence, the IoT paradigm must be able to express users' requests for data access and enforce privacy policies by individual IoT applications or the IoT infrastructure, such that the requests can be verified against the policies in order to decide if they should be granted or denied [12]. Moreover, security attacks are a severe problem for the IoT because of the limited performance of the devices being used, the physical accessibility to sensors, actuators, and objects, the system openness, and that most devices communicate wirelessly [12].

The following points are important for the acceptance and widespread adoption of new IoT technologies and services. We need green IoT technologies for a minimal resource consumption and IoT must consider information security and data privacy to ensure user consent which is difficult to achieve in the IoT because of its deployment, mobility, and complexity [26]. Moreover, hampered by the rapid IoT growth, standardization is required to lower the entry barriers for new service providers and improve the interoperability, compatibility, and reliability of different applications and systems [26]. To address the above challenges, we need fundamental research across networking, device hardware, pricing, human-computer interface, and data science in both industrial and academic laboratories [16, 40]. At this point, the IoT research being done is mainly focused on technology. Once the IoT technology matures, the research needs to broaden into the fields of management, operations, law, economics, and sociology [136].

Bibliography

[1] M. Haus, M. Waqas, A. Y. Ding, Y. Li, S. Tarkoma, and J. Ott. Security and Privacy in Device-to-Device (D2D) Communication: A Review. *IEEE Communications Surveys & Tutorials*, 19(2):1054–1079, 2017. doi:10.1109/COMST.2017.2649687.

[2] M. Haus, A. Y. Ding, and J. Ott. LocalVLC: Augmenting Smart IoT Services with Practical Visible Light Communication. In *Proceedings of the 20th IEEE International Symposium on 'A World of Wireless, Mobile and Multimedia Networks' (WoWMoM)*, pages 1–9, 2019. doi:10.1109/wowmom.2019.8793022.

[3] M. Haus, J. Ott, and A. Y. Ding. DevLoc: Seamless Device Association using Light Bulb Networks for Indoor IoT Environments. In *Proceedings of the Fifth IEEE/ACM International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 231–237, 2020. doi:10.1109/IoTDI49375.2020.00030.

[4] M. Haus, A. Y. Ding, and J. Ott. Multimodal Co-Presence Detection with Varying Spatio-Temporal Granularity. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 1–7, 2020. doi:10.1109/PerComWorkshops48775.2020.9156105.

[5] M. Haus, A. Y. Ding, Q. Wang, J. Toivonen, L. Tonetto, S. Tarkoma, and J. Ott. Enhancing Indoor IoT Communication with Visible Light and Ultrasound. In *Proceedings of the 53rd IEEE International Conference on Communications (ICC)*, pages 1–6, 2019. doi:10.1109/icc.2019.8762001.

[6] M. Haus, A. Y. Ding, and J. Ott. Managing IoT at the Edge: The Case for BLE Beacons. In *Proceedings of the 3rd ACM MobiCom Workshop on Experiences with the Design and Implementation of Smart Objects (SmartObjects)*, pages 41–46, 2017. doi:10.1145/3127502.3127510.

[7] M. Haus, J. Krol, A. Y. Ding, and J. Ott. Feasibility Study of Autonomous Drone-based IoT Device Management in Indoor Environments. In *Proceedings of the*

1st ACM SIGCOMM Workshop on Mobile AirGround Edge Computing, Systems, Networks, and Applications (MAGESys), pages 1–7, 2019. doi:10.1145/3341568.3342105.

[8] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context Aware Computing for the Internet of Things: A Survey. *IEEE Communications Surveys & Tutorials*, 16(1):414–454, 2014.

[9] K. Ashton. That 'Internet of Things' Thing. *RFiD Journal*, 2009.

[10] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, 17(4):2347–2376, 2015.

[11] D. Miorandi, S. Sicari, F. de Pellegrini, and I. Chlamtac. Internet of Things: Vision, Applications and Research Challenges. *Ad Hoc Networks*, 10(7):1497–1516, 2012.

[12] J. A. Stankovic. Research Directions for the Internet of Things. *IEEE Internet of Things Journal*, 1(1):3–9, 2014.

[13] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, 1(1):22–32, 2014.

[14] IDC. The Growth in Connected IoT Devices Is Expected to Generate 79.4ZB of Data in 2025, According to a New IDC Forecast, 2019. URL: https://www.idc.com/getdoc.jsp?containerId=prUS45213219 [last checked 16.06.2020].

[15] Statista. Industrial Internet of Things Market Size Worldwide from 2017 to 2025, 2020. URL: https://www.statista.com/statistics/611004/global-industrial-internet-of-things-market-size/ [last checked 16.06.2020].

[16] L. Atzori, A. Iera, and G. Morabito. The Internet of Things: A Survey. *Computer Networks*, 54(15):2787–2805, 2010.

[17] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[18] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini. Convergence of MANET and WSN in IoT Urban Scenarios. *IEEE Sensors Journal*, 13(10):3558–3567, 2013.

[19] A. S. Hornby, S. Wehmeier, and M. Ashby, editors. *Oxford Advanced Learner's Dictionary of Current English*. Oxford University Press, 7 edition, 2005.

[20] Ericsson. Mobility Visualizer, 2019. URL: `https://www.ericsson.com/en/mobility-report/mobility-visualizer?f=13&ft=3&r=1&t=18&s=9,10&u=1&y=2014,2024&c=1` [last checked 16.06.2020].

[21] N. Banerjee, S. Agarwal, P. Bahl, R. Chandra, A. Wolman, and M. Corner. Virtual Compass: Relative Positioning to Sense Mobile Social Interactions. In *Pervasive Computing*, volume 6030 of *Lecture Notes in Computer Science*, pages 1–21. Springer, Berlin, 2010.

[22] Statista. New Internet of Things (IoT) Connections by 2025, 2020. URL: `https://www.statista.com/statistics/1101127/new-iot-connections-by-2025/` [last checked 16.06.2020].

[23] G. Fodor, E. Dahlman, G. Mildh, S. Parkvall, N. Reider, G. Miklós, and Z. Turányi. Design Aspects of Network Assisted Device-to-Device Communications. *IEEE Communications Magazine*, 50(3):170–177, 2012.

[24] A. Botta, W. de Donato, V. Persico, and A. Pescapé. Integration of Cloud Computing and Internet of Things: A Survey. *Future Generation Computer Systems*, 56:684–700, 2016.

[25] Lee and K. Lee. The Internet of Things (IoT): Applications, Investments, and Challenges for Enterprises. *Business Horizons*, 58(4):431–440, 2015.

[26] L. D. Xu, W. He, and S. Li. Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics*, 10(4):2233–2243, 2014.

[27] I. Chlamtac, M. Conti, and J. J.-N. Liu. Mobile Ad Hoc Networking: Imperatives and Challenges. *Ad Hoc Networks*, 1(1):13–64, 2003.

[28] L. Pelusi, A. Passarella, and M. Conti. Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks. *IEEE Communications Magazine*, 44(11):134–141, 2006.

[29] P. M. Mell and T. Grance. SP 800-145. The NIST Definition of Cloud Computing. URL: `https://csrc.nist.gov/publications/detail/sp/800-145/final` [last checked 16.06.2020].

[30] Q. Zhang, L. Cheng, and R. Boutaba. Cloud Computing: State-of-the-Art and Research Challenges. *Journal of Internet Services and Applications*, 1(1):7–18, 2010.

[31] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta. Secure Integration of IoT and Cloud Computing. *Future Generation Computer Systems*, 78:964–975, 2018.

[32] Cisco. Global Cloud Index Projects Cloud Traffic to Represent 95 Percent of Total Data Center Traffic by 2021, 2020. URL: `https://newsroom.cisco.com/press-release-content?type=webcontent&articleId=1908858` [last checked 16.06.2020].

[33] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie. Mobile Edge Computing: A Survey. *IEEE Internet of Things Journal*, 5(1):450–465, 2018.

[34] H. Chang, A. Hari, S. Mukherjee, and T. V. Lakshman. Bringing the Cloud to the Edge. In *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 346–351, 2014.

[35] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal*, 4(5):1125–1142, 2017.

[36] X. Sun and N. Ansari. EdgeIoT: Mobile Edge Computing for the Internet of Things. *IEEE Communications Magazine*, 54(12):22–29, 2016.

[37] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Communications Surveys & Tutorials*, 19(4):2322–2358, 2017.

[38] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

[39] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere. Edge-centric Computing: Vision and Challenges. *ACM SIGCOMM Computer Communication Review*, 45(5):37–42, 2015.

[40] M. Chiang and T. Zhang. Fog and IoT: An Overview of Research Opportunities. *IEEE Internet of Things Journal*, 3(6):854–864, 2016.

[41] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog Computing and Its Role in the Internet of Things. In *Proceedings of the 1st Workshop on Mobile Cloud Computing (MCC)*, pages 13–15, 2012.

[42] N. Kato. On Device-to-Device (D2D) Communication [Editor's Note]. *IEEE Network*, 30(3):2, 2016.

[43] A. Asadi, Q. Wang, and V. Mancuso. A Survey on Device-to-Device Communication in Cellular Networks. *IEEE Communications Surveys & Tutorials*, 16(4):1801–1819, 2014.

[44] M. Wang and Z. Yan. A Survey on Security in D2D Communications. *Mobile Networks and Applications*, pages 1–14, 2016.

[45] R. Alkurd, R. M. Shubair, and I. Abualhaol. Survey on Device-to-Device Communications: Challenges and Design Issues. In *Proceedings of the 12th IEEE International New Circuits and Systems Conference (NEWCAS)*, pages 361–364, 2014.

[46] J. Liu, N. Kato, J. Ma, and N. Kadowaki. Device-to-Device Communication in LTE-Advanced Networks: A Survey. *IEEE Communications Surveys & Tutorials*, 17(4):1923–1940, 2015.

[47] F. Ghavimi and H.-H. Chen. M2M Communications in 3GPP LTE/LTE-A Networks: Architectures, Service Requirements, Challenges, and Applications. *IEEE Communications Surveys & Tutorials*, 17(2):525–549, 2015.

[48] Y. Zou, X. Wang, and W. Shen. Physical-Layer Security with Multiuser Scheduling in Cognitive Radio Networks. *IEEE Transactions on Communications*, 61(12):5103–5113, 2013.

[49] J. Cao, M. Ma, H. Li, Y. Zhang, and Z. Luo. A Survey on Security Aspects for LTE and LTE-A Networks. *IEEE Communications Surveys & Tutorials*, 16(1):283–302, 2014.

[50] X. Lin, J. Andrews, A. Ghosh, and R. Ratasuk. An Overview of 3GPP Device-to-Device Proximity Services. *IEEE Communications Magazine*, 52(4):40–48, 2014.

[51] M. Haus. System Approach Towards Private Proximity Services. In *Adjunct Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp Adjunct)*, pages 417–422, 2016. `doi:10.1145/2968219.2971352`.

[52] K. G. Shin, Xiaoen Ju, Zhigang Chen, and Xin Hu. Privacy Protection for Users of Location-based Services. *IEEE Wireless Communications*, 19(1):30–39, 2012.

[53] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel. A Classification of Location Privacy Attacks and Approaches. *Personal and Ubiquitous Computing*, 18(1):163–175, 2014.

[54] L. Šikšnys, J. R. Thomsen, S. Šaltenis, and M. L. Yiu. Private and Flexible Proximity Detection in Mobile Social Networks. In *Proceedings of the 11th International Conference on Mobile Data Management (MDM)*, pages 75–84, 2010.

[55] X. Lin, H. Hu, H. P. Li, J. Xu, and B. Choi. Private Proximity Detection and Monitoring with Vicinity Regions. In *Proceedings of the 12th International ACM Workshop on Data Engineering for Wireless and Mobile Access (MobiDE)*, pages 5–12, 2013.

[56] B. Mu and S. Bakiras. Private Proximity Detection for Convex Polygons. In *Proceedings of the 12th International ACM Workshop on Data Engineering for Wireless and Mobile Access (MobiDE)*, pages 36–43, 2013.

[57] M. Knappmeyer, S. L. Kiani, E. S. Reetz, N. Baker, and R. Tonjes. Survey of Context Provisioning Middleware. *IEEE Communications Surveys & Tutorials*, 15(3):1492–1519, 2013.

[58] S. Brands and D. Chaum. Distance-Bounding Protocols: Extended Abstract. In *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques (Eurocrypt)*, pages 344–359, 1993.

[59] A. Francillon, B. Danev, and S. Capkun. Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars. In *Proceedings of the 18th Network and Distributed System Security Symposium (NDSS)*, pages 1–15, 2011.

[60] G. P. Hancke and M. G. Kuhn. An RFID Distance Bounding Protocol. In *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM)*, pages 67–73, 2005.

[61] J.-Y. Lee and R. A. Scholtz. Ranging in a Dense Multipath Environment using an UWB Radio Link. *IEEE Journal on Selected Areas in Communications*, 20(9):1677–1683, 2002.

[62] S. Čapkun, L. Buttyán, and J.-P. Hubaux. SECTOR: Secure Tracking of Node Encounters in Multi-Hop Wireless Networks. In *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, pages 21–32, 2003.

[63] C. Meadows, P. Syverson, and L. Chang. Towards More Efficient Distance Bounding Protocols for Use in Sensor Networks. In *Proceedings of the International Conference on Security and Privacy in Communication Networks (SECURECOMM)*, pages 1–5, 2006.

[64] T. Higuchi, H. Yamaguchi, and T. Higashino. Clearing a Crowd: Context-Supported Neighbor Positioning for People-Centric Navigation. In *Proceedings of the 10th International Conference on Pervasive Computing (Pervasive)*, pages 325–342, 2012.

[65] D. Tsonev, S. Videv, and H. Haas. Towards a 100 Gb/s Visible Light Wireless Access Network. *Optics Express*, 23(2):1627–1637, 2015.

[66] Q. Wang, D. Giustiniano, and O. Gnawali. Low-Cost, Flexible and Open Platform for Visible Light Communication Networks. In *Proceedings of the 2nd International Workshop on Hot Topics in Wireless (HotWireless)*, pages 31–35, 2015.

[67] H. Wu, Q. Wang, J. Xiong, and M. Zuniga. SmartVLC: When Smart Lighting Meets VLC. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, pages 212–223, 2017.

[68] Q. Wang, M. Zuniga, and D. Giustiniano. Passive Communication with Ambient Light. In *Proceedings of the 12th International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, pages 97–104, 2016.

[69] Q. Wang and M. Zuniga. Passive Sensing and Communication Using Visible Light: Taxonomy, Challenges and Opportunities. *arXiv*, pages 1–6, 2017.

[70] X. Xu, Y. Shen, J. Yang, C. Xu, G. Shen, G. Chen, and Y. Ni. PassiveVLC: Enabling Practical Visible Light Backscatter Communication for Battery-free IoT Applications. In *Proceedings of the 23rd ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 180–192, 2017.

[71] S. Schmid, T. Richner, S. Mangold, and T. R. Gross. EnLighting: An Indoor Visible Light Communication System Based on Networked Light Bulbs. In *Proceedings of the 13th IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9, 2016.

[72] T. Li, Q. Liu, and X. Zhou. Practical Human Sensing in the Light. In *Proceedings of the 14th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 71–84, 2016.

[73] S. Zhu and X. Zhang. Enabling High-Precision Visible Light Localization in Today's Buildings. In *Proceedings of the 15th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 96–108, 2017.

[74] C. An, T. Li, Z. Tian, A. T. Campbell, and X. Zhou. Visible Light Knows Who You Are. In *Proceedings of the 21st ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 39–44, 2015.

[75] Z. Tian, K. Wright, and X. Zhou. The darkLight Rises: Visible Light Communication in the Dark. In *Proceedings of the 22nd ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 2–15, 2016.

[76] M. Ibrahim, V. Nguyen, S. Rupavatharam, M. Jawahar, M. Gruteser, and R. Howard. Visible Light based Activity Sensing using Ceiling Photosensors. In *Proceedings of the 3rd Workshop on Visible Light Communication Systems (VLCS)*, pages 43–48, 2016.

[77] Y. Yang, J. Hao, J. Luo, and S. J. Pan. CeilingSee: Device-Free Occupancy Inference through Lighting Infrastructure Based LED Sensing. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 247–256, 2017.

[78] H. Lee, T. H. Kim, J. W. Choi, and S. Choi. Chirp Signal-Based Aerial Acoustic Communication for Smart Devices. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 2407–2415, 2015.

[79] V. Gerasimov and W. Bender. Things that Talk: Using Sound for Device-to-Device and Device-to-Human Communication. *IBM Systems Journal*, 39(3.4):530–546, 2000.

[80] A. Matic, O. Mayora-Ibarra, A. Maxhuni, and V. Osmani. Virtual Uniforms: Using Sound Frequencies for Grouping Individuals. In *Adjunct Proceedings of the ACM International Conference on Pervasive and Ubiquitous Computing (UbiComp Adjunct)*, pages 159–162, 2013.

[81] D. Gijsbrecht, F. Heller, J. Schöning, and F. Kawsar. Grouve: Spontaneous Proximal Group Formation with Ultrasonic Sound Waves. In *Proceedings of the 8th*

EAI International Conference on Mobile Computing, Applications and Services (MobiCASE), pages 140–141, 2016.

[82] V. Mavroudis, S. Hao, Y. Fratantonio, F. Maggi, C. Kruegel, and G. Vigna. On the Privacy and Security of the Ultrasound Ecosystem. In *Proceedings of the 17th Privacy Enhancing Technologies Symposium (PETS)*, pages 1–18, 2017.

[83] B. Schilit, N. Adams, and R. Want. Context-Aware Computing Applications. In *First Workshop on Mobile Computing Systems and Applications (WMCSA)*, pages 85–90, 1994.

[84] 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 *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing (HUC)*, pages 304–307, 1999.

[85] A. Ghose, C. Bhaumik, and T. Chakravarty. BlueEye: A System for Proximity Detection using Bluetooth on Mobile Phones. In *Adjunct Proceedings of the ACM International Conference on Pervasive and Ubiquitous Computing (UbiComp Adjunct)*, pages 1135–1142, 2013.

[86] A. Varshavsky, A. Scannell, A. LaMarca, and E. de Lara. Amigo: Proximity-Based Authentication of Mobile Devices. In *Proceedings of the 9th International Conference on Ubiquitous Computing (UbiComp)*, pages 253–270, 2007.

[87] P. Sapiezynski, A. Stopczynski, D. Kofoed Wind, J. Leskovec, and S. Lehmann. Inferring Person-to-Person Proximity Using WiFi Signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2):1–20, 2017.

[88] Y. Zheng, M. Li, W. Lou, and Y. T. Hou. Location Based Handshake and Private Proximity Test with Location Tags. *IEEE Transactions on Dependable and Secure Computing*, 14(4):406–419, 2017.

[89] W.-T. Tan, M. Baker, B. Lee, and R. Samadani. The Sound of Silence. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 1–14, 2013.

[90] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell. SoundSense: Scalable Sound Sensing for People-Centric Applications on Mobile Phones. In *Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 165–178, 2009.

[91] R. Jin, L. Shi, K. Zeng, A. Pande, and P. Mohapatra. MagPairing: Pairing Smartphones in Close Proximity Using Magnetometers. *IEEE Transactions on Information Forensics and Security*, 11(6):1306–1320, 2016.

[92] M. Miettinen, N. Asokan, T. D. Nguyen, A.-R. Sadeghi, and M. Sobhani. Context-Based Zero-Interaction Pairing and Key Evolution for Advanced Personal Devices. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 880–891, 2014.

[93] R. Mayrhofer and H. Gellersen. Shake Well Before Use: Intuitive and Secure Pairing of Mobile Devices. *IEEE Transactions on Mobile Computing*, 8(6):792–806, 2009.

[94] K. A. Nguyen, R. N. Akram, K. Markantonakis, Z. Luo, and C. Watkins. Location Tracking Using Smartphone Accelerometer and Magnetometer Traces. In *Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES)*, pages 1–9, 2019.

[95] M. Maier, C. Marouane, M. Klette, F. Dorfmeister, P. Marcus, and C. Linnhoff-Popien. SURFtogether: Towards Context Proximity Detection Using Visual Features. In *Proceedings of the 3rd International Conference on Context-Aware Systems and Applications (ICCASA)*, pages 86–91, 2014.

[96] D. Schürmann, A. Brüsch, N. Nguyen, S. Sigg, and L. Wolf. Moves like Jagger: Exploiting Variations in Instantaneous Gait for Spontaneous Device Pairing. *Pervasive and Mobile Computing*, 47:1–12, 2018.

[97] H. Zhang, W. Du, P. Zhou, M. Li, and P. Mohapatra. DopEnc: Acoustic-based Encounter Profiling Using Smartphones. In *Proceedings of the 22nd ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 294–307, 2016.

[98] C. Peng, G. Shen, and Y. Zhang. BeepBeep: A High-Accuracy Acoustic-Based System for Ranging and Localization using COTS Devices. *ACM Transactions on Embedded Computing Systems*, 11(1):1–29, 2012.

[99] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye. Push the Limit of WiFi based Localization for Smartphones. In *Proceedings of the 18th ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 305–316, 2012.

[100] P. C. Ng, J. She, and S. Park. High Resolution Beacon-Based Proximity Detection for Dense Deployment. *IEEE Transactions on Mobile Computing*, 17(6):1369–1382, 2018.

[101] I. Agadakos, J. Polakis, and G. Portokalidis. Techu: Open and Privacy-Preserving Crowdsourced GPS for the Masses. In *Proceedings of the 15th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 475–487, 2017.

[102] B. Thiel, K. Kloch, and P. Lukowicz. Sound-based Proximity Detection with Mobile Phones. In *Proceedings of the 3rd International Workshop on Sensing Applications on Mobile Phones (PhoneSense)*, pages 1–4, 2012.

[103] A. F. Westin. *Privacy and Freedom*. Atheneum, 1970.

[104] C. Bettini and D. Riboni. Privacy Protection in Pervasive Systems: State of the Art and Technical Challenges. *Pervasive and Mobile Computing*, 17:159–174, 2015.

[105] D. Christin, C. Roßkopf, M. Hollick, L. A. Martucci, and S. S. Kanhere. IncogniSense: An Anonymity-Preserving Reputation Framework for Participatory Sensing Applications. *Pervasive and Mobile Computing*, 9(3):353–371, 2013.

[106] A. Pfitzmann and M. Hansen. Terminology for Talking about Privacy by Data Minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management. URL: `http://dud.inf.tu-dresden.de/Anon_Terminology.shtml` [last checked 16.06.2016].

[107] J. Petit, F. Schaub, M. Feiri, and F. Kargl. Pseudonym Schemes in Vehicular Networks: A Survey. *IEEE Communications Surveys & Tutorials*, 17(1):228–255, 2015.

[108] D. Eckhoff, R. German, C. Sommer, F. Dressler, and T. Gansen. SlotSwap: Strong and Affordable Location Privacy in Intelligent Transportation Systems. *IEEE Communications Magazine*, 49(11):126–133, 2011.

[109] A. R. Beresford and F. Stajano. Mix Zones: User Privacy in Location-Aware Services. In *Proceedings of the Second IEEE Conference on Pervasive Computing and Communications Workshops (PERCOMW)*, pages 127–131, 2004.

[110] Rongxing Lu, Xiaodong Li, T. H. Luan, Xiaohui Liang, and Xuemin Shen. Pseudonym Changing at Social Spots: An Effective Strategy for Location Privacy in VANETs. *IEEE Transactions on Vehicular Technology*, 61(1):86–96, 2012.

[111] Y. Pan and J. Li. Cooperative Pseudonym Change Scheme Based on the Number of Neighbors in VANETs. *Journal of Network and Computer Applications*, 36(6):1599–1609, 2013.

[112] G. Corbellini, K. Aksit, S. Schmid, S. Mangold, and T. Gross. Connecting Networks of Toys and Smartphones with Visible Light Communication. *IEEE Communications Magazine*, 52(7):72–78, 2014.

[113] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta. Luxapose: Indoor Positioning with Mobile Phones and Visible Light. In *Proceedings of the 20th ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 447–458, 2014.

[114] Gummeson, Jeremy, J. McCann, C. Yang, D. Ranasinghe, S. Hudson, and A. Sample. RFID Light Bulb: Enabling Ubiquitous Deployment of Interactive RFID Systems. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 1(2), 2017.

[115] Q. Wang, D. Giustiniano, and M. Zuniga. In Light and In Darkness, In Motion and In Stillness: A Reliable and Adaptive Receiver for the Internet of Lights. *IEEE Journal on Selected Areas in Communications*, 36(1):149–161, 2018.

[116] A. U. Guler, T. Braud, and P. Hui. Spatial Interference Detection for Mobile Visible Light Communication. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–10, 2018.

[117] International Telecommunication Union. International Morse Code: Recommendation ITU-R M.1677-1. URL: `http://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.1677-1-200910-I!!PDF-E.pdf` [last checked 16.06.2020].

[118] A. Narayanan, N. Thiagarajan, and M. Lakhani. Location Privacy via Private Proximity Testing. In *Proceedings of the 18th Network and Distributed System Security Symposium (NDSS)*, pages 1–17, 2011.

[119] L. Cohen. Generalization of the Wiener-Khinchin Theorem. *IEEE Signal Processing Letters*, 5(11):292–294, 1998.

[120] S. Schmid, J. Ziegler, G. Corbellini, T. R. Gross, and S. Mangold. Using Consumer LED Light Bulbs for Low-Cost Visible Light Communication Systems. In *Proceedings of the 1st ACM MobiCom Workshop on Visible Light Communication Systems (VLCS)*, pages 9–14, 2014.

[121] S. Schmid, T. Bourchas, S. Mangold, and T. R. Gross. Linux Light Bulbs: Enabling Internet Protocol Connectivity for Light Bulb Networks. In *Proceedings of the 2nd International Workshop on Visible Light Communications Systems (VLCS)*, pages 3–8, 2015.

[122] B. S. Everitt, S. Landau, M. Leese, and D. Stahl. Measurement of Proximity. In *Cluster Analysis*, Wiley Series in Probability and Statistics, pages 43–69. 2011.

[123] D. Ferreira, V. Kostakos, and A. K. Dey. AWARE: Mobile Context Instrumentation Framework. *Frontiers in ICT*, 2:1–9, 2015. URL: `https://awareframework.com` [last checked 16.06.2020].

[124] M. Haus, A. Y. Ding, and J. Ott. Proximityness: Dataset for Evaluation of Co-Presence Detection, 2020. URL: `https://crawdad.org/tum/proximityness/20200218/` [last checked 16.06.2020].

[125] T. Xu and Y. Zhou. Systems Approaches to Tackling Configuration Errors. *ACM Computing Surveys*, 47(4):1–41, 2015.

[126] M. Wang and J. Brassil. Managing Large Scale, Ultra-Dense Beacon Deployments in Smart Campuses. In *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 606–611, 2015.

[127] EdTech Staff. Bluetooth Beacons Could Improve Student Experience on Higher Ed Campuses, 2017. URL: `http://www.edtechmagazine.com/higher/article/2017/02/bluetooth-beacons-could-improve-student-experience-higher-ed-campuses` [last checked 16.06.2020].

[128] Bluvision. BEEKS Beacon Maker, 2016. URL: `https://play.google.com/store/apps/details?id=com.bluvision.beaconmaker` [last checked 16.06.2020].

[129] E. Bregu, N. Casamassima, D. Cantoni, L. Mottola, and K. Whitehouse. Reactive Control of Autonomous Drones. In *Proceedings of the 14th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 207–219, 2016.

[130] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr. Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). *Neurocomputing*, 307:72–77, 2018.

[131] N. Carey. Establishing Pedestrian Walking Speeds. *Portland State University*, 2005.

[132] A. Di Luzio, A. Mei, and J. Stefa. Mind Your Probes: De-Anonymization of Large Crowds Through Smartphone WiFi Probe Requests. In *Proceedings of the 35th IEEE International Conference on Computer Communications*, pages 1–9, 2016.

[133] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers. The SIMON and SPECK Families of Lightweight Block Ciphers. URL: `https://eprint.iacr.org/2013/404` [last checked 16.06.2020].

[134] H. Sun, K. Sun, Y. Wang, and J. Jing. TrustOTP: Transforming Smartphones into Secure One-Time Password Tokens. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 976–988, 2015.

[135] N. Karapanos, C. Marforio, C. Soriente, and S. Čapkun. Sound-Proof: Usable Two-Factor Authentication Based on Ambient Sound. In *Proceedings of the 24th USENIX Security Symposium*, pages 483–498, 2015.

[136] A. Whitmore, A. Agarwal, and L. Da Xu. The Internet of Things - A Survey of Topics and Trends. *Information Systems Frontiers*, 17(2):261–274, 2015.

Publication 1

M. Haus, M. Waqas, A. Y. Ding, Y. Li, S. Tarkoma, and J. Ott. Security and Privacy in Device-to-Device (D2D) Communication: A Review. *IEEE Communications Surveys & Tutorials*, 19(2):1054–1079, 2017. doi:10.1109/COMST.2017.2649687

This thesis includes the accepted version of our article and not the final published version.

Publication Summary

We have analyzed Device-to-Device (D2D) communication for data exchange among physically close devices. Mobile operators are the driving force of the D2D development to enhance network performance and support proximity services based on short range communications. We identified two system models of D2D communication: a) standalone D2D without infrastructure and b) network-assisted D2D with infrastructure, e.g., access points, base stations. In addition, to reveal unique properties of D2D, we compared D2D communication with machine-to-machine (M2M) communication and mobile ad hoc networks (MANETs). Our goal is to improve the understanding of researchers and developers with respect to problems and potential solutions for D2D security and privacy. Therefore, we defined the requirements for security and privacy in D2D and analyzed their relations whether the requirements are contradicting or supporting, and we provided a detailed D2D attack and threat model. One major part of our review is about security solutions for D2D communication with a classification into five different domains: key management, authentication and authorization, confidentiality and integrity, availability and dependability, and secure routing and transmission. The other main part is about privacy solutions for D2D communication, we analyzed multiple privacy approaches and classified them into the five following areas: access control, obfuscation, anonymity, cryptography, and application-oriented privacy involving communication privacy, device and application privacy, and location privacy. On this basis, we performed a detailed discussion of existing D2D solutions according to security and privacy requirements and identified open problems that deserve future research. Our intention is that the presented insights serve as reference guide for further development of D2D security and privacy solutions considering open problems.

Security and Privacy in Device-to-Device (D2D) Communication: A Review

Michael Haus, Muhammad Waqas, Aaron Yi Ding, *Member, IEEE,* Yong Li, *Senior Member, IEEE,* Sasu Tarkoma, *Senior Member, IEEE,* and Jörg Ott, *Member, IEEE*

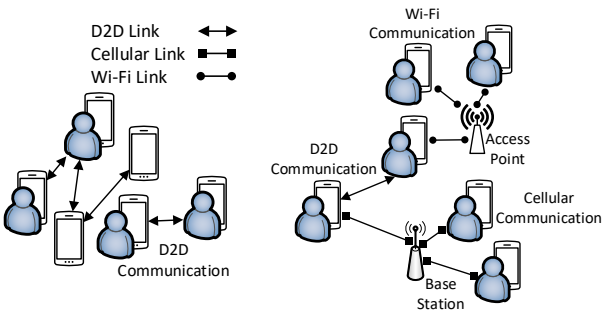
***Abstract*—Device-to-Device (D2D) communication presents a new paradigm in mobile networking to facilitate data exchange between physically proximate devices. The development of D2D is driven by mobile operators to harvest short range communications for improving network performance and supporting proximity-based services. In this article, we investigate two fundamental and interrelated aspects of D2D communication, security and privacy, which are essential for the adoption and deployment of D2D. We present an extensive review of the state-of-the-art solutions for enhancing security and privacy in D2D communication. By summarizing the challenges, requirements, and features of different proposals, we identify lessons to be learned from existing studies and derive a set of "best practices". The primary goal of our work is to equip researchers and developers with a better understanding of the underlying problems and the potential solutions for D2D security and privacy. To inspire follow-up research, we identify open problems and highlight future directions with regard to system and communication design. To the best of our knowledge, this is the first comprehensive review to address the fundamental security and privacy issues in D2D communication.**

***Index Terms*—Device-to-device (D2D) communication, security, privacy.**

I. Introduction

INFORMATION exchange between people has been fundamentally changed by new technologies, such as mobile computing and wireless communication. In spite of rapid advancements, mobile techniques like cellular networks are infrastructure-dependent. The connectivity of mobile users is confined to the coverage of base stations and direct communication between mobile devices is not permitted [1]. The traffic is routed via a core network, even if source and destination are in close proximity to one another. This inflexibility limits the potential of data exchange between mobile users. Especially, when considering the shift in personal computing from stationary PCs and heavier laptops to mobile devices. In 2012, smartphones and tablets outsold PCs and notebooks fivefold and the gap will further increase up to tenfold in 2018 [2], [3]. As a result of this shift to mobile devices, the mobile data traffic is expected to grow to 30.6 exabytes per month by 2020, an eightfold increase over 2015 [4]. Therefore, we need new communication technologies that can scale network capacity and enable data exchange on-demand over the right network connections.

Device-to-Device (D2D) communication represents a promising technique to enable devices to communicate directly without the interaction of access points or base stations [5]. The basic concept of D2D is first proposed in [6] for data exchange between peer nodes. Several studies [5], [7], [8] analyzed the concept of using D2D in cellular networks. However, a conventional cellular system does not allow devices to directly communicate with each other, instead all communications take place through the base stations [8]. The aim of D2D is to leverage the physical proximity of communicating devices to extend the cellular coverage mostly in sparse environments [9]. D2D communication should complement traditional cellular networking services. Thereby, resource sharing of spectrum and energy between cellular and D2D communication is a critical design factor [9], [10].



(a) Standalone D2D (b) Network-assisted D2D

Fig. 1. System models: (a) standalone D2D without infrastructure and (b) network-assisted D2D with infrastructure.

Two major models of D2D communication networks are shown in Fig. 1: standalone D2D in Fig. 1(a) and network-assisted D2D in Fig. 1(b). According to [5], [10], the standalone D2D can be defined as:

> D2D enables devices to communicate directly with each other without traversing fixed network infrastructures such as access points or base stations.

The standalone D2D relies on local hardware capabilities and fixed infrastructure such as access points or base stations is not a prerequisite. Thus, D2D devices must be able to organize communications by themselves. The local connectivity of D2D communication is motivated by two aspects: (1) geographic validity, where the locally relevant content is of little interest to the rest of the world; and (2) temporal validity, which states that the information is only valid for a limited amount of time. In contrast, the network-assisted D2D requires infrastructure, such as base stations or access points, for communication organization and resource utilization, as shown in Fig. 1(b) [11].

At application level for D2D communication, service discovery [15] enables content sharing among devices in prox-

TABLE I
COMPARISON OF SHORT-RANGE WIRELESS TRANSMISSION TECHNIQUES [7], [12], [13], [14]

Wireless technology	NFC	UWB	ZigBee	Bluetooth 4.0	WiFi Direct	LTE Direct
Max. transmission distance	0.2 m	10 m	100 m	100 m	200 m	500 m
Max. data rate	424 kb/s	480 Mb/s	250 kb/s	24 Mb/s	250 Mb/s	13.5 Mb/s
Device discovery	Radio-frequency identification	Manual pairing	ID broadcast or coordinator assistant	Manual pairing	ID broadcast and embed soft access point	Service broadcast
Application	Contactless payment systems	location and tracking systems, auto radar	Home entertainment, environmental monitoring	Object exchange, peripherals connection	Content sharing, group gaming	Content sharing, local advertising

imity and community detection [16] explores nearby communication partners. To illustrate the impact of communication range on D2D applications, we depict the short-range wireless technologies for D2D communication in Table I. As shown in the table, D2D communication can utilize various technologies such as Ultra-wideband (UWB), Near Field Communications (NFC), ZigBee, Bluetooth, WiFi-Direct or LTE Direct [12]. Typical D2D applications and services include cellular data offloading, relaying, gaming, content distribution, and group communication [10], [17], [5]. Some representative D2D prototype systems are FlashLinQ, DataSpotting, and Relay-By-Smartphone, which can provide a discovery range from 100 m up to 1 km [18], [17].

A. Comparing D2D with M2M and MANETs

Other communication paradigms similar to D2D include the Machine-to-Machine Communication (M2M), also known as Machine Type Communication (MTC) [5], and Mobile Ad Hoc Networks (MANETs).

We highlight the differences between D2D, M2M and MANETs to show the distinct properties of D2D communication. According to [5], [19], [10], [20], M2M communication can be defined as:

> Data communication among machines or devices that does not require human mediation nor impose specific restrictions on communication ranges.

M2M communication is based on traditional cellular networks, e.g., 3G and LTE [10]. The communication between machines is routed through core networks via base stations and M2M servers, even if source and destination are proximate to one another. In comparison, D2D communication presumes a distance limit between devices and relies only on local device capabilities without centralized infrastructure support. Moreover, M2M is application-oriented and technology-independent, whereas D2D is technology-dependent and focuses on proximity services, which assumes opportunistic connectivity [5]. The main application of M2M is to automatically collect and deliver measurement information. D2D communication, as a new communication pattern, can be used for M2M communication to improve network performance and reduce transmission delay [10]. Some unique features of M2M include: provision of communication between a massive number of devices, small and infrequent data transmission, reduced need to recharge mobile devices [20].

One distinct difference between D2D and MANETs is the communication spectrum. MANETs work mainly on an unlicensed spectrum making spectrum control difficult and interference a major issue [10]. In contrast, D2D can use both a licensed and an unlicensed spectrum depending on the usage. The control mode is also different. In MANETs each node performs system operations autonomously, whereas in D2D the operations can be performed through the cooperation between D2D nodes or using cellular infrastructure. In addition, the routing patterns vary. D2D uses mainly single hop transmission, instead of multi-hop routing commonly used in MANETs [10].

In the following we highlight the advantages and disadvantages of D2D communication. One major benefit of D2D comes from the stronger anonymity and content privacy because shared information is not stored at a central storage. Moreover, D2D offers better performance by improving spectrum re-usage and system throughput owing to the direct routing of D2D traffic [9], [1]. D2D switches from infrastructure path to direct path for offloading cellular traffic [21], [9], [22], [23]. These properties lead to high data rates, low end-to-end transmission delay and energy saving [1], [10]. D2D also entails some drawbacks. The standalone D2D utilizes only device-managed links in which centralized relay or channel management is not possible [9], whereas with operator controlled links for the network-assisted D2D the base station can partially manage relay and channel selections. The interference management in D2D communication requires thorough research attention [9].

B. Security and Privacy

Our work focuses on security and privacy as two fundamental and interrelated aspects of D2D communication, which are essential for the adoption and deployment of D2D. In the following, we highlight specific challenges that are not addressed by traditional approaches.

The missing of central authority such as access points or base stations is the characteristic disparity between standalone D2D and traditional infrastructure-based communication. As a result, the resource-constrained end user devices must take care of functionalities such as auditing and logging that are

usually managed by a centralized entity. Besides that, D2D communication mainly relies on device discovery to detect communication peers, which is done via broadcasting messages over wireless channels. This allows an attacker to locate and track D2D users, thus violating location privacy. Regarding data privacy, D2D can prevent an adversary from attacking a central communication point for stealing private information. However, D2D users still need to protect sensitive content via private information retrieval, e.g., using homomorphic encryption. Furthermore, as D2D users are typically spontaneous and self-managed, security and privacy enforcement in D2D will be more challenging to realize compared with in traditional centralized environments.

To refine the scope, we concentrate on the standalone D2D because it introduces several unique system-level challenges by operating in a distributed networking environment without central coordination. Our contributions are as follows:

- We provide an extensive review of latest work in D2D domain with respect to security and privacy.
- Compared with previous work on D2D security, we provide a thorough discussion dedicated to D2D privacy.
- We further derive a set of best practices and identify open problems to inspire future work on D2D security and privacy.

The remaining sections of this paper are organized as follows: In Section II we present background and research challenges for security and privacy in D2D communication. We summarize existing approaches in Sections III and IV. In Section V, we discuss the reviewed solutions, highlight "best practices", and identify open problems. Finally, we present the concluding remarks in Section VI.

II. Security and Privacy in D2D

The discussion on security issues for wireless ad-hoc networks started many years ago [24] and there are still open problems. The 3GPP Security Workgroup (SA3) has identified six vulnerability categories for the security and privacy domain [25]:

1) Physical attacks
2) Compromised credentials
3) Configuration attacks
4) Protocol attacks
5) Attacks on core networks
6) User data and privacy attacks

Especially for D2D, connections between proximate devices are vulnerable to security threats due to: (1) direct wireless connection, (2) mobility of end users and (3) privacy issues in social applications [10].

The greater the number of devices that adopt D2D communication, the greater the interest of adversaries to attack these networks (e.g., communication networks becoming the target of cyber-attacks [26], [27], [28], [29], [30]). This stresses the importance of security and privacy in the design of new wireless mobile communication. According to a recent study [31], security and privacy are open issues for D2D.

Given that the existing proposals in the wireless ad hoc domain form a good solution base, although not directly for D2D communication [32], we focus on recent work that directly addresses the security and privacy challenges for D2D.

A. Security and Privacy Requirements for D2D

1) Security: The information exchange between D2D users is more vulnerable due to the exposed nature of wireless communication. Secure wireless communication must satisfy the requirements of authenticity, privacy, confidentiality, integrity, and availability [33] to provide protection against different attacks, such as Denial of Service, masquerading, eavesdropping [34], [35]. We highlight the following security requirements for D2D communication:

a) *Authentication and Authorization*: The goal of authentication is to evaluate who you are. It verifies the possession of a private key or a secret. The prerequisite is to assign an identity to a key or secret. This requires key revocation, in case of a lost or stolen private key where the key is no longer associated with the user identity. In contrast, authorization verifies and grants what you are permitted to do. First the D2D system authenticates the user and then grants the user with pre-defined allowed actions. On this basis, we can uniquely identify each D2D user to distinguish between authorized D2D users and non-authorized users. Authentication and authorization are important to protect D2D communication against impersonation and masquerading attacks.

b) *Availability and Dependability*: Authorized D2D users should be capable of accessing a wireless network anytime and anywhere, even under DoS or DDos attacks. DoS attacks are more difficult to detect in D2D networks because D2D does not rely on centralized infrastructure [26]. For example, a jamming attack can be anonymously started and adversely affect communication between D2D users.

c) *Non-Repudiation*: Non-repudiation guarantees that authentication can be asserted to be genuine and not be refuted later. For instance, a system that prevents an attacker who was authenticated before to deny authorship of messages later. Besides that, non-repudiation is mostly a legal concept rather than a cryptographic one [36]. Usually the legal concept refers to non-repudiation of origin, of transfer, and of delivery. Correlated with non-repudiation, one major problem in cooperative D2D environments is trust, which escalates the risk of collusion attacks if one D2D device trusts another device to attest some aspect of non-repudiation.

d) *Secure Routing and Transmission*: In the presence of adversaries, the information must be securely exchanged among D2D users. We have to ensure that only intended D2D users are able to read the messages. Moreover, any modification of a message during the transmission from sender to receiver must be prevented.

e) *Confidentiality*: D2D service controls the data access to ensure that only authorized D2D users can access it [37]. For instance, symmetric key encryption (SKE) uses a shared key between D2D nodes to encrypt the data before transmission.

f) *Integrity*: The goal of integrity is to provide accurate and reliable information among D2D users without modification or falsification. Data integrity may be violated if the attacker compromises a node and launches malicious attacks, such as message injection or false reporting [38].

The protection mechanism for standalone D2D must consider that the direct connections between proximate devices are more vulnerable due to limited computational capacity of mobile devices for security related computations [39].

2) Privacy: In contrast to security, which has a clear and widely accepted definition, there exists no commonly used definition for privacy. In addition, the term privacy covers a large field of concepts with different interpretations [40], [41], [42]. That is a surprising fact especially given that privacy is one of the most important concepts of our time and yet remains one of the most elusive notions [43]. The following definitions show the evolving understanding of privacy from a social-oriented explanation to a more technique-conscious definition.

One of the oldest and most cited privacy definition is from the 19th century by Warren and Brandeis: the "right to be let alone" [44]. Another traditional definition of privacy is "the state of being alone and not watched or disturbed by other people" [45]. Altman realized that privacy is a "boundary regulation process whereby people optimize their accessibility along a spectrum of 'openness' and 'closedness' depending on context" [46]. Thus, the user has to share data to some extent otherwise no useful, or only limited, services are possible. Westin supports that statement by specifying privacy as a "personal adjustment process" [47] to find a balance between "desire for privacy with the desire for disclosure and communication". Most of today's privacy understanding is based on Westin's explanation from 1967 [47, p. 7]:

> Privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others.

Altman and Westin were referring to nonelectronic environments with limited privacy violation. Today, personal information is accessible anytime and anywhere by billions of people. Hence, the D2D system must consider the following privacy considerations for managing sensitive user data. *Transparency and minimization*, D2D users must be aware of which data they are sharing and the D2D system collects only absolutely required data to provide a specific D2D service. A good idea is to make the user data gathered by the D2D system available to the D2D user [48]. *Sensitivity of personal data* is highly subjective and context-dependent. Therefore, the tools to specify user preferences must be flexible to allow different degrees of data publication. Which user data is transmitted and to what extent to the D2D service. *Access control*, individual user has selective control over their personal data [46], [49]. *Risk management and data protection*, minimize future privacy risks by protecting data that is no longer under direct control of the user [50]. The D2D communication must be protected by some form of encryption. Our privacy requirements for D2D are as follows [41], [51]:

TABLE II
LEGEND FOR SECURITY AND PRIVACY REQUIREMENTS

Security Requirements		Privacy Requirements	
AA	Authentication and Authorization	AI	Anonymity and Indistinguishability
AD	Availability and Dependability	U	Unlinkability
NR	Non-Repudiation	CP	Context Privacy
SRT	Secure Routing Transmission	D	Deniability
CI	Confidentiality and Integrity		

a) *Anonymity and Indistinguishability*: hide the identity of origin and destination of a D2D conversation from an adversary.
b) *Unlinkability*: different sessions of D2D communication of the same user should not be linkable. An adversary cannot link D2D communication activities of a particular D2D user to create a user's profile, which contains a great deal of personal information.
c) *Context Privacy*: adversary is not able to learn context information during the D2D access, e.g., user location, talk time, type of service request.
d) *Confidentiality and Integrity*: interactions between D2D user and service include confidentiality and integrity protection.
   - *Confidentiality*: attacker cannot read messages transmitted between two D2D users. This can be achieved by cryptographic mechanisms, like stream ciphers to prevent eavesdropping.
   - *Integrity*: message during transmission cannot be modified. Modifications include changing, deleting, creating, delaying or replaying messages. Integrity can be ensured by other cryptographic mechanisms like hash functions.
e) *Deniability*: being able to plausibly deny a certain action, such as sending a message.

The legend for security and privacy requirements used in the following discussions is presented in Table II.

B. Relations between Security and Privacy Requirements

The previous Section, "Security and Privacy Requirements for D2D", defined the necessary D2D system requirements and in this section we discuss the relationship between the requirements as depicted in Fig. 2.

One challenge for realizing security and privacy in D2D communication is related to conflicting requirements. True anonymity hides the user's identity from eavesdroppers, service providers and even other communication partners. However, we are unable to detect illegal user behavior if the user can launch attacks anonymously. Anonymity conflicts with authentication, the process by which the user identity must be revealed for verification. User identity can be used as an unique identifier by the attacker to track users and leak sensitive information. This potential traceability contradicts

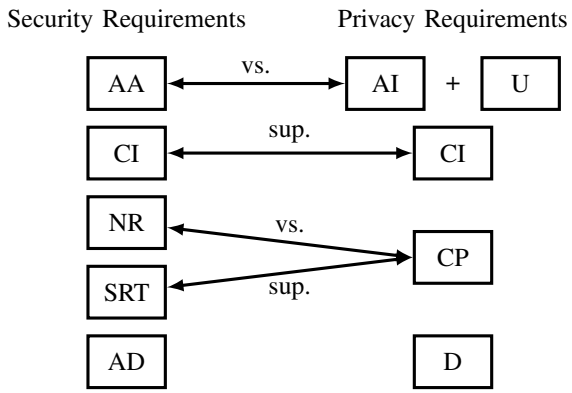


Fig. 2. Relationships between security and privacy requirements for D2D. Contradicting requirements are identified with "vs.". Supporting requirements are identified with "sup.". The requirements AD and D have no relation to other security or privacy requirements.

another privacy requirement, the unlinkability of an user. The basic idea to solve these contradicting requirements is to use anonymous authentication.

In non-repudiation, the message originator is verifiable to avoid data leakage by false notifications from the adversary. On the other hand, context privacy protects the data involved during the D2D communication, such as user location, conversation partners, and talk time. It is easier for the attacker to find the associated conversation data, when using a verifiable message originator. The security requirement of secure routing and transmission supports context privacy and should also protect against passive traffic analysis. Otherwise, only the content is secured against attackers but the adversary is still able to find the communicating parties by analyzing the amount and frequency of exchanged messages.

The following two security and privacy requirements share the same goals. In the security domain, confidentiality guarantees that only intended D2D users are able to access the data. Similarly, for privacy it is important that the attacker cannot read messages transmitted between two parties, which must be considered for secure transmissions. The integrity requirement defines the same goal between the two domains security and privacy. The information among targeted D2D users is not modifiable by unauthorized users.

In contrast, the following requirements have no direct relation to other requirements: the security requirement, Availability and Dependability to ensure user access at anytime and anywhere even during attacks, and the privacy requirement Deniability.

C. Attack and Threat Model

We need a clear adversary model for D2D to properly evaluate security and privacy protection mechanisms. The adversary model specifies at least: (1) the parts of the personal information being transferred and/or processed to which the adversary has access, (2) external or background knowledge to which the attacker has access, and (3) can different adversaries collude [48].

TABLE III
D2D THREAT MODEL FOCUS ON THREE DIMENSIONS: ACTIVE OR PASSIVE / INSIDER OR OUTSIDER / LOCAL OR EXTENDED ATTACK TOGETHER WITH TARGET ENTITY

	Insider & Local	Outsider & Extended
	Target: Mobile Device	Target: Wireless Connection
Active	malware & ransomware, app rewriting, hijacking, information leakage, social engineering, masquerading	jamming, denial of service, session hijacking, impersonation, replay, delay, drop, repudiation, data corruption
Passive	location tracking, context monitoring	eavesdropping, man-in-the-middle, traffic analysis

For our attack and threat model we analyzed two central entities: the mobile device and the wireless connection for communication with other nearby mobile devices. D2D inherently provides a strong anonymity because it misses the central authority like a base station. Usually, the central authority has access to a broader range of data, which increases the risk of potential attacks and threats. Our threat model is based on three dimensions [52]:

1) *Insider vs. Outsider*: The inside attacker is an authenticated user in the network and can communicate with other members. The outside attacker is a non-authentic intruder with less privileges than the insider, which leads to less threats.
2) *Active vs. Passive*: An active attacker can directly modify the network or mobile device to obtain sensitive information. For instance, modifications include change, delete, create, delay or replay of messages. On the other hand, the passive attacker acts in the background and does not affect the mobile device or network. The adversary listens, collects, and analyzes data. Once the passive attacker has access to the system, it is hard to detect this adversary.
3) *Local vs. Extended*: The local attack is limited in scope and adversely influence only a few systems. An extended attacker can control multiple entities scattered across the network.

Our threat model with corresponding attacks is shown in Table III. In this table, the attack pattern is described as active or passive and the attack scope involves either a mobile device and/or a wireless connection. The certain attack can be further influenced by internal or external background knowledge of the attacker and by the number of compromised entities. For instance, the classification of location privacy attacks results in four different types of attacks: single or multiple position attack, context linking attack, and compromising a trusted third party (TTP) [54]. Table IV shows the potential attacks to D2D security and privacy as identified in our threat model.

III. Security Solutions for D2D

D2D communication is vulnerable to diverse attacks due to the broadcast nature of wireless communication [53]. For example, an attacker can easily gain critical or private information by secretly listening to the unprotected communication among devices. We categorize the selected security solutions

TABLE IV
POTENTIAL ATTACKS IN D2D COMMUNICATION [30], [52], [53]

No	Attack	Description
1	(Distributed) denial of service	Attacker floods the wireless channel with generated messages to disrupt communication. D2D is more vulnerable to DoS attacks because of real-time constraints for the D2D communication. To overcome this problem, we can switch to another wireless channel.
2	Man-in-the-middle attack	Adversary is positioned between sender and receiver and sniffs any information being sent between the two nodes.
3	Masquerading	Attacker tries to pretend it is another authenticated communication partner by using a false identity. The behavior is similar to the impersonation attack.
4	Impersonation	Launch an attack using the identity of other mobile devices, e.g., MAC or IP address. This is often the first step for additional, more sophisticated attacks.
5	Session hijacking	Attacker spoofs the victim's IP address and determines the sequence number expected by the target node. Afterwards, the adversary performs a DoS attack on the victim node and impersonates this node to continue the session with the target node.
6	IP spoofing	Malicious node manipulates IP packets, particularly the headers.
7	Bandwidth spoofing	Adversary has unauthorized access to the bandwidth of a legitimate user.
8	Eavesdropping	Mobile hosts share the same wireless medium and broadcast signals over airwaves, which can be easily intercepted by receivers tuned to the proper frequency. Thus, the attacker can read exchanged messages and is able to inject fake messages to manipulate other users.
9	Jamming	Transmitter generates a strong signal to disrupt communications. As a result, the transmitted messages are corrupted or lost.
10	Location spoofing	Attacker sends fake location information to disturb the D2D group formation. In addition, the adversary is able to imitiate artifical locations to confuse D2D group members.
11	Inference attack (context data leakage)	Attacker eavesdrops a wireless channel for various purposes, such as location tracking and context monitoring. These techniques aim at infer user behavior and whereabouts. For example, the threats associated with location tracking are stalking, mugging, burglary of unoccupied home. The adversary tries to recognize user activities by movement traces, such as frequent visits to a hospital or a night club, to obtain sensible data.
12	Malware attack (mobile data leakage)	The users' mobile device is compromised by malware and/or ransomware. The malicious program can be a trojan, worm, virus or botnet/spyware and is able to attack both operating systems and user applications. Thereby, the attacker reveals private information. The malicious program can spread through the network and slow down the entire mobile system or cause damage.
13	Free-riding attack	Selfish D2D users are not willing to share their own resources with other D2D users resulting in reduced system utilization and availability for D2D communication.
14	Trust manipulation attack	Adversary forges its trust value so that other D2D users believe that he will act in a reliable and trustworthy way. For example, to attract D2D communication requests.

into five domains: (1) key management, (2) authentication, (3) confidentiality and integrity, (4) availability and dependability, and (5) secure routing and transmission, as highlighted in Fig. 3.

Key management and authentication services guarantee that data originates from authentic entities. Key management is a crucial issue to achieve several security requirements especially for distributed systems like D2D communication. Key management generates, stores and exchanges cryptographic keys among legitimate users. Authentication provides mutual authentication and secure group communication. Confidentiality and integrity prevent leakage of exchanged data to illegitimate users. Another domain of security is availability and dependability to maintain satisfactory user experience. For instance, any node is able to launch a Denial of Service (DoS) attack to disturb D2D communication. Therefore, availability and dependability ensures that D2D communication is available even under DoS or DDos attacks. Finally, secure routing and transmission protects data transmission among authentic users.

A. Key Management

Key management is a basic procedure for security to generate, store, exchange and update keys [55]. In group communication, key management is crucial when members join or leave the group using shared keys.

Yeh *et al.* [56] proposed key agreement and batch authentication for peer-to-peer (P2P) based online social networks (OSNs). Their security framework offers embedded key authentication and requires less messages to authenticate several users. It applies three different batch authentication protocols: one-way hash function for lower computational cost, ElGamal proxy encryption to exchange information among users, and a certificate based protocol guarantees non-repudiation of transactions. The work of [57] also used batch authentication to offer an efficient one-to-many authentication approach for P2P based networks.

In the following, we discuss Attribute Based Encryption (ABE) for secure data exchange in delay tolerant networks (DTNs). Sudarsono and Nakanishi [58] implemented ABE for authenticating routing messages. The routing node encrypts the symmetric key using ABE and then distributes it to all

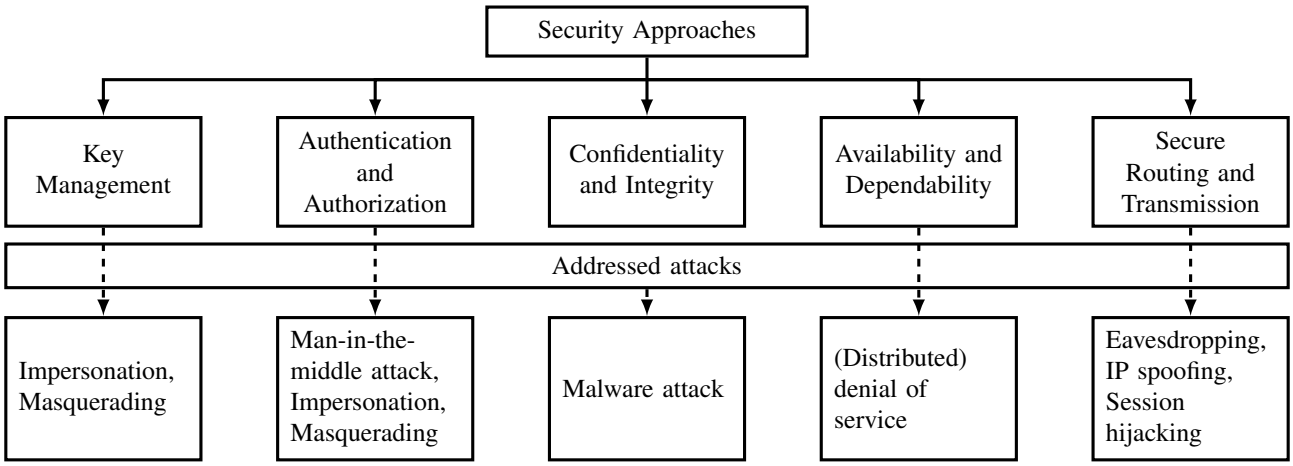


Fig. 3. Classification of security approaches in D2D communication and addressed attacks.

participating nodes. Only those nodes that match a specific attribute policy are able to extract the key. The routing message itself is encrypted via Advanced Encryption Standard (AES). Hur and Kang [59] proposed an attribute-based secure data retrieval scheme using CP-ABE for decentralized DTNs. This solution allows immediate attribute revocation, which enhances backward and forward secrecy of confidential data. Moreover, their approach resolved the key escrow problem.

Jaiswal *et al.* [60] proposed a group key agreement (AGKA) protocol based on Elliptic Curve Cryptography (ECC). The users securely communicate via a session key, which is received from a trusted third party. In [61], the authors discussed issues of group dynamics and key management for secure group communication. A secure group communication computes and distributes group keys with minimal communication and computation cost.

In M2M networks, most approaches use Group Key Agreement (GKA) and Group Key Management (GKM). Each M2M device shares a group key with other devices in the same group. Similar approaches are presented in [62] and [63] for group based authentication in M2M networks. Zhang *et al.* [62] used group based authentication and GKA. In this work, each M2M device pre-shares an additional secret key with other M2M devices of the same group. This shared key is used for local authentication with the serving network. The authors in [63] proposed a lightweight group authentication protocol for M2M communication based on message authentication codes. The so-called LGTH framework authenticates all M2M devices and reduces the authentication overhead.

The authors of the paper [64] analyzed a dynamic updating policy for GKA in M2M LTE-A networks. Their approach uses an asynchronous secret share along with Diffie-Hellman key exchange for authentication in LTE-A networks. The authority of M2M devices is dynamically updated in their approach.

Cao *et al.* [65] aim to increase the security of M2M devices. Their approach used a group based access authentication by aggregation signature. The network simultaneously trusts a group of devices and generates independent session keys with each device using the group based keying.

Another important aspect of D2D communication is to securely find localized content in the network. Searchable encryption (SE) creates an encrypted search index generated over a data collection to protect the content without appropriate tokens. The authors of [66] and [67] analyzed SE and suggested a multi-keyword ranked search operation over encrypted data.

B. Authentication and Authorization

Authentication is a key factor for secure D2D communications to resist a multitude of attacks. It must be ensured that only authorized devices can use the D2D service. There are two types of authentication: entity authentication and data authentication.

[68] aims to design a joint operation protocol comprising routing control and group key agreement. The work is based on ideas related to the dual operation of infrastructure and ad hoc D2D mode. The approach proposed by these authors controls the D2D network and manages the group key in self-organized groups of ad hoc nodes based on their IP addresses. The authors of [69] and [70] considered key agreement and key management to provide authentication in D2D communications. Shen *et al.* [69] introduced a secure and efficient key agreement protocol for transmission in D2D communications. The authentication is based on Diffie-Hellman key agreement and commitment schemes. The secure key agreement enables two mobile devices to establish a shared secret key for D2D communication without prior knowledge. This technique is robust against man-in-the-middle attacks.

On the other hand, the authors in [70] presented key exchange protocols for end-to-end security. The D2D users can hide their identity and group information during the communication. Public Key Cryptography (PKC), based on digital signature, and mutual authentication provide user authentication, non-repudiation, traceability, and integrity. Symmetric encryption further ensures data confidentiality.

The proposal in [71] introduced an end-to-end authentication which is implemented using ECC based Identity-based Cryptography (IBC). This facilitates system implementation on constrained IoT devices. The architecture consists of a

trusted authority (TA) on the border gateway. Each owner of IoT subnet can assign subnet ID and maintain a TA on the border gateway. The border gateway manages authentication and trust of TA keys to avoid additional communication load and latency. The revocation of a public key in IBC also revokes the identity. To overcome this problem of public key revocation in IBC, the identities in their approach are locally assigned IPv6 addresses. These addresses can be renewed whenever trust to a local device requires revocation.

Zhang *et al.* [72] proposed a Secure Data Sharing (SeDS) protocol for D2D communication in LTE-A networks. SeDS is based on Diffie-Hellman Key Exchange (DHKE) and HMAC digital signature to provide authentication and malicious node detection. If the transmitted data originates from an illegitimate provider or is altered by adversaries, the receiver is able to detect the event by signature verification and send a feedback message. Security management schemes are necessary to enable authentication of user content. Goratti *et al.* [73] suggested a security communication protocol to establish direct links between D2D devices. The protocol broadcasts a beacon to nearby devices to set up a D2D communication and then uses a random pre-distribution encryption key for authentication.

Key generation via physical layer is especially interesting for D2D communications. The secret key generation (SKG) takes advantage of the randomness and reciprocity of wireless communication channels to ensure secure communications. However, there are different passive and active attacks on physical layer security. The passive attacks include channel probing and randomness abstraction. The active attacks include disruptive jamming and channel manipulation. Therefore, the authors in [74] analyzed the security strength of physical layer key generation based on channel reciprocity and randomness. Their approach combines user generated randomness and channel randomness to create a shared secret key under active attacks. This secret key generation via the physical layer is used to establish direct communication links between transmitter and receiver.

Another scenario considers cooperative relaying for a better randomness in channel variation and a higher key generation rate. Thai *et al.* [75] presented a secret key generation scheme with multiple untrusted relays. The key generation scheme is designed with zero forcing and minimum mean square error (MMSE) channel estimator for untrusted relays. Chen *et al.* [76] used another relay mechanism to create a full duplex jamming scheme for secret key generation.

C. Confidentiality and Integrity

Confidentiality and integrity are important for D2D communication to secure the user contents and enable legitimate users to decrypt content.

We can use a key extraction protocol based on Channel State Information (CSI) to avoid leakage of key information. Usually, such approaches extract keys from the measurement of individual sub carriers. The problem is that CSI measurements from neighboring users have strong correlations. Hence, the attackers can calculate the key in a relatively short time window. Xi *et al.* [77] proposed a fast secret key extraction protocol called KEEP to overcome these problems. KEEP uses a validation mechanism to obtain secret keys from CSI measurements of all users.

Information theoretic security is able to generate secret keys to achieve data confidentiality, integrity and authentication. Chen *et al.* [78] showed a power allocation technique for the generation of secret keys in relay based LTE-A networks. The impact of power allocation on the SKG rate improved network security.

Sun *et al.* [79] introduced cooperative key generation to set up shared secret keys between devices. Cooperative key generation enables two users to select neighbors as relays and directly extract a secret key from the wireless channels among them. The main issue is the self-interest of mobile users to act as relays without sufficient reward. For this purpose, the authors illustrated a game theoretical approach called SYNERGY to encourage cooperative key generation. In SYNERGY, the cooperative key generation is formulated as a coalition game. The algorithm partitions all involved nodes into multiple disjoint coalitions. Every node in a coalition is strongly encouraged to support other nodes in the same coalition to establish secret keys for rewards.

Tata and Kadoch [80] presented a secure load balancing algorithm called Selective Ad hoc on Demand Multipath Distance Vector (LBS-AOMDV). The objective is to reduce the impact of confidentiality attacks by preventing eavesdroppers from obtaining information from legal users. LBS-AOMDV is based on multipath coded information transmissions, data splitting, and data shuffling schemes. The packets are divided into segments. Afterwards, each segment is shuffled with respect to the random sequence position (RSP). Thus, the number of intercepted packets decreases and the eavesdropper receives less meaningful information. LBS-AOMDV assumes that only source and destination know the RSP, which is encrypted at the transmission begin.

In order to establish social relationships between D2D users, Guo *et al.* [81] proposed a privacy preserving mutual authentication scheme. This scheme first identifies social relationship based on similar user attributes. Then, the D2D users are able to share their encrypted content and only users with similar attributes can decrypt the content. Another work [82] keeps data confidential, detects misbehavior of service providers, and is broadly applicable to popular social networks, such as Facebook. The clients collaborate to ensure data confidentiality and integrity when using an untrusted service provider. The untrusted service provider cannot deviate from the correct execution without being detected. Therefore, the data shared among users is signed by the data provider to ensure data authority. The signed data will be re-signed by the transmitter to guarantee the transmission and provide evidence for the data sharing event.

D. Availability and Dependability

Availability guarantees that the authorized user is able to access the D2D communication. Denial of service is referred to as non-availability of a service that should be available.

Liu *et al.* [83] considered secure transmission in large-scale cellular networks with energy-constrained D2D transmitters. The authors introduced Wireless Power Transfer Policy (WPTP) and an information signal model to enable wireless energy harvesting and secure information transmission. The information signal model uses stochastic geometry to model, analyze, and evaluate the performance of the network. The system's security performance is determined by power outage probability and secrecy throughput. The results show that the secrecy performance is improved by increasing the densities of multi-antenna equipped power beacons and D2D receivers. As an extension, Liu *et al.* [84] demonstrated the power technique for secure D2D communication in large-scale cellular networks. The power transfer model includes three wireless power transfer policies: Cooperative Power Beacons Power Transfer (CPB-PT), Best Power Beacon Power Transfer (BPB-PT) and Nearest Power Beacon Power Transfer (NPB-PT). The authors used the power outage probability to characterize the power transfer reliability of the proposed three policies. For the information signal model, the authors created a comparative framework with two receiver selection schemes: Best Receiver Selection (BRS) and Nearest Receiver Selection (NRS). The objective of BRS and NRS is to examine various network parameters, such as density of D2D receivers, threshold transmit power. As a result, BRS achieves better secrecy performance than NRS, but incurs additional overhead.

Chuan *et al.* [85] studied a large scale D2D enabled cellular network in the presence of eavesdroppers via stochastic geometry. They studied SINR distribution of cellular links, D2D links, and eavesdropping links. The results show that cellular links are not reduced by introducing D2D links. Furthermore, the interference from D2D communications can be exploited to enhance physical layer security of cellular communications. The main limitation of their study is the fixed communication mode, either cellular or D2D, for each user. The users should be able to change the communication mode.

The authors of [86] presented a solution based on Identity Based Encryption (IBE) to secure the exchanged D2D messages during discovery and communication. A pseudonym-based scheme is applied to ensure user privacy and update private keys. In addition, the Elliptic Curve Digital Signature Algorithm (ECDSA) provides non-repudiation.

Zhang *et al.* [87] examined physical layer security in D2D communication as an underlay to cellular networks. They state that D2D generates interference when it accesses the spectrum of cellular users and hence decreases the channel's secrecy capacity. In contrast, D2D increases the secrecy capacity of the system. To address this problem, the authors formulated the radio resource allocation as a weighted bipartite graph and introduced the Kuhn Munkers Algorithm (KMA) to find the maximum sum secrecy capacity for both cellular and D2D users. The results show that the system's secrecy capacity linearly increases with increasing number of cellular and D2D users.

E. Secure Routing and Transmission

The information exchange between D2D users must be secured. Luo *et al.* [88] developed a Stackelberg game in which cellular users are considered as leaders and D2D users are considered as followers. This approach maximizes the rate of cellular users and secrecy capacity of D2D links by optimizing the transmission power and channel access of D2D links. Another work [89] studied the physical layer security in multi tier heterogeneous cellular networks (HCNs). The framework provides secure transmission under stochastic geometry. The authors used an average received signal power (ARSP) policy in which the users can only create a connection with the base station providing highest ARSP value. The link quality is improved by adjusting a larger access threshold of SINR.

Chu *et al.* [90] studied the secrecy rate optimization problem with multiple D2D communications. The work considers two optimization problems: robust power minimization and robust secrecy rate maximization. Their approach used an approximation solution based on Bernstein-type inequality and S-procedure to solve these optimization problems. The Bernstein-type inequality-based approach performs better than the S-procedure regarding achieved secrecy rates.

Another paper [91] applied an interference avoidance scheme for cooperative D2D communication in cellular systems. The cooperative D2D users communicate bi-directionally with each other and also serve simultaneously as relays to assist the two-way transmission between two cellular users. However, the cellular and D2D links share the same spectrum, which creates mutual interference. To overcome this problem, the authors use two different approaches. The first approach is a CSI-free criterion, which aims at system SEP optimization and low complexity. The second approach is a CSI-based criterion for security and reliability with high complexity. Panaousis *et al.* [92] used a Secure Message Delivery (SMD) protocol to securely transmit data from source to destination. Their approach finds a solution for the secure message delivery game. The defenders are D2D users that identify all legitimate network devices. The attackers introduce different malicious messages into the D2D network.

In the following, we discuss secure transmission protocols for ad hoc networks. The authors of [93] analyzed a secure policy agreement for open-privacy routing in wireless communications. Their contributions are as follows: (1) how to obtain an open-privacy policy using Secure Policy Agreement (SPA) mechanisms in on-demand location centric MANET routing, and (2) how to combine SPA with Privacy Routing (SPA-P) protocol for better privacy. The solution achieves a high throughput, low delay and low network overhead. In [94], the authors proposed Inspired Biotic Hybrid Cryptography (IBHC) to protect ad hoc wireless networks against heterogeneous attacks. The SRPAHA protocol enables cryptographically secure communication among nodes using Hybrid DNA-based Cryptography (HDC). HDC requires less communication bandwidth and memory as compared to existing ARAN schemes. The authors of [95] use puncturable encryption to achieve forward secure encryption in store and forward messaging systems, such as email and SMS.

Regarding secure routing protocols that are based on trust management, Chen *et al.* [96] applied dynamic trust management for secure routing optimization. The approach introduced

two social trust metrics: healthiness and unselfishness to deal with malicious and misbehaving nodes. Their results showed that the trust based secure routing protocol outperforms Bayesian trust-based routing and PROPHET. Moreover, trust-based epidemic routing (TBER) is proposed in [97] to address the selfish problem. TBER does not only affect selfish nodes to collaborate with others, it also detects and rejects malicious nodes to send messages. Another idea to reveal misbehaving nodes is to take advantage of an Information Centric Network (ICN) [98]. The ICN monitors and stores all information exchanged in DTNs. Simultaneously, the ICN searches for malicious nodes and selects an alternative transmission path, so that packets arrive at the destination securely. Furthermore, the approach proposed in [99] applied a co-operative scheme called combined faith value (CFV) to reduce the harmful effects of malicious nodes in the network. The node performance in the past is examined by querying neighbor nodes. The node is treated as friendly until it satisfies a pre-defined threshold defined by CFV. A recent work [100] used Fawkes Routers to verify node interactions.

IV. Privacy Solutions for D2D

Proximity-aware applications based on D2D and mobile social networks are facing various privacy challenges, such as location privacy, identity privacy, trust and malicious attacks [101]. For example, 46 % of teen users and 35 % of adults turn off location tracking features due to privacy concerns [102]. Thus, privacy is a key concern in D2D communication to prevent the leakage and illegal usage of sensitive data. We categorize the selected privacy solutions into four domains: access control, obfuscation, anonymity, and cryptography (Fig. 4). The Section "Application-Oriented Privacy" further highlights D2D application scenarios for the reviewed privacy solutions. These scenarios include communication privacy, location privacy, and device-specific privacy.

Access Control ensures a fair use of personal information by using rules or trust-based mechanisms between individuals [40]. For instance, sharing sensible information over D2D with a family member is allowed, but will be denied with a stranger. Anonymity approaches take advantage of pseudonyms to create ambiguity among mobile users. Therefore, we achieve the dissociation of information about an individual to hide the person's identity. The key limitation of anonymity is the need to authenticate the user. In contrast, obfuscation techniques degrade the quality of information, such as the person's location to protect user identity. Obfuscation and Anonymity are similar in that both strategies attempt to hide data in order to protect privacy, but obfuscation is explicitly a spatial approach to location privacy [40]. Finally, the cryptographic approaches have been extensively used to secure wireless communication and to enforce confidentiality of services.

A. Access Control

The idea of access control is to grant or deny a given service provider or other users the right to perform a given action on user's private information. The user should decide whether to share this information or not during the D2D communication. Therefore, the mobile user needs additional mechanisms to control information flow. We can identify three different context-aware access control techniques [48]. In the first technique, the authorization with Discretionary Access Control (DAC) depends on the identity of the subject and is well suited in unstructured domains like generic Internet services. In the second technique, Role-Based Access Control (RBAC) takes advantage of the subject role within a structured organization, such as a company or hospital. The functional role simplifies the definition of access control policies. And in the third technique, the Mandatory Access Control (MAC) uses a sensitivity level assigned to each object and a policy defines which sensitivity level is allowed to access the private information. Most systems for access control use semantic web technologies, such as OWL ontologies, RDF or SWRL to model privacy policies, user context or roles.

In the following, we show examples of access control systems. Behrooz and Devlic [103] proposed a DAC system to control the granularity of the released information. This technique is based on the definition of complex situations via ontology-based context models and support of social relationships. Another access control system called SensorSafe [104] aims at protecting personal sensor data. The level of data disclosure is determined by a broker based on trust among users. The raw sensor data is abstracted to context labels, such as "noise" or "conversation".

The rigidity of merely two possible actions, grant or deny (all or nothing), is a major weakness of existing access control systems. In reality, users need more flexibility by using obfuscation to disclose information at different levels of granularity. There is a demand to define varying levels of data granularity. Therefore, the notion of trust [105] can be helpful to build privacy levels. In principal, we can classify mechanisms for trust establishment into two different categories: credential-based and reputation-based trust [42]. Credential-based trust obtains and verifies credentials of an entity. Usually the credentials are digital certificates, which are maintained by a public-key management (PKI) to ensure bindings of public keys to identities. Methods of reputation-based trust compute trust levels using the history of the entity's past behavior or recommendations by other users.

Personal Data Store (PDS) is another idea to store, manage, and deploy all important personal information in a highly secure and structured way. The individual users get a central point of control for their personal data, such as contact information, preferences, and friend lists. Several approaches build a PDS for better data control and security. The work of [106] proposed a framework known as openPDS, which can collect, store, and manage third party access to personal metadata. However, the framework requires user effort to manage the storage and data access to third parties, and the design does not support user feedback. Haddadi *et al.* [107] proposed a similar framework called Databox, which is a networked device that collects all personal data and provides data control and anonymization of sensitive information.

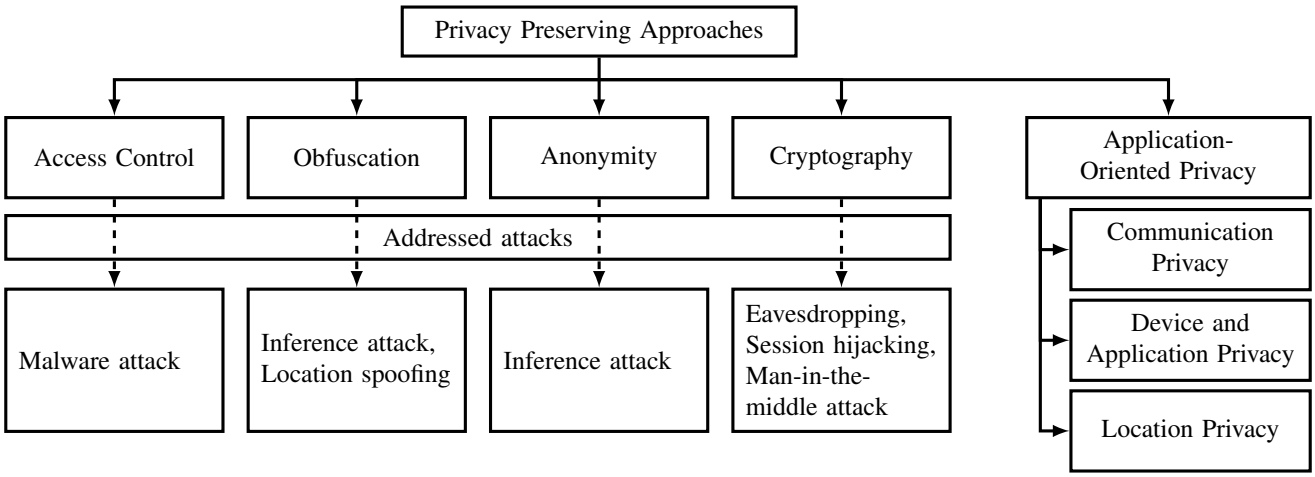


Fig. 4. Classification of privacy preserving approaches [48] and addressed attacks.

B. Obfuscation

The D2D environment is highly dynamic and the status of surrounding users can change frequently. There are multiple possible communication parties and we share data with different levels of sensitivity depending on context factors, such as trust relationships. For instance, consulting a stranger for a train timetable is much less sensitive because the information is available to the public. We share various context data during the D2D communication, e.g., our location, access time, and depending on the D2D implementation much more sensitive data. If there are several potential discussion partners, we need a grouping mechanism based on relative distance and additional context data to describe the identity of the place, including background noise, illumination, humidity, and so on.

Regarding obfuscation, private data is associated with a sensitivity level, which depends on the information accuracy. The less accurate the information or data, the less sensitive is the data. The goal of obfuscation is to degrade the quality of information and protect the user identity. Usually, obfuscation methods are based on generalizing the information or by providing fake information to achieve the aforementioned goal. There has been extensive research on location obfuscation (see details in Section "Location Privacy").

In this section, we present system approaches that can automatically adjust the context data to the current situation according to user preferences, discussion partner, location, and time. Wishart *et al.* [108] used an ontological representation of context data in which users are able to define preferences by setting an obfuscation level applied to data based on the current situation. For example, the user specifies a preference to disclose the current activity only to friends. Especially for D2D group communication, we need privacy protection during the exchange of context data among a group of people. The approach of Franz *et al.* [109] negotiates a privacy policy among all group members including which data is published and at which accuracy. For instance, a group of travelers visit Europe and to allow new people to join the group, information about the group like the current location and cultural interests should be published. However, the group member Alice only allows release of her location at the city level and Bob prefers to hide his membership completely.

C. Anonymity

Obfuscation hides the user identity by reducing the data accuracy. This may result in a negative impact on the service quality. Anonymity-based techniques overcome this problem by protecting the user identity without sacrificing the information accuracy.

However, the security approaches for D2D need authentication, which contradicts anonymity. The PrimeLife project [110] defined an anonymous authentication by adopting cryptographic primitives to prove attributes to a third party without revealing the user identity. In the D2D domain, we need to couple the anonymization technique with a reputation mechanism to create trust among the anonymous conversation entities. In this way, the mobile users feel more comfortable and are willing to share more sensitive content, even if they are sharing content with strangers. The work of Christin *et al.* [111] anonymously verifies the reputation score of users by using periodically changing pseudonyms associated with a reputation level. The cryptographic blind signatures are used to prove the source reputation without revealing individual user identity.

We prefer anonymity techniques that are not dependent on centralized user-trusted entities due to the opportunistic D2D communication. Boutsis and Kalogeraki [112] share users' trajectory paths across mobile devices. Each user knows only a small part of the trajectory and cannot identify the information source. Anonymity mechanisms should consider malicious users who may take advantage of anonymity for illegal actions. In this case, it is necessary to identify the user. The PEACE framework [41] splits all critical information like user identity and group secret keys into two parts and distributes them across different entities, such as group manager and network provider. No entity can determine user's essential attributes or compromise privacy unless two entities collude. The collusion between two entities allows the identification of users performing illegal actions. The PEACE framework as stated above

achieves user access control, user accountability, k-anonymity and non-linkability through the separation of powers.

Pseudonyms are another idea to achieve anonymity. By definition, a pseudonym is an identifier of a subject other than one of the subject's real names [113]. Petit *et al.* [114] identified two essential pseudonym requirements to ensure privacy. A new pseudonym should always be available in case of pseudonym change and a pseudonym must have a validity period to avoid tracking. However, since each pseudonym is unique, all corresponding messages are linkable. We need additional techniques to exchange pseudonyms between mobile users for non-linkability. These mechanisms can be categorized into three groups:

- *Periodical change*: randomize the period to change pseudonyms. Eckhoff *et al.* [115] designed a time-slotted pseudonym pool with swapping functionality. Every mobile user has a pseudonym pool and uses each pseudonym for a specific time slot.
- *Context-based mix zone*: detect and create a dynamic mix zone in social spots such as crowded environments [116]. Inside the mix zone users don't send position updates. Each user receives a new pseudonym when leaving the mix zone [117].
- *Collaboration*: nearby users communicate with each other to synchronize their pseudonyms to confuse the adversary. Pan and Li [118] proposed a cooperative pseudonym scheme based on the number of surrounding users. The mobile device monitors the neighbors within a certain radius. The pseudonym exchange occurs only when the predefined threshold of nearby users is reached.

D. Cryptography

In this section we review cryptographic techniques applicable to D2D communication. We have to include cryptographic mechanisms to increase the reliability of security and privacy approaches for D2D. Our focus is on lightweight mechanisms due to resource constraints of mobile devices with respect to computation power and energy consumption.

The presented cryptographic approaches achieve several privacy goals, such as anonymity, unlinkability, content privacy, confidentiality, and integrity when exchanging messages between mobile users. A widely used standard approach is the Public Key Infrastructure (PKI) in which each participant has private and public keys to authenticate messages. However, the PKI should be modified to fulfill several privacy requirements. Certificates shouldn't contain identifying information about the owner. And keys should be changed periodically to avoid linking of signed messages by the same certificate. Raya and Hubaux [119] presented an approach where each user obtains two certificates. An unique long-term identity together with a key pair and multiple pseudonyms associated with anonymous key pairs to sign messages. Key management and distribution is a major problem for heterogeneous environments like D2D. Nagy *et al.* [120] state that the problem of sharing public and private keys to securely communicate is not solved. They leverage single sign on and authorization mechanism like OAuth 2.0 of a social network (e.g., Facebook) to avoid the key management problem.

Multi-party and distributed cryptographic protocols are important for D2D because they fit the natural properties of standalone D2D environments in which users are distributed without mutual trust. We introduce the idea of Identity-based Cryptography (IBC) [121]. In IBC, each mobile user is able to create a public key through locally available information, such as a phone number or email address. This removes the need to certify the public key and we are able to directly exchange certificates within messages. Nevertheless, IBC requires a centralized trusted authority, which owns a master private key to generate private keys for each user.

Signature schemes, such as group signature, provide anonymity and unlinkability for mobile users. Each group member has a private key and signs messages anonymously on behalf of the group. Other members use a shared group key to verify signed messages without revealing who signed them.

Homomorphic encryption (HE) is another interesting class of cryptographic schemes for D2D communication, especially when requesting data from untrusted entities. HE allows users to perform operations on encrypted cipertext without knowing the original data [122]. Thereby, HE produces the same encrypted result on cipertext as operations executed on plaintext. This is important for environments where the computation occurs on different servers that don't trust each other. Two known homomorphic cryptosystems are Paillier [123] and ElGamal [124]. The proposed systems are semantically secure so that it is impossible to derive any information about the plaintext, given its ciphertext and public key. Paillier decrypts arbitrarily large plaintexts very efficiently, but operations like multiplication and exponentiation are expensive. In contrast, ElGamal's scheme is more efficient regarding computational cost, though it only decrypts small plaintext values. For instance, Mu and Bakiras [125] applied homomorphic encryption to privately identify whether friends are within a nearby distance without revealing the actual user identities.

We can apply Private Information Retrieval (PIR) to protect content in D2D communication. The receiver queries data and the sender does not discover anything about the specific data requested. PIR ensures the privacy of the receiver. Solutions based on PIR usually aim at retrieving information from the nearest neighbor with respect to the current user position [48]. Ghinita *et al.* [126] applied PIR to answer queries without learning or revealing any information about the query. To achieve this goal, PIR relies on the quadratic residuosity assumption; a computationally difficult task to find the quadratic residues for the product of two large primes [54], [126]. The PIR approach does not require a trusted third party and offers strong privacy guarantees. Its major disadvantage is a high computation and communication overhead, which is a concern for resource constrained D2D mobile devices.

Finally, Searchable Encryption (SE) is a new approach applicable to D2D to enable private search on external storage. Bösch *et al.* [127] provided an extensive review on provably secure searchable encryption. The main idea is to encrypt a search index generated over data collection so its content is hidden without appropriate tokens. The tokens can only be generated with a secret key [122]. The search process is as

follows: given a token for a keyword, an user can retrieve pointers to the encrypted data files containing the keyword.

E. Application-Oriented Privacy

In this section, we summarize application-oriented privacy schemes for D2D communication including communication privacy, device and application privacy, and location privacy.

1) Communication Privacy: The environment, in which D2D communication is used, frequently changes with respect to the number of D2D communication partners. D2D communication refers to dynamic, self-forming, self-organizing (autonomous) peer-to-peer networks [34]. The D2D system has no central authority in contrast to conventional infrastructure-based last-hop-wireless networks, where the network provider acts as TTP [34]. In standalone D2D, the adversary must break in a number of D2D devices to achieve a reasonable amount of user information. On the other hand, when an attacker compromises D2D nodes, the attack detection takes more time, which is a benefit for the adversary.

Currently, wireless systems are very limited regarding user privacy and are not satisfactory [41]. Global System for Mobile Communications (GSM) provides a low level of anonymity, mainly protecting the user identity from an eavesdropper by using short-term temporary mobile subscriber identity (TMSI). We needed additional mechanisms to reach the goal of privacy-preserving communications to protect the content and identity of communicating users.

In addition to standard approaches against eavesdropping, we can use pseudonyms and signature-based techniques to enhance user privacy. Public key based approaches can be challenging to deploy because of the distributed nature of D2D communications. Symmetric-key encryption or Identity-Based Cryptography (IBC) [32], [121] are preferred, instead of infrastructure-dependent schemes. IBC enables message encryption and signature verification. The public key in IBC is derived from unique identity information, such as a phone number or email address and the private key is generated by a private key generator (PKG) [34]. The Hierarchical Identity-based Cryptography (HIBC) is an extension of IBC and considers multiple geographical regions for which different PKGs for each region are needed. As a result, IBC is not better than traditional PKI regarding authentication, although it is preferential due to less required network connectivity.

Anonymous authentication is another important aspect for communication privacy. The basic idea is to hide the particular user identity, but at the same time verify the legitimacy of the user [41]. There are three major signature schemes to achieve anonymous authentication. The blind signature [128] in which message content is disguised from its signer. The user obtains the blind signature from the service provider and unblinds it to use as an authentication token. The ring signature [129] in which the actual signer declares a set of possible signers to compute a message signature by using his or her own secret key and the public keys of others. The recipient verifies the signature from one of the declared signers and is able to exchange authoritative secrets in an anonymous manner. The main drawback of these two schemes is the irrevocable anonymity, which does not support the detection of illegal user behavior or insider attacks. The group signature [130] uses k-anonymity to achieve user privacy. The verifier only checks whether a group member has signed the message. This scheme has the ability to revoke user anonymity to account for malicious users.

Cryptographic mechanisms to protect message contents are vulnerable to traffic analysis. For example, the message paths can be revealed due to detection of source and destination by measuring the transmission rate. In this case, we need randomized communications to achieve anonymity. Koh *et al.* [131] introduced randomness in routing paths by phantom receivers and allowed the actual destination node to randomly forward messages to random phantom receivers. In general, existing privacy-preserving network schemes can be classified into non-network coding [132], [133] and network coding [134], [135]. The authors of [132] randomly injected dummy packets into the routing path to create multiple routes. Mehta *et al.* [133] hid the source and destination by using fake sources and receivers to periodically generate dummy traffic. The work of [134] proposed homomorphic encryption with network coding to enhance user privacy. Network coding provides an intrinsic mixing feature, such as Mix-net [136], where the mix nodes reorder and shuffle transmitted messages. In [135], the authors combined network coding with the Onion routing concept to achieve unlinkability.

2) Device and Application Privacy: The security and privacy of the mobile device is important for secure D2D communication because the mobile device executes applications to enable D2D services.

In the following, we highlight key characteristics of mobile security and privacy [137]. The mobile device is strongly personalized because the device owner is its unique user. In addition, mobile devices are most of the time connected to a wireless network to use helpful services like navigation. Finally, the technology convergence in which a single mobile device combines different technologies allows a series of attacks. For example, a privacy infringing attack on a mobile device can leak a user's phone-related information, e.g., contacts, messages, call logs or information derived from sensors. Such an attack can corrupt the integrity and confidentiality of D2D-based services.

Device-oriented privacy refers to a mobile trusted platform that can fulfill several attributes of a basic security mechanism for mobile devices [138]:

- *Platform integrity*: we need to verify the integrity of the platform code. Boot time integrity alone is insufficient, since the attacker can still modify the system after the boot process. Thus, we need a trusted software component that continuously monitors the platform integrity and repairs modified components automatically [139].
- *Secure storage*: a common way to secure storage is a confidential and integrity-protected device-specific key that can be accessed only by authorized code.
- *Isolated execution*: each software component is isolated and can only access other resources of the mobile platform with extra permission. The isolated execution in

combination with secure storage constitutes a trusted execution environment.
- *Device authentication*: external service is able to verify the authenticity of the mobile device.
- *Attestation and provisioning*: external service provider verifies that the device is running a compliant platform version.

Application-oriented privacy is mainly related to monitoring and analyzing mobile applications. The survey reported in [140] provides a recent and comprehensive overview on securing Android phones. The most active research areas in this domain include untrusted application analysis [141], [142] and continuous runtime monitoring [143], [144], [145], [146]. As an application analysis approach, FlowDroid [141] detects privacy leaks through static source code analysis. It performs a flow, context, object, and field-sensitive static taint analysis on Android apps. AppIntent [142] applies static and dynamic code analysis to execute the app in a real or virtual environment. The goal is to check if a data transmission by an app is intended by the user. The static taint analysis generates an event graph including all actions that can lead to a data transmission. Afterwards, the symbolic execution is based on this graph and produces a sequence of UI interactions and data inputs that yield to a data transmission.

For continuous runtime monitoring, the most notable applications with corresponding applied technique to prevent sensitive information leakage are TaintDroid (dynamic taint analysis), BayesDroid (bayesian-based privacy), MockDroid (resource access mocking), TISSA (resource access mocking), AppFence (dynamic taint analysis and resource access mocking), and LP-Guardian (location access regulation) [147], [140]. TaintDroid [143], [144] detects inter-application privacy leaks by applying dynamic taint analysis to observe potential privacy-infringing behavior. It marks any data from sensitive sources as tainted. AppFence [145] identifies the disclosure of data that has been obfuscated, encrypted or transmitted via SSL. This applied technique combines data shadowing of MockDroid and TISSA with taint analysis as in TaintDroid. A recent system called Haystack [146] aims at monitoring encrypted and non-encrypted network communication on mobile phones to inform the user in case of data leakage. A major disadvantage of all of these approaches is the required rooting of the mobile operating system, only Haystack runs entirely in the user space.

The mobile operating system, like Android, provides additional privacy protection [148]. The mobile application must explicitly declare required access to system resources and the permission mechanism of Android ensures that only these system resources are accessed. This is an all-or-nothing approach and in reality we need a more fine-grained permission access control as suggested in the work of Shen *et al.* [149]. These authors proposed flow permissions to provide additional information regarding how apps leverage standard Android permissions and resources.

The mobile operating system uses a sandbox mechanism to identify and isolate application resources; however, the malware DroidDream has broken this sandbox and stolen large amounts of private data. Thus, we need a stronger separation of mobile applications like the approach proposed by Wu *et al.* [150] known as AirBag, which is a lightweight OS-level virtualization to isolate and prevent malware from infecting systems.

The mobile application that realizes the D2D communication should directly consider the privacy-by-architecture principle during the system design phase. This architecture reaches a higher security level by minimizing personal data, using anonymization, client-side storage, and client-side processing [50]. Multiple studies [151], [152], [153], [154] have shown that users want a mechanism to select different security and privacy levels depending on the target group. Several design principles have been identified to facilitate the implementation of privacy-aware applications [155]. The privacy-by-policy principle is related to process-oriented strategies to protect personal data and their relationships by anonmyization, pseudonyms, encryption or k-anonymity. The privacy-by-architecture principle refers to data-oriented strategies to inform data subjects when processing personal data or using privacy policies for data access control.

3) Location Privacy: The heavy usage of location information makes mobile users different from desktop users. Location-based Services (LBS) use a TTP, which receives location data from the mobile users to provide location-specific information, mostly for navigation tasks. This centralized architecture is vulnerable to multiple adversaries and a typical attacker is the service provider itself [48]. In D2D architecture, the first step is to detect mobile devices located nearby before we are able to establish a network connection between potential conversation partners. D2D users are often in close proximity to one another due to the short range of wireless communications making location privacy all the more important in D2D. The term location privacy describes the sensitive association between user identity and location. The following section provides a detailed overview of techniques to maintain location privacy.

The work of Wernke *et al.* [54] provides an in-depth analysis of location privacy attacks and available protection mechanisms. The protection targets include:
- *User identity*: attacker derives user's identity by position information and context data (visited objects as quasi-identifiers).
- *Position*: semantic of location defines criticality of position information, e.g., infer the health status of a user based on frequency of hospital stays.
- *Time*: the time records required for validation of spatial information. In some scenarios, the spatial information is only critical when combined with time. For example, home and work locations can be inferred by the frequency of visited places and the time being spent their.

The adversary knowledge and the attack type are strongly influence the effectiveness of the protection techniques. The attacker knowledge can be classified into two dimensions: temporal information and context information [54]. Temporal knowledge refers to, whether the attacker receives a single user position or continuous position updates, such as movement trajectories. Besides that, if the adversary has access to additional context knowledge beyond spatiotemporal information, such

as maps, building opening hours or a phone book to narrow possible whereabouts. Many privacy approaches assume a weak adversary taking into account single user positions without context information [54]. However, a more realistic privacy scheme should consider a more advanced adversary to guarantee sufficient protection.

In the following section, we classify and highlight approaches for location privacy [156], [54], [157], [40]. These approaches focus mainly on anonymity and obfuscation.

Anonymity techniques aim at the dissociation of information about an individual, such as location from the mobile user to hide the person's identity. Most approaches are based on k-anonymity, a general privacy concept, which stipulates that the target object is indistinguishable from the other $k-1$ objects. Gruteser and Grunwald [158] introduced the concept of k-anonymity for location privacy. The location server acts as a trusted anonymizer and calculates the obfuscation area containing k users based on previously reported positions from mobile users. Afterwards, the location-based service receives only the obfuscation area and is not able to uniquely identify a specific user. Many other approaches extended the k-anonymity concept to enhance privacy protection. The most prominent extensions are strong k-anonymity, l-diversity, t-closeness, p-sensitivity, and historical k-anonymity [54]. Dürr *et al.* [159] applied position sharing to improve the privacy of mobile users. The obfuscated positions are split into position shares and distributed among non-trusted location servers (LS). Thus, each LS has information with only limited precision and the attacker must compromise multiple LSs to acquire sufficient location information to identify users. The approach of position dummies is another concept used to hide the user's identity [160]. The user sends multiple false positions ("dummies") to the LS together with true user position. The advantage of dummy positions is that a TTP is not needed but it is difficult to create dummies not distinguishable from true user position [54].

A special type of anonymity is pseudonymity: the individual is anonymous, but maintains a persistent identity, a pseudonym [40]. Beresford *et al.* [117] proposed an idea to define areas called mix zones. The user does not send position updates and changes its pseudonym with all other users within the mix zone. This approach protects the user identity because the attacker cannot correlate different pseudonyms. The Caché system [161] enhances privacy by pre-fetching location content in large geographic blocks during the night for use the next day. The content is locally accessed when actually needed. This approach increases the bandwidth and storage requirements.

Obfuscation mechanisms degrade the quality of information about a person's location to protect user identity. In general, obfuscation does not require a TTP. Three distinct techniques can be identified from the literature to degrade the quality of location information: (1) Inaccuracy: actual location differs from transmitted location, (2) Imprecision: the region is larger than the actual location, and (3) Vagueness: linguistic terms describe the geographic position [40]. Gutscher et al. proposed an approach based on coordinate transformation [162]. The mobile user performs simple geometric operations, such as shift or rotation over the positions, before sending them to the LS. The transformation function must be distributed among the clients to recover the original position. SpaceTwist [163] is a more advanced approach for location privacy. The user sends a so-called anchor, a fake location to the LS. Afterwards, the user receives multiple data points over the anchor point with various distances to the anchor. Then the mobile user calculates the query results based on his precise position and the data points received. This method achieves location privacy but incurs higher query and communication costs. Further approaches for location privacy use trajectory transformation [164], path cloaking [165] or virtual trip lines [166]. Many obfuscation-based techniques face the challenge that the adversary can significantly reduce the obfuscation area by map knowledge. For instance, the attacker can infer the movement form, for example, a car. With the aid of a road map, the attacker is able to narrow down the user location. One solution to this problem is landscape-aware obfuscation as proposed by [167]. This approach expands the obfuscation area based on a probability distribution function defining the probability that a user is located in a specific area.

Another class of approaches for location privacy include encryption and Private Information Retrieval (PIR). Mu and Bakiras [125] proposed a secure two-party computation protocol based on public key homomorphic encryption for private proximity detection. In this proposal, it is infeasible to derive any information about the plaintext given ciphertext and public key. A secure two-party computation jointly computes a function based on the inputs without revealing input to other parties. Other authors use a centralized client-server architecture for private and flexible proximity detection [168]. Users map their location into four grid cells and send the encrypted location by one-to-one encryption shared among the other users to the server. The server calculates the proximity based on encrypted location and shortest Euclidean distance. Mascetti *et al.* [169], [170] proposed a set of protocols including Hide&Crypt to share a secret key and use secure multi-party computation to encrypt locations before transmission. The idea of PIR [171] is that the location server answers queries without learning or revealing any information of the query. PIR provides stronger and provable location privacy. The technique does not disclose spatial information and prevents any type of location-based attack. The significant computational overhead is a major drawback, particularly for resource restricted mobile devices.

Many approaches in the area of location privacy assume a TTP as service provider, but it is questionable whether the assumption of a TTP is realistic for D2D communication due to a missing central authority. Thus, we prefer TTP-independent solutions based on direct collaboration of mobile users, obfuscation or PIR-based methods [172].

V. Discussion

In this section we outline the security and privacy solutions for D2D communication, which were reviewed in this paper. We highlight the lessons and "best practices" derived from our review of the existing work. We also identify open problems that deserve further investigation.

A. Overview of D2D Security and Privacy Solutions

We categorize the security solutions highlighted in Table V and Table VI based on targeted scenarios and security requirements. We focus on network security with regard to cryptographic design [27], [173], [69], [77], pairing and discovery [174], [86], [92], and distributed algorithms [79], [80], [175], [87]. The application scenarios span across M2M [55], DTN [58], [59], public safety [73] and mobile networks [68], [70], [72], [83], [85], [87], [88], [91]. The D2D security requirements include non-repudiation (NR), authentication and authorization (AA), confidentiality and integrity (CI), availability and dependability (AD), and secure routing and transmission (SRT), as referred in Table II. We also highlight in Table V and Table VI the main technique applied in each proposal and the corresponding security requirements. We deliberately select work published from year 2012 up to 2016 in order to reflect the latest advancements on top of the security research in mobile ad hoc networks [24], [176], [177]. The solutions included in this paper shall provide us with a snapshot of the most recent work dedicated to D2D security.

For D2D privacy solutions shown in Table VII and Table VIII, we categorize them based on scenarios and privacy requirements. To reflect the attacks depicted in Table III, we focus on two dimensions: device privacy and network privacy. For device privacy, we cover access control [103], [104], [106], [107], privacy policy [108], [109], application analysis [141], [142], [144], data leakage [145], [146], and mobile operating systems [149], [150]. Concerning network privacy, we consider anonymity [41], [115], [116], [118], [131], [132], [133], [134], [135], trust [111], access control [41], communication [119], [120], [121], [126], [131], [132], [133], [134], [135], storage access [127], private proximity testing [125], and location privacy [112], [117], [158], [159], [160], [161], [162], [163], [167], [168], [169], [171]. The privacy requirements include anonymity and indistinguishability (AI), unlinkability (U), content privacy (CP), confidentiality and integrity (CI), and deniability (D), as shown in Table II. For each paper reviewed, we summarize the research technique employed for preserving privacy and the conformed privacy requirements.In difference to the conventional reviews that treat privacy as a branch of security aspects [178], [179], [32], [5], [39], [53], we aim to provide a comprehensive selection of privacy schemes (from 2003 till 2015) that can be applied to D2D communication.

B. Lessons Learned and Best Practices

Based on the reviewed papers, we derive a set of lessons learned and "best practices" to be considered in implementing and deploying D2D security and privacy solutions. The key criteria for security and privacy solutions include D2D device consideration, physical layer design, user aspects, and solution compatibility.

1) Device Diversity and Limitation: Owing to the technology advancement in mobile and wireless communication, the devices used in D2D communication are becoming diverse, ranging from wearable devices, smartphones, tablets to smart vehicles. These devices typically deploy different software stacks and exhibit a distinct set of traits in terms of mobility, computing capability, and use cases. This diversity is a key concern in applying security and privacy schemes in D2D environments. Regarding software stack, the security holes in operating systems, as indicated in [48], can result in severe privacy breaches regardless of the protection mechanisms deployed on the application level. To complicate the situation, the fragmentation of mobile operating systems has put further pressure on the limited time available for software development. Hence, developers tend to prioritize service functionality over security and privacy features. Besides software, research proposals typically take these practical factors for granted (e.g., to simplify assumptions) resulting in a limited application scope. To this end, we recommend the adoption of security and privacy schemes on a case by case basis by considering the characteristics of devices, system software and application scenarios. The solution tables summarized in this article can serve as a reference to match dedicated scenarios to solution requirements.

Practical limitations, such as battery life and processing units on mobile devices, also restrict the usage of security and privacy schemes that tend to be power-consuming and computation-demanding. This is especially important for low end devices used in D2D communication. Several reviewed proposals [173], [174], [69], [83], [84], [92], [180] aim to optimize authentication, encryption, and key management. We recommend system level energy-efficient solutions such as Odyssey [181], ErdOS [182], and Blue-Fi [183] to compensate the introduced security overhead by improving the overall system energy saving. In this respect, there are sufficient research studies on mobile energy efficiency [184], [185], [186], [187], [188] that can be considered in the context of D2D (details of energy efficient techniques are beyond the scope of this article).

2) Physical Layer Considerations: The existing cellular security architecture is defined by five security levels comprising (i) network access security, ii) network domain security, (iii) user domain security, (iv) application domain security, and (v) non 3GPP domain security [20]. The security architecture of LTE systems has enlisted basic security aspects including the D2D security 1) between 3GPP networks and the proximity service (ProSe) function/application server, 2) between D2D devices and ProSe function/application server, and 3) between individual D2D devices [39].

Aside from physical layer considerations in conventional MANET security [24], [176], [177], physical layer security in D2D communication also deserves our attention. In specific, physical layer security schemes attempt to create security cardinal by analyzing the physical characteristics of wireless channels between D2D devices. The security studies by Wang *et al.* [10] underlined several scenarios and use cases for D2D. The security threats consist of impersonation attack, threats related to data transmission security and UE mobility and privacy. A general perception is that the D2D security framework that can unify security solutions is not yet matured.

3) User Perspectives: Raising user awareness of security and privacy threats is a key step to boost the adoption of the proposed schemes for D2D communication. Most users are

TABLE V
COMPARISON OF D2D SECURITY SOLUTIONS

Approach				Security Requirements				
Ref	Year	Target Scenario	Technique Employed	NR	AA	CI	AD	SRT
[55]	2013	Network - Key Management	Public key crypto system to secure M2M systems including key generation, encryption, and decryption.	-	Y	-	-	-
[56]	2012	Network - Key Management	Key agreement and batch authentication for P2P based OSNs. Therefore, it applies one-way hash function, ElGamal proxy encryption, and certificate based protocol.	Y	Y	Y	-	-
[57]	2014	Network - Key Management	Batch authentication to offer an efficient one-to-many authentication approach for P2P based networks.	-	Y	Y	-	-
[58]	2014	Network - Key Management	ABE for authenticating routing messages. The routing node encrypts the symmetric key using ABE and then distributes it to all participating nodes. Only those nodes that match a specific attribute policy are able to extract the key.	-	Y	Y	-	Y
[59]	2014	Network - Key Management	Attribute-based secure data retrieval scheme using CP-ABE. The approach provides attribute revocation, fine-grained access policy over attributes, and solves the key escrow problem.	-	Y	Y	-	Y
[60]	2015	Network - Key Management	Group key agreement protocol based on ECC. The users securely communicate via a session key, which is received from a trusted third party.	-	Y	-	-	-
[61]	2015	Network - Key Management	Many-to-many group key management protocol based on ECC for key distribution.	Y	Y	-	-	-
[62]	2012	Network - Key Management	Group based authentication and GKA allows each M2M device to share secret keys with other M2M devices of the same group.	-	Y	-	-	-
[63]	2013	Network - Key Management	Lightweight group authentication protocol for M2M communication based on message authentication codes.	-	Y	-	-	-
[64]	2016	Network - Key Management	Asynchronous secret share along with Diffie-Hellman key exchange for authentication in LTE-A networks.	-	Y	-	-	-
[65]	2012	Network - Key Management	Group based access authentication by aggregation signature.	-	Y	-	-	-
[66]	2016	Network - Key Management	Multi-keyword ranked search operation over encrypted data to securely find localized content.	-	Y	-	-	-
[67]	2016	Network - Key Management	Extension work on multi-keyword ranked search operation.	-	Y	-	-	-
[68]	2014	Network - Authentication	Joint operation protocol to control the D2D network and manage the group key in self-organized groups of ad hoc nodes.	-	Y	-	-	Y
[69]	2014	Network - Authentication	Diffie-Hellman key agreement and commitment schemes for transmission in D2D communications.	-	Y	-	-	-
[70]	2014	Network - Authentication	PKC based on digital signature along with mutual authentication for end-to-end security.	-	Y	Y	-	-
[72]	2015	Network - Authentication	SeDS protocol based on DHKE and HMAC digital signature to provide authentication and malicious node detection.	Y	Y	Y	Y	-
[73]	2014	Network - Authentication	Protocol broadcasts a beacon to nearby devices to set up a D2D communication and then uses a random pre-distribution encryption key for authentication.	-	Y	-	Y	-
[74]	2015	Network - Authentication	Use channel randomness to create a shared secret key for direct communication links.	-	Y	-	-	-
[75]	2016	Network - Authentication	Secret key generation scheme for untrusted relays.	-	Y	-	-	-
[76]	2015	Network - Authentication	Full duplex relay jamming scheme for secret key generation.	-	Y	-	-	-
[77]	2015	Network - Confidentiality and Integrity	Fast secret key extraction protocol called KEEP to obtain secret keys from CSI measurements.	Y	-	Y	-	-
[78]	2015	Network - Confidentiality and Integrity	Power allocation technique for the generation of secret keys in relay based LTE-A networks.	-	-	Y	-	-
[79]	2014	Network - Confidentiality and Integrity	Cooperative key generation to set up shared secret keys between devices.	-	-	Y	-	-
[80]	2015	Network - Confidentiality and Integrity	Secure load balancing algorithm names as LBS-AOMDV to reduce the impact of confidentiality attacks.	Y	-	Y	-	-
[81]	2014	Network - Confidentiality and Integrity	Privacy preserving mutual authentication, in which only users with similar attributes can decrypt the content.	-	-	Y	-	-
[82]	2014	Network - Confidentiality and Integrity	Clients collaborate to ensure data confidentiality and integrity when using an untrusted service provider.	-	-	Y	-	-
[83]	2015	Network - Availability and Dependability	Wireless Power Transfer Policy (WPTP) and an information signal model to enable wireless energy harvesting and secure information transmission.	-	-	-	Y	Y
[84]	2016	Network - Availability and Dependability	Wireless power transfer policies for secure D2D communication including CPB-PT, BPB-PT, and NPB-PT.	-	-	-	Y	Y
[85]	2015	Network - Availability and Dependability	Interference management scheme to enhance physical layer security.	-	-	-	Y	Y
[86]	2015	Network - Availability and Dependability	IBE to secure the exchanged D2D messages during discovery and communication.	Y	-	-	Y	-
[87]	2014	Network - Availability and Dependability	Kuhn Munkers Algorithm (KMA) to find the maximum sum secrecy capacity for both cellular and D2D users.	-	-	-	Y	-

TABLE VI
CONTINUED COMPARISON OF D2D SECURITY SOLUTIONS

Approach				Privacy Requirements				
Ref	Year	Target Scenario	Technique Employed	NR	AA	CI	AD	SRT
[88]	2015	Network - Secure Routing and Transmission	Stackelberg game to maximize the rate of cellular users and secrecy capacity of D2D links.	-	-	-	-	Y
[89]	2016	Network - Secure Routing and Transmission	ARSP policy in which the users can only create a connection with the base station providing highest ARSP value.	-	-	-	-	Y
[90]	2015	Network - Secure Routing and Transmission	Approximation solution based on Bernstein type inequality and S-procedure to optimize power consumption and secrecy rate.	-	-	-	-	Y
[91]	2015	Network - Secure Routing and Transmission	Interference avoidance scheme for cooperative D2D communication in cellular systems.	-	-	-	-	Y
[92]	2014	Network - Secure Routing and Transmission	SMD protocol to securely transmit data from source to destination.	-	-	-	-	Y
[93]	2014	Network - Secure Routing and Transmission	Secure policy agreement for open-privacy routing in wireless communications.	-	-	-	-	Y
[94]	2015	Network - Secure Routing and Transmission	IBHC to protect ad hoc wireless networks against heterogeneous attacks.	-	-	-	-	Y
[95]	2015	Network - Secure Routing and Transmission	Puncturable encryption to achieve forward secure encryption in store and forward messaging systems.	Y	-	-	-	Y
[96]	2014	Network - Secure Routing and Transmission	Dynamic trust management for secure routing optimization.	-	Y	-	-	Y
[97]	2014	Network - Secure Routing and Transmission	TBER scheme to detect and reject malicious nodes.	-	Y	-	-	Y
[98]	2015	Network - Secure Routing and Transmission	ICN monitors all information exchanged in DTNs to detect misbehaving nodes and select alternative links.	-	Y	-	-	Y
[99]	2014	Network - Secure Routing and Transmission	CFV to reduce the harmful effects of malicious nodes in the network.	-	Y	-	-	Y
[100]	2015	Network - Secure Routing and Transmission	Fawkes Routers to verify node interactions.	-	Y	-	-	Y

concerned about personal data protection on mobile devices, as indicated in [189], [190]. A great majority among reviewed users worry about stealing personal information and identity (84 %), and loss of privacy (83 %). About half of users, 49 % would feel more comfortable if they had better control of their private information. Regardless of the general awareness, D2D users might still underestimate the potential threats following exposure of their sensitive information, leading to the perception that security and privacy are unnecessary abstractions. This observation suggests that we should not only enforce security and privacy on devices and communication channels, we should also have effective tools [146], [107], [106] that can manage external access to personal data and explain the effects of data leakage to users.

For D2D privacy, one vital concern deals with user mobility datasets, which are widely used in mobility modeling and location privacy research. A study of human mobility data over 15 months on one and a half million individuals revealed that the uniqueness of human mobility traces is high [191]. The findings indicate that even coarse or blurred mobility datasets provide little anonymity. It is hence possible to re-identify the traces of a targeted individual with the support of a few additional pieces of information (e.g., four spatio-temporal points). As pointed out in [48], privacy protection mechanisms derived from the database anonymity notions are typically based on the predefined background knowledge of possible adversaries. If the adversarial knowledge is different from the assumption, the protected user identity can be easily revealed. Since mobility data is among the most sensitive data we can collect about individuals, we emphasize this lesson in processing mobility datasets and urge a more comprehensive privacy awareness in D2D research.

4) Solution Compatibility and Deployability: Cellular operators are the main driving force for D2D communication [192], [21], which have identified a set of use cases and applications, such as public safety and proximity services. It is important for security and privacy proposals to consider the compatibility with existing and upcoming mobile networks such as LTE/4G and 5G. Regarding the security and privacy proposals dedicated to mobile networks [68], [70], [72], [83], [85], [87], [88], [91], compatibility has been discussed within the context of general mobile access. Based on this observation, we recommend an explicit reference to the 3GPP standards [193], [194] when designing new solutions for D2D security and privacy. We should also be aware of the potential incompatibility between the business models that profit on personal data and the privacy schemes that reduce the fidelity of personal information.

A user friendly and transparent design is preferred regarding deployability. Good examples are the HayStack [146] and Securebox [195] approaches, which strive to detect privacy leakage and security threats on mobile devices in a non-intrusive manner. Based on our observations, a purely infrastructure-independent D2D design is not realistic to meet all the requirements of security and privacy in the current phase. An intermediate step could be a hybrid infrastructure-assisted design in which one mobile node has access to the cellular network and can provide services to other mobile devices, such as group anonymous authentication [70]. This special node can act as a gateway / entry point to the infrastructure and services. The direct benefit is that we can adopt existing security and privacy models for a centralized environment, such as secure multi-party computation (SMC), fully homomorphic encryption (FHE), and one-way trapdoor

TABLE VII
COMPARISON OF D2D PRIVACY SOLUTIONS

Approach				Privacy Requirements				
Ref	Year	Target Scenario	Technique Employed	AI	U	CP	CI	D
[103]	2011	Device - Access Control	DAC system based on ontology-based context model to specify complex situations and relationships.	-	-	Y	-	Y
[104]	2012	Device - Access Control	Broker based on trust among users defines level of data disclosure. The raw sensor data is abstracted to context labels, e.g., “noise” or “conversation”.	-	-	Y	-	Y
[106]	2014	Device - Access Control	Similar to differential privacy: framework receives questions submitted by an application and provides only the answer, e.g., play next song, which is calculated within the safe environment of openPDS. Thereby, the framework reduces the dimensionality of metadata.	-	-	Y	-	Y
[107]	2015	Device - Access Control	Fine-grained data access control by using privacy-preserving data analytic techniques, such as differential privacy and homomorphic encryption. Only release the irreversible data aggregation result, so that de-anonymisation becomes impossible.	Y	-	Y	-	Y
[108]	2007	Device - Privacy policy	Ontological representation of context data organized as hierarchy. User sets an obfuscation level applied to released data based on current situation: disclose activity only to friends.	Y	-	Y	-	Y
[109]	2012	Device - Privacy policy	Negotiates a privacy policy among all group members including which data is published and at which accuracy.	Y	-	Y	-	Y
[141]	2014	Device - Application Analysis	Performs flow, context, object, and field-sensitive static taint analysis to detect privacy leaks.	-	-	-	Y	-
[142]	2013	Device - Application Analysis	Static and dynamic code analysis to execute the app in a real or virtual environment. The goal is to identify data transmissions that are not intended by the user.	-	-	-	Y	-
[144]	2014	Device - Application Analysis	Dynamic taint analysis detects privacy-infringing behavior. It marks any data from sensitive sources as tainted.	-	-	-	Y	-
[145]	2011	Device - Data Leakage	Data shadowing together with taint analysis to identify the disclosure of data that has been obfuscated, encrypted or transmitted via SSL.	-	-	-	Y	-
[146]	2015	Device - Data Leakage	Monitors encrypted and non-encrypted network communication by an integrated TLS proxy. The user is informed when the Aho-Corasick algorithm founds sensitive data, e.g., OS fingerprints or contact details in the network data stream.	-	-	-	Y	-
[149]	2014	Device - Mobile Operating System	Flow permissions to provide additional information, how apps leverage standard Android permissions and resources.	-	-	-	Y	-
[150]	2014	Device - Mobile Operating System	Lightweight OS-level virtualization to isolate and prevent malware from infecting systems.	-	-	-	Y	-
[41]	2009	Network - Anonymity and Access control	Separation of powers: split all critical information like user identity and group secret keys into two parts and distribute them across entities, such as group manager and network provider.	Y	Y	-	-	Y
[111]	2013	Network - Anonymity and Trust	Anonymously verify the reputation score of users by periodically changing pseudonyms associated with a reputation level. Moreover, using blind signatures to prove the source reputation without revealing the individual user identity.	Y	Y	-	-	-
[115]	2011	Network - Anonymity	Every mobile user has a time-slotted pseudonym pool with swapping functionality and use each pseudonym for a specific time slot.	Y	Y	-	-	-
[116]	2012	Network - Anonymity and Location privacy	Detect and create a dynamic mix zone in social spots, e.g., crowded environments. Inside the mix zone users don’t send position updates and receive new pseudonyms when leaving the mix zone.	Y	Y	-	-	-
[118]	2013	Network - Anonymity	Cooperative pseudonym scheme based on the number of surrounding users. The mobile device monitors the neighbors within a certain radius and exchanges the pseudonym when the predefined threshold of nearby users is reached.	Y	Y	-	-	-
[119]	2005	Network - Secure Communication	Each user obtains two types of certificates: (1) unique long-term identity and a key pair and (2) multiple pseudonyms associated with anonymous key pairs to sign messages.	Y	Y	-	Y	-
[120]	2013	Network - Secure Communication	Sharing public and private keys to securely communicate is not solved. This approach leverages single sign on and authorization mechanism like OAuth 2.0 of a social network (e.g., Facebook) to avoid the key management problem.	Y	Y	-	Y	-
[121]	2007	Network - Secure Communication	Identity-based Cryptography (IBC): each mobile user is able to create a public key through locally available information like phone number or email address.	Y	Y	-	Y	-
[126]	2008	Network - Secure Communication	Private Information Retrieval (PIR) to answer queries without learning or revealing any information about the query.	Y	Y	-	Y	-
[125]	2013	Network - Private Proximity Testing	Homomorphic encryption, e.g., Paillier or ElGamal to privately identify whether friends are within a nearby distance without revealing the actual user identities.	Y	Y	-	-	-
[127]	2015	Network - Secure Storage Access	Searchable Encryption (SE) enables private search on external storage. SE encrypts a search index generated over a data collection, so its content is hidden without appropriate tokens.	Y	-	-	-	-

TABLE VIII
CONTINUED COMPARISON OF D2D PRIVACY SOLUTIONS

Approach				Privacy Requirements				
Ref	Year	Target Scenario	Technique Employed	AI	U	CP	CI	D
[131]	2015	Network - Anonymous Communication	Network coding and opportunistic routing to introduce randomness in routing paths. The actual destination node randomly forwards messages to random phantom receivers.	Y	Y	Y	-	-
[132]	2008	Network - Anonymous Communication	Randomly inject dummy packets into the routing path to create multiple routes.	Y	Y	Y	-	-
[133]	2012	Network - Anonymous Communication	Hides the source and destination by using fake sources and receivers to periodically generate dummy traffic.	Y	Y	Y	-	-
[134]	2011	Network - Anonymous Communication	Homomorphic encryption with network coding, which provides an intrinsic mixing feature to reorder and shuffle transmitted messages.	Y	Y	Y	Y	-
[135]	2012	Network - Anonymous Communication	Combination of network coding and Onion routing to achieve unlinkability.	Y	Y	Y	-	-
[112]	2013	Network - Location privacy	Position sharing across mobile devices. Each user knows only a small part of the trajectory and cannot identify the information source.	Y	Y	-	-	-
[117]	2004	Network - Location privacy	Define areas called mix zones, in which the user does not send position updates and changes its pseudonym with all other users within the mix zone.	Y	Y	Y	-	-
[158]	2003	Network - Location privacy	k-anonymity: location-based service receives only an obfuscation area containing k users. The target object is indistinguishable from the other k-1 users.	Y	Y	Y	-	-
[159]	2011	Network - Location privacy	Obfuscated positions are split into position shares and distributed among non-trusted location servers (LS). Attacker must compromise multiple LSs to acquire sufficient location information to identify users.	Y	Y	Y	-	-
[160]	2009	Network - Location privacy	User sends multiple false positions ("dummies") to the location server together with true user position.	Y	Y	Y	-	-
[161]	2011	Network - Location privacy	Pre-fetching location content in large geographic blocks during the night. At the next day, only local data access when actually needed.	Y	Y	Y	-	-
[162]	2006	Network - Location privacy	Mobile user performs simple geometric operations, such as shift or rotation over the positions before sending them to the location server.	Y	Y	Y	-	-
[163]	2011	Network - Location privacy	User sends a so-called anchor, a fake location to the location server. Afterwards, user requests data over the anchor point to hide the actual position.	Y	Y	Y	-	-
[167]	2009	Network - Location privacy	Landscape-aware obfuscation, which expands the obfuscation area based on a probability distribution function defining where the user is probably located.	Y	Y	Y	-	-
[168]	2010	Network - Location privacy	Users send their encrypted location by one-to-one encryption shared among the other users to the location server. The server calculates the proximity based on encrypted location and shortest Euclidean distance.	Y	Y	Y	Y	-
[169]	2009	Network - Location privacy	Hide&Crypt protocol to share a secret key and use secure multi-party computation to encrypt locations before transmitting.	Y	Y	Y	Y	-
[171]	2012	Network - Location privacy	Location server uses Private Information Retrieval (PIR) to answer queries without learning or revealing any information of the query.	Y	Y	Y	Y	-

function [196]. Although standardization is a promising way to boost the deployment of security protocols, it is worthwhile to be aware of the efforts and time needed for standardization processes [197].

C. Open Problems

Security and privacy in wireless communications are not newly emerged problems and have been broadly studied [34], [179], [127], [178], [198], [32], [199], [137]. However, there are special concerns for D2D communication owing to new application requirements and use cases. We list open issues that deserve further research. The key criteria we selected include motivation, requirement gaps, quantification, and legal considerations. These aspects are essential to the adoption of D2D and have not yet been fully investigated.

1) User Incentive: It is essential to stimulate users to actively participate in D2D communication, because D2D communication relies on the cooperation of mobile users. The participating entities in D2D are more spontaneous and self-managed in contrast to traditional infrastructure-based communication where auditing and logging are managed by a centralized entity (e.g., in cellular access). As pointed out in [79], D2D users are rational and selfish in nature, which may hinder security operations, such as key generation and distribution. Meanwhile, new attacks continue to occur on new applications and use cases, and on communication channels as well as on device hardware and software. It is hence crucial to enforce security and privacy on D2D communication. While various proposals exist in the broad wireless communication context [175], [200], [118], [135], [201], [202], [203], [204], [205], [206], the effectiveness of applying these incentive / cooperative schemes to D2D communication is not yet evident. In particular for resource constrained D2D devices, how to compensate the power consumption and computing resources needed for security operations is still an open issue. Further investigations are therefore required to explore novel techniques to motivate D2D users.

2) Requirement Gap and Conflict: Through our review, we found one blind-spot in D2D security requirements: non-repudiation (NR), which is poorly supported by existing proposals. The purpose of NR is to provide data verification and data origination [72]. NR is based on cryptographic methods using symmetric or asymmetric techniques to fulfill the following properties:

- approval of message content
- verification of the origin of message content
- proof of message by receiver
- acknowledgment of received message by recipient

The above mentioned NR objectives are necessary so that legitimate D2D users cannot deny transmission or receipt of messages. As a result, the D2D users act cooperatively during data processing and transmission [86], [95]. However, approaches for NR have received little attention in D2D communication and only a few research articles have been published about NR for D2D. Particularly, the dynamic environment with changing conversation partners and different device capabilities in terms of processing power and available energy poses a challenge for NR.

Besides the conflicting requirements highlighted in Section "Security and Privacy Requirements for D2D" (Fig. 2), other conflicting parties are service quality vs. privacy and security. For example, encryption schemes fulfill multiple requirements of privacy and security but can be too heavyweight to achieve the real-time constraints of D2D communication. How to strike a balance among contradicting requirements deserves future studies. The key is to balance user preferences, security and privacy requirements, and service quality.

3) Quantification and Evaluation Tools: Quantification is one open issue for D2D privacy, which is needed for measuring and illustrating the effects of privacy. Regarding quantification models, the k-anonymity [207] and differential privacy [208] models have been widely used in the database community. In the D2D context, a generic analytical framework was proposed recently by Shokri [209], which formalizes and quantifies location privacy to cover user, adversary, attacks, and protection mechanisms. The framework uses a Bayesian Stackelberg game to model conflicting objectives where the goal of the users is to maximize privacy and the adversary tries to minimize the location estimation error for reliable tracking. This approach is available via the tool Location-Privacy Meter [210]. One important finding of their evaluation [210] is that the popular metrics like k-anonymity and entropy are not correlated with the adversary success and therefore inappropriate as location privacy metrics. Aside from the location privacy aspect, the existing literature offers little insight on quantification models and evaluation tools dedicated to D2D communication. We believe these areas deserve further investigation, because metrics and evaluation tools are necessary for objectively comparing different proposals against the security and privacy requirements.

4) Legal and Regulation Concerns: The ethical and legal requirements are non-negligible factors in D2D security and privacy research, due to the connection with national security and public safety [211], [212]. By complying to regulations, we do not intend to prohibit profitable business models. On the other hand, effective regulations are equally important to enforce the deployment of security and privacy solutions in practice. Recently, WhatsApp introduced end-to-end encryption for their application communications [213], [214]. This step should reassure WhatsApp users that their personal communication is secure. The Patriot Act from 2001 eventually forces software vendors to ensure data access for US authorities. At the South by Southwest (SXSW) event, Barack Obama also made clear that the US government must be able to access information when it is entitled to do so under a lawful warrant [215]. In this regard, a crucial and open question is: who is watching the watchers? Microsoft has sued the US Government because the American investigators accessed Microsoft cloud data in secrecy without the awareness of Microsoft customers [216]. D2D communication may face tougher regulation because it offers a decentralized and opportunistic communication pattern, which requires more surveillance efforts.

VI. Conclusion

We review the state-of-the-art solutions to tackle security and privacy challenges in Device-to-Device (D2D) communication. The reviewed approaches span across a variety of D2D prospects, such as network communication, peer discovery, proximity services, and location privacy. In addition to the conventional review on security, we also provide a detailed discussion on D2D privacy. We summarize and compare the existing solutions according to security and privacy requirements. Based on the analysis, we further derive "best practices" and identify open problems that deserve future research. With respect to lessons learned, the major considerations include device diversity, resource limitation, user incentive, solution deployability, requirement conflicts, evaluation tools and legal concerns. We hope that the discussion presented in this review will serve as a reference guide for researchers and developers to facilitate the design and implementation of D2D security and privacy solutions.

References

[1] F. Ghavimi and H.-H. Chen, "M2M Communications in 3GPP LTE/LTE-A Networks: Architectures, Service Requirements, Challenges, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 525–549, 2015.

[2] Gartner, "Gartner Says Worldwide PC, Tablet and Mobile Phone Combined Shipments to Reach 2.4 Billion Units in 2013." [Online]. Available: http://www.gartner.com/newsroom/id/2408515 (visited on 07.04.2016).

[3] ——, "Worldwide Device Shipments to Grow 1.9 Percent in 2016, While End-User Spending to Decline for the First Time." [Online]. Available: http://www.gartner.com/newsroom/id/3187134 (visited on 06.04.2016).

[4] Cisco, "Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020," 03.02.2016. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html (visited on 02.06.2016).

[5] A. Asadi, Q. Wang, and V. Mancuso, "A Survey on Device-to-Device Communication in Cellular Networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.

[6] Y.-D. Lin and Y.-C. Hsu, "Multihop Cellular: A New Architecture for Wireless Communications," in *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2000, pp. 1273–1282.

[7] D. Feng, L. Lu, Y. Yuan-Wu, G. Y. Li, S. Li, and G. Feng, "Device-to-Device Communications in Cellular Networks," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 49–55, 2014.

[8] M. N. Tehrani, M. Uysal, and H. Yanikomeroglu, "Device-to-Device Communication in 5G Cellular Networks: Challenges, Solutions, and Future Directions," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 86–92, 2014.

[9] N. Kato, "On Device-to-Device (D2D) Communication [Editor's Note]," *IEEE Network*, vol. 30, no. 3, p. 2, 2016.

[10] M. Wang and Z. Yan, "A Survey on Security in D2D Communications," *Mobile Networks and Applications*, pp. 1–14, 2016.

[11] R. Alkurd, R. M. Shubair, and I. Abualhaol, "Survey on Device-to-Device Communications: Challenges and Design Issues," in *Proceedings of the IEEE 12th International New Circuits and Systems Conference (NEWCAS)*, 2014, pp. 361–364.

[12] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2007, pp. 46–51.

[13] Qualcomm Technologies, "Creating a Digital 6th Sense with LTE Direct," 2015. [Online]. Available: https://www.qualcomm.com/media/documents/files/creating-a-digital-6th-sense-with-lte-direct.pdf (visited on 07.10.2016).

[14] ——, "LTE Direct Trial: White Paper," 2015. [Online]. Available: https://www.qualcomm.com/media/documents/files/lte-direct-trial-white-paper.pdf (visited on 07.10.2016).

[15] M. Girolami, S. Chessa, and A. Caruso, "On Service Discovery in Mobile Social Networks: Survey and Perspectives," *Computer Networks*, vol. 88, pp. 51–71, 2015.

[16] N. Kayastha, D. Niyato, P. Wang, and E. Hossain, "Applications, Architectures, and Protocol Design Issues for Mobile Social Networks: A Survey," *Proceedings of the IEEE*, vol. 99, no. 12, pp. 2130–2158, 2011.

[17] J. Liu, N. Kato, J. Ma, and N. Kadowaki, "Device-to-Device Communication in LTE-Advanced Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 1923–1940, 2015.

[18] K. W. Choi and Z. Han, "Device-to-Device Discovery for Proximity-Based Service in LTE-Advanced System," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 1, pp. 55–66, 2015.

[19] Y. Zou, X. Wang, and W. Shen, "Physical-Layer Security with Multiuser Scheduling in Cognitive Radio Networks," *IEEE Transactions on Communications*, vol. 61, no. 12, pp. 5103–5113, 2013.

[20] J. Cao, M. Ma, H. Li, Y. Zhang, and Z. Luo, "A Survey on Security Aspects for LTE and LTE-A Networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 283–302, 2014.

[21] X. Lin, J. Andrews, A. Ghosh, and R. Ratasuk, "An Overview of 3GPP Device-to-Device Proximity Services," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 40–48, 2014.

[22] A. Aijaz, H. Aghvami, and M. Amani, "A Survey on Mobile Data Offloading: Technical and Business Perspectives," *IEEE Wireless Communications*, vol. 20, no. 2, pp. 104–112, 2013.

[23] A. Pyattaev, K. Johnsson, S. Andreev, and Y. Koucheryavy, "Proximity-Based Data Offloading via Network Assisted Device-to-Device Communications," in *Proceedings of the IEEE 77th Vehicular Technology Conference (VTC Spring)*, 2013, pp. 1–5.

[24] F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks," in *Proceedings of the 7th International Workshop on Security Protocols*, 1999, pp. 172–182.

[25] 3rd Generation Partnership Project, "Feasibility Study on Remote Management of USIM Application on M2M Equipment: Technical Report 33.812," May 2007. [Online]. Available: ftp://ftp.3gpp.org/tsg_sa/WG3_Security/TSGS3_55_Shanghai/Docs/S3-091154.zip

[26] H. Huang, N. Ahmed, and P. Karthik, "On a New Type of Denial of Service Attack in Wireless Networks: The Distributed Jammer Network," *IEEE Transactions on Wireless Communications*, vol. 10, no. 7, pp. 2316–2324, 2011.

[27] M. Shirvanian and N. Saxena, "Wiretapping via Mimicry: Short Voice Imitation Man-in-the-Middle Attacks on Crypto Phones," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2014, pp. 868–879.

[28] S. Mascetti, L. Bertolaja, and C. Bettini, "A Practical Location Privacy Attack in Proximity Services," in *Proceedings of the 14th IEEE International Conference on Mobile Data Management (MDM)*, 2013, pp. 87–96.

[29] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A Survey on Jamming Attacks and Countermeasures in WSNs," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 42–56, 2009.

[30] B. Wu, J. Chen, J. Wu, and M. Cardei, "A Survey of Attacks and Countermeasures in Mobile Ad Hoc Networks," in *Wireless Network Security*. Springer, 2007, pp. 103–135.

[31] N. Panwar, S. Sharma, and A. K. Singh, "A Survey on 5G: The Next Generation of Mobile Communication," *Physical Communication*, vol. 18, pp. 64–84, 2016.

[32] R. Di Pietro, S. Guarino, N. V. Verde, and J. Domingo-Ferrer, "Security in Wireless Ad-Hoc Networks - A Survey," *Computer Communications*, vol. 51, pp. 1–20, 2014.

[33] Y. S. Shiu, S. Y. Chang, H. C. Wu, S. C. H. Huang, and H. H. Chen, "Physical Layer Security in Wireless Networks: A Tutorial," *IEEE Wireless Communications*, vol. 18, no. 2, pp. 66–74, 2011.

[34] D. Ma and G. Tsudik, "Security and Privacy in Emerging Wireless Networks [Invited Paper]," *IEEE Wireless Communications*, vol. 17, no. 5, pp. 12–21, 2010.

[35] H. Kumar, D. Sarma, and A. Kar, "Security Threats in Wireless Sensor Networks," *IEEE Aerospace and Electronic Systems Magazine*, vol. 23, no. 6, pp. 39–45, 2008.

[36] IETF, "Internet Security Glossary," 2007. [Online]. Available: https://tools.ietf.org/html/rfc4949 (visited on 12.01.2017).

[37] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 6th ed. Pearson, 2014.

[38] X. Lin, "CAT: Building Couples to Early Detect Node Compromise Attack in Wireless Sensor Networks," in *Proceedings of the Global Telecommunications Conference (GLOBECOM)*, 2009, pp. 1–6.

[39] M. Wang and Z. Yan, "Security in D2D Communications: A Review," in *Proceedings of the IEEE Trustcom/BigDataSE/ISPA*, 2015, pp. 1199–1204.

[40] M. Duckham and L. Kulik, "Location Privacy and Location-Aware Computing," in *Dynamic & Mobile GIS: Investigating Change in Space and Time*. CRC Press, 2006, pp. 34–51.

[41] W. Lou and K. Ren, "Security, Privacy, and Accountability in Wireless Access Networks," *IEEE Wireless Communications*, vol. 16, no. 4, pp. 80–87, 2009.

[42] B. Könings, F. Schaub, and M. Weber, "Privacy and Trust in Ambient Intelligent Environments," in *Next Generation Intelligent Environments*. Springer, 2016, pp. 133–164.

[43] D. J. Solove, *Understanding Privacy*. Harvard University Press, 2008.

[44] S. D. Warren and L. D. Brandeis, "The Right to Privacy," *Harvard Law Review*, no. 4, pp. 193–220, 1890.

[45] A. S. Hornby, S. Wehmeier, and M. Ashby, Eds., *Oxford Advanced Learner's Dictionary of Current English*, 7th ed. Oxford University Press, 2005.

[46] I. Altman, *The Environment and Social Behavior: Privacy, Personal Space, Territory, Crowding*. Brooks/Cole, 1975.

[47] A. F. Westin, *Privacy and Freedom*. Atheneum, 1970.

[48] C. Bettini and D. Riboni, "Privacy Protection in Pervasive Systems: State of the Art and Technical Challenges," *Pervasive and Mobile Computing*, vol. 17, pp. 159–174, 2015.

[49] D. J. Solove, "A Taxonomy of Privacy," *University of Pennsylvania Law Review*, vol. 154, no. 3, pp. 477–560, 2006.

[50] S. Spiekermann and L. F. Cranor, "Engineering Privacy," *IEEE Transactions on Software Engineering*, vol. 35, no. 1, pp. 67–82, 2009.

[51] J. Heurix, P. Zimmermann, T. Neubauer, and S. Fenz, "A Taxonomy for Privacy Enhancing Technologies," *Computers & Security*, vol. 53, pp. 1–17, 2015.

[52] M. Raya and J.-P. Hubaux, "Securing Vehicular Ad Hoc Networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.

[53] P. Gandotra, R. K. Jha, and S. Jain, "A survey on device-to-device (d2d) communication: Architecture and security issues," *Journal of Network and Computer Applications*, vol. 78, pp. 9 – 29, 2017.

[54] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, "A Classification of Location Privacy Attacks and Approaches," *Personal and Ubiquitous Computing*, vol. 18, no. 1, pp. 163–175, 2014.

[55] J. R. Shih, Y. Hu, M. C. Hsiao, M. S. Chen, W. C. Shen, B. Y. Yang, A. Y. Wu, and C. M. Cheng, "Securing M2M With Post-Quantum Public-Key Cryptography," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 106–116, 2013.

[56] L. Y. Yeh, Y. L. Huang, A. D. Joseph, S. W. Shieh, and W. J. Tsaur, "A Batch-Authenticated and Key Agreement Framework for P2P-Based Online Social Networks," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1907–1924, 2012.

[57] H. Yang and V. A. Oleshchuk, "An Improvement of the Batch-Authentication and Key Agreement Framework for P2P-based Online Social Networks," in *Proceedings of the International Conference on Privacy and Security in Mobile Systems (PRISMS)*, 2014, pp. 1–4.

[58] A. Sudarsono and T. Nakanishi, "An Implementation of Secure Data Exchange in Wireless Delay Tolerant Network Using Attribute-Based Encryption," in *Proceedings of the Second International Symposium on Computing and Networking*, 2014, pp. 536–542.

[59] J. Hur and K. Kang, "Secure Data Retrieval for Decentralized Disruption-Tolerant Military Networks," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 16–26, 2014.

[60] P. Jaiswal, A. Kumar, and S. Tripathi, "Design of Secure Group Key Agreement Protocol using Elliptic Curve Cryptography," in *Proceedings of the International Conference on High Performance Computing and Applications (ICHPCA)*, 2014, pp. 1–6.

[61] S. Sharma and C. R. Krishna, "An Efficient Distributed Group Key Management Using Hierarchical Approach with Elliptic Curve Cryptography," in *Proceedings of the IEEE International Conference on Computational Intelligence Communication Technology (CICT)*, 2015, pp. 687–693.

[62] Y. Zhang, J. Chen, H. Li, W. Zhang, J. Cao, and C. Lai, "Dynamic Group Based Authentication Protocol for Machine Type Communications," in *Proceedings of the 4th International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, 2012, pp. 334–341.

[63] C. Lai, H. Li, R. Lu, R. Jiang, and X. Shen, "LGTH: A Lightweight Group Authentication Protocol for Machine-Type Communication in LTE Networks," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2013, pp. 832–837.

[64] J. Li, M. Wen, and T. Zhang, "Group-Based Authentication and Key Agreement With Dynamic Policy Updating for MTC in LTE-A Networks," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 408–417, 2016.

[65] J. Cao, M. Ma, and H. Li, "A Group-based Authentication and Key Agreement for MTC in LTE Networks," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 1017–1022.

[66] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[67] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp. 2546–2559, 2016.

[68] Y. Jung, E. Festijo, and M. Peradilla, "Joint Operation of Routing Control and Group Key Management for 5G Ad Hoc D2D Networks," in *Proceedings of the International Conference on Privacy and Security in Mobile Systems (PRISMS)*, 2014, pp. 1–8.

[69] W. Shen, W. Hong, X. Cao, B. Yin, D. M. Shila, and Y. Cheng, "Secure Key Establishment for Device-to-Device Communications," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2014, pp. 336–340.

[70] R. H. Hsu and J. Lee, "Group Anonymous D2D Communication with End-to-End Security in LTE-A," in *Proceedings of the IEEE Conference on Communications and Network Security (CNS)*, 2015, pp. 451–459.

[71] T. Markmann, T. C. Schmidt, and M. Wählisch, "Federated End-to-End Authentication for the Constrained Internet of Things Using IBC and ECC," in *Proceedings of the ACM Conference on Special Interest Group on Data Communication (SIGCOMM)*, 2015, pp. 603–604.

[72] A. Zhang, J. Chen, R. Q. Hu, and Y. Qian, "SeDS: Secure Data Sharing Strategy for D2D Communication in LTE-Advanced Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2659–2672, 2016.

[73] L. Goratti, G. Steri, K. M. Gomez, and G. Baldini, "Connectivity and Security in a D2D Communication Protocol for Public Safety Applications," in *Proceedings of the 11th International Symposium on Wireless Communications Systems (ISWCS)*, 2014, pp. 548–552.

[74] K. Zeng, "Physical Layer Key Generation in Wireless Networks: Challenges and Opportunities," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 33–39, 2015.

[75] C. D. T. Thai, J. Lee, and T. Q. Quek, "Physical-Layer Secret Key Generation with Colluding Untrusted Relays," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1517–1530, 2016.

[76] G. Chen, Y. Gong, P. Xiao, and J. A. Chambers, "Physical Layer Network Security in the Full-Duplex Relay System," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 574–583, 2015.

[77] W. Xi, X.-Y. Li, C. Qian, J. Han, S. Tang, J. Zhao, and K. Zhao, "KEEP: Fast Secret Key Extraction Protocol for D2D Communication," in *Proceedings of the IEEE 22nd International Symposium of Quality of Service (IWQoS)*, 2014, pp. 350–359.

[78] K. Chen, B. B. Natarajan, and S. Shattil, "Secret Key Generation Rate with Power Allocation in Relay-Based LTE-A Networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2424–2434, 2015.

[79] J. Sun, X. Chen, J. Zhang, Y. Zhang, and J. Zhang, "SYNERGY: A Game-Theoretical Approach for Cooperative Key Generation in Wireless Networks," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2014, pp. 997–1005.

[80] C. Tata and M. Kadoch, "Secure Multipath Routing Algorithm for Device-to-Device Communications for Public Safety over LTE Heterogeneous Networks," in *Proceedings of the 3rd International Conference on Future Internet of Things and Cloud (FiCloud)*, 2015, pp. 212–217.

[81] L. Guo, C. Zhang, H. Yue, and Y. Fang, "PSaD: A Privacy-Preserving Social-Assisted Content Dissemination Scheme in DTNs," *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, pp. 2903–2918, 2014.

[82] A. J. Feldman, A. Blankstein, M. J. Freedman, and E. W. Felten, "Social Networking with Frientegrity: Privacy and Integrity with an Untrusted Provider," in *Proceedings of the 21st USENIX Security Symposium (USENIX Security)*, 2012, pp. 647–662.

[83] Y. Liu, L. Wang, S. A. R. Zaidi, M. Elkashlan, and T. Q. Duong, "Secure D2D Communication in Large-Scale Cognitive Cellular Networks with Wireless Power Transfer," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2015, pp. 4309–4314.

[84] ——, "Secure D2D Communication in Large-Scale Cognitive Cellular Networks: A Wireless Power Transfer Model," *IEEE Transactions on Communications*, vol. 64, no. 1, pp. 329–342, 2016.

[85] C. Ma, J. Liu, X. Tian, H. Yu, Y. Cui, and X. Wang, "Interference Exploitation in D2D-Enabled Cellular Networks: A Secrecy Perspective," *IEEE Transactions on Communications*, vol. 63, no. 1, pp. 229–242, 2015.

[86] E. Abd-Elrahman, H. Ibn-khedher, H. Afifi, and T. Toukabri, "Fast Group Discovery and Non-Repudiation in D2D Communications using IBE," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2015, pp. 616–621.

[87] H. Zhang, T. Wang, L. Song, and Z. Han, "Radio Resource Allocation for Physical-Layer Security in D2D Underlay Communications," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2014, pp. 2319–2324.

[88] Y. Luo, L. Cui, Y. Yang, and B. Gao, "Power Control and Channel Access for Physical-Layer Security of D2D Underlay Communication," in *Proceedings of the International Conference on Wireless Communications Signal Processing (WCSP)*, 2015, pp. 1–5.

[89] H.-M. Wang, T.-X. Zheng, J. Yuan, D. Towsley, and M. H. Lee, "Physical Layer Security in Heterogeneous Cellular Networks," *IEEE Transactions on Communications*, vol. 64, no. 3, pp. 1204–1219, 2016.

[90] Z. Chu, K. Cumanan, M. Xu, and Z. Ding, "Robust Secrecy Rate Optimisations for Multiuser Multiple-Input-Single-Output Channel with Device-to-Device Communications," *IET Communications*, vol. 9, no. 3, pp. 396–403, 2015.

[91] L. Sun, Q. Du, P. Ren, and Y. Wang, "Two Birds With One Stone: Towards Secure and Interference-Free D2D Transmissions via Constellation Rotation," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8767–8774, 2016.

[92] E. Panaousis, T. Alpcan, H. Fereidooni, and M. Conti, "Secure Message Delivery Games for Device-to-Device Communications," in *Decision and Game Theory for Security*. Springer, 2014, pp. 195–215.

[93] D. V. S. Babu and P. C. Reddy, "Secure Policy Agreement for Privacy Routing in Wireless Communication System," in *Proceedings of the International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, 2014, pp. 739–744.

[94] E. S. Babu, C. Nagaraju, and M. K. Prasad, "A Secure Routing Protocol Against Heterogeneous Attacks in Wireless Adhoc Networks," in *Proceedings of the Sixth International Conference on Computer and Communication Technology (ICCCT)*, 2015, pp. 339–344.

[95] M. D. Green and I. Miers, "Forward Secure Asynchronous Messaging from Puncturable Encryption," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, 2015, pp. 305–320.

[96] I. R. Chen, F. Bao, M. Chang, and J. H. Cho, "Dynamic Trust Management for Delay Tolerant Networks and Its Application to Secure Routing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1200–1210, 2014.

[97] X. Liang, J. Qin, M. Wang, D. Wang, and J. Wan, "An Effective and Secure Epidemic Routing for Disruption-Tolerant Networks," in *Pro-*

ceedings of the Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2014, pp. 329–333.

[98] V. Priya and B. Sakthisaravanan, "Information Centric Network for Secure Data Transmission in DTN," in *Proceedings of the International Conference on Innovation Information in Computing Technologies (ICIICT)*, 2015, pp. 1–4.

[99] A. K. Gupta, I. Bhattacharya, P. S. Banerjee, and J. K. Mandal, "A Co-operative Approach to Thwart Selfish and Black-Hole Attacks in DTN for Post Disaster Scenario," in *Proceedings of the Fourth International Conference of Emerging Applications of Information Technology (EAIT)*, 2014, pp. 113–118.

[100] F. Garay, E. Rosas, and N. Hidalgo, "Reliable Routing Protocol for Delay Tolerant Networks," in *Proceedings of the IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS)*, 2015, pp. 320–327.

[101] A. Mocktoolah and K. K. Khedo, "Privacy Challenges in Proximity based Social Networking: Techniques & Solutions," in *Proceedings of the International Conference on Computing, Communication and Security (ICCCS)*, 2015, pp. 1–8.

[102] K. Zickuhr, "Location-Based Services," 2013. [Online]. Available: http://www.pewinternet.org/2013/09/12/location-based-services/ (visited on 10.05.2016).

[103] A. Behrooz and A. Devlic, "A Context-Aware Privacy Policy Language for Controlling Access to Context Information of Mobile Users," in *Proceedings of the Third International ICST Conference (MOBISEC)*, 2011, pp. 25–39.

[104] S. Chakraborty, Z. Charbiwala, H. Choi, K. R. Raghavan, and M. B. Srivastava, "Balancing Behavioral Privacy and Information Utility in Sensory Data Flows," *Pervasive and Mobile Computing*, vol. 8, no. 3, pp. 331–345, 2012.

[105] J.-H. Cho, K. Chan, and S. Adali, "A Survey on Trust Modeling," *ACM Computing Surveys*, vol. 48, no. 2, pp. 1–40, 2015.

[106] Y.-A. de Montjoye, E. Shmueli, S. S. Wang, A. S. Pentland, and T. Preis, "openPDS: Protecting the Privacy of Metadata through SafeAnswers," *PLoS ONE*, vol. 9, no. 7, 2014.

[107] H. Haddadi, H. Howard, A. Chaudhry, J. Crowcroft, A. Madhavapeddy, and R. Mortier, "Personal Data: Thinking Inside the Box," 2015. [Online]. Available: https://arxiv.org/abs/1501.04737 (visited on 29.10.2016).

[108] R. Wishart, K. Henricksen, and J. Indulska, "Context Privacy and Obfuscation Supported by Dynamic Context Source Discovery and Processing in a Context Management System," in *Proceedings of the 4th International Conference on Ubiquitous Intelligence and Computing (UIC)*, 2007, pp. 929–940.

[109] E. Franz, T. Springer, and N. Harder, "Enhancing Privacy in Social Applications with the Notion of Group Context," in *Proceedings of the International Conference for Internet Technology and Secured Transactions (ICITST)*, 2012, pp. 112–118.

[110] "PrimeLife." [Online]. Available: http://primelife.ercim.eu (visited on 17.05.2016).

[111] D. Christin, C. Roßkopf, M. Hollick, L. A. Martucci, and S. S. Kanhere, "IncogniSense: An Anonymity-Preserving Reputation Framework for Participatory Sensing Applications," *Pervasive and Mobile Computing*, vol. 9, no. 3, pp. 353–371, 2013.

[112] I. Boutsis and V. Kalogeraki, "Privacy Preservation for Participatory Sensing Data," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2013, pp. 103–113.

[113] A. Pfitzmann and M. Hansen, "Terminology for Talking about Privacy by Data Minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management," 2010. [Online]. Available: http://dud.inf.tu-dresden.de/Anon_Terminology.shtml (visited on 12.04.2016).

[114] J. Petit, F. Schaub, M. Feiri, and F. Kargl, "Pseudonym Schemes in Vehicular Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 228–255, 2015.

[115] D. Eckhoff, R. German, C. Sommer, F. Dressler, and T. Gansen, "SlotSwap: Strong and Affordable Location Privacy in Intelligent Transportation Systems," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 126–133, 2011.

[116] Rongxing Lu, Xiaodong Li, T. H. Luan, Xiaohui Liang, and Xuemin Shen, "Pseudonym Changing at Social Spots: An Effective Strategy for Location Privacy in VANETs," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 86–96, 2012.

[117] A. R. Beresford and F. Stajano, "Mix Zones: User Privacy in Location-Aware Services," in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW)*, 2004, pp. 127–131.

[118] Y. Pan and J. Li, "Cooperative Pseudonym Change Scheme Based on the Number of Neighbors in VANETs," *Journal of Network and Computer Applications*, vol. 36, no. 6, pp. 1599–1609, 2013.

[119] M. Raya and J.-P. Hubaux, "The Security of Vehicular Ad Hoc Networks," in *Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, 2005, pp. 11–21.

[120] M. Nagy, N. Asokan, and J. Ott, "PeerShare: A System Secure Distribution of Sensitive Data among Social Contacts," in *Proceedings of the 18th Nordic Conference on Secure IT Systems (NordSec)*, 2013, pp. 154–165.

[121] A. Kate, G. M. Zaverucha, and U. Hengartner, "Anonymity and Security in Delay Tolerant Networks," in *Proceedings of the 3rd International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm)*, 2007, pp. 504–513.

[122] C.-T. Huang, L. Huang, Z. Qin, H. Yuan, L. Zhou, V. Varadharajan, and C.-C. J. Kuo, "Survey on Securing Data Storage in the Cloud," *APSIPA Transactions on Signal and Information Processing*, vol. 3, pp. 1–17, 2014.

[123] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," in *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques (EUROCRYPT)*, 1999, pp. 223–238.

[124] T. Elgamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[125] B. Mu and S. Bakiras, "Private Proximity Detection for Convex Polygons," in *Proceedings of the 12th International ACM Workshop on Data Engineering for Wireless and Mobile Access (MobiDE)*, 2013, pp. 36–43.

[126] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private Queries in Location Based Services: Anonymizers are not Necessary," in *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, 2008, pp. 121–132.

[127] C. Bösch, P. Hartel, W. Jonker, and A. Peter, "A Survey of Provably Secure Searchable Encryption," *ACM Computing Surveys*, vol. 47, no. 2, pp. 1–51, 2015.

[128] D. Chaum, "Blind Signatures for Untraceable Payments," in *Proceedings of the 2nd International Cryptology Conference (CRYPTO)*, 1982, pp. 199–203.

[129] R. L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," in *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT)*, 2001, pp. 552–565.

[130] D. Chaum and E. van Heyst, "Group Signatures," in *Proceedings of the 10th Annual International Conference on Theory and Application of Cryptographic Techniques (EUROCRYPT)*, 1991, pp. 257–265.

[131] J. Y. Koh, J. C. M. Teo, D. Leong, and W.-C. Wong, "Reliable Privacy-Preserving Communications for Wireless Ad Hoc Networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2015, pp. 6271–6276.

[132] Ying Jian, Shigang Chen, Zhan Zhang, and Liang Zhang, "A Novel Scheme for Protecting Receiver's Location Privacy in Wreless Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 10, pp. 3769–3779, 2008.

[133] K. Mehta, Donggang Liu, and M. Wright, "Protecting Location Privacy in Sensor Networks against a Global Eavesdropper," *IEEE Transactions on Mobile Computing*, vol. 11, no. 2, pp. 320–336, 2012.

[134] Y. Fan, Y. Jiang, H. Zhu, J. Chen, and X. S. Shen, "Network Coding Based Privacy Preservation against Traffic Analysis in Multi-Hop Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 10, no. 3, pp. 834–843, 2011.

[135] P. Zhang, C. Lin, Y. Jiang, P. P. Lee, and J. C. Lui, "ANOC: Anonymous Network-Coding-Based Communication with Efficient Cooperation," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 9, pp. 1738–1745, 2012.

[136] D. L. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[137] M. La Polla, F. Martinelli, and D. Sgandurra, "A Survey on Security for Mobile Devices," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 446–471, 2013.

[138] N. Asokan, J.-E. Ekberg, K. Kostiainen, A. Rajan, C. Rozas, A.-R. Sadeghi, S. Schulz, and C. Wachsmann, "Mobile Trusted Computing," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1189–1206, 2014.

[139] M. S. Kirkpatrick, G. Ghinita, and E. Bertino, "Resilient Authenticated Execution of Critical Applications in Untrusted Environments," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 4, pp. 597–609, 2012.

[140] Sufatrio, D. J. J. Tan, T.-W. Chua, and V. L. L. Thing, "Securing Android: A Survey, Taxonomy, and Challenges," *ACM Computing Surveys*, vol. 47, no. 4, pp. 1–45, 2015.

[141] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau, and P. McDaniel, "FlowDroid: Precise Context, Flow, Field, Object-Sensitive and Lifecycle-Aware Taint Analysis for Android Apps," in *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, 2014, pp. 259–269.

[142] Z. Yang, M. Yang, Y. Zhang, G. Gu, P. Ning, and X. S. Wang, "AppIntent: Analyzing Sensitive Data Transmission in Android for Privacy Leakage Detection," in *Proceedings of the ACM SIGSAC Conference on Computer & Communications Security (CCS)*, 2013, pp. 1043–1054.

[143] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones," in *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation (OSDI)*, 2010, pp. 393–407.

[144] W. Enck, P. Gilbert, S. Han, V. Tendulkar, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones," *ACM Transactions on Computer Systems*, vol. 32, no. 2, pp. 1–29, 2014.

[145] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall, "These Aren't the Droids You're Looking For: Retrofitting Android to Protect Data from Imperious Applications," in *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS)*, 2011, pp. 639–652.

[146] A. Razaghpanah, N. Vallina-Rodriguez, S. Sundaresan, C. Kreibich, P. Gill, M. Allman, and V. Paxson, "Haystack: In Situ Mobile Traffic Analysis in User Space," 2015. [Online]. Available: https://arxiv.org/abs/1510.01419v1 (visited on 21.04.2016).

[147] M. Haris, H. Haddadi, and P. Hui, "Privacy Leakage in Mobile Computing: Tools, Methods, and Characteristics," 2014. [Online]. Available: http://arxiv.org/abs/1410.4978 (visited on 21.04.2016).

[148] H. Liang, D. Wu, J. Xu, and H. Ma, "Survey on Privacy Protection of Android Devices," in *Proceedings of the IEEE 2nd International Conference on Cyber Security and Cloud Computing (CSCloud)*, 2015, pp. 241–246.

[149] F. Shen, N. Vishnubhotla, C. Todarka, M. Arora, B. Dhandapani, E. J. Lehner, S. Y. Ko, and L. Ziarek, "Information Flows as a Permission Mechanism," in *Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering (ASE)*, 2014, pp. 515–526.

[150] C. Wu, Y. Zhou, K. Patel, Z. Liang, and X. Jiang, "AirBag: Boosting Smartphone Resistance to Malware Infection," in *Proceedings of the 21th Annual Network and Distributed System Security Symposium (NDSS)*, 2014, pp. 1–13.

[151] A. B. Brush, J. Krumm, and J. Scott, "Exploring End User Preferences for Location Obfuscation, Location-Based Services, and the Value of Location," in *Proceedings of the 12th ACM International Conference on Ubiquitous Computing (UbiComp)*, 2010, pp. 95–104.

[152] T. Burghardt, E. Buchmann, J. Müller, and K. Böhm, "Understanding User Preferences and Awareness: Privacy Mechanisms in Location-Based Services," in *Proceedings of the Confederated International Conferences, CoopIS, DOA, IS, and ODBASE (OTM)*, 2009, pp. 304–321.

[153] A. P. Felt, S. Egelman, and D. Wagner, "I've Got 99 Problems, But Vibration Ain't One: A Survey of Smartphone Users' Concerns," in *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM)*, 2012, pp. 33–44.

[154] B. P. Knijnenburg, A. Kobsa, and H. Jin, "Preference-based Location Sharing: Are More Privacy Options Really Better?" in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2013, pp. 2667–2676.

[155] J.-H. Hoepman, "Privacy Design Strategies," in *Proceedings of the 29th IFIP International Conference on ICT Systems Security and Privacy Protection (SEC)*, 2014, pp. 446–459.

[156] G. Ghinita, *Privacy for Location-Based Services*. Morgan & Claypool, 2013, vol. 4.

[157] J. Krumm, "A Survey of Computational Location Privacy," *Personal and Ubiquitous Computing*, vol. 13, no. 6, pp. 391–399, 2009.

[158] M. Gruteser and D. Grunwald, "Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking," in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2003, pp. 31–42.

[159] F. Durr, P. Skvortsov, and K. Rothermel, "Position Sharing for Location Privacy in Non-Trusted Systems," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2011, pp. 189–196.

[160] P. Shankar, V. Ganapathy, and L. Iftode, "Privately Querying Location-Based Services with SybilQuery," in *Proceedings of the 11th International Conference on Ubiquitous Computing (UbiComp)*, 2009, pp. 31–40.

[161] S. Amini, J. Lindqvist, J. Hong, J. Lin, E. Toch, and N. Sadeh, "Caché: Caching Location-Enhanced Content to Improve User Privacy," in *Proceedings of the 9th International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2011, pp. 197–210.

[162] A. Gutscher, "Coordinate Transformation - A Solution for the Privacy Problem of Location Based Services?" in *Proceedings of the 20th IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, 2006, pp. 354–360.

[163] M. L. Yiu, C. S. Jensen, J. Møller, and H. Lu, "Design and Analysis of a Ranking Approach to Private Location-Based Services," *ACM Transactions on Database Systems*, vol. 36, no. 2, pp. 1–42, 2011.

[164] M. Terrovitis and N. Mamoulis, "Privacy Preservation in the Publication of Trajectories," in *Proceedings of the 9th International Conference on Mobile Data Management (MDM)*, 2008, pp. 65–72.

[165] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Preserving Privacy in GPS Traces via Uncertainty-Aware Path Cloaking," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS)*, 2007, pp. 161–171.

[166] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson, "Virtual Trip Lines for Distributed Privacy-Preserving Traffic Monitoring," in *Proceedings of the 6th International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2008, pp. 15–28.

[167] C. A. Ardagna, M. Cremonini, and G. Gianini, "Landscape-Aware Location-Privacy Protection in Location-Based Services," *Journal of Systems Architecture*, vol. 55, no. 4, pp. 243–254, 2009.

[168] L. Šikšnys, J. R. Thomsen, S. Šaltenis, and M. L. Yiu, "Private and Flexible Proximity Detection in Mobile Social Networks," in *Proceedings of the 11th International Conference on Mobile Data Management (MDM)*, 2010, pp. 75–84.

[169] S. Mascetti, C. Bettini, D. Freni, X. S. Wang, and S. Jajodia, "Privacy-Aware Proximity Based Services," in *Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware (MDM)*, 2009, pp. 31–40.

[170] D. Freni, "Privacy-Preserving Techniques for Proximity Based LBS," in *Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware (MDM)*, 2009, pp. 387–388.

[171] K. G. Shin, Xiaoen Ju, Zhigang Chen, and Xin Hu, "Privacy Protection for Users of Location-based Services," *IEEE Wireless Communications*, vol. 19, no. 1, pp. 30–39, 2012.

[172] A. Solanas, J. Domingo-Ferrer, and A. Martínez-Ballesté, "Location Privacy in Location-Based Services: Beyond TTP-based Schemes," in *Proceedings of the 1st International Workshop on Privacy in Location-Based Applications (PiLBA)*, 2008, pp. 12–23.

[173] X. Jin, J. Sun, R. Zhang, and Y. Zhang, "SafeDSA: Safeguard Dynamic Spectrum Access Against Fake Secondary Users," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2015, pp. 304–315.

[174] E. Chung, J. Joy, and M. Gerla, "DiscoverFriends: Secure Social Network Communication in Mobile Ad Hoc Networks," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2015, pp. 7–12.

[175] S. A. M. Ghanem and M. Ara, "Secure Communications with D2D Cooperation," in *Proceedings of the International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, 2015, pp. 1–6.

[176] J.-P. Hubaux, L. Buttyán, and S. Capkun, "The Quest for Security in Mobile Ad Hoc Networks," in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*, 2001, pp. 146–155.

[177] J. H. Cho, A. Swami, and I. R. Chen, "A Survey on Trust Management for Mobile Ad Hoc Networks," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 562–583, 2011.

[178] H. Chen, Y. Xiao, X. Hong, F. Hu, and J. Xie, "A Survey of Anonymity in Wireless Communication Systems," *Security and Communication Networks*, vol. 2, no. 5, pp. 427–444, 2009.

[179] R. Bista and J.-W. Chang, "Privacy-Preserving Data Aggregation Protocols for Wireless Sensor Networks: A Survey," *Sensors*, vol. 10, no. 5, pp. 4577–4601, 2010.

[180] L. Nobach and D. Hausheer, "Towards Decentralized, Energy- and Privacy-Aware Device-to-Device Content Delivery," in *Proceedings of the 8th IFIP International Conference on Autonomous Infrastructure, Management, and Security (AIMS)*, 2014, pp. 128–132.

[181] J. Flinn and M. Satyanarayanan, "Energy-Aware Adaptation for Mobile Applications," in *Proceedings of the Seventeenth ACM Symposium on Operating Systems Principles (SOSP)*, 1999, pp. 48–63.

[182] N. Vallina-Rodriguez and J. Crowcroft, "ErdOS: Achieving Energy Savings in Mobile OS," in *Proceedings of the Sixth International Workshop on MobiArch*, 2011, pp. 37–42.

[183] G. Ananthanarayanan and I. Stoica, "Blue-Fi: Enhancing Wi-Fi Performance Using Bluetooth Signals," in *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2009, pp. 249–262.

[184] N. Vallina-Rodriguez and J. Crowcroft, "Energy Management Techniques in Modern Mobile Handsets," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 179–198, 2013.

[185] S. Tarkoma, M. Siekkinen, E. Lagerspetz, and Y. Xiao, *Smartphone Energy Consumption: Modelling and Optimization*, 1st ed. Cambridge University Press, 8 2014.

[186] W. Sun, Z. Yang, X. Zhang, and Y. Liu, "Energy-Efficient Neighbor Discovery in Mobile Ad Hoc and Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1448–1459, 2014.

[187] B. Han, J. Li, and A. Srinivasan, "On the Energy Efficiency of Device Discovery in Mobile Opportunistic Networks: A Systematic Approach," *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 786–799, 2015.

[188] S. Nath, "ACE: Exploiting Correlation for Energy-Efficient and Continuous Context Sensing," *IEEE Transactions on Mobile Computing*, vol. 12, no. 8, pp. 1472–1486, 2013.

[189] J. M. Urban, C. J. Hoofnagle, and S. Li, "Mobile Phones and Privacy," *BCLT Research Paper Series*, no. 2103405, 10.07.2012.

[190] Microsoft, "Location Based Services Usage and Perceptions Survey Presentation," 2011. [Online]. Available: https://www.microsoft.com/en-us/download/details.aspx?id=3250 (visited on 12.04.2016).

[191] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the Crowd: The Privacy Bounds of Human Mobility," *Scientific reports*, vol. 3, March 2013.

[192] 3GPP, "Proximity-based Services (ProSe)," 2015. [Online]. Available: http://www.3gpp.org/DynaReport/23303.htm (visited on 18.04.2016).

[193] ——. (2015) Proximity-based Services (ProSe); Security Aspects. [Online]. Available: http://www.3gpp.org/DynaReport/33303.htm (visited on 31.05.2016).

[194] ——. (2015) Group Communication System Enablers for LTE. [Online]. Available: http://www.3gpp.org/DynaReport/23468.htm (visited on 31.05.2016).

[195] I. Hafeez, A. Y. Ding, L. Suomalainen, A. Kirichenko, and S. Tarkoma, "Securebox: Toward Safer and Smarter IoT Networks," in *Proceedings of the 1st ACM CoNEXT Workshop on Cloud-Assisted Networking (CAN)*, 2016.

[196] J. Zhou, Z. Cao, X. Dong, and X. Lin, "Security and Privacy in Cloud-Assisted Wireless Wearable Communications: Challenges, Solutions, and Future Directions," *IEEE Wireless Communications*, vol. 22, no. 2, pp. 136–144, 2015.

[197] A. Y. Ding, J. Korhonen, T. Savolainen, M. Kojo, J. Ott, S. Tarkoma, and J. Crowcroft, "Bridging the Gap Between Internet Standardization and Networking Research," *SIGCOMM Computer Communication Review*, vol. 44, no. 1, pp. 56–62, 2013.

[198] M. Conti, J. Willemsen, and B. Crispo, "Providing Source Location Privacy in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1238–1280, 2013.

[199] V. P. Illiano and E. C. Lupu, "Detecting Malicious Data Injections in Wireless Sensor Networks," *ACM Computing Surveys*, vol. 48, no. 2, pp. 1–33, 2015.

[200] L. Buttyán and J.-P. Hubaux, *Security and Cooperation in Wireless Networks: Thwarting Malicious and Selfish Behavior in the Age of Ubiquitous Computing*. Cambridge University Press, 2008.

[201] H. A. U. Mustafa, M. A. Imran, M. Z. Shakir, A. Imran, and R. Tafazolli, "Separation Framework: An Enabler for Cooperative and D2D Communication for Future 5G Networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 419–445, 2016.

[202] S. Du, H. Zhu, X. Li, K. Ota, and M. Dong, "MixZone in Motion: Achieving Dynamically Cooperative Location Privacy Protection in Delay-Tolerant Networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4565–4575, 2013.

[203] E. Hossain, D. I. Kim, and V. K. Bhargava, *Cooperative Cellular Wireless Networks*. Cambridge University Press, 2011.

[204] H. Chen and W. Lou, "Making Nodes Cooperative: A Secure Incentive Mechanism for Message Forwarding in DTNs," in *Proceedings of the 22nd International Conference on Computer Communications and Networks (ICCCN)*, 2013, pp. 1–7.

[205] Y. Zou, X. Wang, and W. Shen, "Optimal Relay Selection for Physical-Layer Security in Cooperative Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 10, pp. 2099–2111, 2013.

[206] A. A. de Freitas and A. K. Dey, "Using Multiple Contexts to Detect and Form Opportunistic Groups," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW)*, 2015, pp. 1612–1621.

[207] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[208] C. Dwork, "Differential Privacy," in *Proceedings of the 33rd International Conference on Automata, Languages and Programming (ICALP)*, 2006, pp. 1–12.

[209] R. Shokri, "Quantifying and Protecting Location Privacy," *it - Information Technology*, vol. 57, no. 4, pp. 257–263, 2015.

[210] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying Location Privacy," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, 2011, pp. 247–262.

[211] G. Fodor, S. Parkvall, S. Sorrentino, P. Wallentin, Q. Lu, and N. Brahmi, "Device-to-Device Communications for National Security and Public Safety," *IEEE Access*, vol. 2, pp. 1510–1520, 2014.

[212] A. Kumbhar, F. Koohifar, I. Guvenc, and B. Mueller, "A Survey on Legacy and Emerging Technologies for Public Safety Communications," *IEEE Communications Surveys & Tutorials*, pp. 1–29, September 2016.

[213] B. Budington, "WhatsApp Rolls Out End-To-End Encryption to its Over One Billion Users," 07.04.2016. [Online]. Available: https://www.eff.org/de/node/91131 (visited on 17.04.2016).

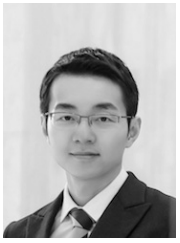
[214] WhatsApp, "WhatsApp Encryption Overview: Technical White Paper," 04.04.2016. [Online]. Available: https://www.whatsapp.com/security/ (visited on 06.04.2016).

[215] M. DeBonis, "Obama at SXSW: 'Absolutist View' on Digital Privacy cannot prevail," 11.03.2016. [Online]. Available: https://www.washingtonpost.com/news/post-politics/wp/2016/03/11/obama-at-sxsw-absolutist-view-on-digital-privacy-cannot-prevail/ (visited on 17.04.2016).

[216] B. Smith, "Keeping Secrecy the Exception, not the Rule: An Issue for both Consumers and Businesses," 14.04.2016. [Online]. Available: https://blogs.microsoft.com/on-the-issues/2016/04/14/keeping-secrecy-exception-not-rule-issue-consumers-businesses/ (visited on 17.04.2016).

Michael Haus is currently a Ph.D. student at Technical University of Munich. His research focus is on privacy for resource-constrained devices and the design of context-aware mobile systems, especially proximity-based applications. He obtained his B.Sc. from the Department of Computer Science, Munich University of Applied Sciences in 2012 and his M.Sc. degree in Robotics, Cognition and Intelligence from the Technical University of Munich in 2014.

Muhammad Waqas received his B.Sc. and M.Sc. degrees from the Department of Electrical Engineering, University of Engineering and Technology Peshawar, Pakistan in 2009 and 2014, respectively. He is currently working towards the Ph.D. degree in FIB LAB at the Department of Electronic Engineering, Tsinghua University, Beijing China. His current research interests are in the areas of networking and communications, including cooperative communication, security, resource allocation, device-to-device communication, and social networks.



Aaron Yi Ding is a postdoc associate and project leader at Technical University of Munich (TUM). His research interests include mobile edge computing, IoT security, and system networking. He obtained his M.Sc. and Ph.D. both with distinction from University of Helsinki. He was a visiting scholar at Columbia University in 2014 and at University of Cambridge in 2013 advised by Prof. Henning Schulzrinne and Prof. Jon Crowcroft, respectively. He has been awarded the ACM SIGCOMM Best of CCR and Nokia Foundation Scholarships.

Yong Li (M'09-SM'16) received the B.S. degree in electronics and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 2007 and the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2012. He is currently a Faculty Member of the Department of Electronic Engineering, Tsinghua University. Dr. Li has served as General Chair, TPC Chair, TPC Member for several international workshops and conferences, and he is on the editorial board of three international journals. His papers have total citations of more than 2000 (five papers exceed 100 citations, Google Scholar). Among them, eight are ESI Highly Cited Papers in Computer Science, and four received conference Best Paper (run-up) Awards. He received IEEE 2016 ComSoc Asia-Pacific Outstanding Young Researchers.

Sasu Tarkoma (M'06-SM'12) is a Professor of Computer Science at the University of Helsinki and Head of the Department of Computer Science. He is also affiliated with the Helsinki Institute for Information Technology HIIT. His research interests are Internet technology, distributed systems, data analytics, and mobile and ubiquitous computing.He has authored four textbooks and has published over 160 scientific articles. His research has received several Best Paper awards and mentions, for example at IEEE PerCom, ACM CCR, and ACM OSR. He has seven granted US Patents. He is a member of the editorial board of the Computer Networks Journal and member of organizing and scientific committees of many international conferences.

Jörg Ott holds the Chair for Connected Mobility at Technical University of Munich in the Faculty of Informatics since August 2015. He is also an adjunct professor at Aalto University, where he was a professor of networking technology with a focus on protocols, services, and software from 2005 until 2015. He is interested in understanding, designing, and building Internet-based (mobile) communication systems and services. His research focus is on network and system architectures, protocols, and applications for mobile systems. His research interests further comprise measuring, modeling, analyzing, and predicting network characteristics and application performance as well as preserving user privacy. Present applications range from scalable services for urban areas to localized networked services independent of cloud and Internet providers to extending the reach of the Internet to remote areas.

Publication 2

M. Haus, A. Y. Ding, and J. Ott. LocalVLC: Augmenting Smart IoT Services with Practical Visible Light Communication. In *Proceedings of the 20th IEEE International Symposium on 'A World of Wireless, Mobile and Multimedia Networks' (WoWMoM)*, pages 1–9, 2019. doi:10.1109/wowmom.2019.8793022

Publication Summary

We can take advantage of the ubiquitous light sources around us for visible light communication (VLC). The idea is to enrich existing lighting infrastructure, e.g., light-emitting diode (LED) lamps in residential and office settings, with data communication. Most researchers improve VLC systems in terms of increased communication performance or novel functionality. As a result, existing VLC solutions suffer from high energy overhead or perceptible light flickering effects. The unpleasant light flickering limits the deployment and possible scenarios of VLC. This leads to the crucial question of how to achieve a practically deployable VLC with low power footprint? Given that common VLC modulation schemes raising light flickering effects, we proposed LocalVLC with a Morse-code inspired modulation scheme which is able to operate on off-the-shelf LEDs with low energy overhead. We transmit data as a series of light on and off periods where we use timely different light off phases to detect letters, words, and messages. The LocalVLC platform follows two principles to enable services: 1) deployable at low-cost by using off-the-shelf components, such as a basic programmable board with general-purpose input/output (GPIO) support, and 2) practical platform that imposes as little constraints as possible for indoor IoT usage, like a flexible orientation and an open-box design with application programming interfaces (APIs) for developer. We have implemented and evaluated a system prototype based on the LocalVLC design. Our evaluation includes micro-benchmarks of the LocalVLC system performance, like throughput, maximum transmission distance, field of view, and latency. Moreover, we compare our LocalVLC encoding with Manchester code in terms of throughput and energy consumption. As a result, our LocalVLC prototype supports up to 10 meters of range and providing a good throughput with low error rate and energy consumption. Compared to the Manchester encoding, LocalVLC achieves 8x improvement on both throughput and energy consumption. LocalVLC does not need specialized processing hardware, such as FPGA or micro-controller, it encodes data into high frequency light pulses and thus overcomes the light flickering effect. To demonstrate the practicality of our proposal, we developed two LocalVLC-based solutions: private service discovery for smart buildings and automated authorization for smart homes.

LocalVLC: Augmenting Smart IoT Services with Practical Visible Light Communication

Michael Haus
Technical University of Munich
haus@in.tum.de

Aaron Yi Ding
Delft University of Technology
aaron.ding@tudelft.nl

Jörg Ott
Technical University of Munich
ott@in.tum.de

***Abstract*—Visible Light Communication (VLC) emerges as a communication technology for Internet of Things (IoT) services with appealing benefits not present in existing radio-based communication. However, current VLC designs commonly require dedicated LED lights to emit modulated light beams which entail high energy overhead and unpleasant visual experiences due to the perceptible light blinking effects for end users. This greatly limits the deployment and applicable scenarios of VLC. In this paper, we design and develop LocalVLC, a practical and low-cost VLC system that can be used as a standard light source to augment smart IoT services. LocalVLC introduces a novel Morse-code inspired modulation scheme that can operate on off-the-shelf LEDs with low energy overhead. It can effectively overcome the light flickering by encoding data into high frequency light pulses without requiring extra processing hardware such as FPGA or micro-controller. We have implemented and evaluated a full-fledged system prototype based on LocalVLC design. Under practical settings, our LocalVLC prototype can support up to 10 meters of range, and attain reasonable throughput (up to 1.4 kB/s) with low error rate and energy consumption. Comparing with the widely adopted Manchester encoding, LocalVLC yields 8x improvement on both throughput and energy consumption. In addition, we demonstrate the practicality of LocalVLC through two IoT use cases where we developed two lightweight LocalVLC-based solutions using low-cost off-the-shelf hardware to exemplify the usage of LocalVLC for indoor service discovery and smart home key management.**

I. Introduction

The motivation behind visible light communication (VLC) is to reuse the ubiquitous light sources around us for data communication. For instance, the widely used Light-Emitting Diode (LED) lamps in residential and office settings can be used to add data communication as an additional functionality of the lighting infrastructure (e.g., by installing low-cost modulation unit to existing LED lights) [1], [2]. This is an appealing benefit of VLC since it introduces minimal deployment overhead in terms of hardware replacement. VLC also offers security and privacy benefits by confining its communication range within a boundary (e.g., office room) because the light signals cannot penetrate concrete walls. In addition, VLC can effectively ward off the interference with existing radio-based communications (Bluetooth and Wi-Fi), on which many IoT services depend. Besides that, VLC can serve as a feedback channel for users, e.g., as in our use case for smart homes changing the light color to indicate a successful user authorization.

However, in spite of those distinct advantages, many VLC designs focus mainly on the communication performance and novel functionality [3]–[10]. This results in a dilemma for deploying VLC in practice because existing solutions either entail high energy overhead or exhibit unpleasant visual experiences due to the perceptible light flickering effects for end users [11]. The light flickering effect greatly limits the deployment and applicable scenarios of VLC (e.g., to be used as a regular light source for indoor). In light of this challenge, the DarkLight design [11] tackles the problem sphere by emitting extremely-low luminance of light pulses, which makes the lighting device appear as a unnoticeable "dark" bulb. Although DarkLight addressed the flickering issue through an unconventional design, their solution provides a shorter communication range (1.3 m) and cannot replace existing regular light sources compared to our custom light bulb. Specifically, a dedicated DarkLight bulb should be installed besides normal LED lamps where DarkLight services are needed which causes light pollution by overlapping light signals for illumination and communication.

This leads to the fundamental question of adopting VLC: **how to achieve a practically deployable VLC with low cost and power footprint?** Given that common VLC modulation schemes can hardly keep light pulses imperceptible and hence causing light flickering effects [11], we tackle the usability challenge for VLC by proposing a holistic system solution named LocalVLC. In its core, LocalVLC introduces a Morse-code inspired modulation scheme that can operate on off-the-shelf LEDs with low energy overhead. We have implemented and evaluated a full-fledged system prototype based on LocalVLC design. In practical settings, our LocalVLC prototype can support up to 10 meters of range and attain reasonable throughput (up to 1.4 kB/s) with low error rate and energy consumption. Compared to the widely adopted Manchester encoding, LocalVLC yields 8x improvement on both throughput and energy consumption. Our design can effectively overcome the light flickering effect by encoding data into high frequency light pulses but does not require extra processing hardware such as FPGA or micro-controller. As inspired by but different from DarkLight, LocalVLC can be deployed as standard light source to overcome the light pollution problem by replacing existing lighting. The unique features of VLC can benefit many IoT scenarios as we demonstrate by two exemplary use cases: indoor service discovery and smart home key management.

In a nut shell, our work makes the following contributions:

- We design and develop LocalVLC, a ready-to-deploy system solution to address the crucial challenge faced by conventional VLC designs: usability in practical deployment. LocalVLC strikes a balance between cost and complexity to eliminate the light flickering effect, making it feasible to be embedded as a regular light source into the infrastructure.
- We introduce a dedicated modulation scheme for VLC as inspired by Morse coding. The LocalVLC modulation enables off-the-shelf hardware to operate at high frequency and sustain low power footprint. Based on the testbed evaluation, our LocalVLC prototype can support up to 10 meters of range and attain reasonable throughput with low error rate and energy consumption. Comparing with the open-source solution with widely used Manchester encoding, LocalVLC yields 8x improvement on both throughput and energy consumption.
- We further demonstrate the practicality of our proposal by developing lightweight LocalVLC-based solutions for two exemplary scenarios: proximity based service discovery and automation of key management for smart homes. Those solutions shed light on how we can harness VLC to augment future IoT services.

II. LocalVLC Platform

LocalVLC aims to provide a communication platform with several wireless communication channels to enable practical user services. We realize a custom light bulb prototype as shown in Fig. 1 to combine mid-range radio-based communication such as Wi-Fi or Bluetooth with VLC. We benefit from the naturally limited VLC communication range for use cases such as indoor service discovery and key management for smart homes presented in Section III. For our VLC transmissions we seek for a robust and yet simple encoding scheme for creating a low-rate signaling channel. On this basis, we propose Morse encoding as lightweight data encoding for VLC to achieve a broadcast with low processing overhead. Our custom light bulb allows replacing existing illumination infrastructure and avoid light pollution, at which different visible lights are overlapping for illumination and communication. We are able to overcome this problem by simultaneously using visible light for illumination and communication.

A. Hardware Platform for LocalVLC

The LocalVLC hardware platform consists of a power supply for the BeagleBone Black. During normal operation the battery is loaded and provides the power for the BeagleBone Black. The battery improves the service availability of LocalVLC and maintains the lighting in case of a power blackout. The BeagleBone Black offers an API for VLC and controls the LED transmitter and wireless modules such as Wi-Fi and Bluetooth.

Our LocalVLC platform follows two principles to enable practical services:

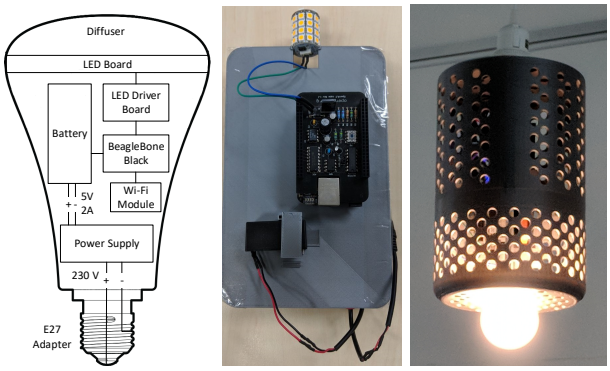


Fig. 1: LocalVLC hardware architecture; installed light bulb components; our deployed 3D-printed LocalVLC light bulb.

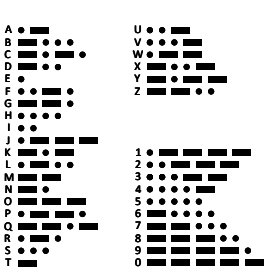
1) **Deployable at low-cost** by using off-the-shelf components. LocalVLC requires only basic programmable boards (in our case BeagleBone Black at $60) with general-purpose input/output (GPIO) support. Due to the low processing overhead, it is possible to run LocalVLC on micro-controllers. The light-emitting diodes (LED) used by LocalVLC are low-cost.
2) **Practical platform** that imposes as little constraints as possible on typical indoor IoT usage. This implies a practical working range in normal indoor situations, flexible orientation, easy portability on devices with ambient light sensors, and an open-box design with adaptive APIs for developer.

B. Morse Code Definition for VLC Signaling

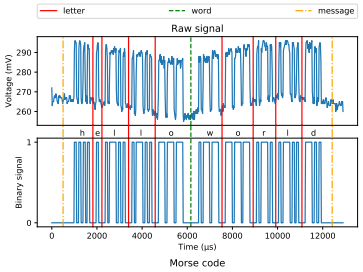
To enable robust and efficient VLC signaling, we use the Morse code defined by the International Telecommunication Union [12] for data encoding. In general, the Morse code is based on two signs, a *dot* as the smallest, time base unit, and a *dash* is about three time units. Fig. 2(a) presents the ISO basic Latin alphabet and Arabic numerals encoded in Morse code. For instance, the letter "B" consists of one dash followed by three dots. We transmit data as a series of light on and off periods, each light on phase is followed by a light off phase. To detect letters, words and messages, we use timely different light off phases. The space between parts of the same letter is one time unit, the space between letters is three time units, the space between words is seven time units, and the space between messages is ten time units.

C. LocalVLC Modulation

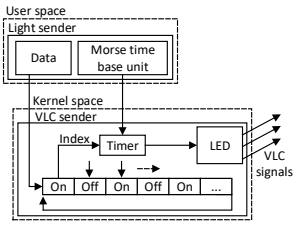
How does the LocalVLC Morse encoding works based on a practical example? We illustrate the processing of a raw light signal, i.e., "hello world", in Fig. 2(b). LocalVLC uses Algorithm 1 for signal decoding. In specific, we quantize the raw voltage signal with a mean threshold to get a binary sequence of light on and off phases (lines: 1-4). In the next step, we detect the change points from light on and off phases and calculate the duration of each phase (lines: 5-11). To improve the robustness, the letter threshold is dynamically



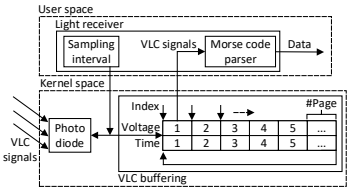
(a) International Morse code



(b) Process a raw light signal to recognize letters, words, and messages.



(c) Processing chain for VLC sender



(d) Processing chain for VLC receiver

Fig. 2: LocalVLC Morse encoding

computed by the mean over all light off phases (lines: 12-14). As predefined by Morse code, the duration thresholds for words and messages are the multiply product of the letter threshold (line: 15). Subsequently, we categorize all light on phases into dot or dash signals based on the corresponding duration (lines: 16-23). This signal series of letters is split into single letters and decoded via a dictionary (lines: 26-28). Finally, we recognize word or message stops to add the correct formatting sign, either a white space or line break (lines: 29-34).

How do the LocalVLC sender and receiver work? We implemented two Linux kernel modules to send and receive data encoded in Morse code. For the VLC sender in Fig. 2(c) the most important parameter is the Morse time base unit which specifies the period for a dot, the smallest time unit for Morse encoding. On this time basis, all signs [a-z, 0-9] can be encoded into different light on and off phases which trigger two real-time kernel timers to switch the LED between on and off state. The VLC sender periodically transmits the same information, e.g., service identifier or password token, for a limited period of time. On the receiving side in Fig. 2(d), we implemented another Linux kernel module which samples the raw light signal via the photodiode. We receive the voltage in mV from the photodiode, where a higher voltage value indicates a light on phase and a lower voltage value indicates a light off phase. We have tested different sampling intervals, i.e., how often voltage values are sampled. This affects the VLC buffering to store and access light signals from the kernel module. Our buffering of light signals dynamically adapts to the available system's memory, to meet different memory constraints ranging from IoT boards to micro-controllers. We

Algorithm 1: Morse Code Processing of LocalVLC

input : voltage $\leftarrow (v_1, v_2, ..., v_n)$, t $\leftarrow (t_1, t_2, ..., t_n)$, morse-code-dict
output: message

Step 1: preprocessing

1 $\overline{v} \leftarrow \frac{1}{n}\left(\sum_{i=0}^{n} \text{voltage}_i\right)$
2 **for** $i \leftarrow 0$ **to** n **do**
3 voltage-on-off [i] $\leftarrow \begin{cases} 1, & \text{if } v_i > \overline{v} \\ 0, & \text{otherwise} \end{cases}$
4 **end**
5 **for** $i \leftarrow 0$ **to** n **do**
6 changepoint [i] $\leftarrow$ voltage-on-off [i+1] - voltage-on-off [i]
7 **end**
8 changepoint-pos $\leftarrow$ `seek` (changepoint = 1 or = -1)
9 **for** $i \leftarrow 0$ **to** n **do**
10 duration [i] $\leftarrow$ t[changepoint-pos [i+1]] - t[changepoint-pos [i]]
11 **end**

Step 2: parsing

12 voltage-on-off $\leftarrow$ voltage-on-off [changepoint-pos]
13 voltage-off-pos $\leftarrow$ `seek` (voltage-on-off == 0)
14 $\theta_{\text{letter}} \leftarrow \frac{1}{n}\left(\sum_{i=0}^{n} \text{duration}[\text{voltage-off-pos}_i]\right)$
15 $\theta_{\text{word}} \leftarrow 2.5 \cdot \theta_{\text{letter}}, \quad \theta_{\text{msg}} \leftarrow 4.5 \cdot \theta_{\text{letter}}$
16 voltage-on-pos $\leftarrow$ `seek` (voltage-on-off == 1)
17 duration-on $\leftarrow$ duration [voltage-on-pos]
18 $\theta_{\text{dash}} \leftarrow \frac{1}{n}\left(\sum_{i=0}^{n} \text{duration-on}_i\right)$
19 dash-pos $\leftarrow$ `seek` (voltage-on-off == 1 and duration $> \theta_{\text{dash}}$)
20 voltage-on-off [dash-pos] = dash
21 letter-pos $\leftarrow$ `seek` (voltage-on-off == 0 and duration $> \theta_{\text{letter}}$)
22 letters $\leftarrow$ `split` (voltage-on-off, letter-pos)
23 duration-off $\leftarrow$ duration [letter-pos]

Step 3: translation

24 message $\leftarrow \emptyset$
25 **for** $i \leftarrow 0$ **to** n **do**
26 letter-on-pos $\leftarrow$ `seek` (letters$_i$ == 1 or == 3)
27 letter-pattern $\leftarrow$ letters$_i$ [letter-on-pos]
28 message $\leftarrow$ `append` (morse-code-dict [letter-pattern])
29 **if** duration-off$_i$ $> \theta_{msg}$ **then**
30 message $\leftarrow$ `append` ("\n")
31 **end**
32 **else if** duration-off$_i$ $> \theta_{word}$ **then**
33 message $\leftarrow$ `append` (" ")
34 **end**
35 **end**

save voltages and the relative time chunked into pages. We control the maximum page size and number of pages based on available memory and sampling interval to provide sufficient information for Morse code parsing in terms of throughput and response time. After parsing the VLC signals, we apply error correction based on a majority rule, i.e., the VLC receiver selects the most frequently received information via a sliding time window of a few milliseconds.

III. Use Cases of LocalVLC

To apply LocalVLC, our demo [13] has covered the indoor wireless (Wi-Fi) authentication scenario to avoid manual distribution and tedious input of passwords for user login. In this section, we further illustrate two IoT oriented use cases to demonstrate how we can practically benefit from the VLC communication capabilities.

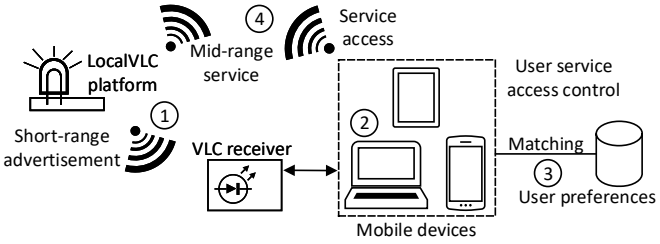


Fig. 3: Indoor service discovery using LocalVLC

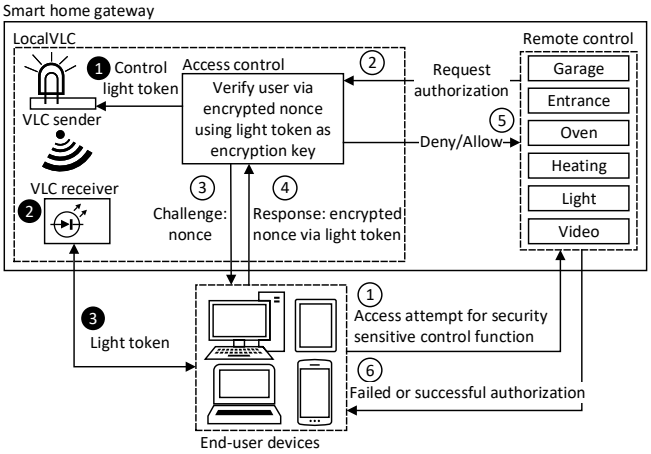


Fig. 4: Key management for smart homes using LocalVLC

A. Indoor Service Discovery

LocalVLC enables proximity-based indoor service discovery without revealing users' private information (e.g., location tags, wireless interface meta data). Our lightweight solution as illustrated in Fig. 3 aims to refine the range of service discovery and improve users' privacy by transforming conventional service advertisements into LocalVLC based signaling. In this setup, users will remain passive (no need of GPS, Wi-Fi or Bluetooth Low Energy (BLE) discovery beacons), collecting advertisements via VLC signaling when approaching the service area, without any need to associate with service hubs. For instance, in a shopping mall with a dense distribution of shops, it is challenging to realize spatially fine-grained service advertisement for commercial coupons (e.g., nowadays manually distributed at the shop-front). Comparing with service announcements over Wi-Fi or Bluetooth, LocalVLC can flexibly reduce the "visibility" of service advertisements only to what is immediately relevant in the vicinity.

By default Morse code lacks support for binary data. Therefore, we apply Base16 encoding to transmit non-alphabetical data such as encrypted service advertisements. The work flow of our solution is illustrated in Fig. 3. The principle is to use short-range VLC advertisements ① dedicated for proximity and location oriented services. The service advertisement is encrypted and includes a location identifier, the password to access the service and a description for the user interface. The VLC receiver obtains and processes the VLC transmitted data. The service advertisements are accumulated over time and shown to users according to pre-defined preferences ②. From a management perspective, users can define preferred services through LocalVLC for specific times and locations. On this basis, the user preferences are matched with the collected service advertisements ③ for carrying out operations (e.g., turning on wireless interfaces or GPS to access certain services). If the user allows the advertised service, the device can access the service flexibly through, for instance, a standard mid-range radio-based communication like Wi-Fi or BLE ④.

B. Key Management for Remote Control of Smart Homes

A smart home incorporates a communication network that connects the key electrical appliances and services, and allows to be remotely controlled, monitored or accessed. Via LocalVLC we fully automate the key management for remote control of smart homes to improve the usability of system's security. To secure remote control of smart homes, existing systems typically use one authentication factor, user name and password, while more secure home controls utilize two-factor authentication. For example, the second authentication step could be a time-based one-time password (TOTP) scheme. A user will generate a random secret key and share the secret key and the validity period with the home control. On this basis, the user generates a new TOTP and enters it at the home control. The drawback of this standard TOTP scheme is the ongoing manual interaction between user and home control. The secret key has to be exchanged, e.g., via a QR code and the user has to manually generate new passwords for each authorization attempt.

To avoid time-consuming user interactions, we integrate LocalVLC into a smart home gateway to realize an automated key management for the remote control of smart homes. We categorize the functionality of the home control into sensitive (e.g., open the entrance door) and standard control functions such as light on and off. The standard control functions can be used remotely and at home. The sensitive control functions can only be used at home via an automated authorization scheme to enhance usability of system's security. Our adapted authorization scheme uses VLC as an out-of-band channel to exchange secret keys and integrates a challenge-response mechanism for on-demand access requests. In this way, we fully automate the user authorization to avoid manual interactions and enabling continuous re-authorization in the background and automatic key revocation based on physical proximity.

Fig. 4 shows our approach to achieve an automated key management for the remote control of smart homes. The home control gateway consists of two independent action flows, distribution of light tokens via LocalVLC and the user authorization at the remote control of the smart home. As basis for user authorization, the access control of LocalVLC generates encryption keys using TOTP and broadcasts it via the VLC sender ❶. The VLC receiver ❷ continuously obtains the light signals and selects the up-to-date encryption key based on a majority rule. The user's device is cable-bound to

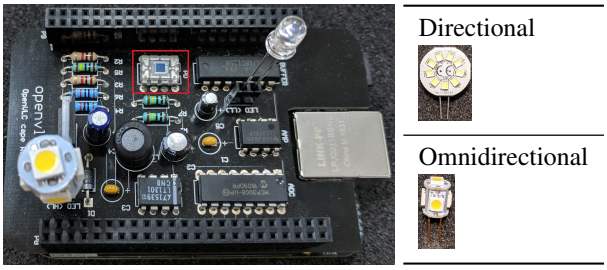


Fig. 5: Evaluation platform with two LED types as transmitters. The photodiode acts as a receiver (highlighted red).

the light receiver ❸ and receives light tokens. On this basis, the user can attempt to access a sensitive home control function via our Android app for smart homes ①. That triggers the remote control of the smart home to request an authorization from the access control of LocalVLC ②. The access control sends a nonce to the end user device ③, the Android app for smart homes increments the received nonce and encrypts it via the Speck cipher [14] using the out-of-band light token as encryption key ④. The access control of LocalVLC performs the same action and compares the encrypted nonce from the end user device with the own nonce. In case the encrypted nonces are equal ⑤, LocalVLC grants access, otherwise denies it. Finally, the Android app for smart home control ⑥ displays the authorization result to the end user. Moreover, we benefit from the visible light as feedback channel for users, i.e., change the color of the light bulb to indicate a successful user authorization or a failed user authorization. The lightweight Speck block cipher is designated for performance limited IoT devices.

IV. Evaluation

The evaluation of LocalVLC is divided into three parts. First, we analyze the LocalVLC system's performance in terms of throughput, latency, error rate, communication field of view (FoV), transmission distance, and impact of ambient light. Second, we compare our LocalVLC data encoding based on Morse code with Manchester encoding regarding throughput and energy consumption. Finally, we evaluate our key management for smart homes based on LocalVLC with regard to latency of light tokens, authorization success rate, and the duration of successful and failed user authorization.

Test Environment: For the evaluation, we use the VLC platform in Fig. 5 from the openVLC project [6] with two different LED types as transmitter and a photodiode as VLC receiver because it is widely deployed on mobile devices, e.g., as ambient light sensor. In all evaluation runs, except for range evaluation, our testbed consists of a sender and receiver in a distance of 50 cm. The sender continuously transmits a test string: "abcdefghijklmnopqrstuvwxyz1234567890". The evaluation encompasses 100 rounds, at which the receiver collects the VLC transmitted data for a duration of ten seconds to compute throughput, error rate, and energy consumption.

A. Micro-benchmarks of LocalVLC System Performance

What are the best operational parameters for LocalVLC encoding based on Morse code? We determine the best VLC parameters in terms of throughput and error rate during VLC transmissions using two different LED types as VLC transmitter: directional and omnidirectional LED. The evaluation includes two test parameters, the sampling interval at the receiver and the Morse time base unit at the sender, both of which affect the throughput and error rate. In detail, at the receiver we use a sampling interval (µs) in the range of [5, 10, 15, 20, 25, 30, 50, 100, 150], which determines how often the photodiode is sampled to receive voltage values and thereby data. At the sender side, we analyze the Morse time base unit which means the time period of one dot, in other words the smallest time base unit to encode the to be transmitted data. Due to different ignition times of the omnidirectional and directional LEDs, we use a different set of time periods for the Morse time base unit (µs): [5, 10, 20, 50, 100, 150] for the omnidirectional and [100, 110, 120, 130, 140, 150] for the directional LED. In case of the omnidirectional LED, Fig. 6 shows the throughput and error rate for different sampling intervals and Morse time base units. On this basis, the best working VLC parameters are a sampling interval of 30 µs and a Morse time base unit of 50 µs. With these parameters we achieve a throughput of 1423.35 Bytes/s without errors. Using the directional LED, Fig. 7 presents the results for throughput and error rate. In this case the best working VLC parameters are a sampling interval of 50 µs and a Morse time base unit of 130 µs resulting in a throughput of 517.68 Bytes/s without errors. The hardware limitations of the directional LED cause a lower sampling interval and a decreased throughput. Regarding a flickering effect at the VLC sender, we transmitted shorter (10 signs) and longer (254 signs) messages and determined a recognizable LED flickering with a Morse time base unit of 150 µs ~6.66 kHz. The frequency range of our VLC sender is above this range using the directional or omnidirectional LED.

Besides the sampling interval and Morse time base unit from the previous section, the page size, how many light signals are buffered, also influences the LocalVLC system performance in terms of throughput and response time until the VLC data is available. **What is the best page size to buffer light signals with regard to a quick response and a high throughput?** Our signal buffering as described in Fig. 2(d) uses a ring buffer and the number of pages scale with the total memory size and page size. The page size of the buffer determines how long it takes until one page is filled with light signals and influences the throughput and user experience in terms of how quick the data is available. Our results in Fig. 8 show the throughput for different page sizes. With a larger page size the throughput increases due to a smaller processing overhead. However, as shown in Table I the response time until the VLC data is available also increases. A fast response time is important for a good user experience, hence we choose a page size of 10k values for the buffering of VLC signals and accept the slightly decreased throughput by 5.35 % resulting

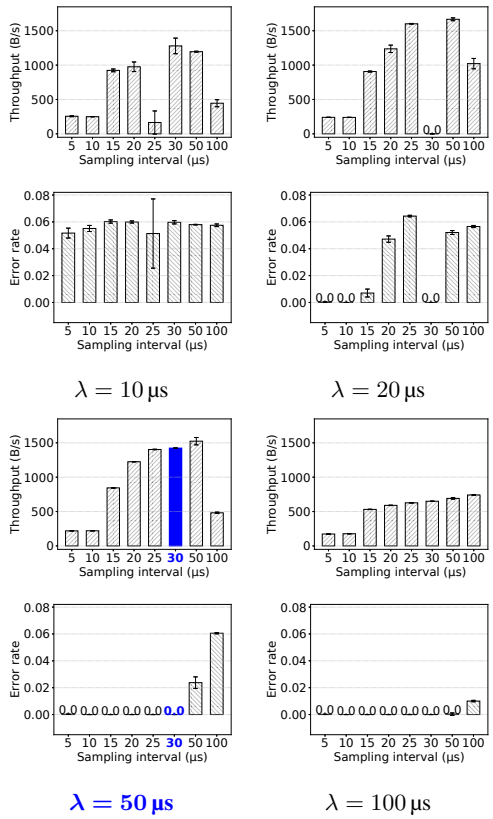


Fig. 6: Evaluation of VLC parameters for LocalVLC encoding using *omnidirectional LED*. We have identified the best working parameters (highlighted blue): sampling interval of 30 µs and Morse time base unit of 50 µs (λ) ~20 kHz.

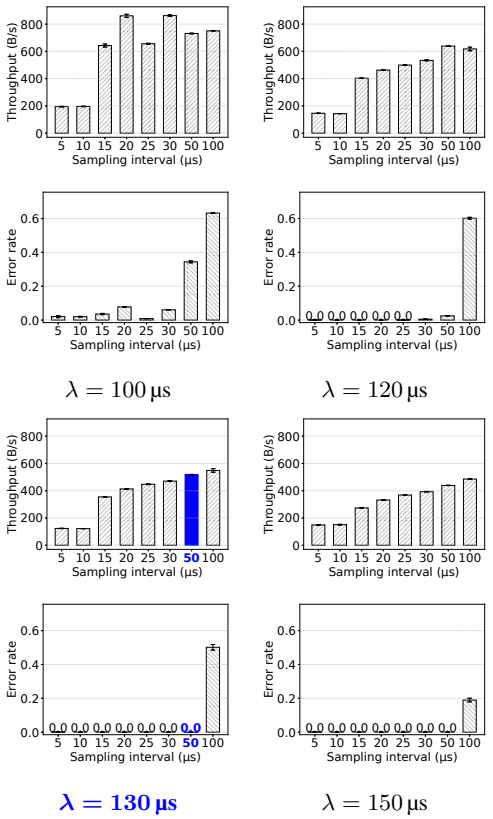


Fig. 7: Evaluation of VLC parameters for LocalVLC encoding using *directional LED*. We have identified the best working parameters (highlighted blue): sampling interval of 50 µs and Morse time base unit of 130 µs (λ) ~7.69 kHz.

in 1500.81 B/s compared to a page size of 30k values with a higher throughput of 1585.57 B/s but a three times greater response time.

What are the VLC communication characteristics regarding maximum range, field of view (FoV), latency, and impact of ambient light? Fig. 9 shows the maximum achievable transmission range with regard to the error rate. The directional LED reaches a maximum distance of 10 m, the omnidirectional LED is able to cover a distance of 3 m. With a larger distance between sender and receiver the error rate increases to a level at which the communication is no longer usable. The throughput only counts the received number of characters within a time frame regardless of whether the data is correct, which is shown by the error rate. We measure the FoV in Fig. 10, the omnidirectional LED obtains a range of 165°–50° and the directional LED achieves a FoV of 175°–5°. In practice, we can utilize mirrors to steer the light signal to dynamically adapt the communication distance and FoV at the given situation. To illustrate the overhead of LocalVLC, the latency using the directional LED results in 0.58 s ± 0.01 s which increases by 25 % (0.77 s ± 0.01 s) if using the omnidirectional LED.

Besides that, we analyze the effect of ambient light at

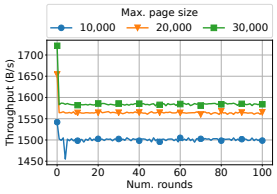


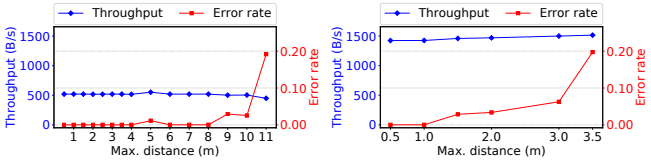
Fig. 8: Throughput (B/s) vs. page size

Page size	Response time
10k	0.3 s
20k	0.6 s
30k	0.9 s

TABLE I: Response time (s) vs. page size

Evaluation of VLC buffer

the omnidirectional LED with a weak light signal and the directional LED with a strong, beaming light signal. Our expectation is that, the stronger the ambient light, the higher the error rate and the lower the throughput. The results in Table II highlight that the directional LED with a throughput between 517 to 654 B/s is less influenced by the ambient light compared to the omnidirectional LED. Nevertheless, with a stronger ambient light the error rate increases significantly limiting a reasonable communication performance using the directional LED. In contrast, the omnidirectional LED only works reliably at a low ambient light intensity. With a slight increase of the



(a) Directional LED (b) Omnidirectional LED

Fig. 9: Maximum range of VLC communication with different LED transmitters regarding throughput (B/s) and error rate

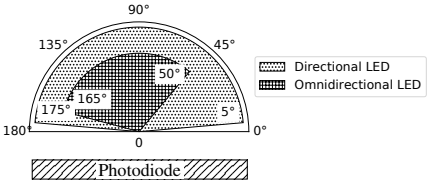
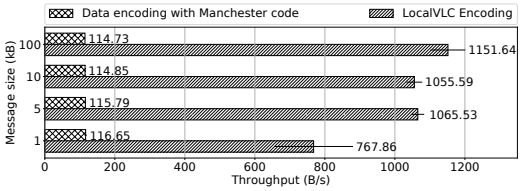
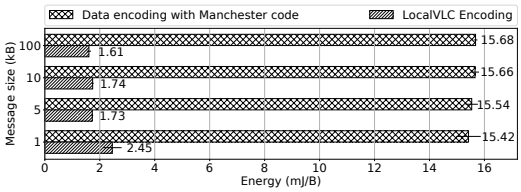


Fig. 10: VLC communication angle: opening and closing field of view at the photodiode (receiver)



(a) Throughput (B/s)



(b) Energy consumption (mJ/B)

Fig. 11: Performance comparison between LocalVLC encoding and Manchester encoding

ambient light, the throughput drops below 25 B/s and the error rate makes it impossible to successfully transmit data. We encounter an unforeseen effect at the omnidirectional LED, with the highest ambient light intensity, the throughput slightly increases and the error rate drops by 50 % compared to the medium intensity of ambient light.

B. Comparison of LocalVLC Encoding with Manchester Code

To highlight performance differences with regard to throughput and energy consumption, we compare our LocalVLC encoding based on Morse code with a baseline using On-Off keying modulation with Manchester code and Reed-Solomon error correction code. In the following, we describe the testbed for the performance measurements. We only use the omnidirectional LED due to missing hardware support by Manchester encoding for the directional LED. As previously determined our LocalVLC encoding works best with a sampling frequency of 20 kHz and Morse time base unit of 50 µs. The data encoding with Manchester code utilizes a 50 kHz sampling frequency. To measure the energy consumption, we use the high voltage Monsoon power device by powering our hardware platform (BeagleBone Black) with 5 V. During the data transmission, we measure the current (mA) and voltage (V) to compute the required energy in Joule per Byte. We measure the energy consumption only at the receiver because the energy consumption at the sender is mainly influenced by the LED power and the data encoding scheme has minor influence on the system's energy. Fig. 11(a) shows in average a 8.75 times higher throughput of LocalVLC encoding with 1010.16 B/s compared to Manchester encoding achieving 115.51 B/s. We obtain a similar result for the energy consumption as shown in Fig. 11(b) because the energy $E = U \cdot I \cdot t$ is mainly dependent on the transmission time. LocalVLC encoding consumes 8.27 times less energy as Manchester encoding, in total numbers 1.88 mJ/B compared to 15.58 mJ/B. Based on these results, our LocalVLC encoding achieves a several orders of magnitude better performance regarding throughput and energy consumption as the usual Manchester encoding. The superior performance of LocalVLC is mainly based on a more efficient Morse encoding compared to Manchester code. Since Morse code takes advantage of encoding pulses and pauses with different length and pauses are more frequent, while the Manchester code requires a high and a low pulse for sending each bit. Furthermore, our error correction, selecting the most frequently received information over a time window, does not decrease the possible payload and is computation-wise faster compared to the Reed-Solomon error correction code.

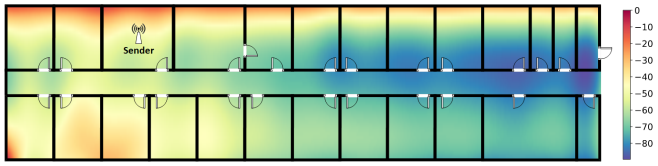
C. Evaluation of Key Management for Smart Homes

We evaluate our key management for smart homes based on out-of-band LocalVLC transmission and automated user authorization scheme. For this environment our testbed experiment in Fig. 12 reveals the competitive advantage of VLC where the coverage of the VLC signals is naturally limited by spatial barriers like walls and doors whereas mid-range radio-based communication covers the entire space. This causes new threats such as localization attacks [15] whereat the adversary track individuals in their home from outside walls by analyzing reflections of ambient Wi-Fi transmissions.

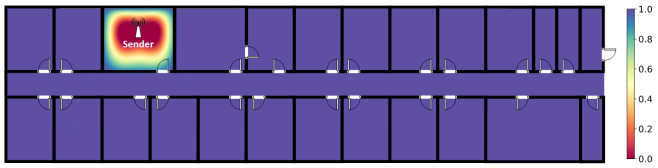
Test setting: Our home control testbed consists of two BeagleBone boards, one acts as home control gateway with integrated LocalVLC for automated key management via VLC transmitted password tokens referred as light tokens. The home control continuously broadcasts light tokens with a predefined refresh period at which the light token randomly changes. The second BeagleBone acts as VLC receiver and selects the light token from the most frequently received messages via a sliding time window. The users' smartphone and the BeagleBone receiver are treated as one device. To perform the automated key management for user authorization at the smart home control, the users' smartphone connects to the

TABLE II: Impact of ambient light at VLC with regard to throughput (B/s) and error rate

Indoor ambient light		Directional LED		Omnidirectional LED	
Level	Intensity (lx)	Throughput (B/s)	Error rate	Throughput (B/s)	Error rate
Low	6.34 ± 0.2	517.08 ± 1.84	0.0 ± 0.0	1423.25 ± 1.84	0.0 ± 0.0
Mid	18.73 ± 0.22	575.22 ± 8.44	0.17 ± 0.01	1.79 ± 0.52	0.71 ± 0.08
High	39.53 ± 0.28	654.09 ± 20.27	0.38 ± 0.04	22.66 ± 1.52	0.35 ± 0.01



(a) Signal strength of Bluetooth low energy (BLE)



(b) Error rate during visible light communication

Fig. 12: Comparison of signal propagation

access point of the home control and tries to access a sensitive home control function, e.g., open the entrance door, which automatically triggers a challenge-response request from the home control gateway to the users' smartphone. Due to space limits we only show the evaluation results for the directional LED as the omnidirectional LED achieves similar results.

How robust is the automated key management? We measure the success rate of user authorizations with a varying token refresh period ranging from 5 s to 60 s. The token period defines the duration at which the home control generates and broadcasts new light tokens. Table III shows the success rate of 5,000 user authorizations. The larger the token refresh period, the fewer failed authorization attempts due to reducing the change frequency of light tokens. The success rate of user authorizations ranges from lowest 84 % to 99 % in case of a token refresh period of 60 s. Due to the latency to receive a light token, each time we change the light token, for a short period of time the users' client has not the up-to-date valid light token and hence the authorization fails. As a result, the less frequently we change the broadcast light token, the more successful are the user authorizations. For a satisfying user experience, a practical system design should allow users to use sensitive home control functions for some limited time (e.g., VLC latency) after initial successful user authorization.

How efficient is the automated key management? In case of a successful user authorization, the users' client has the up-to-date light token, Table III presents the duration ranging from 0.15 s to 0.2 s until the home control functionality is available for the user. In contrast, the user authorization fails,

TABLE III: Evaluation of key management for smart homes

	Token period			
Result	5 s	10 s	30 s	60 s
Success rate	86.68 %	93.3 %	97.62 %	98.84 %
Duration success	0.17 s	0.15	0.18 s	0.2 s
Duration fail	0.72 s	0.73 s	0.76 s	0.75 s

if the users' client does not have the currently valid light token. Using a directional LED and a token refresh period of 30 s, the consecutively failed user authorization attempts took in average 0.76 s. We identify the latency as major influence factor for a successful authorization, if we consider, that the directional LED has a latency to distribute light tokens of 0.6 s ± 0.16 s. Hence, the larger the token refresh period, the better the success rate of the user authorization.

V. Related Work

Regarding VLC transmissions, previous VLC systems [6] utilize Manchester encoding for data transmission causing annoying light blinking. Another work [11] takes advantage of short periods of light signals to overcome the unpleasant visual experience of VLC but requires additional transmission hardware besides the existing illumination infrastructure and increases the light pollution, overlapping illumination and VLC signals working at the same visible light spectrum. LocalVLC improves the usability of VLC by overcoming the LED flickering effect and enables easy VLC deployments based on our 3D printed custom light bulb to replace existing light infrastructure. Thereby, we avoid light pollution by providing illumination together with communication capabilities via VLC.

VLC use cases include LED lights integrated in the ceiling, for indoor localization [16], human identification [17], occupancy detection [18], gesture recognition [11], and activity detection [19]. Another approach [20] utilizes passive light communication at which the environment modulates ambient light signals for data transmission. The reflections caused by the object's surface are received via a photodiode and decoded to read passive information. Our use case with seamless key management for smart homes enables automatic key revocation by fully automating user authorization based on the distance-limited nature of VLC transmissions. Other works

for contextual co-presence enforce proximity by comparing ambient information, e.g., sound [21], acceleration [22], temperature [23], Wi-Fi, LTE, BLE signals [24], [25], or audio signals [26].

VI. Conclusion

LocalVLC is a ready-to-deploy system solution to address the crucial challenge faced by conventional VLC designs: usability in practical deployments. LocalVLC provides a pleasant visual experience by avoiding noticeable flickering effect based on our modulation scheme. Moreover, we are able to equip the existing lighting infrastructure with LocalVLC through a custom light bulb. Some assumptions made in LocalVLC include the repeating transmission of a limited amount of signaling data, e.g., a few bytes, instead of bulk transmission, and a customized error correction by selecting the most frequently transmitted data over a time sliding window. LocalVLC supports transmissions of up to 10 m with a single light source and a throughput ranging from 517.68 B/s to 1423.35 B/s depending on the LED transmitter. Our evaluation reveals that the encoding scheme adopted by LocalVLC provides 8.75 times higher throughput and 8.27 times power saving compared to existing Manchester encoding. Regarding our use cases, LocalVLC can enable seamless key management for smart homes by means of a robust and fast user authorization with 99 % success rate and a duration of 0.2 s.

For future work, we aim to improve the general end-device support for VLC. Most user devices and IoT environments do not have sufficient hardware capabilities for real-time processing of VLC signals and require add-on hardware. Our goal is to shrink the VLC receiver to an appropriate (e.g., coin sized) volume for everyday usage towards ubiquitous VLC. The intended VLC sticker shall be easily attached to different devices for VLC transmissions and supply itself with energy from the light source.

References

[1] H. Haas and S. Dimitrov, *Principles of LED Light Communications: Towards Networked Li-Fi*. Cambridge University Press, 2015.

[2] D. Karunatilaka, F. Zafar, V. Kalavally, and R. Parthiban, "LED Based Indoor Visible Light Communications: State of the Art," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1649–1678, 2015.

[3] G. Corbellini, K. Aksit, S. Schmid, S. Mangold, and T. Gross, "Connecting Networks of Toys and Smartphones with Visible Light Communication," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 72–78, 2014.

[4] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor Positioning with Mobile Phones and Visible Light," in *Proceedings of the 20th International Conference on Mobile Computing and Networking (MobiCom)*, 2014, pp. 447–458.

[5] D. Tsonev, S. Videv, and H. Haas, "Towards a 100 Gb/s Visible Light Wireless Access Network," *Optics express*, vol. 23, no. 2, pp. 1627–1637, 2015.

[6] Q. Wang, D. Giustiniano, and O. Gnawali, "Low-Cost, Flexible and Open Platform for Visible Light Communication Networks," in *Proceedings of the 2nd International Workshop on Hot Topics in Wireless (HotWireless)*, 2015, pp. 31–35.

[7] X. Xu, Y. Shen, J. Yang, C. Xu, G. Shen, G. Chen, and Y. Ni, "PassiveVLC: Enabling Practical Visible Light Backscatter Communication for Battery-free IoT Applications," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2017, pp. 180–192.

[8] Gummeson, Jeremy, J. McCann, C. Yang, D. Ranasinghe, S. Hudson, and A. Sample, "RFID Light Bulb: Enabling Ubiquitous Deployment of Interactive RFID Systems," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 1, no. 2, 2017.

[9] Q. Wang, D. Giustiniano, and M. Zuniga, "In Light and In Darkness, In Motion and In Stillness: A Reliable and Adaptive Receiver for the Internet of Lights," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 1, pp. 149–161, 2018.

[10] A. U. Guler, T. Braud, and P. Hui, "Spatial Interference Detection for Mobile Visible Light Communication," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2018, pp. 1–10.

[11] Z. Tian, K. Wright, and X. Zhou, "The darkLight Rises: Visible Light Communication in the Dark," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2016, pp. 2–15.

[12] International Telecommunication Union, "International Morse Code: Recommendation ITU-R M.1677-1," 2009.

[13] M. Haus, A. Y. Ding, C. Xu, and J. Ott, "Demo: Touchless Wireless Authentication via LocalVLC," in *Proceedings of the 16th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2018, p. 531.

[14] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The SIMON and SPECK Families of Lightweight Block Ciphers," Cryptology ePrint Archive, 2013.

[15] Y. Zhu, Z. Xiao, Y. Chen, Z. Li, M. Liu, B. Y. Zhao, and H. Zheng, "Adversarial WiFi Sensing," Arxiv, 2018.

[16] S. Zhu and X. Zhang, "Enabling High-Precision Visible Light Localization in Today's Buildings," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2017, pp. 96–108.

[17] T. Li, Q. Liu, and X. Zhou, "Practical Human Sensing in the Light," in *Proceedings of the 14th International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2016, pp. 71–84.

[18] Y. Yang, J. Hao, J. Luo, and S. J. Pan, "CeilingSee: Device-Free Occupancy Inference through Lighting Infrastructure Based LED Sensing," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2017, pp. 247–256.

[19] M. Ibrahim, V. Nguyen, S. Rupavatharam, M. Jawahar, M. Gruteser, and R. Howard, "Visible Light based Activity Sensing using Ceiling Photosensors," in *Proceedings of the 3rd Workshop on Visible Light Communication Systems (VLCS)*, 2016, pp. 43–48.

[20] Q. Wang and M. Zuniga, "Passive Sensing and Communication Using Visible Light: Taxonomy, Challenges and Opportunities," *arxiv.org*, pp. 1–6, 2017.

[21] W.-T. Tan, M. Baker, B. Lee, and R. Samadani, "The Sound of Silence," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2013, pp. 1–14.

[22] J. Lester, B. Hannaford, and G. Borriello, ""Are You with Me?" – Using Accelerometers to Determine If Two Devices Are Carried by the Same Person," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, A. Ferscha and F. Mattern, Eds. Springer, 2004, vol. 3001, pp. 33–50.

[23] B. Shrestha, N. Saxena, H. T. T. Truong, and N. Asokan, "Drone to the Rescue: Relay-Resilient Authentication using Ambient Multi-Sensing," in *Proceedings of the 18th International Conference on Financial Cryptography and Data Security*, 2014, pp. 349–364.

[24] Y. Zheng, M. Li, W. Lou, and Y. T. Hou, "Location Based Handshake and Private Proximity Test with Location Tags," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 4, pp. 406–419, 2017.

[25] I. Agadakos, J. Polakis, and G. Portokalidis, "Techu: Open and Privacy-Preserving Crowdsourced GPS for the Masses," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2017, pp. 475–487.

[26] H. T. T. Truong, J. Toivonen, T. D. Nguyen, S. Tarkoma, and N. Asokan, "Proximity Verification Based on Acoustic Room Impulse Response," 2018.

Publication 3

M. Haus, J. Ott, and A. Y. Ding. DevLoc: Seamless Device Association using Light Bulb Networks for Indoor IoT Environments. In *Proceedings of the Fifth IEEE/ACM International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 231–237, 2020. doi:10.1109/IoTDI49375.2020.00030

This thesis includes the accepted version of our article and not the final published version.

Publication Summary

Spontaneous device associations are of particular interest for indoor IoT environments by opening up new opportunities for users to share resources or information. We aim to support two different types of proximity-based applications targeted for users and IoT. Therefore, we take advantage of proximity as a group association technique, i.e., devices find each other when they are in a dedicated space or within a close distance. We introduced the DevLoc framework for device grouping to combine radio-based communication, like Wi-Fi covering larger areas, with visible light signaling to achieve more fine-granular device associations, impossible to recognize with propagating Wi-Fi. In contrast to existing systems for device grouping and to facilitate applications with different spatial expansion of proximity, we provided a complete framework to manage the lighting infrastructure and control the spatial granularity of device grouping. Our custom light bulb is a central part of DevLoc and integrated into a light bulb network to emit light patterns for seamless device associations. In addition, we use a master-slave mechanism for our light bulbs being able to control the user proximity and reduce the effort to adapt lighting infrastructure for DevLoc. Our system supports two modes of device associations depending on the desired application which requires a binding between different entities. For proximity-based services we need device-to-device grouping using only the device's light signals for signal matching and for location-based services we need device-to-area grouping using the device's light signal and an area's reference light signal for signal matching. With respect to a real-world deployment of DevLoc, we evaluated the propagation characteristics of VLC including the impact of ambient light and the field of view (FoV) at the receiver's photodiode. Furthermore, we analyzed the performance of DevLoc in varying environments using two simulations: a) single room with static users and b) multiple rooms with moving users. On this basis, we identified that in general over all test cases the machine learning based device grouping works best in comparison to distance and correlation based device grouping.

DevLoc: Seamless Device Association using Light Bulb Networks for Indoor IoT Environments

Michael Haus
Technical University of Munich
haus@in.tum.de

Jörg Ott
Technical University of Munich
ott@in.tum.de

Aaron Yi Ding
Delft University of Technology
aaron.ding@tudelft.nl

***Abstract*—For indoor IoT environments, spontaneous device associations are of particular interest where users establish a connection in an ad-hoc manner to enable serendipitous interaction. For instance, between a user's personal device and devices the user encounters in the surrounding environment. Our system for device grouping named DevLoc takes advantage of ubiquitous light sources around us to perform continuous device grouping based on the similarity of light signals. To control the spatial granularity of user's proximity, we provide a configuration framework to manage the lighting infrastructure through customized visible light communication. We support two modes of device associations to achieve a binding between different entities: device-to-device and device-to-area allowing either proximity-based or location-based services. Our device grouping includes several methods where in general the machine learning based signal similarity performs best compared to distance and correlation metrics.**

***Index Terms*—Mobile ad hoc networks, Network services, Ubiquitous and mobile devices, Similarity measures, Machine learning approaches**

I. Introduction

We witness a proliferation of wireless devices, such as laptops, mobile phones, tablets, IoT boards, and more. Their wireless capabilities enable flexible formation of ad-hoc groups. Dynamic group association opens up new opportunities for users to spontaneously share resources or information. We aim to support two different types of proximity applications targeted for end users and Internet of Things (IoT). We highlight three use cases for proximity-based, user-oriented applications [1]: 1) Alice is a tourist, rides on the subway and wants to ask locals for the best way to the museum, 2) Bob, a student lands at his college airport and wants to check if anyone from his college is currently at the airport and can give him a ride to campus, and 3) Carol is a manager who wants to automatically record who is present at her daily meetings or share data during a meeting with colleagues and customers. On the other hand, we highlight two use cases for proximity-based IoT applications: 1) IoT boards upload location-tagged data that allows data filtering and data merging from multiple devices at the same area and 2) location-based access policy for consumer smart home platforms [2], e.g., Amazon Echo or Google Home. Therefore, we focus on proximity which has been identified as a group association technique where devices find one another when they are brought within a close distance or a dedicated space [3]. Proximity identifies potential group members and device association refers to the technique that connect group members.

Our system named DevLoc uses visible light signaling for continuous device grouping because light sources are ubiquitous around us ensuring practicality. DevLoc relies on Wi-Fi as the primary communication means in combination with visible light. Since visible light does not pass through opaque objects, it is a good candidate to realize distance-bounding wireless communication compared to the electromagnetic waves of Wi-Fi which easily penetrate physical barriers. Via visible light we achieve more fine-granular device associations based on light bulb coverage which are impossible to recognize with propagating Wi-Fi. On this basis, we can automatically generate meaningful data sharing policies among device groups to define with whom sharing or aggregating data. We offload the task to specify data sharing policies to lower communication layers which are typically handled as part of the application layer in wireless systems used today. Compared to Wi-Fi or other communication technologies, we justify the use of visible light by enabling or automating specific use cases based on the unique characteristic to be sensitive to spatial barriers. This compensates the downsides of visible light such as lack of hardware support at mobile devices. We adopt a master-slave mechanism for light bulbs to minimize the adaption of existing lighting infrastructure for DevLoc.

In contrast to existing systems for device grouping and to facilitate applications with different spatial expansion of proximity, we provide a complete framework to manage the lighting infrastructure and control the spatial granularity of device grouping. We overcome the main disadvantage of location tags [1] that users have no control over the spatial granularity of proximity where the notion of neighborhood is entirely dependent on the type of location tag. Therefore, we enrich the lighting infrastructure by adding light signaling to the widely used Light-Emitting Diode (LED) lamps in residential and office settings. Our custom light bulbs integrate illumination with visible light signaling to automatically link physically nearby devices via the similarity of light patterns. Besides that, we preserve user privacy by comparing low-level context such as surrounding light signals instead of higher-level context like ambient sound. Users may not be comfortable with the idea of sharing context with strangers, even if doing so increases their access to timely and relevant information [4]. Our generated light patterns are ephemeral,

unpredictable nonces associated with a location like a shared pool of entropy between all users at a given location at a given time [1]. The two key properties of light patterns for device grouping are: 1) reproducibility that two measurements at the same place and time yield tags matching with high probability, and 2) unpredictability that an adversary not at a specific place or time is unable to produce a tag that matches the tag measured at that location at that time.

Our key contributions are summarized as follows:

1) We design and develop DevLoc for efficient device grouping based on visible light signaling. We can adapt the lighting infrastructure based on a custom light bulb to control the spatial granularity of user's proximity.
2) We evaluate the physical channel of VLC regarding a real-world deployment of DevLoc. In addition, we perform two simulations for static device-to-device grouping and dynamic device-to-area grouping where we analyze the performance of signal comparison methods.

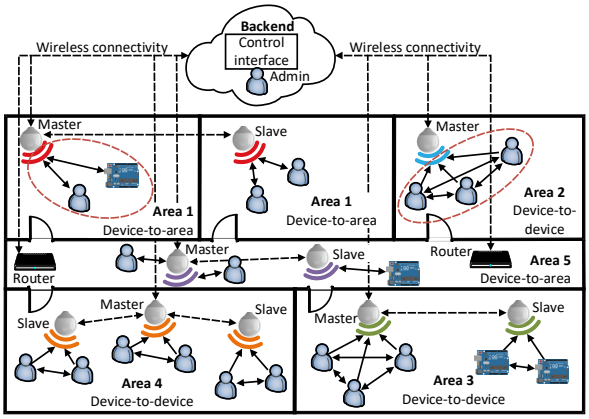
II. Related Work

DevLoc is related to the areas of device pairing, device coupling, device association, and device grouping. Our device grouping is a guidance technique based on proximity in the real world without human interaction. For an overview, the work of [5] categorizes techniques for device associations in the following way: 1) guidance techniques where users act in the real world in order to connect devices via contact, alignment, 2) input focuses on user actions such as trigger commands, entering data, or direct manipulation, 3) enrollment is based on one-time registration of devices with an identity, and 4) matching describes approaches where users compare output of the involved devices to confirm a connection.

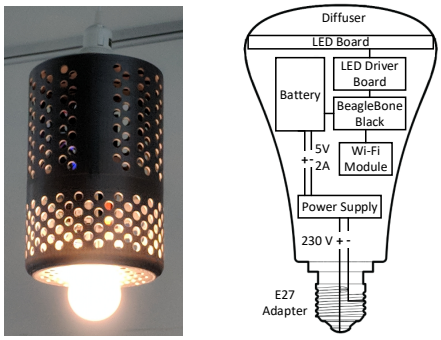
To infer close proximity of users, visible light positioning [6]–[8] is out of scope because we are not interested in the user's position to protect the user's privacy. Instead, we use context information such as ambient light patterns to recognize nearby devices due to the distance-restricted nature of light. In this context, to detect the proximity of devices, other approaches use radio signals [9], ambient audio [10], ambient noise and luminosity [11], accelerometer data caused by hand shaking [12], gait cycle detection of moving users [13], and magnetometer readings of very close devices [14]. The existing work aims to connect mainly two devices whereas DevLoc enables group associations and thereby we are able to flexibly control the granularity of device's proximity. Group association is not just an extension of pairwise association with more users [3]. Instead, many people expect that group association is a single-step procedure rather than multiple pairings. The user study of [3] reveals that close proximity is popular for groupwise associations, but also not rated highly for simplicity.

III. DevLoc System for Device Grouping

We introduce the DevLoc framework for device grouping in Fig. 1(a) where radio-based communication like Wi-Fi covers larger areas and penetrates spatial barriers such as walls, doors.



(a) System overview of DevLoc combining Wi-Fi routers and light bulbs for device associations among users and IoT boards.



(b) DevLoc implements the proposed framework on our customized LocalVLC system [15] to enable visible light signaling for device grouping. We show the deployed 3D-printed light bulb together with the hardware platform.

Fig. 1: DevLoc for seamless device association

We enrich those with visible light signaling to control the spatial granularity of device's proximity based on the master-slave principle of the light bulbs covering larger rooms or across multiple rooms. The different colors at the light bulbs refer to different light patterns used for device associations. The dotted red circles highlight the association among different entities: a) device-to-device using solely the device's light signals for signal comparison or b) device-to-area using the device's light signal and an area's reference light signal for signal matching. DevLoc aims to associate user's mobile devices like tablets, smartphones, laptops, and static IoT boards for data sharing and aggregation. Our custom light bulb in Fig. 1(b) inspired by [16]–[18] is a central part of DevLoc which establishes a Wi-Fi link to the lighting configuration framework and emits light patterns at a high frequency, invisible for human eyes, to seamlessly group devices. This allows us to replace existing illumination infrastructure and we are able to limit the problem of light pollution, at which different visible lights are overlapping for illumination and communication. Our light bulbs to realize device grouping are an implementation choice and not fundamental to the system design. In the following, we describe the setup and working principle of DevLoc.

A. Control the Spatial Granularity of Device Associations

The first task of the administrator is to specify the geographic structure of the device associations by selecting proximity areas. For example, such as in Fig. 1(a) via room numbers for area one and region names like corridor for area five. Initially each light bulb and Wi-Fi router registers itself at the lighting configuration framework running at the backend. Hence, DevLoc knows the light bulbs for all defined areas and randomly chooses for each region one of the light bulbs as master light bulb and the remaining light bulbs act as slaves. We randomly generate a light pattern adapting over time for each master light bulb and the slave light bulb(s) simply broadcast the same light pattern given from the master light bulb. On this basis, we can flexibly define the spatial granularity of device proximity based on the master-slave mechanism of our light bulbs by changing the groups of light bulbs covering different areas. For instance, we cover larger regions by using the same light pattern in different rooms which are semantically the same area such as area one in Fig. 1(a) covering two rooms compared to other areas limited by the room boundaries. The achievable spatial granularity of device associations is defined by the size of rooms and regions like corridors, and the number and distribution of light bulbs. For the most fine-granular proximity, each light bulb works on its own as master light bulb without any slave light bulbs. Our custom light bulb provides a communication range of up to 10 m. The master-slave mechanism of our light bulbs ensures a minimum of technical adaptions on existing illumination because only the master light bulbs need computing power to perform the device associations. The slave light bulbs require only a Wi-Fi connection to receive the commands from the master light bulb to broadcast a specific light pattern.

B. Triggering Device Associations

We combine Wi-Fi routers and our custom light bulbs for triggering device associations. The second task of the administrator is to specify one Wi-Fi router for each master light bulb which continuously monitors the wireless connections of the Wi-Fi router. Due to the larger Wi-Fi coverage the binding is 1:m meaning one router is linked to multiple master light bulbs. If there are no device groups yet and in case of changes on the Wi-Fi connections, each linked master light bulb requests the continuously broadcasted light pattern received from the client(s) and initiates the device association to infer which devices are in the same light communication range instead of being only in the same Wi-Fi coverage. In case of a new Wi-Fi client, the master light bulb performs the signal comparison to infer the matching device group without affecting other devices. When a Wi-Fi device disappears at the router the master light bulb removes this single client from existing device groups. Besides that, the mobility of user's also affects the triggering of device associations. In case of static users it is sufficiently to observe the Wi-Fi connections for device grouping. However, for users moving between different areas but still connected to the same Wi-Fi router we need to manually start the device grouping for current device groups via a predefined period like every few seconds. We do not use signal strength changes of the user's Wi-Fi connection to update the device association because it is sensitive to reflections and shadowing due to moving and static objects, location and distance of users to the router, and layout and material of the building. The Wi-Fi signal strength can change unexpectedly and gives excessive false positives and false negatives causing frequent device association updates which decrease the user experience.

C. Entities for Device Association: Device and Area

As illustrated in Fig. 1(a), the third and last task of the administrator is to define the mode of device associations for each master light bulb depending on the desired application using location-based services (LBS) or proximity-based services (PBS). Device-to-device association for PBS and device-to-area association for LBS. LBS are based upon the absolute position of a user to answer the question "where we are?". In contrast, PBS are based upon context information to find co-location with other points of interest to answer the question "who are we with?". The goal of LBS and PBS is to improve the users' daily lives by providing a personalized service to enable sharing of location information and location-aware information retrieval. We identify three main differences between device-to-device and device-to-area associations: 1) trigger point in time of the device grouping, 2) required number of clients for device association, and 3) signal comparison among different entities influencing the associated binding either device-to-device or device-to-area.

In case of device-to-device associations, triggering the device grouping requires at least two connected clients at the router which is linked to one or multiple master light bulbs. To match a Wi-Fi client to a device group, the master light bulb randomly chooses one client from each existing device group for signal comparison. The established binding between nearby clients lacks the information in which specific geographic region the clients are positioned. Hence, we can only realize PBS like data sharing among close users. LBS are not feasible such as indoor localization of a single or group of users.

On the other hand for device-to-area associations, the master light bulb(s) start the device grouping immediately after the client connected to the Wi-Fi router and compare the client's signal to the area's reference signal. We establish a direct binding between the device and area, and hence we know which device is in which region and at the same time which other devices are nearby. Via LBS including PBS we are able to offer more user services. In addition, there is no restriction in terms of the number of connected clients, e.g., to have at least two connected clients for device-to-device binding. Device-to-device grouping provides less location-specific information compared to the device-to-area grouping.

D. Generation and Detection of Light Patterns

Our custom light bulb broadcasts random light patterns for device grouping. We independently generate a random series of light on and off periods and merge them afterwards to a

light pattern. The duration of each light on and off period is in the range of [1, 5] ms. The minimum duration is defined via the fastest sampling rate achievable by the hardware of our light receiver, how fast the photodiode can be sampled. We determine the maximum duration of each light on and off period by avoiding unpleasant visual experience where light flickering effects are visible. The sender periodically broadcasts the light pattern for a limited time period. The length of the light pattern must be a multiple of two to be able to distinguish different light patterns, i.e., after each light on period appears a light off period. To improve the detection rate of light patterns at the light receiver, we check for each light on and off series if the time periods are sufficiently diverse that each duration is more than 10 % different from the other periods. Our light receiver samples the raw light signal via a photodiode and receives the voltage in mV where a higher voltage indicates a light on period and a lower voltage indicates a light off period.

How to detect cycles in the light signal? To find repeating patterns, we apply the cycle detection algorithm from [13] on our light pattern. The algorithm achieves a reliable signal segmentation based on normalization and supports signal similarity of arbitrary co-aligned sensor data. The algorithm's input is a vector of voltage amplitudes $z = (z_1, ..., z_n)$ and the output is a sequence of consecutive light signal cycles. We utilize auto-correlation and distance calculation to find repetitive signal parts. The auto-correlation is efficiently calculated via the Wiener-Khinchin theorem [19] with complexity $n \cdot log(n)$

$$\begin{aligned} F_R(f) &= FFT[z] \\ S(f) &= F_R(f) \cdot F_R^*(f) \qquad * \hat{=} \text{conjugate} \\ R(\tau) &= IFFT[S(f)] \end{aligned}$$

where z are the voltage amplitudes. The resulting auto-correlation $R(\tau)$ leads to m non-ambiguous local maxima $\zeta = \arg\max(R(\tau)) = \{\zeta_1, ..., \zeta_i ..., \zeta_m\}$. We calculate the distances between all local maxima and a mean distance

$$\delta_{\text{mean}} = \left\lceil \frac{\sum_{i=1}^{m-1} \zeta_{i+1} - \zeta_i}{m-1} \right\rceil$$

where δ_{mean} can be used to select minima indices from z which represent signal cycles. To be specific, each local maxima defines a start point and δ_{mean} a search range to find the local minima

$$\begin{aligned} \mu &= \{\mu_1, ...\mu_i, ..., \mu_{m-1}\} \\ \mu_i &= \arg\min(z_{\zeta_i}, z_{\zeta_i+1}, ..., z_{\zeta_i+\delta_{\text{mean}}}) \end{aligned}$$

Every μ_j represents the index of a minimum in z limited to the range of δ_{mean}. The indices in μ are used to split the voltage amplitude z into light patterns

$$\begin{aligned} Z &= \{Z_1, ..., Z_i, ..., Z_q\} \\ Z_i &= (z_{\mu_{\frac{i}{2}}}, ..., z_{\mu_i}, ..., z_{\mu_{\frac{i+1}{2}}-1}) \text{ with } i = \{2, 4, ..., q\} \end{aligned}$$

This method works reliably only for simple light patterns with a maximum length of six on and off periods. Hence, we introduce our own method to detect light patterns taking into account the period of each light on and off phase. We summarize the light signal into a list of periods

$$\hat{z} = \{(s_1, d_1), ..., (s_n, d_n)\}$$

where $s_n \in \{0, 1\}$ describes whether the light is on or off and $d_i \in \mathbb{Z}$ specifies the duration of each period. We merge similar signal parts with a difference smaller than 10 % because the light sender introduced a 10 % dissimilarity among the light on and off periods to enhance the robustness of signal cycle detection. The remaining unique signal parts define the parts of the signal pattern together with the length. We overlay the light signal with a time window defined by the pattern length to identify the light pattern. We don't know the start position of the light pattern resulting in $m = {}^{\text{len}(\hat{z})}/_2 + 1$ possible candidates as light pattern.

E. Implementation Details of DevLoc

We use the small, low-cost, single-board computer BeagleBone Black as system and development platform. We have implemented two Linux kernel modules to broadcast light patterns at the light bulb and receive light patterns at the user's mobile device. We use MQTT for the communication among light bulbs. Each master light bulb subscribes to the central backend to receive the configured light pattern which is further published to the slave light bulb(s). In addition, we take advantage of Python twisted, an event-driven network programming framework where we provide short callbacks to receive and send data between light bulbs and user clients.

IV. Evaluation of Device Associations via DevLoc

We evaluate the physical channel of VLC with respect to a real-world deployment of DevLoc including the impact of ambient light and the field of view (FoV) at the receiver's photodiode. In addition, to analyze the performance of DevLoc in varying environments, we simulate two different environments with static and moving users to identify for each case the best working device grouping in terms of high detection accuracy and low runtime.

A. Properties of VLC Physical Channel

Regarding a real-world deployment of DevLoc, we have evaluated in [15] the impact of ambient light at VLC by using two different LEDs as VLC transmitter, an omnidirectional LED with a weak light signal and a directional LED with a strong, beaming light signal. The results have shown that the directional LED is less influenced by the ambient light compared to the omnidirectional LED. Nevertheless, with a stronger ambient light similar to an active light source or direct sunlight the performance to detect signal patterns using the directional LED drops significantly. In contrast, the omnidirectional LED only works reliably at a low ambient light intensity. In addition, we measure the FoV at the photodiode of the receiver. The omnidirectional LED obtains a range of 165°–50° and the directional LED achieves a FoV of 175°–5°. As future work, we plan to enhance the robustness of

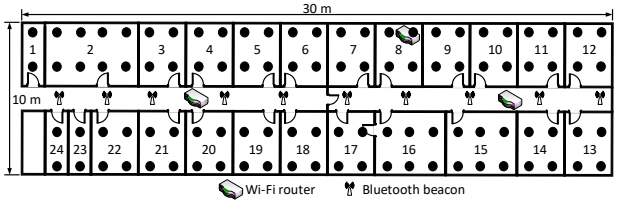


Fig. 2: We use our university lab as simulation environment for device associations which consists of 24 rooms with different sizes. We take real traces of the Wi-Fi and Bluetooth environment at different positions over the university lab for comparison with device localization.

DevLoc by analyzing the effect of overlapping light patterns from different light bulbs whether we are able to separate and identify the different signals. We will adopt the algorithm in [20] which uses orthogonal codes to detect and isolate adjacent light sources, e.g., light markers for object identification.

B. Simulation Settings for Device Association

We evaluate DevLoc via a dedicated simulator running two different simulations with static and dynamic users. To support this, we have previously identified the best working parameters for device grouping which are summarized and italicized in Table I. We perform a trace-driven simulation with persisted environment data from our university lab as shown in Fig. 2. Thereby, we compute statistical features (min, max, median, var, std, mean, sum, length) and time-series tailored features via tsfresh [21] for light patterns. To enable the simulations, each grouping client uses three different real traces: Wi-Fi and Bluetooth scans, and random light patterns with varying length. We achieve a realistic simulation by imitating the network latency between the grouping server and the clients by a random waiting time before each client sends the requested environment data to the grouping server. Thereby, we choose a random start within the sensing range for light patterns, Wi-Fi and Bluetooth scans, and we randomly select a sampling period within the identified best working sampling ranges.

C. Static Device-to-Device Simulation of Device Grouping

Simulation settings In the static simulation no user is moving and each user remains in the same room. The grouping server waits until all devices are connected and starts the device grouping. Table I shows the parameters for the static simulation. We perform the device grouping using random light patterns with a varying length $\in \{2, 4, 6, 8, 10\}$ and for at least two users up to ten users.

Simulation results By using 10-fold cross validation, Table II presents the best working device grouping technique (highlighted in bold) in terms of a fast reasoning and a reasonable total result, meaning the average over accuracy, precision, recall, and F1-score. The ML-based device grouping performs similar or slightly better than the signal similarity metrics like Spearman and Pearson. In contrast, the device localization using Bluetooth and Wi-Fi features works far worse. Regarding the runtime of our device grouping, the median time to receive data for device grouping is about 1.41 s which stands for 83.93 % of total time in comparison to the device grouping with 0.27 s and 16.07 % of the total time.

TABLE I: Settings from parameter estimation (italic) and simulation parameters (bold) for device grouping

Simulation	Parameters	
Static & Dynamic	**Similarity metrics**	Pearson, Spearman
	Similarity threshold	0.7
	Similarity equalize method	DTW
	Localization classifiers	Content-based filtering, random forest, SVM
	Sampling period localization	5 s
	Similarity classifiers	Random forest, extra trees, gradient boosting
	Sampling period to train similarity classifiers	50 ms
Static	**Light patterns**	[2, 4, 6, 8, 10]
	Users	[2, 3, 4, 5, 6, 7, 8, 9, 10]
Dynamic	**Grouping frequency**	[10, 20, 30] s
	Users	[3, 5, 10]
	Rooms	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

For a thorough evaluation we further analyze the performance using different number of grouping users. We reach the highest total result of 0.97 with six grouping users because with less users the grouping signals lack significant patterns and with more users the noise over the grouping signals increases leading to a higher error rate for device grouping. In the following, we show the number of grouping users with descending total result in brackets, meaning the average over accuracy, precision, recall, and F1-score: 6 (0.97), 4 (0.94), 3 (0.93), 8 (0.93), 5 (0.92), 9 (0.92), 10 (0.92), 7 (0.92), 2 (0.87). Moreover, we evaluate the performance of light patterns with different lengths for device grouping. The light pattern with four random on and off periods works best compared to a decrease of 9 % using the worst light pattern with ten random periods. In detail, we present the average performance of light patterns with different lengths sorted by descending total result in brackets: 4 (0.97), 2 (0.95), 6 (0.91), 8 (0.9), 10 (0.88).

D. Dynamic Device-to-Area Simulation of Device Grouping

Simulation settings In comparison to the static simulation, the users are moving between different rooms in the dynamic simulation. Fig. 2 shows our simulation environment for device associations where the rooms are positioned in a rectangular grid with an intra room distance of 2 m and inter room distance of 3 m and we calculate the distances among all room combinations. For each user we calculate a random path between the rooms using the duration of one simulation iteration of 20 min and distribute the time as duration of stay over the rooms using a multinomial distribution. As a result, the user's random path is a list of tuples with duration of stay

TABLE II: Best working classifiers and features for device grouping with static and moving users

Simulation	Grouping technique	Feature type	Result	Runtime	Accuracy	Precision	Recall	F1-score
Static	Gradient boosting	Selected tsfresh	.96	4.02 s (8)	1	.94	.94	.94
	Gradient boosting	Selected statistical	.83	0.75 s (4)	.89	.81	.81	.8
	Gradient boosting	**Statistical**	**.81**	**0.45 s (2)**	**1**	**.75**	**.75**	**.75**
	Spearman	Light pattern	.81	0.49 s (3)	1	.75	.75	.75
	Spearman	Duration of light pattern	.81	0.42 s (1)	1	.75	.75	.75
	Pearson	Light signal	.64	2.06 s (5)	.25	.77	.79	.76
	Random forest	Bluetooth	.43	2.64 s (7)	.34	.48	.52	.38
	SVM	Wi-Fi	.31	2.61 s (6)	.32	.2	.44	.26
Dynamic	Pearson	Duration of light pattern	.95	0.46 s (3)	1	.93	.93	.93
	Pearson	Light pattern	.95	0.47 s (4)	1	.93	.93	.93
	Pearson	**Light signal**	**.95**	**0.28 s (2)**	**.95**	**.95**	**.95**	**.95**
	Gradient boosting	Selected tsfresh	.93	1.58 s (8)	.9	1	.9	.93
	Gradient boosting	Selected statistical	.93	0.53 s (5)	.9	1	.9	.93
	Content-based filtering	Bluetooth	.84	0.59 s (6)	.95	.81	.81	.81
	Content-based filtering	Wi-Fi	.84	0.61 s (7)	1	.78	.78	.78
	Extra trees	Statistical	.83	0.26 s (1)	.75	1	.75	.83

for each room where the user stays and moves to the next room if the duration of stay expired. For example, user A has the random path: [(120, 1), (300, 3), ...] which specifies that the start position is in room 1 and after 120 s the user moves to room 3 and stays there for 5 min, and so forth. Thereby, we randomly create user groups for each room, i.e., at which simulation time how many users are in the same room. During the simulation each user chooses a random movement speed in the range of 1.25 to 1.53 m/s (4.5–5.5 km/h) [22] for each movement between rooms. If the users are in motion they are in the corridor and not associated with any room. For device grouping, each room is associated with unique location-dependent environment data including Wi-Fi and Bluetooth scans, and light patterns and acts independently of other rooms. Table I shows the parameters for dynamic device-to-area simulation covering grouping frequency, number of users, and number of rooms.

Simulation results Via 10-fold cross validation, Table II shows the best working device grouping (highlighted in bold) with respect to a fast runtime and a reasonable total result, meaning the average across accuracy, precision, recall, and F1-score. In contrast to the static simulation, the device grouping based on similarity metrics works slightly better compared to ML-based device grouping. The device localization using Wi-Fi and Bluetooth features achieves a similar result. With respect to the runtime of our device grouping, the median time to receive data for device grouping is about 0.43 s (71.67 % of the total time) in comparison to the device grouping with 0.17 s (28.33 % of the total time). Moreover, we analyze the performance of device grouping with a varying number of rooms, sorted after decreasing total result in brackets: 1 (0.99), 2 (0.96), 3 (0.92), 5 (0.9), 6 (0.89), 4 (0.87), 8 (0.84), 7 (0.82), 9 (0.79), 10 (0.76). The device grouping works best with less rooms because the more rooms the higher the chance that the user lacks the up-to-date light pattern of the designated room due to movement or decoding issues. Besides that, the frequency of device grouping with 20 s works best, the accuracy of device grouping decreases by 16 % with 30 s and with a 10 s frequency the accuracy decreases another 8 %.

To sum up, Table II shows two different best working classifiers and features for device grouping depending on the use case either for static or moving users. The scenario with several rooms and moving users is more realistic in practice and we favor this approach for device association.

V. Conclusion

DevLoc is a ready-to-deploy system solution to enable seamless device grouping based on visible light signaling for data sharing and aggregation. Our custom light bulb broadcasts light patterns so that clients detect cycles in the light patterns for device grouping. Our evaluation of DevLoc via two simulations with a single room and static users and multiple rooms with moving users reveals that in general the machine learning based signal similarity performs best compared to distance and correlation metrics.

To ensure that DevLoc protects the user's privacy during device grouping at the light bulb, we will analyze how to apply fully homomorphic encryption for time-series data where multiple parties compute whether they are nearby without learning each other's inputs. Moreover, by using our Morse-code inspired modulation scheme from LocalVLC [15], we plan to encode a location identifier emitted by our custom light bulb(s) for device grouping and compare the results with the device grouping via light signal patterns in terms of robust device grouping in presence of light interference.

REFERENCES

[1] A. Narayanan, N. Thiagarajan, and M. Lakhani, "Location Privacy via Private Proximity Testing," in *Proceedings of the 18th Annual Network and Distributed System Security Symposium (NDSS)*, 2011.

[2] S. Mare, L. Girvin, F. Roesner, and T. Kohno, "Consumer Smart Homes: Where We Are and Where We Need to Go," in *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2019, pp. 117–122.

[3] M. K. Chong and H. W. Gellersen, "How Groups of Users Associate Wireless Devices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2013, pp. 1559–1568.

[4] A. A. de Freitas and A. K. Dey, "Using Multiple Contexts to Detect and Form Opportunistic Groups," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW)*, 2015, pp. 1612–1621.

[5] M. K. Chong, R. Mayrhofer, and H. Gellersen, "A Survey of User Interaction for Spontaneous Device Association," *ACM Computing Surveys*, vol. 47, no. 1, pp. 1–40, 2014.

[6] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor Positioning with Mobile Phones and Visible Light," in *Proceedings of the 20th International Conference on Mobile Computing and Networking (MobiCom)*, 2014, pp. 447–458.

[7] Z. Yang, Z. Wang, J. Zhang, C. Huang, and Q. Zhang, "Lightweight Visible Light Positioning for Wearables," *GetMobile: Mobile Computing and Communications*, vol. 19, no. 3, pp. 18–21, 2015.

[8] ——, "Wearables Can Afford: Light-Weight Indoor Positioning with Visible Light," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2015, pp. 317–330.

[9] A. Varshavsky, A. Scannell, A. LaMarca, and E. de Lara, "Amigo: Proximity-Based Authentication of Mobile Devices," in *Proceedings of the 9th International Conference on Ubiquitous Computing (UbiComp)*, 2007, pp. 253–270.

[10] D. Schürmann and S. Sigg, "Secure Communication Based on Ambient Audio," *Secure Communication Based on Ambient Audio*, vol. 12, no. 2, pp. 358–370, 2013.

[11] M. Miettinen, N. Asokan, T. D. Nguyen, A.-R. Sadeghi, and M. Sobhani, "Context-Based Zero-Interaction Pairing and Key Evolution for Advanced Personal Devices," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2014, pp. 880–891.

[12] R. Mayrhofer and H. Gellersen, "Shake Well Before Use: Intuitive and Secure Pairing of Mobile Devices," *IEEE Transactions on Mobile Computing*, vol. 8, no. 6, pp. 792–806, 2009.

[13] D. Schürmann, A. Brüsch, N. Nguyen, S. Sigg, and L. Wolf, "Moves like Jagger: Exploiting variations in instantaneous gait for spontaneous device pairing," *Pervasive and Mobile Computing*, vol. 47, pp. 1–12, 2018.

[14] R. Jin, L. Shi, K. Zeng, A. Pande, and P. Mohapatra, "MagPairing: Pairing Smartphones in Close Proximity Using Magnetometers," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1306–1320, 2016.

[15] M. Haus, A. Y. Ding, and J. Ott, "LocalVLC: Augmenting Smart IoT Services with Practical Visible Light Communication," in *Proceedings of the 20th IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2019, pp. 1–9.

[16] S. Schmid, J. Ziegler, G. Corbellini, T. R. Gross, and S. Mangold, "Using Consumer LED Light Bulbs for Low-Cost Visible Light Communication Systems," in *Proceedings of the 1st ACM MobiCom Workshop on Visible Light Communication Systems (VLCS)*, 2014, pp. 9–14.

[17] Gummeson, Jeremy, J. McCann, C. Yang, D. Ranasinghe, S. Hudson, and A. Sample, "RFID Light Bulb: Enabling Ubiquitous Deployment of Interactive RFID Systems," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 1, no. 2, 2017.

[18] S. Schmid, T. Bourchas, S. Mangold, and T. R. Gross, "Linux Light Bulbs: Enabling Internet Protocol Connectivity for Light Bulb Networks," in *Proceedings of the 2nd International Workshop on Visible Light Communications Systems (VLCS)*, 2015, pp. 3–8.

[19] L. Cohen, "Generalization of the Wiener-Khinchin Theorem," *IEEE Signal Processing Letters*, vol. 5, no. 11, pp. 292–294, 1998.

[20] A. U. Guler, T. Braud, and P. Hui, "Spatial Interference Detection for Mobile Visible Light Communication," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2018, pp. 1–10.

[21] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package)," *Neurocomputing*, vol. 307, pp. 72–77, 2018.

[22] N. Carey, "Establishing Pedestrian Walking Speeds," *Portland State University*, 2005.

Publication 4

M. Haus, A. Y. Ding, and J. Ott. Multimodal Co-Presence Detection with Varying Spatio-Temporal Granularity. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 1–7, 2020. doi:10.1109/PerComWorkshops48775.2020.9156105

This thesis includes the accepted version of our article and not the final published version.

Publication Summary

Today's pervasive computing environments are characterized by a plethora of sensing and communication-enabled devices that diffuse themselves among different users. Built-in sensors and telecommunication infrastructure allow co-presence detection. By identifying semantically close entities, we can realize different context-aware applications, such as data offloading, distributed ad hoc networking, romantic matchmaking. In contrast to our work, most of existing studies only analyze device-to-device proximity and not the proximity of device groups, without evaluating the impact of other factors, such as user mobility and hardware heterogeneity on proximity reasoning. We can infer the proximity of mobile devices in two different ways: 1) by calculating distances between entities using coordinates from a positioning system, and 2) by computing similarities of context information that is sensitive to the user's activity or location. We focus on the latter case, i.e., similarity of sensor or wireless data for user's proximity. We aim to support developers building better context-aware applications by a deepened understanding of which set of context information is appropriate for co-presence detection. The main problem for our co-presence detection is a practical dataset, including several proximity verification sets and sensor data of mobile devices where we ensure the physical co-presence of user devices. Therefore, we conducted a study with 126 subjects as part of the lecture on 'social computing', over three months, that resulted in a multimodal dataset for co-presence detection. Our major study question is whether multimodal sensor data (barometer, magnetometer, and accelerometer) are suitable to achieve a co-presence detection with varying spatio-temporal granularity? We showed that sensor modalities are suitable for co-presence detection with a signal distance ratio of 5.6x among nearby and remote users. To evaluate the impact of user mobility on co-presence detection, we identified user groups with different mobility behavior and we found that the user's mobility has a negligible effect on the co-presence detection. Regarding the impact of device heterogeneity on co-presence detection, we quantified the effect of different sensing ranges and sensitivities of device sensors by creating device groups with same and mixed sensor hardware for co-presence detection. As a result, the device heterogeneity has a major impact decreasing the co-presence accuracy by 47 %. With respect to usability, we analyzed the impact of the energy consumption of different device sensors for proximity sensing on the limited battery capacity of mobile devices. The system idle dominates the total energy consumption of the smartphone with 98 %, compared to the phone sensors with only 2 %.

Multimodal Co-Presence Detection with Varying Spatio-Temporal Granularity

Michael Haus[†], Aaron Yi Ding[*], Jörg Ott[†]

[†]Department of Computer Science, Technical University of Munich, Germany
[*]Department of Engineering Systems and Services, Delft University of Technology, Netherlands

***Abstract*—Pervasive computing environments are characterized by a plethora of sensing and communication-enabled devices that diffuse themselves among different users. Built-in sensors and telecommunication infrastructure allow co-presence detection. In turn, co-presence detection enables context-aware applications, like those for social networking among close-by users, and for modeling human behavior. We aim to support developers building better context-aware applications by a deepened understanding of which set of context information is appropriate for co-presence detection. We have gathered a multimodal dataset for proximity sensing, including several proximity verification sets, like Bluetooth, Wi-Fi, and GSM encounters, to be able to associate sensor's data with a spatial granularity. We show that sensor modalities are suitable to recognize the spatial adjacency of users with different spatio-temporal granularity. We find that individual user mobility has only a minor, negligible effect on co-presence detection. In contrast, the heterogeneity of device's sensor hardware has a major negative impact on co-presence detection. To reveal energy pitfalls with respect to usability, we perform an energy analysis pertaining to the usage stemming from different sensors for co-presence detection.**

***Index Terms*—Co-presence detection, Multimodal sensor dataset, User mobility, Device heterogeneity, Sensor energy use**

I. Introduction

Portable devices accompany mobile users almost everywhere surrounded by a pervasive wireless infrastructure that is typically composed of Cellular, Wi-Fi, and Bluetooth. Besides providing connectivity, these infrastructures offer an unprecedented opportunity for co-presence detection which is further supported by sensors typically included in mobile devices, such as smartphones and tablets. A quantitative measure of co-presence defines two individuals as "close" when their similarity of context information is large [1]. Context describes any information that can be used to characterise the situation of a person, place, or object that is considered relevant to the interaction between a user and an application [2]. The ability to identify semantically close entities enables context-aware applications, such as data offloading [3], distributed ad hoc networking [4], romantic matchmaking, and social networking [5]. Mobile users of social networks mainly rely on their virtual online communities, which lack the "physical" and contextual interactions among users. We can augment social networks with local interactions by using proximity as a metric to determine who is discoverable on a network of spontaneously and opportunistically connected nodes.

The proximity of mobile devices is typically inferred in two ways: 1) by calculating distances between entities using coordinates from a positioning system, and 2) by computing similarities of context information that is sensitive to the user's activity or location. In this work, we focus on the latter case, i.e., similarity of sensor or wireless data for user's proximity. We address the question, whether multimodal sensor data are suitable to achieve a co-presence detection with varying spatio-temporal granularity? The main problem is a practical dataset, including several proximity verification sets and sensor data of mobile devices where we ensure the physical co-presence of user devices. Therefore, we conducted a study with 126 subjects as part of the lecture on "social computing", over three months, that resulted in a multimodal dataset for co-presence detection (details in Section III). Our study identifies how effective sensor modalities, e.g., barometer, are to detect physical proximity of users with different spatio-temporal granularities. Moreover, we highlight the impact of device heterogeneity due to different sensor hardware and user's mobility involving movement patterns and variability on co-presence detection of spontaneous groups.

We summarize our contributions as follows:

- We gathered a multimodal dataset for co-presence detection, including multiple proximity verification sets, to be able to associate sensor's data with spatial granularity. Our dataset is publicly available [6] as an anonymized subset.
- We show that sensor modalities are suitable for co-presence detection with a signal distance ratio of 5.6x among nearby and remote users. We quantify the impact of device heterogeneity where the co-presence accuracy decreases by 47 %, while the user's mobility has a negligible effect on the co-presence detection.
- We perform an energy analysis on mobile devices to assess the energy demand of different sensors for co-presence detection. The system idle dominates the total energy consumption of the smartphone with 98 %, compared to the phone sensors with only 2 %.

II. Related Work

Our work can be positioned within the field of proximity detection or co-presence detection (used interchangeably in this paper). A number of approaches have been proposed for co-presence detection using the similarity of Bluetooth

signals [7], Wi-Fi signals [8], ambient sound [9], images [10], and accelerometer data [11]. Some works concentrate on the estimation of face-to-face interaction among users up to 1.5 m using Bluetooth signals [12], proximity sensors [13], or comparing magnetometer readings to link devices in close proximity of a few centimeters [14]. Other solutions for co-presence detection require infrastructure support such as beacons for Bluetooth low energy (BLE) [15] or ultrasound [16] to emit messages to recognize user's co-presence. In contrast to our work, most of existing studies only analyze device-to-device proximity and not the proximity of device groups, without evaluating the impact of other factors, such as user mobility and hardware heterogeneity on proximity reasoning. The system in [17] is similar to our analysis considering a multitude of sensor data for group detection. However, their work lacks a direct comparison of different sensor data regarding proximity accuracy, e.g., whether the similarity of accelerometer data is higher compared to barometer pressure. We do not aim to mitigate context-manipulation attacks [18] which prevent co-presence detection or trick the proximity reasoning to include remote users into a group of nearby users.

III. Proximity Data Collection

In accordance with ethical requirements, we have undergone a standardized process via the data protection officer of the institute covering appropriate data protection and respecting user's privacy; we conveyed explicitly the purpose of data collection, the type of data gathered from different people, how and where the data is stored, and who would process and use the data. Finally, our data collection is built on participants who approved our privacy agreement to join the study.

A. Sensing Framework

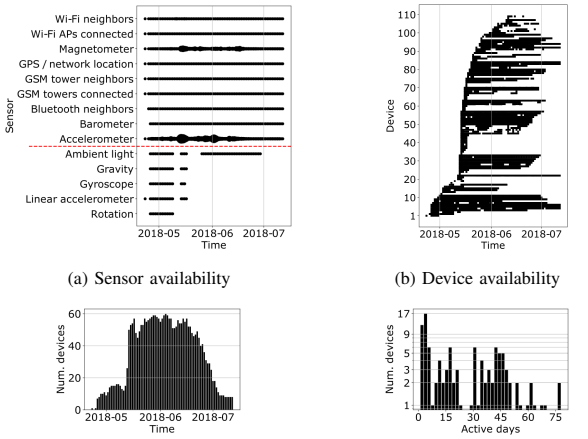
We used the AWARE framework [19] (version 4.0.708) in a client and server setting to gather sensor data from mobile devices. The AWARE server runs on a Linux server located at the department with Apache, MySQL database, PHP, and a Mosquitto MQTT broker, which enables TLS transmissions between student's mobile devices and the AWARE server. On the client side, the AWARE app is available for Android and iOS devices. Table I shows the configured study data to be collected from the student's mobile devices. For the sensor sampling rates, we consider the trade-off between energy and memory consumption, and whether the sampled sensor data being usable for co-presence detection. Our proximity verification sets include GPS and network locations, and Bluetooth, Wi-Fi, and GSM encounters. To obtain a sufficiently fine-grained Bluetooth verification of user's proximity, we distributed 50 BLE beacons over the campus to cover main entrances, lecture halls, library, and cafeteria.

B. Dataset Preparation for Proximity Analysis

Our proximity data collection contains sensor data from 126 devices. We do not consider the following sensor data for co-presence detection due to limited (few users) sampling points: ambient light, ambient temperature, gravity, gyroscope, linear accelerometer, and rotation. This results in a cleaned proximity data collection including 110 mobile devices with 69 % Android devices and 31 % iOS devices. To reduce the demanded resources of I/O and CPU when using the gathered data, we convert the raw data stored in MySQL dumps (122 GiB) to Apache Parquet (23 GiB).

TABLE I
Overview of Study Dataset

Data characteristic	Sampling rate	Study data
User activity	5 s	(Linear) accelerometer
User position	5 min	GPS, network
User environment	5 min	Barometer, magnetometer, temperature, light, gravity, gyroscope, rotation, GSM towers, Bluetooth and Wi-Fi devices



(a) Sensor availability

(b) Device availability

(c) Number of active devices

(d) Histogram of active days

Fig. 1. Overview of proximity data collection

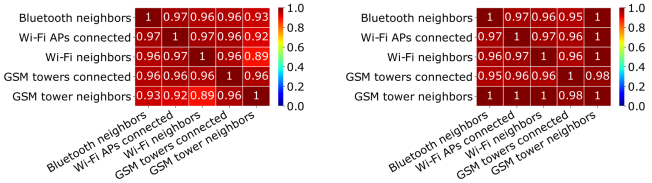
C. Overview of Proximity Data Collection

As our co-presence detection is dependent on sensor data sensed from multiple devices at the same time and place, Fig. 1(a) proofs that most of the desired sensor data are evenly distributed over time being able to infer user's proximity. Additionally, Figs. 1(b) and 1(c) confirm that we have enough active users who contributed sensor data over a longer period. With respect to the possible size of device groups, Fig. 1(d) presents the number of active devices broken down by days of collected data. We select the following wireless and sensor data targeted for proximity reasoning: Wi-Fi access points (APs) connected, Wi-Fi neighbors, Bluetooth neighbors, GSM towers connected, GSM tower neighbors, GPS and network location, accelerometer, barometer, and magnetometer.

IV. Verification Sets for Proximity Reasoning

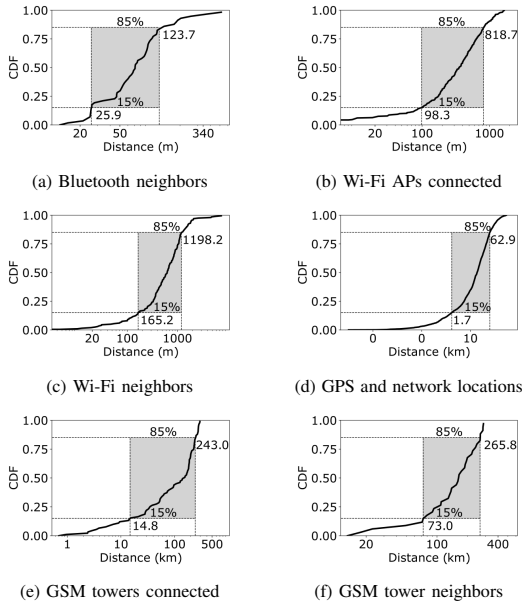
A. Performance Comparison of Proximity Verification Sets

We use several proximity verification sets to check the spatial adjacency of device groups inferred by sensor data,



(a) Close-by devices (b) Remote devices

Fig. 2. Pearson correlation among different proximity verification sets for performance comparison of co-presence detection



(a) Bluetooth neighbors (b) Wi-Fi APs connected

(c) Wi-Fi neighbors (d) GPS and network locations

(e) GSM towers connected (f) GSM tower neighbors

Fig. 3. Spatial granularity of proximity verification sets based on users' encounters of location-tagged wireless devices

such as magnetometer. To be able to compare the verified co-presence results deduced from sensor data, we analyze that each proximity verification set achieves a similar performance in detecting device groups. Hence, we compute the correlation of verified device groups based on Bluetooth neighbors, Wi-Fi APs connected, Wi-Fi neighbors, GSM towers connected, and GSM tower neighbors as shown in Fig. 2. By using a moving time window of two hours over the proximity verification sets, we limit the runtime and identify close-by devices for each proximity verification set via encountered Bluetooth devices, Wi-Fi APs, or GSM cell towers. Afterwards, we remove duplicate device groups and determine remote devices. We compute the mean Pearson correlation for both device groups: close-by and remote devices, if we recognized device groups from at least two different proximity verification sets for each time window, e.g., Wi-Fi and GSM neighbors.

For devices in proximity, Fig. 2(a) shows the mean Pearson correlation between different proximity verification sets. The correlation is evenly distributed over all verification sets and ranges between 0.89 and 0.97. Fig. 2(b) presents a similar result with a slightly increased correlation among proximity verification sets for remote devices ranging between 0.95 and 1. Additionally, the mean group size of nearby devices amounts to four devices compared to distant device groups with 11 devices.

B. Spatial Granularity of Proximity Verification Sets

To be able to associate sensor's data with a spatial granularity for co-presence detection, we compute the geographic expansion of each proximity verification set by using the user's daily moving distance based on encountered location-tagged wireless devices. Therefore, we manually link our self-distributed BLE beacons with latitude and longitude coordinates. For the positions of the Wi-Fi APs, we use a list from our IT department with the MAC address of each Wi-Fi access point and the nearest room number; our institution's room finder provides latitude and longitude for each room on campus. For the positions of the GSM cell towers, we use a publicly available dataset[1]. Based on this, we group the data of each proximity verification set after user devices joining the positions of encountered BLE beacons, Wi-Fi APs, or GSM cell towers. To achieve a more accurate user's daily moving distance, we resample the scans of surrounding Bluetooth devices, Wi-Fi APs, and GSM cell towers by using the median scan period and we take the mean of user positions for each scan. Finally, to get a user's daily moving distance, we sum over the geodesic distances for a series of user positions. Fig. 3 presents the ascending spatial granularity of different proximity verification sets, among 15 % and 85 % of all users.

[1]https://www.opencellid.org

V. Evaluation of Multimodal Group Proximity

Our aim is to analyze whether sensor modalities are effective for co-presence detection by using proximity verification sets with a variety of spatial granularity, such as Bluetooth, Wi-Fi, and GSM tower neighbors. Our sensor data (barometer, magnetometer, and accelerometer) is dependent on the user's location or activity with the assumption that people share the same context similarity. We define the verification of users' proximity in the following way: at least two or more user devices sense the same wireless device, e.g., Bluetooth device, Wi-Fi access point, and GSM cell tower, within a limited time period like 10 min. To identify remote devices which are not close to each other, we take the difference among all devices and nearby devices within the proximity period. To ensure timely aligned sensor's data across user devices for co-presence detection, our sensing framework performs the clock drift correction of user devices during the daily data upload to the data collection server.

A. Time Periods of Device Encounters for Proximity Detection

As prerequisite for a meaningful co-presence detection, we identify the best encounter times in terms of most devices in proximity, a sufficient number of remote devices, and the largest set with same sensed sensor data across user devices. To compute the aforementioned proximity statistics to select

TABLE II
FOR EACH VERIFICATION SET WE IDENTIFY THE MOST EFFECTIVE SENSOR MODALITY TO DETECT CO-PRESENCE FULFILLING DIFFERENT SPATIO-TEMPORAL GRANULARITY

Verification set	Sensor data	Spatial granularity	Proximity period	Proximity signal distance $\bar{\delta}_p$	Non-proximity signal distance $\bar{\delta}_{np}$	Signal distance ratio $\bar{\delta}_{np}/\bar{\delta}_p$
Bluetooth neighbors	Accelerometer	26–124 m	25 min	992.9	1888.5	1.9
Wi-Fi APs connected	Magnetometer	98–819 m	20 min	652.9	1455.8	2.2
Wi-Fi neighbors	Barometer	165 m–1.2 km	30 min	0.7	13.2	18.3
GSM towers connected	Magnetometer	15–243 km	15 min	772.5	2181.8	2.8
GSM tower neighbors	Magnetometer	73–266 km	30 min	2906.3	9353.8	3.2

the best encounter times for co-presence detection, we use a moving non-overlapping time window of two hours for each combination of proximity verification set, sensor data, user groups with different mobility, and device groups with varying sensor hardware. Given the dataset diversity and the nature of proximity detection over short periods, we choose an empirical two hour time window to strike a balance between granularity and fidelity, comparing with the proximity time window of 5–30 minutes.

B. Results of Co-Presence Detection

We perform our co-presence detection for the best encounter times of 55 different parameter sets, including proximity verification sets, sensor data, user groups, and device groups. We use multiple proximity periods $\in$ [5, 10, 15, 20, 25, 30] min to evaluate the time granularity of user's co-presence. To verify the user's proximity, two user devices have to encounter the same wireless device, e.g., access point or BLE beacon, within the proximity time window. We aim at recent proximity encounters and hence set the time range to 5–30 min.

We use the dynamic time warping distance named δ to compute the similarity of sensor data across different user devices. For comparison, we calculate the signal similarity within each group of devices in proximity, defined by the proximity verification set and between each device in proximity and all distant devices. We take the mean of signal distances among close-by devices $\bar{\delta}_p$ as well as between remote devices $\bar{\delta}_{np}$. Our assumption is that the signal similarity among close-by devices is higher compared to that of remote devices. We use the raw sensor signal to evaluate the basic performance of our co-presence detection.

As a result, based on the maximum signal distance ratio $\bar{\delta}_{np}/\bar{\delta}_p$ between the proximity $\bar{\delta}_p$ and non-proximity signal distance $\bar{\delta}_{np}$, Table II presents the most effective sensor modality and proximity period for each proximity verification set with a varying spatial granularity. A signal distance ratio of one means no difference in the signal similarity among nearby and remote devices, hence co-presence detection is not possible. The larger the signal distance ratio the better for proximity reasoning. The proximity period shows how much time elapses before we can infer the most effective co-presence detection. In addition, we are able to associate sensor's data with a spatial granularity and proximity period as shown in Table III, given by the maximum signal distance ratio.

TABLE III
SPATIAL GRANULARITY AND PROXIMITY PERIOD FOR EACH SENSOR MODALITY

Sensor data	Spatial granularity	Proximity period	Proximity signal distance $\bar{\delta}_p$	Non-proximity signal distance $\bar{\delta}_{np}$	Signal distance ratio $\bar{\delta}_{np}/\bar{\delta}_p$
Accelerometer	26–124 m	25 min	992.9	1888.5	1.9
Barometer	165 m–1.2 km	30 min	0.7	13.2	18.3
Magnetometer	165 m–1.2 km	30 min	4320.1	28337.9	6.6

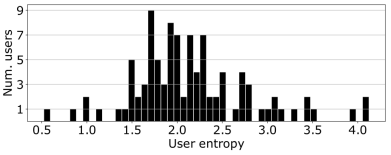


Fig. 4. Distribution of user entropy based on positions from encountered and connected Wi-Fi access points

To sum up, we see a clear distinction of signal similarity among nearby and distant devices, allowing for co-presence detection with a varying spatio-temporal granularity. Via the identified spatial granularity and proximity period, depending on the use case, developers of context-aware applications are able to choose the appropriate wireless or sensor data for co-presence detection. For instance, the magnetometer data offers the most diverse spatial granularity of user's co-presence from a few hundred meters up to kilometers, compared to the accelerometer data with a working range between 30–100 m.

C. Impact of User Mobility on Co-Presence Detection

To evaluate the impact of user mobility on co-presence detection, we compute the mean entropy per user based on positions of encountered wireless devices (e.g., BLE beacons and Wi-Fi APs) or directly via sensed GPS and network positions. We use the entropy to define the user mobility via random device encounters at different locations covering both movement patterns and variability; this is better than the user's

TABLE IV
IMPACT OF USER MOBILITY ON CO-PRESENCE DETECTION

Sensor data	User entropy	Mean proximity period	Mean signal distance ratio $\bar{\delta}_{np}/\bar{\delta}_p$
Accelerometer	1.9	19.2 min	1.3
	2.2	21.2 min	1.6
	3.1	17.5 min	1.3
Barometer	1.9	17.5 min	9.3
	2.2	17.5 min	9.5
	3.1	–	–
Magnetometer	1.9	18.8 min	3
	2.2	18.1 min	1.9
	3.1	21.9 min	2.1

daily moving distance, which neglects the randomness of user behavior. For instance, a user moves several hundred meters each day but only between two positions resulting in a higher moving distance but low entropy. The randomness of user's mobility is more crucial for co-presence detection.

We compute the user entropy for each proximity verification set based on users' encounters of location-tagged wireless devices and apply x-means clustering to find user groups with different mobility behavior. We select user groups inferred by the connected Wi-Fi APs because we can cover 88.2 % of all users, i.e., 97 of 110 users. Fig. 4 shows the distribution of user entropy including two user groups: 34 users with a mean entropy of 3.05 meaning high mobility and 63 users with a mean entropy of 1.86 meaning low mobility. For comparison, we treat all users as a third binned user group with a mean entropy of 2.16 meaning medium mobility.

For each sensor, Table IV shows the impact of user mobility on co-presence detection using three user groups with different mobility behavior. For the barometer sensor, the user group with the highest mobility entropy was too sparse and no proximity encounters could be found. The user entropy, reflecting the users' mobility, has only a minor impact on the mean proximity period and mean signal distance ratio of each user group. For example, there is no trend wherein users with a higher mobility have more or less encounters with other users compared to less randomly moving users.

D. Impact of Device Heterogeneity on Co-Presence Detection

We present a two-fold analysis of device heterogeneity, including sensor hardware statistics and quantifying the effect of different sensing ranges and sensitivities of device sensors on our co-presence detection using sensor's signal similarity.

Regarding the diversity of mobile device's sensor hardware, Fig. 5(a) illustrates that 70 % of all device sensors are produced by only three vendors and Fig. 5(b) shows per sensor that only 30 % of all device sensors are unique and only 10 % of all device sensors are from different vendors. In more detail, Fig. 5(c) presents sensor components and vendors for each sensor, on average, we have 17 unique sensors from six vendors. Many mobile devices from different manufacturers are using the same sensor hardware. This leads to a reduced impact of device heterogeneity on our co-presence detection.

TABLE V
IMPACT OF DEVICE DIVERSITY ON CO-PRESENCE DETECTION

Sensor data	Sensor hardware	Mean proximity period	Mean signal distance ratio $\bar{\delta}_{np}/\bar{\delta}_p$
Accelerometer	mixed	16.9 min	1.2
	same	22.2 min	1.5
Barometer	mixed	17.5 min	2.9
	same	17.5 min	15.9
Magnetometer	mixed	21.9 min	1.7
	same	17.9 min	2.8

Besides that, we quantify the impact of device heterogeneity on our co-presence detection. To this end, we enrich nearby devices defined by our proximity verification set with sensor names or device models, in case of missing hardware information. Different device models like iPhone 6, iPhone 6s, and iPhone 6s Plus are handled as one device model because they use the same sensor hardware[2]. We split the nearby devices according to their sensor hardware or device model to achieve a potentially higher signal similarity among devices in proximity using only the same sensor hardware. We treat remote devices as one device group regardless of their sensor hardware. Table V presents the impact of device diversity on co-presence detection. Per device group, the mean proximity period remains the same across different sensor hardware. In contrast, the mean signal distance ratio increases over device groups separating close-by and distant devices, as expected, if we are only using devices with the same sensor hardware for each proximity group. The co-presence detection with accelerometer data slightly improves by 1.25x, similarly to magnetometer data with 1.6x. The barometer data achieves the greatest improvement with a 5.5x greater signal distance ratio, compared to mixed sensor hardware for co-presence detection.

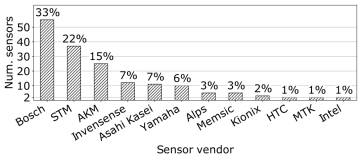
VI. ENERGY ANALYSIS FOR CO-PRESENCE DETECTION

With respect to usability, we analyze the impact of the energy consumption of different device sensors for proximity sensing on the limited battery capacity of mobile devices.

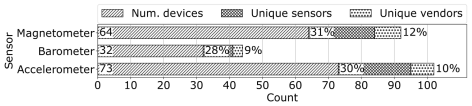
A. Testbed for Sensor Energy Measurements

Our proximity dataset contains five Samsung Galaxy S5 devices (model: SM-G900F). As a sampling device for our sensor energy measurements we use the Samsung Galaxy S5 with Android 6.0.1, in which we replaced the detachable battery with a Monsoon high-voltage power monitor. The Monsoon device directly powers the smartphone with 3.85 V and we take the energy measurements, e.g., time, voltage, and current, via the Python library of the Monsoon power monitor.

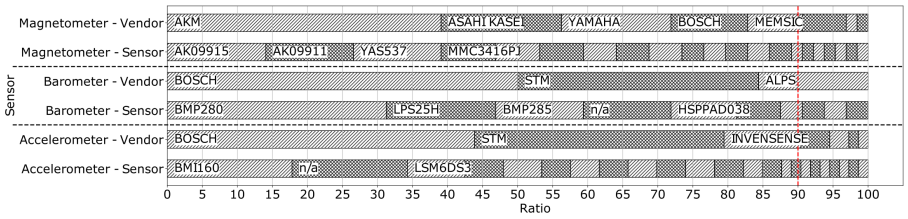
[2]https://www.ifixit.com/Teardown



(a) Ratio of sensor vendors over all devices



(b) Ratio of unique sensor components and vendors per sensor over all devices



(c) Sensor components and vendors per sensor over all devices

Fig. 5. Sensor hardware statistics for device heterogeneity

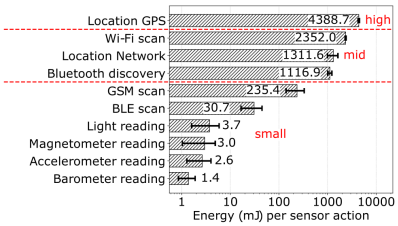
To compute the energy consumption of each device sensor, our Android test application performs different sensor actions, e.g., magnetometer reading and Wi-Fi scan, during the energy measurements lasting one minute in each of our ten evaluation rounds. To purely compute a sensor's energy, we measure the energy used by the smartphone's idle (with disabled wireless connections, including GPS, Wi-Fi, Bluetooth, and GSM) and for a specific sensor's energy we only activate the corresponding wireless interface. For instance, we only activate the Bluetooth interface for Bluetooth discovery or BLE scan. We take as many sensor readings (n_{sensor}) as possible and count them to normalize the consumed energy, resulting in a time-independent energy scale in mJ for each sensor action. The default sampling rate is 10 s or as fast as the sensor is able to provide the information, e.g., GPS location every 30 s.

B. Impact of Sensor Energy Consumption for User's Proximity

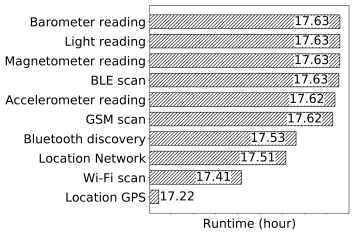
For our energy measurements we record the time, voltage, and current to compute the energy consumption $E(mJ) = U(V) \cdot I(mA) \cdot t(s) = \sum_{i=1}^{n} U_{t_i} \cdot I_{t_i} \cdot (t_{i+1} - t_i)$. We show the energy consumption for each sensor action E_{sa} in Fig. 6(a), defined by $E_{sa} = {E_{\text{sensor}} - E_{\text{system idle}}}/{n_{\text{sensor}}}$. We apply k-means with three clusters on the median energy consumption for each sensor action, to classify the different sensor actions into three energy consumption levels: small with 34.2 mJ, medium with 1558.5 mJ, and high with 4341.9 mJ as shown in Fig. 6(a). For location requests the network provider determines the device location based on cell towers and Wi-Fi APs, whereas the GPS provider determines the device location using satellites.

We aim to highlight the effect of the sensor's standalone energy consumption on the limited battery capacity of mobile devices. Therefore, we compute the smartphone's runtime t_s using different device sensors for proximity detection as in

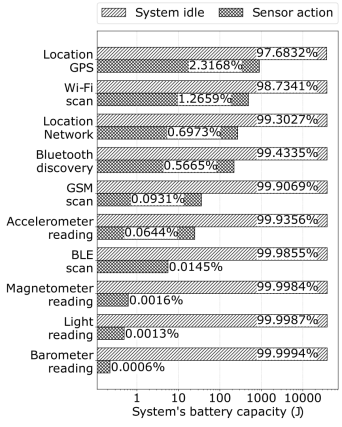
$$
\begin{aligned}
E_{\text{battery}} &= E_{\text{system idle}} + E_{\text{sensor}} \\
&= U \cdot I \cdot t_s + E_{sa} \cdot {t_s}/{s_r} \\
\rightarrow t_s &= \frac{E_{\text{battery}} \cdot s_r}{U \cdot I \cdot s_r + E_{sa}}
\end{aligned}
$$



(a) Standalone sensor's energy consumption with energy levels*



(b) Smartphone runtime using different sensors



(c) Energy ratio between smartphone's system idle and sensor action*

Fig. 6. Energy analysis for co-presence detection (*logarithmic scale)

where E_{sa} is the energy consumption for each sensor action from Fig. 6(a) and s_r is the sampling rate of each device sensor for our proximity data collection in Table I. We use the battery capacity of the Samsung Galaxy S5 with $E_{\text{battery}} = 10.78\,Wh$, defined by $E(Wh) = {}^{Q(mAh)\cdot U(V)}/_{1000}$, electric charge defined as $Q = 2800\,mAh$. For voltage U and current I we take the mean voltage and current from our energy measurements. The battery life of the smartphone in idle state without sensor actions and disabled wireless interfaces is 17.63 hours. Fig. 6(b) shows that the different sensor actions have only a minor effect on the smartphone's runtime t_s. For instance, the standalone sensor's energy consumption to receive GPS locations is 3193x higher compared to barometer readings, whereas the smartphone's runtime only decreases by 25 minutes. Hence, we analyze the ratio of energy consumption between the system idle and sensor actions. Fig. 6(c) highlights that the system idle dominates the total energy consumption of the smartphone with $\approx 98\,\%$. We cannot recognize an effect on the energy relation between system idle and sensor actions if the smartphone is moving or not.

VII. Conclusion

Our study on co-presence detection focuses on using wireless and sensor data from mobile devices. For proximity sensing, we collected a multimodal dataset from 126 participants over three months. We associate the collected data with an effective spatial and time granularity and we identify which sensor data from mobile devices is appropriate for proximity detection. Furthermore, we show that user mobility has only a minor impact on the proximity reasoning and that the device heterogeneity with diverse sensor hardware heavily affects the co-presence detection. Finally, we have conducted an energy analysis of different device sensors for proximity detection. The idle system consumes the most battery capacity of the mobile device while the effect of sensor reading is negligible.

For future work, we plan to further analyze the timely performance of co-presence detection throughout the day, e.g., morning, noon, and evening, and how the user activity, e.g., standing, sitting, and moving, affects the proximity reasoning. Moreover, we plan to enrich our co-presence detection by estimating the social relation among users. For the ground truth of social relationships, we have conducted a survey among the participants of our data collection including the type of relationship, presented as ranked categories, e.g., friend, classmate, and stranger. Our aim is to better understand the social dynamics of a group of people related to proximity, e.g., how the strength of social ties correlates with the spatial adjacency.

Acknowledgment

The authors would like to thank Georg Groh and the study participants for their support in making this work possible. In addition, we thank the reviewers and Leonardo Tonetto for their meaningful feedback.

References

[1] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, "Measurement of Proximity," in *Cluster Analysis*, ser. Wiley Series in Probability and Statistics. Wiley, 2011, pp. 43–69.

[2] 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 *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing (HUC)*, 1999, pp. 304–307.

[3] A. Pyattaev, K. Johnsson, S. Andreev, and Y. Koucheryavy, "Proximity-Based Data Offloading via Network Assisted Device-to-Device Communications," in *Proceedings of the IEEE 77th Vehicular Technology Conference (VTC Spring)*, 2013, pp. 1–5.

[4] J. Li, J. Jannotti, De Couto, Douglas S. J., D. R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2000, pp. 120–130.

[5] Y. Wang, A. V. Vasilakos, Q. Jin, and J. Ma, "Survey on Mobile Social Networking in Proximity (MSNP): Approaches, Challenges and Architecture," *Wireless Networks*, vol. 20, no. 6, pp. 1295–1311, 2014.

[6] M. Haus, A. Y. Ding, and J. Ott. (2020) Proximityness. [Online]. Available: http://proximity.cm.in.tum.de

[7] A. Ghose, C. Bhaumik, and T. Chakravarty, "BlueEye: A System for Proximity Detection using Bluetooth on Mobile Phones," in *Proceedings of the ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication (UbiComp Adjunct)*, 2013, pp. 1135–1142.

[8] P. Sapiezynski, A. Stopczynski, D. Kofoed Wind, J. Leskovec, and S. Lehmann, "Inferring Person-to-person Proximity Using WiFi Signals," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 1–20, 2017.

[9] W.-T. Tan, M. Baker, B. Lee, and R. Samadani, "The Sound of Silence," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2013, pp. 1–14.

[10] M. Maier, C. Marouane, M. Klette, F. Dorfmeister, P. Marcus, and C. Linnhoff-Popien, "SURFtogether: Towards Context Proximity Detection Using Visual Features," in *Proceedings of the 3rd International Conference on Context-Aware Systems and Applications (ICCASA)*, 2014, pp. 86–91.

[11] K. A. Nguyen, R. N. Akram, K. Markantonakis, Z. Luo, and C. Watkins, "Location Tracking Using Smartphone Accelerometer and Magnetometer Traces," in *Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES)*, 2019, pp. 1–9.

[12] S. Liu, Y. Jiang, and A. Striegel, "Face-to-Face Proximity Estimation Using Bluetooth On Smartphones," *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 811–823, 2014.

[13] C. Martella, A. Miraglia, M. Cattani, and M. van Steen, "Leveraging Proximity Sensing to Mine the Behavior of Museum Visitors," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2016, pp. 1–9.

[14] R. Jin, L. Shi, K. Zeng, A. Pande, and P. Mohapatra, "MagPairing: Pairing Smartphones in Close Proximity Using Magnetometers," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1306–1320, 2016.

[15] P. C. Ng, J. She, and S. Park, "High Resolution Beacon-Based Proximity Detection for Dense Deployment," *IEEE Transactions on Mobile Computing*, vol. 17, no. 6, pp. 1369–1382, 2018.

[16] B. Thiel, K. Kloch, and P. Lukowicz, "Sound-based Proximity Detection with Mobile Phones," in *Proceedings of the 3rd International Workshop on Sensing Applications on Mobile Phones (PhoneSense)*, 2012, pp. 1–4.

[17] A. A. de Freitas and A. K. Dey, "Using Multiple Contexts to Detect and Form Opportunistic Groups," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW)*, 2015, pp. 1612–1621.

[18] H. T. T. Truong, J. Toivonen, T. D. Nguyen, C. Soriente, S. Tarkoma, and N. Asokan, "DoubleEcho: Mitigating Context-Manipulation Attacks in Copresence Verification," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2019.

[19] D. Ferreira, V. Kostakos, and A. K. Dey, "AWARE: Mobile Context Instrumentation Framework," *Frontiers in ICT*, vol. 2, pp. 1–9, 2015.

Publication 5

M. Haus, A. Y. Ding, Q. Wang, J. Toivonen, L. Tonetto, S. Tarkoma, and J. Ott. Enhancing Indoor IoT Communication with Visible Light and Ultrasound. In *Proceedings of the 53rd IEEE International Conference on Communications (ICC)*, pages 1–6, 2019. doi:10.1109/icc.2019.8762001

This thesis includes the accepted version of our article and not the final published version.

Publication Summary

Over the time multiple wireless technologies have been developed to enable large scale communications for indoor environments. An important domain for IoT involving lightweight sensors, smart boards, and user's mobile devices. We analyzed and compared different wireless communication technologies in terms of their suitability for indoor IoT communication, e.g., maximum transmission distance, maximum data rate. Thereby, we identified visible light communication (VLC) and ultrasound communication as promising candidates to further enrich the ecosystem for indoor IoT communication. Use cases dedicated for VLC and ultrasound communication include indoor localization and gesture recognition. We have developed two communication modules using off-the-shelf components for visible light and ultrasound communication and evaluated their network performance and energy consumption. In addition, we show the efficacy of our communication modules by applying them in a practical indoor IoT scenario to realize secure IoT group communication sharing distance-bounded information among nearby devices. Our prototype runs on off-the-shelf Android smartphones which are able to collect input data from Wi-Fi, ambient sound, VLC, and ultrasound. In the beginning, each device advertises via Wi-Fi Direct its CPU utilization, available battery power, and memory. On this basis, our prototype chooses the most powerful device as the supernode handling the device grouping. A device can be part of the group communication, if the ambient sound and Wi-Fi traces among the peers is similar or it is able to receive the distance-limited data transmitted via VLC or ultrasound. Besides that, we proposed the MEC^2-Hub for infrastructure-supported device grouping. We aim to enhance the integration of emerging communication technologies, such as visible light and ultrasound, with radio-based communication, like Wi-Fi or Bluetooth. MEC^2-Hub extends the idea of multipath protocols, such as multipath TCP (MPTCP), to realize multiple communication paths via different communication media. Each network subflow in MEC^2-Hub can use a combination of physical transmission medium, e.g., visible light or ultrasound, with different properties regarding transmission range and data rate.

Enhancing Indoor IoT Communication with Visible Light and Ultrasound

Michael Haus[†], Aaron Yi Ding[‡], Qing Wang[††], Juhani Toivonen[*], Leonardo Tonetto[†], Sasu Tarkoma[*], Jörg Ott[†]

[†]Department of Computer Science, Technical University of Munich, Germany
[‡]Department of Engineering Systems and Services, Delft University of Technology, Netherlands
[††]Department of Electrical Engineering, KU Leuven, Belgium
[*]Department of Computer Science, University of Helsinki, Finland

***Abstract*—The number of deployed Internet of Things (IoT) devices is steadily increasing to manage and interact with community assets of smart cities, such as transportation systems and power plants. This may lead to degraded network performance due to the growing amount of network traffic and connections generated by various IoT devices. To tackle these issues, one promising direction is to leverage the physical proximity of communicating devices and inter-device communication to achieve low latency, bandwidth efficiency, and resilient services. In this work, we aim at enhancing the performance of indoor IoT communication (e.g., smart homes, SOHO) by taking advantage of emerging technologies such as visible light and ultrasound. This approach increases the network capacity, robustness of network connections across IoT devices, and provides efficient means to enable distance-bounding services. We have developed communication modules using off-the-shelf components for visible light and ultrasound and evaluate their network performance and energy consumption. In addition, we show the efficacy of our communication modules by applying them in a practical indoor IoT scenario to realize secure IoT group communication.**

***Index Terms*—IoT, Visible light, Ultrasound, Multi-access, Edge computing, Proximity-aware device grouping**

I. Introduction

The demands for network capacity are steadily increasing due to the dense deployment of connected devices. For instance, almost half a billion mobile devices were added globally in 2016 and the global mobile data traffic is estimated to increase sevenfold between 2016 and 2021 [1]. Emerging applications such as VR/AR are demanding low latency and high computing capabilities for real-time interactions. In this respect, edge computing is one important development, which leverages the physical proximity of communicating devices to establish short communication paths. The edge approach offers the following network properties: high throughput, low latency, and reliability, all leading to an improved service completion time [2]. To realize resilient services, approaches like Wi-Fi HaLow, LoRa, SigFox, and NB-IOT address special requirements of IoT communications such as massive connectivity, frequent and small amount of transmitted data. In our context, IoT communication includes typical lightweight sensors, programmable boards, and user's mobile devices like smartphones or tablets. One major problem is how to scale communication over the limited wireless spectrum. Wi-Fi and Bluetooth often interfere with each other in densely deployed IoT networks. We can utilize emerging communication mechanisms such as Visible Light Communication (VLC) and ultrasound to bypass wireless interference. Combined with a smart IoT device management platform [3], we can orchestrate different IoT and edge devices to fully leverage wireless technologies. For instance, when detecting jamming condition of Wi-Fi channels, switch to VLC for data transmission. Thereby, we are able to enhance network performance and save energy by avoiding redundant transmissions.

A unique property of VLC and ultrasound is that the communication range is naturally restricted by territorial obstacles, thus providing the basis for distance-bounding services. A distance-bounding service ensures an upper distance limit between sender and receiver. For example, seamless car entry systems verify if the car's key is within a certain distance, otherwise the doors cannot be opened and the engine cannot be started. In contrast, mid-range radio-based communications like Bluetooth or Wi-Fi cause additional overhead to measure the round trip time between sender and receiver and estimate the distance between them. Due to the limited communication distance, visible light and ultrasound can help enhancing privacy and security of IoT communications where their data exchange can be easily restricted through obstacles like doors, walls, and windows. Radio waves penetrate such spatial barriers and are hence exposed to eavesdropping and interception attacks. From a deployability perspective, ultrasound is easy to deploy and flexible owing to wide support by off-the-shelf smartphones. VLC has also seen significant advances such as the open-source platform OpenVLC [4].

In this work, we exploit emerging communication technologies, VLC, and ultrasound, to utilize the advantages of different electromagnetic spectrum for enhancing indoor IoT communication. In Section II, we analyze user mobility in terms of required transmission distance and compare different wireless communication technologies regarding their suitability for indoor IoT communication. In addition, we highlight use cases for VLC and ultrasound communication in Section III. Besides that, Section IV provides details of our VLC and ultrasound communication modules and we evaluate Wi-

This work was supported by the TUM Living Lab Connected Mobility Project, the Bavarian Ministry of Economic Affairs and Media, Energy and Technology (StMWi) through the Center Digitisation, Bavaria, and in part by the Academy of Finland grant number 314008.

Fi, Bluetooth, VLC, and ultrasound in terms of transmission distance, data rate, and energy consumption. In Section V, we implement a secure group communication service using VLC and ultrasound to share distance-bounded information among proximate devices. Section VI highlights open questions for future research.

We summarize our contributions as follows:

1) We explore the feasibility of two non-radio based communications, VLC and ultrasound, to support indoor IoT communication.
2) We develop communication modules for VLC and ultrasound and evaluate the prototypes with respect to communication distance, data rate, and energy consumption.
3) We apply our VLC and ultrasound modules to realize secure group communication with an automated key management. This service prototype illustrates a pragmatic use case in augmenting IoT services.

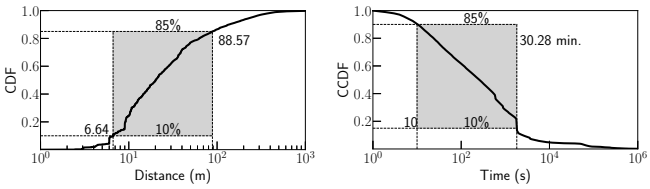
II. Indoor IoT Communication

Indoor communication is an important domain for IoT where multiple wireless technologies have been developed to support large scale communications. We provide a brief overview of indoor IoT communication technologies like Wi-Fi, Bluetooth (BT), VLC, and ultrasound.

The frequency range of visible light, 430 THz to 790 THz, is 1200 times greater compared to the scope of electromagnetic waves with 3 Hz to 300 GHz. This may help solving the network capacity problem of wireless radio-based communications. Besides that, we take advantage of ultrasound by using sound waves between 20 kHz to 24 kHz, to transmit information between devices which is inaudible for humans and can be used as out-of-band channel. Wireless interference is another disadvantage for radio-based technologies, which can negatively affect the network performance. For example, in our testbed we observed a decrease of Wi-Fi throughput in presence of Bluetooth Low Energy (BLE) beacons by 12.12 % (16.89 MB/s without BLE, and 14.84 MB/s with BLE).

For practicality, we have analyzed the mobility of users, i.e., walking distance, to show whether VLC and ultrasound are suitable for indoor IoT communications in terms of viable communication range. The dataset [5] contains the associations between 6202 users and 500 Wi-Fi access points with relative positions within university buildings. To detect a user movement, we analyze whether the associations between user and access point changes over time. Fig. 1(a) shows the users' walking distance, ranging from 6.64 m (10 % of all users) to 88.57 m (85 % of all users). Regarding transmitted network data, another recent study analyzed the user's data consumption and revealed that 85 % of all users consume about 100 MB per day [6].

By comparing maximum transmission distance and data rate, as shown in Table I, we can indicate which communication technology is suitable for indoor IoT communications. Existing ultrasound prototypes using commercial off-the-shelf smartphones provide low bit rates. This greatly limits the possible use cases and hence ultrasound is most applicable as



(a) Users' walking distance (b) Users' connection time

Fig. 1: Analysis of user's mobility pattern to highlight appropriate wireless communications with respect to transmission distance and connection time

out-of-band channel, e.g., transmit encryption keys, but not for bulk data transmission. Meanwhile, VLC is a viable solution as it covers a broader range of user movements and its achievable data rate is sufficient for common IoT communication tasks.

The communication performance of visible light and ultrasound are mainly impacted by environmental conditions such as ambient light or ambient sound. As a distinctive attribute, the transmission range of those two emerging communication technologies is greatly limited by spatial barriers such as doors, walls, and windows. This makes it appropriate for distance-bounding services without additional computation overhead like with radio-based communication.

III. Use Cases for VLC and Ultrasound

VLC has been enabling many applications related to IoT, such as accurate indoor localization [9], human sensing, encounter detection [10], gesture recognition, and so on. Since visible light does not pass through opaque objects, it is a good candidate to realize distance-bounding wireless communication to improve its security performance. Therefore, it can be used in many potential applications, especially those that require close interaction. For example, convenient and secure payment in supermarkets (no need to approach close to the reader to "touch" it for payment, which is required with NFC in order to ensure security) and robots control in smart factories (robots are allowed to access some resources through interactions only if they are physically within the delimited distance).

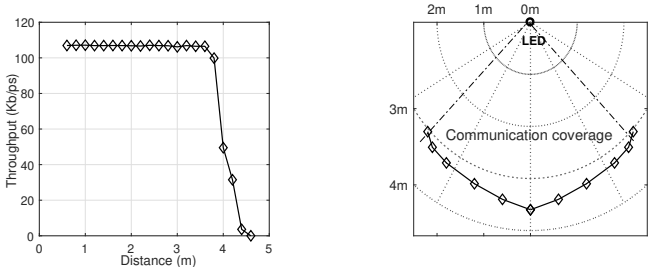
Ultrasound supports a range of use cases including device pairing, proximity detection, user-tailored advertisements or as mobile payment system in taxis. In case of automated device grouping and device pairing [11], [12], the speaker emits inaudible tones which are captured only by physically proximate devices. For instance, to organize group activities, e.g., a meeting or to share documents with its members. Besides that, ultrasound is widely used for proximity marketing [13]. In environments like casinos, museums, retail, airports, the user gets location-tailored advertisement based on user tracking. In shopping malls, stores track the in-store user behavior.

IV. Communication Modules and Evaluation

We use non-radio technologies such as VLC and ultrasound to supplement and enrich conventional radio-based communication for IoT communication. Our communication modules

TABLE I: Comparison of communication technologies for indoor IoT communications [7], [8]

Communication Technology	Max. transmission distance	Max. data rate	Influence factors	Advantages
Wi-Fi	100 m	7 Gbit/s	Interference with other radio-based technologies	Unlicensed spectrum allows cost-efficient implementation
Bluetooth	100 m	24 Mbit/s	Manual pairing for device connection	Low power consumption
Visible Light	30 m	15 Gbit/s	Line of sight transmission	Privacy enhanced communication by distance restriction
Ultrasound	25 m	56 kbit/s	Low data rates and error prone decoding due to overlapping frequencies	Reliable mechanism for device grouping



(a) Throughput vs. distance (b) Communication coverage

Fig. 2: Experimental results of our VLC module

enable visible light- and ultrasound-supported applications. Based on live testbed experiments, we present our insights and evaluation results for these two modules in terms of throughput, transmission range, and energy consumption.

A. Visible Light Communication Module

Our VLC module is built around the low-cost platform BeagleBone Black (BBB) which costs around $60. We use a Philips 4.7 W LED as the transmitter which is powered by a 24 V DC voltage. The LED is disassembled by removing the AC-DC converter that can slow down the transition speed between ON and OFF states. We adopt an advanced Adaptive Multiple Pulse Position Modulation (AMPPM) [4] scheme at the transmitter that can support dimming, instead of simple On-Off-Keying (OOK) modulation. At the receiver, incoming light signals are first sensed by a photodiode (SFH206K) and then amplified by an amplifier (TLC237). Analog signals from the amplifier are converted to digital signals by the ADC (ADS7883) and then sampled by the BBB micro-controllers' Programmable Realtime Units (PRUs) for further computation.

The evaluation results of the throughput achieved at various distances between transmitter and receiver are shown in Fig. 2(a). The transmitter and receiver are aligned. We can observe that our low-end VLC system can work at a maximum communication distance of 3.7 m. It achieves a throughput of up to 107 kb/s which is enough for most of the IoT applications. In addition, we carry out experiments to test the VLC communication coverage and present the results in Fig. 2(b). We can observe that the communication range of VLC is limited, which can be well controlled by using different types of LEDs.

Comparing our testbed results with the higher VLC performance of 15 Gb/s indicated in Table I, the performance gap is caused by the different flavors of VLC platforms using a diverse range of hardware. In addition, the testbed setting in terms of distance range and intensity of ambient light affects the perceived throughput. Our VLC platform proves that even with an off-the-shelf IoT board and low cost LED transmitters, the performance of our VLC module still satisfies the throughput requirement of IoT applications. We note that the timing function provided in the Linux kernel limits the sampling rate which becomes a major bottleneck for our VLC module. To overcome the bottleneck and achieve a higher throughput (e.g., up to several Mb/s), we could use a dedicated field programmable gate array (FPGA) or a separate microcontroller to perform signal sampling. For instance, another VLC system [14] takes advantage of laser diodes and is able to achieve better utilization of the visible light spectrum, reaching a throughput of ~15 Gb/s.

B. Ultrasound Communication Module

To modulate ultrasound messages, we are using an Orthogonal Frequency-Division Multiplexed On-Off Keying (OFDM-OOK) scheme. Thereby, we use eight frequencies to address eight bits in a byte and one frequency for a parity check, encoding each bit in the byte in parallel to the same symbol. For each symbol we use a fixed duration of 46.4 ms (2048 samples at 44.1 kHz) and a guard interval of the same length between the symbols to prevent Inter-Symbol Interference (ISI). To define the start and end of the message, we use a preamble and postamble with all bits on and thrice the regular pulse length. To demodulate an ultrasound message, we need to:

1) convey synchronization via preamble and postamble of the message recording
2) perform a Short Time Fourier Transform (STFT) with a sample size matching the symbol length used for modulation
3) compute a signal threshold to differentiate between bit one and zero. Therefore, we inspect the amplitudes on the frequencies of interest in different samples and for each frequency separately.
4) extract the modulated byte sequence via computed signal threshold.

TABLE II: Evaluation results of Wi-Fi and Bluetooth compared to our communication modules including VLC and ultrasound

Communication Technology	Max. transmission distance	Max. data rate	Energy consumption
Wi-Fi	30 m	1.05 Mbit/s	sender: 3.26 µJ/Byte receiver: 8.72 µJ/Byte
Bluetooth	10 m	718.16 Kbit/s	sender: 3 µJ/Byte receiver: 4.81 µJ/Byte
Visible Light	4.5 m	500 Kbit/s	sender: 8.42 µJ/Byte receiver: 8.32 µJ/Byte
Ultrasound	50 cm	64 bit/s	sender: 25,530 µJ/Byte receiver: 31,834 µJ/Byte

In our experiments, we use commercial off-the-shelf smartphones without special audio hardware. Today's smartphones are equipped with speakers and microphones which are capable to produce and capture sound at frequencies up to 22 kHz – 24 kHz. We tested our ultrasound modulation on a pair of Lenovo Phab 2 Pro phablets and achieved bit rates of 64 bit/s with bit error rates of less than 3 % on a distance of 50 cm. To enhance demodulation robustness we use Reed-Solomon error correction. In comparison, related prototypes achieve bit rates between 8 bit/s and 1280 bit/s with a communication range from 5 cm to 25 m [15], [16].

The achieved bit rate of our ultrasound modulation is appropriate for use cases where small messages are exchanged over limited communication range as needed, for example by device pairing or key exchange protocols. The bit rate can be increased through specialized audio hardware, such as in literature [17], or through a choice of a different modulation. For an overview, the authors of [16] explored several data modulation techniques in terms of their capabilities and differences.

C. Evaluation

To highlight the usability of VLC and ultrasound in IoT environments, Table II shows the maximum transmission distance, data rate, and energy consumption for VLC and ultrasound compared to Wi-Fi and Bluetooth. For Wi-Fi energy measurements, we attached a Wi-Fi USB adapter and created an access point via hostapd to directly connect sender and receiver. The high voltage Monsoon power device measures the energy by powering our hardware platform (BeagleBone Black) with 5 V for VLC and Wi-Fi energy measurements. For ultrasound and Bluetooth, the energy measurements were taken from an Android smartphone with a detachable battery. To compute the energy measurements for Wi-Fi, Bluetooth, VLC, and ultrasound, we have taken the difference to the system's basis energy consumption, BeagleBone black and Android smartphone. During the data transmission, we measured the current (mA), power (mW) and voltage (V) and calculated the required energy in Joule per Byte. With respect to the results, Bluetooth provides the lowest energy consumption in contrast to ultrasound communication with the significantly highest energy consumption. The VLC sender requires 1.6 times more energy as the VLC receiver mainly caused by the high power LED at the sender side to transmit the encoded data via visible light. The energy consumption of VLC and ultrasound is significantly higher compared to Wi-Fi and Bluetooth, which is a drawback for IoT environments with many battery-powered devices. VLC and ultrasound prototypes with specialized hardware can overcome this problem by increased data rates and lower energy consumption.

V. Secure IoT Group Communication

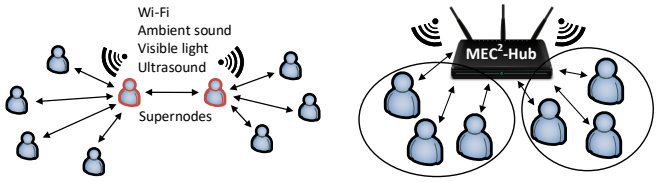
A. Mobile Device Grouping

To illustrate the usage of VLC and ultrasound in practice, we have developed a secure group communication protocol using our communication modules for proximity-aware device grouping. Fig. 3(a) illustrates the setting of our secure group communication solely based on mobile devices. We identify certain mobile devices, e.g., smartphones and tablets, as "supernodes" because of their stronger hardware performance compared to other nearby devices. To broadcast and receive VLC messages, we connect the mobile device via Wi-Fi to our VLC platform as add-on device mentioned in Section IV-A. As out-of-band channel, the supernode broadcasts messages or tokens via VLC and/or ultrasound which are used for device grouping and to secure the radio-based communication. Due to the limited VLC and ultrasound communication range, only mobile clients within a certain range are able to receive the broadcasted VLC and/or ultrasound message and hence are eligible to use the associated service, e.g., device grouping. By using these distance-limited token broadcasts, we are able to automate and ease the key management among IoT and mobile devices without user interactions like machine-to-machine communications. To refine the scope, our current prototype does not consider relay attacks. An adversary relays signaling data to a distant client which is then wrongly included into the device group.

We have implemented our automated device grouping on off-the-shelf Android smartphones, which can aggregate input data from Wi-Fi, ambient sound, VLC, and ultrasound. A device is eligible to participate in the group communication, if the ambient sound among the peers is similar or it is able to receive the data transmitted via VLC or ultrasound. Once the device grouping service is triggered, each device advertises via Wi-Fi Direct its CPU utilization, available battery power, and memory. On this basis, the most powerful device in proximity is selected as the supernode to handle the device grouping.

For Wi-Fi similarity, each device collects three Wi-Fi scans including SSID, BSSID, RSSI, and frequency. For ambient sound similarity, every device creates sound features from 10 s recordings of the ambient environment including: 1) power spectrogram to quantify changes in frequency, 2) Mel Frequency Cepstral Coefficients (MFCC) which mimics the human's perception, and 3) a landmark fingerprint [18] generated from most robust amplitude peaks. The supernode compares these Wi-Fi and ambient sound features for automated device grouping. During experiments in different environments, we have encountered the following settings as working best. For Wi-Fi similarity using the Pearson correlation with a similarity threshold of 0.74 and for ambient sound similarity using the landmark fingerprint with a hash-based offset similarity of 0.7. In addition, our prototype utilizes VLC and ultrasound for device grouping. The supernode broadcasts an ultrasound and VLC signal with an encoded identifier. We infer that a device is in vicinity to the supernode, if the normalized string similarity based on the Levenshtein edit distance between broadcasted word and decoded identifier is greater than 0.8. At least one proximity indicator, either VLC or ultrasound, has to be true to infer that the end device is in vicinity.

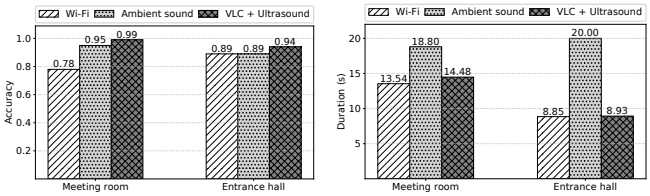
We have evaluated our prototype with off-the-shelf smartphones over ten evaluation rounds in two different testbeds. In each testbed, one closed and one open space, we placed two test devices within the proximity to each other and one device outside of the proximity range. The closed space refers to a meeting room with size of $4.5 \times 3.7 = 16.65\,\mathrm{m}^2$. The proximity is defined by the room boundaries, i.e., the device is within the room. For the second testbed, open space, we use the university entrance hall, which is crowded and noisy. In contrast to the closed environment, proximity is defined by a distance threshold of 5 m. In comparison to Wi-Fi similarity, Fig. 4(a) shows the accuracy of each device grouping mechanism in terms of correctly predicted devices in vicinity. In the closed space, i.e., meeting room, compared to the Wi-Fi based device grouping, using ambient sound achieves a 22 % higher accuracy and the combination of VLC and ultrasound communication performs 27 % better. In the open space, i.e., entrance hall, the proximity accuracy of ambient sound decreases by 6 % and the combination of VLC and ultrasound decreases by 5 %. Since the environment contains more disturbing noise which negatively affects the sound spectrum as proximity indicator. In contrast, the proximity accuracy using Wi-Fi features increases by 11 %. This indicates that Wi-Fi signals are preferably used as coarse-grained proximity indication. Besides that, Fig. 4(b) shows the duration until the devices are grouped together. Using ambient sound features for device grouping takes significantly longer compared to Wi-Fi and the combination of VLC and ultrasound communication which achieve similar results. To sum up, the combination of VLC and ultrasound communication for device grouping outperforms Wi-Fi and ambient sound based device grouping in terms of accuracy and duration.



(a) Device grouping solely based on user's mobile devices

(b) Device grouping supported by infrastructure

Fig. 3: Organization of IoT group communication



(a) Accuracy

(b) Duration

Fig. 4: Experimental results of our proximity-aware device grouping compared to using Wi-Fi

B. *Infrastructure-Supported Device Grouping*

The user's mobility mainly affects the deployment of the reasoning of our previously presented device grouping, e.g., on the mobile devices or access point. If the user is constantly moving the corresponding mobile device is frequently changing its access point. In this case, the device grouping as shown in Fig. 3(a) should be handled directly on the mobile devices. Fig. 1(b) shows the user's connection time to an access point which ranges from 10 min. (10 % of all users) to 30 min. (85 % of all users). Hence, the users are static enough that the device grouping can be offloaded to an access point as shown in Fig. 3(b). Therefore, we introduce our communication platform named MEC^2-Hub which supports multi-access mobile edge computing (MA-MEC) by exploiting the integration of emerging communication technologies such as visible light and ultrasound, together with radio-based communication like Wi-Fi or Bluetooth. MEC^2-Hub utilizes the advantages of different electromagnetic spectrum to realize services such as secure IoT group communication. MEC^2-Hub is intended to run at the edge of the network, such as wireless access points or gateways to enable edge communication paths. Fig. 5 shows our proposed platform which extends the idea of multipath protocols, such as multipath TCP (MPTCP) [19] to support multiple communication paths via different communication media. Each network subflow in MEC^2-Hub can use a combination of physical transmission medium, such as visible light or ultrasound with different properties regarding transmission range and data rate. The multipath protocols in MEC^2-Hub allow us to dynamically switch between network interfaces at runtime without reconnecting as the mobile device's IP address is decoupled from a specific network connection. The MEC^2 socket API is a major component in our platform allowing applications to interact with the MEC^2-Hub networking stack.

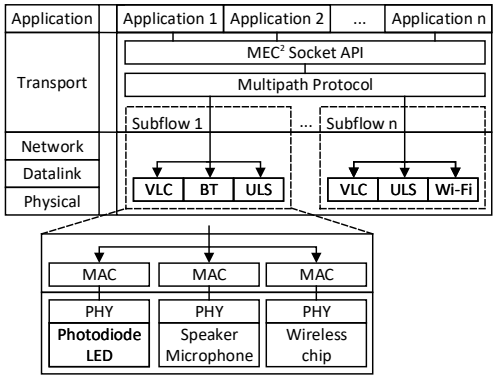


Fig. 5: MEC^2-Hub as communication platform for infrastructure-supported device grouping (ULS: ultrasound)

The underlying multipath protocols utilize feasible network paths via subflows for each network connection and distribute application data across those subflows.

VI. Open Questions and Challenges

VLC support for mobile devices. Existing VLC platforms require dedicated hardware boards. This greatly limits the flexibility in mobile environments. Meanwhile, most end-user devices such as smartphones are already equipped with the necessary hardware, i.e., photodiode for receiver and LED as transmitter. However, off-the-shelf devices lack support for real-time signal processing which is required for VLC. An improved support for VLC on off-the-shelf devices can greatly promote the adoption of VLC in the IoT domain.

Energy efficiency of VLC and ultrasound communications. To illustrate the impact, we have measured the power consumption of Bluetooth, Wi-Fi, VLC, and ultrasound. Comparing the energy consumption with Bluetooth, VLC consumes 124x more and ultrasound goes up to 7343x. For a better adoption of VLC and ultrasound in IoT domain, future research is needed to tackle the energy issue in VLC and ultrasound communications, spanning across hardware, protocol, and software implementations.

VII. Conclusion

Challenging requirements for indoor IoT communication include low latency, secure connectivity, and high reliability for a large number of heterogeneous IoT applications. To fulfill these requirements, we exploit two emerging communication technologies, visible light and ultrasound, and leverage their diverse electromagnetic spectrum to complement the conventional radio-based IoT communication. We have developed the communication modules and evaluated them in testbed environments. Our experimental study sheds light on how to apply those technologies in practice and illustrates pragmatic use cases to augment various IoT services. To demonstrate the efficacy of our approach, we further implement a practical service on off-the-shelf devices for securing IoT group communication.

References

[1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021," March 28, 2017.

[2] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar, and J. Ott, "Consolidate IoT Edge Computing with Lightweight Virtualization," *IEEE Network*, vol. 32, no. 1, pp. 102–111, Jan 2018.

[3] M. Haus, A. Y. Ding, and J. Ott, "Managing IoT at the Edge: The Case for BLE Beacons," in *Proceedings of the 3rd Workshop on Experiences with the Design and Implementation of Smart Objects (SmartObjects)*, 2017, pp. 41–46.

[4] H. Wu, Q. Wang, J. Xiong, and M. Zuniga, "SmartVLC: When Smart Lighting Meets VLC," in *Proceedings of the 13th International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, 2017, pp. 1–12.

[5] D. Kotz, T. Henderson, I. Abyzov, and J. Yeo, "CRAWDAD dataset dartmouth/campus," Downloaded from https://crawdad.org/dartmouth/campus/20041109/movement, Nov. 2004.

[6] B. Alipour, L. Tonetto, A. Y. Ding, J. Ott, R. Ketabi, and A. Helmy, "Flutes vs. Cellos: Analyzing Mobility-Traffic Correlations in Large WLAN Traces," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2018, pp. 1–12.

[7] D. Feng, L. Lu, Y. Yuan-Wu, G. Y. Li, S. Li, and G. Feng, "Device-to-Device Communications in Cellular Networks," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 49–55, 2014.

[8] A. Gupta and R. K. Jha, "A Survey of 5G Network: Architecture and Emerging Technologies," *IEEE Access*, vol. 3, pp. 1206–1232, 2015.

[9] T. Li, Q. Liu, and X. Zhou, "Practical Human Sensing in the Light," in *Proceedings of the 14th International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2016, pp. 71–84.

[10] H. Zhang, W. Du, P. Zhou, M. Li, and P. Mohapatra, "DopEnc: Acoustic-based Encounter Profiling Using Smartphones," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2016, pp. 294–307.

[11] A. Matic, O. Mayora-Ibarra, A. Maxhuni, and V. Osmani, "Virtual Uniforms: Using Sound Frequencies for Grouping Individuals," in *Proceedings of the ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication (UbiComp Adjunct)*, 2013, pp. 159–162.

[12] D. Gijsbrecht, F. Heller, J. Schöning, and F. Kawsar, "Grouve: Spontaneous Proximal Group Formation with Ultrasonic Sound Waves," in *Proceedings of the 8th EAI International Conference on Mobile Computing, Applications and Services (MobiCASE)*, 2016, pp. 140–141.

[13] V. Mavroudis, S. Hao, Y. Fratantonio, F. Maggi, C. Kruegel, and G. Vigna, "On the Privacy and Security of the Ultrasound Ecosystem," in *Proceedings of the 17th Privacy Enhancing Technologies Symposium (PETS)*, 2017, pp. 1–18.

[14] D. Tsonev, S. Videv, and H. Haas, "Towards a 100 Gb/s Visible Light Wireless Access Network," *Optics Express*, vol. 23, no. 2, pp. 1627–1637, 2015.

[15] H. Lee, T. H. Kim, J. W. Choi, and S. Choi, "Chirp Signal-Based Aerial Acoustic Communication for Smart Devices," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 2407–2415.

[16] V. Gerasimov and W. Bender, "Things that Talk: Using Sound for Device-to-Device and Device-to-Human Communication," *IBM Systems Journal*, vol. 39, no. 3.4, pp. 530–546, 2000.

[17] W. Jiang and W. M. Wright, "Ultrasonic Wireless Communication in Air using OFDM-OOK Modulation," in *Proceedings of the International Ultrasonics Symposium (IUS)*, 2014, pp. 1025–1028.

[18] Avery Li-chun Wang, "An Industrial-Strength Audio Search Algorithm," in *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)*, 2003, pp. 1–7.

[19] C. Paasch and O. Bonaventure, "Multipath TCP," *Communications of the ACM*, vol. 57, no. 4, pp. 51–57, 2014.

Publication 6

M. Haus, A. Y. Ding, and J. Ott. Managing IoT at the Edge: The Case for BLE Beacons. In *Proceedings of the 3rd ACM MobiCom Workshop on Experiences with the Design and Implementation of Smart Objects (SmartObjects)*, pages 41–46, 2017. doi: 10.1145/3127502.3127510

Publication Summary

The number of deployed Internet of Things (IoT) devices is steadily increasing and the management of them is becoming crucial, especially for large scale IoT deployments in which ad hoc management becomes untenable. We identified three main challenges for IoT device management: 1) we need a unified management process for IoT devices installed at various locations, 2) lack of well-defined IoT management procedure with all necessary device operations for each phase of the IoT device life cycle, and 3) time consuming and error prone manual efforts are needed to configure IoT infrastructures. We proposed the iConfig framework to tackle these challenges using a programmable edge module being able to cover multiple end-devices, such as smartphones, wearables, and smart boards. For instance, we successfully tested iConfig on Android smartphone and smartglass (MAD Gaze X5). Supported by our edge modules, iConfig enforces a unified configuration procedure by connecting various IoT devices to its management backend. Furthermore, iConfig covers the whole life cycle of IoT devices: register devices, manage configurations, monitor devices, and debug devices. We avoid misconfiguration of devices, a dominant cause of system failures, due to automated configuration of IoT devices. We showcase different add-on services, to highlight the long term benefits of iConfig: 1) global view of distributed IoT devices to monitor them and localize broken devices for replacement, 2) collect maintenance data, such as battery voltage, to debug IoT devices and identify malfunction devices, and 3) ensure up-to-date software versions by automatically distributed software and firmware updates. Based on our evaluation, we highlighted that we can save up to 83 % of the configuration time due to the iConfig automated device configuration with minimal user interaction compared to the manual configuration of IoT devices. Thereby, we recognized another drawback of manual device configuration, the susceptibility to errors, only 1/3 of all manual configurations were entirely correct. With respect to the interaction between users and surrounding IoT devices, our user study and testbed experiments revealed that the conventional, well-known screen-keyboard setup is far from optimal because they restrict the user mobility. To overcome this problem, we realized a second prototype dedicated for wearables, e.g., smartglass, combined with speech control allowing a more fluent interaction during user movement and limit the distraction of user attention.

Managing IoT at the Edge: The Case for BLE Beacons

Michael Haus
Technical University of Munich
haus@in.tum.de

Aaron Yi Ding
Technical University of Munich
ding@in.tum.de

Jörg Ott
Technical University of Munich
ott@in.tum.de

ABSTRACT

Managing IoT devices in urban areas is becoming crucial because the majority of people living in cities and the number of deployed IoT devices are steadily increasing. In this paper we present iConfig, an edge-driven platform dedicated to manage IoT devices in smart cities. The goal is to address three major issues in current IoT management: registration, configuration, and maintenance. The core of iConfig is its programmable edge module, which can be deployed across smartphones, wearables, and smart boards to configure and interact with physically proximate IoT devices. Through testbed experiments and usability studies, we reveal the hardship and hidden pitfalls in managing IoT devices, especially for low budget devices like Bluetooth Low Energy (BLE) beacons. Our system evaluation shows that iConfig can effectively address the aforementioned IoT management challenges by harnessing the mobile and edge cooperation. To inspire community contributions, we further present concrete use cases to illustrate how iConfig can reduce operational cost and facilitate future edge-centric IoT research.

KEYWORDS

IoT Configuration; Management of Edge Devices

1 INTRODUCTION

54 % of the world's population lives in urban areas and by 2050, it could be 66 % [16]. Moreover, the number of deployed Internet of Things (IoT) devices is steadily increasing and projected to reach approximately 50 billions in 2020 [15]. Managing urban areas and its applications is hence becoming important. Urban IoTs support the smart city concept [14] which integrates traditional and modern information and communication technology (ICT) for a unified and simple access to services for the city administration and the residents [21, 22]. The aim is an enhanced use of public resources, improving quality of services for citizens while reducing operational costs of public administration [21]. Besides that, smart city environments are based on a multitude of devices, such as smartphones, sensors, embedded systems, smart meters [14]. Each of these devices have their own purpose, only together they are able to satisfy all service requirements of smart cities. For instance, IoT boards serve as local gateways, collecting sensor data, and provide backend connectivity. In contrast, standalone Bluetooth Low Energy (BLE) beacons are cheap and simple to attach to many objects serving mainly for indoor localization and proximity detection of devices.

In spite of a growing demand for IoT management, there is still a lack of tools to seamlessly manage large scale IoT deployments in which ad hoc management becomes untenable. We need an up-to-date overview of all distributed devices during different phases of their life cycle, including installation, registration, user customization, and device control. Moreover, the IoT configuration framework should be vendor independent and support a diverse range of devices [5]. We identified three main challenges for IoT management. First, IoT devices are installed at various locations which is difficult to handle in large scale deployments and requires a unified management process. Second, there is no well-defined IoT management procedure that includes all necessary device operations corresponding to each phase of the IoT device life cycle. This causes management overhead and information fragmentation. Third, manual efforts are needed to configure distributed IoT infrastructures which is time consuming and error prone and hence increase operational costs [20].

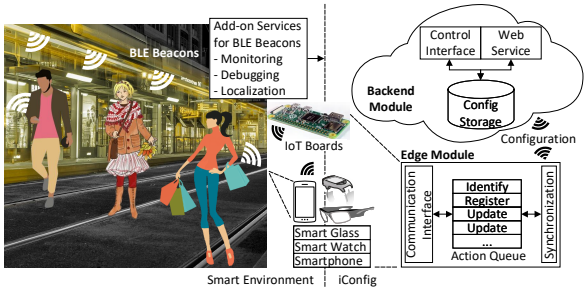


Figure 1: iConfig in the context of smart environments

To tackle these challenges, we propose iConfig, an edge-driven platform that takes care of all installed IoT devices. The framework covers the entire IoT device life cycle: registration, configuration management, monitoring, and debugging. Moreover, our design aims to control the full spectrum of IoT devices, from high end IoT boards to low budget BLE beacons. iConfig enables automated edge device management and hence minimizes operational cost. Fig. 1 shows iConfig in the context of smart environments taking advantage of programmable devices to run edge modules on smart glasses or IoT boards. This allows iConfig to connect various IoT devices to its management backend, which enforces a unified configuration procedure. As an example, we successfully tested iConfig on Android smartphone and smart glass (MAD Gaze X5). In particular, the iConfig edge modules are dedicated for users interested in managing and interacting with IoT environments.

The design principles of iConfig include the following: 1) automatic configuration of IoT devices to avoid misconfigurations which become one of the dominant causes of system failures [20], 2) easy to use frontend, 3) device orchestration via a global view, and 4) open platform for developers to enable add-on services.

The key contributions are summarized as follows:

- We analyzed and identified key properties of IoT device management that must be attained to manage large scale deployments in smart cities.

- We designed and implemented iConfig, an edge-driven platform dedicated for IoT device management. We demonstrate the efficacy of iConfig via prototype implementations, which target at BLE beacons without backend connectivity.
- Our usability study and testbed experiments further uncover hidden aspects in IoT management that are important and deserve future research.

The rest of the paper is structured as follows. Section 2 defines requirements for IoT device management. Section 3 introduces BLE beacons and Section 4 highlights different use cases of iConfig. In Section 5 we present the system architecture, workflow, and implementation details of iConfig. The evaluation in Section 6 consists of performance tests regarding memory usage and system scalability. In addition, we conducted a user study to show differences between manual and automatic device configuration. Section 7 presents related work and Section 8 provides a discussion about user interactions with their surrounding devices and contains details about add-on services via iConfig. We conclude and outline future work in Section 9.

2 REQUIREMENTS FOR IOT DEVICE MANAGEMENT

A major challenge for IoT is the management and configuration of pervasive deployments, especially for IoT devices without backend connectivity. We identified three stages of IoT device management.

The first stage covers device deployment and registration, in which the device is identified and initially configured with default settings. Afterwards, the device is registered at a backend service combined with location information. This manual first step is done only once per device.

Second, automated configuration of device parameters is the fundamental service for IoT device management. The configuration parameters depend on the specific device. Thereby, we classify IoT devices into two different types: standalone and connectivity devices. Standalone devices are only equipped with limited short-range transmission techniques, such as near field communication (NFC), ultra-wideband (UWB), ZigBee, and/or Bluetooth. These IoT devices (e.g., BLE beacons) have no backend link and require additional edge modules for device management. The edge modules can run on platforms which support short-range communication and provide a backend link. In contrast, connectivity devices (e.g., smart home gateways) have a connection to the backend and offer easier device management.

Third and final stage of IoT device management refers to the centralized backend service which receives device registrations together with configuration data. This enables multiple add-on services: 1) monitoring of distributed IoT devices via global view, including configuration status or localization of broken devices for replacement, 2) debugging of IoT devices via collected maintenance data, such as uptime or battery voltage to identify malfunction devices, 3) software and firmware updates distributed via central backend service which ensures up-to-date software versions and improves IoT security, and 4) parameter updates for a set of devices. Thereby, the device selection can be based on different criteria, such as close proximity among IoT devices.

Our case study targets BLE beacons, which represent one of the most challenging classes of IoT devices due to missing backend connectivity. Furthermore, the number of deployed beacons is expected to be much higher than the number of IoT boards. Thus, it's important to have a scalable framework which covers the defined requirements for IoT device management.

3 BLE BEACONS

The battery powered BLE beacons are small-size wireless devices that transmit a short-range BLE signal to mobile computing devices (e.g., smartphones) [18]. Via BLE, users' devices are notified of the beacon proximity by receiving signals which contain contextual information, typically about indoor surroundings and its contents. Thus, the receiving end is able to perform location aware actions, such as accessing specific URLs for marketing purposes [18]. For example, the Los Angeles International Airport uses Bluetooth beacons to track and dispatch wheelchairs for passengers in need of assistance [11]. In another scenario, universities use beacons to help students navigate through the library, guiding them to resources, study spaces, and services in the library [6].

Our beacons [4] are capable of sending three different BLE message formats and can be locked via a password. The primary beacon standards are Eddystone and iBeacon. Eddystone is a protocol specification that defines a BLE message format for beacons [8]. It describes different frame types to transmit device identifier, URL, or telemetry data containing battery voltage, beacon temperature, count of advertised packets, and uptime. In contrast, the iBeacon standard [1] transmits only a device identifier. Besides that, the beacon vendor added an own proprietary BLE message format named sBeacon to transmit another fixed device identifier imprinted on the beacon.

4 ICONFIG USE CASES

The ability of iConfig to programmatically adjust device parameters facilitates different use cases. For example, we used iConfig to automate the setting of BLE parameters for Wi-Fi interference avoidance as an add-on service (Section 8.2 provides further details). Besides that, iConfig supports debugging and monitoring of BLE beacons by collecting maintenance data, such as uptime and battery voltage of each device. The long term benefits of iConfig come from add-on services based on the programmable interface of iConfig. In the context of a smart city management for dense, co-deployed IoT devices, the energy-aware configuration of devices is important. For instance, the developer can use iConfig to add a module to the iConfig backend to turn off devices routinely at specific time.

5 ICONFIG DEVICE MANAGEMENT

The goal of iConfig is to manage various IoT devices by utilizing programmable edge platforms. In our case study, we focused on the management of BLE beacons as low budget IoT devices. This section presents the system architecture of iConfig and the corresponding working flow for IoT device management. In addition, we provide implementation details regarding software libraries and BLE communication.

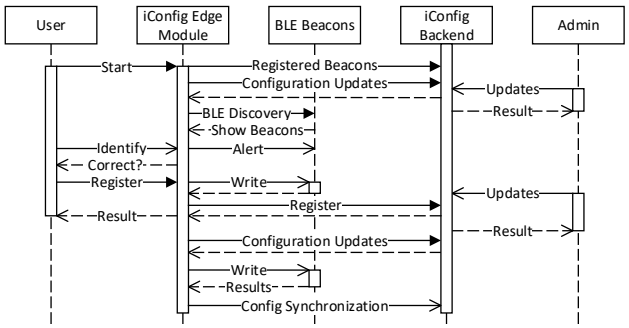


Figure 2: iConfig workflow

5.1 System Architecture and Key Features

The iConfig system architecture, as illustrated in Fig. 1, consists of two major modules: the mobile edge module and the backend module. The framework is able to identify, register, and update IoT devices (in our case BLE beacons). Supported by our speech recognition, a user wearing an iConfig-enabled smart glass can discover, register, and configure BLE beacons while walking around. The edge module is intended to run on mobile devices (e.g., smart glasses, smartphones, tablets) and on static devices like IoT boards. The backend module of iConfig runs at a centralized infrastructure, such as a local server or in the cloud.

The iConfig edge module uses a central action queue for all beacon operations to synchronize user actions (identify and register) with automated updates of beacon settings. Thereby, user interactions are prioritized over automatic beacon updates. Each device action occurs via threads to immediately start the next action from the central operation queue. During registration and update of BLE beacons, the edge module collects maintenance data which enables monitoring of beacon health and detection of broken beacons. To update BLE beacons, we aggregate received signal strength indication (RSSI) for each beacon and update beacons according to descending RSSI sum, which means nearest beacons first. Besides that, the iConfig edge module works in two different modes: offline and online depending on connectivity to the backend. In offline mode, beacon configurations are stored and loaded from local storage for later synchronization to the backend.

The iConfig backend stores all device data at a central storage. We ensure that only beacons are updated where BLE parameters are actually changed. Furthermore, the iConfig backend provides a control interface including status about device update, device functionality (e.g., broken BLE beacon), device health, and offers localization via indoor map and place image.

5.2 Workflow

The main task of the user is to register IoT devices (in our case BLE beacons). Therefore, the user holds a smartphone running the iConfig edge module and walks around to discover nearby BLE beacons. At the application start, the edge module automatically received registered beacons via iConfig backend and shows only unregistered beacons, when they are discovered by the user. Fig. 2 presents the iConfig workflow.

After discovering unregistered beacons, the user is able to identify one beacon at a time. The beacon shows a red light as feedback for device identification. When the beacon identification was successful, the user can register the beacon to the iConfig backend with additional information for device localization, such as nearest room number, picture of device place. Afterwards, during registration, the edge module automatically configures the BLE beacon with a default configuration including password, iBeacon, and Eddystone identifier to ensure that the device is ready to use. Finally, the edge module synchronizes all configuration data to the backend. The registration has to be done only once per beacon.

For advanced management, the iConfig backend provides a control interface for the administrator, including a global view about installed BLE beacons. The administrator has technical knowledge and is responsible to manage the registered BLE beacons. Therefore, the administrator is able to adapt multiple device configurations at once. This is possible by linking devices to groups and groups to configurations.

The device update of a BLE beacon is independent of user actions and automatically triggered when two conditions are satisfied: 1) adapted BLE configuration from administrator via iConfig backend available, and 2) the beacon is currently discovered by the user via iConfig edge module. The update process runs in the background of the iConfig edge module, only recognizable by blinking red lights as feedback of successful configuration.

5.3 Implementation Details

We implemented two prototypes of the iConfig edge module for different device types: smartphone and smart glass. We take advantage of speech recognition to enable hands-free device configuration. On the smartphone the speech recognition can be optionally activated, on the smart glass it is automatically activated to allow a convenient usage of the iConfig edge module.

For beacon configuration, the Eddystone standard provides two Bluetooth Gatt services for the adaption of all Eddystone parameters. However, this service was not available for our beacons and does not cover all BLE parameters, such as the configuration of sBeacon, iBeacon, and password. We hence used the communication library provided by the beacon vendor [2], which uses a customized communication interface for beacon configuration. The communication library works asynchronously via callbacks to provide the result to set beacon parameters. The iConfig edge module is able to configure the following parameters:

- sBeacon: transmission power and advertisement rate.
- iBeacon: device identifier, transmission power, and advertisement rate.
- Eddystone beacon: device identifier, URL, and transmission power and advertisement rate for packets: telemetry (TLM), identifier (UID), and URL.
- Password

The iConfig backend is implemented in Python and provides a REST API and a control interface via cherrypy. The beacon device data can be stored in any database. In our case, we use MongoDB to store device configurations, one document per beacon. Moreover, we use an URL shortening service to overcome the limit of 17 bytes for the Eddystone URL field. For system security, the password of

each beacon is stored encrypted in MongoDB via AES encryption. In addition, SSL secures the wireless network connection between iConfig edge and backend module. Thus, common attacks such as man-in-the-middle or sniffing are not possible.

6 EVALUATION

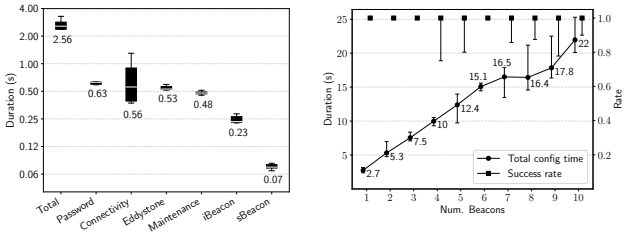
We analyzed the system performance of iConfig regarding memory usage and configuration scalability over multiple beacons. Moreover, we break down the configuration time for one beacon to show the duration of each configuration part. Additionally, we conducted a user study to highlight drawbacks by manually configure IoT devices.

6.1 System Performance

Regarding the memory usage of iConfig edge module, we measured a deployment on Android smartphone (OS 7.1.1) in offline and online mode during configuration of ten BLE beacons. In online mode, the edge module used 6.50 ± 0.98 MB similar to offline mode with a memory usage of 6.47 ± 0.96 MB. These results show that the memory footprint of the iConfig edge module is small enough to run on programmable IoT devices, e.g., smart glass, IoT boards.

The next evaluation part of iConfig refers to the beacon configuration. As illustrated in Fig. 3 (a), we break down the configuration time for one beacon over 20 rounds. Thereby, we measured a total configuration time of 2.56 s shared over six different configuration phases. The connectivity and maintenance phase takes almost half of the configuration time (41 %). The connectivity phase consists of connect and disconnect time. It shows the largest time fluctuation due to interference among BLE beacons. Regarding device configuration, to set the password takes most of the time (25 %). To configure Eddystone packets, including telemetry, URL, and identifier sum up to 21 % of the configuration time. iBeacon and vendor specific sBeacon takes less time. Our results show the worst case, in which all configurable fields are adapted. Usually, the identifiers for Eddystone and iBeacon are set only once during device registration.

For scalability testbed and to evaluate iConfig in a dense deployment, we placed ten beacons in a circle with a diameter of 1 m around the smartphone running iConfig edge module. Thereby, we evaluated ten cases (from one to ten beacons) each over 20 rounds. All BLE parameters for transmission power (5 dBm - 80 m) and advertisement rate (10 Hz - 10 times a second) were set to the maximum, which reflects the worst case setting in terms of interference among BLE beacons. Our evaluation yielded 18 unauthenticated connect errors (meaning BLE configuration is not possible) out of 1100 connect attempts, i.e., rate of 1.64 %. Fig. 3 (b) illustrates scalability results when configuring ten BLE beacons against ten cases each over 20 rounds. The configuration time shows a linear increase over all beacons, on average an increase of 2.2 s per beacon. In addition, we evaluated the success rate which describes whether configuration parameters were correctly set. A success rate of 100 % means that all parameters are correctly set to the predefined value. In our evaluation, the lowest success rate was 75 % during the configuration of four beacons. In most cases, iConfig achieved a success rate of 100 %. In our testbed with a dense deployment of BLE beacons and high interference among devices, iConfig is still



(a) Break down of configuration time for one beacon

(b) Configuration time and success rate from one to ten beacons

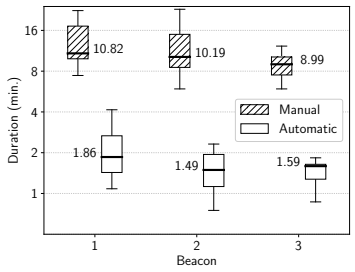
Figure 3: iConfig system evaluation

able to achieve a reasonable configuration time per beacon with high success rates.

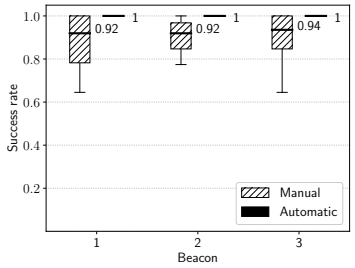
Lessons learned from dense deployment - beacon placement matters: In the first attempt, we placed the smartphone in the middle, which is running the iConfig edge module and on each side a row of five beacons with a distance of 1 m between them. The two beacons on each side with shortest distance (1 m) to the smartphone had an advantage and were mostly successfully configured. The configuration of remaining beacons mostly failed due to unauthenticated connection attempts caused by interference between BLE beacons. To overcome this problem, we rearranged the testbed setting and placed all beacons in a circle with a diameter of 1 m around the smartphone running iConfig edge module. Thus, all beacons were equally privileged in terms of distance to configuration controller (in our case smartphone).

6.2 Usability Study

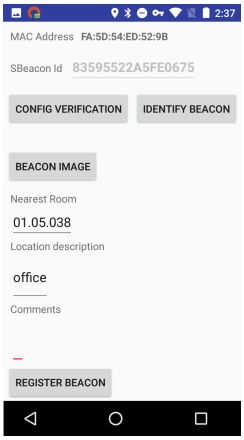
Our user study serves two purposes: 1) to reveal the gap between a manual and an automatic configuration system, and 2) to test the application interface of the iConfig edge module. In total, ten persons participated in the study, all PhD students or Postdocs with a strong background in computer science. The user study consisted of a questionnaire and the configuration of BLE beacons. The participants manually configured three beacons with a predefined configuration using the vendor application [3]. In the next step, the same beacons were registered via iConfig, in which the default configuration was automatically written to each BLE beacon. For each beacon, we calculated the success rate by comparing the predefined configuration with actual beacon settings. The configuration time for the vendor app was taken manually. Fig. 4 (a) shows the configuration time per beacon. The manual configuration took in average six times longer than the iConfig automatic configuration, which reflects a time saving of 83 %. The results show a slight decrease of configuration time when the user is more familiar with the configuration system. Fig. 4 (b) presents the success rate, how many parameters were correctly set at the beacon. In case of manual configuration, the lowest success rate over all beacons was 58 % and the median is around 92 %. In total, only 1/3 of all manual configurations were entirely correct. On the other hand, our iConfig framework achieved for all BLE configurations the success rate of 100 %. In general, this shows that the manual configuration of BLE beacons is time consuming and error prone due to type errors by users. Fig. 4 (c) presents the iConfig edge module to register BLE



(a) Config duration per beacon



(b) Config success rate per beacon



(c) iConfig edge module

ACTION	UPDATE STATUS	MAC ADDRESS	BEACON ID	NEAREST ROOM	MAINTENANCE
Edit	up-to-date	C9:ED:20:D2:6A:5F	8EC968106F2	01.05.038	PKT COUNT: 35155830 / VOLTAGE: 3.6000001 / UPTIME: 459853408
Edit	up-to-date	EB:26:C9:E4:DC:5B	C28C3BC4AE8	01.05.038	PKT COUNT: 24238715 / VOLTAGE: 3.6000001 / UPTIME: 459856770
Edit	up-to-date	D6:82:27:05:E4:0B	2F0446B54A6	01.05.038	PKT COUNT: 31240056 / VOLTAGE: 3.6000001 / UPTIME: 459853016

(d) iConfig control interface (reduced due to space limits)

Figure 4: Results of user study and screenshots of iConfig framework to administrate IoT devices

beacons and Fig. 4 (d) illustrates the iConfig control interface which is part of the backend module.

In the following questionnaire, most participants rated the manual configuration via the vendor app as difficult. On the other hand, the automatic configuration by iConfig was entirely rated as easy. The configuration process is faster and device registration requires less manual input by the user. Moreover, we asked the participants of the user study for useful features of the iConfig backend. The monitoring of beacon health and the localization of beacons achieved highest consent.

7 RELATED WORK

A majority of the related work is dedicated to services taking advantage of beacon deployments, while BLE beacons themselves are not considered. The work in [17] presented a system where an IoT hub is dynamically selected from a changing set of users' devices. The IoT hub is responsible for configuring a set of services running on proximate devices to ensure an efficient management of available resources. Harris *et al.* [9] used BLE-tagged products for inventory control. Their work included a detailed analysis of BLE regarding signal propagation, deployment, and protocol attributes. In our case, iConfig is dedicated for IoT device management including BLE beacons.

Indoor localization and proximity detection are major use cases for BLE beacons. Faragher *et al.* [7] found that BLE positioning systems provide a higher accuracy compared with Wi-Fi fingerprinting. The authors of [13] proved the feasibility of using BLE beacons for proximity detection in working places. The detailed analysis of BLE parameters, such as advertising interval and transmission power shows the impact of these settings on the proximity detection mechanism. Another use case is highlighted by Michalevsky *et al.* [12] using cryptographic secret handshakes over BLE protocol. Their proposal can enable private communication among nearby devices without central servers, which addresses a crucial concern for privacy in device-to-device (D2D) communication [10].

8 DISCUSSIONS

8.1 User Interactions

A key observation from our user study and experiments is that the interaction among users, their smart gadgets, and surrounding IoT devices via the conventional screen-keyboard setup is far from optimal. Especially for smart cities with a multitude of services empowered by IoT devices, the system interaction should be more natural and fluent. Even when users adopt iConfig on smartphones, the keyboard input is still hindering the user experience no matter how automated iConfig has made the entire configuration process. This is the main reason for our second prototype dedicated for wearables (e.g., smart glass). The combination of speech recognition and hands-free devices can enable a more integrated interaction during user movement and limit the distraction of user attention. iConfig is endeavored to enhance user experience and to streamline management of large scale IoT deployments, especially for low budget devices such as BLE beacons without backend connectivity.

8.2 Add-On Services

iConfig provides the core functionality for device management such as automated device updates, monitoring, and debugging of IoT devices. Furthermore, the centralized backend allows orchestrating devices via a global view and thereby iConfig is able to serve as platform for developers to enable add-on services. For instance, we used iConfig to realize an add-on service for Wi-Fi interference avoidance by reducing transmission frequency and range of BLE beacons because both wireless technologies work on the same frequency of 2.4 GHz. The results are interesting due to the increasing usage of BLE-enabled devices in areas with existing Wi-Fi networks, such as BLE beacons for indoor localization. The testbed consisted of ten BLE beacons which are placed around a Wi-Fi access point in a distance of a few centimeters. iConfig automatically adapted BLE parameters, such as transmission power or packet advertisement rate to measure the impact of BLE beacons on existing Wi-Fi networks via different BLE configurations. The worst case refers to highest transmission power (5 dBm - 80 m) and highest advertisement rate (10 Hz - 10 times a second) for BLE beacons. On the other hand, we used in the realistic setting an advertisement rate of 2.5 s and a transmission power of -20 dBm (12 m). The Wi-Fi throughput was measured via wrk benchmark [19] over ten rounds for each of the six different file sizes. Fig. 5 shows the impact of

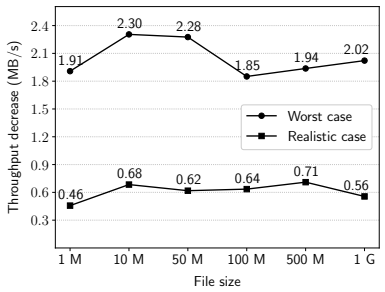


Figure 5: Decrease of Wi-Fi throughput by BLE beacons

BLE beacons on a Wi-Fi network. In the worst case scenario, the Wi-Fi throughput decreases by -12.12 % (w/o BLE: 16.89 MB/s, w/: 14.84 MB/s) and in the realistic case by -5.96 % (w/o BLE: 10.22 MB/s, w/: 9.61 MB/s).

9 CONCLUSION AND FUTURE WORK

We designed and implemented iConfig for IoT device management in smart cities. The goal is to streamline the management of large scale IoT deployments, especially for low budget devices such as BLE beacons. We have evaluated iConfig in two ways. In the usability study, we revealed the hardship introduced by existing mechanisms. Our results highlighted the time saving and higher success rates of iConfig owing to its automatic configuration with minimal user interaction. The system evaluation of iConfig further showed a small memory footprint of the mobile edge module, making it suitable for various smart devices. We also evaluated the robustness and scalability of iConfig by configuring multiple BLE beacons in a testbed. The configuration time scales linearly with high success rates. Moreover, we discussed several use cases enabled by iConfig, such as dynamic adjustment of beacon parameters for IoT testbeds, and efficient detection of broken devices.

For future work, we plan to enhance the mobile edge module. An immediate step is to port the current Android implementation to a Linux environment, which can be deployed on multiple programmable IoT boards. The fixed installation of IoT boards provides the benefit of a continuous access to nearby BLE beacons. This will ensure a more reliable and faster update process by using the iConfig platform to unify the management of edge devices. Another extension is to improve synchronization redundancy when several iConfig edge modules are in offline mode. Based on the last synchronization time with the iConfig backend module, multiple iConfig edge modules can carry different configuration versions for the same device. If the iConfig edge modules are proximate to BLE beacons, they independently start to update the devices in which some BLE beacons are several times updated and the last written configuration is active, which is maybe not the latest. To solve the problem via an extension, a lock functionality on the local device itself provides a reliable mechanism to avoid simultaneous device access and inefficient device updates. In combination with a locally stored timestamp or version number of the currently active device configuration we would be able to avoid inconsistent device configurations.

REFERENCES

[1] Apple. 2017. iBeacon. (2017). Retrieved July 26, 2017 from https://developer.apple.com/ibeacon/

[2] Bluvision. 2014. BEEKS Beacons SDK. (2014). Retrieved July 26, 2017 from http://developer.bluvision.com/developer/beeks-beacons-sdk/

[3] Bluvision. 2016. BEEKS Beacon Maker. (2016). Retrieved July 26, 2017 from https://play.google.com/store/apps/details?id=com.bluvision.beaconmaker

[4] Bluvision. 2016. Specification Sheet: iBeek Sensor Beacon. (2016). Retrieved July 26, 2017 from http://bluvision.com/wp-content/uploads/2016/12/Specs-iBEEK1.6.pdf

[5] Enri Dalipi, Floris van den Abeele, Isam Ishaq, Ingrid Moerman, and Jeroen Hoebeke. 2016. EC-IoT: An Easy Configuration Framework for Constrained IoT Devices. In *Proceedings of the IEEE 3rd World Forum on Internet of Things (WF-IoT)*. 159–164.

[6] EdTech Staff. 2017. Bluetooth Beacons Could Improve Student Experience on Higher Ed Campuses. (2017). Retrieved July 26, 2017 from http://www.edtechmagazine.com/higher/article/2017/02/bluetooth-beacons-could-improve-student-experience-higher-ed-campuses

[7] Ramsey Faragher and Robert Harle. 2015. Location Fingerprinting With Bluetooth Low Energy Beacons. *IEEE Journal on Selected Areas in Communications* 33, 11 (2015), 2418–2428.

[8] Google. 2017. Eddystone. (2017). Retrieved July 26, 2017 from https://github.com/google/eddystone

[9] Albert F. Harris, Vansh Khanna, Guliz Seray Tuncay, and Robin Hillary Kravets. 2016. Smart LaBLEs: Proximity, Autoconfiguration, and a Constant Supply of Gatorade(TM). In *Proceedings of the First IEEE/ACM Symposium on Edge Computing (SEC)*. 142–154.

[10] Michael Haus, Muhammad Waqas, Aaron Yi Ding, Yong Li, Sasu Tarkoma, and Jörg Ott. 2017. Security and Privacy in Device-to-Device (D2D) Communication: A Review. *IEEE Communications Surveys & Tutorials* 19, 2 (2017), 1054–1079.

[11] Joe Bates. 2016. LAX uses Bluetooth Beacon Technology to improve the Efficiency of Wheelchair Operations. (2016). Retrieved July 26, 2017 from http://www.airport-world.com/news/general-news/5976-lax-uses-bluetooth-beacon-technology-to-improve-wheelchair-operations.html

[12] Yan Michalevsky, Suman Nath, and Jie Liu. 2016. MASHaBLE: Mobile Applications of Secret Handshakes over Bluetooth LE. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom)*. 387–400.

[13] Alessandro Montanari, Sarfraz Nawaz, Cecilia Mascolo, and Kerstin Sailer. 2017. A Study of Bluetooth Low Energy Performance for Human Proximity Detection in the Workplace. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 1–10.

[14] Hans Schaffers, Nicos Komninos, Marc Pallot, Brigitte Trousse, Michael Nilsson, and Alvaro Oliveira. 2011. Smart Cities and the Future Internet: Towards Cooperation Frameworks for Open Innovation. In *The Future Internet*. Lecture Notes in Computer Science, Vol. 6656. Springer, 431–446.

[15] Statista. 2017. Internet of Things (IoT): Number of Connected Devices Worldwide from 2012 to 2020 (in Billions). (2017). Retrieved July 26, 2017 from https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[16] United Nations. 2014. World Urbanization Prospects. (2014). Retrieved July 26, 2017 from https://esa.un.org/unpd/wup/publications/files/wup2014-highlights.Pdf

[17] Kirill Varshavskiy, Albert F. Harris III, and Robin Kravets. 2016. MiHub: Wearable Management for IoT. In *Proceedings of the Workshop on Wearable Systems and Applications (WearSys)*. 1–6.

[18] Michael Wang and Jack Brassil. 2015. Managing Large Scale, Ultra-Dense Beacon Deployments in Smart Campuses. In *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 606–611.

[19] Will Glozer. 2017. wrk - a HTTP benchmarking tool. (2017). Retrieved August 09, 2017 from https://github.com/wg/wrk

[20] Tianyin Xu and Yuanyuan Zhou. 2015. Systems Approaches to Tackling Configuration Errors. *ACM Computing Surveys* 47, 4 (2015), 1–41.

[21] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. 2014. Internet of Things for Smart Cities. *IEEE Internet of Things Journal* 1, 1 (2014), 22–32.

[22] Vladimir Zdraveski, Kostadin Mishev, Dimitar Trajanov, and Ljupco Kocarev. 2017. ISO-Standardized Smart City Platform Architecture and Dashboard. *IEEE Pervasive Computing* 16, 2 (2017), 35–43.

Publication 7

M. Haus, J. Krol, A. Y. Ding, and J. Ott. Feasibility Study of Autonomous Drone-based IoT Device Management in Indoor Environments. In *Proceedings of the 1st ACM SIGCOMM Workshop on Mobile AirGround Edge Computing, Systems, Networks, and Applications (MAGESys)*, pages 1–7, 2019. doi:10.1145/3341568.3342105

Publication Summary

The effective management of connected IoT devices becomes challenging due to the scale of deployments. In addition, much of the deployed IoT technology is designed to be invisible. This limits the services that can be provided and raises concerns with respect to user privacy, e.g., users are not aware about sensing devices taking place in any given area, and system security, e.g., organizations might not know which IoT devices are connected to their network. With an up-to-date overview of all distributed devices including their locations, we can take full advantage of IoT deployments while avoiding threats, such as security concerns and user privacy. As extension of iConfig, we built a new edge module dedicated for drone-based IoT device management. We explored the feasibility of using small COTS drones to create indoor maps of Wi-Fi and Bluetooth Low Energy (BLE) devices. Thereby, we can further reduce operational costs for device management by being independent of users and able to autonomously collect environment data for device maps. A drone is most appropriate for our purpose to explore indoor environments because it can move more freely compared to robots moving on the ground with obstacles on the way. The drone carries the drone controller, e.g., smartphone, and device detection platform to achieve a fully autonomous area exploration, avoiding user involvement and dependency to a ground-control station. We have implemented two different area exploration strategies: random direction and boundary following which are simple to realize and satisfy our needs to explore the area and detect devices. Our evaluation in several real-world testbeds showed that the limited battery capacity of the drone is the main problem to explore larger indoor areas for automated device management. In addition, compared to the random direction strategy, the flight control using boundary following for indoor area exploration achieves the best results with respect to a quick device detection and small localization error. To limit the negative impact of the restricted battery capacity of the drone, we optimized the flight path of the drone to visit more IoT devices using simulations with larger regions and a varying number of IoT devices. Our evaluation revealed, in terms of most discovered IoT devices and least explored area, the best path optimization uses a sampling-based graph of the drone's environment combined with Dijkstra path planning. A further optimization is to identify special points of interest, so-called hot spots, where the drone hovers at a certain position and reaches multiple IoT devices at once without flying to each device. This behavior saves the limited energy of the drone and significantly increases the number of discovered IoT devices.

Feasibility Study of Autonomous Drone-based IoT Device Management in Indoor Environments

Michael Haus
Technical University of Munich

Jan Krol
Technical University of Munich

Aaron Yi Ding
Delft University of Technology

Jörg Ott
Technical University of Munich

ABSTRACT

Future computing environments are embedded with many sensors for applications like augmented reality. Much of the deployed Internet of Things (IoT) technology is designed to be invisible. To support user's privacy awareness, a map of surrounding sensing devices is beneficial to determine the nature of data collection taking place in any given area. Moreover, security and governance issues are among the challenges IoT poses to organizations which might not know exactly which IoT devices are connected to their network. For instance, many employees bringing their own devices to the workplace. We explore the feasibility to use small COTS drones to create indoor maps of wireless devices. These comprehensive device maps serve as basis for device localization and monitoring to enhance user privacy and network security. We analyze the impact of our device management platform at the drone's energy consumption and evaluate the device detection rate, explored area, and localization error. Due to the restricted battery capacity of the drone, we simulate larger areas with a varying number of IoT devices to highlight the limits of our drone-based device management platform regarding area exploration and reachable IoT devices.

KEYWORDS

IoT device management, COTS drones, Indoor mapping, Device localization and monitoring, User privacy, Network security

1 INTRODUCTION

The term "IoT" is an umbrella keyword covering various aspects related to the extension of the Internet and Web into the physical realm [13]. In 2019, an estimated amount of 27 billion connected IoT devices are deployed worldwide [18] and the effective management of these IoT devices becomes challenging due to the scale of deployments. For example, in a harbor logistic warehouse the port authority and different shipping vendors gradually use wireless tags and sensing kits to trace and log their products for transaction and book keeping purposes. Since the position of those sensing devices are often not well tracked and sometimes moved incidentally by maintainers, we need an automatic mechanism to detect and trace them. Furthermore, the lack of awareness about spatially distributed IoT assets both limits the services that can be provided and raises concerns with respect to user privacy and system security.

Security and governance issues are among the challenges IoT poses to organizations stemming from the widespread adoption of IoT devices, their diversity, standardization obstacles, and inherent mobility. For instance, smart cameras and smoke detectors enhance security; smart thermostats, smart light bulbs and sockets facilitate power savings; and so forth. Organizations might not know exactly which IoT devices are connected to their network, particularly caused by the trend that employees bringing their own IoT devices (BYOIoT) to the workplace. For example, with the use of wearables in the healthcare and business service/consulting industries. Surveying this BYOIoT trend, also 25–50 % of remote employees connected at least one IoT device to the enterprise network [4]. A situation which threatens the security and integrity of the network and the devices. To support user's privacy awareness, a map of surrounding sensing devices is beneficial to determine the nature of data collection taking place in any given area. It does not protect users against deliberate covert surveillance, but we are able to inform users about the data capture upon approaching a region.

With mapped IoT devices, an up-to-date overview of all distributed devices including their locations and capabilities, we can fully harness IoT deployments while avoiding potential threats in terms of security concerns and user privacy. We build upon our ground work [6, 7] by developing new modules that are dedicated for drone-based IoT device management. This further reduces operational costs to be independent of users and able to autonomously gather data for device maps. We explore the feasibility of using small COTS drones to create indoor maps which consist of Wi-Fi and Bluetooth Low Energy (BLE) devices. These comprehensive device maps serve as basis for device localization and monitoring to enhance network security and user privacy. Via repeated drone flights we can perform device presence detection and our indoor device mapping is able to identify new devices and track changes in the environment.

Our contributions are summarized as follows:

(1) We realize a drone-enabled IoT device management to autonomously create indoor maps containing wireless devices. This serves for device monitoring and localization improving user privacy and network security.
(2) We analyze the impact of our device management platform at the drone's energy consumption. Moreover, we identify the best working area exploration strategy in terms of explored area over time, device detection rate, and localization error.
(3) To unwind the restricted battery capacity of the drone, we simulate larger areas with different number of IoT devices to optimize the flight path reaching more devices. After the initial area exploration the IoT device positions are known and we are able to tweak the flight path of the drone.

2 PLATFORM FOR DEVICE MANAGEMENT

To enrich our platform in Fig. 1, we build upon our ground work [6, 7] by developing new modules that are dedicated for drone-based IoT device management. The drone has to be as easy as possible to control autonomously, small enough to fly indoors, and powerful enough to carry a smartphone for on-board control and an IoT board for device detection. We use the DJI Mavic Air to analyze the feasibility of using COTS drones for indoor device management.

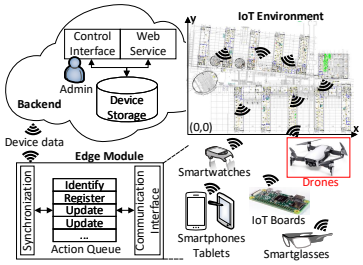
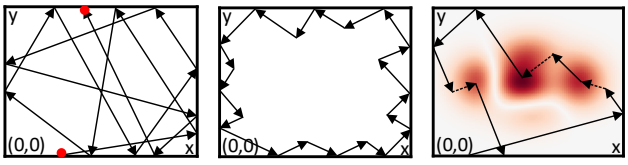


Figure 1: Platform overview for IoT device management. The edge modules run at multiple end-devices which are static user-independent (IoT boards) and mobile user-dependent (smartwatches, smartglasses, smartphones). We focus on drones as mobile user-independent end-devices.

(a) Assembly view **(b) Schematic view**

Figure 2: Platform for autonomous device management



(a) Random direction **(b) Follow boundary** **(c) Hybrid**

Figure 3: Strategies for area exploration

Drone platform assembly To achieve fully autonomous area exploration, our drone controller and device detection platform flies with the drone as shown in Fig. 2(a), avoiding user involvement and dependency to a ground-control station. The schematic view in Fig. 2(b) shows the smartphone controlling the drone and powering the Raspberry Pi Zero W. During startup the Raspberry Pi begins to continuously perform device detection by receiving Wi-Fi probe requests and BLE messages. Our drone flies at an altitude of 1.8 m to more easily explore areas by avoiding spatial barriers such as chairs, tables. Inspired by the reactive control of [3] at which control decisions are taken only upon observed changes, we actively control the flight of the drone only in case of encountered obstacles. We take advantage of the drone's API and move the drone via virtual sticks which are translated into movement.

Area exploration We have implemented two different area explorations: random direction and boundary following. These are sufficiently simple to realize and fulfill our target to explore the area and recognize devices. With the control mode random direction as illustrated in Fig. 3(a) (start and end position highlighted with red dots), the drone flies up to an obstacle like a wall and randomly chooses another direction until no obstacle blocks the drone. Fig. 3(b) shows the principal working flow of boundary following, the drone flies in its heading until it recognizes an obstacle, then it turns right and left to follow the boundary as closely as possible. For future work, the hybrid control mode in Fig. 3(c) applies by default the random direction strategy. When during movement the drone detects an increasing signal strength, we allow the drone to follow this signal (dotted line). At the peak of the signal strength the drone falls back to random direction control mode.

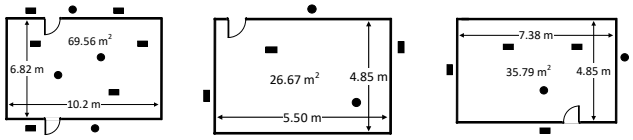
Device detection For the device detection of Wi-Fi devices we use the Nexmon firmware patch [16] to enable the Wi-Fi monitor mode at the Raspberry Pi Zero W. On this basis, tcpdump collects Wi-Fi probe requests of surrounding devices and concurrently we perform channel hopping to find as many Wi-Fi devices as possible. The discovery of BLE devices occurs via a Python script receiving broadcasted BLE messages. The device detection of all Wi-Fi and BLE devices includes list entries with timestamp, MAC address, and received signal strength indicator (RSSI).

Relative positioning To calculate a relative position of each encountered device and be able to generate device maps, we define a relative coordinate system for our university building. Each room has a reference frame with an origin at which the drone starts to explore the room. From this start position, the drone estimates its own position (x, y) via time, gyroscope (direction), and velocity. Before the flight we perform a time synchronization between smartphone and Raspberry Pi to later identify the device locations via matched log timestamps. During each second of the flight, the smartphone logs the current relative position and time. After the drone's flight, we are post processing the gathered data and for each device we select encounters with strongest RSSI, i.e., at this time the drone was nearest to the wireless device. As result, we create a device list with MAC address, relative location (x, y), time, and signal strength.

Limitations Our device detection is limited to active wireless devices, entirely passive devices cannot be recognized. Besides that, the drone's obstacle avoidance restricts indoor area exploration to medium- and large-sized areas. In our experiments, we found that the drone maintains a safety distance of 1.8 m to obstacles, mainly motivated by outdoor usage and weather conditions like wind. Hence, it is not possible to move the drone in small rooms, e.g., $\leq 12\,m^2$, or corridors with a width of two or three meters.

3 EVALUATION

We analyze our drone-based device management regarding initial area exploration to create precise device maps. To be specific, we evaluate the device detection rate, explored area, and localization error over time. After the initial area exploration, we know the device positions to optimize the flight route of the drone to reach more IoT devices within a limited flight time with one battery load. Therefore, we simulate larger areas with a varying number of IoT devices to show the impact of path generation algorithms and hot spots in terms of maximum reachable devices. By hovering at a hot spot, the drone is able to reach multiple IoT devices at once which is more efficient than to fly to each device individually.



(a) Show room: 6 x Wi-Fi, 4 x BLE **(b) Meeting room: 3 x Wi-Fi, 2 x BLE** **(c) Office room: 4 x Wi-Fi, 2 x BLE**

Figure 4: Test environments for autonomous area exploration of ●: BLE devices and ▬: Wi-Fi devices

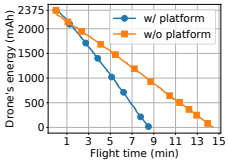


Figure 5: Impact of device management platform at the drone's energy drain

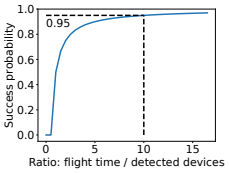
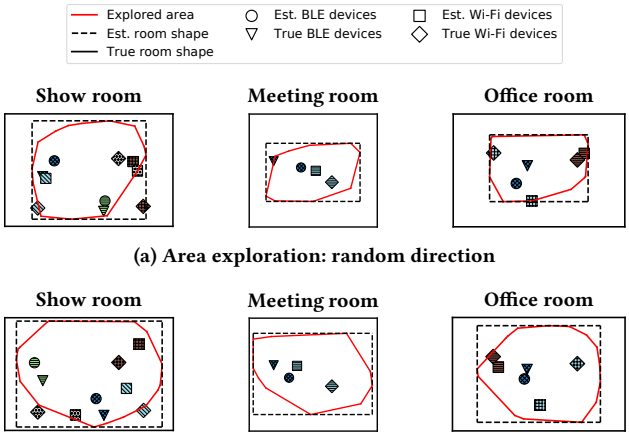


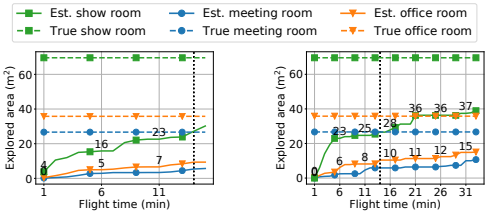
Figure 6: Asses the area exploration of unknown territory (success probability)



(a) Area exploration: random direction

(b) Area exploration: boundary following

Figure 7: Maps of indoor area exploration



(a) Random direction **(b) Boundary following**

Figure 8: Experimental evaluation of explored area

3.1 Device Management Platform in Testbed

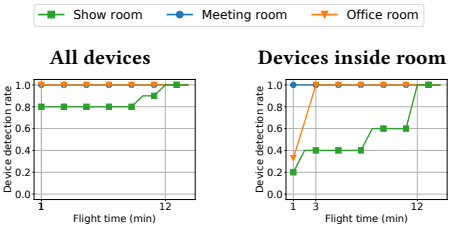
We have chosen three real-world test environments as shown in Fig. 4 to evaluate our indoor device management platform in terms of explored area, device detection rate, and localization error. We placed a varying number of Raspberry PIs acting as BLE or Wi-Fi device inside and outside of each room representing smartwatches, printers, BLE beacons, IoT sensor boards, and so forth. For an useful localization mechanism it is most important to distinguish between devices located inside and outside of the room. Only in larger areas like in our simulation with the university hall (Section 3.2) the device position inside the area gains importance.

Energy consumption The main impact of our device management platform at the theoretical maximum flight time of 21 min is the additional weight of the smartphone with 143 g to control the drone and the Raspberry Pi Zero W with 9 g to gather wireless information. Fig. 5 shows the energy consumption over time with and without our platform: the battery drains at -164.18 mAh/min without our platform which results in a maximum flight time of 14.45 min. In contrast, with our device management platform, the battery drain increases to -277.08 mAh/min which results in a decreased flight time of 8.51 min, a decrease of 41.1 %.

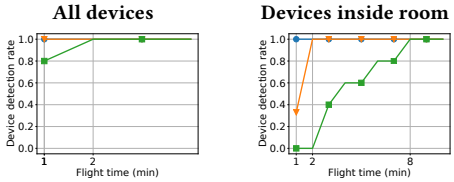
Device map For a qualitative evaluation, Fig. 7(a) and Fig. 7(b) show the device mapping for each testbed with identified devices inside the room and their true and estimated positions (for each device an own color and shade). The area explorations are performing well independent of the room size, most device positions are estimated closely to the true position. The outer relative coordinates from the drone define the convex hull illustrated as explored area. Based on the knowledge that room shapes are mostly rectangular, we take the maximum coordinates from the convex hull in each direction, and lines through these points define the room boundary.

Explored area We evaluate the area explorations presented in Fig. 3 where we implemented the random direction and boundary following. Fig. 8 presents the explored area over time compared to the true area of the testbed. The explored area corresponds qualitatively to the actual room size. On average, random direction discovers 0.94 m^2/min and boundary following obtains 0.66 m^2/min. We take the same point in time (14 min) to compare the explored area over time (highlighted via a dotted line). Random direction explores 15.09 m^2 covering 30.32 % of the true room area. On the other hand, the area exploration using the boundary following results in 14.28 m^2 covering 29.76 % of the true area. In general, our area exploration is too slow to discover a reasonable area size within the limited flight time of one battery load.

Device detection We classify devices to be within the room based on the assumption that devices inside the room, nearby the drone without an obstacle in between, achieve a stronger signal strength compared to devices outside of the room. Over three test environments with varying room size, we experimentally identified RSSI thresholds of -45 dB for Wi-Fi devices and -50 dB for BLE devices to be able to distinguish devices inside and outside of the room. For each testbed, Fig. 9 shows the accuracy and duration until all devices (inside and outside of the room) are recognized and classified to be inside the room. To detect all devices inside and outside of the room, the flight mode with boundary following takes on average 1.33 min compared to a 3.5 times increase of 4.67 min using random direction. In addition, to classify the devices to be within the room, the random direction control mode (5.33 min) lasts

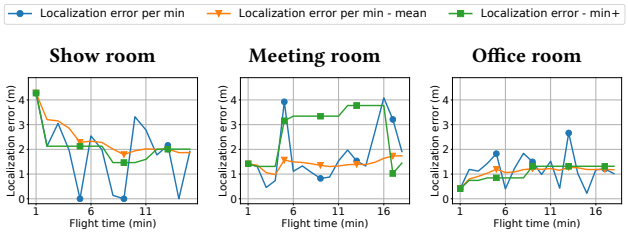


(a) Area exploration: random direction

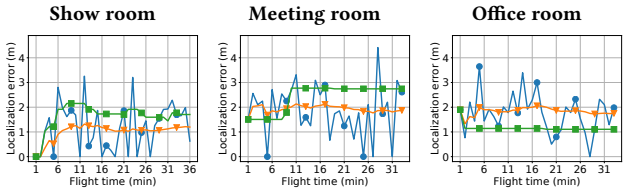


(b) Area exploration: boundary following

Figure 9: Experimental evaluation of device detection



(a) Area exploration: random direction



(b) Area exploration: boundary following

Figure 10: Experimental evaluation of localization error

1.5 times longer than the boundary following (3.67 min). The area exploration using boundary following is faster compared to random direction, which allows a good device detection within restricted flight time.

Localization error For each device we take the device position(s) with the strongest signal strength and compute the average localization error over all of them. We calculate three different localization errors by varying the time scale of the input data. First, we compute a localization error for each data collected over one minute. Second, we average the localization errors over data collected within one minute. Third, we compute localization errors by taking only one device position with strongest signal strength for each device from current and previous data collected over one minute time intervals (min+). The localization error over time highlights the varying RSSI precision. The mean localization error achieves the smallest positioning error. Using the boundary following control, the localization error results on average in 1.59 m ± 0.19 m compared to the random direction with 1.67 m ± 0.35 m. The second most precise device positions are computed over minute intervals of gathered data. The flight mode using random direction gains a localization error of 1.6 m ± 0.96 m which is similar to the boundary following of 1.61 m ± 0.91 m. With more collected data to estimate device positions, the computed localization error increases. The boundary following control mode results on average in a smaller localization error of 1.74 m ± 0.38 m compared to 1.97 m ± 0.66 m using random direction.

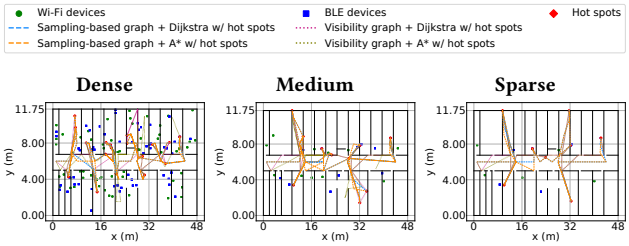
Explore unknown territory Usually we explore areas with an unknown number of IoT devices. To assess when we are done exploring an unknown territory, we are averaging our results for the device detection rate and flight time over different area explorations and testbeds to compute a success probability. Fig. 6 shows whether the area is well enough explored depending on the ratio of flight time and number of detected devices. This means that most IoT devices are found and we can classify them to be inside the room, and the localization error is around 2 m.

Summing up, the indoor area exploration using boundary following achieves a superior performance compared to random direction. Boundary following is faster to detect and classify devices and obtains a smaller localization error, the size of the explored area is similar to random direction.

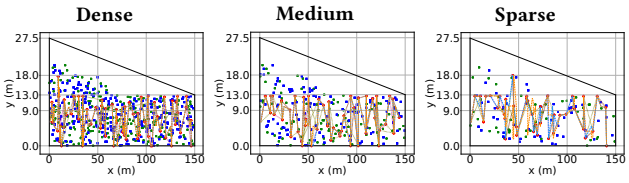
3.2 Optimizing Device Coverage in Simulation

In terms of reachable IoT devices, the limited battery capacity of the drone is the most severe limitation for our device management platform. After the initial area exploration the device positions are known. On this basis, we simulate larger areas with different number of IoT devices to optimize the flight path of the drone to reach more IoT devices with one drone's battery load.

Simulation settings As prerequisite for the simulation, we have measured the drone's velocity and energy drain rate during the flights in our real-world test environments similar to the simulation environments regarding spatial arrangement. In the university lab with more obstacles the drone achieves a velocity of 0.9 m/s in comparison to 1.5 m/s in the university hall as one large room with fewer spatial barriers. We set the energy limit of the drone to ≥ 35 %, i.e., with the remaining energy the drone is able to fly back to a predefined position to change or recharge the battery. Based on a detailed building map, we have modeled two different simulation environments, our university lab with 564 m^2 and the university hall with 3038 m^2. Fig. 11 highlights our simulation for one exemplary device distribution (for readability we omit the flight paths without hot spots). We use a random device distribution with different number of IoT devices ranging from dense, medium to sparse. For an area of 25 m^2 similar to a single room, we randomly distribute five devices (dense), two devices (medium), and one device (sparse). Each IoT device is either a Wi-Fi or BLE device and we randomly select a wireless range of [5, 15] m for Wi-Fi devices and [1, 7] m for BLE devices. We simulate a maintenance task for each IoT device by a random waiting time between 5 to 10 s. Table 1 presents the number of devices, the determined hot spots and their ratio for all



(a) Simulation: university lab (area: 564 m^2, drone velocity: 0.9 m/s)



(b) Simulation: university hall (area: 3038 m^2, drone velocity: 1.5 m/s)

Figure 11: Visualization of exemplary simulation runs

Table 1: Simulation details with number of devices, hot spots, and their ratio over different device distributions

Simulation	University lab			University hall		
Devices	Dense	Mid	Sparse	Dense	Mid	Sparse
Wi-Fi	59	11	5	233	93	46
BLE	56	12	7	287	122	56
Total	115	23	12	520	215	102
Hot spots	19	10	8	65	46	39
Ratio (%)	16.5	43.5	66.7	12.5	21.4	38.2

simulation environments. With more IoT devices, the number of hot spots are increasing while the ratio to the total number of devices is decreasing. With less devices, multiple hot spots are covering only one IoT device.

Exemplary simulation results For our exemplary simulations in Fig. 11, the visibility graph applying Dijkstra path planning performs best with a median distance of 200.1 m in case with hot spots and 489.3 m without hot spots. The visibility graph with A* path planning obtains on average a longer flight route of 176.1 m with hot spots and 600.5 m without hot spots. The sampling-based graph using A* or Dijkstra path planning achieve similar results of 216.3 m with hot spots and 568.9 m without hot spots. The hot spots save on average 61 % of the flight distance.

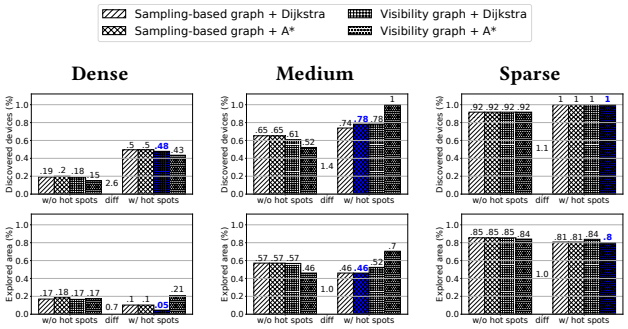
Path generation To compute the flight path of the drone, we generate two different graphs of IoT devices: sampling-based graph and a visibility graph using the map data from our simulations like in Fig. 11. After the graph construction representing the simulation environment, we apply common Dijkstra and A* path planning to find the shortest flight path of the drone visiting IoT devices. We compare four different path generation approaches: sampling-based graph using Dijkstra and A* path planning, and visibility graph with Dijkstra and A* path planning. Our simulation results over ten rounds in Fig. 12 show the discovered IoT devices by the drone with one battery load compared to the total number of IoT devices (Table 1). Over all simulation runs, the drone's energy limit is 39.5 %±2.61 % of the battery capacity with 2375 mAh (≈ 938 mAh). This lowers the average flight time to 303.84 s ± 26.15 s compared to 464.77 s ± 87.73 s without an energy limit leading to a decrease of 34.63 %. The target of our simulation is to find the most efficient path generation, i.e., reach a maximum number of IoT devices by a minimum amount of explored area. Therefore, we consider the ratio between discovered IoT devices (%) and explored area (%) to highlight the best performing path planning for each environment and device distribution. Over all simulation runs, the sampling-based graph applying Dijkstra path planning performs best. In comparison, the sampling-based graph using A* achieves a median ratio of 98.55 % of discovered IoT devices and explored area. This is similar to the median ratio of 98.35 % using the visibility graph with Dijkstra path planning. With a median ratio of 87.11 % of discovered IoT devices and explored area the visibility graph applying A* path planning performs worst. The Dijkstra path planning finds mostly the best path even without utilizing a search heuristic as used by A* path planning.

Hot spots To further optimize the flight path and minimizing the flight time, we compute so-called hot spots at which the drone can reach as many IoT devices as possible without movement. We calculate the intersection points among the wireless ranges of all IoT devices to find the positions of the hot spots. The difference factor in Fig. 12 shows the average increase in discovered IoT devices and decrease in explored area. The effect of the hot spots diminishes from a dense to a sparse device distribution. In case of a dense distribution of IoT devices, we achieve the highest impact with an increase of discovered IoT devices by 2.6 (university lab) and 2.4 (university hall). There is no significant performance difference between our two test environments. The university lab consists of 23 rooms with an average room size of 25 m^2 and many spatial barriers such as walls in contrast to the university hall as one large room with 3038 m^2.

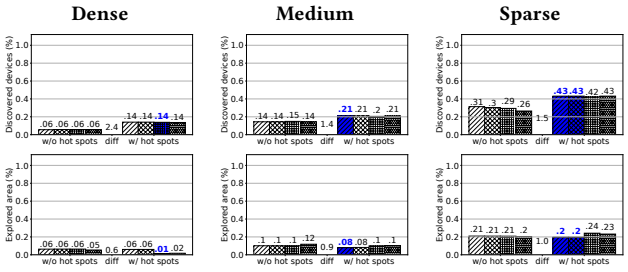
Over all simulations it is not possible to reach all IoT devices even in case of hot spots with one battery load and an energy limit of 35 % to safely fly back to a predefined position. Only in the university lab with a sparse and medium number of distributed IoT devices we are able to discover all IoT devices. The university hall is too large to be entirely discovered with one battery load. Over all simulations, hot spots increase the number of reachable IoT devices by 39.76 %.

4 USE CASES: PRIVACY AND SECURITY

Our central backend for IoT device management in Fig. 1 provides add-on services via a global map of locally installed wireless IoT devices for enhanced user privacy and network security. The drones as mobile user-independent end-devices identify local devices and transmit this information to our device management backend. To improve user privacy, we synchronize MAC addresses, device positions and device types (Wi-Fi or BLE) to end user devices. Combined with indoor localization at the user device we can inform users about nearby wireless sensing devices which potentially infringing their privacy. Regarding enhanced network security, we



(a) Simulation: university lab



(b) Simulation: university hall

Figure 12: Simulation results of two test environments, we highlight the best performing path generation regarding most discovered IoT devices and least explored area

are able to distinguish between remote and local wireless devices by comparing network scans with our indoor maps via included MAC addresses. Hence, network administrators can more easily recognize unauthorized wireless devices and via white lists of network devices we can discover locally installed malicious wireless devices.

5 RELATED WORK

Existing work [1, 2, 10, 14, 15] mainly focus on specialized drones, control of the drone, or using energy consuming and computation heavy vision data to create indoor maps. In contrast, our device management platform only analyzes the wireless signals to localize devices for indoor mapping.

Users are unaware of the locations and purposes of IoT devices which are infringing their privacy by sensing data in the background. Hence, it is important to catalog wireless devices with detailed information about their locations and capabilities to support awareness regarding user privacy [17]. A lot of work [8, 11] analyzes network traffic by applying machine learning models like random forest or convolution neural network to identify devices within the network, their device type, and detect anomalous deviations in communication patterns [19]. In our work, we are passively sensing network traffic of Wi-Fi and BLE devices to detect their presence and from the received signal strength (RSSI) we infer the location of the devices.

Maps of surrounding devices can be generated from three sources: 1) authorities, 2) users, and 3) data provided by the infrastructure. Data from the infrastructure removes the need of human effort. For instance, analysis of electric signals to identify home appliances [5], or using network traffic to automatically detect IoT devices [12]. In our case, we create indoor maps together with installed IoT devices and their locations. We need a moving data collector to gather required data and be independent of humans to ensure a uniform quality of the device maps. We have chosen a drone to be most suitable for our intended purpose which can move more freely within indoor areas compared to robots moving on the ground [9] and obstacles on the way like tables, chairs, and staircases.

6 CONCLUSION

By our control design of the drone, the control unit, e.g., smartphone, flies with the drone, we are independent of any ground control station. The drone's battery capacity is the main limiting factor for automated area exploration. In our real-world testbed, we found that the flight control using boundary following for indoor area exploration works best in terms of faster device detection and smaller localization error compared to the random direction strategy. Our simulation revealed that the sampling-based graph applying Dijkstra path planning achieves the best path generation in terms of most discovered IoT devices and least explored area. Hot spots which cover multiple IoT devices at once are significantly improving the number of discovered IoT devices. However, it is still not possible to reach all IoT devices distributed over the entire space with one drone's battery load.

For future work, we plan to implement the hybrid area exploration as described in Fig. 3(c) to gain further insights about the drone's flight path and precision of estimated IoT device locations. To reduce the time to explore areas, we can simultaneously fly multiple drones in different areas and merge the gathered data to generate an overall device map. Besides that, we seek to avoid the additional weight of the smartphone to control the drone by moving the control application to the Raspberry Pi Zero W which is currently only gathering the wireless data like Wi-Fi probe requests and BLE messages for device detection.

REFERENCES

[1] Pompilio Araujo, Rodolfo Miranda, Diedre Carmo, Raul Alves, and Luciano Oliveira. 2017. Air-SSLAM: A Visual Stereo Indoor SLAM for Aerial Quadrotors. *IEEE Geoscience and Remote Sensing Letters* 14, 9 (2017), 1643–1647.

[2] Yingcai Bi, Hailong Qin, Mo Shan, Jiaxin Li, Wenqi Liu, Menglu Lan, and Ben M. Chen. 2016. An Autonomous Quadrotor for Indoor Exploration with Laser Scanner and Depth Camera. In *Proceedings of the 12th IEEE International Conference on Control and Automation (ICCA)*. 50–55.

[3] Endri Bregu, Nicola Casamassima, Daniel Cantoni, Luca Mottola, and Kamin Whitehouse. 2016. Reactive Control of Autonomous Drones. In *Proceedings of the 14th International Conference on Mobile Systems, Applications and Services (MobiSys)*. 207–219.

[4] Calero. 2015. 3 Ways the Internet of Things will Impact Enterprise Security. https://www.calero.com/mobility-service-support/3-ways-the-internet-of-things-will-impact-enterprise-security/

[5] Sidhant Gupta, Matthew S. Reynolds, and Shwetak N. Patel. 2010. ElectriSense: Single-Point Sensing Using EMI for Electrical Event Detection and Classification in the Home. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing (UbiComp)*. 139–148.

[6] Michael Haus, Aaron Yi Ding, Pan Hui, and Jörg Ott. 2017. Demo: iConfig - What I See is What I Configure. In *Proceedings of the 12th ACM Workshop on Challenged Networks (CHANTS)*. 1–2.

[7] Michael Haus, Aaron Yi Ding, and Jörg Ott. 2017. Managing IoT at the Edge: The Case for BLE Beacons. In *Proceedings of the 3rd Workshop on Experiences with the Design and Implementation of Smart Objects*. 41–46.

[8] Lei Bai, Lina Yao, Salil S. Kanhere, Xianzhi Wang, and Zheng Yang. 2018. Automatic Device Classification from Network Traffic Streams of Internet of Things. *CoRR* abs/1812.09882 (2018).

[9] Ren C. Luo and Chun C. Lai. 2012. Enriched Indoor Map Construction Based on Multisensor Fusion Approach for Intelligent Service Robot. *IEEE Transactions on Industrial Electronics* 59, 8 (2012), 3135–3145.

[10] Aravindh Mahendran, Ayush Dewan, Nikhil Soni, and K. Madhava Krishna. 2013. UGV-MAV Collaboration for Augmented 2D Maps. In *Proceedings of Conference on Advances In Robotics (AIR)*. 1–6.

[11] Yair Meidan, Michael Bohadana, Asaf Shabtai, Juan David Guarnizo, Martín Ochoa, Nils Ole Tippenhauer, and Yuval Elovici. 2017. ProfilIoT: a Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis. In *Proceedings of the Symposium on Applied Computing (SAC)*. 506–509.

[12] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N. Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. 2017. IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT. In *Proceedings of the 37th IEEE International Conference on Distributed Computing Systems (ICDCS)*. 2177–2184.

[13] Daniele Miorandi, Sabrina Sicari, Francesco de Pellegrini, and Imrich Chlamtac. 2012. Internet of Things: Vision, Applications and Research Challenges. *Ad Hoc Networks* 10, 7 (2012), 1497–1516.

[14] Guillaume Pepe, Massimo Satler, and Paolo Tripicchio. 2015. Autonomous Exploration of Indoor Environments with a Micro-Aerial Vehicle. In *Proceedings of the Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*. 43–52.

[15] Arindam Saha, Soumyadip Maity, and Brojeshwar Bhowmick. 2018. Indoor Dense Depth Map at Drone Hovering. In *Proceedings of the 25th IEEE International Conference on Image Processing (ICIP)*. 96–100.

[16] Matthias Schulz, Daniel Wegemer, and Matthias Hollick. 2017. Nexmon: The C-based Firmware Patching Framework. https://nexmon.org

[17] Peter Shaw, Mateusz Mikusz, Petteri Nurmi, and Nigel Davies. 2019. IoT Maps: Charting the Internet of Things. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications (HotMobile)*. 105–110.

[18] Statista. 2019. Internet of Things (IoT): Number of Connected Devices Worldwide from 2015 to 2025 (in Billions). https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[19] Thien Duc Nguyen, Samuel Marchal, Markus Miettinen, Minh Hoang Dang, N. Asokan, and Ahmad-Reza Sadeghi. 2018. DÏoT: A Crowdsourced Self-learning Approach for Detecting Compromised IoT Devices. *CoRR* abs/1804.07474 (2018).

Non-Evaluation Relevant Parts and Publications

Contents

1 Non-Evaluation Relevant Parts and Publications

This chapter contains non-evaluation relevant material based on the following publications which did not undergo a full peer-review and are not relevant for grading:

8. M. Haus, J. Ott, and A. Y. Ding. Enabling Seamless Device Association with DevLoc using Light Bulb Networks for Indoor IoT Environments. *arXiv*, pages 1–14, 2020. URL: `https://arxiv.org/abs/2005.07731` [last checked 16.06.2020]

9. M. Haus, A. Y. Ding, C. Xu, and J. Ott. Demo: Touchless Wireless Authentication via LocalVLC. In *Proceedings of the 16th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, page 531, 2018. `doi:10.1145/3210240.3211119`

10. M. Haus, A. Y. Ding, P. Hui, and J. Ott. Demo: iConfig - What I See is What I Configure. In *Proceedings of the 12th ACM MobiCom Workshop on Challenged Networks (CHANTS)*, pages 29–31, 2017. `doi:10.1145/3124087.3124103`

1.1 iConfig: Device Discovery via Speech Control on Wearables

The key observation identified in our user study in Publication 6 is that the conventional screen-keyboard setup is far from optimal for the interaction among users, their smart gadgets, and surrounding IoT devices. The system interaction should be more natural and fluent. Moreover, by our lab experiments, we encountered the problem how to easily recognize and integrate new IoT devices into the management backend. The one-time manual registration of IoT devices should be as easy as possible to reduce the risk that not all IoT devices will be registered leading to an incomplete device map. The lack of awareness about spatially distributed IoT assets limits the services that can be provided. This is the reason for our second prototype dedicated for wearables as shown in Fig. 1.1 from Publication 10. As most natural way, the included speech recognition

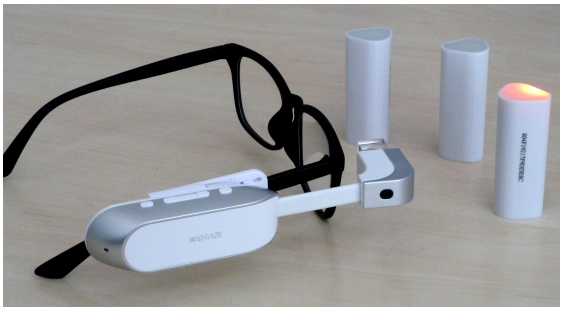


Figure 1.1: Smartglass with speech control for IoT device management [3]

runs completely offline on the smartglass for hands-free device configuration during user movement limiting the distraction of user attention.

For device discovery and registration the speech control can be activated via the key phrase *geronimo*. The user can control the iConfig edge module using the following keywords: *select* physically closest BLE beacon identified by strongest signal strength, *target* specific BLE beacon chosen by unique device id, *identify* BLE beacon via visual feedback (blinking red light), and *register* BLE beacon at the iConfig backend including an image of the device which is controllable via speech, e.g., zoom in and out. With respect to the robustness of our speech recognition, we evaluated multiple metrics for string similarity of the speech input, such as Euclidean, Levenshtein, and Jaro-Winkler distance. The aim is to enhance the quality of our speech recognition to avoid misinterpretation, e.g., the number three is sometimes recognized as 'free'. In our evaluation, the Jaro-Winkler distance, designed for short strings, achieved the best results to calculate the string similarity of the speech input.

1.2 iService: User-Oriented and Privacy-Aware Proximity Services

As a minor use case from Publication 9, our automated authentication for wireless networks achieves a 'touchless' experience using distance-bounding VLC. We improve the usability by avoiding manual distribution and tedious input of passwords for login. We use LocalVLC from Publication 2 for M2M communication to ease the setup of Wi-Fi networks including devices like smartphone, tablet, laptop as well as IoT devices like sensor boards. Other mechanisms to exchange credentials still require human interaction, such as WPS or QR codes. The video `https://www.youtube.com/watch?v=e8kjfDNmVSA` [last checked 16.06.2020] shows our working demo. Via LocalVLC we broadcast the

Table 1.1: Best working classifiers and features to predict semantic device groups [1]

Device group	Testbed	Classifier	Feature type	AUC	CS
Personal	Dense	Naive bayes	Contact frequency per week - sum	.99	23
	Medium	Extra trees	Grouping time per week - mean	.97	17
	Sparse	Ada boost	Contact frequency per week	.98	56
Family & friends	Dense	Naive bayes	Contact frequency per week - sum	.99	17
	Medium	Gradient boosting	Grouping time per week - std	.85	72
	Sparse	Ada boost	Contact frequency per week	.98	55
Well-known & stranger	Dense	Ada boost	Grouping time per week - sum	.98	16
	Medium	Ada boost	Contact frequency per week	.99	22
	Sparse	Ada Boost	Contact frequency per week	.99	45

network name and continuously generated time-based one-time passwords (TOTPs) for the wireless login. The user's end-device retrieves the VLC transmitted login data and continually scans for nearby wireless networks. In case of spotting a matching network name, it performs the automated authentication without any manual interaction.

1.2.1 Data Access Control to Enhance User Privacy

As a practical extension we analyze patterns of device associations, e.g., frequency and time of encounters, to infer different semantic device groups: personal, family & friends, and well-known & stranger devices. On this basis, we are able to define data sharing policies for device groups who may exchange which data with whom. For instance, the user distributes sensible information only among personal devices and deny sharing with unknown devices. We generated an artificial log of device associations to mimic different environments with a varying number of devices per device group and different grouping times. The sparse environment includes three devices per device group and a grouping time in the range [10, 60] min, the medium environment has six devices per group and a grouping time of [20, 120] min, and the dense environment uses a grouping time of [30, 180] min and nine devices per device group. We created 43 different feature sets using the following information: 1) how long the devices are associated together per event, per day, and per week, 2) how often the devices are associated together per day and per week, and 3) the time ratio between association time and entire time frame per day and per week. We model the problem as multi-class classification to identify

Table 1.2: Hardware specifications of platforms for the runtime analysis of different libraries for fully homomorphic encryption [1]

System	CPU	RAM
Server	40x Intel Xeon E5-2630 2.2 GHz	768 GB
Next unit of computing (NUC) PC	4x Intel Core i5-6260U 1.8 GHz	16 GB
IoT board	1x ARM AM3358 1 GHz	0.5 GB

the best working classifiers and features to predict device groups. Using 10-fold cross validation, identified by the average area under the curve (AUC) and cold start in days (CS), Table 1.1 presents for each device group and the aforementioned testbeds the best working classifier and feature type to predict semantic device groups. Especially important is the cold start as success criterion meaning after which time we are able to reliably predict the device group, the earlier the better. The cold start is defined by a threshold of 80 %, from this point in time (day), all result metrics including accuracy, precision, recall, and F1-score are above this threshold.

1.2.2 Homomorphic Encryption to Improve User Privacy

We apply fully homomorphic encryption to improve the user's privacy during device associations at our custom light bulb. Multiple parties compute whether they are nearby without learning each other's inputs. This protects the location data, e.g., light patterns, against a wide range of attacks, because it reveals no sensitive information to anyone. Our system model consists of a trusted party as service provider, mobile users, and IoT devices. The service provider is usually considered untrusted and should not learn the proximity test results. Our custom light bulb acts as trusted party and service provider, e.g., data sharing, among nearby mobile devices. Typically fully homomorphic encryption is applied only to a small amount of data to compute the distance between two points of interest, e.g., latitude and longitude of attractions. In contrast, we apply fully homomorphic encryption for a large quantity of time-series data, such as light patterns, and analyze whether it is practically usable. To evaluate the performance of fully homomorphic encryption, we performed a runtime analysis of two state-of-the-art libraries: HElib [4] and SEAL [5] on three platforms with different performance capabilities, described in Table 1.2. We measured the time to compute the euclidean and cosine distance as similarity measure between light patterns with varying vector lengths and averaged the results. Our performance results showed that the HElib library works fastest at the server, the NUC is about 33 % slower, and the IoT board is 11.7

times slower compared to the NUC. Moreover, the SEAL library also performs best at the server, the NUC is 19 % slower, and the IoT board takes 51 times longer. In a nutshell, the runtime analysis motivates the need of new cryptographic primitives for homomorphic encryption on time-series data, being too slow to be usable in practice with a runtime of about 30 s per distance computation whereas we require a maximum runtime of 0.5 s per distance calculation.

Bibliography

[1] M. Haus, J. Ott, and A. Y. Ding. Enabling Seamless Device Association with DevLoc using Light Bulb Networks for Indoor IoT Environments. *arXiv*, pages 1–14, 2020. URL: `https://arxiv.org/abs/2005.07731` [last checked 16.06.2020].

[2] M. Haus, A. Y. Ding, C. Xu, and J. Ott. Demo: Touchless Wireless Authentication via LocalVLC. In *Proceedings of the 16th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, page 531, 2018. `doi:10.1145/3210240.3211119`.

[3] M. Haus, A. Y. Ding, P. Hui, and J. Ott. Demo: iConfig - What I See is What I Configure. In *Proceedings of the 12th ACM MobiCom Workshop on Challenged Networks (CHANTS)*, pages 29–31, 2017. `doi:10.1145/3124087.3124103`.

[4] IBM. HElib. URL: `https://github.com/homenc/HElib` [last checked 16.06.2020].

[5] Microsoft. SEAL. URL: `https://github.com/Microsoft/SEAL` [last checked 16.06.2020].

Publication 8

M. Haus, J. Ott, and A. Y. Ding. Enabling Seamless Device Association with DevLoc using Light Bulb Networks for Indoor IoT Environments. *arXiv*, pages 1–14, 2020. URL: `https://arxiv.org/abs/2005.07731` [last checked 16.06.2020]

Publication Summary

Spontaneous device associations are of particular interest to open up new opportunities for users to share resources or information. We introduced the DevLoc framework for device grouping to combine radio-based communication, like Wi-Fi covering larger areas, with visible light signaling to achieve more fine-granular device associations, impossible to recognize with propagating Wi-Fi. We are able to manage the lighting infrastructure and control the spatial granularity of device grouping. Our custom light bulb is a central part of DevLoc and integrated into a light bulb network to emit light patterns for seamless device associations. In this work we extended our existing DevLoc framework regarding user privacy. First, we added data access control by analyzing patterns of device associations, e.g., frequency and time of encounters, to identify different semantic device groups. On this basis, we are able to define data sharing policies for device groups who may exchange which data with whom. For example, the user distributes sensible information only among personal devices and deny sharing with unknown devices. Second, we applied fully homomorphic encryption to improve the user's privacy during device associations at our custom light bulb. Multiple parties compute whether they are nearby without learning each other's inputs. This protects the location data, e.g., light patterns, against a wide range of attacks, because it reveals no sensitive information to anyone.

Enabling Seamless Device Association with DevLoc using Light Bulb Networks for Indoor IoT Environments

Michael Haus
Technical University of Munich
haus@in.tum.de

Jörg Ott
Technical University of Munich
ott@in.tum.de

Aaron Yi Ding
Delft University of Technology
aaron.ding@tudelft.nl

To enable serendipitous interaction for indoor IoT environments, spontaneous device associations are of particular interest so that users set up a connection in an ad-hoc manner. Based on the similarity of light signals, our system named DevLoc takes advantage of ubiquitous light sources around us to perform continuous and seamless device grouping. We provide a configuration framework to control the spatial granularity of user's proximity by managing the lighting infrastructure through customized visible light communication. To realize either proximity-based or location-based services, we support two modes of device associations between different entities: device-to-device and device-to-area. Regarding the best performing method for device grouping, machine learning-based signal similarity performs in general best compared to distance and correlation metrics. Furthermore, we analyze patterns of device associations to improve the data privacy by recognizing semantic device groups, such as personal and stranger's devices, allowing automated data sharing policies.

***Index Terms*—Mobile ad hoc networks, Network services, Ubiquitous and mobile devices, Similarity measures, Machine learning approaches**

I. Introduction

The capabilities of wireless devices, such as laptops, mobile phones, tablets, IoT boards, enable flexible formation of ad-hoc groups. New opportunities arise for users in physical proximity by dynamic group association to spontaneously share resources or information. We support two different types of proximity applications aimed for end users and Internet of Things (IoT). We envision two use cases for user-oriented, proximity-based applications [1]: 1) Alice is a tourist, rides on the subway and wants to ask locals for the best way to the museum, and 2) Carol is a manager who wants to automatically record who is present at her daily meetings. In addition, we emphasize two use cases for proximity-based IoT applications: 1) location-tagged data from IoT boards facilitate data merging and filtering in case we have the same information from multiple IoT boards placed in the same area and 2) location-based access policy for consumer smart home platforms [2], e.g., Amazon Echo or Google Home. To realize such applications, we explore proximity as a group association technique where devices find one another when they are brought within a close distance of each other or in a dedicated space [3]. Thereby, proximity identifies potential group members and device association refers to the technique that connect (a subset of) those potential group members.

Our system for continuous and seamless device grouping named DevLoc uses visible light signaling because light sources are ubiquitous around us ensuring practicality. DevLoc combines visible light and Wi-Fi as the primary communication means. Compared to the electromagnetic waves of Wi-Fi which easily penetrate physical barriers, visible light does not pass through opaque objects and hence it is a good candidate to realize distance-bounding wireless communication. Based on the distance-limited nature of visible light, we achieve more fine-granular device associations which are impossible to recognize with propagating Wi-Fi. To compensate the downsides of visible light communication (VLC), such as lack of hardware support at mobile devices, e.g., to receive the data, we provide the design of a light tag for pervasive VLC. The light tag usable as sticker can be easily attached to different end-user devices enabling light transmissions. Furthermore, by the analysis of log files of device associations we can infer different semantic device groups, e.g., personal devices, based on the frequency and time of device encounters. This allows us to automatically generate meaningful data sharing policies between devices associated with a certain type such as personal, family, etc. to define with whom sharing or aggregating data. Hence, we are able to move the task to specify data sharing policies to lower communication layers, usually handled as part of the application layer in wireless systems used today. Moreover, to minimize the adaption effort to introduce DevLoc, we adopt a master-slave principle for light bulbs of existing lighting infrastructure. Only the master light bulb requires computation power to perform device associations, the slave light bulb simply broadcasts the light pattern received from the master light bulb via a Wi-Fi interface.

In contrast to existing systems for device grouping, we provide a complete framework to manage the lighting infrastructure and control the spatial granularity of device grouping. As a result, we are able to facilitate applications with different spatial expansion of proximity and overcome the main disadvantage of location tags [1] that users have no control over the spatial granularity of proximity. We enrich the lighting infrastructure by adding light signaling to the widely used Light-Emitting Diode (LED) lamps in residential and office settings. To automatically link physically nearby devices based on the similarity of light patterns, our custom light bulbs combine illumination with visible light signaling. Our generated light patterns for device grouping are unpredictable nonces associated with a location [1]. For instance, like a shared pool of entropy between all users at a given location at a given time. In the following, we state the two key properties of light patterns for device grouping [1]: 1) reproducibility

meaning that two measurements at the same place and time match with high probability, and 2) unpredictability so that an adversary at another location is unable to produce a location tag that matches the tag measured at the actual location at that time.

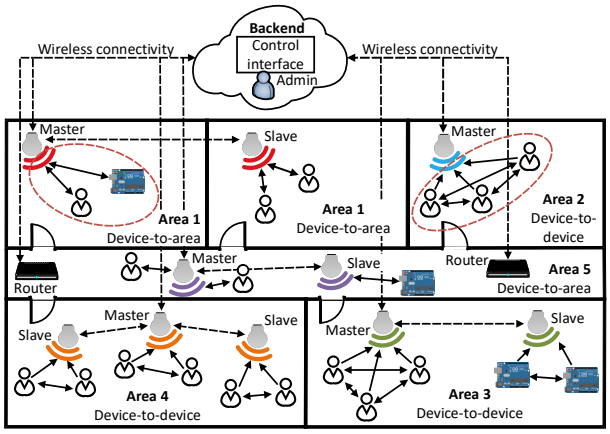
In a nutshell, our work makes the following contributions:

1) We introduce DevLoc for seamless device grouping taking advantage of boundary-limited visible light signaling. We build a custom light bulb to enrich the lighting infrastructure being able to control the spatial granularity of user's proximity.
2) To qualify the feasibility of real-world deployments of DevLoc, we analyze the propagation characteristics of VLC, e.g., maximum achievable detection range of light patterns. Furthermore, we perform a feature selection for light patterns and we analyze the performance of several signal comparison methods via two simulations for static device-to-device grouping and dynamic device-to-area grouping.
3) To enhance data privacy and ease the setup of data sharing, we extract different features from logs of device associations and classify them to infer semantic device groups such as personal and stranger's devices. On this basis, we are able to create data sharing policies like sensitive information can be only shared among personal devices.

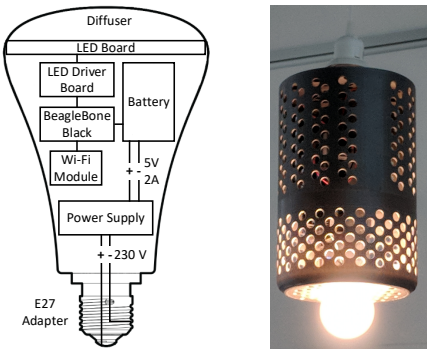
II. Related Work

DevLoc deals with the areas of device coupling, device grouping, device association, and device pairing. Our device association is a guidance technique without human interaction on the basis of user's proximity in the real world. The work of [4] provides an overview by classifying techniques for device grouping in the following way: 1) input aims at user actions like triggering commands, entering data, or direct by manipulating, 2) enrollment uses on-time registration of devices with an identity, 3) guidance takes advantage of users acting in the real world to link devices via contact or alignment, and 4) matching involves different approaches where users compare the output of the involved devices to acknowledge a connection.

Visible light positioning such as in [5]–[7] is out of scope because we are not interested in the user's position to protect the user's privacy. We only need to infer whether users are nearby. Hence, we use context information such as ambient light to recognize proximate devices based on the distance-limited nature of light. In this regard, to detect the co-presence of devices, other system approaches use ambient audio [8], ambient noise and luminosity [9], accelerometer data caused by hand shaking [10], radio signals [11], magnetometer readings of very close devices [12], and gait cycle detection of moving users [13]. The existing work aims to link mainly two devices whereas DevLoc enables group associations. Group association is not simply an extension of pairwise association with additional devices [3]. Rather than multiple device pairings, many people expect that group association is a single-step procedure. The user study of [3] states that



(a) DevLoc's system combines Wi-Fi routers and light bulbs to enable device associations for mobile users and IoT boards.



(b) DevLoc is based on the customized LocalVLC system [14] to realize visible light signaling for device grouping. We present the hardware platform of the deployed 3D-printed light bulb.

Fig. 1: Enable seamless device associations using DevLoc

groupwise associations are not rated highly for simplicity, but close proximity is a popular technique to link devices.

III. Device Associations via DevLoc System

Fig. 1(a) presents the DevLoc framework for device grouping to combine radio-based communication like Wi-Fi covering larger areas and non radio-based communication such as light which is spatially more fine-grained due to walls, doors. We enrich existing lighting with visible light signaling for device grouping. Based on a master-slave principle of the light bulbs, we are able to semantically link multiple rooms or regions and thereby flexibly control the user's spatial granularity. We use different colors in Fig. 1(a) to illustrate varying light patterns at the light bulbs for device associations. The dotted red circles highlight the association among different entities: a) device-to-device using only the device's light signals or b) device-to-area using the device's light signal and an area's reference light signal for signal comparison. The goal of DevLoc is to ease data sharing among mobile user devices like tablets, smartphones, laptops, and static IoT boards. Inspired by [15]–[17] and as central part of DevLoc, our custom light bulb in Fig. 1(b) establishes a Wi-Fi link to the lighting configuration framework and broadcasts light patterns at a high frequency. On this basis, we can replace

existing illumination units and hence we are able to restrict the problem of light pollution, where different visible lights would be overlapping for illumination and communication. We now describe in more detail the setup and working principle of DevLoc.

A. Adaptable Spatial Granularity of Device Grouping

DevLoc allows to select proximity areas to define the geographic structure of the device associations. For example, Fig. 1(a) uses room numbers for area one and region names like corridor for area five. The lighting configuration framework runs at the backend and, initially, each light bulb and Wi-Fi router registers itself with the backend. As a result, DevLoc knows all light bulbs and their specific areas and randomly selects for each region one of the light bulbs as master light bulb, the remaining ones act as slaves. The backend randomly creates a light pattern for each registered master light bulb and the slave(s) broadcast the same light pattern with the master-slave mechanism for the light bulbs. We can dynamically choose the spatial granularity of device proximity by adapting the groups of light bulbs covering different regions. We can use the same light pattern over different rooms which are semantically the same region, e.g., area one in Fig. 1(a) to link two rooms. The size of rooms and regions like corridors, and the number and distribution of light bulbs define the achievable spatial granularity of device groupings. To achieve the most fine-granular user proximity, each light bulb works on its own as master. In our experiments, we identified the communication range of our custom light bulb of up to 10 m. Moreover, the master-slave mechanism of our light bulbs allows a minimum of technical adaptions on existing illumination. Only the master light bulbs need computing power to perform the device groupings, the slave light bulbs require only a radio connection, e.g., Wi-Fi or Bluetooth, to receive the commands from the corresponding master bulb.

B. Triggering Device Grouping

We combine each master light bulb with a Wi-Fi router as a channel to the central configuration framework to maintain light patterns and for later device interaction. The light bulb continuously monitors the wireless connections of the Wi-Fi router and triggers device groupings. Due to the larger Wi-Fi coverage, one router can be combined with multiple master light bulbs. If there are no device groups yet and the Wi-Fi connections are changing, each linked master light bulb requests the continuously broadcasted light pattern received from the client(s). After receiving the client's data, the master light bulb initiates the device grouping to infer which devices are in the same light communication range instead of being only in the same Wi-Fi coverage. In case of a new Wi-Fi client, the master light bulb runs the signal matching to infer the matching device group without affecting other devices. When a Wi-Fi device disappears at the router, the master light bulb deletes this single client from existing device groups.

Moreover, user mobility may also trigger device groupings. For static users who don't move between rooms it is enough to observe the Wi-Fi connections for device grouping. In contrast, we need to manually start the device association via a predefined period, e.g., every few seconds, if users move between multiple regions but still connected to the same Wi-Fi router. To update the device grouping, we do not use signal strength changes of the user's Wi-Fi connection because it can change unexpectedly and yields excessive false positives and false negatives causing frequent device grouping updates.

C. Two Modes for Device Grouping: Device and Area

To support either location-based services (LBS) or proximity-based services (PBS), we are able to specify the mode of device groupings for each master light bulb: device-to-area grouping for LBS and device-to-device grouping for PBS. LBS needs to answer the question "where we are?" based on the absolute position of a user. In contrast, PBS needs to answer the question "who are we with?" based on context information to find co-location with other points of interest. We encounter three main differences between device-to-device and device-to-area groupings: 1) trigger point in time of the device association, 2) required number of user clients for device association, and 3) signal comparison between different entities which affect the resulting binding either device-to-device or device-to-area.

For device-to-device groupings we need at least two connected user clients at the Wi-Fi router to start the device grouping. To link a Wi-Fi client to a specific device group, the master light bulb randomly selects one client from each existing device group for signal matching. The participating user clients only know which other clients are nearby and not at which indoor region they are located. Thereby, we can only realize PBS like data sharing among close-by users and LBS are not feasible, e.g., sharing the menu of the cafeteria since the users are nearby to the canteen, because location-related information is missing using the device-to-device grouping.

For device-to-area groupings, after the user client connected to the Wi-Fi router the corresponding master light bulb(s) immediately start the device association and compare the client's signal to the area's reference signal. We achieve a direct binding between the device and area. Thereby, we know which device is in which area and at the same time which other devices are close-by. There is no limitation with respect to the number of connected user clients, e.g., at least two connected clients for device-to-device association. In general, device-to-device associations provide less location-specific information compared to device-to-area groupings.

D. Generation and Recognition of Light Patterns

Our custom light bulb emits randomly generated light patterns for device associations. We independently create a random series of light on and off periods and combine them resulting in a light pattern. The duration of each light on and off period is in the range of [1, 5] ms. The minimum duration is constrained via the hardware of our light receiver, specifically, how fast the photodiode can be sampled. The maximum duration of each light on and off period is determined by avoiding unpleasant visual experience where light flickering

effects are visible by human eyes. The light sender emits the light pattern in a loop for a restricted amount of time. To be able to differentiate light patterns, the length of the light pattern must be a multiple of two, i.e., after each light on period appears a light off period. To enhance the recognition rate of light patterns at the light receiver, we introduce a 10 % duration difference among light on and off series so that the time periods are sufficiently distinct. The photodiode at the light receiver samples the raw light signal as voltage in mV: a higher voltage refers to a light on period and a lower voltage refers to a light off period.

How to detect reoccurring patterns in the light signal? We use the cycle detection algorithm from [13] to find repeating patterns in our light signal. The algorithm supports signal matching of arbitrary co-aligned sensor data and reaches a reliable signal segmentation based on normalization. The algorithm's input expects a vector of voltage amplitudes $z = (z_1, ..., z_n)$ and the result is a sequence of consecutive light signal patterns. We use auto-correlation and distance calculation to find repeating signal parts. We can efficiently calculate the auto-correlation via the Wiener-Khinchin theorem [18] with complexity $n \cdot log(n)$

$$\begin{aligned} F_R(f) &= FFT[z] \\ S(f) &= F_R(f) \cdot F_R^*(f) \qquad * \hat{=} \text{conjugate} \\ R(\tau) &= IFFT[S(f)] \end{aligned}$$

where z are the voltage amplitudes. The auto-correlation $R(\tau)$ results in m non-ambiguous local maxima $\zeta = \arg\max(R(\tau)) = \{\zeta_1, ..., \zeta_i ..., \zeta_m\}$. We compute the distances among all local maxima and a mean distance

$$\delta_{\text{mean}} = \left\lceil \frac{\sum_{i=1}^{m-1} \zeta_{i+1} - \zeta_i}{m-1} \right\rceil$$

at which δ_{mean} can be used to choose minima indices from z representing signal patterns. To find the local minima μ, every local maximum specifies a start point and δ_{mean} a search range

$$\begin{aligned} \mu &= \{\mu_1, ...\mu_i, ..., \mu_{m-1}\} \\ \mu_i &= \arg\min(z_{\zeta_i}, z_{\zeta_i+1}, ..., z_{\zeta_i+\delta_{\text{mean}}}) \end{aligned}$$

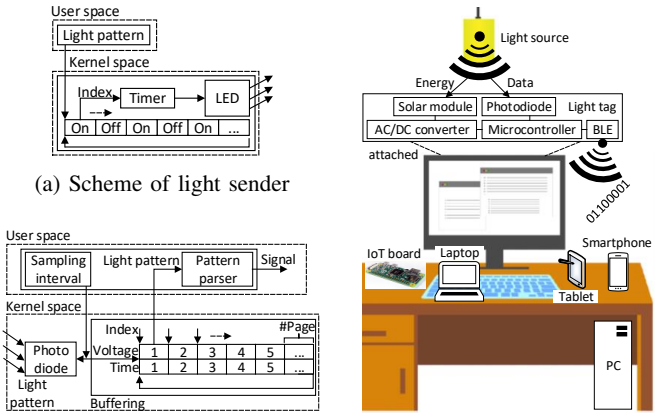
Each μ_j refers to an index of a minimum in z restricted to the range of δ_{mean}. The indices in μ are used to separate the voltage amplitude z into light cycles

$$\begin{aligned} Z &= \{Z_1, ..., Z_i, ..., Z_q\} \\ Z_i &= (z_{\mu_{\frac{i}{2}}}, ..., z_{\mu_i}, ..., z_{\mu_{\frac{i+1}{2}}-1}) \text{ with } i = \{2, 4, ..., q\} \end{aligned}$$

In our experiments, the rate of successfully extracted light patterns decreases significantly in case of sudden changes of light patterns caused by light interference. Therefore, we implement our own method to recognize light cycles considering the period of each light on and off phase. We define the light signal as a list of periods

$$\hat{z} = \{(s_1, d_1), ..., (s_n, d_n)\}$$

where $s_n \in \{0, 1\}$ specifies if the light is on or off and $d_i \in \mathbb{Z}$ refers to the duration of each period. We combine similar signal parts with a difference smaller than 10 % because the light sender introduced a 10 % signal margin between the light on and off periods to improve the robustness of signal pattern detection. The remaining unique signal parts define the signal pattern including the period of each phase. To identify the light pattern, we overlay the light signal with a time window specified by the pattern length which is defined for the system.



(a) Scheme of light sender

(b) Scheme of light receiver

(c) Light tag for ubiquitous visible light signaling

Fig. 2: Implementation details of VLC sender and receiver used by DevLoc. As future work we show the design of a light tag for improved end-device support of VLC.

E. Technical Details of DevLoc

As system and development platform for our custom light bulb, we use the small, low-cost, single-board computer BeagleBone Black. We have implemented two Linux kernel modules to broadcast and receive light patterns. On the sender side, our custom light bulb broadcasts random light patterns via the light sender as shown in Fig. 2(a). The different light on and off periods of the signal pattern trigger two real-time kernel timers to switch the LED between on and off state. The custom light bulb consists of a power supply for the BeagleBone Black and during normal operation the battery is loaded and provides the power for the BeagleBone Black. The battery improves the service availability of DevLoc and maintains the lighting in case of a power blackout. In addition, the BeagleBone Black offers an API for visible light signaling and controls the LED transmitter and wireless modules such as Wi-Fi. On the receiving side, the light receiver in Fig. 2(b) samples the raw light signal via the photodiode. We have tested different sampling intervals, i.e., how often voltage values are sampled, and a sampling rate of 20 µs works reliably to detect light patterns. This affects the signal buffering to store and access light signals from the kernel module. We save voltages and the relative time chunked into pages and control the maximum page size and number of pages based on the sampling rate to provide sufficient information for signal parsing and available system's memory.

We use MQTT for the communication among light bulbs. Via the subscription to the central backend, each master light bulb receives the configured light pattern which is then further published to the slave light bulb(s). Moreover, we use Python

twisted as event-driven network programming framework to receive and send data between light bulbs and user clients. We establish a TLS network connection between the device grouping server and clients. The light bulb uses four different messages to query data for device grouping including raw light signal, detected light pattern, Wi-Fi, and Bluetooth scan data. The clients transmit the data in chunks of lines and the device grouping server buffers for each client the received raw data and merges them before the device association. The message format consists of a message type, a payload length, and the payload itself.

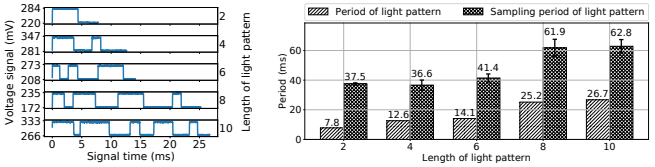
Regarding the VLC receiver, we aim to improve the end-device support for visible light signaling. Many user devices like personal computers do not have the required hardware such as a photodiode to sense light signals, while mobile user devices like smartphone and tablet provide a photodiode, e.g., for ambient light, but lack the ability to process light signals in real-time and require add-on hardware. For future work, we plan to shrink the light receiver to an appropriate (e.g., coin sized) volume for everyday usage towards ubiquitous visible light signaling. The foreseen light tag in Fig. 2(c) acts as proximate communication hub which can be easily attached to different end-user devices enabling light transmissions. The light source acts as energy source and at the same time as transmission medium. Thereby, the light tag works passively meaning it awakes for operation via energy induced by the solar module. During operation the light tag receives and processes light transmitted data in real-time and broadcasts it via Bluetooth Low Energy (BLE) to nearby end-user devices.

IV. Evaluation of Device Associations via DevLoc

We evaluate the propagation characteristics of VLC to qualify the feasibility of real-world deployments of DevLoc. Besides that, we emulate two varying environments with static and moving users to highlight the performance of DevLoc in different environments. We identify, for each case, the best working device association with regard to high detection accuracy and fast runtime. Therefore, we perform a thorough parameter estimation of our device grouping including the sampling periods for light patterns, device localization for comparison, and training classifiers. Moreover, we select the best performing distance and correlation metrics and determine the most suitable time-series features for light patterns.

A. Propagation Characteristics of VLC

With regard to the use of DevLoc in the real world, we have evaluated the maximum attainable range of light patterns for two different LEDs as VLC transmitter in a dark room without interference from the surrounding light [14]. We used two LEDs: an omnidirectional LED with a weak light signal and a directional LED with a strong, beaming light signal. The directional LED reaches a maximum distance of 10 m, while the omnidirectional LED can cover a distance of 3 m. Furthermore, we identified via an experiment the FoV at the photodiode of the VLC receiver, the entire FoV ranges from 180° to 0°. The omnidirectional LED receives a range of 165°–50°, meaning opens at 165° and closes at



(a) Random light patterns (b) Average sampling period

Fig. 3: Sampling periods to detect light patterns with a varying length for device association

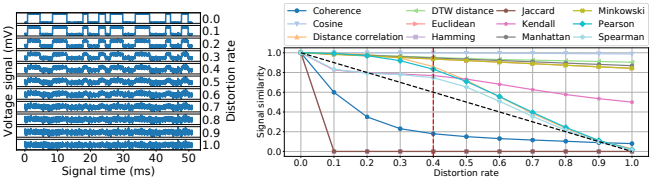
50°, and the directional LED achieves a FoV of 175°–5°. During our experiments, we have recognized that the impact of ambient light is decisive for VLC. Obviously, the directional LED is less sensitive to the ambient light compared to the omnidirectional LED. Nevertheless, with an active light source or direct sunlight acting as ambient light, the performance to detect signal patterns from the directional LED drops significantly. On the other hand, the omnidirectional LED only works reliably at a low ambient light intensity. For future work, we will adopt the algorithm in [19] which uses orthogonal codes to detect and isolate adjacent light sources. Thereby, we plan to enhance the robustness of DevLoc by supporting overlapping light patterns from different light bulbs.

B. Parameter Estimation: Sampling Periods of Light Patterns for Similarity Metrics

Our device grouping takes advantage of random light patterns in different spatial areas to establish device groups for data sharing. Which sampling periods are required to successfully detect light patterns? On the receiver side, we analyze the sampling time to be able to detect a valid light pattern within the time-series of voltage values. The light receiver is only able to recognize the light pattern when it starts repeating. We use five different light patterns for our evaluation with a varying length of light on and off periods as shown in Fig. 3(a). For each specific light pattern and over ten different test rounds, we choose a random start position within the light pattern and we extract a raw voltage signal via a monotonically increasing sampling time until all detected light patterns are valid. We classify a raw light signal as valid if all extracted light patterns have the same length $\in \{2, 4, 6, 8, 10\}$ and the duration of each light on and off phase is above 1 ms known by the generation of light patterns. Fig. 3(b) shows the necessary sampling periods to successfully recognize a reoccurring light pattern compared to the duration of a single light pattern. The sampling time is on average 3.1 times longer than the raw signal pattern. In our evaluation we use the identified sampling ranges for each length of signal pattern to randomly choose a raw voltage signal which is large enough for a reasonable signal comparison.

C. Parameter Estimation: Similarity Metrics for Light Patterns

We identify the best working similarity metrics for light patterns in terms of highest accuracy for group detection. We analyze the behavior of the similarity metrics by comparing



(a) Distorted light signals (b) Signal similarity of distorted signals

Fig. 4: Analysis of signal similarity using distorted light signals for several distance and correlation metrics

TABLE I: Best working similarity and equalize methods for device association including similarity threshold and runtime

Similarity measure	Equalize method	Average metric	Similarity threshold	Runtime
Pearson	DTW	0.93	0.8	0.532 s
Spearman	DTW	0.89	0.9	0.532 s
DTW distance	DTW	0.89	0.7	0.506 s

raw light signals with the same increasingly distorted light signal such as in Fig. 4(a). Per similarity metric and signal distortion rate, the evaluation result in Fig. 4(b) shows the median similarity over ten rounds and each light pattern. The desired property of the similarity metric is that the similarity decreases in case of increasing dissimilarity between two time-series signals. Hence, we identify the following reasonable similarity metrics at a signal distortion rate of 40 %, highlighted as dotted red line in Fig. 4(b): Spearman with a similarity threshold of 0.74, Pearson with a similarity threshold of 0.83, and distance correlation with a similarity threshold of 0.86.

Besides that, we simulate a testbed with two clients for device grouping where we perform ten evaluation rounds for each combination among all light patterns $\in \{2, 4, 6, 8, 10\}$ to find the best working similarity metric and equalize method to unify signal lengths being able to compare them. In each run, we apply the similarity metrics mentioned in Fig. 4(b) and, as input, we randomly choose two light patterns and equalize their signal length using the methods $\in \{$fill, cut, dynamic time warping (DTW)$\}$. Table I presents the three best working combinations of similarity metric, equalize method, and signal threshold in terms of highest average metric including accuracy, precision, recall, and F1-score weighted with 0.8 and lowest runtime weighted with 0.2. To calculate the result metrics, we assume that the two simulated clients are in the same region if the light signals have the same repeating pattern. We use these identified best working similarity measures to limit the runtime of our simulations for device grouping.

D. Parameter Estimation: Device Localization as Reference for Device Grouping

We use well-known device localization as reference to compare the results of our device grouping based on light patterns. We include a common indoor localization based on the similarity of Wi-Fi and Bluetooth signals [20]. Via an Android app we gather a list of Wi-Fi router and Bluetooth beacons

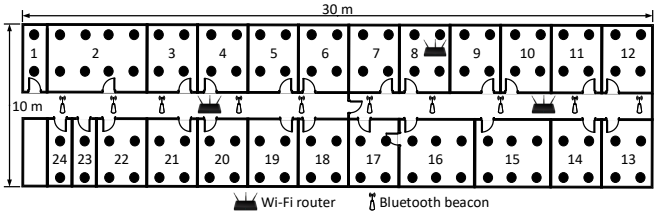
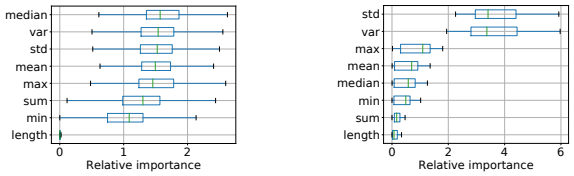


Fig. 5: We model our university lab as simulation environment for device associations. For comparison with device localization, we take real traces of the Wi-Fi and Bluetooth environment at different positions.



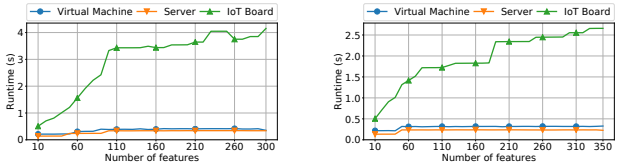
(a) Single light patterns for dynamic device-to-area simulation (b) Combination of light patterns for static device-to-device simulation

Fig. 6: Selection of statistical features for light patterns via relative importance of trained machine learning models

containing MAC addresses and signal strengths (RSSI) for each measurement point (●) at our university lab as shown in Fig. 5. We evaluate the sampling period, i.e., how many traces are required to achieve a reasonable localization accuracy. For a supervised machine learning approach with 10-fold cross validation, we have taken the following feature subset from the work in [21] to compare the list of Wi-Fi routers seen at two different measurement points, the same applies for the list of Bluetooth beacons:

- Number of overlapping devices
- Size of the union of the two lists
- Jaccard distance between the size of the intersection and the size of the union of the two lists
- Number of non-overlapping devices
- Manhattan distance of RSSI of overlapping devices
- Euclidean distance of RSSI of overlapping devices
- Spearman correlation of RSSI of overlapping devices
- Pearson correlation of RSSI of overlapping devices
- Share top device based on strongest RSSI
- Share at least one top device based on RSSI range $\pm$ 6 dB

We merge the different measurements for each room and perform binary classification among all rooms with multiple sampling periods $\in$ [2, 5, 10, 15, 20, 25, 30] s applying content-based filtering, support vector machine (SVM), and random forest. The content-based filtering uses the shortest cosine distance among room measurements to identify the user's room. On average, the sampling period with 5 s achieves the highest accuracy for Wi-Fi and Bluetooth localization.



(a) Single light patterns for dynamic device-to-area simulation (b) Combination of light patterns for static device-to-device simulation

Fig. 7: Runtime analysis of tsfresh features for light patterns to highlight the performance difference among test platforms

TABLE II: Testbed specifications for runtime analysis of feature selection used by device grouping

System	CPU	RAM
Server	40x Intel Xeon E5-2630 2.2 GHz	768 GB
Virtual machine (VM)	4x Intel Xeon E5-2630 2.2 GHz	32 GB
Next unit of computing PC (NUC)	4x Intel Core i5-6260U 1.8 GHz	16 GB
IoT board	1x ARM AM3358 1 GHz	0.5 GB

E. Parameter Estimation: Feature Selection for ML-based Device Grouping

Meaningful features are important for device grouping based on machine learning (ML) to achieve a good performance. Our feature selection identifies the features with highest entropy, i.e., information content, and lowest runtime. To find the most robust features in terms of distorted light patterns, we include light patterns with increasing white noise from 0 % to 100 %.

Raw light patterns for different simulations We use the combination of light patterns with different lengths for the static device-to-device simulation of our device grouping because it consists of only one room where we change the light pattern over time to keep the device groups up-to-date. Each master light bulb performs the proximity reasoning and is trained with a combination of light patterns with different lengths $\in \{2, 4, 6, 8, 10\}$. In contrast, we use single light patterns for the dynamic device-to-area simulation of device grouping because it contains several rooms where the associated light pattern for each room remains the same over time. Each master light bulb is trained only for a specific, single light pattern with the same length.

Feature types We compute statistical features and time-series tailored features via tsfresh [22] for single and combination of light patterns. Tsfresh performs a time series feature extraction on basis of scalable hypothesis tests combining 63 time series characterization methods to identify the most meaningful features from a total of 794 time series features.

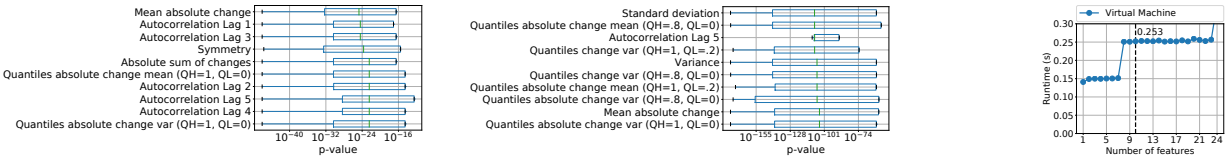
Best statistical features For feature selection of statistical features we take advantage of three different machine learning models: extra trees, gradient boosting, and random forest to identify the most important features. Fig. 6 shows the average relative importance of each statistical feature. In case of single light patterns, the relative importance is uniformly distributed over all features and only the feature length does not provide sufficient entropy. On the other hand, with the combination of light patterns, the variance and standard deviation outperforms all other features by 68 %.

Runtime analysis of time-series features to select simulation platform We apply tsfresh for time-series feature selection to identify the most meaningful features of light patterns. Thereby, we highlight the runtime of feature calculation to evaluate the practicality of our proposed system where we aim to run the device associations directly at the light bulb which embeds an IoT board with limited hardware capabilities. Fig. 7 presents the feature runtime for single and combination of light patterns on three different test platforms described in Table VII. We sort the features according to their decreasing information content based on the hypothesis testing from tsfresh. For runtime comparison we calculate the relative runtime $= {}^{\text{runtime}}/_{\text{number of features}}$ among the test platforms. In case of the single light patterns, tsfresh computes 300 features where the server achieves the fastest performance with a relative runtime of 2.63 ms per feature, the virtual machine is about 31 % slower with 3.46 ms per feature, and the IoT board is 6.2 times slower with 21.28 ms per feature. Using the combination of light patterns we achieve a similar runtime where tsfresh calculates 350 different time-series features. The server reaches a relative runtime of 2.11 ms per feature, the virtual machine with 3.17 ms, and the IoT board is by far the slowest platform with 14.99 ms per feature. Based on our findings from the feature runtime, we choose the virtual machine (VM) as simulation platform to evaluate DevLoc because the IoT board as desired platform is too slow when we repeat the device grouping at a higher rate. Note that this is just for the purpose of evaluation, the IoT board is still considered in other parts of the evaluation.

Best time-series features To improve the detection accuracy and limit the runtime of our device grouping, we select the ten most meaningful features of light patterns which are shown in Figs. 8(a) and 8(b). We use the p-value or probability value from tsfresh to select the most important features, the probability of finding the observed results when the null hypothesis is true. The lower the p-value the more significant is the feature. To ensure a reasonable performance for device grouping, Fig. 8(c) presents a detailed runtime analysis of features of light patterns computed at the virtual machine used as simulation platform for device grouping. It takes 253 ms to compute the ten most meaningful features for device associations which is fast enough to ensure the validity of our simulation results considering that we repeat device grouping every few seconds among multiple users.

F. Parameter Estimation: Sampling Period of Light Patterns to Train ML-based Device Grouping

We perform an offline evaluation of our device grouping to identify the best working sampling period for light patterns to train different classifiers. In our experiment, the device association ranges between two and ten grouping clients and we apply 10-fold cross validation for the classifiers: extra trees, gradient boosting, and random forest. These are trained via sampling periods $\in [30, 120]$ ms for different light patterns with signal length $\in \{2, 4, 6, 8, 10\}$ and selected tsfresh

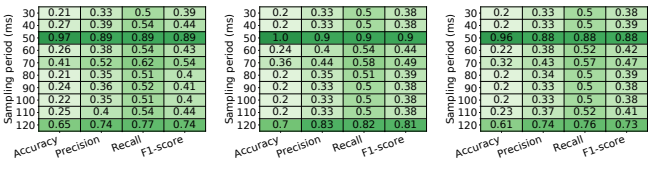


(a) Best ten features of single light patterns for dynamic device-to-area simulation

(b) Best ten features for combination of light patterns for static device-to-device simulation

(c) Detailed runtime analysis of best time-series features at our simulation platform

Fig. 8: Via tsfresh we select the ten most meaningful features of each light pattern type for device grouping. Moreover, we highlight the runtime to calculate these features at the virtual machine used as simulation platform for device grouping.



(a) Extra trees (b) Gradient boosting (c) Random forest

Fig. 9: Parameter estimation of sampling period to train ML-based device association

features from Fig. 8. We take the average results of our ML-based device grouping including accuracy, precision, recall, and F1-score. Fig. 9 shows that the sampling period of 50 ms achieves the best result with 0.91 over all classifiers compared to a sampling period of 120 ms with 0.74. The classifiers: extra trees, gradient boosting, and random forest achieve similar results.

G. Simulation Parameters for Device Grouping

To evaluate DevLoc, we run two different simulations with static and dynamic users based on a dedicated simulator. Table III shows the summarized best working parameters for device grouping (italicized). With persisted environmental data from our university lab as shown in Fig. 5, we perform a trace-driven simulation where each grouping client uses three different real traces: Wi-Fi and Bluetooth scans, and random light patterns with varying length. To achieve a realistic simulation, we emulate the network latency between the grouping server and the clients. Each client waits a random time within a predefined time range before sending the requested environment data to the grouping server. Thereby, we select a random start time within the sensing range for light patterns, Wi-Fi and Bluetooth scans. In addition, we randomly choose a sampling period within the identified best working sampling ranges.

H. Static Device-to-Device Simulation of Device Grouping

Simulation settings No user in the static simulation moves and every user stays in the same room. The grouping server immediately starts the device grouping when all devices are connected. The parameters for the static simulation are shown in Table III. We run the device grouping using random light patterns with different length $\in \{2, 4, 6, 8, 10\}$ and from at least two users up to ten users.

TABLE III: Summarized settings from parameter estimation (highlighted in italic) and simulation parameters (highlighted in bold) for device grouping.

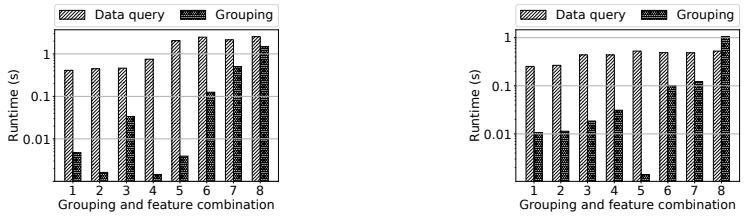
Simulation	Parameters	
Static & Dynamic	*Sampling period to train similarity classifiers*	50 ms
	Similarity classifiers	Random forest, extra trees, gradient boosting
	Sampling period localization	5 s
	Localization classifiers	Content-based filtering, random forest, SVM
	Similarity equalize method	DTW
	Similarity threshold	0.7
	Similarity metrics	Pearson, Spearman
Dynamic	**Rooms**	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
	Users	[3, 5, 10]
	Grouping frequency	[10, 20, 30] s
Static	**Users**	[2, 3, 4, 5, 6, 7, 8, 9, 10]
	Light patterns	[2, 4, 6, 8, 10]

Simulation results Through 10-fold cross-validation, highlighted in bold in Table IV, we identify the best working device grouping technique in relation to a fast reasoning and a reasonable overall result, i.e., the average over accuracy, precision, recall, and F1-score. The device localization using Bluetooth and Wi-Fi features works worst whereas ML-based device grouping performs similar or slightly better than the signal similarity metrics such as Spearman and Pearson. Moreover, Fig. 10(a) presents the runtime of each method for device grouping sorted in ascending order. To perform device grouping, it takes around 1.41 s to receive data (83.93 % of total time) compared to the actual device grouping with 0.27 s (16.07 % of the total time).

For a thorough evaluation, we further analyze the performance of light patterns with different lengths for device grouping. The light pattern with four random on and off periods works best compared to a decrease of 9 % using the worst light pattern with ten random periods. The performance of light patterns with different lengths sorted by descending total result in brackets, i.e., average over accuracy, precision, recall, and F1-score: 4 (0.97), 2 (0.95), 6 (0.91), 8 (0.9), and 10 (0.88). Furthermore, we evaluate the performance using different number of grouping users. With six users we reach

TABLE IV: We identify best working classifiers and features for device grouping via simulations with static and moving users

Simulation	Grouping technique	Feature type	Result	Runtime	Accuracy	Precision	Recall	F1-score
Dynamic	Extra trees	Statistical	.83	0.26 s (1)	.75	1	.75	.83
	Content-based filtering	Wi-Fi	.84	0.61 s (7)	1	.78	.78	.78
	Content-based filtering	Bluetooth	.84	0.59 s (6)	.95	.81	.81	.81
	Gradient boosting	Selected statistical	.93	0.53 s (5)	.9	1	.9	.93
	Gradient boosting	Selected tsfresh	.93	1.58 s (8)	.9	1	.9	.93
	Pearson	**Light signal**	**.95**	**0.28 s (2)**	**.95**	**.95**	**.95**	**.95**
	Pearson	Light pattern	.95	0.47 s (4)	1	.93	.93	.93
	Pearson	Duration of light pattern	.95	0.46 s (3)	1	.93	.93	.93
Static	SVM	Wi-Fi	.31	2.61 s (6)	.32	.2	.44	.26
	Random forest	Bluetooth	.43	2.64 s (7)	.34	.48	.52	.38
	Pearson	Light signal	.64	2.06 s (5)	.25	.77	.79	.76
	Spearman	Duration of light pattern	.81	0.42 s (1)	1	.75	.75	.75
	Spearman	Light pattern	.81	0.49 s (3)	1	.75	.75	.75
	Gradient boosting	**Statistical**	**.81**	**0.45 s (2)**	**1**	**.75**	**.75**	**.75**
	Gradient boosting	Selected statistical	.83	0.75 s (4)	.89	.81	.81	.8
	Gradient boosting	Selected tsfresh	.96	4.02 s (8)	1	.94	.94	.94



(a) Static device-to-device grouping (b) Dynamic device-to-area grouping

Fig. 10: Runtime analysis of different device groupings and signal features. To identify the grouping technique and feature type, the number of grouping and feature combination matches with the runtime order in Table IV mentioned in brackets.

the highest total result of 0.97 because with less users the grouping signals miss crucial patterns and with more users the noise in the grouping signals grows which leads to a higher error rate for device associations. In detail, we show the number of grouping users with descending total result in brackets: 6 (0.97), 4 (0.94), 3 (0.93), 8 (0.93), 5 (0.92), 9 (0.92), 10 (0.92), 7 (0.92), and 2 (0.87).

I. Dynamic Device-to-Area Simulation of Device Grouping

Simulation settings In contrast to the static simulation, in the dynamic device-to-area simulation the users are moving between different rooms receiving varying light patterns. Our modeled simulation environment for device associations is shown in Fig. 5. The rooms are positioned in a rectangular arrangement with an inter room distance of 3 m and intra room distance of 2 m. We compute the distances among all room combinations. Using the duration of one simulation iteration of 20 min, we calculate a random path between the rooms for each user. Thereby, we distribute the random time as duration of stay over different rooms using a multinomial distribution. As a result, the user's random path is a list of tuples with duration of stay and room, e.g., user A has the random path [(1, 120), (3, 300), ...]. This means that the start position is in room 1 and after 120 s the user moves to room 3 and stays there for 5 min, and so forth. Hence, we randomly create user groups for each room at a specific time and for each movement between rooms every user chooses a random movement speed in the range of 1.25 to 1.53 m/s (4.5–5.5 km/h) [23]. If the users are in motion between two rooms they are in the corridor and not associated with any room. For device grouping, each room acts independently of other rooms and is linked with a unique location-dependent environment data containing Wi-Fi and Bluetooth scans, and light patterns. The overview of parameters for dynamic device-to-area simulation in Table III covers grouping frequency, number of users, and number of rooms.

Simulation results Table IV shows the best working device grouping (highlighted in bold) using 10-fold cross validation in terms of a fast runtime and a reasonable overall result, meaning the average over accuracy, precision, recall, and F1-score. Compared to the static device-to-device simulation, the device grouping based on similarity metrics works slightly better than ML-based device grouping. Further, the device localization using Wi-Fi and Bluetooth features reaches a similar result. The runtime of each method for device grouping is shown in

Fig. 10(b) with ascending runtime from 0.26 s to 1.58 s. The median time is around 0.43 s (71.67 % of the total time) to receive data for device grouping whereas the device grouping itself lasts 0.17 s (28.33 % of the total time). Besides that, the frequency of device grouping with 20 s works best, the accuracy of device grouping decreases by 16 % with 30 s and with a 10 s frequency the accuracy decreases another 8 %. Furthermore, we analyze the performance of device grouping with a varying number of rooms, sorted after decreasing overall result in brackets: 1 (0.99), 2 (0.96), 3 (0.92), 5 (0.9), 6 (0.89), 4 (0.87), 8 (0.84), 7 (0.82), 9 (0.79), and 10 (0.76). The device grouping works best with less rooms because the more rooms the higher the risk that the user miss the up-to-date light pattern of the designated room due to movement between rooms.

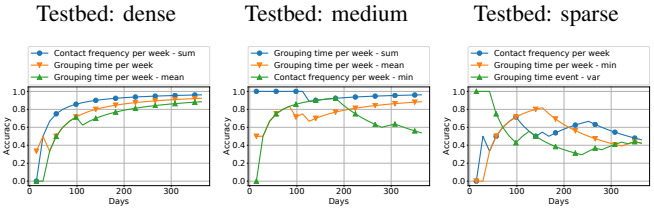
Summarizing, Table IV presents two different best working classifiers and features for device grouping depending on the use case either with static or moving users. In both cases, we have an equal distribution between machine learning based and similarity based device grouping. Moreover, similar grouping techniques perform best, mainly the feature types are changing in varying conditions. Nevertheless, we favor the approach for device grouping in the scenario with multiple moving users because it is more realistic in practice.

V. Practical Extension of DevLoc

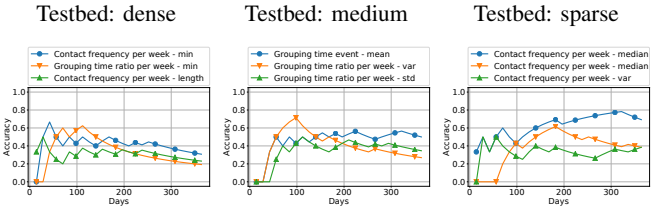
DevLoc provides the basis to connect devices for data sharing among users. As a practical extension to enhance data privacy and ease the setup of data sharing, we analyze logs of device associations in terms of grouping patterns, e.g., time and frequency, to detect semantic device groups like personal and stranger's devices. On this basis, we are able to create automated data sharing policies, such as sharing sensitive data only among personal devices.

A. Artificial Log of Device Associations

In practice, DevLoc records device associations for further analysis to find semantic device groups. For evaluation we generate an artificial log of device associations to be able to analyze the device groups across different testbeds instead of using real-world data sets [24], [25] of social networks. To generate the log of device associations, we define a calendar for the simulation time including days $\in \{$all, holiday, weekend, workday$\}$ and time slots structuring the hours of the day: all $\in [0, 24]$, night morning $\in [0, 5]$, morning $\in [5, 10]$, forenoon $\in [10, 12]$, noon $\in [12, 14]$, afternoon $\in [14, 17]$, evening $\in [17, 21]$, and evening night $\in [21, 24]$. On this basis, we specify three different device groups $\in \{$personal, family & friends, well-known & stranger$\}$ with corresponding time encounter rules. For personal and family & friends devices: morning[workdays], evening[workdays], noon[all], afternoon[all], evening[all] in contrast to well-known & stranger devices where all encounter times are allowed: all[all], i.e., entirely random device encounters. We determine the possible device encounters based on the corresponding rule of the device group and then randomly distribute the device encounter time over the simulation duration, in our case one year (365 days). Finally, we clean the generated log by removing log entries with single and duplicated devices.



(a) Detection accuracy of the three best performing features to cluster devices into three single device groups



(b) Detection accuracy of the three best performing features to cluster devices into seven mixtures of device groups

Fig. 11: Detection granularity of different device groups over time: personal, family & friends, and well-known & stranger across testbeds with varying numbers of devices: dense, medium, and sparse.

B. Semantic Log Analysis of Device Associations

Features for Log Analysis We create 43 different feature sets of three or ten dimensional features using the following information:

- how much time the devices are associated together per event, per day, and per week,
- how often the devices are associated together per day and per week, and
- the time ratio between association time and entire time frame per day and per week.

In addition, we calculate statistical measures like average and standard deviation of multiple combined features per event and per week. We use the statistical measures all together in a ten dimensional feature vector or we treat them individually as three dimensional feature vectors.

Testbeds for Log Analysis To simulate different test environments and compare the results, we introduce three environments with varying numbers of devices per device group and different grouping times, could be also more groups. The sparse environment includes three devices per device group and a grouping time in the range [10, 60] min, the medium environment has six devices per group and a grouping time of [20, 120] min, and the dense environment uses a grouping time of [30, 180] min and nine devices per device group.

Detection granularity of device groups over time Fig. 11 shows the clustering accuracy over time for each test environment with single or mixtures of device groups. We sort each cluster estimation after the mean accuracy in descending order and select the three best performing clustering methods.

TABLE V: Best working classifiers and features to predict semantic device groups

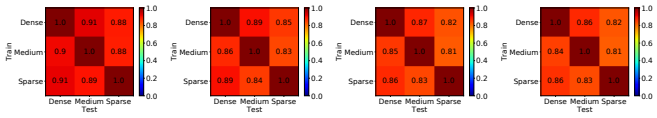
Device group	Testbed	Classifier	Feature type	AUC	Cold start (days)	Accuracy	Precision	Recall	F1-score
Personal	Dense	Naive bayes	Contact frequency per week - sum	.99	23	.95	.92	.93	.92
	Medium	Extra trees	Grouping time per week - mean	.97	17	.93	.87	.92	.89
	Sparse	Ada boost	Contact frequency per week	.98	56	.96	.94	.93	.94
Family & Friends	Dense	Naive bayes	Contact frequency per week - sum	.99	17	.94	.92	.92	.92
	Medium	Gradient boosting	Grouping time per week - std	.85	72	.89	.84	.81	.83
	Sparse	Ada boost	Contact frequency per week	.98	55	.96	.94	.95	.95
Well-known & Stranger	Dense	Ada boost	Grouping time per week - sum	.98	16	.97	.97	.93	.95
	Medium	Ada boost	Contact frequency per week	.99	22	.97	.95	.95	.95
	Sparse	Ada Boost	Contact frequency per week	.99	45	.98	.97	.96	.96

Our results show that in the dense and medium testbeds, we are able to reliably identify the three single device groups, in the sparse testbed the input data is not sufficient to estimate the device groups over time. The same holds true to detect the seven mixtures of device groups, over all testbeds it is impossible to reliably recognize the device groups. Our further analysis is restricted to three single device groups, if we can predict these device groups, we can also define data policies for a mixture of them.

In detail, at each time step t we calculate the accuracy based on the estimated number of clusters via different clustering methods and the true number of clusters. Thereby, we highlight the temporal behavior to detect the number of device groups based on the log of device associations. Our log includes either three single device groups: personal, family & friends, well-known & stranger or seven mixtures of device groups, e.g., personal + family & friends. We use the following clustering methods to the range between 2 to 9 clusters:

- K-Means with elbow method using the total within sum of squares to measure the cluster compactness
- K-Means using silhouette score
- Hierarchical clustering using silhouette score
- Gaussian mixture using Bayesian Information Criterion
- Gaussian mixture using Akaike Information Criterion
- X-Means using K-Means++ for initial cluster centers

Find the best working features and classifiers to predict device groups Table V presents for each device class and testbed the best working classifier and feature type including the average area under the curve (AUC), cold start in days, accuracy, precision, recall, and F1-score. We model the problem to identify the best working classifiers and features to predict device groups as multi-class classification with the following device classes: 0 – personal, 1 – family & friends, and 2 – well-known & stranger. In total, we use 43 feature sets with a series of different classifiers via 10-fold cross validation. The classifiers include extra trees, gradient boosting, SVM, random forest, naive Bayes, and Ada boost. The cold start is the success criterion meaning after which time we are able to reliably predict the device class, the earlier the better. The cold start is defined by a threshold of 80 %, from this point in time (day), all result metrics including accuracy, precision, recall, and F1-score are above this threshold. Furthermore, we consider only prediction results with a cold start in the first quarter of the overall timeline of one year.



(a) Accuracy (b) Precision (c) Recall (d) F1-score

Fig. 12: Average prediction result of the best performing classifiers and features to infer semantic device groups across testbeds with different number of devices

Stability to predict devices groups over time and across testbeds Table VI shows the most common combinations of classifier and feature per train and test environment among best performing prediction methods for semantic device groups. To be specific, we identify which combination of classifier and feature type works best over several test environments with varying numbers of devices and different device encounter times. Via 10-fold cross-validation, we train and test the classifier for each combination of {dense, medium, sparse} testbed and time of testing. Figs. 12(a) to 12(d) illustrate the average prediction result of the best performing classifiers and features, i.e., highest average detection accuracy over all device classes: personal, family & friends, and well-known & stranger to predict semantic device groups.

We encounter a potential privacy leakage by storing and analyzing device associations to infer semantic device groups. Countermeasures can be the anonymization of device activity logs, store the activity logs for a short period of time, and blockchain-secured device logging, or perform analysis in home only locally on the access point.

We conclude the evaluation of DevLoc to enable seamless device grouping based on visible light signaling. First, we analyze the characteristics of the VLC physical channel and afterwards we perform a thorough parameter estimation including pre-selection of distance and correlation measures for light patterns and feature selection for ML-based device association. On this basis, we run two device grouping simulations with a single room and static users and several rooms with moving users to find for each case the best working device grouping method. Besides that, we use the log of device associations to

TABLE VI: Most common combination of classifier and feature among best prediction methods for semantic device groups

Train \ Test	Dense	Medium	Sparse
Dense	Ada Boost + Contact frequency per day	Naive Bayes + Grouping time ratio per week	Naive Bayes + Grouping time ratio per week
Medium	SVM + Grouping time per week - min	Ada Boost + Contact frequency per day	SVM + Grouping time per week - mean
Sparse	SVM + Grouping time per week	SVM + Grouping time per week	Ada Boost + Contact frequency per day

infer semantic device groups like personal, family or stranger's devices to support data sharing policies, with whom sharing which data. For future extension, we plan to calculate a trust score [26] based on the log of device interactions from DevLoc to further enrich the semantic meaning of devices regarding allowed data sharing among devices.

VI. Discussion

Due to the configuration framework of DevLoc based on visible light signaling integrated in surrounding lighting, we can support fine-granular device associations per room or region. In this way, we can realize our previously defined use cases for end users and IoT applications. By using the similarity of distance-limited light patterns, we are able to help Alice asking for the best way on the subway or Carol to record who is present at here meetings. Combined with LocalVLC [14], to incorporate our Morse-code inspired modulation scheme that can operate on off-the-shelf LEDs with low energy overhead, we are able to transmit data via light, e.g., location identifier, and not only light patterns. Thereby, we can support IoT applications, such as merging and filtering of location-tagged data from IoT boards and location-based access policies for consumer smart home platforms.

To enhance the user's privacy for DevLoc, we will use private proximity testing at the light bulb during device grouping. Thereby, as part of the secure multi-party computation (SMC) problem, multiple parties are able to compute whether they are nearby without learning each other's inputs. Homomorphic encryption [27] and garbled circuit [28] are two main techniques to solve the SMC problem, at which homomorphic encryption is more efficient compared to garbled circuit [29]. Usually homomorphic encryption is applied to a small amount of data, e.g., latitude and longitude, to compute the distance between two points of interest. In the appendix we analyze the runtime of fully homomorphic encryption for time-series data such as light patterns and we pinpoint the need of new cryptographic primitives for practical use.

Moreover, we plan to enhance DevLoc to be more resilient against adversaries performing relay attacks to trick our device grouping to include distant clients into the device group by relaying location-dependent light patterns from the intended space to remote clients. Besides that, we have to consider mitigation strategies against other attacks like spoofing the area's light reference signal that clients use for association or periodically starting the device grouping every few seconds may introduce a frequent opportunity for potential attackers to identify themselves as part of the group.

VII. Conclusion

DevLoc is a ready-to-use system solution that provides a seamless device grouping based on visible light signaling for data sharing. Our customized light bulb transmissions allows clients to detect cycles in the light patterns for device grouping. We perform a thorough evaluation of DevLoc via two different simulations with a single room and static users and multiple rooms with moving users. Thereby, we analyze the performance of several device grouping methods: signal similarity based on distance and correlation metrics, ML-based signal similarity, and as baseline the device localization using Wi-Fi and Bluetooth traces. Finally, we take advantage of the device grouping log to infer semantic device groups like personal, family or stranger's devices to enhance the data privacy, with whom sharing which data.

References

[1] A. Narayanan, N. Thiagarajan, and M. Lakhani, "Location Privacy via Private Proximity Testing," in *Proceedings of the 18th Annual Network and Distributed System Security Symposium (NDSS)*, 2011.

[2] S. Mare, L. Girvin, F. Roesner, and T. Kohno, "Consumer Smart Homes: Where We Are and Where We Need to Go," in *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2019, pp. 117–122.

[3] M. K. Chong and H. W. Gellersen, "How Groups of Users Associate Wireless Devices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2013, pp. 1559–1568.

[4] M. K. Chong, R. Mayrhofer, and H. Gellersen, "A Survey of User Interaction for Spontaneous Device Association," *ACM Computing Surveys*, vol. 47, no. 1, pp. 1–40, 2014.

[5] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor Positioning with Mobile Phones and Visible Light," in *Proceedings of the 20th International Conference on Mobile Computing and Networking (MobiCom)*, 2014, pp. 447–458.

[6] Z. Yang, Z. Wang, J. Zhang, C. Huang, and Q. Zhang, "Lightweight Visible Light Positioning for Wearables," *GetMobile: Mobile Computing and Communications*, vol. 19, no. 3, pp. 18–21, 2015.

[7] ——, "Wearables Can Afford: Light-Weight Indoor Positioning with Visible Light," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2015, pp. 317–330.

[8] D. Schürmann and S. Sigg, "Secure Communication Based on Ambient Audio," *Secure Communication Based on Ambient Audio*, vol. 12, no. 2, pp. 358–370, 2013.

[9] M. Miettinen, N. Asokan, T. D. Nguyen, A.-R. Sadeghi, and M. Sobhani, "Context-Based Zero-Interaction Pairing and Key Evolution for Advanced Personal Devices," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2014, pp. 880–891.

[10] R. Mayrhofer and H. Gellersen, "Shake Well Before Use: Intuitive and Secure Pairing of Mobile Devices," *IEEE Transactions on Mobile Computing*, vol. 8, no. 6, pp. 792–806, 2009.

[11] A. Varshavsky, A. Scannell, A. LaMarca, and E. de Lara, "Amigo: Proximity-Based Authentication of Mobile Devices," in *Proceedings of the 9th International Conference on Ubiquitous Computing (UbiComp)*, 2007, pp. 253–270.

[12] R. Jin, L. Shi, K. Zeng, A. Pande, and P. Mohapatra, "MagPairing: Pairing Smartphones in Close Proximity Using Magnetometers," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1306–1320, 2016.

[13] D. Schürmann, A. Brüsch, N. Nguyen, S. Sigg, and L. Wolf, "Moves like Jagger: Exploiting variations in instantaneous gait for spontaneous device pairing," *Pervasive and Mobile Computing*, vol. 47, pp. 1–12, 2018.

[14] M. Haus, A. Y. Ding, and J. Ott, "LocalVLC: Augmenting Smart IoT Services with Practical Visible Light Communication," in *Proceedings of the 20th IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2019, pp. 1–9.

[15] S. Schmid, J. Ziegler, G. Corbellini, T. R. Gross, and S. Mangold, "Using Consumer LED Light Bulbs for Low-Cost Visible Light Communication Systems," in *Proceedings of the 1st ACM MobiCom Workshop on Visible Light Communication Systems (VLCS)*, 2014, pp. 9–14.

[16] Gummeson, Jeremy, J. McCann, C. Yang, D. Ranasinghe, S. Hudson, and A. Sample, "RFID Light Bulb: Enabling Ubiquitous Deployment of Interactive RFID Systems," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 1, no. 2, 2017.

[17] S. Schmid, T. Bourchas, S. Mangold, and T. R. Gross, "Linux Light Bulbs: Enabling Internet Protocol Connectivity for Light Bulb Networks," in *Proceedings of the 2nd International Workshop on Visible Light Communications Systems (VLCS)*, 2015, pp. 3–8.

[18] L. Cohen, "Generalization of the Wiener-Khinchin Theorem," *IEEE Signal Processing Letters*, vol. 5, no. 11, pp. 292–294, 1998.

[19] A. U. Guler, T. Braud, and P. Hui, "Spatial Interference Detection for Mobile Visible Light Communication," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2018, pp. 1–10.

[20] H. S. Maghdid, I. A. Lami, K. Z. Ghafoor, and J. Lloret, "Seamless Outdoors-Indoors Localization Solutions on Smartphones," *ACM Computing Surveys*, vol. 48, no. 4, pp. 1–34, 2016.

[21] P. Sapiezynski, A. Stopczynski, D. Kofoed Wind, J. Leskovec, and S. Lehmann, "Inferring Person-to-person Proximity Using WiFi Signals," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 1–20, 2017.

[22] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package)," *Neurocomputing*, vol. 307, pp. 72–77, 2018.

[23] N. Carey, "Establishing Pedestrian Walking Speeds," *Portland State University*, 2005.

[24] S. Firdose, L. Lopes, W. Moreira, R. Sofia, and P. Mendes, "CRAWDAD dataset copelabs/usense (v. 2017-01-27)," Downloaded from https://crawdad.org/copelabs/usense/20170127, Jan. 2017.

[25] R. I. Ciobanu and C. Dobre, "CRAWDAD dataset upb/hyccups (v. 2016-10-17)," Downloaded from https://crawdad.org/upb/hyccups/20161017, Oct. 2016.

[26] D. Chatzopoulos, M. Ahmadi, S. Kosta, and P. Hui, "OPENRP: A Reputation Middleware for Opportunistic Crowd Computing," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 115–121, 2016.

[27] C. Gentry, "Fully Homomorphic Encryption using Ideal Lattices," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, M. Mitzenmacher, Ed., 2009, pp. 169–178.

[28] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (SFCS)*, 1982, pp. 160–164.

[29] P. Hallgren, M. Ochoa, and A. Sabelfeld, "InnerCircle: A Parallelizable Decentralized Privacy-Preserving Location Proximity Protocol," in *Proceedings of the 13th Annual Conference on Privacy, Security and Trust (PST)*, 2015, pp. 1–6.

[30] "Ibm," https://github.com/homenc/HElib, 2019.

[31] "Microsoft SEAL (release 3.2)," https://github.com/Microsoft/SEAL, Feb. 2019, microsoft Research, Redmond, WA.

APPENDIX

Practicality of Fully Homomorphic Encryption for Private Proximity Testing To protect the user's privacy during device grouping at the light bulb, we apply private proximity testing as an instance of the secure multi-party computation (SMC) problem where multiple parties compute whether they are nearby within a specific distance threshold without learning each other's inputs. This protects the location data, e.g., light patterns, against a wide range of attacks, because it reveals no sensitive information to anyone. Homomorphic encryption [27] and garbled circuit [28] are two main techniques to solve the SMC problem, at which homomorphic encryption is more efficient compared to garbled circuit [29].

Our system model for proximity detection consists of a trusted party, a service provider, mobile users, and static IoT devices. The service provider is usually considered untrusted and should not learn the proximity test results. Our custom light bulb acts as trusted party and service provider, e.g., data sharing, among the nearby mobile devices. Usually homomorphic encryption is applied to a small amount of data to compute the distance between two points of interest, e.g., latitude and longitude of attractions like for LBS. In contrast, we analyze the runtime of fully homomorphic encryption for time-series data such as light patterns whether it is practically usable.

Testbeds To evaluate the performance of homomorphic encryption, we use the libraries: HElib [30] and SEAL [31] to compute the euclidean and cosine distance as similarity measure between two light patterns on three different platforms: server, NUC, and IoT board as described in Table VII.

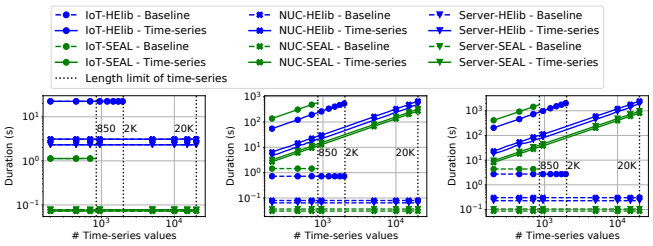
TABLE VII: Testbed's hardware specifications for runtime analysis of fully homomorphic encryption

System	CPU	RAM
Server	40x Intel Xeon E5-2630 2.2 GHz	768 GB
Virtual machine (VM)	4x Intel Xeon E5-2630 2.2 GHz	32 GB
Next unit of computing PC (NUC)	4x Intel Core i5-6260U 1.8 GHz	16 GB
IoT board	1x ARM AM3358 1 GHz	0.5 GB

Performance results Figs. 13(a) to 13(c) show the duration to initialize two different time-series each with a size in the range of [200, 21k] and compute the euclidean and cosine distance for proximity detection. We encounter data limits due to restricted system's memory: on the IoT board the SEAL library achieves a maximum time-series length of 850 values compared to HElib with a size of 2k, and on the NUC only HElib reaches a data limit of 20k values. We analyze the library performance per test platform over all operations $\in$ {initialization, euclidean, cosine} and time-series values. The HElib library works fastest at the server, the NUC is about 33 % slower, and IoT board is 11.7 times slower compared to the NUC. In comparison, the SEAL library performs best at the server, the NUC is 19 % slower, and the IoT board takes 51 times longer.

We compute the runtime per operation over all test platforms and time-series values. The initialization of time-series vectors takes on average 0.43 s using SEAL and 9 s with HElib, hence HElib is around 11.67 times slower per vector element. For the euclidean distance, the HElib library requires between [36, 569] s and the SEAL library achieves [56, 397] s, the relative runtime per vector element of HElib with 0.11 s is 2.16 times faster compared to SEAL (0.24 s). For the cosine distance, the HElib library takes on average between [135, 2137] s and using the SEAL library lasts [166, 1187] s, the relative runtime per vector element of HElib with 0.42 s is 1.72 times faster compared to SEAL with 0.72 s.

We compare the runtime performance of the baseline using two-dimensional positions like for LBS with our generated light patterns ranging between 1762 to 3717 raw voltage signals. We compute the average runtime for each homomorphic encryption library over all test platforms and operations at which the light pattern contains on average 1417 voltage values. The HELib library is slower with a runtime of 0.72 s for the baseline and the light pattern takes around 63.39 s, with a normalized delta of 90.99 % over the time-series length. The faster SEAL library achieves a runtime of 0.09 s for the baseline and 31.8 s for the larger light signal which results in a normalized delta of 63.34 %.



(a) Initialization (b) Euclidean distance (c) Cosine distance

Fig. 13: Performance of fully homomorphic encryption over three different testbeds: server, NUC, and IoT board using two libraries: HElib and SEAL. The baseline refers to the distance calculation using only two-dimensional positions compared to time-series with hundreds of values.

In a nutshell, our aim is to enhance user's privacy by applying homomorphic encryption to secure the time-series data processing of our device grouping. The runtime analysis motivates the need of new cryptographic primitives for efficient time-series data analysis. The up-to-date homomorphic encryption is too slow to be usable in practice with a runtime of about 30 s per distance computation whereas we require a maximum runtime of 0.5 s per calculation.

Publication 9

M. Haus, A. Y. Ding, C. Xu, and J. Ott. Demo: Touchless Wireless Authentication via LocalVLC. In *Proceedings of the 16th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, page 531, 2018. doi:10.1145/3210240.3211119

Publication Summary

We aim to automate indoor wireless (Wi-Fi) authentication based on LocalVLC. The idea is to use distance-bounding VLC for machine-to-machine communication to ease the setup of Wi-Fi networks. Thereby we improve the usability by avoiding manual distribution and tedious input of passwords for login. The workflow: LocalVLC continuously generates time-based one-time passwords (TOTPs) and setups an access point with SSID and generated password and broadcasts this security credential data. Via an add-on device, the user's smartphone is able to receive the VLC transmitted login data. The smartphone continually scans for nearby wireless networks and in case of a matching SSID, it can perform automated wireless authentication without any manual interaction. In comparison, other mechanisms to exchange credentials such as WPS or QR codes still require human interaction.

Demo: Touchless Wireless Authentication via LocaLVLC

Michael Haus
Technical University of Munich
haus@in.tum.de

Aaron Yi Ding
Technical University of Munich
ding@in.tum.de

Chenren Xu
Peking University
chenren@pku.edu.cn

Jörg Ott
Technical University of Munich
ott@in.tum.de

CCS CONCEPTS

• **Security and privacy** → *Multi-factor authentication*; • **Hardware** → *Emerging optical and photonic technologies*;

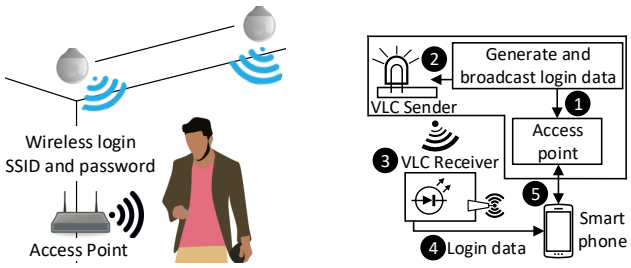
KEYWORDS

Visible light communication; Distance-bounding services

Visible light communication (VLC) has been enabling many applications related to Internet of Things (IoT) such as activity detection, occupation detection, and human sensing. Due to the emerging concern on IoT security and privacy, enforcing distance boundary is becoming a highly desired attribute for various IoT services. In theory, such distance boundary could be achieved by combining dedicated system design and communication technologies. In practice, we rarely find IoT services that can fully benefit from this safeguard feature. This is mainly due to the complexity of exploiting various radio-based communication technologies to attain the distance limit in different IoT environments. Since visible light does not pass through opaque objects, it is a good candidate to reinforce the spatial barrier control over different IoT services that demand for distance-bounding wireless communication.

We present LocaLVLC , a ready-to-deploy platform that takes advantage of VLC to realize distance-bounding services. The goal is to build a user friendly VLC-based system that can harness the unique property of visible light to respect spatial barriers like doors and walls. Our design suits for dense wireless IoT deployment as visible light does not interfere with other existing wireless infrastructure (e.g., Wi-Fi, Bluetooth Low Energy). LocaLVLC introduces a lightweight Morse data encoding adjusted for VLC to deliver fine-grained and low-cost distance boundary control. We have implemented a full-fledged platform prototype to demonstrate the practicality of LocaLVLC .

Our use case (Fig. 1(a)) aims to automate indoor wireless (Wi-Fi) authentication. We use VLC for machine-to-machine communication to ease the setup of Wi-Fi networks. LocaLVLC streamlines the credential management and achieves a "touchless" authentication experience in a distance-bounding manner, avoiding manual distribution and tedious input of passwords for login. In comparison, other mechanisms to exchange credentials such as WPS or QR codes still require human interaction. LocaLVLC covers many target devices including common Wi-Fi equipped devices (e.g., smartphone, tablet, laptop) as well as IoT devices like sensor boards. The workflow of our demo is shown in Fig. 1(b) where LocaLVLC continuously generates time-based one-time passwords (TOTPs) and setups an access point with SSID and generated password ❶. For the wireless network, LocaLVLC broadcasts the security credential data



(a) Transmit credentials via LocaLVLC for touchless wireless authentication

(b) Workflow of LocaLVLC

Figure 1: Use case and overview of LocaLVLC

including SSID and password ❷. Via an add-on device equipped with a photodiode ❸, the user's smartphone is able to retrieve the VLC transmitted login data ❹. The smartphone continually scans for nearby wireless networks and in case of spotting a matching SSID ❺, it can perform automated wireless authentication without any manual interaction.

We note that potential data leakage of VLC signals through windows may limit the practicality of LocaLVLC for credential management. Our evaluation reveals that such data leakage is possible but only with a directional, shining LED used as transmitter. To address interference among light bulbs, e.g., user's device receives VLC signals simultaneously from multiple sources, we can use a spatial distribution of light bulbs to avoid interference, enhance LocaLVLC with time synchronization for VLC signals, or integrate a dedicated algorithm for spatial interference detection.

Our demo includes two IoT boards to setup the LocaLVLC platform. One board serves as an AP and broadcasts login data via VLC for the managed network. The second board acts as a light receiver, which connects to a smartphone via MQTT. As the light receiver constantly retrieves the login data, we plot both the raw and processed light signal in real-time to demonstrate the encoding scheme. To assess the quality of service, we show a live authentication monitor on the AP board including success rate of authentication attempts and latency.

Publication 10

M. Haus, A. Y. Ding, P. Hui, and J. Ott. Demo: iConfig - What I See is What I Configure. In *Proceedings of the 12th ACM MobiCom Workshop on Challenged Networks (CHANTS)*, pages 29–31, 2017. doi:10.1145/3124087.3124103

Publication Summary

With respect to human-computer interaction, a key observation from our user study is that the conventional screen-keyboard setup is far from optimal for the interaction among users, their smart gadgets, and surrounding IoT devices. The system interaction should be more natural and fluent. This is the reason for our second prototype dedicated for wearables. As most natural way, the included speech recognition runs completely offline on the smartglass for hands-free device configuration during user movement limiting the distraction of user attention.

Demo: iConfig – What I See is What I Configure

Michael Haus
Technical University of Munich
haus@in.tum.de

Aaron Yi Ding
Technical University of Munich
ding@in.tum.de

Pan Hui
Hong Kong University of Science and Technology
panhui@cse.ust.hk

Jörg Ott
Technical University of Munich
ott@in.tum.de

ABSTRACT

Managing IoT devices in urban areas is becoming crucial because the majority of people living in cities and the number of deployed IoT devices are steadily increasing. In this demo, we present iConfig, an edge-driven platform dedicated to manage densely deployed IoT devices in smart cities. Our goal is to address three challenging issues in current IoT management: registration, configuration, and maintenance. The core of iConfig is its programmable edge module, which can be deployed across smartphones, wearables, and smart boards to configure and interact with physically proximate IoT devices. Our system evaluation shows that iConfig can effectively address the aforementioned IoT management challenges by harnessing mobile and edge cooperation.

KEYWORDS

IoT Configuration; Management of Edge Devices

1 INTRODUCTION

Managing IoT devices in urban areas is becoming important because the majority of people living in cities and the number of deployed IoT devices are steadily increasing. Urban IoTs support the smart city concept which integrates traditional and modern information and communications technology (ICT) for a unified and simple access to services for the city administration and the residents [3]. The aim is an enhanced use of public resources, improving quality of services for citizens while reducing operational costs of public administration [3]. Currently, there is a lack of instruments to seamlessly manage large IoT deployments in which ad hoc management is becoming untenable, especially for IoT devices without Internet connectivity. Therefore, we propose iConfig, an edge-driven platform taking advantage of wearable and edge computing to run edge modules on smart glasses or IoT boards. This allows iConfig to consolidate various connected devices to its management backend, which enforces a unified configuration procedure. We successfully tested iConfig on Android smartphone and smart glass (MAD Gaze X5). The design principles of iConfig include: 1) automatic configuration of IoT devices to avoid misconfigurations which become one of the dominant causes of system failures [2], 2) easy to use frontend, 3) orchestrate devices via global view, and 4) serve as platform for developer to enable add on services.

In this demo, we use iConfig to configure Bluetooth Low Energy (BLE) beacons, which represent one of the most challenging classes of IoT devices due to missing Internet connectivity. These battery powered BLE beacons are small-size wireless devices that transmit a short-range BLE signal to mobile computing devices (e.g., smartphones) [1]. Via BLE, users' devices are notified of the beacon proximity by receiving signals which contain contextual information, typically about indoor surroundings and its contents. Thus, the receiving end is able to perform location aware actions, such as accessing specific URLs for marketing and assistance purposes. The ability of iConfig to programmatically adjust device parameters facilitates different use cases: 1) set up an automated testbed for research projects, 2) debug and monitor IoT devices, and 3) energy aware management of IoT devices which are deployed in smart buildings.

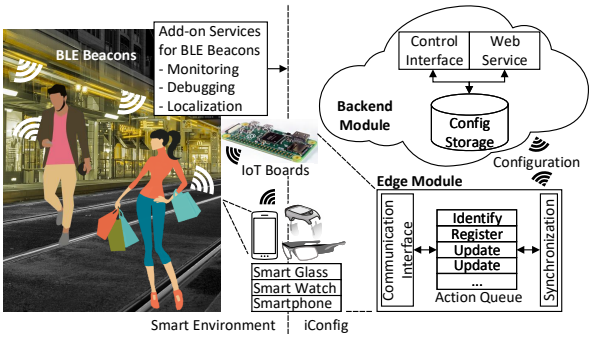


Figure 1: iConfig in the context of smart environments

2 ICONFIG FOR IOT MANAGEMENT

The iConfig system architecture, as illustrated in Fig. 1, consists of two major modules: mobile edge module and backend module. The framework is able to identify, register, and update IoT devices (as in our case BLE beacons). Supported by our speech recognition, a user wearing an iConfig-enabled smart glass is able to discover, register, and configure BLE beacons while walking around. Fig. 2 shows the demo hardware including smart glass and BLE beacons. In the discovery phase, the edge module shows only unregistered beacons. After discovering unregistered beacons, the user is able to register one beacon at a time. The beacon shows a red light as feedback for device identification. When the beacon identification was successful, the user can register the beacon to the iConfig backend with additional information for device localization, such as nearest room number, picture of device place. Afterwards, the edge module automatically configures the BLE beacon with a default configuration, including password, URL, transmission power, and packet advertisement rate. Finally, the edge module synchronizes all configuration data to the backend. The registration has to be done only once per beacon. For advanced management, iConfig backend

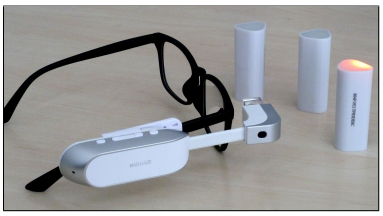


Figure 2: Smart glass configures blinking beacon

Table 1: Configuration time (s) for one to ten beacons

1	2	3	4	5	6	7	8	9	10
2.7	5.3	7.5	10	12.4	15.1	16.5	16.4	17.8	22

provides a control interface for the administrator, including a global view of installed BLE beacons. Device updates are independent of user actions and automatically triggered when two conditions are satisfied: 1) adapted BLE configuration from the administrator via iConfig backend available, and 2) the BLE beacon is currently discovered by the user via iConfig edge module.

3 EVALUATION

We analyzed the system performance of iConfig regarding memory usage and configuration scalability over multiple beacons. Additionally, we conducted a user study to highlight the drawback of manual configuration, which is untenable in dense device deployments.

We measured the memory usage of the iConfig edge module on Android smartphone (OS 7.1.1) in offline and online mode during configuration of ten BLE beacons. In online mode, the edge module used 6.50 ± 0.98 MB similar to offline mode with a memory usage of 6.47 ± 0.96 MB. These results show that the memory footprint of the iConfig edge module is small enough to run on programmable IoT devices, e.g., smart glass, IoT boards.

For configuration scalability, we evaluated ten cases (from one to ten beacons) each over 20 rounds in a dense testbed deployment. Our evaluation yielded 18 unauthenticated connect errors (meaning BLE configuration is not possible) out of 1100 connect attempts, i.e., rate of 1.64 %. Table 1 shows a linear increase of configuration time over all beacons, on average an increase of 2.2 s per beacon. In addition, we evaluated the success rate which describes whether configuration parameters were correctly set. The lowest rate was 75 % during configuration of four beacons. In most cases, iConfig achieved 100 % success rate.

Our user study revealed the gap between a manual and an automatic device configuration system. Ten participants manually configured three beacons with a predefined configuration using the vendor application. In the next step, the same beacons were configured via iConfig, in which the default configuration was automatically written to each BLE beacon. The manual configuration took in average six times longer than iConfig automatic configuration, which reflects a time saving of 83 %. Moreover, only 1/3 of all manual configurations were entirely correct.

4 HANDS-FREE DEVICE CONFIGURATION

We take advantage of speech recognition to enable hands free device configuration and implemented two prototypes for different device types: smartphone and smart glass. On the smartphone the speech recognition can be optionally activated, on the smart glass it is automatically activated to allow a convenient usage of the iConfig edge module. The speech recognition works completely offline on the smart glass and can be activated via the key phrase *geronimo* for device discovery and registration. Afterwards, the user is able to control iConfig via the following speech commands: *select* nearest beacon automatically detected via strongest signal strength, *target* specific beacon via beacon id, *identify* beacon via visual feedback, and *register* beacon at iConfig backend. Furthermore, we implemented a custom camera control via speech recognition to capture an image of beacon placement for easier localization.

To enhance the robustness of our speech recognition (e.g., number three sometimes recognized as “free”), we used several metrics for string similarity of the speech input: Euclidean, Levenshtein, and Jaro-Winkler distance. In our setting, the Jaro-Winkler distance which is designed for short strings provided the best results to calculate the string similarity of iConfig speech input.

5 DISCUSSION AND OUTLOOK

A key observation from our user study and experiments is that the interaction among users, their smart gadgets and surrounding IoT devices via the conventional screen-keyboard is far from optimal. Especially for smart cities with a multitude of services empowered by IoT devices, the system interaction should be more natural and fluent. This is the main reason for our prototype dedicated for wearables (e.g., smart glass). The combination of speech recognition and hands-free devices can enable a more integrated interaction during user movement and limit the distraction of user attention. Hence, iConfig is our endeavor to enhance user experience and to streamline the management of large scale IoT deployments, especially for low-budget devices such as BLE beacons without backend connectivity.

We plan to use iConfig-enabled drones for achieving autonomous configuration of IoT devices which are deployed in more challenging environments, such as high ceilings, steel-making factories.

REFERENCES

[1] Michael Wang and Jack Brassil. 2015. Managing Large Scale, Ultra-Dense Beacon Deployments in Smart Campuses. In *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 606–611.

[2] Tianyin Xu and Yuanyuan Zhou. 2015. Systems Approaches to Tackling Configuration Errors. *ACM Computing Surveys* 47, 4 (2015), 1–41.

[3] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. 2014. Internet of Things for Smart Cities. *IEEE Internet of Things Journal* 1, 1 (2014), 22–32.

6 DEMONSTRATION PROCEDURE

In this demonstration, we present a typical use case of our platform to configure BLE beacons via smart glass over speech control. For comparison, we provide a smartphone to configure the beacons. The testbed layout is depicted in Fig. 3. The laptop runs the iConfig backend server together with a MongoDB database which is accessible via a local Wi-Fi hotspot. The iConfig backend server receives the beacon configurations and enables add-on services. Besides

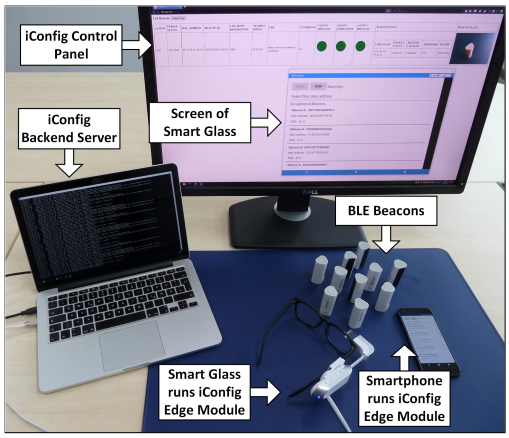


Figure 3: Demo setting

that, the laptop projects the screen content of the smart glass to a larger monitor. This monitor shows the GUI of the iConfig edge module running on the smart glass and the control panel of the iConfig backend. For device configuration, our demo includes a smart glass and a smartphone running the iConfig edge module to identify, configure, and update BLE beacons. We provide ten BLE beacons for configuration.

Our demo includes the following devices: 1) smart glass running iConfig edge module with speech control for hands-free device configuration, 2) smartphone running iConfig edge module for conventional manual input or optional speech control, 3) ten BLE beacons for device configuration, 4) a laptop to run the iConfig backend server and to project the screen content of the smart glass to an external monitor, and 5) a headphone for improved speech control in noisy environments.

We would need space for a poster board and a table, big enough for one laptop and a 24-inch monitor. Our prototype system requires approximately 30 minutes to setup.

In addition, we need a table and power for the demo. We also prefer to have a large monitor with HDMI or VGA cable to show the screen content of the smart glass running the iConfig edge module. We will cast the control panel of the iConfig backend module on the monitor as well.