TECHNISCHE
UNIVERSITÄT
DARMSTADT

MAKING ZERO-INTERACTION PAIRING AND AUTHENTICATION PRACTICAL IN THE INTERNET OF THINGS

Am Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

DISSERTATION

zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
von

MIKHAIL FOMICHEV

Erstreferent: Prof. Dr.-Ing. Matthias Hollick
Korreferent: Prof. Jun Han, PhD

Darmstadt 2021
Hochschulkennziffer D17

Mikhail Fomichev, *Making Zero-interaction Pairing and Authentication Practical in the Internet of Things*, Dissertation, Technische Universität Darmstadt, 2021.

Fachgebiet Sichere Mobile Netze
Fachbereich Informatik
Technische Universität Darmstadt
Jahr der Veröffentlichung: 2021
Tag der mündlichen Prüfung: 30. August 2021
URN: urn:nbn:de:tuda-tuprints-197688

In memory of my beloved Alice, who will always have a special place in my heart.

ABSTRACT

The proliferation of the Internet of Things (IoT) requires establishing and maintaining secure communication between smart devices to ensure user privacy and trustworthiness of IoT systems. Zero-interaction pairing (ZIP) and zero-interaction authentication (ZIA) are recent techniques that allow pairing or authenticating devices without user involvement utilizing devices' physical context (e.g., ambient audio). Compared to centralized security solutions for the IoT such as public-key infrastructure (PKI) and conventional user-assisted pairing and authentication methods (e.g., entering a password), ZIP and ZIA schemes promise improved user experience, as they do not require users to participate in pairing or authentication procedures, and easy deployment, as they rely on on-board sensors of smart devices. However, we find that proposed ZIP and ZIA schemes are still immature, requiring improvements in three areas: security, usability, and deployability. In this thesis, we advance the domain of ZIP and ZIA in these three areas as follows. First, we analyze state-of-the-art ZIP and ZIA schemes both theoretically and empirically using real-world data that we collect. Our findings reveal that these schemes show reduced security and usability under realistic conditions, and we identify reasons why this reduction occurs. Second, we improve on ZIP, proposing a novel ZIP architecture called *FastZIP* combining a recently introduced Fuzzy Password-Authenticated Key Exchange (fPAKE) protocol, which has stronger security properties than the cryptographic primitives used by the state-of-the-art ZIP schemes, and sensor fusion, which allows building robust context from multiple sensor modalities, each capturing a distinct physical phenomenon. We demonstrate, collecting real-world data using off-the-shelf devices, that *FastZIP* has higher security guarantees than state-of-the-art ZIP schemes against brute-force offline and predictable context attacks (e.g., context replay) and significantly shorter pairing time, improving the usability of our scheme. Third, we develop a new copresence detection method named *Next2You*; copresence detection is a core part of any ZIA scheme. *Next2You* utilizes channel state information (CSI), which captures a unique wireless context of an environment (e.g., a room), and neural networks. Through our real-world experiments using off-the-shelf smartphones, we demonstrate that *Next2You* outperforms state-of-the-art copresence detection methods in two ways: (1) it achieves accurate copresence detection in challenging cases of low-entropy context (e.g., empty room with few events occurring) and insufficiently separated environments (e.g., adjacent rooms), thus is more secure and (2) *Next2You* requires devices to only have ubiquitous Wi-Fi chipsets, without a need for extra sensors (e.g., microphones), improving the deployability of our method. Fourth, we publicly release the collected context data and codebase of the above contributions, enhancing the reproducibility in the domain of ZIP and ZIA.

ZUSAMMENFASSUNG

Die Verbreitung des Internets der Dinge (IoT) erfordert den Aufbau und die Aufrechterhaltung einer sicheren Kommunikation zwischen vernetzten Geräten, um die Privatsphäre der Benutzer und die Vertrauenswürdigkeit von IoT-Systemen zu gewährleisten. Zero-Interaction-Pairing (ZIP) und Zero-Interaction-Authentication (ZIA) sind neuere Techniken, die das Pairing oder die Authentifizierung von Geräten ohne Benutzereingriff unter Verwendung des physischen Kontexts der Geräte (z. B. Umgebungsgeräusche) ermöglichen. Im Vergleich zu zentralisierten Sicherheitslösungen für das IoT wie Public-Key-Infrastrukturen (PKI) und herkömmlichen benutzergestützten Pairing- und Authentifizierungsmethoden (z. B. Eingabe eines Passworts) versprechen ZIP- und ZIA-Verfahren eine verbesserte Benutzererfahrung, da sie keine Mitwirkung durch die Nutzer erfordern, und eine einfachere Umsetzung, da sie auf eingebauten Sensoren von vernetzten Geräten basieren. Wir stellen jedoch fest, dass die vorgeschlagenen ZIP- und ZIA-Verfahren noch unausgereift sind und in drei Bereichen verbessert werden müssen: Sicherheit, Benutzbarkeit und praktische Umsetzbarkeit. In dieser Doktorarbeit erweitern wir den Stand der Forschung in diesen drei Bereichen wie folgt: Zunächst analysieren wir aktuelle ZIP- und ZIA-Verfahren sowohl theoretisch als auch empirisch unter Verwendung von uns gesammelter realistischer Daten. Unsere Ergebnisse zeigen, dass diese Verfahren unter realistischen Bedingungen eine reduzierte Sicherheit und Benutzbarkeit aufweisen, und wir identifizieren Gründe für diese Reduzierung. Zweitens verbessern wir ZIP, indem wir eine neuartige ZIP-Architektur namens *FastZIP* vorschlagen, die ein kürzlich eingeführtes Fuzzy Password Authenticated Key Exchange (fPAKE)-Protokoll verwendet, das über stärkere Sicherheitseigenschaften verfügt als die kryptographischen Primitive aktueller ZIP-Verfahren. Es nutzt außerdem eine Kombination verschiedener Sensoren um einen robusten Kontext aus mehreren Sensormodalitäten abzuleiten, die jeweils ein bestimmtes physikalisches Phänomen erfassen. Anhand von realen Daten, die mit handelsüblichen Geräten gesammelt wurden, zeigen wir, dass *FastZIP* höhere Sicherheitsgarantien gegen Brute-Force-Offline- und vorhersehbarer-Kontext-Angriffe (z. B. durch Context Replay) bietet als moderne ZIP-Verfahren. Es benötigt außerdem deutlich weniger Zeit um ein Pairing zu etablieren, was die praktische Benutzbarkeit unseres Verfahrens verbessert. Drittens entwickeln wir eine neue Copräsenz-Erkennungsmethode namens *Next2You*. Copräsenz-Erkennung ist ein Kernbestandteil jedes ZIA-Verfahrens. *Next2You* nutzt Channel-State-Informationen (CSI), die den einzigartigen drahtlosen Kontext einer Umgebung (z.B. eines Raumes) erfassen, und neuronale Netzwerke. Durch unsere Experimente mit handelsüblichen Smartphones zeigen wir, dass *Next2You* die modernen Copräsenz-Erkennungsmethoden in zweierlei Hinsicht übertrifft: (1) es erreicht eine

genaue Copräsenz-Erkennung in schwierigen Fällen mit geringer Variation im Kontext (z. B. einem leeren Raum mit wenigen auftretenden Ereignissen) oder unzureichend getrennten Umgebungen (z. B. benachbarten Räumen) und ist daher sicherer und (2) *Next2You* benötigt lediglich einen handelsüblichen Wi-Fi-Chipsatz und erfordert anders als bisherige Verfahren keine zusätzliche Sensoren (z. B. Mikrofone), was die praktische Umsetzbarkeit unserer Methode verbessert. Viertens veröffentlichen wir die gesammelten Forschungsdaten und die Codebasis der oben genannten Beiträge, um die Reproduzierbarkeit im Bereich von ZIP und ZIA zu verbessern.

ACKNOWLEDGMENTS

I would like to thank several people without whom this thesis would not have been possible. First of all, I express my gratitude to my advisor Prof. Matthias Hollick for his trust, encouragement, and support throughout these years. Second, I thank Prof. Jun Han for agreeing to become the second advisor of this thesis and for his valuable advice.

I would like to thank my colleagues for a pleasant and fruitful working environment, especially Max Maass, Lars Almon, Flor Álvarez, Alejandro Molina, Luis F. Abanto-Leon, and Arash Asadi. I am also grateful to Doris Müller and Ursula Paeckel for their administrative support and help with the university paperwork. Furthermore, I thank my student assistant Timm Lippert for his dedication and hard work.

I thank my friends for their support and a nice time that we have spent together, taking my mind off the PhD hurdles. I am grateful to my parents, who have always supported and believed in me. I thank my son Roman for making me smile and inspiring me to carry on. Finally, I thank my wife Anastasia for her unconditional love and support throughout all these years.

My research has been partially funded by the German Federal Ministry of Education and Research (BMBF) within the Smarter[1] project and the German Research Foundation (DFG) within the SFB1119 CROSSING[2] research center and the Loewe initiative together with the State of Hesse within the NICER[3] project and the BMBF together with the State of Hesse within the ATHENE[4] research center and the University of Oslo within the Parrot[5] project. Calculations for this research were conducted on the Lichtenberg high performance computer of the TU Darmstadt. Also, I thank the Flaticon[6] team for the vector icons that I used in Figure 1, Figure 3, and Figure 4.

1 https://smarter-projekt.de/
2 https://www.crossing.tu-darmstadt.de/
3 https://www.nicer.tu-darmstadt.de/
4 https://www.athene-center.de/
5 https://www.mn.uio.no/ifi/english/research/projects/parrot/
6 https://www.flaticon.com/

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AGC	automatic gain control
AI	artificial intelligence
AP	access point
AUC	Area Under the Curve
BLE	Bluetooth Low Energy
CIR	channel impulse response
CSI	channel state information
CV	cross-validation
DH	Diffie-Hellman
DoS	denial-of-service
ECC	error correction code
ECU	electronic control unit
EER	Equal Error Rate
EKE	Encrypted Key Exchange
EWMA	exponentially weighted moving average

FAR	False Acceptance Rate
FFT	fast Fourier transform
FLOPS	floating point operations per second
fPAKE	Fuzzy Password-Authenticated Key Exchange
FRR	False Rejection Rate
HCI	human-computer interaction
IMD	implantable medical device
IMU	inertial measurement unit
IoT	Internet of Things
IPI	inter-pulse interval
IrDA	Infrared Data Association
LoS	line-of-sight
MAC	message authentication code
MANA	Manual Authentication
MITM	machine-in-the-middle
NFC	near-field communication
NTP	Network Time Protocol
OoB	out-of-band
PBC	Push Button Configuration
PDP	power delay profile
PFS	perfect forward secrecy
PHY	physical
PKI	public-key infrastructure
PoE	Power over Ethernet
QoS	quality of service

RFID	radio-frequency identification
RIR	room impulse response
RRR	Right for the Right Reasons
RSS	received signal strength
SAS	short authentication string
SDP	Secure Device Pairing
SG	Savitzky-Golay
SNR	signal-to-noise ratio
TAR	True Acceptance Rate
TEA	tamper-evident announcement
TEP	tamper-evident pairing
TVOC	total volatile organic compound
VLC	visible light communication
WPS	Wi-Fi Protected Setup
WSN	wireless sensor network
ZIA	zero-interaction authentication
ZIP	zero-interaction pairing
ZIS	zero-interaction security

LIST OF PUBLICATIONS

During the course of writing this thesis, I co-authored several papers and articles that I list below.

JOURNAL AND MAGAZINE ARTICLES

[1] Mikhail Fomichev, Luis F. Abanto-Leon, Max Stiegler, Alejandro Molina, Jakob Link, and Matthias Hollick. "Next2You: Robust Copresence Detection Based on Channel State Information." In: *ACM Transactions on Internet of Things* 1.1 (2021), pp. 1–30. **Part of this thesis.**

[2] Mikhail Fomichev, Flor Álvarez, Daniel Steinmetzer, Paul Gardner-Stephen, and Matthias Hollick. "Survey and Systematization of Secure Device Pairing." In: *IEEE Communications Surveys & Tutorials* 20.1 (2018), pp. 517–550. **Part of this thesis.**

[3] Mikhail Fomichev, Max Maass, Lars Almon, Alejandro Molina, and Matthias Hollick. "Perils of Zero-interaction Security in the Internet of Things." In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3.1 (2019), pp. 1–38. Distinguished Paper Award. **Part of this thesis.**

[4] Mikhail Fomichev, Max Maass, and Matthias Hollick. "Zero-interaction Security-Towards Sound Experimental Validation." In: *GetMobile: Mobile Computing and Communications* 23.2 (2019), pp. 16–21. **Part of this thesis.**

CONFERENCE AND WORKSHOP PAPERS

[5] Mikhail Fomichev, Julia Hesse, Lars Almon, Timm Lippert, Jun Han, and Matthias Hollick. "FastZIP: Faster and More Secure Zero-interaction Pairing." In: *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*. 2021, pp. 440–452. **Part of this thesis.**

COLLABORATIONS AND MY CONTRIBUTION

Research is a collaborative effort, advancing the current state of knowledge. Exchanging ideas and embarking upon interdisciplinary research projects bring together experts from different fields. Contributions presented in this thesis have benefited from fruitful collaborations, I thus use the pronoun "we" when describing the results of this thesis in subsequent chapters. I am grateful to my colleagues and students with whom I had a pleasure to work, namely Flor Álvarez, Daniel Steinmetzer, Paul Gardner-Stephen, Max Maass, Lars Almon, Alejandro Molina, Timm Lippert, Julia Hesse, Jun Han, Max Stiegler, Luis F. Abanto-Leon, and last but not least, to my advisor Matthias Hollick, who is a co-author of all my papers.

When working in teams, it becomes difficult to distill the contributions of each individual author precisely. For all papers that comprise this thesis, each co-author contributed, to a different extent, by exchanging ideas, discussing results, and writing them up into a paper. In the following, I detail the contributions of my co-authors and myself per chapter. In addition, I follow the regulations of the Department of Computer Science at Technical University of Darmstadt and give an account of the parts that include verbatim or revised fragments of previous publications that form this thesis as indicated in the preceding list of publications.[7]

Chapter 1 provides an overview of contributions of this thesis based on the core publications comprising it [1, 2, 5, 3], while Chapter 2 presents relevant background and reviews related work.

Chapter 3 is based on a joint work with Flor Álvarez, Daniel Steinmetzer, and Paul Gardner-Stephen. I came up with the idea for a new system model for device pairing based on physical channels, human-computer interaction (HCI) channels, and application classes, which lies at the core of our work. Together with Flor, we compiled the list of papers to be surveyed, and I conducted the analysis of the papers based on physical channels (Section 3.3) and application classes (Section 3.5). Flor and I analyzed papers based on HCI channels (Section 3.4), while Daniel provided input for discussion points on physical channels, and Paul helped to improve the presentation of our results.

Chapter 4 is based on a joint work with Max Maass, Lars Almon, and Alejandro Molina. I came up with the initial idea for this work, proposing the list of zero-interaction pairing (ZIP) and zero-interaction authentication (ZIA) schemes that we investigated. Together with Max, we reproduced the five ZIP and ZIA schemes from the ground up, designed experiments, and collected context data with the help of Lars. I evaluated three schemes while Max—the remaining two schemes. Alejandro helped

7 References in this chapter refer to my list of publications given on Pages xxi to xxi.

us to evaluate two ZIA schemes based on machine learning. With the feedback from Max, I distilled our practical lessons learned from this work (Appendix B).

Chapter 5 results from a collaboration with Lars Almon, Julia Hesse, Jun Han, and Timm Lippert. I proposed the idea for a faster and more secure ZIP scheme—called *FastZIP*—based on the Fuzzy Password-Authenticated Key Exchange (fPAKE) protocol and sensor fusion. Together with Lars and Timm, we collected context data from moving cars on which *FastZIP* was evaluated. I am the sole author of the *FastZIP* implementation except for the fPAKE part, which was implemented by Timm under the guidance of Julia. Timm also conducted benchmarking of the implemented fPAKE protocol under my supervision (Section 5.5.6), while Lars ran entropy evolution of fingerprints generated by *FastZIP* (Section 5.5.5), all other parts of evaluation were carried out by me. Julia analyzed how to set fPAKE parameters for *FastZIP* (Key Exchange in Section 5.3) and provided necessary background on the fPAKE protocol (Section 5.1), while Jun contributed by constantly providing feedback and improving the presentation of our results.

Chapter 6 is based on a joint work with my master's student Max Stiegler as well as Luis F. Abanto-Leon, Alejandro Molina, and Jakob Link. I suggested the design of our novel copresence detection scheme—named *Next2You*—based on channel state information (CSI) and neural networks, while Max implemented it, and together we conducted CSI data collection. The results obtained by Max (Section 6.5.2), I refactored, extended, and prepared for publication. Luis helped by providing theoretical justification for the use of CSI for copresence detection (Section 6.3.2), while Alejandro guided the machine learning parts of our work (Section 6.3.5, Section 6.5.4), and Jakob assisted our experiments by porting the *Nexmon* CSI extractor onto new devices.

Part I

PRELUDE

1 INTRODUCTION

Pairing and authentication are primary techniques for bootstrapping and maintaining secure communications between devices. The advance of the Internet of Things (IoT) results in a rapid increase in the number of smart devices, which become prevalent in everyday life, ranging from professional activities to leisure and healthcare [128]. Recent reports estimate that the number of connected IoT devices will reach over 75 billion by 2025 [224, 339]. The shift to the IoT paradigm brings many benefits such as the higher level of automation and fine-grained control over processes, saving resources, increasing productivity, and enabling a sustainable environment [118]. However, this shift also has a downside, introducing new security and privacy issues. Specifically, the wide adoption of IoT devices increases the amount of sensitive data that they collect such as user activities, health records, and consumption levels [55, 144, 195]. Protecting these sensitive data is imperative to ensure user privacy and trustworthiness of IoT systems. We see that the current situation is alarming: for example, the ubiquity of sensing aided by machine learning makes it feasible to infer various sensitive data of users via indirect measurements [31]. Furthermore, IoT devices often have weak security, hence are prone to attacks, due to their diversity (e.g., hardware, used protocols) and limited resources (e.g., processing power). As a result, no consistent security mechanisms exists among devices of different vendors, and traditional security techniques such as public-key infrastructure (PKI) cannot be directly applied [153], as they do not scale to billions of devices, cannot accommodate various use cases in which IoT devices may be used, and require significant computational resources. A large number of diverse devices with limited resources requires a new approach to pairing and authentication in the IoT to address the above security and privacy issues.

"As 5G networks roll out, the use of connected IoT devices will accelerate dramatically."

1.1 MOTIVATION

The growing reliance of society on the IoT urges the need to protect not only the data transmitted by IoT devices but also themselves, as they are becoming increasingly targeted by attackers [215, 62]. Traditionally, pairing and authentication have been used to provide confidentiality, integrity, and authenticity of the data exchanged between two devices. Pairing allows two devices without any prior association or any jointly trusted third party to establish a shared secret key, thus their communication can be encrypted and integrity-protected. While authentication allows one device to assure the identity of another device or validate that the received data was indeed sent by the expected device. In the IoT, pairing enables the protection of sensitive data transmitted by devices (e.g.,

The survey by Gartner ranks security risks of IoT systems as a top concern for the next three to five years.

user location) from eavesdropping and tampering. While strong authentication (e.g., two-factor) prevents IoT devices from being controlled by malware [189] and protects against relay attacks on wireless channels [58]. These threats pose a high risk to user privacy and trustworthiness of IoT systems, resulting in substantial financial losses, reputational damage, and even danger to human life [61, 214]. For example, the lack of encryption on transmitted user location makes it possible to track users, violating their privacy or even physical security [2, 400]. Similarly, weak authentication in IoT devices (e.g., default usernames and passwords) allows taking them over and using them in massive denial-of-service (DoS) attacks against critical infrastructure such as road safety systems [345].

Traditionally, pairing and authentication schemes rely on user assistance (e.g., entering a password) to achieve their purpose [33, 50]. However, the skyrocketing number of IoT devices requires a prohibitive amount of user effort to perform pairing or authentication, which does not scale. In addition, many IoT devices are not equipped with user interfaces (e.g., display), making user-assisted pairing and authentication infeasible. These issues in addition to diversity and resource constraints of IoT devices, preventing the use of centralized security mechanisms such as PKI, resulted in the recent advent of zero-interaction pairing (ZIP) [243] and zero-interaction authentication (ZIA) [60]. They enable autonomous pairing and authentication without user interaction by relying on context data (e.g., audio) sensed by devices from their ambient environment.

Zero-interaction pairing and authentication enable more secure and usable IoT, which is currently plagued with privacy and security issues.

The proposed ZIP and ZIA schemes claim high usability, as they exclude a user from pairing and authentication procedures. Also, these schemes rely exclusively on on-board sensors of smart devices, promising to be readily deployable. By eliminating a user effort and requiring no extra hardware, ZIP and ZIA schemes can be executed on many smart devices simultaneously, featuring improved scalability compared to user-assisted methods.

Despite the abovementioned advantages of ZIP and ZIA schemes, a few studies questioned their *security* [324, 325]. However, these findings were obtained in limited settings (e.g., lab environment) considering a single ZIP/ZIA scheme, making it difficult to generalize them. In addition, the *usability* metric encompasses more aspects than only a minimum user effort such as the completion time of ZIP and ZIA schemes or the rate of failed pairing/authentication attempts between legitimate devices in dense real-world deployments (e.g., adjacent rooms with multiple IoT devices). These aspects are vital for a ZIP or ZIA scheme to be usable, but they have not been thoroughly studied by prior research. Since billions of smart devices already exist, the role of *deployability* has become crucial for putting security mechanisms into practice [192, 244, 253]. We are aware of a single previous ZIA scheme that investigates deployability [181], indicating that more research is required in this direction.

To unlock the full potential of ZIP and ZIA schemes, we are motivated to systematically explore them in this thesis with respect to security, usability, and deployability, paving the way for their practical deployment.

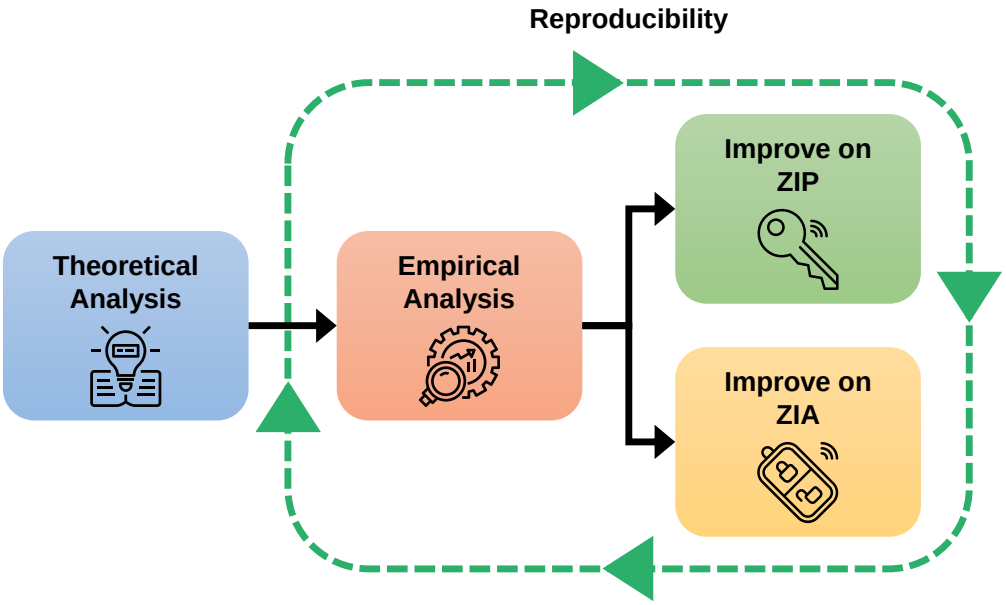


Figure 1: Overview of our approach to improving security, usability, and deployability of ZIP and ZIA schemes.

1.2 CHALLENGES AND GOALS

The ultimate goal of this thesis is to improve *security*, *usability*, and *deployability* of ZIP and ZIA schemes. To achieve this goal, we use the following approach (cf. Figure 1). First, we survey and conduct a theoretical analysis of existing pairing schemes, including ZIP, to identify their systemic weaknesses, guiding our further research. Second, utilizing this input, we reproduce and evaluate state-of-the-art ZIP and ZIA schemes using real-world data, demonstrating their realistic security and usability as well as revealing deployability pitfalls. Third, based on these findings, we improve on both ZIP and ZIA by developing two novel schemes that achieve better security, usability, and deployability compared to state of the art. These three metrics facilitate practical ZIP and ZIA solutions that can be used in real life. However, security, usability, and deployability might be at odds (cf. Figure 2), thus we discuss their trade-offs in ZIP and ZIA schemes that we develop. Finally, we bootstrap the reproducibility in the domain of ZIP and ZIA by providing the first open-source datasets of various context data as well as implementations of existing and our schemes. Each of these steps poses unique challenges that we discuss in the following.

To enable practical ZIP and ZIA, we research their security, usability, and deployability.

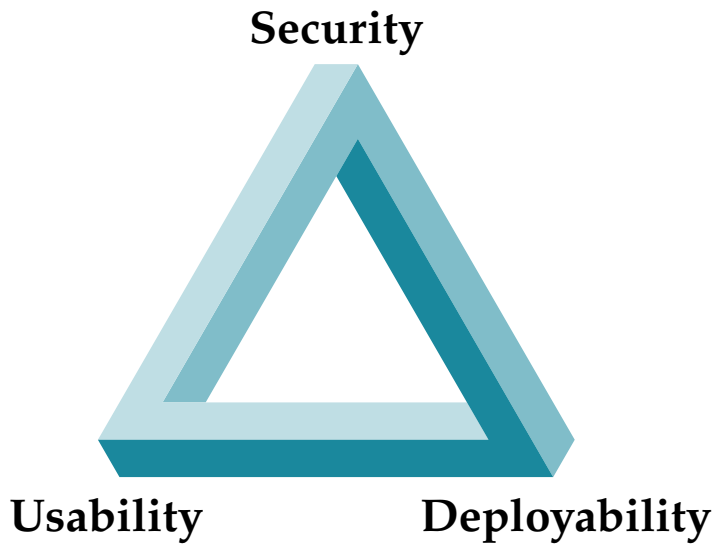


Figure 2: Design approach that we follow when developing new ZIP and ZIA schemes. We aim to improve on security, usability, and deployability simultaneously, however these metrics might be in conflict, thus it is important to identify their trade-offs.

1.2.1 *Theoretical Analysis of Pairing Schemes*

As mentioned above, ZIP is a recent technique, thus we need to identify the shortcomings of existing pairing schemes, including a number of pioneering ZIP works, to be able to improve on them. Such analysis needs to be performed in a systematic way, covering a wide range of pairing schemes, considering their security, usability, and use cases. We find that previous surveys on device pairing do not yield a coherent comparison of different schemes with respect to security, usability, and use cases, thus we conduct our own analysis to fill in this gap.

Incomparability and recurring security incidents of pairing schemes point to deficiencies in their design.

Incomparability and Limited Security of Pairing Schemes. We discover that prior surveys on device pairing cannot justifiably provide a fair comparison between different schemes even against a single metric (e.g., security). Moreover, we find that pairing schemes have been repeatedly compromised via the same attack vector, namely by exploiting the out-of-band (OoB) channel, which is used to ensure that intended pairing devices have established the shared secret key. Such issues suggest fundamental design flaws of existing pairing schemes, which we need to identify and address.

Enhanced Design Approach for Pairing Schemes. To enhance the design of pairing schemes, we need to (1) devise a generic pairing model, encompassing interactions between devices and users as well as application contexts of pairing, (2) characterize properties of these interactions and application contexts that affect security and usability of the schemes, and (3) list which information should be provided by a pairing scheme to make it comparable with others.

1.2.2 *Empirical Analysis of Zero-interaction Pairing and Authentication Schemes*

To compare the security and usability of state-of-the-art ZIP and ZIA schemes, they need to be evaluated on the same context data. Also, to obtain their realistic security and usability performance, the schemes should be investigated in challenging real-world scenarios. The main problems that hinder such analysis are the unavailability of schemes' implementations and the lack of common real-world context datasets.

The lack of common context datasets and unavailability of implementations hinder a fair comparison of ZIP and ZIA schemes.

Reproduction of State-of-the-art ZIP and ZIA Schemes. The common practice of *not releasing* the source code of a scientific publication makes it impossible to compare state-of-the-art ZIP and ZIA schemes. To address this challenge, the schemes need to be reimplemented from the ground up. This is hindered by the limited amount of information on the implementation provided in the publication, missing system parameters (e.g., threshold values), and unspecified versions of the software used in the original implementation.

Collection of Real-world Context Data. The context dataset on which ZIP and ZIA schemes can be comprehensively tested should encompass the following points: (1) contain various context data (e.g., audio, luminosity) that is utilized by different ZIP and ZIA schemes, (2) be collected using a representative number of sensing devices realistically distributed inside the environment such as a room (e.g., on a door, under a table), (3) incorporate context data recorded by heterogeneous sensing devices which are both stationary and mobile, and (4) include context data from different scenarios (e.g., smart home) collected for a sufficient amount of time. Such a comprehensive dataset should then serve as a *benchmark* for future ZIP and ZIA schemes.

1.2.3 *Improve on Zero-interaction Pairing*

Despite the advantages of ZIP schemes in terms of usability and scalability, they suffer from prolonged *pairing time* and vulnerability to *brute-force offline attacks* on a shared secret key as well as attacks caused by the *predictable context* (e.g., replay attack). So far, these limitations have not been addressed by any ZIP scheme we are aware of.

State-of-the-art ZIP schemes require minutes and hours to establish pairing.

Reduction of Pairing Time. The existing ZIP schemes rely on the limited entropy of context to secure a shared secret key. Hence, they need to collect a sufficient amount of context data to provide adequate security, resulting in a prolonged pairing time, ranging from multiple minutes to several hours. To address this challenge, we need to decouple the entropy of a shared key from the entropy of the context as well as find a way to accumulate more entropy in a unit of time.

Brute-force offline attacks and threats due to predictable context endanger current ZIP schemes but have not been addressed so far.

Resistance to Brute-force Offline and Predictable Context Attacks. The brute-force offline attacks are feasible on current ZIP schemes because they utilize a cryptographic primitive called *fuzzy commitments* [175], which turns the entropy extracted from context data directly into the entropy of a shared secret key. The entropy of context data is often limited or biased, allowing the adversary to eventually guess the shared secret key.

Overcoming this problem requires seeking a more suitable cryptographic primitive that can produce a strong shared key from limited entropy of context data. The predictable context attacks (e.g., replay attack) are possible because context is usually represented by a single sensor modality such as acceleration. This modality capturing a specific physical phenomenon (e.g., human gait) generates data that changes within known boundaries. Thus, it can be inferred via indirect observations such as video analysis or replicated in a comparable environment. Addressing this issue requires strengthening context, for example, by constructing it from multiple sensor modalities.

1.2.4 *Improve on Zero-interaction Authentication*

Copresence detection is the core element of a ZIA scheme.

The key component of any ZIA scheme is a *copresence detection* mechanism, allowing one device to verify the physical proximity of another device based on their sensed context. Existing methods show reduced copresence detection accuracy in cases of low-entropy context (e.g., empty room with few events occurring) and insufficiently separated environments (e.g., adjacent rooms), threatening the security of ZIA schemes. Furthermore, state-of-the-art copresence detection methods require devices to be equipped with common sensors such as microphones, limiting the deployability of ZIA. To the best of our knowledge, none of the existing ZIA schemes offers a solution to these problems.

Copresence detection accuracy drops in low-entropy context and insufficiently separated environments.

Resistance to Low-entropy Context and Insufficiently Separated Environments. To provide reliable copresence detection in *low-entropy context* and *insufficiently separated environments*, we need to capture unique physical properties of the surroundings (e.g., geometry, distribution of obstacles), where copresence detection takes place. These properties should be commonly observed by nearby devices while being unobtainable outside their environment. Furthermore, capturing the unique properties of the surroundings should tolerate insignificant changes in the environment such as motion, which might be differently perceived by nearby devices.

Devices with heterogeneous sensors cannot use existing ZIA schemes.

No Common Sensors Requirement. The current copresence detection methods, and thus ZIA schemes, are limited to devices with shared sensors, which is not typical in the IoT, where many devices feature only one dedicated sensor (e.g., power meter). However, IoT devices are connected, thus they are ubiquitously equipped with communication technologies such as Wi-Fi, which can be leveraged for sensing applications [222].

1.2.5 *Bootstrapping Reproducibility*

Context data, implementation, and documentation are crucial for reproducibility of ZIP and ZIA schemes.

The current state of reproducibility in the domain of ZIP and ZIA is inadequate. To date, over a dozen different ZIP and ZIA schemes have been proposed. Out of these, we are aware of only one scheme that makes its implementation publicly available, whereas others release no source code at all. Also, none of the schemes provides open access to the context data on which they were evaluated, not to mention documentation

such as ground truth or the used software version required for reproducibility [23]. This makes it impossible to not only compare different ZIP and ZIA schemes but also scrutinize their findings, preventing further development in the domain.

1.3 OVERVIEW OF CONTRIBUTIONS

To address the above challenges attaining the goal of this thesis, we make the following major contributions: (1) we survey and analyze existing pairing schemes, including ZIP, identifying their key issues and proposing ways to tackle them, (2) we reproduce five state-of-the-art ZIP and ZIA schemes and evaluate them on real-world context data that we collect, (3) we develop a novel ZIP scheme that shortens pairing time while preventing attacks threatening state of the art, (4) we present a new copresence detection scheme that works on devices without shared sensors, achieving higher security under challenging conditions than existing methods, and (5) we ensure the reproducibility of our contributions (2)–(4) by publicly releasing the collected context data, source code of the data collection tools, evaluation stacks, and implementations of the schemes as well as documentation.

We analyze existing and design new ZIP and ZIA schemes.

1.3.1 *Survey and Analysis of Pairing Schemes*

We survey and analyze the current landscape of pairing schemes, being the first to include ZIP as a separate category in pairing. Specifically, we identify that the incomparability of existing pairing schemes stems from their design choices, namely using hardware interfaces as a starting point. Also, we find that numerous attacks on pairing schemes are caused by erroneous assumptions made about OoB channels. Therefore, we devise consistent terminology and a new system model for pairing that is based on *physical* channels, *human-computer interaction (HCI)* channels, and *application classes*. Using these three metrics, we survey existing pairing schemes with respect to their security, usability, and use cases, demonstrating that our approach leads to a more coherent comparison of pairing schemes and allows more sound reasoning about their security. In the course of our survey, we identify critical security, privacy, usability, and deployability issues, which should guide future research.

Our survey reveals principles of designing more robust pairing schemes.

1.3.2 *Evaluation of Zero-interaction Pairing and Authentication Schemes*

To assess realistic security and usability of ZIP and ZIA schemes, we reproduce five state-of-the-art schemes from the ground up and collect comprehensive context data in three real-world scenarios: *connected car*, *smart office*, and *smart office with mobile heterogeneous devices*. Our dataset contains over 239 GB of data from seven sensor modalities: audio, Wi-Fi and Bluetooth Low Energy (BLE) beacons, barometric pressure,

Evaluation under realistic conditions shows significantly reduced security and usability of existing schemes.

humidity, luminosity, and temperature collected by multiple off-the-shelf sensing devices. Based on our data, we demonstrate that the reproduced schemes are challenged by realistic conditions, especially in the cases of low-entropy context (e.g., a room at nighttime) and insufficiently separated environments (e.g., adjacent rooms), showing Equal Error Rates (EERs) between 0.6% and 52.8%, hence both their security and usability are imperiled. Also, the schemes suffer from low generalizability, thus they cannot be easily adapted to being used in different environments.

We release the first comprehensive context dataset for ZIP and ZIA.

1.3.3 *Faster and More Secure Zero-interaction Pairing*

We address the problem of prolonged pairing time and vulnerability to brute-force offline attacks on a shared key as well as attacks caused by predictable context (e.g., replay) from two sides. First, we adapt a recently introduced cryptographic primitive called *Fuzzy Password-Authenticated Key Exchange (fPAKE)* to reduce the amount of entropy that is required from context to secure a shared key. Second, we propose using *sensor fusion*—for the first time in ZIP—allowing us to (1) construct context from multiple sensor modalities, significantly decreasing its predictability and (2) obtain more entropy by accumulating it from heterogeneous sensors simultaneously. Hence, sensor fusion further assists fPAKE in shortening the pairing time. We implement our novel ZIP scheme called *FastZIP* on real hardware and evaluate it for the use case of intra-car device pairing based on context data from 800 km of driving that we collect. *FastZIP* demonstrates up to three times faster pairing time compared to state-of-the-art ZIP schemes, preventing brute-force offline and predictable context attacks.

We utilize a recent cryptographic protocol and sensor fusion to reduce pairing time and improve security.

We prototype and benchmark FastZIP on off-the-shelf IoT devices.

1.3.4 *Robust Copresence Detection Based on Channel State Information*

To achieve reliable copresence detection in low-entropy context (e.g., empty room with few events occurring) and insufficiently separated environments (e.g., adjacent rooms) without requiring common sensors, we design a novel copresence detection scheme named *Next2You* that is based on *channel state information (CSI)*. CSI is a robust sensor modality with valuable location-sensitive properties, and it is generated by default as part of the communication chain between Wi-Fi devices. We find that location-sensitive properties of CSI can be leveraged using *neural networks* to provide reliable copresence detection even under challenging conditions of low-entropy context and insufficiently separated environments. Also, the ubiquity of Wi-Fi in the IoT facilitates the deployment of *Next2You* on devices without shared sensors such as microphones. We implement our scheme on off-the-shelf smartphones and evaluate it on over 90 hours of CSI data that we collect in five real-world scenarios, demonstrating the advantages of *Next2You* compared to state of the art, its real-time performance, and resilience to attacks.

We leverage CSI and neural networks for copresence detection.

We build and evaluate Next2You on off-the-shelf smartphones.

1.3.5 *Ensuring Reproducibility of Our Contributions*

From our contributions, we collect the following context datasets: 239 GB (3 scenarios, 7 sensor modalities) in Section 1.3.2, 6 GB (1 scenario, 4 sensor modalities) in Section 1.3.3, and 10 GB (5 scenarios, 1 sensor modality) in Section 1.3.4. We make the majority of these context data publicly available via a general-purpose open-access repository for research artifacts *Zenodo*[8]. In addition, we host all intermediate results of our evaluations either on *Zenodo* or *Google Drive* to ensure reproducibility. Furthermore, we release the source code of our contributions in Section 1.3.3 to Section 1.3.4, namely data collection and processing tools, evaluation stacks, and implementations of the reproduced and developed ZIP and ZIA schemes on the *GitHub* page[9] of our research group. Both the released context data, intermediate results, and software are documented and supplied with the required metadata for reproducibility [23].

Over 75 GB of various context data that we collect in this thesis is publicly available.

1.4 OUTLINE

The rest of this thesis is structured as follows. Chapter 2 provides definitions and background on ZIP and ZIA and reviews related work. In Chapter 3, we present a comprehensive survey of secure pairing schemes, including ZIP. Chapter 4 follows, where we conduct the first comparative study of ZIP and ZIA schemes under realistic conditions. In Chapter 5, we design and evaluate a novel ZIP scheme, shortening pairing time and improving security. Chapter 6 presents a new copresence detection scheme that achieves better security than state of the art, especially under challenging conditions, and is easy to deploy. Finally, we conclude this thesis in Chapter 7.

8 https://zenodo.org/communities/zis/?page=1&size=20
9 https://github.com/seemoo-lab

2

BACKGROUND AND RELATED WORK

In this chapter, we provide necessary terminology used in this thesis, explain fundamental concepts in zero-interaction pairing (ZIP) and zero-interaction authentication (ZIA), and discuss related work.

2.1 DEFINITIONS

In the following, we define two major concepts: *context* and *colocation* to set the scope of this thesis as well as clarify the difference between *zero-interaction* and *context-based* security.

2.1.1 *Context*

The context is comprised of sensor readings taken in a specific environment at a point in time.

We define *context* as a set of *sensor modalities* (e.g., audio, signal strength, or button pressing event) collected by a device from its ambient environment within a timeframe [94, 98, 144]. We note that the context may not only be represented by sensing physical characteristics of the surroundings (e.g., room temperature) but also by recording sensor data resulted from (e.g., gait) [36, 309, 393] or generated by (e.g., pressing a button) [205, 206, 405] a user. In this thesis, we use the terms *context data* and *context information* interchangeably to mean sensor readings of the ambient environment supplied with metadata such as timestamps [266].

2.1.2 *Colocation*

Colocation definition depends on the use case of ZIP or ZIA.

Colocation refers to the level of physical proximity between devices defined by a ZIP or ZIA scheme to be sufficient for pairing or authentication. In ZIP and ZIA, device colocation is often determined as being inside an enclosed physical space such as an office, room, or car [98, 245, 365]. However, the granularity of colocation varies significantly depending on the use case, for example, in *wearables*, colocated devices are worn by a user [36, 229], while in a *smart home*, colocated devices reside inside the same house [144, 243]. In a number of ZIP and ZIA schemes, the term *copresence* is used instead of colocation to mean the same [58, 92, 365]. In this thesis, we use both colocation (colocated, non-colocated) and copresence (copresent, non-copresent) interchangeably.

2.1.3 *Zero-interaction Security*

We use the term *zero-interaction security (ZIS)* when jointly referring to ZIP and ZIA schemes that share two fundamental properties: (1) they rely on *context* to achieve either paring or authentication, and (2) they do not require *any form* of user interaction. With these two properties in mind, ZIP is defined as the process of establishing a shared secret key between previously unassociated devices, while ZIA is the process of determining colocation between two associated devices (i.e., those that already share a secret key). We highlight a difference between *ZIS* and *context-based* pairing/authentication. While both rely on context to achieve pairing or authentication, the former completely excludes user interaction, while the latter inherently requires it to generate or influence the context, for example, in the form of touching, shaking, tapping, swiping, or twisting devices as well as pressing buttons on them [90, 116, 205–207, 262, 285, 312, 396, 404, 405]. In this thesis, we refer to pairing and authentication schemes that utilize context as well as some form of user interaction (e.g., touching) as *touch-to-access* [173] methods. Thus, ZIS is a subset of context-based pairing and authentication schemes.

Context-based pairing and authentication is a superset of ZIP and ZIA, respectively.

2.2 OVERVIEW OF ZERO-INTERACTION PAIRING AND AUTHENTICATION

We present an overview of ZIP and ZIA, describing a generic setup and operation principle of the two approaches as well as threats they aim to mitigate. The fundamental idea leveraged in ZIP and ZIA is the *similarity of context* observed by colocated devices compared to non-colocated devices.

2.2.1 *Zero-interaction Pairing: Principle and Threats*

In ZIP, colocated devices that do not have any prior association nor any jointly trusted third party establish a shared secret key. Specifically, they (1) sense their context for a predefined timeframe, (2) translate the collected context into a sequence of bits (i.e., fingerprint), and (3) input their fingerprints into a key agreement protocol to derive a shared symmetric key [94, 144, 245]. The context fingerprints obtained by colocated devices serve as a *common source of entropy* for bootstrapping a shared secret key [144]. Despite being similar, the fingerprints of colocated devices are not identical due to imperfections in sensing, thus ZIP schemes utilize fuzzy cryptography [75, 174, 175] to correct mismatches between devices' fingerprints and ultimately share a secret key. Figure 3 shows a typical ZIP setup, where a number of smart devices (e.g., thermostat, laptop, fridge) located inside the same room utilize their shared context for pairing.

In ZIP a shared secret key is obtained in three steps: (1) context sensing, (2) fingerprint derivation, and (3) key exchange.

In this thesis, we focus on ZIP schemes that target pairing between two devices. While group pairing poses several challenges [84, 203, 251], they are mostly addressed on the level of key agreement protocols [75], thus we only mention that the group pairing can

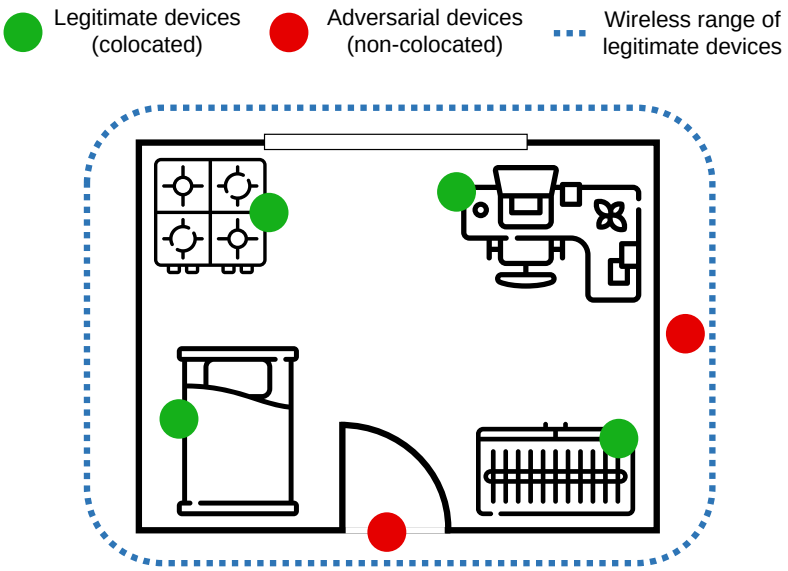


Figure 3: Typical ZIP setup encompassing a number of colocated smart devices that use their on-board sensors to obtain the shared context. The adversarial devices reside outside the space of colocated devices but within their wireless range (i.e., proximately close).

be enabled using novel cryptographic protocols such as Fuzzy Password-Authenticated Key Exchange (fPAKE) (cf. Chapter 5).

As in traditional *user-assisted pairing* schemes (e.g., entering a password), the main threats in ZIP are *impersonation* and *machine-in-the-middle (MITM)* attacks [93, 144, 245]. In the former attack, a non-colocated adversary, residing in the wireless range of a legitimate device, tries to pair with it in order to impersonate another legitimate device, violating *authenticity* of communication between the two legitimate colocated devices (cf. Figure 3). In the latter attack, a non-colocated adversary performs the impersonation attack on two legitimate devices simultaneously, attempting to intercept their communication, thus compromising *confidentiality* and possibly *integrity* of the data exchanged between the two legitimate devices.

Impersonation and MITM are primary attacks against ZIP in this thesis.

In this thesis, we focus on preventing impersonation and MITM attacks in ZIP, considering *denial-of-service (DoS)* attacks that target *availability* of pairing to be outside the scope of our work; we refer to [239, 394] for information on DoS mitigation in pairing. In Chapter 5, we detail how a non-colocated adversary may launch impersonation and MITM attacks on a ZIP scheme by eavesdropping, replaying, or mimicking the context of colocated devices. ZIP schemes are inherently vulnerable to colocated adversaries, which cannot be completely withstood [173, 389], thus we consider such adversaries to be outside the scope of this thesis.

We do not consider attacks against the availabilty of pairing such as DoS.

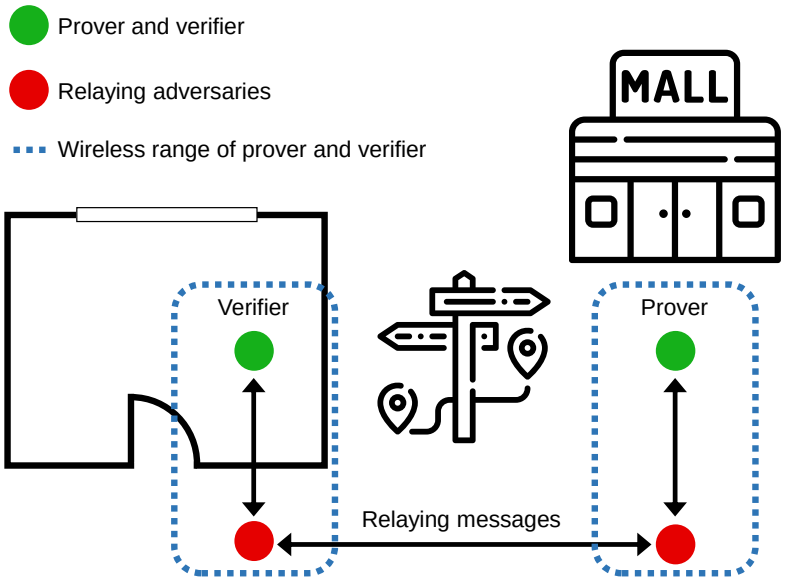


Figure 4: Typical ZIA setup, including the prover and verifier under the relay attack. To ensure colocation, the prover and verifier use the on-board sensors to obtain their context. The context similarity between the prover and verifier measured by the latter determines their colocation, allowing the verifier to authenticate the prover.

2.2.2 *Zero-interaction Authentication: Principle and Threats*

The verifier compares its sensed context against the prover's to decide on their copresence.

ZIA encompasses two devices: a *prover* and *verifier*, where the former tries to prove its colocation to the latter [92, 323, 364, 365]. Specifically, (1) the prover sends a copresence verification request to the verifier over a wireless channel such as Wi-Fi, (2) both devices capture their context using available sensors for a predefined timeframe (e.g., 5 seconds), (3) the prover transmits its context readings to the verifier, and (4) the verifier compares its own context readings with the prover ones and decides whether the two devices are colocated, thus the prover can be authenticated. In ZIA, both devices are assumed to share a secret key, hence their communication is encrypted and can also be integrity-protected (e.g., using authenticated encryption such as AES-GCM [76]), ensuring that an adversary cannot eavesdrop on or tamper with the transmitted context readings. The secret key between the prover and verifier can either be pre-shared or priorly established via secure pairing [93, 394, 404]. To compare context readings, the verifier can either use similarity metrics (e.g., cross-correlation) and check them against the fixed thresholds [136, 181, 317] or compute features from context readings (e.g., mean) and input them to a trained machine learning classifier [323, 364, 365].

ZIA can prevent relay attacks on wireless channels.

ZIA is designed to mitigate *relay attacks* on wireless channels [323, 364, 365], or it can be used as a second authentication factor [181, 229, 230]. In a relay attack, a pair of colluding adversaries, one located near the prover, another—near the verifier, forward

wireless messages between them to trick the verifier into believing that the prover is in proximity, violating its *authenticity* (cf. Figure 4). Hence, the colocation evidence based on context serves as the proof of physical proximity between the two devices, confirming the absence of the relay attack.

In this thesis, we consider relay attacks to be the main threat to ZIA, detailing how a non-colocated adversary can eavesdrop on or try to match the context of colocated devices to mount such an attack in Chapter 6. Similar to ZIP, ZIA schemes are vulnerable to colocated adversaries. While mitigation techniques have been proposed (e.g., by Truong et al. [365]), they severely degrade the practicality of ZIA schemes, thus we consider the colocated adversaries to be outside the scope of this thesis.

As in the case of ZIP, we do not consider colocated adversaries in ZIA.

2.3 RELATED WORK

In this section, we first highlight the differences between traditional user-assisted pairing schemes and ZIP, discussing our contributions in surveying and systematizing pairing schemes with respect to related work. Second, we compare ZIS and touch-to-access methods against several metrics (e.g., security, usability), exposing their trade-offs. Third, we review prominent ZIP and ZIA schemes, emphasizing our contributions to both approaches. Finally, we discuss the shortcomings of context-based pairing and authentication methods that require further attention from the research community.

2.3.1 *User-assisted and Zero-interaction Pairing*

The traditional user-assisted pairing schemes rely on user interaction to achieve pairing, for example, by requiring a user to input a password to devices, compare numeric codes, or press buttons [50, 93, 196]. In these schemes, user actions are the cornerstone of pairing, thus any mistakes, inattentiveness, or reluctance to ensure security from the user side result in vulnerabilities [166, 177]. It goes without saying that an average user is incapable of detecting or preventing active attacks on pairing, and they can easily become a victim of social engineering [170, 211, 366].

Bluetooth pairing is a classic example of user-assisted pairing.

Recall from Chapter 1 that ZIP has the following advantages over user-assisted pairing, which are especially beneficial in the Internet of Things (IoT):

- *Usability*. ZIP schemes perform pairing autonomously, excluding a user from the pairing procedure. Considering a rapid increase in the number of smart devices (e.g., per user [340] or per household [13]), many of which lack user interfaces, motivates the use of ZIP. Furthermore, IoT devices are often deployed in spots that are difficult to reach (e.g., smart light under the ceiling or an electronic control unit (ECU) inside a car trunk), making user-assisted pairing impractical.
- *Scalability*. The aforementioned growth of smart devices without user interfaces makes it infeasible for users to pair multiple devices simultaneously. In contrast,

Usability is a key advantage of ZIP.

ZIP allows pairing a practically unlimited number of colocated devices, which only need to be equipped with sensors.

ZIP relies on sensing to enable scalability and deployability.

- *Deployability*. ZIP schemes are built upon the off-the-shelf sensing capabilities of smart devices without requiring additional hardware (e.g., NFC) or significant processing power that is needed for centralized approaches such as public-key infrastructure (PKI) to perform expensive cryptographic operations.

In Chapter 3, we present a comprehensive survey on device pairing, covering both user-assisted and ZIP schemes [93]. In our survey, we, for the first time, study ZIP as a distinct category of paring schemes. Compared to previous surveys on device pairing, our contributions lie in: (1) a clear differentiation between physical and human-computer interaction (HCI) channels used for pairing and their analysis, (2) introduction of application classes (essentially use case categories) which impose different security/usability requirements on pairing schemes, and (3) revealing reasons for incomparability of different pairing schemes and their weak security. While Section 3.1 details our contributions relative to prior works, in the following, we discuss a few relevant surveys that have been published after our work.

In the following, we review three recent surveys that expand upon findings from our survey.

Caprolu et al. [43] survey ZIP and ZIA schemes that rely exclusively on short-range audio channels, covering both audible and ultrasound spectra. Differently from our work, they study a wider range of attacks on audio channels, including some novel attacks [403], as well as review defense mechanisms that can be built upon short-range audio channels. The authors come to a number of similar conclusions to our survey such as non-confidentiality of audio channels, insufficient attention to privacy issues, and promising potential of multi-sensing approaches.

In their work, Mayrhofer et al. [239] propose realistic adversary models for better analysis of pairing and authentication schemes—this was outlined as one of the future challenges in our survey. Specifically, the authors devise four adversary classes: "Zero effort", "Minimal effort", "Advanced Effort", and "Guaranteed Success" based on the adversary's capabilities (e.g., system knowledge) and required effort (e.g., time and computational resources) to break a pairing or authentication scheme. Using these adversary classes, they survey existing authentication and pairing schemes (both user-assisted and zero-interaction ones), coming to a conclusion that the majority of schemes relies on unrealistically weak adversary models, which explains numerous attacks against them in recent years.

Xu et al. [394] study ZIP schemes that are applicable in the IoT. Specifically, they use hardware interfaces as the base for their taxonomy, paying special attention to new sensing channels (e.g., relying on piezoelectric sensors or sensor fusion). Compared to our survey, the authors cover slightly more hardware interfaces as well as conduct comparative security and performance analysis of ZIP schemes. Despite identifying a number of common security and performance metrics (e.g., key generation rate, key agreement rate, randomness of keys), the authors merely summarize data presented in original publications. This does not yield a meaningful comparison because (1) many

schemes do not report such metrics, which is acknowledged by Xu et al. and (2) ZIP schemes, even those that utilize the same sensor modality, are evaluated in different scenarios and setups, preventing their baseline comparison [93]. In Chapter 4, we evaluate a number of state-of-the-art ZIS schemes in realistic scenarios showing that their security in the real world is significantly lower than originally reported [98].

ZIA has similar advantages over user-assisted authentication.

ZIA, similar to ZIP, exhibits the same usability, scalability, and deployability advantages over user-assisted authentication schemes (e.g., entering verification codes [181]). In this thesis, we do not make specific contributions in surveying or systematizing ZIA schemes, referring an interested reader to existing works such as [58].

2.3.2 *Comparison between Zero-interaction Security and Touch-to-Access Schemes*

Touch-to-access schemes require user input, whereas ZIS is completely autonomous.

We clarify the difference between ZIS and touch-to-access pairing and authentication schemes in Section 2.1.3: while both approaches utilize context for pairing or authentication, the former does not need any user interaction, whereas the latter inherently relies on it. In touch-to-access schemes, a user generates or affects devices' context, for example, by shaking, rotating, or tapping onto devices, which allows them to either capture user input with their sensors (e.g., accelerometer, button pressing event) [90, 205, 206, 209] or sense changes in the ambient environment caused by user actions (e.g., variation in signal strength due to user motion) [116, 207, 404, 405].

We utilize five metrics to compare ZIS and touch-to-access approaches.

Recently, a number of touch-to-access pairing and authentication schemes have been proposed [90, 116, 205–207, 262, 285, 312, 396, 404, 405]. Since they are natural competitors of ZIS schemes, we compare the two approaches in terms of security, usability, completion time[10], scalability, and deployability. We distinguish completion time as a separate metric because it encompasses both usability and deployability aspects, for example, a prolonged completion time may render a pairing or authentication scheme unusable, or time constraints of an application may make the scheme non-deployable.

FAR and FRR are established security and usability metrics, respectively in pairing and authentication systems.

Regarding *security*, touch-to-access schemes often exhibit False Acceptance Rate (FAR) below 1% [105, 116, 205, 206, 229] because they benefit from user input, while many ZIS schemes show higher FARs of around 2–5% [173, 323, 364]. However, touch-to-access schemes have an additional attack vector via mimicry or observation of user actions (e.g., tapping), which becomes a serious threat when assisted by computer vision techniques [36, 206]. In Chapter 5, we present a ZIP scheme that reaches FARs below 0.5% under attack, significantly outperforming state-of-the-art solutions, while in Chapter 6, we develop a copresence detection scheme (i.e., a key building block of ZIA) that achieves even lower error rates.

For *usability*, we compare False Rejection Rate (FRR) of ZIS and touch-to-access schemes. Differently from FAR, ZIS schemes show lower FRRs below 3% [173, 181, 323, 364], while touch-to-access methods often reach FRRs of around 10% [3, 90, 206, 229,

10 The time required to complete a pairing or authentication procedure. In the thesis, we also use terms *pairing time* or *authentication time* to mean the same.

Table 1: Comparison between zero-interaction and touch-to-access pairing/authentication schemes.

Metric	Zero-interaction		Touch-to-access
	State-of-the-art	Our contribution	
Security	○	●	◑
Usability	●	●	○
Completion time	○	◑	●
Scalability	●	●	○
Deployability	◑	●	◑

●–best, ◑–medium, ○–worst.

405]. Thus, we see a clear trade-off of user interaction: while it may help security, it inevitably harms usability because user input is prone to errors due to user mistakes, incorrect usage, or rushing behavior [93].

ZIS schemes suffer from prolonged completion time on the order of minutes and hours.

The *completion time* benefits from user actions, with many touch-to-access schemes showing pairing or authentication time within ten seconds [3, 116, 205, 206, 396]. For ZIS schemes, such time is generally longer—varying from several seconds [181, 310, 364] to minutes or even hours [144, 243, 309]—which is the price for operating autonomously without user interaction. In Chapter 5, we propose a novel architecture for ZIP schemes that significantly reduces pairing time. The recent advances in sensing allow utilizing radio frequency noise for ZIP, reporting pairing time below one second [173].

Touch-to-access schemes will not scale to the upcoming number of smart devices.

ZIS schemes have higher *scalability* compared to touch-to-access because they only require a user to install their devices instead of actively handling them. Given that an average household and user already have over ten and six smart devices, respectively [13, 340], and these numbers are envisioned to increase to dozens of devices [144], it is likely that touch-to-access schemes will overburden users, thus they can only apply to a limited set of devices. For example, touch-to-access schemes can be used to pair or authenticate a few central devices (e.g., smart hub and smartphone) reachable by a user to ensure stronger security, while other smart devices, distributed inside an environment such as a room, can rely on ZIS schemes.

The need for a "helper" device impedes deployability of touch-to-access schemes.

Considering the *deployability*, both ZIS and touch-to-access schemes require devices to be equipped with common sensors (e.g., microphones). Attempts have been made to enable pairing between devices with heterogeneous sensors in ZIS [144] and utilize minimum user interfaces (e.g., a button) for pairing and authentication in touch-to-access [205, 206]. In both cases, higher deployability results in either lower usability [205, 206] or prolonged completion time [144]. Many touch-to-access schemes require an additional helper device [116, 206, 312, 404] such as a smartphone or smartwatch, which assists pairing or authentication in the form of time synchronization, running cryptographic operations, or correlating sensor data with the target device to be

paired or authenticated. These touch-to-access schemes cannot function without additional hardware (e.g., user forgets to wear a smartwatch), hindering their deployability. In Chapter 6, we design a copresence detection scheme that does not require shared sensors, providing improved security than state of the art without compromising usability or completion time.

Table 1 summarizes the above comparison between ZIS and touch-to-access schemes. In this thesis, we focus on ZIS schemes making notable contributions to improving their security, completion time, and deployability.

2.3.3 *Review of Zero-interaction Pairing Schemes*

We review prominent ZIP schemes relevant to this thesis, describing the main drawbacks of existing solutions that we improve upon.

One issue that encompasses all existing ZIP schemes is methodological. Specifically, the reliance of ZIP schemes on context data (i.e., sensor readings collected by devices from their ambient environment) makes access to such data imperative to scrutinize and compare ZIP schemes. Unfortunately, none of the existing ZIP schemes releases context data on which they were evaluated, rendering the comparison of different ZIP schemes impossible. In addition to context acquisition, a ZIP scheme requires (1) *quantization*—the process of translating context data into fingerprint bits and (2) *key exchange* steps (cf. Section 2.2.1). Since these steps include complex signal processing and cryptographic protocols, the implementation of a ZIP scheme ensures the correct comparison between different schemes. We are aware of a single ZIP scheme that makes its implementation[11] publicly available [310]. Thus, the lack of context data and implementation of existing ZIP schemes hinders their *reproducibility* and *comparability*, hence their real-world security and utility cannot be assessed.

Reproducing and comparing different ZIP schemes is difficult due to the lack of context data and source code.

In Chapter 4, we conduct the first comparative study of ZIS schemes, reproducing five state-of-the-art ZIP and ZIA schemes, collecting real-world context data on which we evaluate them, and making both the scheme implementations and context dataset available for future research.

In our study, we consider two ZIP schemes. The pioneering work of Schürmann and Sigg [310] utilizes ambient audio as context, using fuzzy commitments [175] to establish a shared secret key between two colocated devices. This scheme relies on short snippets of ambient audio (i.e., 6 seconds), requiring a tight time synchronization between pairing devices. Such audio snippets are quantized to fingerprints of around 500 bits, which are input to the key exchange protocol based on fuzzy commitments. The authors evaluate their scheme in a set of environments such as an office, street, and student canteen, showing a margin of 20 percentage points in average similarity between fingerprints of colocated and non-colocated devices.

One of the first proposed ZIP schemes uses ambient audio.

11 https://github.com/dschuermann/fuzzy-pairing

Utilizing context changes over time eliminates the need for precise time synchronization between devices.

The work of Miettinen et al. [243] extends upon the scheme of Schürmann and Sigg, removing the necessity for tight time synchronization between pairing devices. Specifically, they devise a quantization method extracting fingerprint bits based on the changes in context commonly observed by colocated devices over time. In addition, they introduce a key evolution approach by performing a number of key exchanges (based on fuzzy commitments as in [310]) to strengthen the shared key between colocated devices over time, confirming their sustained copresence. The scheme by Miettinen et al. considers ambient noise levels and luminosity as context, and they evaluate it in smart home, smart office, and wearable scenarios, reporting the difference from 6 to 20 percentage points in average similarity between fingerprints of colocated and non-colocated devices.

From our study, we find that the above two ZIP schemes exhibit several issues, which are common to many other ZIP schemes, as we show in the following.

Average fingerprint similarities obscure actual security and usability performance.

First, the schemes report average fingerprint similarities and *not* actual error rates (i.e., FAR, FRR, or Equal Error Rate (EER)), which does not show their realistic security and usability performance. Nevertheless, presenting average fingerprint (or bit) similarity is a common practice followed by ZIP schemes [144, 145, 210, 262, 309, 393]. Miettinen et al. [245] come to the same conclusion as we do, indicating the pitfalls of average fingerprint similarities when evaluating ZIP schemes. Second, quantization methods extracting fingerprint bits from context data often produce fingerprints with entropy biases. This opens the door to security vulnerabilities (e.g., guessing attacks), which are not considered by existing ZIP schemes. A study on ZIP schemes utilizing human gait captured by accelerometers as context reports similar findings: out of five popular ZIP schemes, the quantization methods of four generate predictable fingerprints [36].

Fingerprints derived from context are often predictable.

Low-entropy context and sufficiently separated environments are two other hurdles found in our study.

In addition to not computing actual error rates and limited entropy of context fingerprints, we identify two more issues that decrease the security of ZIP schemes: (1) low-entropy context and (2) insufficiently separated environments. The former happens when ZIP schemes use low-entropy context recorded in the environment with little ambient activity (e.g., smart home at nighttime) to derive a fingerprint, which becomes predictable by an adversary. The latter occurs when non-colocated devices are proximately close (e.g., in adjacent rooms separated by a thin wall), hence they inevitably sense some shared context, for example, a loud sound in the case of audio.

Few solutions exist to tackle low-entropy context in ZIP.

Regarding low-entropy context, we are aware of few mechanisms utilized by existing ZIP schemes to address this issue. For example, Miettinen et al. [243] propose the *surprisal* to exclude fingerprints resulting from low-entropy context, but they did not empirically access it. In Chapter 4, we evaluate such a mechanism, finding that it often excludes a significant portion of fingerprints, hindering the scheme's availability without any clear security benefit. In Chapter 5, we introduce an *activity filter*—part of a ZIP scheme that selectively discards context data with insufficient entropy, ensuring that the obtained fingerprints contain enough entropy, and thus are unpredictable.

The issue of insufficiently separated environments is more complex, as it highly depends on the definition of colocation, which is often underspecified in ZIP schemes [98]. There exist two approaches to address this issue. First, is the use of highly varying context, for example, a wireless radio environment captured by channel state information (CSI). In the scheme by Xi et al. [389], two colocated devices leverage correlated CSI readings to establish a shared secret key. The use of such varying context containing ample entropy enables fast pairing and resilience to attacks. However, it limits the scheme's practicality requiring colocated devices to be within five centimeters from each other, which is unrealistically small distance even between wearables, not to mention smart home devices. The second approach that we present in Chapter 5 relies on sensor fusion, namely, utilizing a number of sensor modalities to capture a multidimensional context. As we demonstrate, this approach successfully withstands even proximately close adversaries without imposing any distance restrictions on colocated devices. However, it requires several sensors, trading off deployability for security.

ZIP in insufficiently separated environments requires the use of robust context.

Having identified a number of problems in existing ZIP schemes, we aim to improve on them. Recall from Table 1 that current ZIS schemes have lower security and longer pairing time compared to touch-to-access schemes. In the following, we review state-of-the-art ZIP schemes, discussing their security rationale and pairing time.

The abovementioned scheme by Miettinen et al. [243] strengthens security by utilizing sustained copresence of devices. Specifically, it is difficult for a non-colocated adversary to observe changes happening in the context of colocated devices over time (e.g., fluctuations in noise level). While this approach eliminates adversaries that may sense the context of colocated devices for a short time, it results in excessively long pairing time on the order of several hours, trading off pairing time for security.

Relying on context changes over time prolongs pairing time.

Han et al. [144] propose *Perceptio*, the first ZIP scheme that enables pairing between devices with heterogeneous sensors. Specifically, it relies on "numerically different yet contextually similar" events sensed by colocated devices, for example, a door knock that can be captured by both a microphone and accelerometer. The security of *Perceptio* is based on the inability of a non-colocated adversary to predict events randomly occurring in the environment of colocated devices or missing such events due to attenuation of sensed signals (e.g., audio wave). The colocated devices utilize timing information about the occurrence of common events to derive fingerprint bits. Thus, the number of bits obtainable from the context is determined by the frequency of such events, which is relatively low in typical smart home settings [144], leading to pairing time of hours or even days.

Time can be used as an invariant modality between devices with heterogeneous sensors.

Schürmann et al. [309] present BANDANA, a ZIP scheme that utilizes human gait captured by accelerometers as context to pair wearable devices carried by a user. The security of BANDANA relies on the unique properties of human gait, which is difficult to mimic [321, 393]. The authors extract fingerprint bits by computing energy differences between each gait cycle. With the optimal parameters, BANDANA reports pairing time of 96 seconds, considering that a user moves continuously within this timeframe. Any

Human gait captured by accelerometers is used as a unique context to pair wearables.

stops or standstill periods are not taken into account, suggesting that the pairing time in a realistic setting will be significantly longer.

ZIP schemes utilize other biometrics such as heartbeat.

Similar to BANDANA, Lin et al. [210] study the applicability of another biometric modality for pairing wearables, namely an inter-pulse interval (IPI), i.e., the interval between two peaks in a heartbeat signal, captured by piezo sensors. The security basis again relies on the uniqueness of IPI signals per individual and the difficulty of inferring them even with the aid of video analysis [210]. The authors estimate the pairing time to be approximately 40 seconds under ideal conditions (i.e., maximum entropy, high heartbeat rate, no sensor bias), which can hardly be guaranteed in realistic scenarios.

Offline attacks endanger current ZIP schemes.

The majority of ZIP schemes—those discussed above and many others [145, 173, 348, 362]—are based on fuzzy commitments [175] (or equivalently fuzzy vaults [174]) cryptographic primitives for key establishment. These primitives are by design vulnerable to brute-force offline attacks, which can only be prevented by *long fingerprints containing sufficient entropy*. Our results in Chapter 4 and findings from similar studies [36] suggest that the entropy of fingerprints derived from context is generally limited, threatening practically all ZIP schemes proposed to date with offline attacks. This issue is not addressed by any of the existing works on ZIP we are aware of.

The limited entropy of fingerprints does not only affect the security of ZIP schemes but also their pairing time. Specifically, any entropy biases in fingerprints need to be compensated for by obtaining extra bits from the context, which requires longer data collection, increasing the pairing time. None of the ZIP schemes described above takes this point into account when reporting their pairing time.

Lack of entropy in fingerprints and entropy loss due to error correction reduce security and extend pairing time, which is not considered by existing ZIP schemes.

Another issue that is not considered by existing ZIP schemes is the entropy loss due to error correction. Specifically, fuzzy cryptographic primitives (e.g., fuzzy commitments or vaults) allow fixing a number of mismatching bits between context fingerprints obtained by different devices by utilizing error correction codes (e.g., Reed–Solomon [282]). This requires transmitting the error correction information about context fingerprints in plain text, hence the adversary gains partial knowledge of them. Such entropy loss needs to be accounted for to maintain security by collecting more context data, yet again prolonging the pairing time. Miettinen et al. [245] are the first to point out this problem, however none of the existing ZIP schemes considers entropy loss due to error correction in their security and pairing time analysis.

In Chapter 5, we present a novel architecture for ZIP schemes that addresses the issue of offline attacks and prolonged pairing time. Thus, any existing or future ZIP scheme can adapt or borrow from this architecture to enhance security and reduce the time required to complete pairing. As a proof of concept, we implement and evaluate a ZIP scheme for pairing smart devices inside a moving car, reviewing similar works in Section 5.7.

2.3.4 *Review of Zero-interaction Authentication Schemes*

In the following, we discuss relevant ZIA schemes, indicating the weaknesses of existing solutions and how we address them in this thesis.

In ZIA, the state of context data and source code availability is similar to ZIP.

As in the case of ZIP, ZIA schemes suffer from the same methodological shortcoming of not releasing context data on which they were evaluated and their implementations, making a comparison of different ZIA schemes and further analysis problematic. We are aware of a single line of work [364, 365] that may provide access to the used context data upon request, which still does not qualify it as publicly available [147].

In our comparative study, we scrutinize three ZIA schemes.

Similarly to two ZIP schemes (cf. Section 2.3.3), we reproduce and evaluate three ZIA schemes in our comparative study presented in Chapter 4, releasing the collected context data and schemes' implementations to act as a benchmark for future research. The scheme by Karapanos et al. [181] relies on short snippets (i.e., 3 seconds) of ambient audio recorded by two colocated devices. Such snippets are split into 20 one-third octave bands, within each, a cross-correlation value is computed, to obtain a final similarity score that is the average of the bands' cross-correlation values. This similarity score is checked against a set similarity threshold to decide if two devices are colocated. The authors evaluate their scheme in a number of scenarios, ranging from a quiet office to a train station, showing high accuracy and robustness in distinguishing colocated and non-colocated devices (i.e., EER below 0.2%).

Sensor fusion has been explored in ZIA.

Truong et al. [364] propose using a number of sensor modalities, namely audio, Wi-Fi and Bluetooth signal strengths, and GPS as context for a ZIA scheme. They follow a different approach from Karapanos et al. [181], specifically, computing features from the context data (e.g., cross-correlation, Euclidean distance) and training a machine learning classifier on them to predict copresence. The authors evaluate their scheme in a range of environments such as lecture halls, cafeteria, and streets, reporting FAR and FRR below 2% when fusing all sensor modalities. A similar work by Shrestha et al. [323] explores environmental sensors: temperature, humidity, barometric pressure, and Carbon Monoxide (CO) to capture context. They also rely on context features computed from the collected sensor data, which are input to a machine learning classifier. The scheme is tested in a range of settings such as a library, office room, and café. The authors obtain a slightly higher FAR and FRR of up to 5% when fusing all modalities, arguing that environmental sensors trade lower security and usability for faster authentication time due to the shorter data collection period.

Existing ZIA schemes have been attacked exploiting low-entropy context.

The findings from our comparative study (cf. Chapter 4) for ZIA schemes are similar to that of ZIP. ZIA schemes are especially vulnerable to low-entropy context and insufficiently separated environments, leading to a several-fold increase in error rates. These identified problems are in line with the results of prior research, for example, Shrestha et al. [325] demonstrate that the scheme by Karapanos et al. [181] can be attacked with a success rate of up to 83.2% by injecting a typical smartphone sound (e.g., SMS ringtone) when the scheme is running in a quiet environment.

In the following, we discuss a few other security, usability, and deployability challenges of existing ZIA schemes. Truong et al. [365] and Han et al. [143] study how the colocated adversaries can be resisted in ZIA schemes relying on ambient audio. Truong et al. [365] propose using an acoustic room impulse response (RIR) to obtain a unique signature of physical surroundings, which is difficult to replicate by an adversary. Thus, colocated adversaries residing further than 50 cm away from legitimate devices can be mitigated. Han et al. [143] rely on the physical layer fingerprints of speakers and microphones (e.g., frequency response) to ensure that two legitimate devices are colocated. Specifically, during the fingerprinting phase, legitimate devices obtain their fingerprint profiles by emitting a sound with their speakers and recording it with their microphones, storing such profiles on a trusted server to be further used during ZIA. The main drawback of both these schemes is their dependency on both speakers and microphones, making such solutions impractical for the majority of IoT devices (e.g., smart lock, fitness tracker), which are not equipped with such hardware.

ZIA schemes aiming to prevent colocated adversaries require extra hardware, trading deployability for security.

In their work, Shrestha et al. [324] investigate the security of ZIA schemes in the presence of context manipulating adversaries. Specifically, they study how difficult it is to replicate context sensed by the prover near the verifier in order to trick the latter into believing that the former is colocated. The authors utilize off-the-shelf devices and software to, for example, stream audio via Skype or use a hairdryer to increase the temperature, showing that with such simple tools—context manipulation is feasible. However, they assume that an adversary has physical access to the verifier's sensors to perform manipulation, granting the adversary an unfair advantage.

Context manipulation has been shown feasible using off-the-shelf appliances.

The work by Shepherd et al. [317] investigates the applicability of different sensor modalities for ZIA in the domain of NFC payments. They find that for a typical NFC transaction time of 500 milliseconds, many sensors do not provide reliable context measurements that would allow two devices to detect their copresence. Thus, the authors identify the minimum duration of context acquisition, which determines the lower bound for authentication time for the majority of studied sensor modalities.

Using ZIA for NFC payments is difficult due to a short context acquisition window.

A practical limitation of many ZIA schemes is their reliance on more than one common sensor [229, 321–323, 364], which reduces their deployability, as they cannot be used on devices with heterogeneous sensors, or when one device lacks a specific sensor. Furthermore, the ZIA schemes that utilize context features computed from sensor data and classic machine learning algorithms (e.g., SVM, Random Forest) [229, 321–323, 364], require manual feature engineering, which is not only laborious and error-prone [188] but also imposes extra processing overhead on devices running a ZIA scheme, as context features need to be computed in real-time.

The reliance of ZIA schemes on several sensor modalities and machine learning leads to practical drawbacks.

In Chapter 6, we present a copresent detection scheme, which is the core part of ZIA, addressing the above challenges. First, we utilize a robust sensor modality, namely, CSI that significantly improves the accuracy of copresence detection in low-entropy context (e.g., smart home at nighttime) and insufficiently separated environments (e.g., adjacent rooms). Second, we use neural networks as a machine learning classifier,

which eliminates the need for manual feature engineering and allows us to learn the representation of CSI data (i.e., features) once and reuse it afterwards, facilitating the deployability of our approach. Third, CSI is mandatory information acquired by every Wi-Fi device. Thus, our approach does not require additional hardware sensors (e.g., microphones), and it can be deployed on heterogeneous devices (e.g., smart lock, thermostat) that are only equipped with Wi-Fi chipsets, which are ubiquitous in the IoT [352]. Section 6.1 mentions a few other works in the domain of ZIA, which are less relevant for our discussion here.

We improve security and deployability in ZIA.

2.3.5 *Shortcomings of Context-based Pairing and Authentication Schemes*

Despite the advantages of context-based pairing and authentication schemes over traditional user-assisted methods (cf. Section 2.3.1) they have one major drawback. Specifically, their reliance on context data requires access to the device's sensors and their readings. The sensor data collected from or near users as well as in locations where they reside, work, or spend time serves as an endless source of privacy issues, ranging from user tracking [31] and breaches of physical home security [277] to inferring user input (e.g., password) [330] and analyzing human speech [9].

Privacy issues stemming from sensing is the downside of context-based pairing and authentication.

To violate user privacy, there exist two attack vectors for infiltrating sensor data from smart devices: via (1) insufficiently controlled sensor permissions and (2) side channels. The first attack vector exploits weaknesses of permission-based access control (e.g., overprivileged) of Android, iOS, and other operating systems of smartphones and IoT devices to obtain sensor data without explicit user consent or by confusing users [171, 260, 270, 329, 386]. The mitigation techniques include fine-grained permission control, for example, having partial access to (obfuscated) data in addition to conventional "allow" and "deny" permissions, context-aware policies which take user context (e.g., location, data sensitivity, utility) into account when granting access to sensor data, and transparently binding app permissions to their functionality [171, 260, 270, 329].

User privacy can be violated due to loose sensor permissions and side channel attacks.

The second attack vector exploits physical characteristics of the sensed signals to either eavesdrop on the sensor data (e.g., via signal leakage) or inject a malicious stimulus (e.g., vibration) to affect sensor readings [45, 117, 367]. To mitigate the side channels, a number of techniques have been proposed ranging from masking signals that eliminate acoustic emanations [6, 8] to obfuscating or filtering sensor data to reduce the effect of malicious stimuli injected by an adversary [330, 367] as well as shielding sensors themselves and applying dynamic sampling strategies [117, 367].

Side channel attacks become increasingly common but are difficult to defend against.

The sensitivity of collected sensor data, in addition to numerous attacks to infiltrate it, urges the need for wider adoption of lightweight encryption mechanisms, ensuing its confidentiality on multiple layers (e.g., physical, application) as well as the more pervasive use of security extensions (e.g., TrustZone) for processing sensor data.

Encrypting and securely processing sensor data help protect user privacy.

Part II

CONTRIBUTION

3

SURVEY AND ANALYSIS OF PAIRING SCHEMES

Secure Device Pairing (SDP) has been proposed as a practical alternative to public-key infrastructure (PKI) in ubiquitous computing, allowing two wireless devices that do not have any prior association nor any jointly trusted third party to establish a shared secret key [336]. To establish this key, the two devices (1) execute an unauthenticated key exchange such as Diffie-Hellman (DH) and (2) ensure the authenticity of each other's keys using an auxiliary *out-of-band (OoB)* channel [17, 196]. The OoB channel is used to prevent *machine-in-the-middle (MITM)* attacks, which can easily be mounted on an unauthenticated key exchange performed via a wireless channel [196, 247, 264].

SDP bootstraps secure communication avoiding MITM attacks using OoB channels.

After the pioneering work of Stajano and Anderson [336], many pairing schemes have been proposed, most of which target two classic use cases: (1) paring personal devices of a single user (e.g., TV and headset) and (2) pairing devices of different users (e.g., two smartphones) [50, 177, 187, 370]. With the advent of the Internet of Things (IoT), SDP has become an important mechanism for securing wireless communication between smart devices [296, 328], which is demonstrated by an extensive standardization effort of pairing schemes by Wi-Fi and Bluetooth alliances [29, 87]. However, there are three challenges to applying classic pairing schemes in the IoT. First, the skyrocketing number of devices imposes a prohibitive user effort if pairing requires any user involvement (e.g., entering a password) [296]. Second, the diversity of IoT devices (e.g., processing power, user/wireless interfaces) renders classic pairing approaches impractical, as many devices cannot satisfy the common hardware requirements such as having a display [133]. Third, the user-to-device and device-to-device interaction patterns in the IoT are more varying such as pairing without user involvement [243, 310], thus the pairing assumptions (e.g., adversary model) differ from classic pairing use cases.

Securely pairing IoT devices poses new challenges for existing schemes.

The sound comparison of existing pairing schemes is difficult. Prior SDP surveys struggle to fairly compare proposed pairing schemes even over a single metric (e.g., security and usability), concluding that no universal pairing approach exists [50, 177, 187, 196, 247]. We find that such incomparability is caused by the lack of common information about a pairing scheme required to make a comparison, which happens because of two reasons. First, available hardware interfaces (e.g., buttons) have been the main reason for designing another pairing scheme [3, 272, 332, 333]. Hence, many schemes focus on a specific use case in a limited setting, without considering findings of previous pairing proposals. Second, Ion et al. [166] show that the choice of a pairing scheme is context- and environment-dependent, thus the information on context and environment (which is rarely provided) is imperative to enable schemes' comparison.

Comparing different pairing schemes is hard due their hardware-based design and context-dependency.

Two other issues in SDP are caused by the lack of common understanding on the definition and properties of an OoB channel. First, there exist a dozen definitions of an OoB channel [50, 196], which apply mixed terminology and overlapping adversary capabilities [247, 254]. This is a direct consequence of the design flow based on hardware interfaces. Second, many schemes make unrealistic assumptions about the security properties of an OoB channel (e.g., confidentiality of data transmission), resulting in numerous successful attacks [10, 135, 137, 344].

Our goal is to facilitate the design of robust pairing schemes for the IoT.

The incomparability of different pairing schemes and their recurring security incidents suggest issues in the existing design flow of pairing schemes. Also, we witness the proliferation of *zero-interaction pairing (ZIP)* schemes, which enable pairing between IoT devices without user involvement by sensing devices' ambient environment (e.g., audio) [243, 310, 372, 406]. ZIP schemes aim to improve usability, scalability, and deployability of pairing in the IoT. Hence, in this chapter, we systematize knowledge on SDP, shedding light on how to design more robust and comparable pairing schemes, including a new class of ZIP schemes. We make the following contributions:

- A system model and consistent terminology that facilitates precise description and reasoning about SDP schemes by considering the three components:
  - Physical (PHY) channels,
  - Human-computer interaction (HCI) channels, and
  - Application classes.
- Classification of the existing SDP schemes using this model.
- Identification and analysis of systemic security weaknesses commonly found in such schemes, revealing areas where future SDP research is required.
- Revelation of the rarity with which privacy is considered among current SDP schemes.
- Principles for designing robust SDP schemes.

3.1 RELATED WORK

Quantitative analysis of different pairing schemes is questionable.

In the literature, several surveys have investigated different aspects of SDP. Kumar et al. [196] present the first comparative study to quantify usability and security of various pairing schemes. Our work reveals that quantitative comparison of different SDP schemes is questionable due to the previously taken design decisions, and we qualitatively address the design aspects of SDP to enable meaningful comparison of different SDP schemes. Two other studies from Kobsa et al. [187] and Kainda et al. [177] focus more closely on usability and the role of user actions to achieve security in SDP. Our work has wider scope because we consider the role of the user as one of the fundamental design aspects of SDP in addition to physical communication media and particular use cases.

The work of Mirzadeh et al. [247] provides an extensive survey on security and performance of different cryptographic protocols used in various SDP schemes in addition to presenting classification of OoB channels. In our work, we devise a more fine-grained classification of communication channels in SDP by differentiating between PHY and HCI channels and focus on security issues of those channels instead of cryptographic protocols. Since security weaknesses of various communication channels have resulted in numerous successful attacks on different SDP schemes, we consider our qualitative analysis of those channels as a novel contribution. In their survey, Chong et al. [50] present different modes of user interaction for SDP and analyzed a vast number of SDP schemes using this taxonomy. We refine their findings to classify HCI channels and, additionally, present a set of common security and usability properties to coherently analyze those channels and the SDP schemes relying on them, which has not been done before.

To date, a few surveys on device pairing have separately covered security and usability issues.

In our survey, we focus on SDP schemes proposed for two (or several) devices and consider multi-device SDP outside the scope of this chapter. In comparison to the prior work, our survey is innovative in three aspects. First, we devise a novel system model for SDP, which addresses the security weaknesses of the existing generation of SDP schemes. Second, we propose a new approach to design SDP schemes, which enables their meaningful comparison. Third, we provide a deep insight into the current state of SDP from the point of PHY channels, HCI channels, and application classes as well as present an overview of SDP challenges and perspectives in light of the upcoming IoT.

We propose a new system model for pairing facilitating meaningful comparison of different schemes.

In this section, we have reviewed the related work on SDP and highlighted the contributions of our survey. In the next section, we present our system model and taxonomy.

3.2 SYSTEM MODEL AND TAXONOMY

In this section, we first give a high-level overview of a generalized pairing procedure together with widely-used notations. Second, we address ambiguity in the current terminology by providing clear definitions to describe SDP. Third, we present a system model that illustrates the notion and properties of communication channels as well as facilitates a more unified approach towards the design of pairing schemes. Fourth, we discuss threats in SDP with respect to our system model. We conclude by explaining our taxonomy, which is used to systematize and evaluate proposed pairing schemes.

3.2.1 *Generalized Pairing Procedure*

Traditionally, the pairing procedure has been considered as depicted in Figure 5. The scenario consists of two devices, *D1* and *D2*, which do not share any prior knowledge and would like to pair. That is, two devices need to exchange some secret information, ensuring it came from the correct party, and is not obtained by any third party. In order

A pairing procedure consists of three main steps.

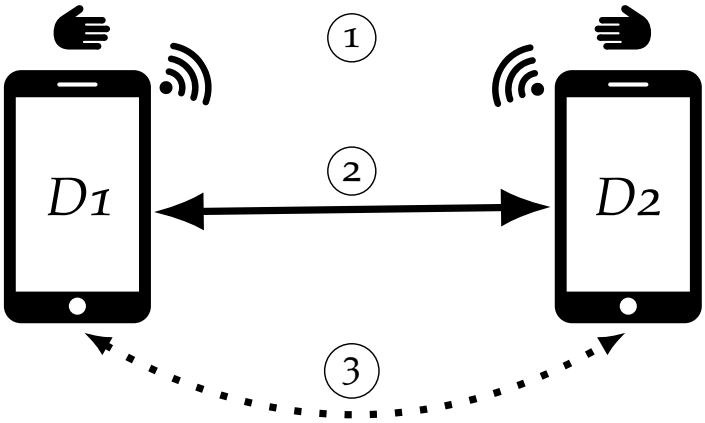


Figure 5: Generalized pairing procedure.

to achieve pairing, three steps need to be followed: ① *discovery*, ② *secret exchange*, and ③ *verification*. In the first step, *D1* and *D2* become aware of each other, which can happen either automatically, for example, using Bluetooth discovery or with user assistance, for example, via physical contact. During the second step, both devices exchange some cryptographic material, for example, public keys or a password, which can later be used to establish secure communication. In the final step, both parties verify the obtained secrets to ensure that the process has not been compromised by an attacker.

To provide a better understanding of the interactions presented in Figure 5, we examine the commonly used notation for SDP. Three main terms are commonly used in the literature: (1) *in-band channel*, (2) *out-of-band channel*, and (3) *user interaction*. By employing the generalized pairing procedure shown above, we demonstrate how those concepts apply using a well-known example [17]. Two devices discover each other, after having been brought together physically by a user (3). Then, they exchange hashes of their public keys over an auxiliary channel (2), followed by a mutual transfer of the corresponding public keys over a wireless radio link (1). Of course, the given example illustrates just one possible case of how the pairing flow can be implemented. There are other variants, for example, where the discovery can happen without user interaction as in [243], or the secret key is first transmitted via the in-band channel and subsequently verified via the OoB channel as in [123].

Traditionally, pairing is achieved with user assistance utilizing in-band and OoB channels.

To gain a deeper understanding of the major pairing concepts, it is important to specify the characteristics of in-band and OoB channels that have been traditionally discussed by the research community. The pioneering work of Balfanz et al. [17] states two related properties that an OoB channel should possess: *demonstrative identification* and *authenticity*, and also that *confidentiality* should not be assumed.

The authenticity is the defining characteristic of the OoB channel, ensuring the infeasibility of forging communications over an OoB channel without being detected, which makes OoB communications so valuable in SDP. In practice, this implies that

OoB channels must possess demonstrative identification, that is, it must be easy to demonstrate that the OoB communication is occurring between the intended parties, for example, by showing the display of a device to another user. Demonstrative identification, thus, implies that the devices must be brought sufficiently close to one another to allow their mutual positive identification by their users. While OoB channels should not be assumed to offer confidentiality, a number of the surveyed SDP schemes depend on the OoB channels being confidential.

Authenticity is the key property of the OoB channel.

The in-band channel, in contrast, has been generally regarded as a communication channel with relaxed security characteristics. That is, it refers to a wireless radio link, which is easily accessible by a powerful attacker [70], and thus deemed as inherently *insecure*.

In-band channel is considered insecure assuming a Dolev–Yao adversary.

So far, we have discussed the core components of SDP along with their prime purposes and vital properties. Yet, there are no precise definitions of the OoB channel and user interaction, which are common in the field. We consider this point to be the principle weakness of the existing terminology. Many researchers have described the OoB channel to be a side PHY channel which is either human-perceptible and/or directly controlled by a user [196]. Nevertheless, there is a number of pairing schemes that intrinsically rely on user actions to accomplish pairing [177]. One example of the latter is Secure Simple Pairing [28], which is a de facto standard for connecting Bluetooth devices securely. Consequently, such disparity results in a situation where one part of the community only considers physical media as the OoB channel, while neglecting the user-assisted channels and vice versa. In addition, communication channels differ in fundamental ways, hence assumptions about the media and attacker models vary significantly, and these are not straightforward to align.

The lack of consistent terminology is a major weakness in the pairing field.

Another issue is that the boundary between the human-assisted OoB channels and user interaction is often blurred. That is, the latter is a more general term that can include the former. However, the essential purpose of the OoB channel is to provide some form of data authenticity. Specifically, a human operator can assist in initiating device pairing during the discovery step, for example, by colocating devices, aligning them, or enabling physical contact. Yet, we argue that only explicit actions, which directly affect the security of the pairing scheme, should be considered as the OoB channel.

The role of user interaction in pairing is often blurred.

3.2.2 *Defining Secure Device Pairing Terminology*

A specific challenge to the comparison and analysis of SDP schemes is the lack of accepted terminology covering such schemes. We, therefore, present the terminology that we use in the remainder of this thesis, both for clarity of explanation here and as a suggestion for a common vocabulary to facilitate communications among practitioners in the future.

We define a number of terms to enable a common vocabulary for SDP.

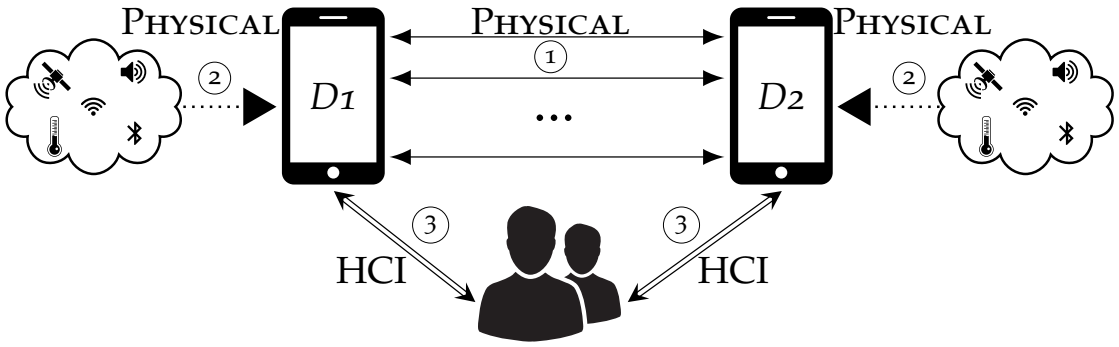


Figure 6: System model for secure device pairing.

- *Pairing* refers to the establishment of a secure communication channel between two or more devices.
- An *application class* represents a particular pairing scenario that is determined by the degree of involvement and level of control that a user has over the pairing devices. An application class covers use cases that share broadly similar security threats and objectives.
- An *SDP scheme* consists of the procedures, cryptographic protocols, and the motivating application class required to securely pair devices.
- An *SDP method* or *SDP procedure* is the sequence of actions required to execute an SDP scheme. While considering method and procedure interchangeable, we avoid the synonym protocol because of the strong association of this word with cryptographic protocols.
- A *party* is someone or something who controls of one or more devices that participate in an SDP procedure.
- A *security domain* is the set of devices, data, policies, and intentions that a single party controls. That is, every device belongs to a security domain, but there may be more than one security domain involved in a given application class.
- A *channel* is a means by which communications occur in an SDP scheme, whether over a physical medium or through an HCI.
- An *HCI channel* is a means of communication where a user acts as the channel by which the communications occurs by undertaking some form of interaction with the devices involved. This could take the form, for example, of a user reading information from the display of two devices and entering confirmation that they match into one of those devices.
- A *PHY channel* is a communication channel that allows data transmission or acquisition over a physical medium. PHY channels can be described by their objective physical characteristics, and where the information is not transferred by a user, that is, it is not an HCI channel.

3.2.3 *System Model*

To address the issues in SDP mentioned previously, we introduce our system model depicted in Figure 6. The objectives of our approach are threefold. First, it takes into account a set of diverse interactions that appear in the context of IoT. Second, it aims to resolve the ambiguities in the pairing concepts, which are currently present in the field. Third, our model facilitates a more unified procedure for the pairing design.

Our system model contains three main components. Particularly, there are (1) *two devices* to be paired *D1* and *D2*, (2) *a varying number of users*, and (3) *the ambient environment* in which device pairing takes place. In addition, several types of distinctive interactions can happen between those elements. First, *D1* and *D2* can communicate with each other by means of various wireless technologies such as Wi-Fi or Bluetooth ①. Second, a device can obtain information about its ambient environment (as in the case of ZIP) such as temperature or location via the sensing capabilities ②. Third, the connection between a human operator and the respective device is established in the form of HCI ③. We further consider the relationship between a user and a pairing device. Specifically, a human operator can either control both devices involved in SDP, a single one, or none at all. Implied in the system model, is the purpose for which the devices are being paired, that is, a use case.

Physical channels, HCI channels, and application classes are the three building blocks of our system model for pairing.

From the above, we can consider an SDP scheme as consisting of the automated communications between two devices over conventional PHY channels, plus the human-assisted communications between the devices over HCI channels. Pairing of devices always occurs for a purpose, that is, it happens within the context of an application class. We, therefore, use three key concepts as the foundation for our system model:

- PHY channels.
- HCI channels.
- Application classes.

Figure 7 illustrates the relationship between those concepts, that is, we consider the channels both PHY and HCI to be orthogonal to the application classes.

The first two concepts specify two fundamentally different types of interactions that can be utilized by a pairing scheme. With this in mind, we further analyze PHY and HCI channels independently to identify the most important features of each class and expose the trade-offs involved. To account for both types of interactions, we understand "device" to mean any physical device that possesses one or more communication channels that can be used to connect to the outside world. SDP is achieved using some set of such channels. Figure 8 depicts an abstract visualization of such a device concept, including a comprehensive list of PHY and HCI channels. A rigorous discussion covering each channel category in detail is provided in Section 3.3 for PHY channels and in Section 3.4 for HCI channels, respectively.

A pairing device possesses both physical and HCI interfaces to the outside world.

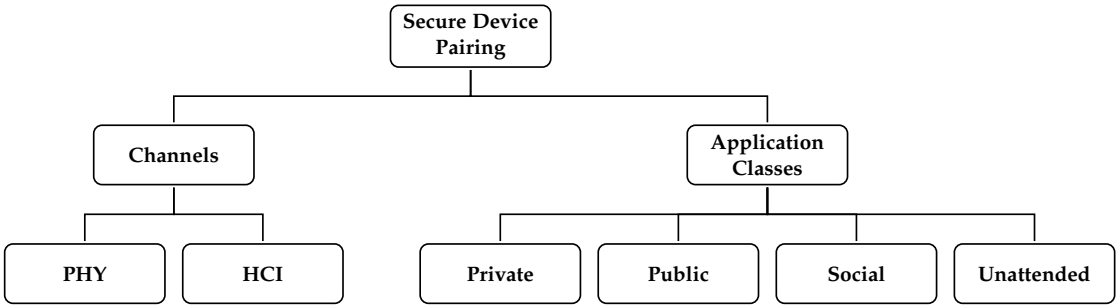


Figure 7: Taxonomy of secure device pairing.

We distinguish four application classes, covering all existing pairing schemes.

As for the application classes, we identify four different cases which cater to classify all the pairing schemes proposed up-to-date. The categories are as follows: (1) *private*, (2) *public*, (3) *social*, and (4) *unattended*. The private class corresponds to a "classic pairing" case where a single user either owns or directly controls two devices that ought to be paired. The public class is related to a single user possessing one device, where the user performs the pairing with some third party infrastructure, for example, a payment terminal, over which they have no control. The social class incorporates two users who would like to securely pair their corresponding devices. The unattended class covers the use case of ZIP, where two colocated devices (e.g., inside the same room), which may belong to the same or different ownership domain such as a person or organization, pair without user involvement.

An application class allows identifying common security and usability requirements for pairing schemes.

For each application class, we present distinct interaction patterns demonstrated by instantiating our system model (cf. Section 3.5). Furthermore, we identify commonalities in the form of adversary capabilities as well as security and usability implications that have to be taken into account for a particular application class. Consequently, it is possible to determine a set of common security and usability properties shared by a group of pairing schemes that have been designed with a specific application class in mind. Section 3.5 explores the potential for such application classes to facilitate the design of better, more coherent SDP schemes.

3.2.4 *Overview of Threats*

The formulation of a detailed adversary model is beyond the scope of this chapter. However, a general overview of relevant threats is still required to meaningfully compare different SDP schemes. We consider two pairing devices D1 and D2, as depicted in Figure 6, assuming that they are not compromised with malware and controlled directly by their respective owners. The goal of an adversary is always to undermine SDP. We focus primarily on attacks against authenticity and confidentiality, as they are the most relevant to the SDP process. We consider two broadly representative

We consider attacks against authenticity and confidentiality in pairing.

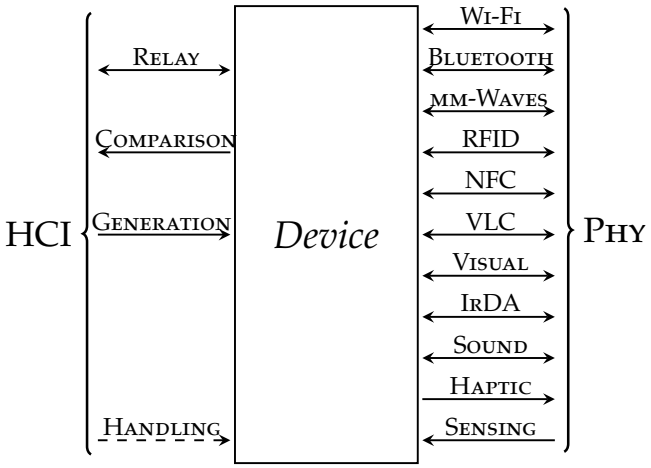


Figure 8: Pairing device with PHY and HCI channels. This model is independent of application classes.

classes of adversaries seeking to undermine SDP: first, those who want to attack via PHY communications channels, and second, those who wish to attack via HCI channels.

3.2.4.1 *Attacks on PHY Channels*

MITM, eavesdropping, and relay attacks are common on physical channels.

An adversary who exploits PHY channels can mount many different attacks. The majority of these attacks are considered particularly severe in SDP, as they undermine the basic assumptions of authenticity and confidentiality that SDP seeks to establish. PHY channels used for data transmission are especially vulnerable to attacks on confidentiality such as MITM and eavesdropping attacks, while PHY channels used for data acquisition, for example, environmental sensing, are susceptible to attacks that undermine authenticity such as relay attacks. For example, the adversary can reproduce the relevant sensor readings by manipulating the temperature or humidity to match that of a remote location.

We do not focus on attacks undermining the availability of pairing such as DoS.

Regarding availability and integrity properties, the adversary can jam or otherwise disturb the communication media, which can result in denial-of-service (DoS) attacks. While DoS is a legitimate security issue because it can prevent SDP, it cannot lead to a false sense of security, that is, where two users mistakenly believe that their devices have securely paired. Hence, DoS can deny availability but not subvert authenticity or confidentiality. This stands in contrast to MITM, eavesdropping, and relay attacks where the devices will behave as though they have paired securely, when, in fact, they have not.

More sophisticated attacks such as selective jamming, disturbing signal parts, or bit-flipping can force retransmissions and increase adversary's chances of compromising SDP. Time-based attacks, for instance, replaying previously captured packages or delaying messages in transmission can also impair the SDP process. In our survey, we

are primarily concerned with attacks on PHY channels that can subvert authenticity and confidentiality, that is, MITM, eavesdropping, and relay attacks. More exotic attacks on various PHY channels are discussed in the respective sections, where relevant.

3.2.4.2 *Attacks on HCI Channels*

There are two main classes of HCI adversaries: the (1) *external attacker*, that is, someone who is not a legitimate pairing party and (2) *internal attacker*, that is, one of the pairing participants.

Observational threats aided by modern technology may compromise HCI channels.

The external adversaries aim to violate the authenticity and confidentiality of the HCI channel by observing user interaction in order to be able to covertly participate in the communications between the pairing devices. For example, an external HCI adversary may monitor the HCI channel during a pairing event in order to also derive the cryptographic material being exchanged during the pairing process.

In contrast, the internal adversaries already have such access but seek to use the process of pairing in order to extract sensitive data from the other participant's device. This may take the form of social engineering. When the attacker's primary objective is to participate in the pairing in order to undermine the privacy of the other party, we label them as *honest-but-curious adversaries*. For example, an honest-but-curious attacker may seek to obtain the telephone number of the other party, contrary to their wishes.

3.2.5 *Taxonomy*

We propose a taxonomy built upon the three key concepts that have been introduced and described above: PHY channels, HCI channels, and application classes, which makes the following two contributions to the field of SDP. First, it provides systematization based on these three key concepts drawn from the design space of SDP. Second, it enables qualitative assessment and meaningful comparison of different pairing schemes. The structure of our taxonomy is given in Figure 7.

In order to investigate devices' channels, we adopt the following framework. First, we identify the most important characteristics that are relevant to a communication channel. For PHY channels such parameters are measurable and objective, whereas the HCI channels are represented by more subjective metrics. Second, we focus on three sets of properties that are vital in the context of SDP, namely: *security*, *usability*, and *ease of adoption*. Third, based on this structure we review the existing pairing schemes to reveal how these properties are addressed by a particular scheme, and what are the trade-offs. With regard to application classes, we first provide a thorough description of each class followed by a discussion on its specific security and usability implications. Second, we map the proposed pairing schemes to the corresponding application classes to provide a more systematic overview of the current state in the field.

We characterize physical and HCI channels with respect to their security, usability, and ease of adoption.

In this section, we have presented and motivated our system model as well as provided the taxonomy, which we use to survey existing SDP schemes. In the next section, we review PHY channels and the corresponding pairing schemes.

3.3 PHYSICAL CHANNELS

We study properties of physical channels to justify how easily they can be attacked.

The PHY channels listed in Figure 8 allow a device to communicate with other devices as well as interact with the ambient environment. We base our analysis on several aspects to conduct the meticulous investigation of PHY channels. Specifically, we consider *channel characteristics*, *known attacks*, and *ease of adoption* to compare different types of communication channels. With regard to security, we identify a set of channel properties that have a direct security impact, namely they indicate the amount of effort necessary to intercept the pairing process. Furthermore, we discuss pairing schemes that utilize each surveyed channel to show how it is employed to achieve pairing. Finally, we summarize the most important findings, presented in Table 2, and discuss key issues from our study of PHY channels.

3.3.1 *Channel Characteristics*

We employ six metrics to characterize a physical channel.

The physical nature of wireless communication channels is described by different properties. The most important ones are the *frequency range* of transmitted electromagnetic or mechanical waves and the achievable *data rates*. In fact, the data rate is defined by technical specifications of a communication protocol such as the available bandwidth, coding, and modulation schemes, rather than the underlying PHY channel. Hence, we consider this as the amount of transmitted information in a unit of time using common state-of-the-art protocols. The frequency and bandwidth of a particular channel comes along with certain propagation characteristics such as *coverage*, human *perceptibility*, *penetration*, and *line-of-sight (LoS)* propagation. We put these properties into security perceptive by refining the observations made by Balfanz et al. [17] on location-limited channels.

3.3.1.1 *Coverage*

Short coverage benefits security.

Defines the maximum nominal distance at which a signal on a PHY channel can be successfully received, that is, differentiated from noise. Naturally, increasing the sensitivity of receivers leads to wider coverage, which makes it a questionable security property. However, the amount of effort and cost required to receive a signal far outside the nominal range are high. For an adversary, a PHY channel with smaller coverage is harder to access, and thus attack.

3.3.1.2 *Perceptibility*

Human perceptibility can help alert a user about the ongoing attack.

Specifies whether a user can perceive the fact of data transmission through major human senses such as sight, hearing, and touch [69]. This property can have both advantageous and harmful repercussions. On the one hand, a benign user can be alerted if some unexpected interaction occurs. On the other hand, an attacker can easily observe such type of communication without any specialized equipment. Nevertheless, this trade-off can be leveraged by secure protocol design so that the benefits of human perceptibility greatly outweigh the risks.

3.3.1.3 *Penetration*

Physical channels that are blocked by obstacles are less easy to eavesdrop on.

The propagation properties of a particular channel depend on the underlying physics. Both electromagnetic and mechanical waves are subject to diffraction, reflection, refraction, scattering, and absorption [343, 356]. With these effects a signal can be partially or entirely blocked, which hinders the communication. Penetration characterizes the ability of the signal to propagate trough solid obstacles such as walls, doors, and furniture. Thus, a communication channel with high blockage, that is, low penetration, effectively limits the operation range of the channel, thus hampering the attacker's ability to access it.

3.3.1.4 *Line-of-sight (LoS)*

Line of sight requires an adversary to intercept within the transmission beam.

Signals propagate throughout the environment and find multiple paths from the transmitter to the receiver over several reflections, diffractions, and refractions. As we stated above, channels are differently affected by these factors. Generally speaking, with higher frequency diffraction and refraction become less significant. With low multi-path components, a channel becomes dependent on LoS, which is a direct path between a transmitter and receiver without obstruction. Correspondingly, non-line-of-sight communication does not require an obstruction-free path between a transmitter and receiver. The LoS enables predominantly directional communication, which hinders the ability of the adversary to stealthily intercept the channel from outside the main transmission beam.

3.3.2 *Known Attacks*

Various attacks have been reported on physical channels, which we summarize.

To better understand the security implications of PHY channels in SDP, we summarize the most prominent attacks that have been reported on various physical media. This list of attacks is by no means exhaustive but provides an overview of common possible attacks vectors. In the literature, many attacks on widespread wireless radio channels can be found. We summarize those and additionally present security implications in other communication channels such as visible light and audio. Overall, our study

provides a deep insight into existing threats and outlines differences in vulnerabilities and security properties among different PHY channels.

3.3.3 *Ease of Adoption*

In order to evaluate how feasible it is for a specific channel to be adopted in the context of SDP, we compare it with a set of available interfaces on widespread hardware such as smartphones. We make this particular choice because of the ubiquitous nature of smartphones, which are viably considered as a gateway for the personal IoT environment [184] as well as the pairing mediator for the IoT devices [349]. Specifically, we employ the hardware characteristics of the fifth generation of Nexus devices [124] as a reference to identify the common interfaces present on an average smartphone.

The presence of a physical channel on commodity hardware facilities its adoption.

3.3.4 *Survey of PHY Channels*

In the following, we survey PHY channels suitable for SDP by focusing on the previously described properties.

3.3.4.1 *Wi-Fi Channel*

Wi-Fi is a wireless communication technology based on a set of IEEE 802.11 standards, and it is used to connect devices within a wireless local area network. The most common Wi-Fi standards such as 802.11 a/b/g/n/ac operate in 2.4 and 5 GHz frequency bands. Other frequency bands, for example, around 60 GHz are also standardized (IEEE 802.11ad) but less frequently used. Due to different propagation characteristics at high frequencies, we discuss IEEE 802.11ad separately in Section 3.3.4.3.

The spectrum at 2.4 GHz and 5 GHz is used for Wi-Fi.

Different modes of operation are available for Wi-Fi: infrastructure, direct, and ad-hoc. The infrastructure mode is established with a centralized access point (AP), which handles all network traffic from connected stations. The latter two modes are formed in a peer-to-peer fashion directly by the devices. While Wi-Fi direct is intended to be applied in use cases where ad-hoc Wi-Fi was previously envisaged for use, its implementation is very different. The primary difference is that Wi-Fi direct internally uses the infrastructure mode, while ad-hoc Wi-Fi remains a separate mode. Because the specification of ad-hoc Wi-Fi has not been substantially updated since the 802.11b standard, more recent Wi-Fi security and performance improvements have not necessarily been incorporated into ad-hoc Wi-Fi implementations. This would be of limited concern, were it not for the continued use of ad-hoc Wi-Fi in certain applications, particularly those where multi-hop mesh networking is required [1, 112]. While some technologies now widely adapt Wi-Fi direct [126, 316], the pure ad-hoc mode is relatively rarely used despite the number of its advantages.

Wi-Fi has three modes of operation: infrastructure, direct, and ad-hoc.

Table 2: Summary of surveyed pairing schemes utilizing PHY channels.

PHY Channel	Channel Characteristics							Attacks	Common Interface	Pairing Scheme
	Description	Frequency Range	Typical Data Rates	Typical Coverage	Perceptibility	Penetration	Line-of-sight			
Wi-Fi (§3.3.4.1, pp. 43)	Wireless Radio Comm.	2.40 GHz–2.48 GHz, 5.03 GHz–5.83 GHz	1 Mbit/s–1 Gbit/s	30 m–250 m	○	High	○	Eavesdropping [4] MITM [4, 108] Jamming [21, 306] DoS [218, 306]	●	PBC [86] Integrity codes [42] TEP [119]
Bluetooth (§3.3.4.2, pp. 46)	Wireless Radio Comm.	2.40 GHz–2.48 GHz	1 Mbit/s–24 Mbit/s	1 m–100 m	○	High	○	Eavesdropping [74] MITM [74, 135] DoS [74]	●	Just Works [28]
mm-Waves (§3.3.4.3, pp. 47)	Wireless Radio Comm.	57 GHz–64 GHz	6.75 Gbit/s	10 m	○	Low	●	Eavesdropping [341]	○	PBC [86]
RFID (§3.3.4.4, pp. 47)	Wireless Radio Comm.	120 kHz–150 kHz, 13.56 MHz, 850 MHz–960 MHz, 2.45 MHz, 5.8 GHz, 3.1 GHz–10.6 GHz,	4 kbit/s–1.5 Mbit/s	10 m (passive) 100 m (active)	○ ○	Medium High	○ ○	Eavesdropping [183] Unauthor. access [63, 186] Relay [63, 103] DoS [183]	○	Noisy tags [46] Adopted-pet [5]
NFC (§3.3.4.5, pp. 49)	Wireless Radio Comm.	13.56 MHz	424 kbit/s	10 cm	○	Medium	○	Eavesdropping [409] Unauthor. access [186] Relay [104]	●	NFC [87] Out-of-band [252]
VLC (§3.3.4.6, pp. 50)	Wireless Visible Comm.	400 THz–800 THz	11 kbit/s–96 Mbit/s	10 m	●	Low	●	Eavesdropping [53]	○	KeyLED [287] Enlighten me! [113] Flashing displays [190]
Visual (§3.3.4.7, pp. 51)	Wireless Visual Comm.	400 THz–800 THz	12 Mbit/s, 324 kbit/s	10 m 20 cm	●	Low	●	Eavesdropping [402] Replay [275]	●	SBVLC [402]
IrDA (§3.3.4.8, pp. 52)	Wireless Infrared Comm.	334 THz–353 THz	2.4 kbit/s–1 Gbit/s	1 m	○	Low	●	Replay [14] Eavesdropping [53]	○	Talking to strangers [17]
Audio (§3.3.4.9, pp. 53)	Wireless Acoustic Comm.	20 Hz–20 kHz	20 bit/s, 4.7 kbit/s	~20 m, ~4 m	●	Medium	○	Eavesdropping [137] Relay [324]	●	Loud and clear [123] HAPADEP [333] Zero-power pairing [141]
Ultrasound (§3.3.4.10, pp. 55)	Wireless Acoustic Comm.	20 kHz–20 MHz	230 bit/s, 2 kbit/s	11 m 2 m	○	Low	○	Eavesdropping [235] Relay [56, 235]	◑	Ultrasonic ranging [237]
Haptic (§3.3.4.11, pp. 56)	Wireless Haptic Comm.	40 Hz–800 Hz	200 bit/s	physical contact	●	Low	○	Eavesdropping [137]	●	Vibrate-to-unlock [302] Shot [344] Vibreaker [7]
Sensing (§3.3.4.12, pp. 57)	Onboard Sensors	n/a	n/a	n/a	◑	n/a	◑	Relay [304, 324] Context-manipulation [324] Reproducible readings [344] GPS: Spoofing, jamming [361]	◑	Amigo [372] Good neighbor [40] Wanda [271] Audio pairing [310] Context-based pairing [243] MagPairing [172] TAG [378]

● = fulfills property; ◑ = partly fulfills property; ○ = does not fulfill property.

The data rates of Wi-Fi communication have increased significantly over the last decade and can exceed 1 Gbit/s [121]. Wi-Fi coverage varies from 30 to 250 meters [12, 18] for indoor and outdoor environments, respectively. The 5 GHz band has a smaller communication range due to a shorter wavelength and higher attenuation as compared to 2.4 GHz [248]. Wi-Fi communication is human-imperceptible and enables omnidirectional transmission with signals propagating through most non-metal objects such as walls, doors, and windows.

Wi-Fi coverage can be up to 250 meters.

Since Wi-Fi channels are inherently broadcast and have wide coverage, they are susceptible to a number of threats [4]. Adversaries may, for example, obtain unauthorized access to intercept transmitted information, inject and modify data in the air with surgical precision, reroute traffic for analysis with MITM attacks [108], or efficiently jam the network to cause a DoS [218, 306]. Such attacks have been shown to be feasible with low effort [21]. Moreover, due to the widespread use of Wi-Fi, identity tracking might threaten user privacy [250].

Numerous attacks have been demonstrated on Wi-Fi communication.

As Wi-Fi chipsets are ubiquitous and integrated in a wide range of devices starting from powerful laptops to resource-constrained sensors, the technology became a de-facto standard for communication of mobile devices. In the following, we describe various pairing schemes which utilize the Wi-Fi channel to accomplish pairing.

Wi-Fi is a ubiquitous technology in the IoT.

Push Button Configuration (PBC) have been introduced as part of standardized Wi-Fi Protected Setup (WPS) [86], which incorporates two other pairing schemes known as "Pin Entry" and "Near Field Communication". The pairing is initiated when a user presses a button on one device (enrollee), which starts searching for a PBC-enabled peer within its range to complete pairing. Once a button is pressed on the second device (registrar), an unauthenticated DH key exchange is performed via the Wi-Fi channel.

Despite the fact that the PBC pairing scheme is implemented on real devices and provides protection against passive eavesdroppers, it is inherently vulnerable to active adversaries who can mount MITM attacks.

Capkun et al. [42] propose *integrity codes (I-codes)*, a security mechanism that enables authentication and integrity protection of messages exchanged over a wireless radio channel. In order to achieve the stated purpose, I-codes rely on three components: unidirectional message coding, on-off keying communication, and the ability of the receiver to determine if the transmitter is within its communication range. The authors demonstrate that authentication through presence can be achieved if communicating devices are aware of each other's reception distance, and they are synchronized with respect to the start of transmission.

We review PBC, integrity codes, and TEP schemes based on Wi-Fi.

The security properties of I-codes are discussed in the presence of a powerful attacker who has full control over a wireless channel except for their inability to disable the whole communication, for example, remove the energy of a signal. Based on I-codes, a new version of the DH protocol is proposed, which the authors claim to be optimal in the sense of transmitted message length and the corresponding security level.

Gollakota et al. [119] suggest *tamper-evident pairing (TEP)*, a scheme that utilizes on-off coding to prevent MITM attacks on the wireless channels. Specifically, the authors introduce a primitive called tamper-evident announcement (TEA), which completely prevents active attackers from either changing the content of a transmitted message or hiding the fact that the message has been sent. To achieve the stated goal, the TEA mechanism introduces silence periods. Particularly, the payload of the TEA message is appended by a sequence of short equal-sized packets called slots in which the transmitter chooses to either send data (on-slot) or remain idle (off-slot).

The TEP scheme uses a bit sequence produced by on-off slots to encode the hash of the TEA payload. In this case, an adversary might tamper with the off-slots by transmitting a signal, while they cannot remove energy from the on-slots. Hence, the attackers that have no physical access to the pairing devices are prevented from tampering with the transmitted signal: they can neither suppress the communication between legitimate devices nor create a capture effect [380].

3.3.4.2 *Bluetooth Channel*

Bluetooth shares the unlicensed spectrum at 2.4 GHz with Wi-Fi, but it is designed for much shorter coverage of a few meters.

Bluetooth is a wireless communication technology which operates in the 2.4 GHz frequency band, and it is used to connect several devices in an ad-hoc manner, thus forming a personal area network [27]. The typical data rates for Bluetooth are 1–3 Mbit/s but can reach 24 Mbit/s [261]. Bluetooth coverage varies from 1 to 100 meters depending on the utilized antennas [74], and it can be used in both indoor and outdoor environments. The physical characteristics of Bluetooth communication are similar to those of 2.4 GHz Wi-Fi, therefore it cannot be sensed by humans, achieves relatively high penetration of solid objects, and does not require LoS for data transmission.

Attacks similar to Wi-Fi have been performed against Bluetooth.

Bluetooth communication is vulnerable to similar security issues as Wi-Fi, despite the fact that a Bluetooth channel is more difficult to access due to its shorter range. In addition, attacks to extend over the nominal communication range, obtain unauthorized data access, or fuzz protocol implementations to reveal vulnerabilities have been shown to be feasible [74] with low hardware requirements. Moreover, MITM attacks [135] as well as DoS [74] are as practical as in Wi-Fi.

Bluetooth is one of the most common communication technologies in smart devices.

Currently, low-power Bluetooth chipsets are pervasive and can be found in billions of devices. Further, we review a prominent pairing scheme that relies on the Bluetooth channel.

Secure Simple Pairing [28] proposed by the Bluetooth SIG is a de facto standard for pairing multiple personal devices. With a recent security enhancement [29], there are now four schemes available for Bluetooth pairing: “Just Works”, “Numeric comparison”, “Passkey Entry”, and “Out-of-band”. However, only the first scheme solely relies on the Bluetooth channel to achieve pairing, whereas others utilize HCI or other PHY channels, for example, near-field communication (NFC), to ensure authenticity.

Secure Simple Pairing is a standardized pairing scheme for Bluetooth devices.

The *Just Works* scheme is used to perform pairing with constrained devices, for example, a headset, which lack convenient input/output capabilities such as a keyboard

or display. Essentially, Just Works is based on an unauthenticated DH key exchange, which provides protection against passive eavesdroppers but is inherently vulnerable to active MITM attackers [28].

3.3.4.3 *mm-Waves Channel*

The unlicensed spectrum for mm-Wave communication is around 60 GHz.

mm-Wave wireless communications operate in a wide frequency band from 30 to 300 GHz. The lower part of the mm-Wave spectrum (30–50 GHz) is considered to be used in cellular and indoor environments with the coverage of up to 200 meters [279], although high-speed outdoor point-to-point links can work over longer distances [157]. At higher frequencies, the unlicensed spectrum around 60 GHz is being standardized (i.e., IEEE 802.11ad [163]), and it is deemed to be actively used for a great variety of indoor applications [88].

With mm-Waves, very high data rates are possible due the wide channel bandwidth available. For example, IEEE 802.11ad achieves transmission speed of up to 6.75 Gbit/s within the coverage area of up to 10 meters [67]. Due to high attenuation and absorption rates, mm-Waves at 60 GHz do not propagate through solid objects, for example, walls, and the LoS requirement is imposed on the mm-Wave communication [256]. Being part of the microwave spectrum, mm-Waves cannot be perceived by humans.

mm-Waves require line-of-sight communication and are easily blocked by walls.

The plausible properties of mm-Waves such as the short range, LoS transmission, and no wall penetration are claimed to provide highly secure operation [157]. However, it has been shown [341] that eavesdropping is possible on a 60 GHz channel through reflections caused by small-scale objects located within a transmission beam. Currently, 60 GHz chipsets can only be found in a few commercial products, for example, [80], but the number of supported devices will undoubtedly increase in the medium term. Next, we describe a pairing scheme which uses the mm-Wave channel.

Devices using mm-Waves appear on the market.

Despite being a relatively new technology, the 60 GHz communication has already adopted pairing schemes from the standardized WPS such as *PBC* [155]. Nevertheless, the PBC pairing is susceptible to MITM attacks as stated above. However, due to the short-range transmission with LoS, an adversary would have to be copresent, that is, in the same room, and interfere within a transmission beam in order to mount such an attack. These actions are much harder to perform stealthily without a benign user noticing them, which is not the case for the legacy Wi-Fi PBC.

60 GHz mm-Waves is part of the Wi-Fi standard adopting the PBC pairing scheme.

3.3.4.4 *Radio-frequency Identification Channel*

radio-frequency identification (RFID) is a wireless communication technology which is used for automatic identification in both indoor and outdoor environments. That is, an RFID system consists of tags (either active or passive), which store the identification information, and readers that query the tags in order to extract and verify that information [383]. The more ubiquitous passive tags have to harvest energy from nearby

RFID reader's interrogating radio waves, whereas the active tags have on-board power supply, for example, a battery.

RFID operates in several frequency bands [383]: Low Frequency (120–150 kHz), High Frequency(13.56 MHz), Ultra-High Frequency (860–960 MHz), Microwave (2.45 and 5.8 GHz), and Ultra-Wide Band (3.1–10.6 GHz). Its typical data rates vary from several to hundreds of kbit/s, depending on the utilized spectra [78]. The coverage reported for the RFID technology ranges from 10 to 100 meters for the passive and active tags, respectively [311, 383]. Regardless of the underlying frequency, RFID communication cannot be sensed by humans. However, the capability of RFID transmission to pass through solid objects depends on the used spectrum as well as the employed antenna, and it is higher for active RFID tags. For sending and receiving data with RFID, LoS is not required.

RFID operates in a range of frequency bands.

There are several security concerns regarding RFID communication. First, its wireless nature poses threats similar to Wi-Fi and Bluetooth, which are especially prominent for active tags operating over longer distances [183]. Second, passive tags are very constrained devices and can promiscuously respond to any reader's request [219] despite being short-range. It has been shown that RFID channels are vulnerable to eavesdropping, unauthorized access, relay, and DoS attacks [63, 103, 183, 186]. The relay attacks on contactless smart cards are especially severe, as they can easily circumvent the security of payment and access control systems [71, 103], and defending against such relay attacks is an active research area [152, 219, 278].

Constrained resources of RFID tags caused severe attacks on them such as relay.

RFID tags are presently ubiquitous and can be found in many applications such as logistics, tracking, and access control [360]. Nevertheless, most consumer devices, for example, smartphones, are not supplied with built-in RFID chips, instead they use the NFC technology. Several representative pairing schemes, which employ the RFID channel, are described below.

End-user devices are not equipped with RFID supporting NFC instead.

Castelluccia and Avoine [46] present a paring scheme called *Nosy tags* for secure key establishment over a wireless RFID channel between a passive tag and reader. Essentially, the pairing scheme relies on noise injection into a public communication channel, which makes the actual signal meaningless for an eavesdropping adversary, but allows the reader to efficiently restore the original message. This idea is implemented by introducing an extra RFID tag (i.e., a nosy tag), which belongs to the reader and shares a secret key with it.

We survey two schemes for pairing RFID tags with readers.

The proposed pairing scheme works as follows. When the reader queries a passive tag within its proximity, the nosy tag generates a sequence of random bits, which prevents the eavesdropper from differentiating between the original message sent by the queried tag and the one injected by the nosy tag. On the reader's side, the generated noise can be subtracted to recover the actual signal.

The authors provide three variants of their pairing scheme based on the nosy tags and analyzed its security against passive attackers. Nevertheless, the pairing scheme can still be circumvented by active adversaries.

Amariucai et al. [5] suggest *Adopted-pet*, an automatic time-based scheme for pairing a passive RFID tag with a reader without any human interaction or additional PHY channels, for example, NFC. The main idea is as follows. A tag can reassure that a particular reader is trusted only if it spends a sufficient amount of uninterrupted time within the proximity of this reader. Specifically, a tag has to be interrogated only by a single reader (i.e., uninterrupted property) for a time period during which the tag gradually transmits pieces of its secret password, which are accumulated by the reader in order to eventually restore the secret.

The authors implement their pairing scheme using a linear-feedback shift register and argue that it is robust against adversaries who can spend numerous interrupted time intervals in the proximity of a victim tag.

3.3.4.5 *Near-field Communication Channel*

NFC is a short-range communication technology to connect devices within a few inches from each other.

NFC is a wireless communication technology which is used to establish point-to-point communication between two devices brought to close proximity. NFC is an offshoot of RFID technology, thus NFC devices can similarly be active or passive [334]. NFC operates in 13.56 MHz frequency band and supports data rates of up to 424 kbit/s [104]. NFC has a very limited coverage of up to 10 cm [104]. Similarly to RFID, NFC communication cannot be perceived by humans, is able to penetrate solid object to a certain degree, and does not require LoS for data transmission.

Relay and range extension attacks are especially severe on NFC.

Initially, the security assumptions about NFC were based on its very short range, and hence severe difficulty for an attacker to access it. However, recently it has been shown that eavesdropping on NFC channels is possible at a distance of up to 240 cm [409]. In addition, unauthorized readings [186] pose a real threat, which can lead to practical relay attacks on the NFC communications [104]. With the advent of mobile NFC payments, the relay attacks on such systems have become a severe security threat [223, 231, 286], which has not yet been fully addressed [134, 242].

The increasing number of devices is being equipped with NFC.

NFC chips are widely deployed and can be found in numerous smartphones and other devices. We present two pairing schemes which utilize the NFC channel.

Near Field Communication [87] is a pairing scheme from the standardized WPS mentioned previously. Specifically, the NFC channel can be used to transmit a hardware generated password from one device that initiates pairing (i.e., enrollee) to another device (i.e., registrar) with which the pairing should be performed. Another pairing setting available with WPS NFC is to exchange hashes of public keys between the enrollee and registrar once they are brought to close proximity, that is, physical contact.

The NFC channel is used to transmit authentication information as part of standardized Wi-Fi and Bluetooth pairing.

The security assumptions of WPS NFC are based on the limited communication range provided by the NFC technology, which is much more difficult for an adversary to eavesdrop.

Out-of-band [252] is a pairing scheme provided by standardized Bluetooth Secure Simple Pairing. It works as follows. Once two devices have discovered each other via the Bluetooth channel, the NFC channel is used to exchange authentication information,

for example, Hash C, Randomizer R, or TK-value, between the devices in order to accomplish pairing.

The security arguments for the Out-of-band scheme rely on the restricted nature of the NFC communication, which cannot be easily accessed by the attacker.

3.3.4.6 *Visible Light Communication Channel*

VLC is perceptible by humans and cannot penetrate non-transparent objects.

visible light communication (VLC) is a wireless communication technology which carries information by modulating light in the visible spectrum that is used for illumination [11]. VLC operates in the 400–800 THz frequency band, and it is widely considered to be used for indoor short range communications [11]. The data rates that can be achieved by the existing standard (i.e., IEEE 802.15.7 [162]) vary from 11.67 kbit/s to 96 Mbit/s [276], although recent research demonstrates throughput of up to 20 Gbit/s [159]. Typically, VLC has the coverage of up to 10 meters [263], and it is perceived by humans via the sight sense. VLC transmission requires LoS and cannot penetrate non-transparent solid objects such as walls and doors.

Eavesdropping on VLC is shown feasible via reflections.

Therefore, VLC communication is concealed, to some extent, from an adversary who is not copresent. However, recently it has been shown that VLC can be efficiently eavesdropped by the attacker located outside of the room where communication happens [53]. Additionally, in [267] it is discussed that the integrity of a VLC channel can be affected by an adversary using a directional light source, for example, a laser.

VLC prototypes according to the standard have been built.

At present, there are no commercial devices, for example, smartphones, that support the standardized IEEE 802.15.7 VLC technology. However, fully functional prototypes [283] have been recently demonstrated, which makes it feasible that the VLC-enabled devices will appear on the market soon. In the following, we review representative pairing schemes that rely on the VLC channel.

Roman and Lopez [287] study the applicability of visible light as OoB in the context of wireless sensor networks (WSNs), where two previously unknown devices want to exchange sensitive information. The authors develop a scheme called *KeyLED* with which two constrained sensors can pair. Particularly, two devices located in close proximity utilize a LED-photosensor pair to set up a short distance communication channel (i.e., a few cm) and transmit their public keys using on-off keying.

The security of the proposed pairing scheme is discussed with respect to eavesdropping, injection, and DoS attacks. The authors claim that even though such threats are feasible, mounting them in practice is difficult, and a benign user who initiates communication between two sensors can be easily alerted in case of the attack.

We discuss three pairing schemes utilizing the VLC channel.

The similar line of work by Gauger et al. [113] investigates an ad-hoc key assignment for devices in WSN. The suggested *Enlighten me!* scheme is considered for two application scenarios: (1) initial key assignment as part of the WSN configuration and (2) dynamic key reassignment of already deployed sensors.

The proposed pairing scheme works as follows. There is a master device (i.e., key sender) which provides a set of sensors (i.e., key receivers) residing within its wireless

range with secret keys using an auxiliary light channel. During the key assignment procedure, the discovery of a receiving device, secret transmission, and verification are achieved with a light source-sensor channel using Manchester coding.

The authors implement two types of key senders: a sensor node lamp with a powerful LED and a smartphone with varying brightness level on a display. For both prototypes, they argue that eavesdropping the transmitted key is difficult to achieve in practice because the resulting VLC channel can be effectively concealed from an outside observer.

Kovačević et al. [190] propose *Flashing displays*, two multichannel deployment schemes for secure initialization of wireless sensors using only a multi-touch screen of a smartphone or tablet as the light source. Particularly, both schemes utilize two channels: wireless radio and VLC, where the former is considered as insecure, and the latter is used as OoB.

The first scheme relies on a visible light channel that is established between a display of a smartphone and a light sensor of a constrained device once it is put on top of the screen. In this setting, several constrained devices can simultaneously receive secret keys, which have to be verified later on over a wireless radio channel.

The second scheme is introduced in order to address a powerful adversary who can still eavesdrop on the VLC channel via electromagnetic emissions of a flashing display. Specifically, the developed mechanism incorporates both the VLC channel, for synchronization purposes, together with customized integrity codes [42]. The authors demonstrate that such a pairing scheme is secure against the attacker who can read the content of the flashing screen at any moment in time.

3.3.4.7 *Visual Channel*

The visual channel utilizes a display-camera pair for communication.

A visual channel enables wireless communication in the visible light spectrum (i.e., 400–800 THz) by utilizing currently abundant LCD-camera hardware. Such real-time transmission is shown feasible at 12 Mbit/s within a distance of 10 meters using large displays and high-speed digital cameras [268]. Another line of research [149, 374, 375] investigates visual communication that can be established with the LCD-camera found in commodity hardware such as smartphones. The results indicate that data transmission at 324 kbit/s is possible in the vicinity of 20 cm.

Main threats to the visual channel come from the ubiquity of shoulder surfing and surveillance.

Such visual channels, whose properties include short range of communications and interference-free operation, are claimed to provide secure transmission [255, 268]. However, the fact that an LCD-camera channel can be observed and easily interpreted by humans comes with a drawback. Specifically, eavesdropping either in the form of shoulder surfing or ubiquitous CCTV is shown to be a real threat [402], especially taking into account the continuous increase of display sizes [19, 338] and recent advances in CCTV [77]. Another security issue is related to the "liveness" of the captured video stream, which can lead to replay attacks [275].

Displays and cameras are common on smartphones.

At present, camera-display peripherals are ubiquitous on numerous devices such as smartphones. Further, we describe a pairing scheme based on the visual channel.

Zhang et al. [402] investigate secure bar-code communication for smartphones. They propose *SBVLC*, a novel approach for secure ad-hoc interactions, which can be established via a short-range LCD-camera channel on mobile devices.

We survey a pairing scheme based on bar-code communication.

The authors suggest a pairing scheme based on SBVLC that works as follows. To pair, two parties utilize a full duplex LCD-camera channel, which is realized as a sequence of QR-codes displayed on the screen of one device and captured by the camera of another device. Specifically, once two smartphones are brought to physical proximity, that is, within a few inches, they start to simultaneously exchange key material using the described visual channel. Afterwards, one of the devices randomly picks a universal hash function which is used to build a shared secret key from the material accumulated by both parties.

The security of the proposed approach is formally analyzed against the eavesdropping adversary by employing 2D and 3D geometric models. In addition, it is shown that proactive rotation of the devices during pairing can enhance security, since the attacker is forced to capture frames simultaneously from both displays to undermine the pairing scheme. Moreover, the authors show that the established key has enough entropy, that is, cannot be recovered, if the adversary misses at least one frame during the key exchange, which further improves security.

3.3.4.8 *Infrared Data Association Channel*

IrDA requires line-of-sight communication and suffers from interference caused by ambient light.

Infrared Data Association (IrDA) is a set of wireless communication technologies that uses the infrared radio spectrum 334–353 THz [257] for point-to-point data transmission. Since IrDA is susceptible to interference from ambient light sources [140], it is mostly considered for indoor applications. With the IrDA communication, high data rates of up to 1 Gbit/s [164] are possible. IrDA has the coverage of up to 1 meter, is human-imperceptible, and requires direct LoS.

Replay attacks have been shown feasible on IrDA.

IrDA is claimed [164] to provide secure data transmission due to its short range, directional operation (a 30° beamwidth), and the fact that infrared communication cannot traverse through solid objects such as walls and doors. However, such claims cause controversy because with the toolkits like *TV-B-Gone* [14] a variant of replay attacks can be mounted over relatively long distances [81]. In addition, the eavesdropping attack through reflections recently demonstrated on VLC [53] can be feasibly applied to the infrared channel since the two media have very similar physical properties.

Few end-user devices feature the IrDA channel.

Presently, many IrDA-enabled devices can be found in consumer electronics and household appliances, although such technology is obsolete on modern smartphones. Nevertheless, there is a growing number of personal devices supplied with IR-blasters [38], which indicates the restored interest to the infrared communication. One such pairing scheme, which makes use of the infrared channel, is described next.

Balfanz et al. [17] suggest a pairing scheme known as *Talking To Strangers*. The core idea behind is to combine demonstrative identification from the user perspective, for example, two physical devices with which a user interacts, with location-limited

channels that aim to provide data authenticity. The latter concept denotes exactly the OoB channel.

The IrDA channel has been proposed as the OoB in a pioneering work on pairing.

The basic pairing scheme proposed by the authors works as follows. First, two devices exchange commitments to their public keys, that is, hashes, over an IrDA communication channel, which serves as the OoB channel. Second, they transfer their corresponding public keys over a wireless radio channel. The wireless communication is verified against the initial commitment that has been transmitted over the infrared channel. In addition, several other schemes are developed upon the basic pairing approach, which deal with constrained devices and group pairing.

The authors discuss the security implications of the proposed pairing and point out that an adversary would have to actively intercept the OoB channel in order to undermine the pairing scheme.

3.3.4.9 *Audio Channel*

The audio channel is formed by a microphone-speaker pair.

Audio is a mechanical pressure wave caused by periodic vibrations within an audible frequency range of 20 Hz–20 kHz [288]. An audio channel, in this case, would be represented by a speaker-microphone pair, where the former generates a sound, and the latter records it.

Audio communication is human-perceptible, and it is affected by ambient factors such as noise.

The line of research [44, 148, 199, 225] investigates the throughput and coverage of the audio channel utilizing different modulation schemes. The results indicate that with inexpensive speakers and microphones found in commodity hardware such as laptops, data rates can go from 20 bit/s to 4.7 kbit/s over distances from 19.7 to 3.89 meters, respectively. Naturally, the audio channel can be perceived by humans via the hearing sense. The audio signal can, to a certain extent, pass through solid objects, for example, walls, although the penetration capability very much depends on the used frequency and environment, which determine the pass loss factors [160]. For transmission, the audio channel does not require LoS, however, the signal reception is largely affected by several aspects: (1) intensity of a sound source, (2) ambient noise, (3) acoustic environment, and (4) directionality and sensitivity of a microphone [327].

Eavesdropping and relay attacks are common on audio.

With respect to security, eavesdropping is shown to be easily achievable using off-the-shelf equipment even for specifically designed short-range sound waves [137]. Moreover, relay attacks are possible since audio streaming tools are highly available on mobile devices such as smartphones [324].

Many smart devices have microphones and speakers.

Currently, microphones and speakers are pervasive in many existing platforms, ranging from simple sensors to powerful laptops. We describe several pairing schemes which utilize the audio channel.

Goodrich et al. [123] suggest an approach to human-assisted authentication of previously unknown devices using the audio channel. The developed *Loud and clear* (LC) pairing scheme requires two devices to be equipped with speakers, or when one device does not have a speaker, it should be supplied with a display.

The LC pairing consists of two phases and works as follows. First, both devices exchange their public keys over a wireless radio channel such as Wi-Fi or Bluetooth. Second, the audio channel is used to transmit the hashes of public keys encoded as MadLib sentences, which can be verified by the user. Specifically, in the case of both devices having speakers, the user has to confirm the equality of the generated audio sequences. While for the speaker-display setting, the user needs to ensure that a sentence played by the first device is similar to the one displayed on the screen of the second device.

The authors analyze the security of the LC pairing scheme and conclude that MITM attacks can be easily detected if the user is diligent when comparing verification audio sequences.

We review three pairing schemes based on the audio channel.

Soriente et al. [333] propose a pairing scheme called *HAPADEP*, which relies on the audio channel to transmit both data and verification information between previously unknown devices.

The HAPADEP pairing scheme consists of two steps and works as follows. First, both devices exchange their public keys over the audio channel using the fast codec, which allows higher transmission speed but makes the signal meaningless for a user. During the second phase, two devices encode the hash of the exchanged cryptographic material using the slow codec and play back the sound, for example, a melody or a MadLib sentence that can be recognized and verified by the user. That is, if both audio sequences heard by the user match, then the pairing is considered to be successful.

The authors provide the implementation of the HAPADEP pairing and conduct a usability study, which reveals that the scheme is generally accepted by the users. Moreover, they discuss the resilience of the proposed pairing to MITM, impersonation, and DoS attacks. The HAPADEP scheme has been cryptographically extended in the unified pairing framework [234] to provide perfect forward secrecy (PFS), which further increases security against MITM attackers.

Halperin et al. [141] investigate security and privacy implications in implantable medical devices (IMDs). Specifically, they reveal that communication between the implant and the medical programmer, used for the collection of sensitive data and IMD reprogramming, happens without encryption or authentication. This opens the door for attacks such as eavesdropping, replay, and DoS. To mitigate the aforementioned threats, the authors propose zero-power pairing, which can be applied to batteryless constrained devices such as passive RFID tags.

The suggested pairing scheme works as follows. The programmer initiating pairing sends a RF signal to power the passive component of the IMD, which, in turn, generates a session key and broadcasts it as a modulated sound wave that is recorded by the programmer's microphone.

The authors reason about the security of the proposed pairing scheme based on two points. First, since the microphone is placed within a few centimeters of a patient's chest, it can easily receive the audio signal. However, it is very difficult to obtain the

same signal from farther distance without dedicated hardware equipment. Second, by utilizing the audio channel for the key exchange, the user is provided with audible and tactile feedback, which brings their attention to the fact of pairing.

3.3.4.10 *Ultrasound Channel*

Lower ultrasound spectrum can be produced/captured by commodity speakers/microphones.

Ultrasound refers to acoustic waves that lie within a frequency range above audible sound (i.e., 20 kHz–20 MHz) [161]. In this spectrum, frequencies higher than 250 kHz are strongly absorbed by the air, and thus are mostly used for medical imaging rather than data transmission [73]. As in the case of audio, the ultrasound channel is formed by an ultrasonic speaker-microphone pair, which are based on the piezoelectric effect to produce high frequency waves [356].

Ultrasound communication is more concealed than audio due to higher absorption and reflection.

Similar to audio, the throughput and transmission range of the ultrasound channel are evaluated in prior work [44, 148, 199, 225, 258]. Specifically, data rates can vary from 230 bit/s to 2 kbit/s at the corresponding distances of 11 and 2 meters. Contrary to audio, the ultrasound communication cannot be sensed by humans. When propagating through air, the ultrasound signal is subject to high reflection and absorption rates caused by solid objects, for example, walls, which makes the ultrasound communication limited to a single room [235]. For data transmission with ultrasound, LoS is not required.

The ultrasound channel is susceptible to eavesdropping and relay attacks.

As for security implications, it is shown that a copresent adversary can eavesdrop and manipulate the ultrasound channel, although the attacker's capabilities largely depend on their position relative to communicating parties [235]. Moreover, relay attacks can be mounted on the ultrasound channel when it is used for a distance estimation as part of the authentication procedure [56, 235].

At the moment, few end-user devices are supplied with dedicated ultrasonic chips. However, the lower part of the ultrasound spectrum can be generated and recorded by non-specialized hardware present on existing smartphones [385] and laptops [258]. A pairing scheme that employs the ultrasound channel is presented below.

Mayrhofer et al. [237] study how secure spontaneous interactions can be established with spatial references. They develop a pairing scheme that utilizes the ultrasound channel for initial device discovery and then implicitly for authenticity verification.

We discuss a prominent pairing scheme relying on ultrasound.

The proposed pairing scheme works as follows. First, two devices become aware of each other and learn their corresponding distances and relative positions by employing the ultrasound sensing. Second, both devices perform an unauthenticated DH key exchange over a wireless radio channel. Third, devices authenticate each other by sending a nonce encrypted with a shared key over a wireless radio channel and transmitting the plain nonce over the ultrasound channel using the interlock protocol [284]. Specifically, the nonce value is split into pieces, and each part is transmitted as a delayed ultrasound pulse to another device, which results in a longer distance than the previously obtained spacial reference. Thus, the receiving device can subtract the initially learned distance

from the received measurement to acquire a part of the nonce and gradually learn the full nonce.

The authors discuss the security of the pairing scheme with respect to eavesdropping, relay, and MITM attacks. They claim that the suggested pairing can mitigate and detect those adversaries even if they have access to both radio and ultrasound channels, assuming a benign user is attentive.

3.3.4.11 *Haptic Channel*

The haptic communication is based on a vibrator motor-accelerometer pair, and it is perceptible by humans.

A haptic channel is formed by low frequency waves within a range of 40–800 Hz that cause tactile sensations [30]. For data transmission, such a channel can be represented by a vibrator motor-accelerometer pair, where the former generates a set of pulses captured by the latter. Recently, it is demonstrated that with advanced modulation and coding schemes, data rates of up to 200 bit/s can be achieved over the haptic channel using off-the-shelf hardware [292]. Obviously, haptic communication requires direct physical contact between the sender and receiver. By its nature, haptic transmission does not propagate well in the air and cannot pass through solid objects, for example, walls. Also, haptic communication is human-perceptible, and it does not require LoS.

Despite the restricted nature of the haptic channel, it has been shown that eavesdropping is possible through acoustic side channels [137].

Vibrator motors and accelerometers are built in many smartphones.

The haptic channel, realized with a vibration motor and accelerometer, can presently be found in numerous end-user devices such as smartphones. Several pairing schemes that use the haptic channel are presented below.

Saxena et al. [302] propose a pairing scheme called *Vibrate-to-unlock*, which is used to establish a shared secret between an RFID tag and smartphone that belong to the same user.

The suggested pairing scheme works as follows. Initially, a smartphone selects a secret PIN (14-bits) and transmits it as an on-off coded sequence of vibrations. An RFID tag that is brought to contact with the vibrating phone records the data with its accelerometer, decodes the PIN, and stores it. After the enrolling step, two devices share a common secret. Later on, when the RFID tag is powered, once in the vicinity of the reader, it can only be unlocked if a user authenticates by proving the possession of the pre-shared PIN with their phone, that is, similarly as described above.

We survey three pairing schemes utilizing the haptic channel.

The authors claim that their scheme has a corresponding security level of the 4-digit PIN prompted at the ATM with 3 attempts. Moreover, they argue that the suggested pairing can mitigate such attacks as user tracking, impersonation, and ghost-and-leech.

Studer et al. [344] investigate security implications of the Bump exchange protocol [37] and reveal that it is vulnerable to MITM attacks. Hence, the authors present a new scheme known as *Shot* to pair two smartphones in a user-friendly manner.

The Shot scheme uses a server, which is considered as an insecure channel between two devices to be paired, and works as follows. The first device (i.e., endorser) hashes its public key and transmits the truncated version of the hash (i.e., 80-bits) to another device

(i.e., verifier) as a sequence of vibrations. This message serves as a pre-authenticator, and it is used by the verifier to bootstrap communication with the server, that is, as a session identifier. By utilizing the server, two devices exchange their identities and public keys. With such information at hand, the verifier can compute the hash of the endorser's public key and compare it with the previously sent pre-authenticator. Once checked, the verifier informs the endorser about the success or failure of the pairing via a binary vibration.

The authors analyze the security of Shot pairing and claim that in can withstand all types of active attacks on the insecure channel given an adversary cannot inject messages into the haptic channel.

Anand and Saxena [7] investigate how the previously proposed Vibrate-to-Unlock scheme [302] can be secured against acoustic side channels [137]. Their approach is to actively inject noise in order to cloak the acoustic leakage emanating from the vibrations. The enhanced pairing scheme called *Vibreaker* utilizes a built-in speaker of a smartphone to generate a masking signal, which makes the acoustic side channel indistinguishable for an eavesdropping adversary. Specifically, the authors explore white noise and vibration noise, for example, the prerecorded representation of audio leakage, as feasible candidates for masking. In Vibreaker, the pairing procedure of Vibrate-to-Unlock is complemented by an extra step when a transmitter injects the masking signal during the PIN transmission through vibrations. The results indicate that both types of noise can efficiently conceal the acoustic side channel even if the attacker applies filtering techniques.

3.3.4.12 *Sensing Channel*

The proliferation of sensors in smart devices has enabled the sensing channel.

A sensing channel is used to obtain information about the ambient environment as well as determine device's location, position, and orientation. Recently, the use of built-in sensors has been proposed for ZIP, where colocated devices (e.g., inside the same room) sense their ambient environment (i.e., context) utilizing it as a source of common entropy to establish a shared secret key [243, 310, 372, 378]. A similar line of work develops zero-interaction authentication (ZIA) schemes, where devices apply context sensing to ensure each other's proximity to protect against relay attacks on wireless channels [63, 136, 323, 364].

We summarize categories of sensor modalities found in the literature.

- Radio (e.g., Wi-Fi, Bluetooth).
- Audio.
- Motion and position (e.g., accelerometer, gyroscope, and magnetometer).
- Location (e.g., GPS).
- Physical (e.g., temperature, pressure, luminosity, and humidity).

Radio and audio are used to obtain information about wireless radio and acoustic channels described above. That is, with Wi-Fi and Bluetooth antennas, signals from

APs and peer devices can be received, while ambient audio can be captured with a microphone. In the case of radio and audio, the sensing range cannot be delimited precisely because it very much depends on the receiving antenna or a microphone, transmitting power, and the channel quality.

Motion and position is measured by a set of sensors that allow a device to detect movement as well as determine its relative position and orientation [125]. Readings from an accelerometer, gyroscope, and magnetometer are easily affected by user actions and the ambient environment, which makes the measurements obtained by similar sensors within some distance highly uncorrelated [172].

ZIP schemes rely on the sensing channel.

Location sensing is represented by the GPS system, which provides the worldwide outdoor positioning within the accuracy of several meters [180]. The GPS technology utilizes several frequency bands such as 1575.42 MHz and 1227.60 MHz for transmission. The data rates available with GPS can go up to 50 bit/s. The GPS communication requires direct LoS since the signals cannot easily pass through non-transparent solid objects such as walls and doors.

Physical sensing is used to capture information about the surrounding environment such as temperature, pressure, and luminosity. Typically, physical characteristics do not vary too much within close proximity, but they differ significantly for various locations, for example, indoor vs. outdoor or neighboring rooms.

Despite the recency of the sensing channel a number of attacks have been reported on it.

Previously, it has been claimed [136] that tampering with the ambient environment is a hard task in which an adversary is unlikely to succeed. However, a more recent study [324] reveals that it is feasible to manipulate readings of different sensors such as radio, audio, and physical in a controlled way using off-the-shelf hardware. Regarding motion and position modalities, it is demonstrated that accelerometer readings can be reproduced with sufficient precision [344]. This vulnerability stems from the limited accuracy of built-in sensors, and it can be further increased if the attacker manages to observe specific user actions, for example, shaking or bumping. As for the GPS location sensor, it has been shown susceptible to attacks such as spoofing and jamming [361].

Currently, many devices are equipped with sensing capabilities, with smartphones having several different ones, although various physical sensors are still not widely deployed. In the following, we review a number of ZIP schemes utilizing the sensing channel.

Varshavsky et al. [372] propose a pairing scheme named *Amigo* to authenticate colocated devices without explicit user involvement. Specifically, they suggest utilizing a common radio profile, which is location- and time-specific, as the indicator of physical proximity.

The pairing scheme works as follows. First, two devices, brought to close proximity, perform an unauthenticated DH key exchange over a wireless radio channel (e.g., Wi-Fi). Second, both devices start monitoring the ambient radio environment for a short period of time and construct a signature containing identifiers and signal strength of the packets received during the snapshot. Finally, two devices exchange their signatures

over a secure channel using a commitment scheme in order to verify if the received and local measurements match.

The authors analyze the security of their pairing scheme and report that Amigo is resilient to attacks such as eavesdropping, MITM, and impersonation.

Cai et al. [40] investigate how to establish secure communication between previously unknown devices without any shared secrets and OoB channels. They propose a pairing scheme called *Good neighbor*, which uses received signal strength (RSS) between multiple antennas of the same device (i.e., receiver) to differentiate if another device (i.e., sender) is nearby or not. Specifically, if the sender is in close proximity to the receiver, the RSS values measured by the two receiver's antennas will be substantially different, which is not the case when the (malicious) sender is far away, irrespective of its transmitting power.

The suggested pairing scheme relies on the correlation between the RSS and physical proximity and works as follows. First, the sending device initiates pairing by requesting a public key of the receiving device once brought close to its first antenna. Second, the sender generates a session key, which is encrypted with the receiver's public key, and starts to repeatedly transmit the session key to the receiver. Meanwhile, the sender needs to be moved to the second antenna of the receiver. Finally, the receiver calculates the ratio of the RSS values obtained from two antennas and checks if the number of consecutive measurements are above a predefined threshold.

The authors evaluate their pairing scheme with respect to a powerful adversary who can eavesdrop on the wireless channel, arbitrarily adjust the transmitting power of her devices, and gain knowledge about the location of receiver's antennas. The results indicate that the proposed pairing can successfully mitigate such an attacker.

We review three pairing schemes based on radio: Amigo, Good neighbor, and Wanda.

Pierson et al. [271] propose *Wanda*, a pairing scheme built upon Good neighbor pairing [40] to securely introduce mobile devices. Conceptually, the "Wand" is realized as a portable hardware device equipped with two antennas located half wavelength apart. Similarly to Good neighbor, the scheme uses signal strength to determine if the Wand and a target device are nearby (i.e., detect primitive). However, Wanda expands upon Good neighbor by utilizing wireless signal reciprocity to securely transmit data between the Wand and the target device via the in-band channel (i.e., impart primitive).

The proposed pairing scheme works as follows. First, a user enables the target device, for example, pressing a button, which starts broadcasting beacon packets and points the Wand to it. Using the RSS ratio of the received beacons from two antennas, the Wand determines if the target device is in close proximity. Second, to send a message, the Wand encodes it as a binary string and transfers one bit at a time. Particularly, a packet transmitted using the closest antenna is considered as "1", and "0" if it is sent from the farthest antenna. To decode the message, the target device calculates the average RSS from all received packets and checks if the RSS of a specific packet is above or below the average, that is, either "1" or "0". Finally, the Wand sends the hash of the transmitted message which can be verified by the target device.

The authors evaluate the security of the Wanda scheme against eavesdropping and malicious packet injection. Their findings demonstrate that the proposed pairing can withstand both types of attacks.

The first "classic" ZIP scheme proposed by Schürmann and Sigg uses ambient audio.

Schürmann and Sigg [310] study how a secure communication channel can be established between two devices in an ad-hoc manner by utilizing ambient audio. Specifically, they propose the first "classic" ZIP scheme that uses audio fingerprints obtained by two devices from the shared ambient environment to derive a common secret key without exchanging any information about the captured audio context.

The suggested pairing scheme works as follows. First, two devices synchronize their clocks by running an NTP-based protocol. Second, two devices start simultaneously recording the ambient audio with their local microphones. The obtained audio fingerprints are very similar but not identical due to noise and sampling effects. Finally, error correction codes (i.e., Reed-Solomon) are applied to obtain identical codewords, which are mapped to the unique secret key.

The authors analyze the security of the proposed pairing scheme with respect to an adversary who is not in the same context but can eavesdrop as well as mount MITM, DoS, and audio amplification attacks. The experimental results confirm that such threats can be successfully mitigated.

Miettinen et al. coined the term "zero-interaction pairing".

Miettinen et al. [243] propose context-based ZIP for IoT devices, which can happen without any user involvement. Specifically, the notion of sustained copresence is employed, meaning that two devices will sense the same context over a substantial period of time if they are in close proximity.

The proposed pairing scheme works as follows. First, two devices derive a shared secret key using an unauthenticated DH key exchange. Second, both devices continuously monitor ambient audio and luminosity in order to obtain contextual fingerprints over time. Using these readings, two devices can iteratively evolve the initial secret key and obtain a new secret key each time two fingerprints are sufficiently similar. Finally, after a number of successful key evolution steps, two devices can authenticate each other and use the evolved secret key for secure communication.

The authors discuss the security of the suggested pairing scheme with regard to an adversary being inside and outside the same context as well as examining context replay attacks. Their findings indicate that the proposed pairing can withstand both types of adversaries and mitigate the replay attacks.

Jin et al. [172] propose a pairing scheme called *MagPairing*, which requires minimum user interaction, and thus yields high usability. Particularly, they leverage magnetometer readings of two smartphones brought to close proximity in order to establish pairing.

The suggested pairing scheme works as follows. First, two devices are tapped, which triggers an authenticated DH key exchange during which both devices measure magnetic fields with their sensors. Second, two devices securely exchange their magnetometer readings via the interlock protocol [284]. Finally, both devices can authenticate each other by comparing if the received and local measurements match.

For security analysis, the authors consider attacks such as eavesdropping, MITM, replay, and reflection. The results show that MagPairing resists the above threats even if a powerful active adversary is within a few centimeters from the pairing devices.

Wang et al. [378] suggest a pairing scheme known as *Touch-and-guard (TAG)* for associating a wearable and another nearby device by utilizing resonant properties of a human hand. Specifically, a shared secret is obtained from a hand touch using vibration motors and accelerometers.

ZIP schemes have been prototyped on smartphones and wearables.

The proposed pairing scheme works as follows. First, a user initiates pairing by touching a target device, for example, a payment terminal, with the hand on which a wristband is worn. Second, the target device generates vibrations, which excite both the device itself and the hand. At this point, both the wearable and the target device record vibrations with their accelerometers. Finally, both devices process their accelerometer data separately without exchanging it in order to extract reciprocal information to eventually generate a shared secret.

The security of the TAG scheme is empirically evaluated against an eavesdropper acting via acoustic side channels. It is shown that the proposed pairing can withstand such attackers even if they are located in proximity. However, the authors admit that the TAG scheme can still be susceptible to advanced visual eavesdroppers who utilize high-speed cameras.

3.3.5 *Discussion*

We present main takeaways from our survey of physical channels.

The results of our survey on PHY channels reveal interesting details. First, the literature makes it clear that there are no known confidential channels despite considerable efforts having been invested in pursuit of this goal. Second, at the time of writing, the most promising communication channels, in terms of security, are not present in the majority of devices. Third, the use of sensors to obtain a shared context has recently been proposed as a new approach for SDP, however it is not without challenges. We expand on these points in the following.

3.3.5.1 *There Are No Confidential Channels*

Confidentiality cannot be guaranteed by any of the PHY channels surveyed, even though this appears to be an explicit or implicit assumption in a number of pairing schemes. As presented in Table 2, all PHY channels that we study are vulnerable to eavesdropping attacks, and those attacks have been successfully mounted in the past. Hence, none of these channels provides a secure transmission medium on its own.

The problem here is twofold. First, the probability of "off-the-shelf" eavesdropping, that is, performed without specialized equipment, has increased tremendously since numerous smart devices nowadays are equipped with various peripherals, for example, cameras and microphones, and the sensing capabilities of commodity hardware con-

No communication channel can provide confidential data transmission only based on its physical properties.

tinue to grow [25]. Second, as shown by Halevi and Saxena [137], side channels pose a real threat because they can completely circumvent the security of the pairing scheme. Specifically, they demonstrate that three pairing schemes (i.e., Zero-power pairing [141], Vibrate-to-unlock [302], and BEDA [332]), which assume confidentiality of the OoB channel, can be successfully attacked by exploiting acoustic side channels.

The importance of side channels as a vital security issue has been recognized by the research community and addressed in recent communication systems [292] and pairing schemes [7, 378]. Nevertheless, new sources of sensitive information leakage are being continuously discovered [293], which raises a fundamental question whether it is feasible to identify and tackle all hidden channels in modern systems. Therefore, we argue that confidentiality of the PHY channel is very hard to achieve and guarantee in practice. Correspondingly, this property should be treated with a great deal of attention when a PHY channel is considered as a candidate for the OoB channel.

Regardless of this state of affairs, where it is questionable that confidential channels are possible in practice, pairing schemes continue to be proposed that rely on secrecy of data transmission, for example, [376].

3.3.5.2 *Most Potentially Secure Channels Missing From Current Commodity Hardware*

The physical characteristics of various PHY channels provide different security properties. We have identified an important trade-off between security and ease of adoption. That is, a number of newer communication channels such as mm-Waves and VLC can offer improved security, however they are not yet ubiquitous.

VLC and mm-Waves channels have plausible security properties, however, they are not yet prevalent on off-the-shelf devices.

In particular, mm-Waves and VLC possess valuable security characteristics. Their short-range communication, LoS requirements, and low penetration rates make them ideal for deployment and use as OoB channels in the IoT domain. Research is, however, still ongoing to improve on both mm-Waves [256] and VLC [263] communications. A further advantage is that both technologies can be efficiently implemented on constrained devices. For example, the antennas required for mm-Wave transmission are very small, and the VLC building blocks such as diodes and photosensors are inexpensive. Hence, these technologies are worth considering for SDP.

The challenge lies in the maturing of these newer channels such that they become widely available on commodity hardware. This requires the action on both researchers and vendors.

3.3.5.3 *Using Environment Sensing*

A different approach to pairing is ZIP, namely utilizing the sensing channel to obtain the shared context, which can be used either as an indicator of physical proximity or as an entropy source to derive a shared secret key. The use of sensor data enables scalable pairing, which is crucial in a distributed and diverse environment such as the IoT. It

also reduces or eliminates the user effort, resulting in more usable and less error-prone pairing.

For example, the use of physical environment sensing [323] and GPS data [220], as already explored in the domain of ZIA, can provide a suitable base to increase security in device pairing as well. The key insights provided by ZIA, both those that are security-enhancing such as fusing multiple sensor modalities [364] as well as those that are adversarial such as context-manipulation threats [324], should also be taken into account in ZIP.

Sensing enables ZIP and ZIA, however, the security guarantees of these approaches are unclear.

Channel state information (CSI) in radio communications provides a different use for sensing. Reciprocal radio channels, meaning that the same antenna is used for transmitting and receiving, lead to correlated channel observations at both sides of a communication link. This correlation of channel observations allows the transmitter and receiver to obtain a common fingerprint of the radio environment that can, in turn, be used to mitigate various types of attacks, including MITM and relay attacks.

Pairing schemes such as Amigo [372], Good neighbor [40], and Wanda [271] as well as [179, 213, 232, 265, 319, 376, 377, 389] rely on such information to ensure that both pairing devices communicate over the same channel. However, the robustness of such channel fingerprinting schemes against spoofing is still an open question under investigation. For example, Zafer et al. [399] have demonstrated an active CSI spoofing attack. Hence, pairing schemes leveraging the radio channel must account for manipulated and forged channel states.

Environment and channel sensing can provide an additional layer of verification. However, it still suffers from similar security limitations as PHY channels. Therefore, the sensing channel cannot, by itself, guarantee that no one is intermediating communications between the pairing devices, for example, through a MITM attack.

In this section, we have investigated and discussed PHY channels along with the SDP schemes utilizing them. In the next section, we review HCI channels and the corresponding pairing schemes.

3.4 HUMAN-COMPUTER INTERACTION CHANNELS

Modern information and communication technologies have become an indispensable part of the human society. The way people live, work, and interact with each other and the environment has changed significantly with the advent of smart devices, social networking, and cloud-based services. Various research and technologies have utilized HCI to provide security in a wide range of applications such e-commerce, home automation, and social networking [65, 168, 299]. With the upcoming IoT, the importance of developing socially compatible security tools based on HCI is becoming more evident [64, 66]. However, relying on human interactions to achieve security often introduces vulnerabilities to the system. Bruce Schneier [303] emphasizes the relevance of the human factor in the system as follows: "... security is only as good as it's weakest

Human-computer interaction has been leveraged for security purposes in different applications.

link, and people are the weakest link in the chain." Hence, the security of the system where a user is involved depends not only on the technical aspects of the system but also on how people understand and use it in addition to the system's capability to mitigate threats and issues introduced by the users themselves [24, 208, 299]. From the pairing perspective, a user also plays an important role with regard to security. Traditionally, the security of pairing schemes has involved an aspect of human supervision, which can take the form of perception, for example, image comparison [269], decision-making, for instance, pressing a button [332], and other interactive techniques, for example, drawing a pattern [315].

We identify a number of HCI channels used for pairing, studying their security and usability properties.

We start our discussion by identifying three points which are the base for rigorous HCI investigation. In particular, we specify several types of *HCI channels*, which have been used in SDP and denote two sets of properties, namely *security properties* and *usability properties* that we study. Afterwards, we review existing pairing schemes that rely on various HCI channels to exhibit the trade-offs between security and usability. Finally, we discuss the most significant insights and implications that have been identified in our survey on HCI channels.

3.4.1 *HCI Channels in Device Pairing*

Four types of HCI channels are investigated in our survey.

Recently, numerous devices with rich input/output capabilities and considerable processing power have become widely available, which has significantly improved the quality of HCI [168]. Correspondingly, many pairing schemes proposed up-to-date rely on some form of user involvement. Chong et al. [50] survey existing pairing schemes by considering user actions required to establish a secure channel between two devices. We refine their findings to obtain the fine-grained categories of user interaction that have been used in SDP.

Specifically, we define three HCI channels that fully satisfy our definition given in Section 3.2.2: *Data relay*, *Data comparison*, and *Data generation*. In addition, we consider *Device handling* which, while not a conventional HCI channel, represents a more passive form of user interaction that is often (implicitly) present in device pairing.

3.4.1.1 *Data Relay*

A channel where a user is prompted to transfer data generated by one pairing device onto another pairing device.

3.4.1.2 *Data Comparison*

A channel where a user is required to compare and analyze data produced by two pairing devices, for example, to verify the correctness or consistency of the information.

3.4.1.3 *Data Generation*

A channel where a user provides common input to both pairing devices simultaneously, for example, shaking, drawing, or first imposes (secret) input on one device and then provides it again on a second device.

3.4.1.4 *Device Handling*

A form of user interaction where a human actor is required to bring pairing devices in proximity, make physical contact, align them, or take similar action.

3.4.2 *Security Properties*

To identify relevant security properties of HCI channels, we use knowledge from adjacent research fields.

The security properties of the pairing schemes based on HCI channels are quite different from the ones purely relying on PHY channels. First, users are the unavoidable source of errors [191, 208] and their behavior as well as attitude towards security sensitive tasks vary significantly [24]. Second, user interaction is subject to observation by both an internal participant, who is curious and an external adversary, who is malicious. To compile the list of representative security properties, we combine issues that have been raised in the pairing community with respect to human factors [110, 301] and complement them with the implications found in the authentication domain [32].

3.4.2.1 *Inattentive User*

Defines if a pairing scheme has certain tolerance to mistakes and errors introduced by the user. In particular, a pairing mechanism that does not verify user input for errors or provide corresponding feedback can be circumvented by the attacker who can impersonate a legitimate device.

3.4.2.2 *Rushing Behavior*

Specifies if a pairing scheme accounts for rushing users who are willing to skip certain steps of the pairing procedure or accept specific conditions without verification in order to speed up pairing.

3.4.2.3 *Consent Tampering*

Determines if a pairing scheme is resilient to consent tampering by a dishonest user. That is, if the user can accept pairing even if the data exchanged between two devices mismatch or conversely, reject pairing even though both devices successfully establish a connection.

3.4.2.4 *User Observation*

Defines if a pairing scheme is resistant to an adversary who can observe user actions during the pairing process. In other words, the attacker does not benefit from learning user interactions, including (secret) data exchanged on the HCI channel and cannot compromise pairing with such information at hand.

3.4.2.5 *Forward Secrecy*

Determines how resilient a pairing scheme from a cryptographic perspective to an eavesdropper who can leverage user observation and the compromise of the long-term keys. That is, if the underlying cryptographic protocol used in the pairing scheme mitigates brute-force offline attacks aided by (secret) data observed on the HCI channel and restricts an adversary to a one-off (online) guessing game. We evaluate this property under the assumption that DH keys used by the underlying cryptographic protocols are ephemeral.

3.4.2.6 *Honest-but-curious*

Specifies if a pairing scheme is susceptible to an honest-but-curious adversary who legitimately participates in the pairing process but tries to learn or infer more information about another pairing party.

3.4.3 *Usability Properties*

We complement usability properties of HCI channels found by prior work on pairing with findings from user-device interaction and authentication domains.

As mentioned previously, the usability of pairing schemes has been a subject in several studies, and a number of works investigate how usability can be enhanced in the case of device pairing [156, 178, 193]. However, many works apply mostly quantitative metrics to evaluate usability such as completion time and error rate [196], which are implementation-dependent. In addition, subjective characteristics such as personal preferences vary with context, as has been previously demonstrated [166]. Thus, there is a lack of a common baseline approach which would allow usability evaluation of pairing schemes more qualitatively and coherently. We aim to remedy this situation by presenting a set of usability properties, which we derive by studying the usability implications in general human-device interaction [133] as well as authentication techniques [32], and projecting the findings onto the pairing domain.

3.4.3.1 *Effortless Initialization*

Defines minimal user effort during the discovery phase of the pairing process. For example, a user is not required to provide any additional information such as a number of participants or preconfigure devices prior to pairing.

3.4.3.2 *No Secret Relay*

Does not prompt users to transfer any (secret) information from one pairing device to another, or if it is required, the length of the relayed data should be minimal.

3.4.3.3 *Automatic Secret Generation*

Specifies that the data used for authentication, for example, cryptographic keys, is generated by pairing devices without requiring any user input or assistance such as shaking or drawing.

3.4.3.4 *Automatic Consistency Check*

Determines user effort, necessary for verifying that information exchanged between pairing devices is similar.

3.4.3.5 *Environmental Insensitivity*

Defines the applicability of the pairing scheme with respect to the ambient environment. For example, a pairing scheme may lead to high error rates or even fail if the environment is too noisy, crowded, or has poor illumination.

3.4.3.6 *Explicit User Feedback*

Specifies if a pairing scheme provides meaningful feedback to the user during and upon the completion of the pairing process. For example, two pairing devices can indicate success by making an appropriate sound and provide explanatory actionable feedback if pairing fails.

3.4.3.7 *Familiarity*

Determines if the user actions imposed by the pairing scheme correspond to the daily user experience [166, 299]. That is, if a pairing scheme relies on well-established interaction patterns, for example, smartphone usage, and requires no extra training for an average user in order to be adopted.

3.4.4 *Survey of HCI Channels*

In this section, we review representative pairing schemes which rely on HCI by focusing on the properties given above.

3.4.4.1 *Manual Authentication (MANA)*

Gehrmann et. al [115] present several *MAN*ual *A*uthentication (MANA) schemes for authenticating DH public keys. They assume that devices have at least one input and/or output interface, for example, a display and/or a keypad. From the user perspective, a human operator plays a crucial role in pairing. Three variants of the MANA scheme were proposed which work as follows:

- MANA I: One device has a display and a simple input interface, for example, a button, while another device has a keypad and a simple output interface, for instance, an LCD panel. The first device computes a random key and a checksum value and displays this data. The user reads the checksum value and the random key from the screen of the first device and inputs this information into the second device. Then, the second device computes the checksum value using the provided random key and compares the two checksums. The outcome of the comparison is indicated as an accept or reject message to the user. Finally, the user enters the result back into the first device.

We survey three types of MANA scheme accommodating different hardware requirements.

- MANA II: Both devices have a display but neither of them a keypad, although they are equipped with a simple input interface, for example, a button. Similar to MANA I, the first device computes the random key and the checksum and displays two values. In addition, the first device sends the random key to the second device over an insecure channel, for example, wireless radio. Afterwards, the second device computes the checksum value and outputs it together with the key. By comparing values displayed by both devices, a user has to either accept the connection if they are equal or reject it otherwise.
- MANA III: Both devices are assumed to have a keypad. The user enters a short random bit-string **R** into both devices. Then, each device generates a random message authentication code (MAC) key and calculates a MAC value over **R** concatenated with a device identifier and the DH-public keys. Afterwards, both devices exchange their corresponding MAC values via a wireless radio channel. Only upon receiving the MAC value from the pairing peer, each device reveals its MAC key. Finally, both devices verify the received MAC values and indicate the result to the user who is required to compare and confirm it. A simpler variant exists in the case one of the devices has only a display, that is, no means of input.

The authors argue that MANA-schemes are robust against MITM attacks given the user diligence in verifying calculated hash values.

Table 3: Summary of surveyed pairing schemes utilizing HCI channels.

Pairing Scheme	HCI Channel				Security Properties						Usability Properties						
	Data Relay	Data Comparison	Data Generation	Device Handling	Inattentive User	Rushing Behavior	Consent Tampering	User Observation	Forward Secrecy	Honest-but-curious	Effortless Initialization	No Secret Relay	Auto. Secret Generation	Auto. Consistency Check	Environment Insensitivity	Explicit User Feedback	Familiarity
MANA I [115]	●	–	–	–	○	○	○	○	○	○	○	○	◑	◑	●	◑	●
MANA II [115]	–	●	–	–	○	○	○	○	○	○	○	●	●	○	●	○	●
MANA III [115]	–	◑	●	–	○	○	○	○	●	○	○	●	○	◑	●	●	●
AP authentication [291]	–	●	–	–	○	○	○	○	●	○	●	●	●	○	○	○	◑
Shake them up! [47]	–	–	–	●	●	○	●	○	○	○	○	◑	●	●	●	○	○
ShaVE [236]	–	–	●	●	●	●	●	◑	●	●	●	◑	◑	●	●	●	◑
ShaCK [236]	–	–	●	●	●	◑	●	◑	○	●	●	◑	○	●	●	●	◑
SAPHE [132]	–	–	●	●	●	●	●	◑	○	●	●	◑	○	●	●	●	◑
Ultrasound authentication [185]	–	●	–	●	○	○	○	◑	○	○	○	●	●	○	○	○	◑
Beep-Blink [272]	–	●	–	–	○	○	○	◑	●	○	●	●	●	○	○	◑	◑
Blink-Blink [272]	–	●	–	–	○	○	○	◑	●	○	◑	●	●	○	◑	○	◑
RhythmLink [209]	–	–	●	–	◑	◑	●	○	○	○	○	○	○	●	◑	●	○
Seeing-is-believing [240]	●	–	–	●	◑	◑	◑	◑	◑	◑	◑	○	●	●	◑	●	●
Visible laser light [238]	–	–	–	●	●	○	○	○	●	○	○	●	●	◑	○	●	○
VIC authentication [300]	–	●	–	●	○	○	○	◑	◑	◑	○	●	●	○	◑	●	●
BEDA (B2B) [332]	–	–	●	–	●	◑	●	○	●	○	●	●	○	●	●	○	○
BEDA (D2B, SV2B, LV2B) [332]	●	–	–	–	●	●	●	○	●	○	●	○	●	●	◑	○	○
Playful security [110]	●	–	–	–	●	●	●	●	–	○	●	○	●	●	●	●	◑
Safeslinger [84]	–	●	◑	–	●	●	●	●	○	◑	○	●	●	◑	●	◑	◑
Synchronized drawing [315]	–	–	●	●	◑	◑	◑	○	●	●	●	●	○	●	●	○	◑
Proximity authentication [202]	–	–	●	●	◑	●	●	◑	●	◑	●	●	○	●	●	○	◑
Checksum gestures [3]	●	–	–	◑	●	●	●	◑	●	◑	●	○	●	●	●	◑	◑

● = fulfills property; ◑ = partly fulfills property; ○ = does not fulfill property; – = n/a.

3.4.4.2 *Access Point Authentication*

Roth et. al [291] suggest a pairing scheme to protect the connection between an AP and a client device against evil twin attacks.

SASs have been used to encode some form of user input in pairing schemes.

The proposed pairing scheme uses short authentication strings (SASs) for key establishment and consists of two phases. In the setup phase, both devices exchange their public keys and a nonce value over an insecure wireless channel. During the authentication phase, a user is required to compare a certain number of color sequences (minimum two) in order to verify that pairing was performed with the intended AP. Specifically, each sequence is comprised of two colors and represents a SAS. Both devices display the sequence of colors, that is, one color at a time, and the user has to verify their equality by pressing a button and proceeding to the next sequence. The number of sequences shown depends on the desired level of security, and eventually, the user is prompted to either accept or reject pairing.

The authors discuss the security of the proposed pairing scheme and conclude that it can withstand evil twin attacks.

3.4.4.3 *Shake Them Up!*

Castelluccia et al. [47] propose a pairing scheme for CPU-constrained devices, for example, sensors, that do not have enough computational power to perform public key cryptography.

The proposed pairing scheme utilizes the anonymous broadcast channel and works as follows. In order to derive a shared secret key, two devices are held together and shaken either by a single user or by two users in close proximity. Meanwhile, both devices broadcast empty packets over an insecure wireless channel. The anonymous broadcast implies that each device sends a packet by setting its own identifier or the identifier of the pairing peer as the source of the message. In this case, an adversary can read the transmitted packets but cannot distinguish the source. In contrast, each pairing device knows if it has sent a particular message or not, which is interpreted by the device as a secret bit 1 or 0, and the shared key can be obtained by observing a predefined number of packets. The shaking is done to thwart signal strength analysis by an attacker to identify the actual sender.

The authors analyze the security of their pairing scheme against an adversary who can read all packets but cannot distinguish the source of the packet and report that it is resilient against MITM and DoS attacks. However, Rasmussen et. al [280] show the vulnerability of this scheme by using radio fingerprinting to identify the sender.

3.4.4.4 *Shake Well Before Use*

Mayrhofer et al. [236] suggest a pairing approach which utilizes accelerometer data generated from distinct movement patterns. Specifically, they propose two schemes to

securely pair devices where a user is required to hold them together and then shake simultaneously.

The first scheme (ShaVE) uses the DH key exchange to derive a shared key over an insecure wireless channel followed by the exchange of accelerometer readings via the interlock protocol [284] to verify the authenticity of pairing devices.

The second scheme (ShaCK) relies on the data captured by the accelerometer to derive a shared secret key. In particular, two devices hash their synchronized feature vectors obtained from the sensor readings and accumulate them until the entropy is sufficient to produce the shared secret key.

The authors discuss the security of the proposed pairing with regard to an active adversary and conclude that both schemes can withstand MITM attacks. However, they concede that the ShaCK variant does not provide forward secrecy, and it is vulnerable to offline guessing attacks.

3.4.4.5 *SAPHE*

Groza and Mayrhofer [132] propose a pairing scheme based on shaking, which improves upon the previous works, for example, ShaCK [236], by devising a more lightweight approach to securely exchange low-entropy vectors obtained from accelerometer data.

The schemes reviewed in sections 3.4.4.3–3.4.4.5 enable pairing by asking a user to shake devices simultaneously.

The suggested pairing scheme employs a hashed heuristic tree and works as follows. First, the commitments between two devices are exchanged in the form of hashes of randomly generated values. Second, the accelerometer data produced by shaking two devices together is recorded and used to obtain a unique secret key on each device. The unique secret keys are extracted by comparing the accelerometer readings to the threshold values obtained from the initial commitments by means of the Euclidian distance. The key extraction algorithm relies on a hashed heuristic tree, which is essentially a search tree, where the accelerometer readings are first sorted in a descending order with respect to the distance from the threshold values, and then bit-by-bit hashing is applied to retrieve the unique secret key. Third, both devices exchange challenges which are nonces encrypted with the individual secret keys, and each device proves the possession of the peer's key by verifying the challenge.

The authors analyze the security of the proposed pairing scheme and claim that their approach provides better resilience to MITM attackers, who try to guess the low-entropy vectors obtained from accelerometer data. However, the authors concede that further research is required to evaluate the resilience of the SAPHE scheme against the adversaries who can observe user interaction.

3.4.4.6 *Ultrasound Authentication*

Kindberg et. al [185] present a pairing scheme which utilizes ultrasound to physically validate two devices and establish a secure channel between them.

The proposed pairing scheme consists of two phases and works as follows. In the locate phase, a user selects a target device to communicate with and makes sure that their personal device (i.e., client) is in LoS with the target. Then, the client sends a message to locate the target, which replies with its designated identifier, for example, network address, over RF and ultrasound channels. The client receives those messages, matches the identifier, and is able to calculate the approximate distance to the target device, which is displayed to a user for verification. During the associate phase, the user points the client device to the target and initiates pairing. The target device replies with the RF message containing its public key together with a random number and simultaneously emits the ultrasound message with the same random number. Upon receipt, the client checks if random numbers from RF and ultrasound messages match and asks the user to confirm the relative position of the target device. Finally, the client encrypts a session key with the target's public key and sends it along with a random number back to the target.

Some pairing schemes require a user to locate and point one pairing device towards another.

The authors argue that the proposed pairing scheme is robust against various spoofing and replay attacks given the adversary is unable to counterfeit ultrasound messages.

3.4.4.7 *Synchronized Audio-visual Patterns*

Prasad and Saxena [272] present two pairing schemes suitable for devices with only basic interfaces such as a pair of LEDs and/or speakers. Specifically, both schemes rely on SASs transmitted by two devices in the form of synchronized audiovisual patterns, for example, blinking LEDs, which have to be compared by a user for equality.

In the first scheme (blink-blink), two devices encode their SASs as sequences of blinking LEDs, and the user is required to compare these sequences and determine if they are synchronous on both devices, for example, green or red LEDs.

Comparing audio or visual patterns is a common form of user interaction leveraged for pairing.

In the second scheme (beep-blink), one device transmits its SAS as a sequence of blinking LEDs while another device encodes the SAS as a series of beeping sounds and silence periods. The user has to verify if these two patterns match such as the LED light corresponds to the sound.

The authors analyze the security of the proposed pairing with regard to a MITM adversary and conclude that both schemes can withstand such attacks, yet the security depends on user diligence when comparing two audiovisual sequences.

3.4.4.8 *RhythmLink*

Lin et. al [209] propose a pairing scheme based on rhythm tapping.

Initially, a user inputs a song rhythm several times on their personal device, for example, a smartphone, to provide some training data and eventually obtain a tapped password, referred to as a tapword. Afterwards, this generated tapword is stored on the user device and used further for pairing.

To pair with a target device, the user inputs the same tapped rhythm into it. Therefore, the target device can compute a tapword and compare it with the pattern stored on the user device by means of the Euclidean distance. The protocol uses elliptic curve cryptography to calculate the Euclidean distance between the tapwords without either device revealing its tapword. To generate a session key, a password authenticated key exchange is used in order to avoid MITM attacks. A device encrypts its model information with this session key and sends the encrypted data to the other device, which decrypts this information and computes the Euclidean distance. Afterwards, both distances are compared. If the distances match, the devices accept pairing.

Prompting a user input in the form of a rhythm-based password allows pairing devices equipped only with buttons.

3.4.4.9 *Seeing-is-believing (SiB)*

McCune et. al [240] propose a pairing scheme based on taking a snapshot of a two-dimensional barcode displayed on the screen of one device by the camera of another device. The two-dimensional barcodes are generated by the devices automatically without any human effort. A user is required to configure the camera and take the snapshot of the 2-D barcode.

To perform pairing, one device sends its public key to another device over an insecure channel, for example, Wi-Fi, and displays a two-dimensional barcode. This barcode represents a visual encoding of the public key sent over the insecure channel. The second device, equipped with the camera, takes a snapshot of the barcode and runs a barcode recognition algorithm in order to process the image and extract the public key. Afterwards, this device compares the data obtained from the barcode with the data received over the insecure channel. If they match, the second device can trust the first device. The barcode-scanning procedure has to be executed by both devices for bidirectional authentication.

The popularity of smartphones enables pairing schemes based on taking a picture of a displayed bar- or QR-code.

The security assumption made by this pairing scheme is that mounting active attacks is difficult without being detected. The authors further analyze the security of their pairing scheme against passive attacks and propose additionally using the DH session key exchange protocol to protect against brute-force attacks.

3.4.4.10 *Visible Laser Light*

Mayrhofer et. al [238] develop a pairing scheme based on visible laser light for personal mobile devices equipped with a laser diode. These personal devices interact with another remote device, which is able to detect the laser light.

The proposed pairing scheme works as follows. First, a user presses a button and turns on the laser on their personal device. This causes the device to begin continuously transmitting messages. When the remote device detects these messages, it generates a response and broadcasts it over a wireless radio channel. Second, both devices start a key agreement protocol, and the target turns on a LED to identify itself. Third, if the LED is activated on the target device expected by the user, they press a second

button triggering an autonomous phase. During this phase, the derived secret key is verified by sending a series of cryptographic challenges via the wireless radio channel and requiring that the responses to the challenges to be transmitted by the laser.

The authors evaluate their pairing scheme in the face of an active adversary attempting to mount a MITM attack. They report that the attack would only succeed if the adversary can compromise the integrity and confidentiality of the laser and wireless radio channels simultaneously.

3.4.4.11 *Visual Authentication Based on Integrity Checking (VIC)*

Saxena et. al [300] improve the SiB pairing scheme by providing mutual authentication between devices to be paired using only a unidirectional visual channel, that is, requiring that only one of the two devices has a camera instead of both.

The proposed pairing scheme employs short authenticated integrity checksums for key agreement and works as follows. First, each pairing device exchanges its public data: a public key and a random bit string over an insecure channel. Second, each device calculates a checksum, in practice a cryptographic hash-function, over this public data, that is, both public keys and random bit strings. Third, one of the devices sends its results to the other device using the visual channel for comparison, that is, the second device uses its camera to read the 2-D barcode displayed by the first device. Fourth, the second device compares the hash transmitted over a display-camera channel by the first device with the locally computed value. If the two values match, the second device accepts the connection and displays a confirmation message to the user. Finally, the first device prompts the user to indicate if the second device accepted the connection or not.

The authors discuss the security of their pairing scheme, indicating that it is resilient to MITM attacks only if the hash function used in the scheme is collision-resistant.

3.4.4.12 *BEDA*

Soriente et. al [332] explore how to pair devices with very limited interface capabilities such as a single button. They propose a pairing scheme which first performs a DH key agreement and then executes the procedure to authenticate the DH public keys.

User interaction has been utilized for pairing devices without common hardware interfaces.

The suggested pairing scheme consists of two phases and works as follows. In the first phase, a short 21-bit secret is distributed between the devices with user assistance. Depending on the available hardware interfaces, this initial secret can either be obtained via the user input provided to both devices (i.e., Button-to-Button) or by relaying the data generated by one device to another device (i.e., Display-, Short Vibration-, Long Vibration-to-Button). In the second phase, the authenticity of the exchanged public key is incrementally verified in a 21-round procedure by using the initial secret.

The security of the proposed pairing depends on the confidentiality of the channel. The authors discuss that their pairing scheme is secure against MITM attacks only if the data exchanged between the devices cannot be eavesdropped. The BEDA scheme

has been cryptographically extended in the unified pairing framework [234] to provide PFS, which further increases the security against MITM attackers.

3.4.4.13 *Playful Security*

Gallego et. al [110] propose a pairing scheme based on the memory game Simon. The suggested scheme uses SASs computed by each device individually, and a user is required to transmit these strings from one device to another device.

The proposed pairing scheme works as follows. One device displays several audio-visual patterns, and the user relays these patterns to another device equipped with the input interface. The first pattern consists of a single color and tone that encodes the first two bits of the SAS. For the next round, two bits will be concatenated to the first pattern. This data forms a new pattern that needs to be similarly transmitted by the user. This iterative process continues until a sufficient number of bits have been successfully exchanged between two devices. If an error occurs in a round, a new pattern will be concatenated with the previous patterns that are exchanged successfully. To avoid synchronization issues, the first device has two buttons. If an error occurs, the user selects previous button to repeat the exchange of the SASs between the devices.

The authors argue that the proposed pairing scheme is robust to human errors, and therefore can mitigate MITM attacks caused by such errors.

3.4.4.14 *Safeslinger*

Farb et. al [84] present a pairing scheme for data exchange with smartphones. That is, users upon a physical encounter can initiate the exchange of their public keys as well as selected contact information and communicate securely afterwards. The SafeSlinger scheme is built upon two cryptographic mechanisms, namely multi-value commitments and group DH key agreement. The pairing scheme requires active user interaction, which includes entering the number of participating devices, selecting the data to be exchanged, and finally, comparing a 3-word phrase which has to be commonly chosen by all users.

Involving a user in the pairing process as part of a game or collective task has been shown feasible.

The authors analyze the security of their pairing scheme and argue that SafeSlinger mitigates attacks such as MITM, group-in-the-middle, impersonation, and sybil attacks by involving the user in the security chain and accounting for user misbehavior.

3.4.4.15 *Synchronized Drawing*

Sethi et. al [315] present a pairing scheme based on physical proximity and commitment-based cryptographic primitives.

The proposed pairing scheme consists of four phases and works as follows. In the first phase, two devices attempt to establish a shared secret using DH or a similar protocol over an insecure channel. In the second phase, fuzzy secrets are extracted from

user input produced by simultaneously drawing the same pattern with two fingers of the same hand, for example, a thumb and index finger, on two touchscreens or surfaces of two devices to be paired. In the third phase, each device sends an unencrypted commitment message to another device which contains a hash of: the (1) device's identifier, (2) fuzzy secret derived from the drawing, (3) random number, and (4) DH-shared key. In the fourth phase, each device encrypts its random number and fuzzy secret obtained in the third phase using the shared secret calculated in the first phase.

By carefully ensuring that both devices complete the third phase before entering the fourth phase, the authors argue that MITM attacks can be prevented.

3.4.4.16 *Proximity Authentication*

Li et. al [202] present a pairing scheme which uses proximity to perform mutual authentication between two devices without using NFC chips.

The common user input such as drawing or gesture has been used to pair mobile smart devices.

The suggested pairing scheme works as follows. First, a user draws a zigzag pattern simultaneously on both devices to be paired using two fingers of the same hand. Second, each device individually derives a set of common features obtained from the drawing. Third, the private set intersection approach [169] is applied to the feature vectors of both devices in order to generate a shared secret key.

The authors discuss the security implications of their pairing scheme and claim that it is secure against dictionary and MITM attacks.

3.4.4.17 *Checksum Gestures*

Ahmed et. al [3] propose a pairing scheme based on SASs, where a continuous gesture is required for encoding authentication information.

The suggested pairing scheme works as follows. First, a user and target devices execute a key exchange protocol based on SASs to obtain a checksum string (at least 20 bits) stored on both devices. Second, the user's device transforms this checksum string into a motion pattern, which is displayed to the user, who is required to reproduce this motion pattern as a continuous gesture on the target device. Third, the input gesture is captured and processed by the target device, which then compares the obtained data with the motion pattern derived locally from the shared checksum string. If both match, the unidirectional communication channel is authenticated between the user and target devices. The security of the proposed pairing scheme is based on the feasibility of gesture recognition technologies, particularly in maintaining sufficiently low false positive and false negative error rates.

The authors analyze the security of their pairing scheme based on the probability of interpreting a false input of an attacker as a correct gesture and report that the probability of a success relay attack is under 5.5%.

3.4.5 *Discussion*

The results of our HCI study are summarized in Table 3 from which we identify and discuss four key points that have important security and usability implications for SDP. First, we identify an important trade-off that exists between passive and active HCI channels. Second, the significance of usability properties, including the provision of explicit user feedback and insensitivity of HCI input to environmental conditions, is considered. Third, security issues resulting from various forms of intentional and unintentional as well as benevolent and malicious user misbehavior are explored. Finally, the vital problem of observation threats for HCI channels is presented, that is, the situation where an attacker can observe and exploit human interaction.

We discuss key findings from our analysis of HCI channels.

3.4.5.1 *Trade-offs Between Passive and Active HCI Channels*

The *handling* channel yields the best results in terms of usability because it requires the minimum amount of user effort. However, such pairing schemes do not give a user fine-grained control over the pairing process and provide less assurance that the pairing is established with the intended device. In contrast, data *relay*, *comparison*, and *generation* require more user involvement but provide better control and assurance of pairing. Yet, these types of interaction are susceptible to user misbehavior and errors, which makes it necessary for users to adequately understand the impact of their actions. For example, if the generation channel is involved, it is not sufficient to only incorporate common user experience. It is additionally required that the user is alerted if the generated secret lacks sufficient entropy for its intended use so that the user can take appropriate action.

Passive user interaction can be used for seamless pairing in non-critical applications.

Hence, we identify an important trade-off between different HCI channels. While passive user interaction can be viably used for pairing in situations where no sensitive information, such as financial or personal data, is involved. Active user participation should be used for more critical applications, for example, bank transactions, where user awareness can be leveraged to increase security in device pairing.

3.4.5.2 *Usability Properties*

Two usability properties which are crucial to augment both usability and security in pairing are providing explicit user feedback and ensuring insensitivity to environmental conditions.

Explicit user feedback and robustness to environmental noise improve usability of pairing.

First, the importance of *explicit user feedback* has been outlined previously [113, 166], yet only a few pairing schemes provide it in a meaningful way. However, the user feedback can not only mitigate input errors and present the evidence of pairing devices, for example, that pairing with the intended device is successful, but also assist a human operator with security advice. For instance, if the user generates data to produce a secret, the pairing mechanism can notify the user if the provided input has sufficient entropy for the intended application or not.

Second, *environment insensitivity* is also vital for maximizing user experience. That is, a pairing scheme should work for the intended use case, irrespective of the ambient conditions that might be reasonably expected to occur. Section 3.5 examines a range of specific use cases, exploring this topic further. The key point is that these two factors interact, for example, a pairing scheme that requires audio comparison and confirmation from the user should not be expected to be used in public scenarios.

3.4.5.3 *Security Issues*

The prior work emphasizes the security issues in pairing stemming from unintentional or deliberate user misbehavior [196, 301, 369]. Interestingly, only two pairing schemes (i.e., Playful Security [110] and Safeslinger [84]) account for such properties as *inattentive user*, *rushing behavior*, and *consent tampering* by design. Table 3 clearly indicates that human interaction by itself does not bring any security benefit if it does not consider threats posed by the user behavior, for example, MANA [115]. Additionally, in our analysis we introduce an *honest-but-curious* participant who tries to obtain more information about the pairing party. The motivation for this stems from a number of application classes that we consider. Since social and public pairing are in scope (cf. Section 3.5), it cannot be assumed that all pairing parties are benign and collaborative. For example, social engineering can be used to infer extra information about another user, or if the sensing channel is involved, another device or participant can leverage this sensor data to violate user privacy. Moreover, *human observation* has not been well addressed in the pairing literature. However, as we show under observation threats, the situation is dire, and this point must be taken into account if the pairing scheme relies on human interaction.

Most existing pairing schemes fail to leverage user interaction to improve security.

3.4.5.4 *Observation Threats*

Regarding observation threats, we focus on security implications in authentication techniques, as the adversary similarly tries to circumvent security by examining user interaction. The examples of malicious observation include, but are not limited to, shoulder surfing, audio or video analysis of the keyboard utilization, and voice recognition. Specifically, Halevi and Saxena [138] show that keyboard acoustic emanations can be successfully used to retrieve (even random) passwords prompted with different typing styles. Similarly, Davis et al. [68] propose a method to extract audio data from the high-speed video analysis in order to perform acoustic eavesdropping without having a microphone. More sophisticated attacks [15] exploit reflection from the objects to reconstruct any confidential data displayed on the screen of a device. Yue et al. [398] apply computer vision techniques to show that it is possible with 95% probability to reconstruct user input on the touchscreen of a mobile device using a low resolution video of user interaction. Recent attacks against voice verification [249] demonstrate that voice impersonation is achievable with the success rate of 90% using only a limited

The advances of surveillance aided by computer vision and ubiquity of sensing make observation of user interaction the major threat to security of HCI channels.

number of victim's voice samples. Overall, observational threats are increasingly easy to achieve, thus their risk should be taken into account when designing SDP schemes.

In this section, we have investigated and discussed HCI channels along with the SDP schemes utilizing them. In the next section, we review the application classes and classify the surveyed SDP schemes accordingly.

3.5 APPLICATION CLASSES

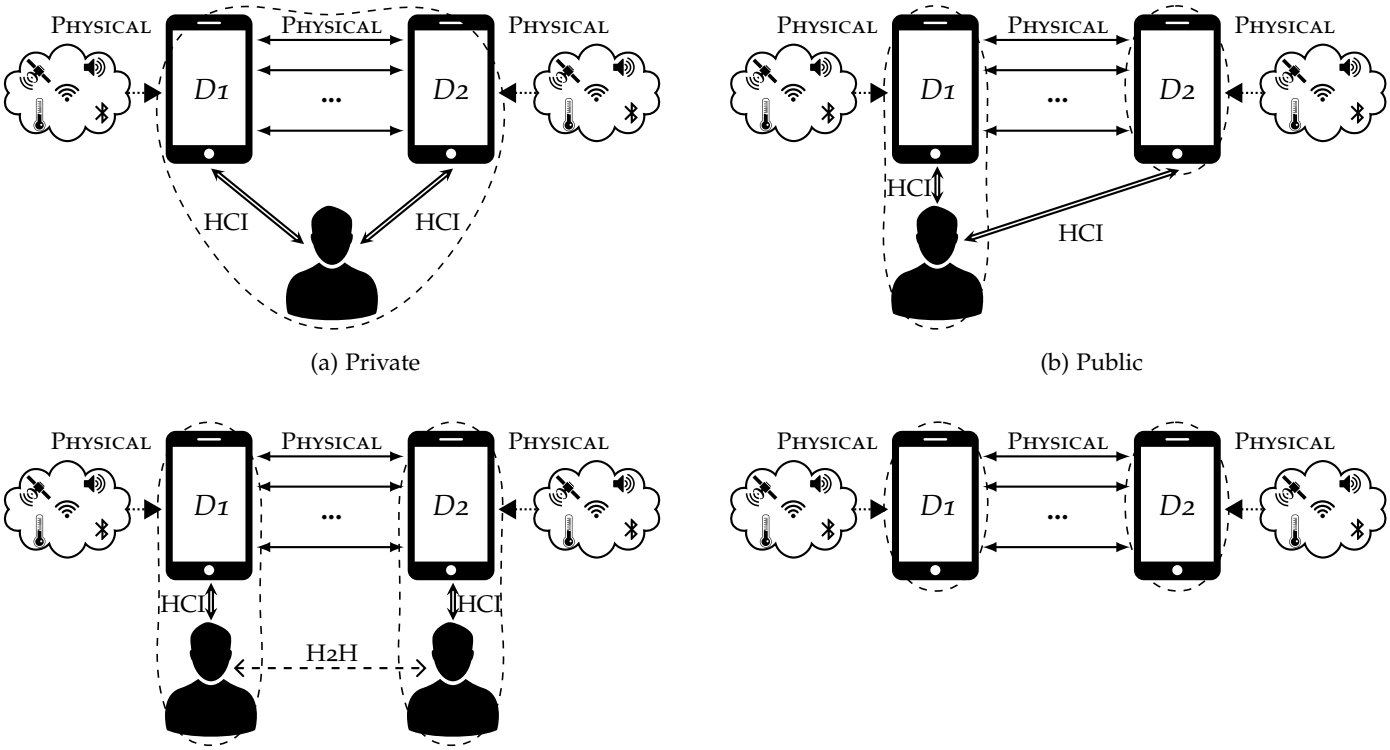
In this section, we explore and analyze the four application classes introduced in Section 3.2. First, we describe each application class with respect to typical interactions as well as its security and usability insights. Second, we categorize the pairing schemes covered in Section 3.3 and Section 3.4 with regard to their application classes and discuss the most interesting results of this classification. Finally, we highlight important open issues in SDP that have been identified in our study of application classes.

We survey pairing schemes with respect to application classes.

3.5.1 *Overview of Application Classes*

An application class covers a set of similar SDP use cases each of which involves a similar degree of involvement and level of user control over the pairing process. We recall the four application classes introduced earlier: (1) private, (2) public, (3) social, and (4) unattended. The private class corresponds to a "classic pairing" case where a single user either owns or directly controls two devices that ought to be paired. The public class is related to a single user possessing one device where the user performs the pairing with some third party infrastructure, for example, a payment terminal over which she has no control. The social class incorporates two users who would like to securely pair their corresponding devices. The unattended class covers the use case of ZIP, where two colocated devices (e.g., inside the same room), which may belong to the same or different ownership domain such as a person or organization, pair without user involvement. Figure 9 depicts the four application classes instantiated from our system model to provide a better understanding of the typical interactions for each application class.

We distinguish four application classes: private, public, social, and unattended.



(a) Private

(b) Public

(c) Social

(d) Unattended

Figure 9: The four application classes. Each application class consists of two devices to be paired, each from a distinct security domain, except for the private application class (a). The boundaries of security domains are marked by dashed lines.

The ownership of the devices being paired plays a critical role in SDP, necessitating its explicit consideration when describing application classes. We recall that a *security domain* is the set of devices, data, policies, and intentions that a single party controls. That is, a security domain refers to the limit of enforcement of security policy by a particular owner or controller of one or more devices. These security domains are especially significant when more than one exists, as it allows for security requirements of pairing devices to be differentially achieved or undermined either by the pairing process or subsequent actions of one of the pairing parties.

The application class defines the level of involvement and control that a user has over the pairing process.

For example, consider Figure 9a and Figure 9c. In Figure 9a, a single user controls all devices, and thus a single security domain exists. Therefore, following a successful pairing procedure, there are only two possibilities: either the policy requirements of the single security domain are met or not. In contrast, for Figure 9c, there are two users, each controlling a separate device D1 and D2, respectively. In this case, if the security policy requirements of each user differ, it may be possible that the security policy of one user is satisfied but not for the other. Similarly, one of the users may later reveal information that, without violating their own security policy, may violate that of the other. That is, the presence of the second security policy allows for a more complex set of outcomes as compared to if there were only a single security domain.

In the following, we expand on the four application classes under consideration.

3.5.1.1 *Private*

Figure 9a depicts the well-known private class which applies when a single user either owns or controls both devices. A good example of this scenario is pairing smart devices that belong to the same person. In such a setting, a rich set of HCI interactions are possible since a user can freely communicate with and handle their portable devices in many ways. The physical interactions between the devices as well as with the ambient environment are user-enabled, and they are only limited by the availability of hardware interfaces on the devices.

In the private class, the focus can be shifted towards usability of pairing.

From a security perspective, private pairing is often performed in a rather restricted environment, for example, home premises or a workplace, where such threats as external observation and communication interception are reduced. The private class consists of a single security domain, that is, all devices are subject to the same security policy requirements because they are controlled by a single party. In this context, the focus of the user tends towards usability due to the combination of reduced perceived threats and the relative frequency of pairing that may occur, especially given the increasing numbers of devices that people own. Hence, usability must be preserved and emphasized even in the face of numerous devices to be paired with one another. Despite the lower perception of risk, it remains important to maintain security.

3.5.1.2 *Public*

The public class, shown in Figure 9b, corresponds to the case where a single user possesses one device but has no control over another device to be paired with. For example, the user wants to pair their personal device, for instance, a smartphone, with a third party infrastructure such as a public AP, printer, or payment terminal.

In the public class, users are willing to trade some usability for security due to higher perceived risk.

In terms of HCI interactions, a human operator has fewer options as compared to the private class because the public infrastructure typically has only a few common user interfaces and cannot be moved, shaken, or handled in a convenient way. Similarly, physical interfaces used for communication between the devices as well as with the ambient environment are restricted and typically cannot be invoked by the user.

From a security perspective, the public class implies a more hostile environment, that is, public places, as compared to the private class. Thus, user actions during the pairing procedure are subject to external observation, which can come in the form of shoulder surfing or ubiquitous CCTV. Additionally, an attacker can stealthily install rouge devices in the public premises to interfere or hijack the pairing process.

The public class incorporates two distinct security domains, namely the user with their device and the infrastructure, which opens a door to a number of threats outlined in the following discussion. In comparison with the private class, users are likely to have an increased perception of security risks in such public scenarios. Therefore, users may reasonably accept some shift in the balance away from usability in order to improve security. However, care must be taken not to reduce usability to the point where users' tolerance is exhausted.

3.5.1.3 *Social*

The social class, illustrated in Figure 9c, represents a case where two different users would like to perform pairing between their personal devices [370]. Pairing two smartphones that belong to different people is a good example of such a scenario. It is obvious from the given example that the social class implies two distinct security domains, that is, two users with their devices. The presence of multiple security domains can result in complicated security outcomes, as previously described.

In the social class, the usability of pairing becomes important so as the possible privacy hurdles.

The reality of users' concerns regarding these complications can be observed, for example, through users' reluctance to hand their personal devices over to others. Explicit human-to-human (H2H) interaction can be used to resolve this concern, that is, to allow users to pair their devices without losing physical possession of them at any point during the pairing process. Since users are interfacing with their devices individually, numerous HCI and physical interactions can be enabled similarly to the private class.

With regard to security, social pairing is vulnerable to external observation. On the one hand, the typical environment for the social class may present lower inherent risk as compared to the public class, for example, by occurring in a private house

instead of in public places. On the other hand, the social pairing may still occur in a public place. Also, social pairing procedures typically involve user interaction, which is particularly at risk of observation attacks. Thus, while the social class can suffer from a similar level of risk to the public class, users' perception of the risk may be lower, potentially reducing their tolerance for security measures that harm usability. Therefore, considerable attention should be given to optimizing user experience for social pairing procedures while still ensuring adequate security.

3.5.1.4 *Unattended*

Figure 9d depicts the unattended class which applies when two devices perform pairing without any user involvement (e.g., ZIP). For example, two IoT devices, for instance, sensors, located nearby can pair, and similarly wearables as well as IMDs can be paired. In the case of wearables and IMDs, a user is present but acts only as a carrier of the devices and does not consciously participate in the pairing process.

Since no user is involved, unattended pairing relies solely on various physical interactions, especially those used for data acquisition, that is, sensing. The key approach employed in unattended pairing is to utilize various sensor capabilities to measure the ambient environment over time. Thus, if two devices continuously sense sufficiently similar contexts, they interpret this as evidence of their physical proximity. When two devices believe that they are in physical proximity, they may then attempt to pair [243, 310]. The ambient environment does not only refer to physical characteristics such as wireless radio, audio, luminosity, or humidity. It can also correspond to measuring the human body, for example, a heartbeat rate [290] or muscle contraction [397] as well as capturing user specific actions, for example, a gait [309, 321], approach trajectory [176], or head movement pattern [204].

The unattended pairing (e.g., ZIP), eliminates user interaction, requiring much stronger focus on security.

In terms of security, the unattended class significantly differs from other application classes since the pairing devices communicate in a standalone fashion without explicit user control. This poses major security challenges such as physical access to the devices by an adversary in addition to their ability to efficiently monitor [242], disturb, or even manipulate [324] the pairing environment without being noticed. Moreover, it is not straightforward to unambiguously define a number of security domains in the unattended class. For example, the proposed ZIP schemes [243, 310] assume that devices originate from the same ownership, for example, either a user or infrastructure, and thus constitute a single security domain. An open question is the pairing of IoT devices which belong to different security domains.

The unattended pairing (e.g., ZIP) is by definition an autonomous process, removing all user interaction. It can be viewed as pushing the usability-security trade-off completely in the direction of usability. It is, therefore, not surprising that the security properties of unattended ZIP schemes are often weaker as compared to the other application classes. Thus, more research is required to devise more secure unattended pairing schemes.

3.5.2 *Classification of Pairing Schemes*

We employ the reported use cases of the pairing scheme as the basis for application class assignment.

To categorize different pairing schemes with respect to their application classes, we use the following approach. First, we consider pairing schemes that are surveyed in the physical and HCI sections. Second, for each pairing scheme, we seek a particular use case discussed by the original authors or look at the specific setting in which the implemented pairing scheme is tested and evaluated. Using this information, we explicitly assign each pairing scheme to one or more of the application classes. Finally, we consider for each pairing scheme whether it could be extended to other application classes either by an implicit reference in the paper or by considering the physical and HCI interactions necessary for a specific pairing scheme and comparing them with interactions possible in each application class. Then pairing schemes that rely on biometry and have been used in the field of IMDs, for example, [290], are outside of the scope of this survey, and thus are not included in these results. The results of our classification are presented in Table 4 and are discussed below.

The majority of existing pairing schemes target the private class, while ZIP addresses the needs of the unattended class.

In line with the prior research, we see that most of the proposed pairing schemes are aimed at the private application class. The public class is the second most targeted application scenario followed by the social and unattended classes, respectively. We also observe that many pairing schemes could be extended to other application classes, especially schemes that implement security mechanisms on the physical layer [42, 119] or utilize context sensing [243, 310]. An interesting trade-off exists between those two groups of pairing schemes. While the former can offer provable security guarantees, it requires low-level changes in the communication stack, which hinders the widespread adoption. In contrast, the latter group can be more easily deployed but lacks clear security guarantees [324].

Commercial pairing schemes claim to be agnostic to application classes, however, their security records often suggest the contrary.

Another observation is related to pairing schemes deployed in commercial products, for example, [28, 86, 87, 252]. Often these schemes are claimed to be applicable to multiple of the application classes, irrespective of whether they are suitable on the basis of their security properties. For example, the PBC scheme [86] is available in both the infrastructure mode as well as for Wi-Fi direct [89]. However, PBC is known to be vulnerable to MITM attacks, and the exposure is much greater in public and social contexts as compared to the private application class. Similar arguments apply to Just Works [28] which is the Bluetooth pairing scheme. Two other pairing schemes provided by the standardized bodies, namely Near Field Communication [87] and Out-of-band [252] rely on the NFC technology to transmit sensitive data, for example, a device generated password, in plain text. Despite being difficult, eavesdropping the NFC channel is not impossible and the chance of successful attack is much higher in public and social scenarios.

Table 4: Summary of surveyed pairing schemes utilizing application classes.

Pairing Scheme	Application classes			
	Private	Public	Social	Unattended
PBC [86]	●	●	●	○
Integrity codes [42]	◑	●	◑	◑
TEP [119]	●	◑	◑	◑
Just Works [28]	●	●	●	○
Noisy tags [46]	◑	●	○	○
Adopted-pet [5]	●	◑	○	●
NFC [87]	●	●	●	○
Out-of-band [252]	●	●	●	○
KeyLED [287]	●	◑	◑	○
Enlighten me! [113]	◑	●	○	○
Flashing displays [190]	●	◑	◑	○
Talking to strangers [17]	◑	●	◑	○
Loud and clear [123]	●	◑	◑	○
HAPADEP [333]	●	◑	◑	○
Zero-power pairing [141]	○	○	○	●
Ultrasonic ranging [237]	◑	◑	●	○
SBVLC [402]	●	●	●	○
Vibrate-to-unlock [302]	●	○	○	○
Shot [344]	◑	○	●	○
Vibreaker [7]	●	○	○	○
Amigo [372]	◑	●	◑	◑
Good neighbor [40]	●	◑	◑	○
Wanda [271]	●	●	◑	○
Audio pairing [310]	◑	◑	◑	●
Context-based pairing [243]	◑	◑	◑	●
MagPairing [172]	◑	◑	●	○
TAG [378]	●	●	○	○
MANA [115]	●	◑	●	○
AP authentication [291]	◑	●	○	○
Shake them up! [47]	●	○	◑	○
Shake well before use [236]	●	○	○	○
SAPHE [132]	●	○	○	○
Ultrasound authentication [185]	◑	●	●	○
Audio-visual patterns [272]	●	●	●	○
RhythmLink [209]	●	●	○	○
Seeing-is-believing [240]	◑	◑	●	○
Visible laser light [238]	◑	●	◑	○
VIC authentication [300]	◑	◑	●	○
BEDA [333]	●	◑	◑	○
Playful security [110]	◑	○	●	○
Safeslinger [84]	◑	○	●	○
Synchronized drawing [315]	●	○	○	○
Proximity authentication [202]	●	●	○	○
Checksum gestures [3]	◑	●	○	○

● = explicitly applies; ◑ = can be applied; ○ = does not apply.

3.5.3 *Discussion*

We discuss three takeaways for our survey of application classes.

Based on the investigation of the application classes, we discuss three open issues that have not been resolved by the prior research in SDP. First, how the presence of multiple security domains introduces complications. Second, what privacy issues arise in the respective application classes. Finally, whether pairing of devices should be valid indefinitely or only for a finite time.

3.5.3.1 *Multiple Security Domains*

Issues arise when pairing devices belong to different security domains. The goals of two pairing parties and the assets they protect can vary. This leads to security, privacy, and usability implications that can affect the adoption of a given pairing scheme. For example, in the public application class, the infrastructure side can provide acceptable user experience and a certain level of security but ignore users' privacy. Since privacy awareness is growing [59], many users may be reluctant to adopt a pairing scheme with such a drawback. The opposite situation is also feasible when the infrastructure side aims to enhance security and privacy, but this occurs at the expense of usability. In this case, users may become confused, as they seek to understand how pairing works. Such confusion could result in high error rates that can negatively affect both security and privacy as well as jeopardize the acceptance of the pairing scheme. Similarly, in the social application class, two users may have completely different attitudes towards security and privacy. Therefore, it should not be assumed that both participants are always attentive, collaborative, and security-motivated. A pairing scheme that is designed to operate in the presence of several security domains should take into account the possible inconsistencies existing between them and the impacts that this can have on user behavior and resulting security.

Security, privacy, and usability requirements of pairing can contradict when devices are controlled by different parties.

3.5.3.2 *Privacy Issues*

Each application class differs from the others in terms of privacy risks and their potential impact. The key privacy issues regarding each application class are summarized below.

The private class is the least problematic since only a single user is involved who directly controls both devices. Therefore, all private information remains within the sphere of control of the user involved. Nonetheless, there exists the potential risk of observation attacks exfiltrating private information.

The public class introduces the risk of user tracking. Consider, for example, a distributed service that allows paying for the petrol in some area. Initially, a user pairs with the terminal on a petrol station. Behind the scenes, the user is being enrolled in the service so that they can easily pay at other stations without the need to pair again. This example is both simple and realistic, and would allow the service to track the users, significantly impacting their privacy.

In the public, social, and unattended classes privacy issues are feasible in the form of user tracking and infiltration of personal information.

The social class is exposed to the risk of honest-but-curious participants. Such a threat can come in different forms, for example, peeking at another person's screen or observing their actions, or making a deliberate mistake to get physical access to the peer pairing device or retrieve extra data. None of the surveyed pairing schemes considers this type of attack. This is, therefore, a topic that justifies attention.

The unattended class is also prone to privacy leakage. The surveyed unattended ZIP schemes rely on context sensing, which is shown to be plagued with privacy issues [51]. Since IoT devices at home or wearables can disclose a great deal of private information about the user and/or their environment, unattended pairing schemes must account for privacy protection during pairing. This presents, perhaps, the most critical privacy issue uncovered during this survey. That is, devices that can pair autonomously and may have access to a considerable amount of private data, currently rely on the pairing mechanisms that do not take privacy into account, and the current state of the art does not yet offer any solution.

3.5.3.3 *Pairing Validity*

Traditionally, the validity of pairing follows the "once and forever" approach.

Historically, the norm for device pairing has been to establish a "once and forever" pairing. However, there are good reasons why this is not always the most sensible approach when instead the alternative may be more appropriate, that is, a temporary or transient pairing. In the private class, once-and-forever makes sense, where, for example, a user wishes to pair their smartphone with their car's infotainment system. In such cases, there exists an expectation of a long-term relationship between the devices, and that they will continue to belong to a single common security domain. In contrast, many pairing scenarios in the public class are more sensibly handled by creating transient relationships between devices, for example, when paying for a parking ticket, printing, or some other short-lived transient activity. In such situations, the devices belong to separate security domains, and the owner of one device has no control over the behavior of the other or its handling of any potentially private data. It, therefore, makes no sense for the pairing relationship to endure indefinitely. Indeed, there may be additional advantages to transient pairing, for example, by preventing the user tracking. An open question is how one should implement short-term pairing in the public application class.

One approach would be to unpair the devices after the necessary operation has been completed. However, it should be seamless and require no human effort, otherwise the usability will be jeopardized. Recently, a similar problem has been researched with respect to deauthentication [158], revealing the nontriviality of designing such schemes in a secure way.

Regarding the social class, both transient and long-term pairing may be applicable, depending on the social context and the amount of trust two people put into each other. For encounters of naturally limited scope or duration, for example, the exchange of contact details at a conference, pairing two devices permanently may be excessive.

In the public and unattended classes short-term pairing makes more sense.

Furthermore, the level of trust between people can degrade, which is another argument against pairing once-and-forever. Short-term pairing can also provide users with better security and privacy assurances, as the pairing is established only on an as-needed basis. This is in stark contrast to long-term pairing, which can be abused by another person or their device, for example, if the other person's device were to be compromised. However, if two users communicate regularly, for example, colleagues, having to repeatedly pair the same devices may be inconvenient.

Finally, considering the unattended class, the once-and-forever paradigm does not take into account the highly dynamic nature of IoT environments. In such environments, it is already common to pair devices only if they are physically colocated. It may, therefore, make sense to unpair devices whenever they conclude that they are no longer in close proximity. Yet, it remains unclear how to handle such unpairing events, including how to determine when the confidence of physical proximity reduces such that unpairing is justified.

In this section, we have discussed the application classes and provided the classification of existing SDP schemes. In the next section, we outline open research challenges and future perspectives in the field of SDP.

3.6 FUTURE CHALLENGES AND PERSPECTIVE

We present main challenges in the pairing field and avenues for future work.

In order to design and build viable pairing schemes, a wide range of challenges and open issues need to be resolved. We discuss several prominent challenges and provide a broad outlook for future research. We begin by explaining the need for creating adaptable SDP schemes that are independent of specific PHY and HCI channels. The importance of including human interaction in the security chain is then discussed in terms of its potential to improve both security and usability. Following this, we explain why it is critical that the design process of a pairing scheme begins with the target use case or application class so that again, security and usability can be maximized for each application. Fourth, we emphasize that SDP schemes currently lack ease of comparability, which hampers the evidence-driven improvement of state of the art for such pairing schemes. Finally, we highlight the problem that user privacy is rarely considered by the current cohort of SDP schemes.

3.6.1 *Adaptable Secure Device Pairing*

As has been shown through the course of this chapter, it is impossible to find a universal pairing solution. The selection of both PHY and HCI channels highly depends on a number of factors, including application classes, the environment and (social) context, potential attacks, the data to be exchanged, and availability of the channel. Thus, we argue that future research should be conducted towards a more general framework for pairing, which would take the aforementioned factors into account and develop

dynamic and customized pairing schemes built upon various PHY and HCI channels. In this case, the best security-usability trade-off can be obtained for a given situation. Such a framework should offer a higher level of abstraction, which would account for adding new factors, for example, in the form of "rules" that influence pairing as well as PHY and HCI channels seamlessly. Finally, we stress that the current design flow in pairing, which starts with the hardware capabilities, should be fundamentally rethought.

No universal pairing solution exists, thus pairing schemes need to be designed more adaptable.

3.6.2 *Including Human Interaction in the Security Chain*

So far, the role of human interaction in SDP has not been fully acknowledged as fundamentally important. Yet, human interaction is unavoidable in device pairing, for example, when a user wants to have more control and assurance of the pairing process. In our study, we have shown that human interaction can be used to improve security if properly utilized. However, users' incentives for pairing and the common HCI practices in pairing have not been well-studied. Surprisingly, few pairing schemes that we review account for mitigating user misbehavior or actually leveraging human involvement to achieve better security. Thus, we advocate for making the HCI component an indispensable part of the pairing design and outline several points that are subject to future investigation. First, having a continuous and transparent feedback loop between a user and pairing mechanism is crucial. As stated before, feedback to the user can mitigate many aspects of user misbehavior. Also, the prior research relies heavily on human-perceptible PHY channels, but the full potential of this property has not yet been realized. For example, with the feedback loop, both security and usability benefits can be obtained such as leveraging user perception to locate the source of the attack to improve security or making a human-device link more interactive to improve usability. Second, more research on basic user experience and its applicability to pairing should be carried out to facilitate the creation of more usable and error-resilient pairing schemes. Finally, we highlight several issues with regard to HCI observation attacks, however a more sophisticated analysis is required to evaluate the security of HCI channels.

Human interaction can provide extra security benefits, however, it is the job of a pairing scheme to leverage it, making the process more efficient, interactive, and transparent for a user.

3.6.3 *Application Class Driven Design*

Many of the surveyed pairing schemes are designed without a particular application class or use case in mind. However, our findings have shown that each application class has unique and often highly divergent security and usability requirements. Similarly, the sensitivity of the data being exchanged varies considerably among use cases [166], ranging from negligible, for example, exchanging contact information at a conference, to critical such as performing internet banking transactions. Therefore, it makes sense to begin the design process of an SDP scheme with the target data, use case, application

Pairing design starting from the application class improves security, usability, and comparability of the schemes.

class in mind. Only in this way can the resulting design be optimized to the particular needs and opportunities afforded by the target use case. This optimization of the security-usability trade-off is critical to ensure the best possible outcome.

3.6.4 *Improving Comparability of Secure Device Pairing Schemes*

A sound comparative analysis of different SDP schemes was previously impractical given the current design approach that starts from hardware capabilities instead of the target application class or use case. While the contributions of this chapter have facilitated comparison of SDP schemes, complications remain, for example, due to the lack of distinction between PHY and HCI channels in most of the SDP schemes surveyed. By shifting the focus to the target use cases and application classes, it becomes possible to identify a set of implementation-independent security and usability metrics. Those metrics could then be used to provide qualitative or quantitative comparisons between different pairing schemes within an application class. Building a more generalized attacker model within an application class would assist in defining such security metrics. Derivation of specific threat models for each of the application classes would be a particularly valuable contribution, as it would allow more objective assessment and comparison of the security properties of proposed pairing schemes.

3.6.5 *Considering User Privacy*

The growing privacy awareness and increasing threat to it from surveillance and sensing, urge the need for privacy-friendly pairing schemes.

Prior research has not adequately addressed privacy issues in SDP. The increasing numbers of users' devices store sensitive information and have advanced sensing capabilities with which many aspects of users' daily life can be directly measured or inferred [274]. Privacy concerns relating to this exist, and attacks that can obtain private data are feasible in the public, social, and unattended application classes. Several channels by which users' privacy can be readily violated are revealed in the process of this survey. While not necessarily new information, it is a clear reminder of the attention required to devise systems that are privacy-preserving. That is why SDP schemes should be designed with user privacy and the specific target use cases as the starting point, rather than physical hardware capabilities or other factors taking the leading role. Further research is also required to uncover hitherto undetected channels by which privacy may be violated so that they can be taken into account in future SDP schemes.

In this section, we have discussed open research challenges and future perspectives in the field of SDP. In the next section, we provide the concluding remarks and summary of this chapter.

3.7 SUMMARY

In this chapter, we survey existing Secure Device Pairing (SDP) schemes, including a new class of zero-interaction pairing (ZIP) schemes relying on context sensing. Specifically, we propose a system model and consistent terminology to facilitate meaningful comparison and analysis of SDP schemes. Our system model is based on the three key components drawn from the design space of SDP: physical (PHY) channels, human-computer interaction (HCI) channels, and application classes.

With regard to PHY channels, the survey reveals that data confidentially of the physical medium is very hard to guarantee in practice. Also, the emerging communication technologies such as visible light communication (VLC) and mm-Waves offer improved security properties. Other opportunities arise from the use of sensing of the shared ambient environment by nearby devices, as in the case of ZIP. Despite ZIP schemes providing improved usability by eliminating user interaction in pairing, their security requires improvement. Also, further research is needed to investigate the deployability of ZIP schemes such as how to deploy them on heterogeneous devices without common sensors or how fast they can pair without user involvement.

We recap contributions of this chapter.

With regard to HCI channels, we highlight the importance of building pairing schemes that are resilient to: (1) user misbehavior, (2) observation of user actions during the pairing process, and (3) honest-but-curious adversaries. It is only when these potential threats are properly considered, that HCI channels can play a trusted role in SDP schemes.

We also introduce application classes as a means of classification of SDP use cases. Through the identification of the target application class, considerable insights can be gained that can be used to guide the design of SDP schemes to optimize the security-usability trade-off for a particular use case. This stands in contrast to the current practice of beginning with physical hardware capabilities, instead of with the target use cases. This shift to use case oriented design is also identified as a necessary requirement to advance the state of the art. It is only by making this change, that SDP schemes within an application can be better compared in the future, whether qualitatively or quantitatively, allowing for evidence-based design and comparison of SDP schemes. Until this occurs, SDP schemes will likely continue to fail to address the security, privacy, and usability requirements of the various use cases.

4

EVALUATION OF ZERO-INTERACTION PAIRING AND AUTHENTICATION SCHEMES

PKI and user-assisted paring/authentication methods do not scale in the IoT.

Zero-interaction pairing (ZIP) and *zero-interaction authentication (ZIA)* schemes have been proposed to establish and maintain secure communication between Internet of Things (IoT) devices [181, 243, 310, 323, 364, 389]. These schemes allow pairing or authenticating devices without user involvement by utilizing sensor data collected from devices' ambient environment (e.g., audio), often called *context information* [266]. We refer to both ZIP and ZIA as *zero-interaction security (ZIS)* schemes.

ZIS has usability, scalability, and deployability advantages over user-assisted schemes.

ZIS schemes have the following advantages in the IoT: (1) they eliminate user effort to pair or authenticate devices imposed by user-assisted methods (e.g., entering a password) [93, 144], (2) they are more practical on resource-limited devices than centralized approaches such as public-key infrastructure (PKI), which have high complexity and limited scalability [79], and (3) they can be built on top of devices' sensing capabilities, reducing modification overhead and facilitating interoperability.

We reproduce five state-of-the-art ZIS schemes and collect real-world data in three scenarios.

To date, a number of ZIS schemes relying on various context information such as audio, luminosity, and acceleration as well as Wi-Fi and Bluetooth have been proposed [181, 243, 309, 310, 321, 323, 364]. Despite the promising results reported by these schemes, prior work raises questions about their practical applicability and security soundness [317, 324, 325]. The evaluation and comparison of the proposed ZIS schemes in realistic IoT scenarios are crucial yet missing. In this chapter, we fill this gap by conducting the *first large-scale comparative study* of existing ZIS schemes. Specifically, we reproduce *five state-of-the-art* ZIS schemes [181, 243, 310, 323, 364] and evaluate their ability to distinguish authorized and unauthorized devices on comprehensive datasets of context information collected in three realistic scenarios: (1) *connected car*, (2) *smart office*, and (3) *smart office with mobile heterogeneous devices*.

State-of-the-art ZIS schemes are challenged under realistic conditions.

In our scenarios, we collect seven different types of context information (cf. Table 5): audio, Wi-Fi and Bluetooth Low Energy (BLE) beacons, barometric pressure, humidity, luminosity, and temperature, obtaining datasets of 2, 214, and 23 GB for car, office, and mobile scenarios, respectively. Evaluating the reproduced schemes on the collected data reveals that they are challenged by our scenarios, demonstrating significantly higher error rates (i.e., between 0.6% and 52.8%) than obtained by the original authors. We also find that many ZIS schemes have limited adaptability to difficult circumstances (e.g., nighttime), and they are frequently not robust, with parameters optimal in one scenario leading to notably higher error rates in the other. To facilitate future research, we publicly release the collected context information, including audio recordings from the mobile scenario, evaluation results for the reproduced schemes as well as source code

Table 5: Context information used by the reproduced ZIS schemes.

ZIS scheme	Audio	BLE	Wi-Fi	Press.	Hum.	Lum.	Temp.	Details
Karapanos et al. [181]	✓	–	–	–	–	–	–	§A.1
Schürmann, Sigg [310]	✓	–	–	–	–	–	–	§A.2
Miettinen et al. [243]	✓	–	–	–	–	✓	–	§A.3
Truong et al. [364]	✓	✓	✓	–	–	–	–	§A.4
Shrestha et al. [323]	–	–	–	✓	✓	–	✓	§A.5

✓ = used; – = not used.

of our data collection tools, evaluation stack, and implementations of the reproduced ZIS schemes. In summary, we make the following contributions:

- We reproduce five state-of-the-art ZIS schemes [181, 243, 310, 323, 364] and design three realistic IoT scenarios from which we collect comprehensive datasets of various context information.
- We evaluate the schemes' performance and robustness for use in different scenarios. We also provide insights into the pitfalls of the reproduced schemes.
- We release the first open-source toolkit, containing datasets of various context information, including audio, together with the source code used to collect these data and implementations of the five ZIS schemes.

4.1 BACKGROUND

In this section, we first recap the terminology from Section 2.1 and also define the context feature. We then present our system and threat models and describe the ZIS schemes that we reproduce and evaluate.

4.1.1 *Terminology*

We recap a number of terms used throughout this chapter.

Context information. We define context information as the data collected from device sensors (e.g., microphones, light sensors), augmented with metadata like timestamps [266].

Context. We refer to a set of context information collected by a device from its ambient environment over time as the context of the device.

Colocation. We define colocation as a set of devices residing in the same physical space. In our scenarios, the spaces are different cars and offices, thus devices within the same car or office are colocated, otherwise non-colocated. The term colocation highly depends on the use case of the ZIS scheme. In the case of wearables, colocated devices

are on the same body [36], whereas for a smart home, colocated devices are inside a house [144].

Context feature. We define context feature as a concise context property computed from context information. Context features are based on a snapshot of context information [181, 310, 323, 364] or on relative changes of context information over time [243]. They calculate a distance or similarity metric between two samples of context information [181, 323, 364] or derive a sequence of bits (i.e., fingerprint) from a sample of context information [243, 310].

4.1.2 *System and Threat Models*

We assume an IoT scenario containing a number of devices that are colocated and equipped with a set of sensors to collect context information. The goal of ZIS schemes is to have two colocated devices establish a shared secret key (i.e., ZIP) or a proof of physical proximity (i.e., ZIA) without user interaction, utilizing context features to secure the process. We assume no established infrastructure and, in the case of ZIP, no prior trust between devices.

Our adversary is based on the models used in the reproduced ZIS schemes. The adversary is an IoT device located in an adjacent car or office. This device can be benign, accidentally trying to pair or authenticate with proximate devices in its wireless range (e.g., IoT device in a neighbor's car), or it can be malicious, intentionally trying to pair or authenticate with non-colocated devices. The adversary is non-colocated with benign devices, thus it can neither observe their context information nor compromise benign devices to circumvent a ZIS scheme. However, the adversary is physically close to benign devices (i.e., adjacent car or office), equipped with the same sensing hardware to collect context information, and can communicate with them over a wireless link.

Our adversary has more capabilities than threat models of the reproduced ZIS schemes.

The goal of the adversary is to obtain similar enough context information to fool benign devices into believing that it is colocated with them. Compared to threat models of the reproduced schemes, our adversary is more powerful as it possesses two extra capabilities. First, it remains permanently in close proximity to benign devices, including times of low-ambient activity such as during the night. Second, due to symmetric deployment of devices in our scenarios, the adversary has much better chances of following the same trends in context information (e.g., lighting conditions) as benign devices.

We purposely make our adversary powerful to evaluate the scheme performance in challenging scenarios. This allows us to establish the worst-case *baseline adversary*, facilitating comparison of the reproduced ZIS schemes (cf. Section 4.4), as well as gain first insights into possible attack vectors for an active adversary [324].

4.1.3 *Reproduced ZIS Schemes*

We reproduce five state-of-the-art ZIS schemes.

To select ZIS schemes for our study, we survey frequently cited schemes published at top security venues in the last five years. We select schemes that target IoT scenarios and utilize different context information or context features. We exclude schemes based on behavioral biometrics, for example, gait [36], gesture [322], or keystroke dynamics [229], as these schemes are designed for wearable IoT scenarios. In the end, we arrive at five schemes which we reproduce from the ground up, relying on the help of the original authors to ensure the correctness of our implementations. We briefly introduce each scheme in its respective result subsection (cf. Section 4.3) and refer to Appendix A for detailed descriptions.

4.2 STUDY DESIGN

We aim to collect a comprehensive dataset of context information.

We design our study to cover the majority of relevant context information used in current ZIS schemes. We select three realistic IoT scenarios: in the first two, we use identical sensing devices to collect context information, minimizing the effects of hardware variations on our results. In the third scenario, we use different sensing devices to evaluate the impact of device heterogeneity. This section describes the design and conduct of our experiments as well as ethical concerns when dealing with sensitive personal data collected in our study.

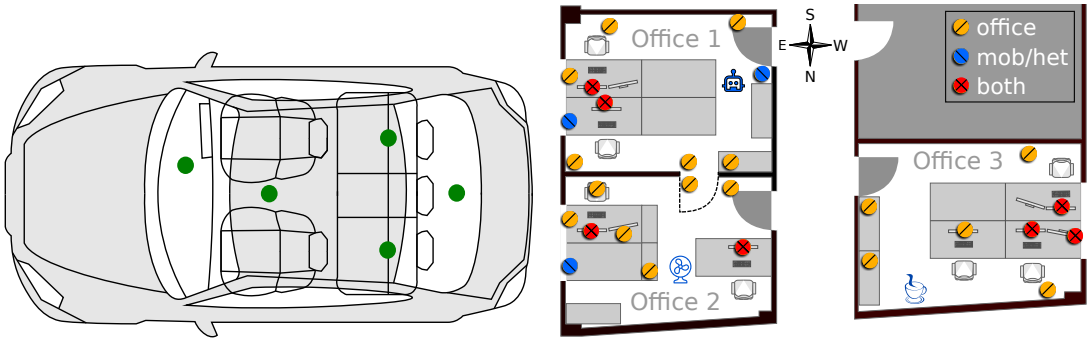
4.2.1 *Data Collection*

Our data collection is conducted using stationary and mobile heterogeneous sensing devices.

The goal of our experiment is to collect a comprehensive real-world dataset of context information that can serve as a baseline for comparing current and future ZIS schemes. In the first two scenarios, we collect data using a *Texas Instruments SensorTag CC2650* and *Raspberry Pi 3*. Audio data is collected using a *Samson Go* USB microphone, which records a mono audio stream with a 16 kHz sampling rate and encodes it using the lossless *FLAC* format. The Raspberry Pi also collects all visible wireless access points (APs) and BLE devices, including their signal strength, every ten seconds. The remaining context information (i.e., accelerometer, barometer, gyroscope, humidity, light intensity, magnetometer, temperature) are recorded using the SensorTag connected to the Raspberry Pi via BLE. [12] Sensor data is recorded with a sampling rate of 10 Hz.

In the third scenario, we additionally use *Samsung Galaxy S6* smartphones and *Samsung Gear S3* smartwatches to collect the same context information. Since those devices are not equipped with temperature and humidity sensors, we combine them with a *RuuviTag+*. We try to obtain the same sampling rate on all our devices, however, the Galaxy S6 limits barometric pressure and luminosity readings to 5 Hz. The summary

12 While accelerometer, gyroscope, and magnetometer are not used by any of the reproduced schemes, we collect their data for use in future schemes.



(a) Car (b) Office and mob/het

Figure 10: Device location map in the car (a) as well as office and mob/het scenarios (b).

of used sensing devices and sampling rates is given in Table 31 in Appendix A. All events that could influence the context information (e.g., windows/doors being opened or closed, people entering or leaving the recording area, and traces of mobile devices) are documented automatically or by hand in a ground truth sheet.

4.2.2 *Scenario 1: Car*

In the first scenario, we use two cars from different manufacturers. Each car is equipped with six sensing devices distributed inside the vehicle, as shown in Figure 10a. The devices occupy similar spots in both cars: one device is placed on top of the dashboard facing the windshield, inside the glove compartment, in between the front seats facing upwards, attached to each handhold above the two rear doors, and put in the middle of the trunk. This placement covers all prominent spots one might expect a sensor or a personal device inside a car (cf. Table 32 in Appendix A).

We collect context information from 120 km of driving.

After setting up the cars, we drive a predefined route of three hours and 120 km (74 M) on the afternoon of an autumn day. The time is chosen to ensure that the collection begins while the sun is visible and ends after sunset, to collect a variety of lighting conditions. The route includes city traffic, country roads, and highway (cf. Figure 40 in Appendix A for a map). We drive both cars close to each other within a distance of 20 m (65 ft), which we vary from time to time. In addition, we take a short break, with the cars parked side by side.

The challenge for the ZIS schemes is to identify colocated devices in a single car, while excluding devices from different cars that might be nearby or just listening to the same radio station.

4.2.3 *Scenario 2: Office*

A typical application for IoT devices is the deployment in a smart home or office. To collect realistic context information in this scenario, we deploy eight sensing devices in three office rooms, as shown in Figure 10b. We put the devices in similar places, representing typical IoT spots: one device is attached to the main screen of a workplace (e.g., smart workstation, several spots), above the windows (e.g., smart shades), near the ceiling lights (e.g., smart lights), in a closed cupboard (e.g., smart pen), near the door at around two meters height (e.g., motion sensor), and in a corner at around 2.5 meters (e.g., environmental sensor). The summary of device locations in the office scenario is given in Table 33 in Appendix A.

In the office scenario, the data is collected for one week.

We collect context information for one full week, resulting in five work days with people present and two days of the weekend, when Offices 1 and 2 are empty, and one person is working in Office 3. Offices 1 and 2 are adjacent and connected with a door, which is closed most of the time. Office 3 is on the opposite side of the floor. All three rooms have a similar setup in terms of size and position of furniture but a different number of participants working in them (i.e., one in Office 2, two in Office 1, and three in Office 3).

The collected dataset is intended for testing ZIS schemes designed for smart homes and offices. The challenge here is to distinguish between the three different rooms. Ideally, a ZIS scheme identifies all colocated devices in one room but excludes all other.

4.2.4 *Scenario 3: Office with Mobile Heterogeneous Devices (Mob/het)*

We extend the office scenario by including both static devices, permanently residing inside offices, and mobile devices carried by users (cf. Figure 10b). We add a number of appliances (i.e., vacuum robot and its station in Office 1, fan in Office 2, coffee machine in Office 3), facilitating device mobility when users move to use them. Each office is equipped with four static devices (i.e., SensorTags), covering similar spots and the appliances: one device is attached to the main screen of a workplace (e.g., smart workstation, several spots), near a power plug (e.g., smart plug), on top of a vacuum robot station (e.g., smart robot station) and coffee machine (e.g., smart coffee maker), near a fan (e.g., smart fan). We equip four participants with three mobile devices each: a laptop (with attached smartphone to collect context information), smartphone, and smartwatch. We also place a smartphone on top of the robot vacuum cleaner. Device locations are summarized in Table 34 in Appendix A.

We record context information with heterogeneous devices that are both stationary and mobile for eight hours.

We collect context information for eight hours from 9 am till 5 pm, representing a typical working day. Over the course of the day participants move freely between the offices to get a cup of coffee, have a meal, or attend a meeting, each time carrying a set of their mobile devices. We also move the vacuum robot between the offices, letting it autonomously run a complete cleaning cycle.

Similarly to the office scenario, the challenge for ZIS schemes is to distinguish devices present in the same office while excluding devices in others.

4.2.5 *Reproducibility and Reusability*

In total, our dataset contains 239 GB of context information, including more than 4250 hours of audio recordings, over 1 billion sensor readings, and over 12 million Wi-Fi and BLE beacons. Computing the context features of the reproduced schemes took over 300 000 CPU hours. The audio-based features were computed using Matlab on a high-performance cluster. The remaining features were implemented in Python on a high-performance server. After compression, they utilize almost 1 TB of disk space. This includes the computed features, aggregated statistics, and metadata for reproduction and validation following the recommendations by Benureau et al. [23].

The collected context dataset exceeds 239 GB.

To facilitate future reuse, we release the source code of the entire data collection and evaluation stack as well as the collected context information in an anonymized form, all intermediate and final data files (including machine learning models), and the code used to generate the visualizations. Privacy concerns prevent us from releasing the audio data recorded in the Car and Office scenarios, but we are able to provide researchers with the audio recordings from the Mobile scenario upon request [96]. See [97] for an index of all released data and code.

4.2.6 *Ethical Considerations*

The study was approved by our institutional ethical review board, data protection officer, and workers' council. Participants gave informed consent for the collection, use, and release of the data. During collection, the audio data was encrypted with keys controlled by the affected participants, requiring their explicit consent and cooperation to decrypt the data for processing. In the mob/het scenario, we gave participants the chance to inspect the recordings before obtaining informed consent for their release.

Data collection has been approved by our IRB

4.3 EVALUATION

In this section, we report on the performance in distinguishing colocated and non-colocated devices for the five reproduced schemes (cf. Table 6). The performance evaluation of each scheme is structured as follows. We first provide a concise overview of the scheme by explaining the context features used to distinguish colocated and non-colocated devices. Then, we explain the methodology of the original scheme and provide details of our evaluation. Next, we present and interpret the performance results of the scheme for each scenario. To quantify the performance, we compute the Equal Error Rate (EER) which is the point of equal False Acceptance Rate (FAR) and

We use FAR, FRR, and EER as performance metrics to evaluate ZIS schemes.

Table 6: Overview of the reproduced ZIS schemes and evaluation results.

Scheme	Conclusion	Best EER		
		Car	Office	Mob/het
Karapanos et al. [181]	Best EER for car scenario among the schemes. Limited robustness on intervals 5 to 15 seconds. Breaks down in heterogeneous setting.	0.006	0.098	0.157
Schürmann, Sigg [310]	Generates fingerprints with good randomness, but shows varying performance on subscenarios, and provides only limited robustness.	0.154	0.241	0.140
Miettinen et al. [243]	Weak fingerprint randomness leads to some EERs exceeding 0.5. Low robustness. Audio-based fingerprints perform better than luminosity-based fingerprints.	0.226	0.120	—
Truong et al. [364]	Achieves the best EER in office and mob/het scenarios, but shows low robustness and high reliance on audio feature, and struggles with heterogeneous settings.	0.104	0.069	0.123
Shrestha et al. [323]	Promising performance in car scenario, which drops in office and mob/het scenarios, low robustness, and high redundancy and ambiguity in features.	0.115	0.247	0.141

False Rejection Rate (FRR). In addition, we assess how much usability the schemes can deliver if a specific security level is required by setting a number of target FARs (between 0.1% and 5%) and analyzing the resulting FRRs.

Table 6 shows an overview of our results.

We evaluate the scheme robustness by analyzing an increase in error rates (either FAR or FRR) from the original EER when applying parameters found to be optimal in one scenario to another. This simulates a scheme being used in a scenario it was not trained on, like an IoT device optimized for office use being deployed in a car. We further summarize each studied scheme by comparing our results with the original findings and providing key takeaways from our evaluation. This facilitates a direct comparison of the different schemes in our scenarios.

We distinguish three subscenarios each in the car and office scenarios.

We introduce subscenarios to investigate the impact of changes in the environment (e.g., time of day, moving vs. parked cars) on the scheme performance. A subscenario represents a subset of context information collected at a specific stage in the scenario. For the car scenario, we distinguish three subscenarios: the *city* and *highway* subscenarios contain context information of the cars driving inside city limits or on the highway, respectively, and the *parked* subscenario includes context information from the time the cars are parked. Similarly, we construct three subscenarios for the office scenario: the *weekday* subscenario contains context information collected from Monday to Friday from 8 am to 9 pm, the *night* includes context information for all seven days from 9 pm to 8 am, and the *weekend* consists of context information from Saturday and Sunday in the timeframe from 8 am to 9 pm. We omit the subscenario evaluation in the mob/het scenario, as there are no specific stages in this scenario.

We assess the performance of all schemes except [364] and [323] on time intervals of 5, 10, 15, 30, 60, and 120 seconds with the length denoted t. The interval represents a timeframe over which context information is aggregated to compute a context feature, for example, a 5 second audio snippet or a 30 second Wi-Fi capture. [364] is evaluated on time intervals of 10 and 30 seconds, as the scheme is less well-suited to an arbitrary interval length due to the used features, while [323] does not use any intervals.

ZIS schemes are evaluated on intervals, ranging from 5 to 120 seconds.

4.3.1 *Karapanos et al.*

Karapanos et al. [181] propose using maximum cross-correlation between snippets of ambient audio from two devices to decide if they are colocated. The cross-correlation is computed on a set of one-third octave bands [165] and averaged to a similarity score. One-third octave bands split the audible spectrum (i.e., 20 Hz to 20 kHz) into 32 frequency ranges of different sizes. To prevent erroneous authentication when audio activity is low, a power threshold is applied to discard audio snippets with insufficient average power. The similarity score is checked against a fixed similarity threshold to decide if two devices are colocated, and can thus be authenticated. Tuning the similarity threshold allows trading usability for security and vice versa. The authors evaluate their scheme in several scenarios such as a quiet office, lecture hall, and café. The scheme details are given in Section A.1.

ZIA scheme by Karapanos et al. utilizes short snippets of ambient audio.

4.3.1.1 *Methodology*

To investigate the scheme performance, we compute similarity scores between colocated and non-colocated devices on different interval lengths. We increase the minimum length of audio snippet and maximum correlation lag to achieve a comparable level of synchronization to the original implementation. These changes have a negligible impact on the similarity score computation, as stated in Section A.1. To understand factors affecting the performance, we analyze the behavior of similarity scores on different octave bands.

We adjust the length of audio snippet and maximum correlation lag.

4.3.1.2 *Car*

We observe EERs between 0.006 and 0.050, decreasing with rising interval length (cf. Table 7). To understand this behavior, we compute the distributions of colocated and non-colocated similarity scores for each interval. Overlaps of these distributions explain the corresponding error rates: in the car scenario, the overlaps range from 1.1% to 8.5%. We observe a clearer separation between colocated and non-colocated similarity scores at longer intervals caused by a sharper drop of non-colocated similarity scores. When targeting low FARs, the resulting FRRs are below 0.2 on the intervals above $t = 15$, dropping rapidly with a growing FAR (cf. Figure 11a).

This scheme achieves lowest EERs among others in the car scenario.

Table 7: EER summary for Karapanos et al.

	Car				Office				Mob/het
t	Full	City	Highway	Parked	Full	Night	Weekday	Weekend	Full
5	0.050	0.071	0.009	0.124	0.141	0.140	0.135	0.143	0.157
10	0.032	0.049	0.003	0.071	0.133	0.132	0.128	0.136	0.168
15	0.026	0.043	0.002	0.060	0.128	0.126	0.123	0.129	0.170*[13]
30	0.017	0.031	0.001	0.022	0.118	0.115	0.116	0.115	0.172
60	0.008	0.014	0.002	0.007	0.107	0.102	0.109	0.099	0.179*
120	0.006	0.010	0.000	0.037	0.098	0.090	0.103	0.081	0.183*

Our octave band analysis shows the profound influence of lower frequencies (below 315 Hz) caused by a running car on the overall similarity score. This explains the lowest EERs reaching 0.0 in the uniform sound environment of a highway (cf. Table 7). The more diverse sound environment of a city shows a severalfold increase in EERs compared to the highway subscenario. Surprisingly, in a low-activity environment of parked cars, the EERs are only a few percentage points above the city subscenario. Investigating this phenomenon reveals that the power threshold discards up to 90% of similarity scores in the parked subscenario, retaining only those scores that result from intense audio activity.

On intervals of 10 and 15 seconds, the scheme shows some robustness.

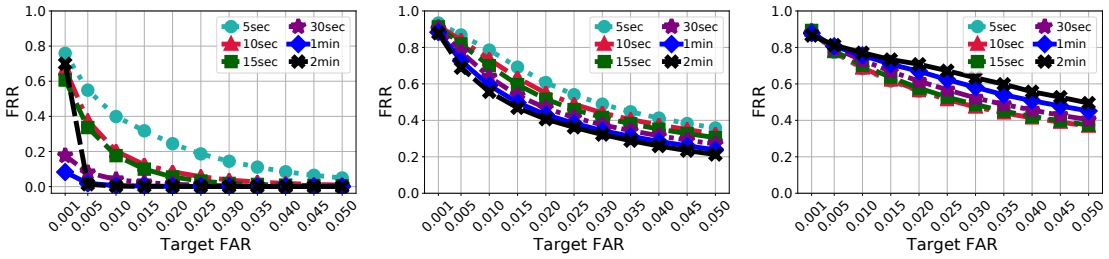
Applying office and mob/het EER thresholds to the car dataset leads to a marginal increase in error rates below 1 percentage point on the intervals $t = 5$ to 15 for the office and on $t = 10, t = 15$ for the mob/het, with other intervals showing severalfold growths in error rates. Among subscenarios, we see limited robustness between quiet (i.e., parked) and active environments (i.e., city and highway) at $t = 120$ as well as when applying city thresholds to the highway dataset for $t = 60, t = 120$.

4.3.1.3 *Office*

The office EERs are significantly higher than the car scenario.

In the office scenario, we observe EERs between 0.098 and 0.141, decreasing with growing interval length (cf. Table 7). We attribute these EERs to larger overlaps between colocated and non-colocated classes, ranging from 19% to 28%. We see a clear trend of higher similarity scores between non-colocated devices in adjacent offices (i.e., Offices 1 and 2 in Figure 10b). Our octave band analysis reveals close resemblance between these scores on lower frequencies below 250 Hz and on higher frequencies above 1250 Hz. Thus, both low frequencies penetrating adjacent offices and high frequency sounds like a police siren can increase non-colocated similarity scores. When targeting low FARs, the resulting FRRs start around 0.9 and never drop below 0.2 (cf. Figure 11b).

13 In cases where FAR and FRR do not match to three digits after the decimal, we average them and denote the result as EER*.



(a) Car (b) Office (c) Mob/het

Figure 11: FRRs with target FARs for Karapanos et al. in the full car, office, and mob/het scenarios.

We observe that higher audio activity of weekdays results in lower EERs on the intervals below $t = 30$. However, on longer intervals, the EERs of low-activity environments (i.e., night and weekend) become lower compared to the weekday. Investigating this phenomenon in more detail reveals two reasons for such a behavior. First, the power threshold retains a few similarity scores originated from intense audio activity in the night and weekend subscenarios. Second, in low-activity environments sounds are infrequent, localized, and short-term, making them easier to capture on longer intervals by colocated devices and less prone to be leaked to non-colocated devices.

Applying car and mob/het EER thresholds to the office dataset results in a minor increase in error rates below 2 percentage points on the intervals $t = 10, t = 15$ for the car and on $t = 10$ for the mob/het, with other error rates rising a few extra percentage points. In subscenarios, we observe robustness only between low-activity environments of night and weekend, showing an increase in error rates below 2 percentage points on all intervals.

4.3.1.4 *Mob/het*

We find the original power threshold to harm the scheme's availability.

Investigating similarity scores in the mob/het scenario reveals that 75% to 100% of the scores generated by smartphones and watches are discarded by the default power threshold (i.e., 40 dB) in the absence of intense audio activity (e.g., running vacuum robot). We adjust the power thresholds for smartphones and watches to 38 and 35 dB, respectively, significantly increasing the scheme availability in the cases of medium audio activity (e.g., low-voiced conversation), while still discarding the similarity scores from quiet environments.

EERs further increase due to mobility and heterogeneity of devices' microphones.

With the new power thresholds, we observe increased EERs between 0.157 and 0.183, rising with interval length, reversing the trend seen in the car and office scenarios (cf. Table 7). Once again, higher EERs are explained by larger overlaps between colocated and non-colocated classes, ranging from 33% to 36%. When targeting low FARs, the resulting FRRs vary almost linearly between 0.9 and 0.37 (cf. Figure 11c).

We find that microphone diversity and device mobility are likely reasons for the reversed EER trend. The similarity scores among heterogeneous devices are generally lower, decreasing significantly towards longer intervals. Our analysis suggests that the main reason for these lowered scores is diverse sensitivity and frequency response of heterogeneous microphones [182]. We empirically observe that smartwatch microphones are optimized for human voice but rather insensitive to low frequencies, while on smartphones low frequencies cause a lot of noise in recordings, and the USB microphones show the best signal quality on a wide frequency range. On longer intervals, device mobility further increases signal variation: the probability of capturing a unique signal (e.g., a keystroke by smartwatch) or wide-band scratching noises (e.g., smartphone rubbing against a pocket) increases.

Applying car and office EER thresholds to the mob/het dataset leads to a minor increase in error rates of up to 1.5 percentage points on the intervals $t = 10, t = 15$ for the car and $t = 10$ for the office, with other intervals showing several percentage points extra growths in error rates.

4.3.1.5 *Conclusion*

The ZIA scheme by Karapanos et al. works generally well.

Our results show that the scheme by Karapanos et al. can reliably distinguish colocated and non-colocated devices in the car scenario but degrades in performance in the office and mob/het. We generally achieve higher EERs compared to the authors who observe an EER of 0.002. Possible reasons for that are the increased distance between colocated devices and sustained closeness of non-colocated devices in our scenarios.

The scheme benefits from intense audio activity but drops in performance in cases of longer distance between devices, mobility, and microphone heterogeneity.

When the scheme is used among homogeneous devices (i.e., car and office scenarios), we observe better performance with increasing interval length and more intense audio activity. The difference between car and office EERs is due to a smaller distance between colocated devices in the car and more intense audio activity, especially on lower frequencies (e.g., on a highway). We see that highway EERs decrease marginally towards longer intervals, suggesting the use of short- to medium-sized intervals in active environments, reducing the runtime overhead of the scheme.

With heterogeneous devices (i.e., mob/het scenario), using longer intervals decreases the scheme performance, and intense audio activity is only beneficial if heterogeneous microphones can similarly record it (e.g., human voice), otherwise the performance will further decrease, especially on longer intervals. Considering that built-in microphones in mobile devices are user interaction oriented, the scheme can benefit from shorter intervals and audio activity in the frequency range of human voice in heterogeneous settings.

Some parameters (e.g., power threshold) need to be tuned on different hardware.

The power threshold allows the scheme to cope with quiet environments, sometimes at the price of excluding a significant portion of the dataset (e.g., parked car), trading off availability for security. However, as we have seen in the mob/het scenario, the power threshold proposed by the authors severely decreases scheme availability already

in the cases of medium audio activity, urging the need to carefully select this parameter, depending on the characteristics of the microphones.

The scheme consistently shows robustness on medium-sized intervals ($t = 10, t = 15$) among our scenarios, suggesting that it can potentially adapt to new environments on these intervals.

4.3.2 *Schürmann and Sigg*

The ZIP scheme by Schürmann and Sigg relies on short recordings of ambient audio.

Schürmann and Sigg [310] propose encoding a snippet of ambient audio into a binary fingerprint to pair two devices. The generated fingerprint consists of 16 individual shorter fingerprints that reflect the energy changes of successive frequency bands in the audio snippet over shorter timeframes. The similarity between the fingerprints derived by two devices informs a pairing decision. These fingerprints need to exhibit good randomness in order to secure a key establishment procedure between devices via fuzzy commitments. The authors evaluate their scheme in a series of deployments, ranging from staged lab measurements to recordings in a busy canteen and near a road. A detailed description of the scheme can be found in Section A.2.

4.3.2.1 *Methodology*

We generate slightly fewer fingerprint bits due hardware limitations.

We evaluate the performance of the scheme by generating fingerprints using different intervals t. Due to hardware constraints, we use a lower audio sampling rate, which reduces the length of the fingerprint from 512 to 496 bits. This change introduces a marginal deviation from the original implementation, as detailed in Section A.2. To evaluate the similarity of the generated fingerprints of two devices, we calculate the similarity percentage as $1 - (\texttt{hamming\_dist}/\texttt{length})$.

Randomness of fingerprints is crucial for a ZIP scheme.

The scheme uses a fixed similarity threshold that distinguishes colocated from non-colocated devices. In addition, we investigate the randomness of the fingerprints by interpreting them as random walks, with 1- and 0-bits representing steps in the positive and negative direction [36]. The outcomes will follow a binomial distribution if the fingerprints are uniformly random. We also investigate bit transition probabilities by interpreting each bit of the fingerprint as a state in a Markov chain.

4.3.2.2 *Car*

In the car scenario, the lowest EER is around 10%, increasing with less diverse audio activity and shorter interval.

We see EERs between 0.154 and 0.271, decreasing with increasing interval length t (cf. Table 8). These error rates correspond to the observed overlaps in similarity between colocated and non-colocated devices, which range between 30% and 51%. When optimizing for a low FAR, the resulting FRRs exceed 0.8 for certain parameters and never drop significantly below 0.3 (cf. Figure 12a). The system performs best in scenarios with diverse sound environments, like driving within city limits, showing consistently lower EERs in the city subscenario (cf. Table 8), dropping as low as 0.096.

Table 8: EER* summary for Schürmann and Sigg.

	Car				Office				Mob/het
t	Full	City	Highway	Parked	Full	Night	Weekday	Weekend	Full
5	0.271	0.228	0.247	0.362	0.419	0.423	0.406	0.440	0.363
10	0.226	0.175	0.199	0.359	0.351	0.365	0.319	0.380	0.257
15	0.211	0.157	0.170	0.361	0.317	0.340	0.267	0.347	0.215
30	0.179	0.121	0.126	0.361	0.277	0.308	0.215	0.309	0.175
60	0.160	0.100	0.106	0.359	0.256	0.287	0.194	0.280	0.154
120	0.154	0.096	0.112	0.328	0.241	0.275	0.178	0.253	0.140

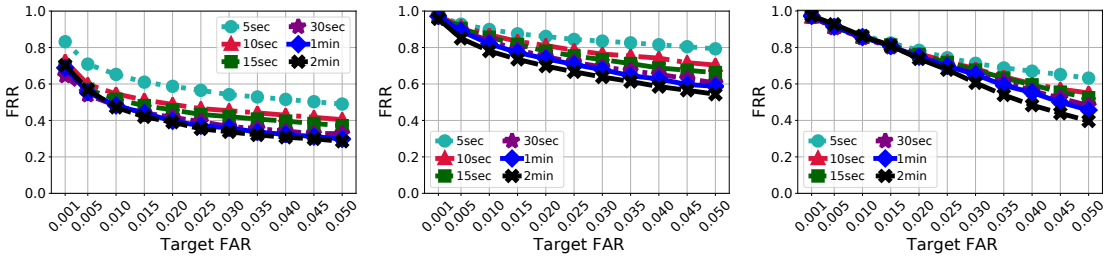
Environments with a uniform sound environment, like driving on the highway, show slightly increased error rates but still remain consistently below the error rates for the full dataset. In low-activity environments like parked cars, the scheme shows significantly increased error rates of up to 0.362—an increase of 0.134 over the city environment with the same parameters.

The produced fingerprint exhibit good randomness.

The fingerprints exhibit good randomness across all devices. Their Markov property is good with $P(b = 1) \approx 0.5$ for all bits. When interpreting fingerprints as random walks, the resulting distribution of endpoints is close to the expected binomial distribution (cf. Figure 13). When splitting the 496-bit fingerprints into their constituent 31-bit fingerprints and analyzing them separately, the random walks show a more varied distribution. Some are close to the expected binomial distribution (cf. Figure 13d), while others show a flatter distribution (cf. Figure 13c), indicating more fingerprints contain a larger number of 1- or 0-bits than expected. Investigating these sensors in more detail, we find that their microphones were affixed to surfaces that vibrated more than average. As fingerprints are derived from variations in signal energy over time, the biased fingerprints may have been caused by periodic variations in the energy induced by the vibrations.

Robustness of the scheme is limited.

Applying the threshold from the office scenario to this dataset results in an increase in error rates between 3.6 and 11.2 percentage points, with the larger changes occurring for $t = 5$ and 120. The mob/het threshold increases the error rates by 1.7 to 7.1 percentage points, with the largest changes for $t = 15$ and 30, while $t = 120$ shows the smallest change. In subscenarios, the most stable results are obtained between city and highway, changing between 4.1 and 9.5 percentage points in both directions. The other combinations show significantly larger error rate increases, in some cases up to 25.7 percentage points. This indicates that the scheme has limited robustness in cases where the environments are similar, but is not robust to larger changes in environmental characteristics.



(a) Car (b) Office (c) Mob/het

Figure 12: FRRs with target FARs for Schürmann and Sigg in the full car, office, and mob/het scenarios.

4.3.2.3 *Office*

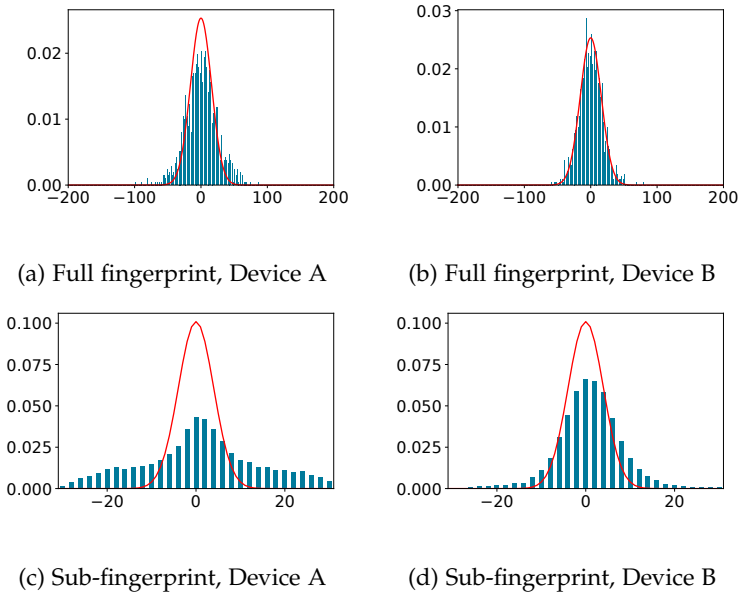
In the office, we observe generally increased EERs, ranging from 0.241 to 0.419 and decreasing with increasing interval lengths (cf. Table 8). These error rates are explained by the higher overlaps between colocated and non-colocated classes, which lie between 48% and 79%. In particular, we observe that the computed similarities between some non-colocated devices exceeded the similarities with all of their respective colocated devices, especially using smaller interval sizes t. Investigating these anomalous pairs in more detail reveals that the high similarities occur mostly at night and on the weekend, i.e., at times of very low ambient activity. However, the question why these particular devices are affected while others behave normally remains unanswered.

In the office scenario, EERs double compared to the car, while fingerprints show sufficient randomness.

When optimizing for a low FAR, the resulting FRRs for the full scenario are universally above 0.5 (cf. Figure 12b). Once again, the system performs best in environments with high audio activity, in this case the weekdays, showing significantly reduced error rates compared to the night and weekend.

The fingerprints again show good randomness, with a strong Markov property and random walks close to the expected distribution for the full fingerprints. When investigating the sub-fingerprints, we observe a slight bias towards 0 in the lowest three bits of some devices, with $P(b = 1) \approx 0.48$. Most of the affected devices are located in Office 2, but there is no discernible pattern in which devices exhibit this behavior and no obvious explanation.

Applying the car threshold to this dataset results in error rate increases of 3.9 to 10.9 percentage points, with the largest changes at $t = 5$ and 120. Conversely, the threshold obtained in the mob/het scenario will increase error rates by 5.6 to 12.6 percentage points, with the largest changes at $t = 10$, 15, and 30. The error rates of the night and weekend subscenarios remain almost completely stable when exchanging their thresholds. All other combinations show larger changes, often showing swings of more than 10 percentage points.



(a) Full fingerprint, Device A

(b) Full fingerprint, Device B

(c) Sub-fingerprint, Device A

(d) Sub-fingerprint, Device B

Figure 13: Distribution of fingerprint random walks of two representative devices for Schürmann and Sigg, Car scenario, t = 10. Expected binomial distribution in red.

4.3.2.4 *Mob/het*

In the mob/het scenario, the EERs are similar to the car.

The error rates in the mob/het scenario exhibit a larger spread than in the other scenarios, with EERs ranging from 0.140 to 0.363 and decreasing with rising interval lengths (cf. Table 8). They, once again, correlate with the overlaps in similarity between colocated and non-colocated devices, which range from 27% to 62%. When optimizing for low FARs, the resulting FRRs range from close to 1.0 to 0.40 (cf. Figure 12c).

The fingerprint randomness of mobile devices is worse than that of stationary devices.

Although the Markov property is universally good, the randomness of the fingerprints shows significant variation. While the fixed sensors show decent randomness, the mobile devices (i.e., smartphones and smartwatches) deviate from the expected distribution, showing similar behavior to the biased sensors in the car scenario. Part of this deviation can likely be explained by the different characteristics of the microphones (cf. Section 4.3.1.4). Devices that were covered (i.e., smartphones in pockets and smartwatches worn under long-sleeved clothing) show the largest deviation from the expected distribution, with strong biases towards sub-fingerprints consisting of mostly 1- or 0-bits. This, is likely related to the movement of cloth over the devices, generating wide-band scratching noises in combination with sound attenuation caused by the clothing.

Applying the car threshold to this dataset results in increases in error rates between 1.5 and 7.4 percentage points, with the largest changes for t = 15 and 30. The office threshold increases error rates by 5.3 to 11.8 percentage points, with the highest increases for t = 15 and 30.

4.3.2.5 *Conclusion*

The scheme by Schürmann and Sigg is unable to reliably distinguish colocated from non-colocated devices in our scenarios. We also observe unexplained high similarities for specific non-colocated device pairs in the office scenario. In particular, the scheme breaks down in environments with low-ambient activity (a limitation also noted by the original authors), however, even in high-activity environments like a driving car, the error rates exceed 10% for almost all parameters. Still, it may be possible to increase the overall performance of the scheme by excluding low-energy samples with a power threshold (e.g., similar to Karapanos et al. [181]).

The ZIP scheme by Schürmann and Sigg breaks in quiet environments.

The fingerprints exhibit good randomness in many cases, however, they struggle with noisy inputs, like vibration- or friction-induced sounds, and will in some cases generate fingerprints that consist almost entirely of 1- or 0-bits. In particular, devices carried in pockets or under long sleeves seem to cause problems.

The produced fingerprints show good randomness except for devices covered by clothes or affected by vibrations.

While the scheme is robust in some pairs of subscenarios, the robustness is very limited. Interestingly, the intervals behave differently for different combinations of scenarios—while the error rates of $t = 120$ are almost unaffected in some pairs, in others, they show very large changes. The same is true for other intervals like $t = 15$.

Schürmann and Sigg do not report error rates in their evaluation, so a direct comparison is impossible. However, the average separation between colocated and non-colocated fingerprints they find is larger than that observed in our scenarios. One possible explanation may be a tighter synchronization of audio signals in their experiment, as their samples were recorded by a single device with two microphones, thus avoiding any problems related to recordings not being exactly in sync. In a practical setting, such a tight synchronization between two devices will be more challenging to achieve (our synchronization method is described in Appendix A).

Direct comparison between the original and our results is impossible because the authors do not compute error rates.

4.3.3 *Miettinen et al.*

The scheme proposed by Miettinen et al. [243] uses two context features, one based on audio and the other on luminosity. In both cases, changes over extended timeframes are recorded and encoded into a binary *context fingerprint* of a fixed length b. The similarity of these fingerprints is then used to decide if devices can establish a connection by serving as a shared secret to bootstrap a key exchange using fuzzy commitments. Due to this usage, the randomness of the fingerprints is, once again, of interest. The authors evaluate their scheme in an office, a home scenario, and a mobile scenario simulating wearable devices. They also propose an optional extension to ensure sufficient fingerprint quality by discarding fingerprints with an insufficient *surprisal*, which measures how unexpected a fingerprint is for the current time of day. However, they do not evaluate the effect of this proposal. More details are given in Section A.3.

The ZIP scheme by Miettinen et al. encodes changes in ambient audio and luminosity over time into fingerprint bits.

Table 9: EER* results for Miettinen et al.

	Audio						Luminosity					
	t=5	10	15	30	60	120	5	10	15	30	60	120
Car												
b=64	0.377	0.389	0.396	0.384	0.382	0.263	0.506	0.505	0.504	0.505	0.501	0.517
128	0.358	0.368	0.370	0.372	0.362		0.507	0.506	**0.492**	0.499	0.516	
256	0.329	0.335	0.328	0.295			0.505	0.499	0.498	0.514		
512	0.344	0.294	0.287				0.497	0.504	0.517			
1024	0.297	**0.226**					0.498	0.522				
Office												
64	0.249	0.228	0.218	0.204	0.202	0.193	0.495	0.491	0.486	0.468	0.444	0.425
128	0.226	0.206	0.203	0.190	0.184	0.172	0.487	0.477	0.469	0.447	0.418	0.406
256	0.212	0.196	0.193	0.180	0.165	0.147	0.483	0.470	0.459	0.421	0.397	0.403
512	0.203	0.188	0.185	0.166	0.136	0.131	0.471	0.454	0.440	0.397	0.362	0.400
1024	0.197	0.184	0.178	0.135	**0.120**	0.126	0.454	0.437	0.426	**0.344**	0.363	0.362
Mob/het†												
64	0.377	0.368	0.364	0.383	0.349	0.314	**0.517**	0.520	0.520	0.521	0.525	0.524
128	0.356	0.344	0.339	0.371	0.325		0.521	0.523	0.522	0.525	0.519	
256	0.331	0.322	0.305	0.365			0.521	0.528	0.520	0.524		
512	0.306	0.308	0.291				0.522	0.526	0.518			
1024	**0.287**						0.525					

Empty cell denotes insufficient data to generate fingerprint. Best value in scenario marked in **bold**.
† Computed on subset.

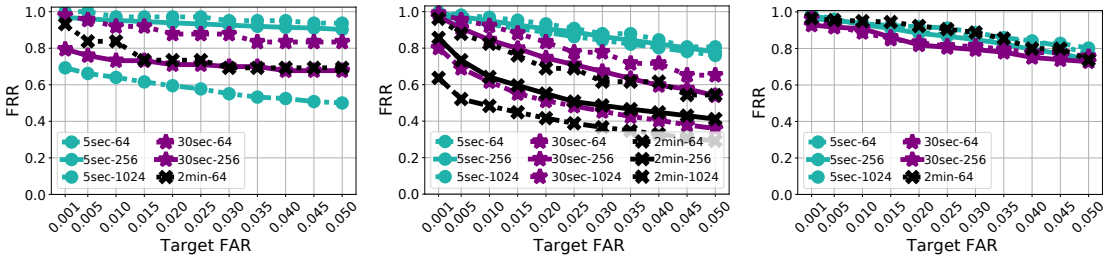
4.3.3.1 *Methodology*

Our methodology is identical to that used for the paper by Schürmann and Sigg (cf. Section 4.3.2). As the fingerprints generated by the scheme span long timeframes (i.e., up to 34 hours), we omit the subscenario evaluation, as allocating fingerprints to specific subscenario timeframes is impossible.

4.3.3.2 *Car*

In the car scenario, the EERs exceed 20% and 49% for audio and luminosity, respectively.

Both luminosity- and audio-based fingerprints show relatively high error rates, with the lowest observed EER* being 0.492 and 0.226, respectively (cf. Table 9). These high error rates can be explained by the large overlap of similarity percentages between the colocated and non-colocated groups, showing overlaps between 83% and 96% for the luminosity fingerprints. The overlaps are lower, but still significant for the audio fingerprints, with overlaps between 39% and 79% being observed. When aiming for



(a) Audio feature, Car (b) Audio feature, Office (c) Audio feature, Mob/het

Figure 14: FRRs with target FARs for Miettinen et al. in the full car, office, and mob/het scenarios for a selection of parameters.

a specific FAR, the resulting FRR is universally above 0.5 for the audio fingerprint (cf. Figure 14a). For the luminosity feature, the FRRs are 1.0 for all targeted FARs, indicating that all samples are rejected, making the scheme usability unacceptable.

Once again, the security of the scheme does not only depend on the error rates but also on the randomness of the generated fingerprints. Here, we observe the luminosity fingerprints to be heavily biased towards zero. The audio fingerprints contain more 1-bits but still do not show sufficient randomness. This limited randomness and high bias also explain the high overlap in the fingerprint similarity distributions. Rejecting fingerprints with insufficient *surprisal* excludes over 90% of the luminosity fingerprints, even for the smallest specified surprisal value, and consistently increases error rates for all attempted thresholds. For audio fingerprints, we evaluate a series of thresholds for different parameters and find that in many cases the error rates do not decrease significantly and in some cases will even increase, unless over 95% of the dataset is excluded.

The resulting fingerprints have weak randomness, while the proposed surprisal decreases scheme's availability without increasing security.

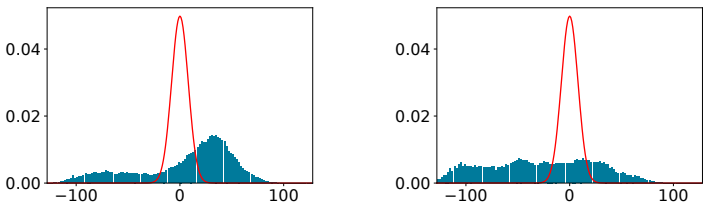
Applying the threshold from the office scenario increases the error rates for audio fingerprints by varying amounts, in some cases remaining stable, in others increasing by close to 25 percentage points, where higher values of b and t result in higher robustness. For luminosity fingerprints, increasing b reduces robustness and can lead to all samples being rejected, while smaller values of b with large t sometimes show stable error rates. With the mob/het threshold, the system rejects all audio fingerprints. On luminosity fingerprints, it shows unpredictable behavior, being robust for certain parameters and rejecting all samples for others, with no discernible patterns observed.

We see limited robustness of the scheme for both audio and luminosity modalities.

4.3.3.3 *Office*

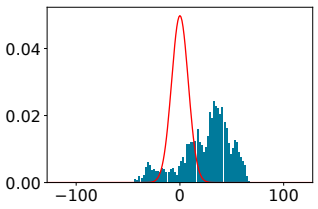
In the office scenario, we observe lower error rates, with EERs between 0.249 and 0.120 for the audio fingerprints (cf. Table 9), which can be explained by the decreased overlaps between the fingerprint similarity percentages of colocated and non-colocated devices (i.e., between 24% and 49%). For the luminosity fingerprints, the error rates remain high, with the lowest observed EER being 0.344, which can be explained by overlaps

The scheme shows its lowest EER of 12% for the audio modality in the office scenario.

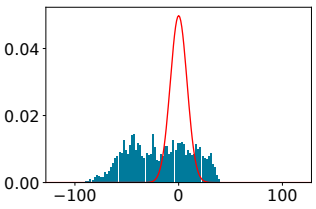


(a) Device A, t = 5

(b) Device B, t = 5



(c) Device A, t = 120



(d) Device B, t = 120

Figure 15: Distribution of fingerprint random walks of two representative devices for audio feature of Miettinen et al., Office scenario, b = 128. Expected binomial distribution in red.

between 80% and 99%. In many cases, FAR and FRR only become equal with thresholds close to 100% similarity, at which point the FAR becomes 0.0 and the FRR 1.0. When aiming for a low FAR, the resulting FRRs remain large for both audio (cf. Figure 14b) and luminosity fingerprints (where the error rate is almost universally 1.0).

Similar to the car, we observe severe randomness biases in the fingerprints.

The luminosity fingerprints consist overwhelmingly of 0-bits, which explains the observed overlaps in similarity percentages. This can be explained by the low variance in luminosity in offices, which are often lit by electric lighting with only very infrequent changes. Audio fingerprints show more variance but are usually biased, with the probability of obtaining a 1-bit varying between 0.4 and 0.63. The distribution within the fingerprint is also unequal, as the fingerprints are almost completely zero at night, leading to further biases (cf. Figure 15). Rejecting fingerprints with insufficient surprisal, once again, excludes most of the dataset in the luminosity feature and only leads to improvements of 1–2 percentage points in the audio fingerprints while excluding 10-20% of the dataset.

Applying the car threshold to the dataset again results in varying increases in the audio fingerprint error rates, following the same trends outlined for this combination in the car section. The luminosity feature accepts almost all fingerprints, with FARs between 0.9 and 1.0. With the threshold from the mob/het scenario, the scheme rejects all audio fingerprints, and either accepts or rejects all luminosity fingerprints, following no discernible pattern.

4.3.3.4 *Mob/het*

In the mob/het scenario, the colocation of devices changes over time, as they move between offices. This makes it impossible to perform a comprehensive evaluation of the scheme proposed by Miettinen et al., as the mobile devices often do not stay colocated with any device long enough to establish a pairing. We, thus, limit our evaluation to a timeframe of approximately 2.5 hours at the beginning of the recording, during which the colocation of all devices remains static.

The mob/het scenario exhibits highest EERs, which are caused by mobility and microphone heterogeneity.

The error rates for both luminosity and audio fingerprints are increased compared to the other scenarios—in some cases significantly (cf. Table 9). The EERs of the luminosity fingerprints are above 0.5 for all combinations of parameters, and the best observed EER of the audio fingerprint exceeds those of the car and office scenarios by more than four percentage points. Aiming for a low FAR will result in unacceptably high FRRs (cf. Figure 14c). For audio fingerprints, this decreased performance can be attributed to the varying microphone characteristics leading to sounds being received with different amplitudes, resulting in deviating fingerprints. The luminosity fingerprints are challenged by the different positions of the mobile devices, which are in some cases carried in pockets, and thus do not receive the same luminosity readings as other devices.

As before, the produced fingerprints are not random.

Luminosity fingerprints remain heavily biased towards zero, and the audio fingerprints also frequently show strong biases towards 1 or 0, following no discernible dependence on the parameters t and b. Using the surprisal thresholds leads to small improvements (i.e., less than 2 percentage points) in the error rates for audio fingerprints, at the cost of excluding 10-20% of the dataset. For luminosity fingerprints, even the smallest threshold excludes 96% of the dataset and does not improve the error rates significantly.

Applying the car threshold to the dataset will result in varying error rates, often rejecting all samples, and never coming close to the original error rates for the audio fingerprint. The luminosity fingerprints will occasionally reach error rates close to the original, following no particular pattern, but will often reject all fingerprints as well. The behavior of the office threshold is similar, rejecting close to all samples for both fingerprint types.

4.3.3.5 *Conclusion*

The ZIP scheme by Miettinen et al. shows prohibitively high error rates in our scenarios.

Our evaluation has shown that the scheme is unable to provide good separation between colocated and non-colocated devices, exhibiting large FARs and FRRs. Low FARs can only be obtained at the cost of large FRRs. The best performance is achieved using audio fingerprints in the office scenario, likely because of the homogeneous hardware and low level of background noise. We also investigate the impact of using the surprisal thresholds proposed by Miettinen et al. and find that it will in some cases slightly increase the performance of the scheme but excludes a significant fraction of the dataset in the process, reducing the availability.

We find that the surprisal proposed to tackle low-entropy context does not achieve its purpose, generating fingerprints of weak randomness.

The randomness of the generated fingerprints is limited, with devices often showing strong biases towards either 1 or 0, enabling adversaries to break the scheme in a practical deployment by guessing the fingerprint. This illustrates the importance of using an environmental data source with sufficient variability (unlike fixed electric lighting) and a quantization scheme that ensures a roughly equal proportion of 1- and 0-bits, for example, [310].

The authors do not report error rates, making a direct comparison of their and our results impossible.

Miettinen et al. do not compute error rates but observe an average colocated luminosity and audio fingerprint similarity of 95% and 91.8%, respectively, using an interval of $t = 120$ in their office scenario. For non-colocated devices, they report similarities between 68% and 88% for luminosity and 62% to 71% for audio. We were unable to achieve this degree of similarity on our dataset.

4.3.4 *Truong et al.*

The ZIA scheme by Truong et al. combines features computed from several context modalities and machine learning.

Truong et al. [364] propose combining multiple types of context information to increase the reliability and performance of ZIA schemes. They collect Wi-Fi, Bluetooth, GPS, and audio data and compute a number of context features, aggregated over a time interval t. Features for the first three modalities are computed based on distances between sets of observed devices and signal strengths, while the audio data is used to calculate the maximum cross-correlation and time-frequency distance between the audio snippets. Colocation is determined using a machine learning classifier, which has been trained with a labeled dataset of colocated and non-colocated features. Due to technical limitations of the used hardware, we were unable to capture GPS data. However, Truong et al. find that the GPS feature contains the least amount of discriminative power in their dataset, which is obtained by having volunteers in two cities collect context information and colocation ground truth data using smartphones and tablets in locations of their choice. The full details of the scheme are given in Section A.4.

4.3.4.1 *Methodology*

We use Gradient Boosting Machine and Random Forest as machine learning classifiers in our evaluation.

To investigate the performance of machine learning colocation prediction, we use the H2O framework [357] to train a set of classifiers and pick the best performers. We evaluate Gradient Boosting Machines (GBMs) [107] and Random Forests (DRFs) [35] as classifiers, and then select the algorithm that gives the best cross-validated performance. These classifiers perform well in a wide range of datasets [85], they are fast, and they can handle instances with missing data directly in the model, allowing us to use instances with missing data in our datasets, which would otherwise have to be discarded. This is desirable, as in the real world, data may be incomplete (e.g., due to missing GPS fixes). These partial instances still provide information about the generating distribution, and therefore are beneficial for the model, as shown by Tang et al. [351]. When building the cross-validation folds, H2O uses stratified sampling. This helps alleviate issues that can

Table 10: Classification results for Truong et al., Car.

Scenario	t	Model	EER	AUC	Acc.
Car	10	GBM	0.111	0.961	88.8%
– City	10	GBM	0.038	0.993	96.1%
– Highway	10	GBM	0.026	0.995	97.4%
– Parked	10	GBM	0.271*	0.813	72.9%
Car	30	GBM	0.104*	0.967	89.6%
– City	30	GBM	0.032	0.995	96.8%
– Highway	30	GBM	0.022	0.997	97.7%
– Parked	30	GBM	0.282*	0.803	71.7%

arise from class imbalances such as in datasets that contain more non-colocated than colocated instances.

To rank the classifiers, we use 10-fold cross-validation (CV) and estimate the Area Under the Curve (AUC), which measures the quality of the predictions irrespective of the selected thresholds. A higher AUC indicates a more accurately discriminative model. Using this measure is valid in our case, as we are interested in lower false accept and false reject errors along the predicting threshold domain.

For training, we employ a 10-fold cross-validation, computing Area Under the Curve as a performance metric in addition to FAR, FRR, and EER.

For the learning, we let H2O split the data into training and validation datasets of 80% and 20%, respectively. H2O will train a set of models independently from each other and automatically perform a parameter search to find optimal parameters for the specific dataset. Once we have found the top performing models, we get the cross-validated predictions $\hat{y} \in [0, 1]$. To convert those predictions to actual classes, we use a threshold T and classify predictions that satisfy $\hat{y} > T$ as colocated. By optimizing the threshold, we balance the values between FAR and FRR to obtain the EERs or our target FARs. We also evaluate the impact of the individual features in the process using the normalized relative importance. Truong et al. evaluate different interval length and come to the conclusion that increasing t above 10 seconds does not significantly increase the performance of the scheme. To validate this result, we evaluate two datasets, with $t = 10$ and 30. We present our results in Table 10 and Table 11.

4.3.4.2 *Car*

In this scenario, we obtain an EER of 0.111 and 0.104 for $t = 10$ and 30, respectively. Cross-correlation and time-frequency distance of the audio recordings account for 85% of the relative feature importance. This is expected, as the driving route passed through many areas without Wi-Fi APs and BLE devices.

In the car scenario, the EERs go as low as 2.5%.

Table 11: Classification results for Truong et al., Office.

Scenario	t	Model	EER	AUC	Acc.
Office	10	GBM	0.084*	0.974	91.5%
– Night	10	GBM	0.08*	0.976	91.9%
– Weekday	10	DRF	0.087	0.973	91.3%
– Weekend	10	GBM	0.071	0.981	92.9%
Office	30	GBM	0.069	0.982	93.1%
– Night	30	GBM	0.063*	0.984	93.6%
– Weekday	30	GBM	0.072*	0.981	92.8%
– Weekend	30	GBM	0.053	0.989	94.6%

When investigating subscenarios, we observe that the parked subscenario exhibits a significantly higher EER than the other subscenarios. In this subscenario, the models also show a lower reliance on audio features, with those features making up only 64% of importance, and a higher precedence being given to Wi-Fi and BLE features. This can likely be explained by the low audio activity in this subscenario, leading the model to use these less reliable features, and thus reducing classification performance.

The machine learning classifiers heavily rely on audio features.

The FRRs, given in Figure 16a, show similar trends: city and highway exhibit the lowest FRRs for the desired FARs, with the parked car significantly above them, and the full dataset somewhere in between. The FRRs also show a steeper drop in the beginning that tapers off later.

To test the robustness of the model, we use it to obtain predictions on the data from the other scenarios, applying the EER threshold determined before. This results in significantly increased error rates for all combinations of scenarios and intervals, with FARs larger than 0.44 for the office scenario, and FRRs in excess of 0.6 for the mob/het scenario, indicating that the model's performance will deteriorate when used on data from a scenario it has not been trained on, and thus is not robust to being operated in different environments.

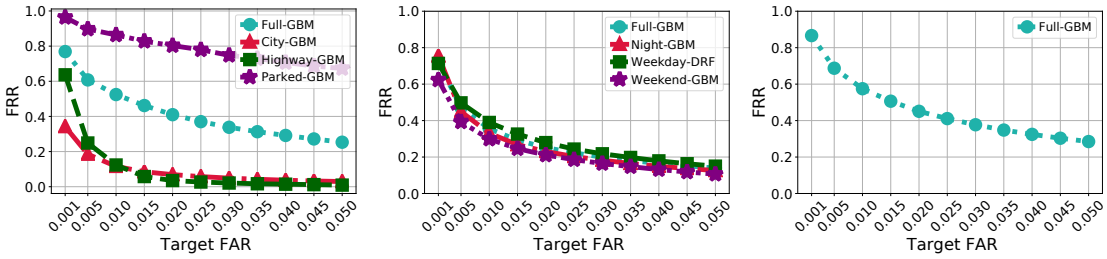
Using the car's model to make predictions on the office and mob/het scenarios leads to a sharp drop in performance.

The robustness of models trained on subscenario datasets shows significant variation. Combinations of the city and highway subscenarios show changes between 0 and 4 percentage points, while combinations involving the parked subscenario show changes between 25 and 82 percentage points. This indicates that the models are robust to small changes in the environment but cannot adapt to significant deviations.

4.3.4.3 *Office*

We observe a slightly improved EER of 0.084 (t = 10) and 0.069 (t = 30). Surprisingly, the Wi-Fi features are not more relevant, despite Wi-Fi being one of the best features

This scheme achieves lowest EERs among others in the office scenario.



(a) Car (b) Office (c) Mob/het

Figure 16: FRRs with target FARs for Truong et al. (t = 10).

reported by Truong et al., and one would expect more stable signals for stationary devices compared to the mobile car scenario. However, our results show that the audio features are even more relevant with a combined relative importance of 91%. To investigate if this is caused by the missing Wi-Fi data in the dataset (cf. Section 4.4), we repeat the analysis, excluding instances where the Wi-Fi data is missing due to a scan error, and obtain unchanged results. Thus, even in a dataset that contains Wi-Fi data for all samples, the feature does not become more relevant for the classifier. The subscenarios show a much more similar behavior than in the previous scenario, with EERs between 0.071 (weekend) and 0.087 (weekday). This trend is also shown in the FRR evaluation in Figure 16b, where the curves are all closely matched.

When running the model on the car dataset for t = 10, we obtain an FAR and FRR of 0.174 and 0.412, respectively, with t = 30 increasing the error even further. Applying it to the mob/het dataset yields error rates of 0.022 and 0.711, respectively, once again, increasing further for t = 30. This shows that the models are sufficiently different such that generalization is low, and therefore robustness of the scheme suffers. Switching between the different subscenarios results in less pronounced changes, but still in some cases doubles the error rates. The scheme appears especially challenged when applying the weekend model to the other subscenarios, often doubling the error rates, while the weekday model is fairly robust, with only minor changes to most error rates. This is likely due to the higher complexity of the weekday dataset, which contains data from a more diverse set of situations.

Using the subscenario's weekday model to make predictions on the weekend and night shows potential for generalization.

4.3.4.4 *Mob/het*

In the mob/het scenario, we obtain EERs* of 0.127 and 0.123, respectively (cf. Table 12). Once again, the most important features are audio-based, although their importance is less pronounced, making up only 60% and 56% of relative feature importance for t = 10 and 30, respectively. Optimizing for a low FAR will result in FRRs between 0.9 and 0.3 (cf. Figure 16c). This lower overall performance and the reduced prominence of the audio features is likely related to the issue of heterogeneous microphone characteristics, which

In the mob/het scenario the importance of audio-based features is lower likely due to microphone heterogeneity.

Table 12: Classification results for Truong et al., Mob/het.

Scenario	t	Model	EER	AUC	Acc.
Mob/het	10	GBM	0.127*	0.946	0.873%
Mob/het	30	GBM	0.123	0.949	0.877%

we previously observe in the scheme proposed by Karapanos et al. (cf. Section 4.3.1.4), as Truong et al. use similar audio features.

Using the model to classify the car and office datasets results in significantly increased error rates (i.e., FAR > 0.7, FRR $\leqslant 0.22$ for all combinations), showing that the model is not robust to different environments.

4.3.4.5 *Conclusion*

The ZIA scheme by Truong et al. achieves relatively low EERs, which are still higher than that reported by the authors.

Our evaluation shows that the scheme can achieve a good EER in some of our scenarios, although it does not reach the error rates of the original paper, which reports a FAR and FRR of 0.0198 and 0.0167 for $t = 10$. We also see that models generated in one scenario show a significant loss in accuracy when being used in another scenario, and that the scheme encounters problems when using heterogeneous microphones. The authors also conduct an experiment where pairs of devices are placed in close proximity (which matches our office scenario), obtaining a FAR of 0.0476, but they do not report the FRR, which prevents a direct comparison.

Contrary to the original results, we find lower reliance of the classifier on Wi-Fi features and better performance on longer intervals.

Contrary to the original evaluation, the classification performance increased with larger intervals. We also see a much higher importance of the audio feature than the original paper and a correspondingly lower importance of the Wi-Fi feature. This is likely related to the collection strategy employed by Truong et al., who collect their dataset in different locations across two cities, which can be easily distinguished by their different Wi-Fi signals.

The subscenario evaluation shows that the system does not work well in environments with low-ambient activity, like cars parked in areas without Wi-Fi and BLE devices. The differences between the subscenarios are less pronounced in the office scenario, where a larger number of Wi-Fi and BLE devices are visible at all times.

In our evaluation, we do not use GPS data and utilize a different machine learning classifier than the authors.

Two factors limit the validity of our results. First, we do not collect GPS data, used by Truong et al. We assume that the impact would have been low in the office and mob/het scenarios, where devices are located close to each other and mostly static, however, it may have improved performance in the car scenario. Second, we use a different classifier than the authors, who utilize a Multiboost classifier [382], which is not supported in H2O. Still, DRFs and GBMs rely on ensemble methods similar to Multiboost, and are unlikely to give significantly worse results.

4.3.5 *Shrestha et al.*

The ZIA scheme by Shrestha et al. follows a similar approach to Truong et al. utilizing environmental sensor modalities.

Shrestha et al. [323] propose combining readings from temperature, humidity, altitude, and precision gas sensors to decide if two devices are colocated. They compute the absolute difference between the readings of two devices and use a Multiboost classifier [382] trained on a labeled dataset to distinguish colocated and non-colocated devices. As our devices did not feature a precision gas sensor, we omit this feature. The sensor readings are not averaged over time intervals but used individually. Their dataset is obtained by collecting data from several locations using a pair of devices. Any data collected at different locations and times is interpreted as non-colocated. Additional details of the scheme are given in Section A.5.

4.3.5.1 *Methodology*

We employ similar methodology to Truong et al., finding that the obtained context features have high potential for compression.

Although the machine learning methodology is identical to that used for the paper by Truong et al. (cf. Section 4.3.4), the characteristics of the datasets and volume of data demand different treatment. One assumption made by any classifier in machine learning is that it estimates a surjective function from a vector of features $\hat{\mathbf{x}}$ to a particular class c, i.e., all unique instances in the dataset map to exactly one class. However, our datasets do not fulfill this requirement, as several identical instances map from the same feature values to different classes. As the classifier has no additional data to base its decision on, it is unable to distinguish these ambiguous instances, and thus can never reach a performance of 100%, i.e., the EER has a lower bound larger than 0. This indicates that more features are needed to discriminate the classes properly. We show the percentage of these ambiguous instances (Amb.) in each dataset in Table 13.

At the same time, it also indicates a potential for compression. Indeed, after analyzing the original office dataset with a size of 81 GB, we observe that many instances are repeated. Therefore, we introduce a preprocessing step before training, where we group all equal instances and keep a count of how many times they appear. These counts are used as weights in the later learning stage, which acts as a lossless compression mechanism. This way, we reduce the dataset to approximately 600 MB, which allows us to train models much faster and with significantly lower computational resources without sacrificing classification performance.

4.3.5.2 *Car*

The scheme achieves its lowest EER around 3.5% in the car scenario.

For the car scenario, we obtain an EER* of 0.115, with the classifier relying almost evenly on all the features to make the predictions. The individual subscenarios achieve even lower EERs, showing rates of 0.034 (parked), 0.08 (highway), and 0.081 (city). This performance correlates with the percentage of ambiguous instances—subscenarios with more ambiguous instances obtaining higher error rates. The low error rate of the parked subscenario is likely related to the use of temperature sensors, which capture

Table 13: Classification results for Shrestha et al.

Scenario	Model	EER	AUC	Acc.	Amb.
Car	DRF	0.115	0.960	88.5%	8.8%
– City	DRF	0.081*	0.977	91.9%	5.5%
– Highway	DRF	0.08	0.979	91.9%	6.8%
– Parked	DRF	0.034*	0.995	96.5%	2.2%
Office	DRF	0.247*	0.834	75.2%	25.5%
– Night	DRF	0.155*	0.911	84.4%	15.5%
– Weekday	DRF	0.271*	0.824	72.9%	25.5%
– Weekend	GBM	0.148*	0.928	85.1%	14.1%
Mob/het	DRF	0.141*	0.942	85.9%	12.2%

the different rates of heat dissipation of the cars after they are parked. When aiming for a specific FAR, the differences between the subscenarios are maintained, with the parked subscenario showing consistently lower FRRs (cf. Figure 17a).

Applying the car's model to office and mob/het data results in significantly worse performance.

The model is not very robust, showing significantly increased error rates when applied to the office or mob/het dataset (i.e., FAR 0.342, FRR 0.503 for office, 0.276 / 0.717 for mob/het). Similarly, applying models specific to one subscenario to another increases the error rates by at least 32 percentage points.

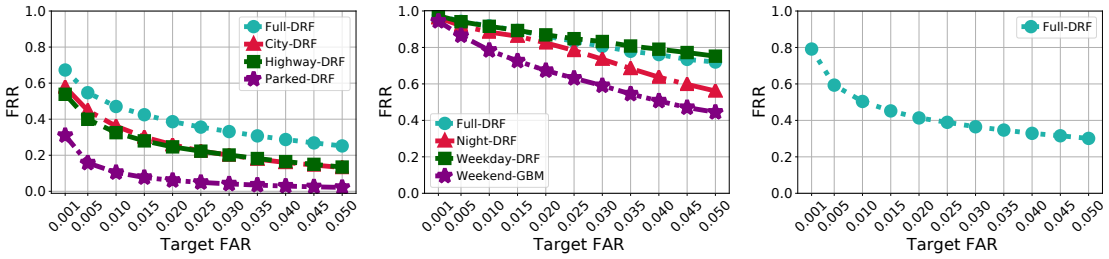
4.3.5.3 *Office*

The EERs in the office are universally above 15%.

In the office, the classifier heavily relies on temperate features to make predictions.

Here, the classifier reaches an EER* of 0.247, showing a significantly lower performance than in the car scenario. It relies strongly on the temperature differences to make the predictions. Such a focus on a feature with a low range of potential values may make the classifier more vulnerable to active attacks, and thus is undesirable. The low performance is mirrored in the subscenarios, with error rates of 0.148 (weekend) to 0.271 (weekday), which is also borne out in the high FRRs when aiming for a specific FAR (cf. Figure 17b). Once again, higher percentages of ambiguous instances translate to higher error rates. We also observe that the percentage of ambiguous instances grows with the size of the dataset. This is to be expected, as a larger dataset has a higher chance of obtaining these instances, as the range of potential values is limited.

When the office model is used on the other two datasets, the error rates increase significantly (i.e., FAR 0.186, FRR 0.693 for car, 0.184 / 0.787 for mob/het), showing that the model is not robust to being used in different environments. Out of the three subscenario models, the weekday model is the most robust, with error rate increasing below 6 percentage points when applied to different subscenario datasets. However, applying the night model to weekday data increases the error rate by 35 percentage



(a) Car (b) Office (c) Mob/het

Figure 17: FRRs with target FARs for Shrestha et al.

points, and the weekend model increases its error rate by over 43 percentage points with the weekday dataset. This shows that the robustness is limited.

4.3.5.4 *Mob/het*

Due to a lack of humidity and temperature sensors, the laptops and robot are not included in the evaluation. The classifier reaches an EER* of 0.141 (cf. Table 13), relying primarily on the altitude readings, with a lower importance given to temperature and humidity. A closer investigation reveals that the phones' barometric pressure readings deviate from those of the watches and SensorTags, showing an offset of approximately 2 hPa. The temperature and humidity readings vary significantly, with the position of the device (e.g., smartphone in a pocket, SensorTag on top of a screen) having a much larger influence than the room they operate in. The error rate is likely related to this challenging environment as well as the number of ambiguous instances, which make up 12.2% of the dataset. When optimizing for a low FAR, the resulting FRR are at least 0.3 (cf. Figure 17c).

In the mob/het scenario, the most important features are based on barometric pressure.

Applying the model to the car dataset results in notably increased error rates (i.e., FAR 0.302, FRR 0.672). The office dataset gives similar error rates (i.e., FAR 0.306, FRR 0.606), showing limited robustness of the model in different environments.

4.3.5.5 *Conclusion*

Overall, the scheme by Shrestha et al. cannot reliably separate colocated from non-colocated devices in most scenarios. This is in stark contrast to the error rates reported by the authors, who obtain an FAR and FRR of 0.0581 and 0.0296, respectively. This deviation can partially be explained by the lack of precision gas features in our datasets, reducing the number of dimensions the models can discriminate on. Another explanation is the more challenging environment our data is collected in—the authors collect their data in widely spaced locations at different times of day, and their non-colocated class consists of pairings between different locations. This also explains the high dis-

The ZIA scheme by Shrestha et al. shows much higher error rates in our scenarios than originally reported.

These worse results might have been due to the lack of precision gas sensor data.

criminative power of the altitude readings reported by the authors, indicating that the scheme will likely have a much better performance if only coarse colocation is required.

The number of ambiguous instances indicates the weakness of using environmental sensor data with a limited range of possible values.

The high number of ambiguous instances shows that the scheme would benefit from incorporating additional sensors to improve its discriminative power. Additionally, features tracking the change in values over time may be a more promising approach, as our dataset shows that these changes are more consistent between devices in the same room than the sensor readings themselves. The results also demonstrate that even if high classification performance can be reached, it is still highly specific to the environment it is collected in and does not transfer well into other environments, namely, the robustness of the scheme is limited.

4.4 DISCUSSION

In this section, we discuss the implications of the obtained results, the limitations of our method, and avenues for future work.

The schemes by Karapanos et al. and Truong et al. demonstrate lowest EERs in our scenarios.

Performance Comparison. Our results, summarized in Table 6, show that the scheme by Truong et al. obtains the best EERs in the office (0.069) and mob/het (0.123) scenarios, while the scheme by Karapanos et al. achieves a significantly lower error rate (0.006) in the car scenario. This indicates that depending on the use case, both are solid choices. We also observe a large variation in performance on different subscenarios, ranging from perfect accuracy (i.e., Karapanos et al., $t = 120$, highway) to significantly degraded performance compared to using the full dataset (i.e., Schürmann and Sigg, $t = 120$, parked), illustrating the importance of fine-grained test scenarios.

All schemes show limited adaptiveness in cases of low-entropy context.

Adaptiveness. Some schemes struggle to adapt to times of low-ambient activity like the night, where we observe a lower separation between colocated and non-colocated devices. These times need to be taken into account when designing a scheme intended for continuous operation. Karapanos et al. and Miettinen et al. introduce measures to reduce the impact of these times by dynamically discarding samples with low-ambient activity [181] or high predictability [243], trading off availability for security.

Schemes' parameters such as thresholds cannot be easily transferred from one environment to another.

Robustness. Even if they can operate in environments with low activity, most schemes suffer from a lack of robustness, namely, parameters that are optimal for one scenario do not give good performance in another. Some schemes can achieve a certain degree of robustness for specific parameters (e.g., interval sizes), most notably that by Karapanos et al., but no scheme is robust with all parameters. The same trend holds when exchanging parameters or models between different subscenarios, like day and night, especially if they have significantly different ambient activity levels. This illustrates the importance of testing schemes in a wide variety of settings. We urge researchers to pay special attention to robustness, facilitating the use of ZIS schemes in a wide variety of different and sometimes unexpected environments, where IoT devices are deployed.

Heterogeneity. Even if schemes can provide good performance in settings with homogeneous devices (i.e., the same hardware), they may still fail when encountering devices

with different characteristics. These challenges have also been encountered in other research fields such as participatory sensing. Examples from our dataset include microphones with varying sensitivity and frequency response [182, 216, 227], which leads to lower correlations, or incorrectly calibrated sensors (e.g., air pressure) measuring with a fixed offset from one another. In addition, the way a device is carried influences the observed sensor data [246]. Schemes need to be able to still provide good results under these conditions if they are intended for use cases, where the used hardware is not carefully controlled by a single party, and should be tested with heterogeneous devices.

Device heterogeneity lowers performance of most schemes.

Colocation Definition. Many schemes do not explicitly state their colocation definition. It is often unclear if they are intended to distinguish personal workspaces inside an office, different rooms, or parts of a city, making it difficult to identify security guarantees that the schemes provide in any specific situation. This hinders a fair evaluation and comparison of these schemes, and makes it hard to determine if our results impact its designated use case. Authors should explicitly define what their scheme considers (non-)colocated to allow for fair comparisons.

Limitations. Technical issues during the recording led to data loss for some features, especially the Wi-Fi captures, which stopped working on some devices. In the mob/het scenario, three devices stopped the data collection before the eight-hour countdown, resulting in partial loss of audio and sensor data. This reduces the amount of available data for the evaluation of features based on these modalities. In the same way, the *SensorTag* platform occasionally delivered incorrect readings for the luminosity, which we detected and excluded. Appendix B details practical challenges we faced in our study such as building a reliable data collection platform, processing the collected data, and releasing it as well as reproducing the existing ZIS schemes from scratch.

Our goal was to compare the different ZIS schemes in a fair manner and as specified by their original authors. While we attempted to stay as close to the published version of the scheme as possible, in some cases, minor changes had to be made to parameters such as interval length or sampling rate. These deviations are noted in Appendix A, and their influence on the results should be negligible. We did not attempt to optimize any parameters aside from interval length and the power threshold of Karapanos et al. for the mob/het scenario in our dataset, so it is possible that some schemes could perform better when they are instantiated with different parameters.

We do not attempt to optimize schemes' parameters, which may improve their performance.

We also note that our scenarios are challenging, as they include devices in isolated positions (e.g., glove compartment, cupboard, pocket), low separation between adjacent offices, and two cars that travel next to each other for extended amounts of time. This is intentional to be able to investigate the performance of the schemes in challenging situations, as consumer IoT deployments rarely follow best practices for deployment, facilitating zero-interaction security. Also, we do not include any scenarios in busy areas like shopping malls, which may show different behavior due to the higher environmental variations, as we choose to focus on the likely application domain of zero-interaction pairing and authentication—consumer applications. Additionally,

obtaining approval and informed consent for a long-term data collection in a public place would have been infeasible in our jurisdiction.

A ZIS scheme for a smart building would pose a different set of challenges.

Future Work. Our chosen scenarios only cover a subset of interesting IoT environments. Other scenarios may pose different challenges for the schemes. For example, in a *smart building* scenario, an ideal ZIS scheme would have to be able to pair or authenticate all devices within the same building while excluding adjacent buildings. In a *café* scenario, schemes would need to be able to distinguish individual tables. Also, we do not include any scenarios where devices operate in environments without any humans for extended amounts of time (e.g., automated factory work floors or storage units), which could pose challenges to many schemes due to the potentially low variation in the context information or different noise characteristics.

The collection of additional datasets will assist efforts to create more adaptive and robust schemes and to understand the limitations of existing ones. Another avenue for future work is the robustness to adversarial settings, where part of the context information can be controlled by an active adversary (e.g., by injecting sound).

4.5 SUMMARY

In this chapter, we reproduce and evaluate five zero-interaction pairing (ZIP) and zero-interaction authentication (ZIA) schemes in three realistic scenarios: (1) connected car, (2) smart office, and (3) smart office with mobile heterogeneous devices, posing different challenges in aspects like environmental noise, context leakage, and times of low-ambient activity. We see that none of the reproduced schemes can perfectly separate devices in all scenarios. The schemes by Karapanos et al. [181] and Truong et al. [364] show promising results, but no scheme reliably outperforms the others in all scenarios. The obtained error rates indicate that zero-interaction security (ZIS) should not be used as the only access control factor, as even a False Acceptance Rate (FAR) of 1% can be prohibitively high for real-world applications. In fact, Karapanos et al. explicitly propose their ZIA scheme as a usable second authentication factor [181] instead of a stand-alone solution.

We find that a good average-case separation of context features aggregated over the whole data does not imply high pairing/authentication performance on individual samples. Thus, ZIP and ZIA schemes should be evaluated in terms of their error rates, both in average cases and individual subscenarios, to get a realistic impression of their performance. Similarly, the evaluation should be performed using a set of heterogeneous devices in real-world environments, testing the limits of the schemes.

We summarize contributions of this chapter.

Our evaluation reveals that in many cases, features based on ambient audio perform best. However, researchers need to take the privacy implications of using audio recordings into account, as this may not be acceptable in some environments like hospitals. In addition, the computational costs of processing have to be considered, as expensive audio processing may be infeasible on resource-constrained devices. Furthermore, we

find that devices with differing microphone characteristics can significantly reduce the performance of sound-based schemes. Thus, we encourage researchers to continue investigating the possibility of using more power-efficient features based on low-power sensors. Also, we observe that instead of using the absolute difference between sensor readings, trends over time may be a more reliable colocation indicator.

We find that the robustness and adaptiveness of many schemes vary dramatically for different scenarios. ZIS schemes should explicitly state which environments they are designed for. Additionally, they should support robustness and adaptiveness, potentially by automatically adapting their internal parameters to their environment, and should be evaluated on data from different scenarios and devices.

Finally, we release the first extensible open-source toolkit [97] for researching zero-interaction security, containing reference implementations of the reproduced schemes, the audio recordings of the mob/het scenario, and over 1 billion samples of labeled sensor data. We also release the data generated by our evaluation to facilitate the reproduction of our results and provide a common benchmarking baseline for future schemes.

5

FASTER AND MORE SECURE ZERO-INTERACTION PAIRING

After having investigated zero-interaction pairing (ZIP) schemes under realistic conditions, revealing their shortcomings (cf. Chapter 4), we improve on ZIP with respect to security and pairing time in this chapter.

To date, a number of ZIP schemes utilizing various sensor modalities to capture context have been proposed [144, 145, 210, 243, 245, 309, 310]. These state-of-the-art schemes have three major limitations: (1) prolonged pairing time, (2) vulnerability to offline attacks, and (3) susceptibility to attacks caused by predictable context (e.g., replay). First, state-of-the-art ZIP schemes suffer from *prolonged pairing time* requiring minutes and hours of context data to establish a shared key [98, 144]. This happens because they use a cryptographic primitive called *fuzzy commitments* [175], where the entropy of a shared key *is equal to* the entropy of fingerprint bits, input to the protocol. Thus, these ZIP schemes need to obtain at least 128 bits of entropy from context to ensure that a shared key provides adequate security [26]. Obtaining these bits takes a prolonged time because many contexts change slowly. Second, state-of-the-art ZIP schemes are by design vulnerable to *offline attacks*, namely an adversary can mount a brute-force attack on a shared key by repeatably guessing the used fingerprints. These schemes can *only* withstand offline attacks if they use (1) long fingerprints (i.e., >128 bits) of (2) high entropy. However, recent works find severe entropy biases (e.g., bit patterns) in fingerprints of state-of-the-art ZIP schemes [36, 98], exposing them to offline attacks. Third, state-of-the-art ZIP schemes are susceptible to context replay, inference, or monitoring attacks due to *predictable context* [36, 98, 145]. Frequently, context becomes predictable because it relies on a single sensor modality (e.g., acceleration). Thus, an adversary can obtain similar context in comparable environments, use better hardware, or employ video analysis.

State-of-the-art ZIP schemes suffer from three limitations, hindering their real-world deployment.

The above three limitations impair the practicality and security of ZIP schemes, hindering their real-world deployment. To overcome these limitations, we propose *FastZIP*, a novel ZIP scheme that achieves shorter pairing time and improved security by addressing the following two challenges. Figure 18 compares *FastZIP* and state-of-the-art ZIP schemes in terms of paring time[14], resistance to offline attacks, and the number of common sensors required for pairing.

First, to shorten pairing time, we need to reduce the number of fingerprint bits while being robust to offline attacks. This is challenging because fewer bits means less entropy in a shared key, easing an offline attack. To address this challenge, we adapt a recently introduced *Fuzzy Password-Authenticated Key Exchange (fPAKE)* protocol [75].

14 We use the shortest pairing time reported in the original publication for each scheme.

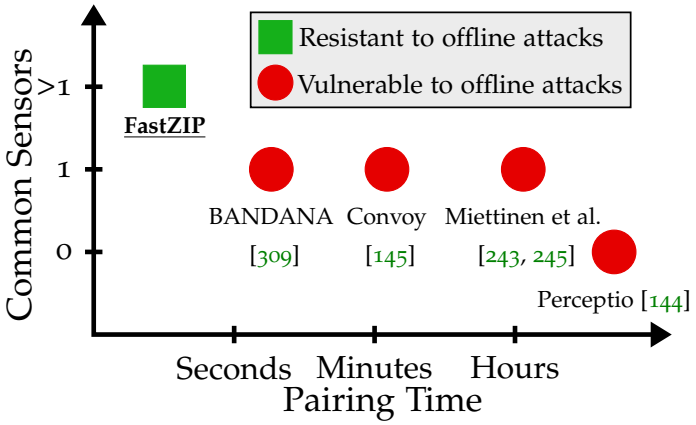


Figure 18: Design space of *FastZIP*: it provides shorter pairing time and improved security compared to state-of-the-art ZIP schemes utilizing more common sensors.

The fPAKE protocol allows preventing offline attacks.

fPAKE establishes a shared key from low-entropy secrets (e.g., short passwords) and is resistant to offline attacks. While fPAKE is an existing protocol, adapting it to ZIP schemes is not trivial. Specifically, we find that fPAKE protection against offline attacks is not always guaranteed in realistic ZIP settings, namely when colocated devices do not yield highly similar fingerprints from context. Thus, we analyze how to set fPAKE parameters to withstand offline attacks even in such settings (cf. Section 5.3). To the best of our knowledge, we are the first to implement fPAKE and demonstrate that it shortens pairing time and improves security of ZIP schemes using real-world data.

We are the first to apply sensor fusion in ZIP.

Second, to defend against predictable context attacks (e.g., replay attacks), we propose a simple form of *sensor fusion* by concatenating fingerprints derived from different sensors, each capturing distinct ambient activity. Applying sensor fusion is not straightforward, as we require a generic method to extract fingerprint bits of sufficient entropy from heterogeneous sensor signals (cf. Section 5.3). Existing methods rely on scenario-specific characteristics of sensor signals (e.g., peak occurrence), thus cannot be directly reused, and they often produce fingerprints with entropy biases [36, 98]. We demonstrate that sensor fusion not only prevents predictable context attacks (e.g., replay) but also assists fPAKE in shortening paring time, as we obtain more bits from context, accumulating entropy faster. Sensor fusion is feasible because smart devices have multiple sensor modalities often *integrated* into one chip, for example, an inertial measurement unit (IMU) contains an accelerometer, gyroscope, and magnetometer [384], while a camera has light and RGB sensors, and a wireless chipset hosts both Wi-Fi and Bluetooth [294].

We evaluate FastZIP for in-car pairing.

We demonstrate the advantages of *FastZIP* by evaluating a novel use case of *intra-car device pairing* (cf. Section 6.5), which is inspired by the growing number of smart devices inside modern cars. For example, the increasing popularity of carsharing and self-driving rides urges the need to pair multiple user devices (e.g., smartphone, earbuds)

with infotainment systems of different cars to enable such services as customized driving experience [151, 298]. Furthermore, electronic control units (ECUs) require pairing with wireless third-party components (e.g., tire pressure monitor) to enable travel efficiency and safety [49, 52, 363]. In both examples, the growing number of devices hinder manual pairing, requiring pairing solutions without user intervention. Despite focusing on in-car pairing, we show how the design of *FastZIP* can generalize to other ZIP use cases (e.g., smart home) to improve pairing time and security in Section 5.6.

FastZIP achieves shorter pairing time and higher security than state-of-the-art ZIP schemes.

Through our real-world experiments, we demonstrate the feasibility of leveraging the context of a moving car to pair devices inside it. Such context is affected by road and traffic conditions, car characteristics such as suspension, and driving patterns, and it can be captured by accelerometer, gyroscope, and barometer sensors [48, 145, 297, 371] that are ubiquitous in user devices (e.g., smartphone) and modern cars. We evaluate *FastZIP* by collecting sensor data from four cars driven over 800 km on different road types, including urban, rural, and highways. In our evaluation, we assume that pairing devices can start measuring context simultaneously by receiving a broadcast command from the car's infotainment system, which is not compromised. *FastZIP* achieves up to *three times* faster pairing compared to state-of-the-art ZIP schemes, shows error rates *below 0.5%* in the presence of a powerful adversary, and runs efficiently on off-the-shelf Internet of Things (IoT) devices. In summary, we make the following contributions:

- We design *FastZIP*, a novel ZIP scheme utilizing fPAKE and sensor fusion to reduce pairing time and improve security.
- We implement *FastZIP* for intra-car device pairing and evaluate it by collecting real-world driving data, demonstrating the effectiveness of *FastZIP*.
- We publicly release the collected data, source code of our evaluation stack, and the first implementation of fPAKE.

5.1 BACKGROUND

We first explain the working principle and shortcomings of fuzzy commitments—a cryptographic protocol used by state-of-the-art ZIP schemes to share a secret key. Then, we detail the fPAKE protocol [75], addressing these shortcomings, that we utilize in *FastZIP*.

ZIP Based on Fuzzy Commitments. Prior work on ZIP relies on fuzzy commitments or vaults [174, 175] to exchange a key K between two devices holding similar fingerprints f, f' [144, 145, 210, 243, 245, 309, 310]. Specifically, Device A chooses a 128-bit key K and sends a commitment $c \leftarrow \mathsf{ECC.Encode}(K) \oplus f$ to Device B, which can recover $K \leftarrow \mathsf{ECC.Decode}(c \oplus f')$ if the fingerprint mismatch $f' \oplus f$ is within the error correction capability of the error correction code (ECC). While conceptually simple, this approach has two disadvantages in the case of ZIP. First, it inherently requires fingerprints f, f' to

Offline attacks are by design mountable on ZIP schemes based on fuzzy commitments.

be at least 140 bits, since they are XORed to an expanded encoding of the 128-bit key[15]. Second, an eavesdropping adversary can capture the commitment c and try decoding it with arbitrarily many fingerprint guesses to obtain the key K. This constitutes an *offline attack* on K, which can only be defended against if the fingerprints have high entropy (i.e., they are hard to guess). In practice, state-of-the-art ZIP schemes already require multiple minutes or even hours to obtain fingerprints >128 bits from context [144, 145, 243, 245, 309]. Even worse, an in-depth entropy analysis reveals that fingerprints of these schemes contain bit patterns or predictable distributions of 0- and 1-bits [36, 98]. Thus, an adversary can more easily guess the fingerprints, exposing state-of-the-art schemes to offline attacks.

The fPAKE protocol has an interactive entropy amplification phase, preventing a passive adversary from mounting an offline attack.

fPAKE Protocol. fPAKE used by *FastZIP* allows reducing the number of required fingerprint bits, hence shortening pairing time, while providing resilience to offline attacks. In essence, fPAKE is also a fuzzy commitment, but instead of creating the commitment from fingerprint f, fPAKE adds an interactive *entropy amplification* phase that turns fingerprints f, f' into high entropy keys $\mathbf{k}, \mathbf{k}'$ with a similar mismatch pattern as f, f' (cf. Figure 19). In entropy amplification, fPAKE leverages an established cryptographic primitive called password-authenticated key exchange (PAKE) [22], which allows two parties to exchange a secure (i.e., 128-bit and uniform) key from a shared short string, such as a password, or even a bit. The PAKE protocol is secure against offline attacks, meaning that the best possible adversarial strategy is to guess the short string and engage in the key exchange. In fPAKE, PAKE is used to amplify the entropy of individual fingerprint bits as follows: Devices A and B run multiple standard PAKE [22] protocols on the individual fingerprint bits in parallel, obtaining key vectors $\mathbf{k}$ and $\mathbf{k}'$, where $\mathbf{k}_i = \mathbf{k}'_i$ if the i-th fingerprint bits matched. Next, Device A chooses a 128-bit secret s and sends a fuzzy commitment $\mathsf{com} \leftarrow \mathsf{ECC.Encode}(s) \oplus \mathbf{k}$ to Device B, which decodes it with $\mathbf{k}'$. Afterwards, Devices A and B confirm to each other that they know s by sending each other hash values $H(s\|0)$ and $H(s'\|1)$ of the secret. Finally, if the hash check succeeds, Devices A and B derive a shared key k_{AB} from s using a key derivation function (KDF).

With fPAKE much shorter fingerprints can be used to protect against offline attack.

Advantages of fPAKE in ZIP. By using high entropy keys in the fuzzy commitment phase (cf. Figure 19), fPAKE prevents an eavesdropping adversary from mounting an offline attack because the adversary only knows $c \oplus \mathbf{k}$, which is a secure encryption of c under a (by the guarantees of PAKE) secure key $\mathbf{k}$. Moreover, even an active adversary (e.g., malicious Device B) can try *exactly one* fingerprint guess f' as input to the interactive entropy amplification phase. If that one guess is too far (i.e., f and f' are dissimilar), even the unbounded adversary cannot recover s, making the offline attack *impossible*. Otherwise, if the guess is "close enough" (cf. Section 5.3), the active adversary can attempt an offline attack. However, Device A waits for the key confirmation h' within a short timeout (e.g., a few seconds), allowing the adversary only this amount

15 The 140 bits are for an expanded encoding allowing up to 10% mismatching fingerprint bits. To allow for 30% mismatch in fingerprints, 205 bits are required.

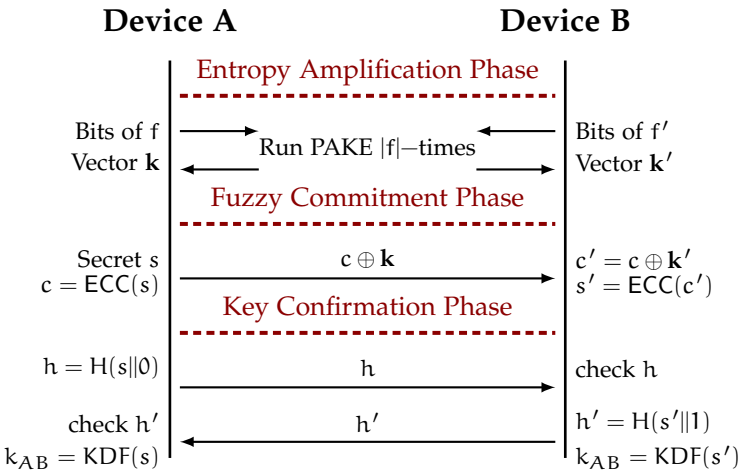


Figure 19: Detailed flow diagram of the fPAKE protocol.

of time to perform the attack. We note that for standard fuzzy commitments the key confirmation upon timeout cannot similarly limit the offline attack, as the adversary does not participate in the protocol.

In Section 5.3, we demonstrate how to leverage the strong security of fPAKE against offline attacks to reduce the required fingerprint sizes *well below* 128 bits in many settings. Also, we empirically show that additional communication overhead of fPAKE (i.e., entropy amplification phase) is negligible compared to up to three times faster pairing time when using fPAKE instead of fuzzy commitments.

5.2 SYSTEM AND THREAT MODELS

We introduce our system model, describing the goal, requirements, and assumptions of *FastZIP*, and our threat model, detailing adversary's goals and capabilities.

System Model. The main *goal* of *FastZIP* is to establish a shared secret key between colocated devices within a trusted boundary (e.g., inside a car) based on the perceived context. We design *FastZIP* to fulfill the following *requirements*: (1) be free of user interaction during pairing (*usability*), (2) have short pairing time (*practicality*), and (3) work on commodity devices equipped with off-the-shelf sensors (*deployability*). To achieve the main goal while satisfying the requirements, we make the following *assumptions*: (1) devices running *FastZIP* do not have any pre-shared secrets, nor any jointly trusted third party, (2) they communicate using Wi-Fi or Bluetooth, and share a common set of sensors such as an accelerometer, gyroscope, and barometer, and (3) they begin measuring context upon receiving the "start" command from the car's infotainment system, which is assumed to be non-compromised.

Threat Model. We consider an adversary whose *goal* is to establish a shared secret key

with a legitimate device while residing outside the trusted boundary. In particular, the adversary attempts to either *impersonate* one of the legitimate devices or acts as a *man-in-the-middle* between a pair of devices. The adversary can neither compromise legitimate devices nor break cryptographic primitives, however, they fully control a wireless channel, are equipped with the same sensing hardware as legitimate devices, and have four attack capabilities. In an *injection attack*, the adversary attempts to pair with legitimate devices using self-chosen context readings. In a *replay attack*, the adversary replays precollected context readings. In a *similar-context attack*, the adversary tries to actively match their context with legitimate devices. In the intra-car pairing, the adversary launching a replay attack replays the precollected context data from a route driven by a victim car carrying legitimate devices, while in a similar-context attack, they actively follow the victim car to capture similar context such as the road bumpiness. The first three attacks require the adversary to participate in the pairing protocol, while in an *offline attack*, they record a successful pairing session and try to compute a shared key from it by repeatedly guessing fingerprints used by legitimate devices.

We consider four types of attacks against FastZIP.

5.3 SYSTEM DESIGN

We present the architecture of *FastZIP*, describing its modules: *activity filter*, *quantization*, and *key exchange*.

System Overview. The main goal of *FastZIP* is to share a symmetric key between a pair of devices utilizing their context. In a moving car the context encompasses road turns, bumpiness, and speed changes [48, 297, 371], and it can be perceived by accelerometer, gyroscope, and barometer sensors that are ubiquitous in smart devices. *FastZIP* works as follows (cf. Figure 20): Devices A and B capture their context using a set of common sensors. The resulting sensor readings are input to the *activity filter* to discard low-entropy context, which can be predicted by an adversary. Afterwards, the filtered sensor readings are input to the *quantization* translating them into a sequence of fingerprint bits. Each device constructs its fingerprint by concatenating sub-fingerprints derived from different sensors (i.e., sensor fusion). These fingerprints are input to the *fPAKE* protocol, which outputs a shared symmetric key if the fingerprints have a sufficient number of similar bits.

Activity Filter. The security of any ZIP scheme relies on the unpredictability of context from outside a trusted boundary (e.g., car interior). The low-entropy context undermines security of ZIP schemes, allowing an adversary to guess fingerprints derived from it [98]. *FastZIP* utilizes the *activity filter* to ensure that fingerprints are obtained from context data with sufficient entropy.

Our activity filter is based on three generic signal metrics.

To estimate the entropy of a sensor signal, we analyze its strength relative to noise and variation. For that, we employ three metrics: *average power*, *signal-to-noise ratio (SNR)*, and the *number of prominent peaks*, which are used to characterize signal's quality [181, 210, 371]. The average power and SNR are applicable to all sensors, while prominent

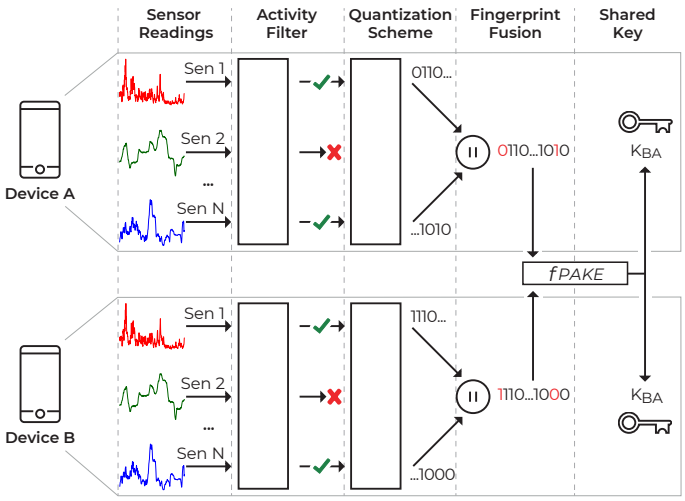


Figure 20: System overview. *FastZIP* takes as input a set of sensor readings from two devices. The readings are quantized to *similar* fingerprints and afterwards input to the fPAKE protocol to share a symmetric key, K_{BA}.

peaks is a complementary metric for rapidly changing modalities (e.g., acceleration), ensuring their sufficient variation. We compute the average power P_s in dB of a discrete sensor signal $s(t)$ as follows:

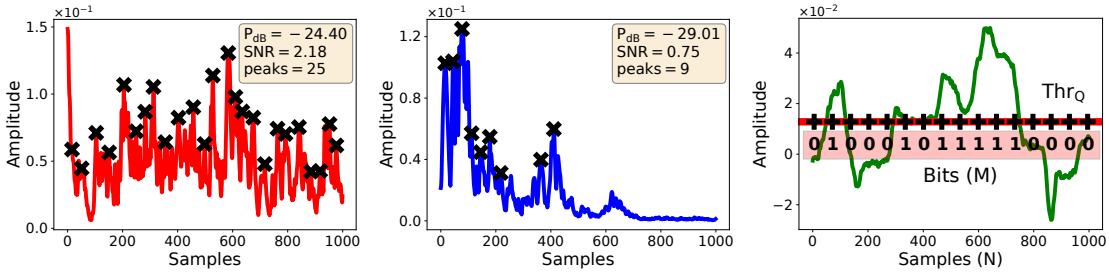
$$P_{s(dB)} = 10 \cdot \log_{10}\left(\frac{1}{T}\sum_{t=1}^{T} s^2(t)\right)$$

We cannot compute SNR as the ratio of signal to noise power, as we do not have the estimate of the latter; estimating noise power will impose additional processing overhead. Thus, we use an alternative definition of SNR as the ratio of mean to standard deviation of a signal: $\text{SNR} = \frac{\mu}{\sigma}$ [39]. To find prominent peaks in a signal, we count peaks that have sufficient height relative to the highest peak, while being within minimum distance Δ_P from each other. Figure 21a and Figure 21b show activity filter metrics computed for two acceleration signals (prominent peaks marked with ✖). We see that the former signal captures continuous activity, exhibiting sufficient entropy, while the latter signal contains noise in its right half, which is reflected in the computed metrics.

Applying the activity filter on a stream of sensor data using the sliding window approach improves availability of FastZIP.

After computing the metrics, we check them against fixed thresholds to discard signals with insufficient entropy, which, in turn, may reduce availability of *FastZIP*. To avoid this, we apply the activity filter on a continuous stream of sensor data using an overlapping sliding window. Thus, parts of the signal containing sufficient entropy are considered in both preceding and following timeslots, making it possible to retain them.

Quantization. *Quantization* translates a sensor signal (e.g., acceleration) to fingerprint bits used in a key agreement protocol. To ensure security, the produced fingerprints



(a) Signal with sufficient entropy (b) Signal with insufficient entropy (c) Signal quantized to bits

Figure 21: Activity filter applied to acceleration signals (a) and (b); Quantization applied to a gyroscope signal (c).

must be sufficiently unpredictable. Prior work [36, 98] finds that quantization methods of state-of-the-art ZIP schemes generate fingerprints with patterns (e.g., containing more 0-bits). We design the quantization of *FastZIP* with three goals in mind: it must (1) generate fingerprints that are random, (2) apply across various sensors, and (3) reveal minimum information about the input sensor signal. The last goal seeks to reduce adversary's knowledge about the input sensor signal leaked by quantization (e.g., signal range [132]).

We quantize short sensor signals of several seconds, producing fingerprints of a few dozen bits. However, our method generalizes to longer signals and fingerprints. The advantage of using short input signals is twofold: (1) it forces an adversary to guess context captured by a sensor within a precision of a few seconds, (2) it requires less processing, improving the runtime performance of *FastZIP*. Our quantization takes a sensor signal S of length N samples as input and outputs a fingerprint f of M bits (cf. Figure 21c). Specifically, we first find a quantization threshold Thr_Q (solid red line in Figure 21c) that splits the signal horizontally into upper and lower parts. The threshold is computed from the median of the signal, ensuring that the same number of samples lie above and below it. This way of selecting Thr_Q leads to improved randomness of the fingerprints (cf. Section 5.5.5) and is efficient to implement. Second, we place the quantization points $p_1, \ldots, p_M$ (marked as ✚ in Figure 21c) equidistantly onto the threshold line within distance $\Delta_Q = \lceil \frac{N}{M} \rceil + \varepsilon$ from each other, covering the signal completely. The number of quantization points and Δ_Q are public parameters customized for each sensor modality (cf. Section 5.5.1). Using public parameters has the advantage of (1) fewer communication rounds, as they do not need to be exchanged during the pairing protocol, and (2) not leaking information about a specific input signal, as the parameters are derived from the class of signals of the same modality, and

Our quantization does not rely on scenario-specific characteristics of sensor signals, thus can be extended to other use cases.

thus are general. Third, we obtain a bit in the fingerprint $f(i)$ by comparing a signal value at the quantization point $S(p_i)$ with the quantization threshold Thr_Q:

$$f(i) = \begin{cases} 1, & S(p_i) > Thr_Q \\ 0, & \text{otherwise} \end{cases}$$

Key Exchange. The main challenge of adapting fPAKE for the use in *FastZIP* is to find which minimum fingerprint sizes[16] are required to securely exchange a 128-bit key, protecting against offline attacks. With fPAKE we can choose *arbitrarily small* fingerprints, which are then amplified to match the size of the encoded secret (cf. Figure 19). By reducing the number of required fingerprint bits, we are able to shorten pairing time, while providing sufficient security.

Before calculating the required fingerprint sizes, we determine sufficient security levels for *FastZIP*. Specifically, we consider two levels: (1) the minimal probability P with which an offline attack is eliminated and (2) the average complexity C of an offline attack. We set $P = 1 - 2^{-20}$, namely an adversary *actively participating* in a million pairing sessions can mount an offline attack in at most one of them (without even learning which one). This level of security is considered adequate for ZIP [245]. We set $C = 2^{60}$, demanding that an offline attack has an average complexity of at least 2^{60} AES decryptions. We consider this complexity sufficient given that attack time is limited to few seconds due to the key confirmation timeouts that we augment the fPAKE protocol with (cf. Figure 19).

We consider two types of security level against offline attacks for FastZIP.

We explain how to calculate the required fingerprint sizes, satisfying our chosen security levels using a 95% similarity threshold as an example. The similarity threshold defines the amount of common bits in two fingerprints needed to obtain a shared secret key. One might think to set the fingerprint size $|f|$ such that the probability of guessing at least $0.95 \cdot |f|$ bits correctly is smaller than 2^{-20}, since the key should be undecodable otherwise. Unfortunately, this is not true: an ECC (used in fPAKE) correcting 5% mismatch between the fingerprints *leaks some information about the encoded secret* until up to $2 \cdot 5\% = 10\%$ mismatch. Thus, an active adversary guessing less than 90% of the fingerprint correctly learns nothing about the secret. However, if the guessed fingerprint is "close enough" (i.e., 90–95% of the bits), the adversary cannot immediately decode the secret but obtains an *ambiguous encoding* from which the secret can be brute-forced. Taking this "security gap" inherent to ECCs into account, we set $|f|$ such that the probability $\sum_{i=m}^{n} \binom{n}{i} / 2^n$ of guessing $m = (2 \cdot Thr - 1) \cdot |f|$ out of $n = |f|$ bits correctly is smaller than 2^{-20}, where Thr is the target similarity threshold.

We take entropy loss due to error correction into account when calculating the minimal probability of an offline attack.

We note that $|f|$ goes to infinity when Thr approaches 75%, since $2 \cdot 25\% = 50\%$ of a random bitstring is easy to guess. Thus, full protection against offline attacks with probability at least $1 - 2^{-20}$ is only possible for thresholds over 75%, requiring

16 Here, we assume fingerprints to be uniformly random bitstrings; entropy biases will increase fingerprint sizes (cf. Section 5.5.5).

Table 14: Offline protection and brute-force complexity for *FastZIP* computing 128-bit keys for different choices of similarity thresholds and fingerprint sizes. Gray boxes mark sufficient security levels. T is described in text.

Similarity Threshold	Fingerprint Bits	Offline Attack $1-P$	Brute-force Complexity C
95%	40	$< 2^{-23}$	$\approx 2^{37}T$
90%	60	$< 2^{-20}$	$\approx 2^{32}T$
85%	80	$< 2^{-12}$	$\approx 2^{60}T$
80%	100	$< 2^{-7}$	$\approx 2^{63}T$
75%	120	≈ 1	$\approx 2^{64}T$
70%	140	≈ 1	$\approx 2^{60}T$

We choose the brute-force complexity of an offline attack considering that the attack is limited to a few seconds.

short fingerprints of 40–60 bits for thresholds above 90% (cf. Table 14). Below 90%, the security of *FastZIP* relies on our other security level measuring brute-force complexity of the offline attack. For estimating this complexity, we think of ECC encodings as consisting of $n = |f|$ parts which are correct or wrong depending on whether the corresponding fingerprint bit was correct or not. We use the following brute-force method to decode an ambiguous encoding: randomly guess which m parts of the encoding are correct, decode only them, and set the secret to be the first result that appears twice. Considering that we do not know how many parts i of the codeword are actually correct, the conditional probability of guessing m out of i correct parts in the n parts long encoding is given by the hypergeometric distribution as $i^{\underline{m}}/n^{\underline{m}}$, which finds its maximum at $i = (n-m)/2$. The complexity of the offline attack is thus lower bounded by $n^{\underline{m}}/(n-m)^{\underline{m}}T$. Here, T is the complexity of ECC.Decode, which is larger than the complexity of one AES decryption.

Table 14 shows the calculated fingerprint sizes providing sufficient security for *FastZIP* based on fPAKE for a range of similarity thresholds. In Section 5.6, we elaborate that these findings are generic, thus can be directly reused by other ZIP schemes.

5.4 INTRA-CAR DEVICE PAIRING

We present *intra-car device pairing*—an exemplary use case of *FastZIP* to pair devices inside a moving car. It enables novel vehicular applications such as pairing user devices for customized driving experience or pairing ECUs for travel efficiency [52, 298]. We first provide the case overview followed by implementation details.

Case Overview. There is a growing number of on-board smart devices in modern cars, including devices of drivers and passengers (e.g., smartphone, earbuds) as well as ECUs and infotainment systems [52, 167]. The prohibitive user effort to pair these devices,

many of which lack user interfaces, justifies the use of *FastZIP* for intra-car device pairing. *FastZIP* utilizes four sensor modalities to capture the context of a moving car: vertical and horizontal acceleration, gyroscope sky-axis, and barometer. Our review of prior work shows that acceleration of a moving car can be decomposed into *vertical* and *horizontal components*, with the former capturing road conditions (i.e., bumpiness), while the latter—driving patterns and traffic conditions (i.e., acceleration/deceleration) [48]. A *gyroscope* measures car's turns and steering directions [371], while a *barometer* captures altitude changes when a car moves along the road [297].

Intra-car device pairing enables innovative vehicular applications.

5.4.1 *Implementation*

Data Collection. We develop an Android app to collect accelerometer, gyroscope, and barometer data at fixed sampling rates (i.e., 100 Hz for accelerometer and gyroscope; 10 Hz for barometer). We convert accelerometer and gyroscope data to the world coordinates, eliminating the effect of device orientation. Before data collection, we perform an Network Time Protocol (NTP) update on smartphones, ensuring consistent data timestamps, which we use to synchronize the start of sensor recordings of colocated devices.

Data Processing. We process the collected sensor data before feeding it into the activity filter. Prior to any processing, we resample the data to the set sampling rates, eliminating the effect of *sampling rate instability* [342]. To decompose acceleration into vertical and horizontal components, we (1) remove the Earth's gravity from the accelerometer data applying a non-overlapping 5-second sliding window and (2) use the estimated Earth's gravity to perform the decomposition [48]. For the gyroscope data, transformed to the world coordinates by our app, we select a Z-axis that is perpendicular to the road surface. We convert the barometer data p to altitude h_{alt} in meters using a standard pressure-height formula [297]:

$$h_{alt} = 44330 \cdot \left(1 - \left(\frac{p}{1013.25}\right)^{\frac{1}{5.255}}\right)$$

After converting the sensor data to a required format, we perform signal smoothing and noise reduction in two steps: (1) applying them on the whole data and (2) on signals of several seconds, partitioning these data. To remove low-frequency noise and smooth the whole data without distorting it (e.g., keep peak locations), we use a Savitzky-Golay (SG) filter with a window length 3 and degree 2 polynomial. Afterwards, we apply a Gaussian filter with a sigma of 1.4 to reduce high-frequency noise.

We perform signal smoothing and noise reduction on the whole data and chunks of the data.

We use the same sequence of filters on sensor signals of several seconds: the SG filter has a window length 5 and degree 3 polynomial for finger-grained smoothing, while the Gaussian filter stays the same. For the acceleration signals, we afterwards apply an exponentially weighted moving average (EWMA) filter to smooth them further, while keeping their significant changes; the EWMA alpha is set to 0.16 and 0.2 for

vertical and horizontal acceleration, respectively. For the altitude signals, we perform mean subtraction before filtering. This helps to (1) remove offset between barometer sensors caused by hardware and temperature variation [99], (2) eliminate atmospheric pressure, accentuating altitude changes in the signal. We adapt filter parameters for signal smoothing and noise reduction from related work [48, 144, 210].

Activity Filter. The activity filter applies to a processed sensor signal of several seconds. We implement it by computing the average power and SNR for all modalities, and counting prominent peaks for vertical and horizontal acceleration (cf. Section 5.3). To pass the activity filter, a signal must have the average power, SNR, and optionally the number of prominent peaks higher than a predefined threshold, which we find empirically. The signal that passes the activity filter is input to the quantization, otherwise it is discarded.

Quantization. We implement quantization, converting a sensor signal to fingerprint bits, as described in Section 5.3. Its parameters (i.e., signal length, number of output bits) are set empirically for each modality (cf. Section 5.5.1). We compute the quantization threshold as a median of the sensor signal; for vertical and horizontal acceleration, we add small Δ to the median, reducing the effect of sensor noise on quantization. We concatenate bits quantized from the sensor signal with bits derived from other modalities likewise before inputting them as one fingerprint to the fPAKE protocol.

***FastZIP* Prototype.** We implement *FastZIP* to evaluate its runtime performance on off-the-shelf IoT devices (cf. Section 5.5.6). We focus on the fPAKE protocol, as the underlying functionality takes either constant (e.g., sensing) or negligible time (e.g., quantization). The *FastZIP* prototype allows two devices with similar fingerprints to establish a shared symmetric key. Our implementation is modular and agnostic to the fingerprint derivation, making it directly reusable by other ZIP schemes. To implement the fPAKE protocol, we use primitives from a Python cryptography library [355]. For the ECC, we utilize Shamir's secret sharing scheme in its error-correcting variant (i.e., introducing redundancy by adding more point-value pairs of the polynomial) [75]. For the PAKE component, we use the Encrypted Key Exchange (EKE) protocol [22], built as Diffie-Hellman key exchange symmetrically encrypted with passwords. Our fPAKE implementation supports two security levels, generating keys of 128- and 244-bits. We enable communication between devices utilizing IP sockets and data serialization [100, 101], allowing us to run the *FastZIP* prototype in real-time. To benchmark our implementation, we employ a Python time module [102].

Our fPAKE implementation is modular and agnostic to a fingerprint derivation process.

5.5 EVALUATION

We present a comprehensive evaluation of *FastZIP* based on the real-world data we collect.

Experiment Setup. We collect accelerometer, gyroscope, and barometer data from four cars driven in a number of scenarios: within a *city*, on *country* roads, on a *highway*,

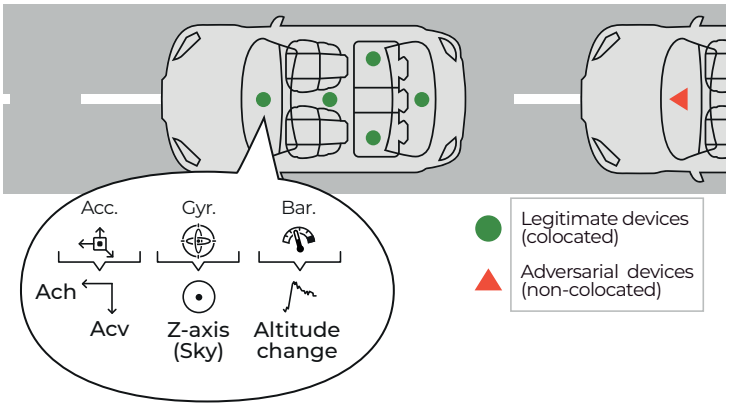


Figure 22: Experiment setup. Smartphones are placed at five various spots inside each of two cars driving the same route.

and inside a *parking* garage for a total of 800 km. To evaluate the suitability of *FastZIP* for various cars, we collect these data using (1) two *similar cars* (Opel Astra wagons; 400 km of driving) and (2) two *different cars* (Škoda Octavia sedan and Volkswagen Golf hatchback; 400 km of driving). In both experiments, we equip two cars with five smartphones each, covering spots where smart devices are typically found [48, 98]: on a dashboard, between front seats, behind driver and passenger seats, and inside a trunk (cf. Figure 22). Then, we collect sensor data from two cars driven as such: (1) one car starts a predefined route, followed by another car after a 10–15 minute lag, (2) two cars drive one after another, changing the distance between each other and the role of a leading vehicle. We cover a similar number of kilometers for *city*, *country*, and *highway* driving. In the *parking* scenario, cars leave an underground garage and return back to it multiple times. To collect the sensor data, we utilize Nexus 5X and Nexus 6P smartphones. After data processing (cf. Section 5.4.1), we use vertical and horizontal acceleration (labeled as *Acv* and *Ach*, respectively), gyroscope sky-axis (*Gyr*), and altitude computed from barometer (*Bar*) in our evaluation.

We collect sensor data from over 800 km of driving of four cars.

Reproducibility and Reusability. We release the collected sensor dataset along with the driven routes map and the source code of our data collection app, evaluation stack, and *FastZIP* prototype [95].

5.5.1 *Methodology*

We evaluate *FastZIP* using several criteria: (1) security and usability, (2) pairing time, and (3) runtime performance. To assess security, we compute False Acceptance Rate (FAR) and evaluate entropy of our fingerprints. A false acceptance occurs when non-colocated devices in different cars (cf. Figure 22) pair because their fingerprints are similar enough. We assess usability by computing True Acceptance Rate (TAR), showing the rate of successful pairings between colocated devices inside the same car. For a detailed

We compute TARs and FARs to quantify the rate of legitimate and adversarial pairings, respectively.

analysis of FARs and TARs, we compute them on the *full* data of an experiment (e.g., *similar cars*) and on the subsets of data corresponding to driving in one of our scenarios: *city*, *country*, *highway*, and *parking*. To evaluate pairing time, we find the amount of context data (in seconds) required to pair securely, while for runtime performance we benchmark the *FastZIP* prototype on the Raspberry Pi.

System Parameters. We use the collected sensor data to find configuration parameters for *FastZIP*'s modules: activity filter, quantization, and fPAKE yielding the best trade-off between security and short pairing time. To find the *length of sensor signal* to derive fingerprint bits, we examine how much sensor data is required to capture typical ambient activity (e.g., car turn by *Gyr*). Our results show that 10 seconds of *Acv*, *Ach*, and *Gyr* data capture typical road bumpiness, acceleration patterns, and car turns, while 20 seconds of *Bar* data is enough to record altitude changes. We set these signal lengths as input to the activity filter and to quantization, using them to empirically find thresholds for activity filter metrics for each sensor modality.

We set system parameters for FastZIP empirically based on our collected data.

To choose the number of *fingerprint bits* output by quantization, we investigate (1) the good ratio between high TAR and low FAR and (2) modality variation. The latter helps us understand how many uncorrelated bits can be extracted from the sensor signal. Based on our findings, we set the number of fingerprint bits to 24 for both *Acv* and *Ach*, 16 for *Gyr*, and 12 for *Bar*. A *similarity threshold* defines the level of similarity between two fingerprints required to establish pairing. To select similarity thresholds, we study how many bits typically differ in the fingerprints of colocated devices. We set the following thresholds, balancing high TAR and low FAR, to be used in the fPAKE protocol: 70.8% (*Acv*), 75% (*Ach*), 93.7% (*Gyr*), and 91.7% (*Bar*).

5.5.2 *Pairing between Colocated Devices*

We compute TARs between each pair of colocated devices inside the same car, providing the average TAR. First, we present TARs for individual sensors (e.g., *Acv*) followed by the evaluation of sensor fusion. Our results are consistent across the *similar* and *different cars* experiments, indicating generalizability of *FastZIP* to various cars. In the following, we provide typical TARs.

TARs of Individual Sensors. Figure 23a depicts TARs for the first car in the *similar cars* experiment. We see that *full* TARs range between 0.84 and 0.91, showing that the individual sensors alone achieve relatively high success rates. However, the TARs of scenarios (e.g., *city*) have higher variation: while *Ach* and *Gyr* exhibit fairly consistent TARs, *Bar* and especially *Acv* show a wider spread of TARs. For *Acv*, the TAR spread is caused by diverse bumpiness perception inside a car affected by such factors as car suspension (e.g., front vs. rear) and surface on which bumpiness is measured (e.g., plastic vs. fabric). These factors become important when a car moves slowly, reducing TARs as in the *city* and *parking*, while higher speed leads to more profound bumpiness, increasing TARs as in the *country* and *highway*. For *Bar*, higher speed

We see that different sensors achieve best TARs in different scenarios.

causes profound altitude changes, improving TARs (e.g., *highway*), while there are few such changes when traveling short distances, reducing TARs (e.g., *parking*). In contrast, *Ach* and *Gyr* show lower TARs when a car moves at constant speed (e.g., *highway*). These sensors benefit from non-monotonic driving with many stops, leading to distinct acceleration patterns (*Ach*) and sharp turns (*Gyr*), as in the *city* and *parking*. Thus, no sensor outperforms the others in all the scenarios, and they show potential for complementing each other.

We analyze TAR deviation inside a car, finding that longer distance between devices leads to lower TARs. This happens because context signals (e.g., road bumpiness) can be attenuated or perceived with varying intensity at distant spots. We find that rapidly changing sensors (i.e., *Acv*, *Ach*) can have up to 20 percentage points of TAR difference between farthest devices, while for gradually changing sensors (i.e., *Gyr*, *Bar*) it is below five percentage points.

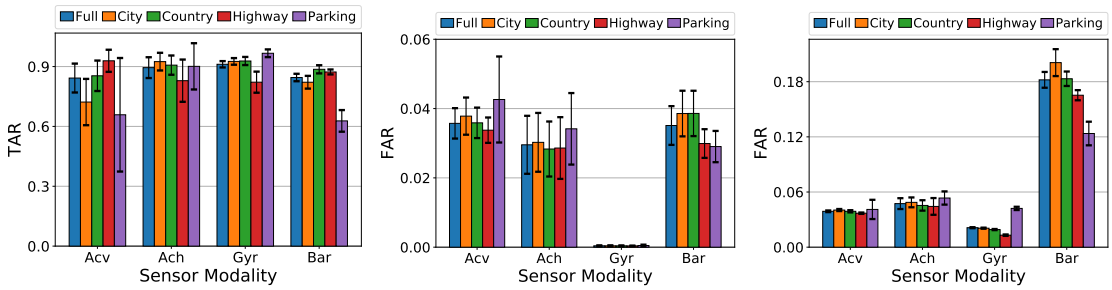
TARs with Sensor Fusion. We fuse sensors by concatenating sub-fingerprints of different modalities derived in the same timeframe. Thus, we obtain more fingerprint bits in less time, speeding up pairing. We explore the fusion of two, three, and all of our sensors. Our findings show that sensor fusion generally increases TARs, while reducing their deviation between devices. This happens because sensors can reinforce each other in the following way: error correction bits unused in the sub-fingerprint of highest similarity allow fixing extra errors in another sub-fingerprint, making the fused fingerprint exceed the similarity threshold, improving the TAR. Such reinforcing effect leads to the fused TAR to be either close to the highest TAR in sensor combination or even exceed it. The latter outcome is typical for sensor combinations including *Ach* and *Gyr*, which often capture co-occuring ambient activity (e.g., decelerate when turning).

Sensor fusion results in higher and less varying TARs among colocated devices inside the same car.

Figure 24a shows a subset of fused TARs for the second car in the *similar cars* experiment. We see that by adding more sensors TARs steadily increase from left to right: ranging from (0.65, 0.89) for individual modalities to (0.85, 0.93) when fusing all of them. With the TAR of 0.9 colocated devices would need 1.1 pairing attempts on average to pair successfully. In few cases, sensors do not reinforce each other, namely combinations including *Acv* and *Bar*, and *Acv* and *Gyr* in the *parking* and *city*, leading to reduced TARs. For *Acv* and *Bar*, both have lowest TARs in these scenarios (cf. Figure 23a), so combining them increases the number of mismatching bits in the fused fingerprint. We find that *Acv* and *Gyr* often capture disjoint ambient activity (e.g., high speed: intense bumpiness but no turns), explaining lower potential for reinforcing each other.

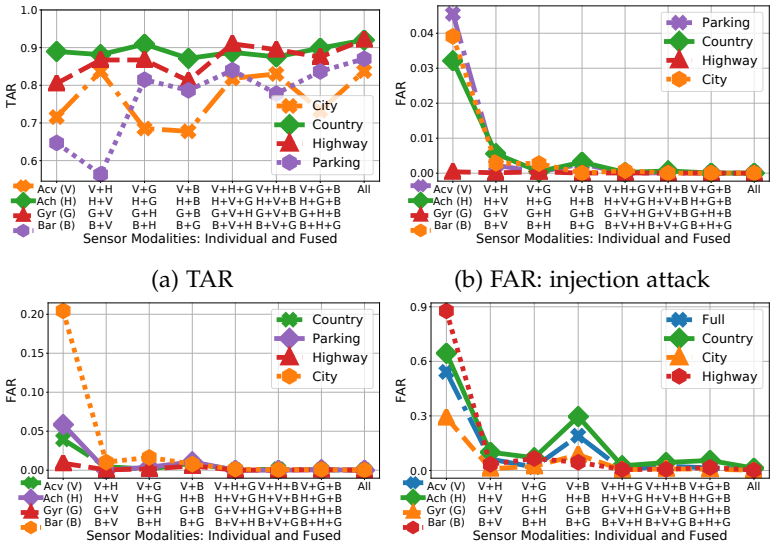
5.5.3 *Resilience to Attacks*

We compute FARs between each pair of non-colocated devices in different cars, presenting the average FAR under *injection*, *replay*, and *similar-context* attacks (cf. Section 5.2). Similar to TAR, we first provide FARs for individual sensors and then evaluate their



(a) TARs (b) FARs: injection attack (c) FARs: replay attack

Figure 23: True Acceptance Rates (TARs) and False Acceptance Rates (FARs) for individual sensors.



(a) TAR (b) FAR: injection attack

(c) FAR: replay attack (d) FAR: similar-context attack

Figure 24: Effect of sensor fusion on True Acceptance Rates (TARs) and False Acceptance Rates (FARs) for representative subsets of scenarios; we connect markers for readability, the plots do not show a time series.

fusion, obtaining consistent results across *similar* and *different cars* experiments. In the following, we provide typical FARs.

Injection Attack. We use sensor data collected inside a parked car capturing noise to pair with legitimate devices. Figure 23b depicts FARs of individual sensors computed for the first car in the *similar cars* experiment. We see that three out of four modalities show FARs above 0.03, making this low-effort attack practical. However, injecting sensor noise does not work on *Gyr* because car turns result in distinct up and down peaks in the signal (cf. Figure 21c) that are not common for noise. With sensor fusion, FARs drop below half a percentage point using two modalities, converging to zero by adding more sensors (cf. Figure 24b). This result is the opposite of the reinforcing effect in TARs, showing that with more sensors differences between non-colocated fingerprints grow, reducing FARs.

Despite low effort, the injection attack succeeds on three out of four sensor modalities.

We also try injecting sensor signals that are collected in a moving car but do not pass the activity filter. In this case, FARs grow by an extra percentage point for *Acv*, *Ach*, and *Bar*, while for *Gyr* they increase by order of magnitude: up to 0.005. Thus, low-entropy sensor signals from a moving car slightly improve the attack, while sensor fusion has the same effect as in Figure 24b.

Replay Attack. We replay sensor signals passing the activity filter from one car to pair with devices in another car; both cars have driven the same route. In the first case, we do not synchronize such replayed signals. Figure 23c depicts FARs of individual sensors for the first car in the *similar cars* experiment. Compared to injection attack, FARs show a fourfold increase for *Gyr* and *Bar*, remaining similar for *Acv* and *Ach*. The altitude change (*Bar*) on a given route has least variation, allowing successfully replay (i.e., FAR of up to 0.2), while other sensors are less affected. We can reach zero FARs by fusing more than two sensors (cf. Figure 24c).

Sensor modalities exhibiting low variation are most vulnerable to replay attacks.

In the second case, we replay sensor signals from periods when both cars drive the same part of the route (e.g., in a city) using a rough timeline of their travel. We see an extra twofold increase in FARs of *Gyr* and *Bar* peaking at 0.07 and 0.38, respectively, while for *Acv* and *Ach* the growth is 1–3 percentage points. Thus, all sensors have FAR above 0.05, making this attack alarming. The sensor fusion leads to zero FARs as in Figure 24c, showing its importance to prevent replay attacks.

Similar-context Attack. We use sensor signals passing the activity filter from one car to pair with devices in another car when two cars drive one after another (cf. Figure 22). We grant the adversary an unfair advantage of matching a single sensor (e.g., *Acv*). It means that they always "guess" the closest fingerprint to the legitimate one; the adversarial and legitimate fingerprints are derived from temporally close sensor signals. Figure 24d depicts the best achievable FARs for this attack. We see that none of individual sensors can prevent the similar-context attack alone, showing FARs between 0.3 and 0.9 (leftmost of the graph). As in the replay, *Bar* that has least variation is the most vulnerable followed by *Ach* and *Acv*. For *Ach* and *Acv*, FARs are caused by shared road conditions such speed limits leading to consistent decelerations (*Ach*) and

The similar-context attack can only be prevented using sensor fusion.

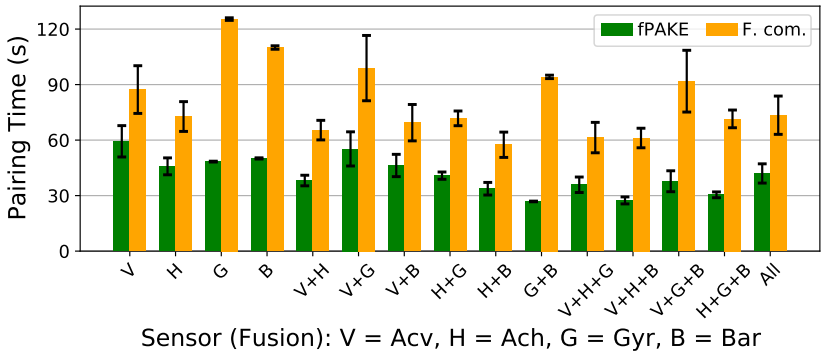


Figure 25: Pairing times obtained from our sensor data for *FastZIP* (fPAKE) and state-of-the-art ZIP schemes (Fuzzy commitments: F. com.).

road cracks resulting in similar bumpiness (*Acv*). *Gyr* is the most robust to this attack because it captures human-specific steering behavior, which varies between drivers.

We see that fusing two sensors cannot prevent the similar-context attack, especially when combining low-varying *Bar* with other modalities (cf. peak in the middle of Figure 24d). By adding three and more sensors, we achieve nearly zero FARs, emphasizing the necessity for sensor fusion to mitigate advanced attacks in ZIP.

5.5.4 *Pairing Time*

We compare pairing time of *FastZIP* utilizing fPAKE and state-of-the-art ZIP schemes based on fuzzy commitments. To enable a fair comparison, we assume the number of fingerprint bits and time to derive them to be the same (cf. Section 5.5.1) and evaluate pairing time on the level of the cryptographic protocol, namely fPAKE vs. fuzzy commitments. First, we calculate how much time it takes to obtain enough fingerprint bits to provide security against offline attacks for fPAKE and fuzzy commitments. For the former, we use findings in Table 14, while for the latter we target a 128-bit fingerprint, accounting for entropy loss due to error correction [245]:

FastZIP requires 20–40 seconds to pair in most situations.

$$|f|_{\mathtt{entropy\_loss}} = |f|_{\mathtt{target}} + 2 \cdot (1 - \mathtt{thr}) \cdot |f|_{\mathtt{target}}$$

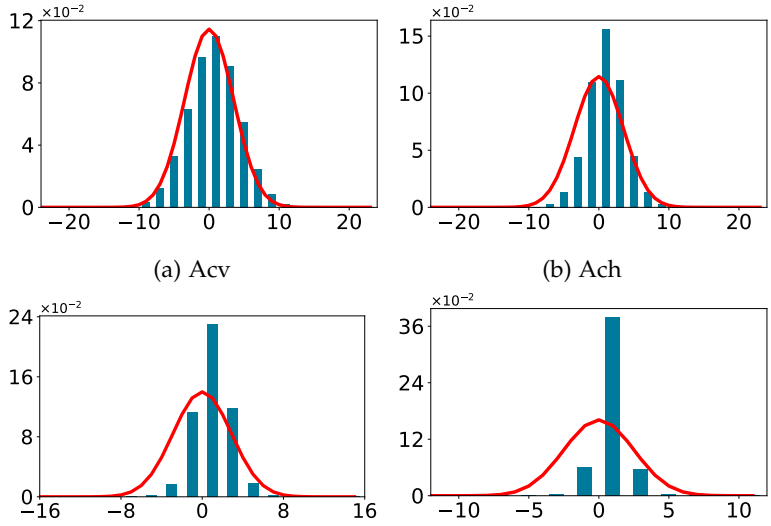
Here, $\mathtt{thr}$ denotes a similarity threshold. Table 15 shows the resulting pairing times, demonstrating that *FastZIP* requires 20–40 seconds to pair in the majority of cases, while state-of-the-art schemes need 1.5–3 times longer time under the same conditions.

Both theoretically calculated and found from our sensor data pairing times demonstrate the advantage of FastZIP.

Second, we evaluate the time required to accumulate fingerprint bits for fPAKE and fuzzy commitments in Table 15 by traversing our collected sensor data with an overlapping sliding window using a 5-second step (cf. Activity Filter in Section 5.3 for reasoning). Figure 25 gives pairing times obtained on the *full* data of the first car in the *different cars* experiment, confirming the 1.5–3 faster pairing time of *FastZIP*. The

Table 15: Calculated pairing times for *FastZIP* (fPAKE) and state-of-the-art ZIP schemes (Fuzzy commitments: F. com.).

Sensor (Fusion)	Sim. Thr.	Fingerprint Bits		Pairing Time (s)	
		fPAKE	F. com.	fPAKE	F. com.
Acv (V)	70.8%	140	203	60	90
Ach (H)	75.0%	120	192	50	80
Gyr (G)	93.7%	50	145	40	100
Bar (B)	91.7%	60	147	100	260
V+H	72.9%	130	198	30	50
V+G	80.0%	100	180	30	50
V+B	77.8%	110	185	80	120
H+G	82.5%	90	173	30	50
H+B	80.5%	100	178	60	100
G+B	92.9%	55	147	40	120
V+H+G	78.1%	110	185	20	30
V+H+B	76.7%	120	188	40	80
V+G+B	82.7%	90	173	40	80
H+G+B	84.6%	80	168	40	80
All	80.2%	100	179	40	60



(a) Acv (b) Ach

(c) Gyr (d) Bar

Figure 26: Distribution of fingerprint random walks for different sensors. Expected binomial distribution in red.

calculated and obtained from our data pairing times are close to each other; the latter pairing times for *Bar* and its fusion combinations are even smaller, as the length of the *Bar* signal (i.e., 20 seconds) is significantly bigger than the sliding window step. We see that pairing times obtained from our data shorten in the case of profound ambient activity (e.g., *Acv* on *highway*), and pairing time consistency inside a car depends on device location for *Acv* and *Ach*, while it is stable for *Gyr* and *Bar*.

5.5.5 *Entropy of Fingerprints*

We do not find obvious entropy biases in our fingerprints.

To evaluate entropy of fingerprints produced by *FastZIP*, we (1) examine them for biases (e.g., bit patterns) and (2) estimate their min-entropy. To identify biases, we represent our fingerprints as random walks, with 1- and 0-bits showing steps in positive and negative directions [36, 98]. The result follows a binomial distribution if fingerprints are uniformly random. We also study bit transition probabilities, interpreting each bit position in a fingerprint as a state in a Markov chain. Figure 26 depicts the results of random walks for individual sensors. The distributions for all sensors are centered around the mean, indicating that overall fingerprints have the equal number of 0- and 1-bits. We see that more unique fingerprints can be generated from modalities with higher variation (e.g., *Acv*). The Markov property is close to 0.5 for all sensors, showing that the probability of each bit in a fingerprint to be 0 or 1 is equal. These findings reveal no biases in our fingerprints, indicating that our quantization achieves its design goals (cf. Section 5.3).

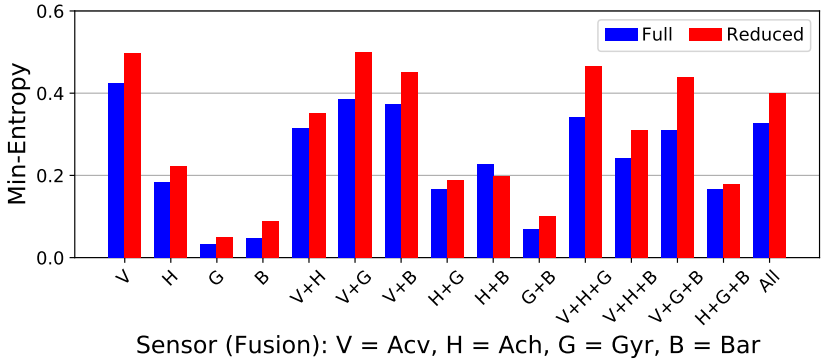


Figure 27: Min-entropy of *FastZIP* fingerprints estimated by NIST SP800-90B test suite (entropy in 1 bit).

To assess min-entropy, we apply the NIST *SP 800-90B* test suite [337, 368]. It consists of ten entropy estimators and is widely used [41, 194, 401]. Figure 27 shows the estimated min-entropy for fingerprints of individual and fused sensors. We obtain 0.43 bits of entropy for *Acv*, 0.19 bits for *Ach*, and below 0.05 bits for both *Gyr* and *Bar*, confirming our findings in Figure 26. Sensor fusion has a positive impact on min-entropy, which either stays close to the highest min-entropy in the combination or exceeds it. The fact that min-entropy increases when combining different sensors, indicates that they are uncorrelated, preventing the adversary from inferring one sensor signal from another. We consider the obtained entropy results to be conservative because the SP 800-90B suite is known to underestimate min-entropy [410], and it makes a fair assessment given $>10^6$ data samples, which we do not have. We find that dependency between consecutive bits is a decisive factor in lowering min-entropy of our fingerprints. To check if this is caused by quantization parameters, we halve the number of bits in our fingerprints (cf. *Reduced* in Figure 27), seeing only a modest increase in min-entropy. Thus, min-entropy in our fingerprints is restricted by the lack of entropy in the sensor data. In Section 5.6, we elaborate on attainable entropy from our sensor data.

With sensor fusion, we obtain more fingerprint bits by compromising very little entropy.

Figure 27 shows that the majority of fused fingerprints have 30–40% of truly random bits. Thus, we need to collect more data to provide security, increasing pairing time of *FastZIP* by 2.5–3 times to 75–120 seconds. For state-of-the-art ZIP schemes, pairing time grows by 7 times, reaching several minutes, as they are more affected by non-random bits due to longer fingerprints.

5.5.6 *Prototype Performance*

We benchmark our *FastZIP* prototype on the Raspberry Pi 3 Model B, recording its performance in terms of computation and communication overhead. Specifically, we randomly sample 2000 fingerprints for each fusion combination, deploying them on

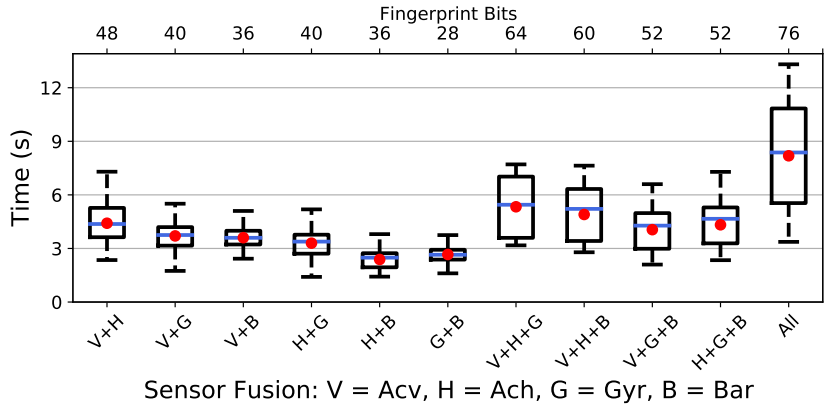


Figure 28: Performance of *FastZIP* for 128-bit key output.

FastZIP requires several seconds to run on the Raspberry Pi.

two Raspberry Pis (i.e., 1000 fingerprints on each) connected via a Wi-Fi router. We measure the execution time to establish a 128-bit symmetric key on each device, showing the average performance in Figure 28[17]. We observe a maximum time of around 4.4 seconds for two sensors (i.e., *Acv* + *Ach*), growing to 8.2 seconds when fusing all of them. The execution time depends on the fingerprint size, and its deviation from the average performance increases with a lower similarity threshold (e.g., *Acv* + *Ach* vs. *Gyr* + *Bar*). We see that 60–80% of the execution time accounts for the communication overhead, which can be reduced using a direct link between devices. Rising the output key size from 128 to 244 bits proportionally increases the execution time. Overall, *FastZIP* runs efficiently on off-the-shelf IoT devices, imposing only a few seconds of overhead. Our prototype is Python-based without performance optimization techniques. Its building blocks (cf. Section 5.4.1) can be reimplemented in C to deploy *FastZIP* on more constrained devices.

5.6 DISCUSSION

We provide relevant discussion points for *FastZIP*.

Generalizability. We show how to adapt the building blocks of *FastZIP*: activity filter, quantization, and fPAKE to be used for ZIP in other use cases (e.g., smart home, wearables). Our activity filter utilizes generic metrics: average power, SNR, and a number of prominent peaks that can be computed on any sensor signal. To find metrics thresholds, excluding low-entropy signals, we examine metrics of sensor signals of fixed length (e.g., 10 seconds), capturing strong and weak ambient activity. With this approach, we obtain thresholds suitable for different cars and road types. Similar results are reported for the average power threshold of audio signals recorded in different

17 We use fingerprints from our evaluation. Accounting for entropy loss in the fingerprints (cf. Table 15) will increase the execution time by a few seconds.

places [181]. Hence, thresholds for activity filter metrics can be determined once per use case and sensor type. Our activity filter can be easily adapted by wearable ZIP utilizing human gait captured by the accelerometer. Specifically, all three metrics (prominent peaks can mark gait cycles) are computed on gait signals of chosen length (e.g., 30 seconds), while metrics thresholds can be derived using public gait data of moving and still users [309].

All building blocks of FastZIP can be adapted to other ZIP use cases.

FastZIP quantization has worked well on four sensor modalities. To apply it for other ZIP use cases, two parameters need to be adjusted: (1) length of input sensor signal and (2) number of output fingerprint bits. These parameters are set empirically based on the duration and variation of scenario-specific ambient activity captured by the sensor signal. For example, in a smart home, a door knock event lasting a few seconds can be recorded by the microphone and accelerometer [144]. The former signal has a higher variation, thus our quantization can be set to output more fingerprint bits from it.

Our fPAKE findings (cf. Table 14) are generic for a given similarity threshold and security level, hence directly reusable by other ZIP schemes. For a different choice of similar threshold/security level, the required fingerprint size in bits providing protection against offline attacks can be computed, as explained in Section 5.3.

Entropy of Sensor Data. The min-entropy presented in Figure 27 results from sensor data collected on high-quality flat roads, giving the lower bound of attainable entropy. We do not cover gravel, forest, or mountain roads that have profound bumpiness (*Acv*) and sharp turns (*Gyr*). Also, we do not have representative data from hectic metropolis driving, which should reveal distinct acceleration patterns (*Ach*) as well as from hilly regions with rapidly changing altitude (*Bar*). These different road and traffic conditions have a high potential for increasing entropy in sensor data [145]. Another way of obtaining more entropy from sensor data is customized quantization. Our quantization focuses on (1) extracting bits from heterogeneous sensors and (2) reducing entropy biases in fingerprints, hence it may not be optimal in the amount of attainable entropy. Prior work explores various quantization methods [36, 131], some of which can be adapted to *FastZIP*.

We expect more entropy can be extracted from sensor data collected in a running car.

Deployment Considerations. To deploy *FastZIP* in a real car setting, a few points need to be considered. First, devices are expected to continuously sense their context before they establish pairing, eliminating the need for time synchronization [144, 245] (i.e., each device extracts fingerprints bits from common parts of context passing the activity filter). In other words, devices observe common context events (e.g., road bump) in the same timeline (i.e., similar to [144]) and can buffer them, tolerating clock offset between devices. For this to work, devices must maintain the same sampling rate of context measurements and start them simultaneously (e.g., upon a broadcasted command). For example, a major component of the car (e.g., infotainment system) can broadcast such a command when a car is started. Since devices in the same car are located nearby, they will receive this command almost at the same time. To further eliminate the effect of different devices receiving the broadcasted command at negligibly different times

We envision FastZIP to be used on a stream of sensor data, capturing context.

and account for overhead to trigger sensing, devices can start measuring context upon the command reception after a short pause (e.g., 5 seconds); for this, they do not need synchronized clocks as well. Furthermore, *FastZIP* extracts much fewer bits from context signals (e.g., 24 bits from 10 seconds) as compared to existing ZIP schemes (e.g., 128 bits from 5 seconds in [393] or 512 bits from 6 seconds in [310]), making *FastZIP* less susceptible to several millisecond offsets between these signals. Specifically, we try injecting 5–7 millisecond offsets between context signals that we used to evaluate *FastZIP*, finding that it would reduce TARs of individual sensors (cf. Figure 23a) by a maximum 10% for *Acv*, 7% for *Ach*, and below 5% for *Gyr* and *Bar*, while FARs remain the same. We consider this reduction to be acceptable for a proof-of-concept *FastZIP*, however further research can investigate how to eliminate the effect of synchronization errors in real deployments. Since *FastZIP* requires a few dozen seconds to pair, collecting context for this time using low-power sensors will not impose much overhead. Second, each device is expected to learn parameters of the scheme (e.g., quantization) prior to pairing: in *FastZIP*, it can happen upon the scheme installation, as commonly assumed in ZIP [144, 309, 388]. Before trying to pair, each device can advertise its desired security level (e.g., 128- or 244-bits in fPAKE), pairing with those devices that support the same security level.

The current version of FastZIP works on stationary devices deployed inside a car.

Limitations. We evaluate *FastZIP* using devices fixed inside a car interior, covering the likely use case of pairing between a mounted user device (e.g., smartphone) and an infotainment system. However, users may interact with their devices, affecting accelerometer and gyroscope readings. Differentiating between human and vehicle motion in the sensor data collected inside a moving car is an open research question [48]. We envision that predicting sensor data resulted from human motion [388] and filtering it afterwards [309] can help address this question.

5.7 RELATED WORK

To date, a number of ZIP schemes utilizing various sensors (e.g., microphone, accelerometer) to capture context have been proposed [144, 145, 210, 243, 245, 309, 310]. The state-of-the-art ZIP schemes rely on the fuzzy commitments cryptographic primitive [175] to establish a shared secret key. Other cryptographic alternatives include customized extensions of fuzzy commitments [309] or the EKE protocol [131]. However, these extensions do not have proven security guarantees. The majority of proposed ZIP schemes rely on a single common sensor to capture context. The existing schemes utilizing fuzzy commitments and context based on a single sensor modality suffer from (1) prolonged pairing time, (2) vulnerability to offline attacks, and (3) attacks caused by the predictable context (e.g., replay). *FastZIP* overcomes these limitations by a novel design, namely combining the fPAKE protocol [75] and multi-sensor context constructed by combining multiple sensor modalities (i.e., sensor fusion).

Table 16: Comparison with state-of-the-art ZIP schemes.

Scheme	Use Case	Time (s)	(FAR, FRR)	Bias
Schürmann and Sigg [310]†	In-car	120	(0.10, 0.10)	Low
Miettinen et al. [243]†	In-car	1280	(0.23, 0.23)	High
Convoy [145]	In-car	300	-	-
Miettinen et al. [245]	Home	5640	(0.03, 0.02)	-
Perceptio [144]	Home	8280	-	-
BANDANA [309]	Wearables	96	-	High
FastZIP	In-car	20	(0.0, 0.06)	Low

†evaluated in [98]. We show best achievable results for each scheme.

Table 16 compares *FastZIP* and prominent state-of-the-art ZIP schemes in terms of pairing time, error rates, and entropy biases in the fingerprints. We note that this comparison is indicative, as we use the information reported in the original publication for each ZIP scheme. *FastZIP* has the shortest pairing time among the schemes, including those that are used for in-car pairing, while achieving low error rates. This shortest pairing time is due to the combination of fPAKE and sensor fusion, which can together give a 3–9 reduction in pairing time (cf. Section 5.5.4). However, pairing time also highly depends on the used context (e.g., continuous gait [309] vs. infrequent knock [144]) and quantization method (e.g., in [243] one bit is derived from two minutes of sensor data).

We compare pairing time, error rates, and fingerprint entropy biases of FastZIP and state-of-the-art ZIP schemes.

The schemes [310] and [243] utilizing ambient audio and noise levels, respectively, are evaluated for in-car pairing [98], showing error rates above 0.1. Despite audio and noise level context varying significantly in a running car, the fingerprints of those schemes contain entropy biases (e.g., more 0-bits). *Convoy* that uses road bumpiness captured by the accelerometer for pairing is vulnerable to the context replay attack [145], however the resulting FAR is not reported. ZIP schemes for pairing smart home devices [144, 245] may achieve comparable error rates to *FastZIP*, requiring, however, at least two orders of magnitude longer time. This time will further increase in the case of entropy biases, which are not evaluated by the considered schemes. We note that the longest pairing time of *Perceptio* [144] is a tradeoff, as the scheme enables pairing between devices with heterogeneous sensors (e.g., microphone and accelerometer). For ZIP schemes targeting wearables such as BANDANA [309], utilizing human gait captured by the accelerometer, the pairing time is closest to *FastZIP*. However, such schemes often show bit patterns in their fingerprints and are vulnerable to video-based attacks [36].

Our review of related ZIP work reveals important results: entropy biases of various level of severity exist in fingerprints of all schemes. This is worrying, as the state of the art relies on fuzzy commitments, where high entropy of fingerprints is imperative to prevent offline attacks. Also, none of the works explicitly accounts for entropy biases

(e.g., by saying how many more bits need to be collected). The impact of entropy biases is less severe in fPAKE, as it limits the offline attack in time and number of attempts. We notice that many ZIP schemes use previous versions of NIST statistical tests [20] to find entropy biases, reporting results for only passed tests, without further investigation [200, 210, 393]. Thus, we urge researchers to scrutinize the entropy of fingerprints derived from context with recent NIST tests [368] and additional tools such as in [36, 98].

5.8 SUMMARY

We recall contributions of this chapter.

In the age of the Internet of Things (IoT), securing wireless communication of smart devices is crucial to protect their data. Zero-interaction pairing (ZIP) allows establishing a shared secret key between devices based on their physical context (e.g., ambient audio). In this chapter, we propose *FastZIP*, a novel ZIP scheme that significantly reduces pairing time while providing stronger security than state-of-the-art ZIP schemes. The main contribution of *FastZIP* is its innovative design combining the Fuzzy Password-Authenticated Key Exchange (fPAKE) protocol and sensor fusion. We implement and empirically evaluate *FastZIP* in the exemplary use case of intra-car device pairing, demonstrating that *FastZIP* (1) reliably pairs devices inside the same car, achieving up to three times faster pairing than state-of-the-art ZIP schemes, (2) is secure against various attacks, and (3) runs efficiently on off-the-shelf IoT devices.

6

ROBUST COPRESENCE DETECTION BASED ON CHANNEL STATE INFORMATION

After evaluating zero-interaction authentication (ZIA) schemes in realistic scenarios, identifying their limitations (cf. Chapter 4), we improve on ZIA with respect to security and deployability in this chapter.

Copresence detection is a necessary prerequisite to ZIA—a technique that allows one device to authenticate another based on their physical proximity. To date, a number of context-based copresence detection schemes utilizing different sensor modalities have been proposed [136, 181, 229, 317, 321, 323, 364, 365]. Some schemes are already used in commercial products such as Futurae Authentication [109] and Apple Auto Unlock [346], making their security and utility crucial for real-world applications. However, these state-of-the-art schemes have two major limitations. First, they show reduced copresence detection accuracy, hence lower security, in the cases of *low-entropy context* (e.g., empty room with few events occurring) and *insufficiently separated environments* (e.g., adjacent rooms). In the first case, the context becomes predictable, allowing the adversary to guess or manipulate it in a controlled manner [98, 325]. In the second case, close environments partly share the context (e.g., loud sound), confusing copresence detection if several devices start it simultaneously [98, 347]. Second, the state-of-the-art context-based schemes require devices to be equipped with common sensors such as microphones, limiting their utility because many Internet of Things (IoT) devices have only a single sensor (e.g., power meter) [144].

State-of-the-art copresence detection schemes have two major limitations.

The above two limitations impair the security and utility of context-based copresence detection schemes, hindering their adoption in the IoT. We address these limitations by proposing *Next2You*, a novel copresence detection scheme based on channel state information (CSI). Figure 29 shows the design space of *Next2You* in comparison to state-of-the-art schemes. Specifically, *Next2You* has higher security, achieving accurate copresence detection in low-entropy context and insufficiently separated environments, and is deployable on devices with heterogeneous sensors, while performing similarly in terms of completion time and distance at which copresence detection is viable.

To the best of our knowledge, we are the first to demonstrate the feasibility of copresence detection based on CSI, leveraging its two advantages. First, CSI is mandatory information generated when Wi-Fi-enabled devices communicate. The ubiquity of Wi-Fi in IoT devices [352] and increasing CSI availability in them [142, 308, 390] allow *Next2You* to run on devices that do not have common sensors but are equipped with Wi-Fi (e.g., a laptop and smart plug). Second, CSI is known to be location-sensitive, capturing variation of a wireless channel, which is affected by the distance between

We are the first to show the feasibility of CSI-based copresence detection for ZIA.

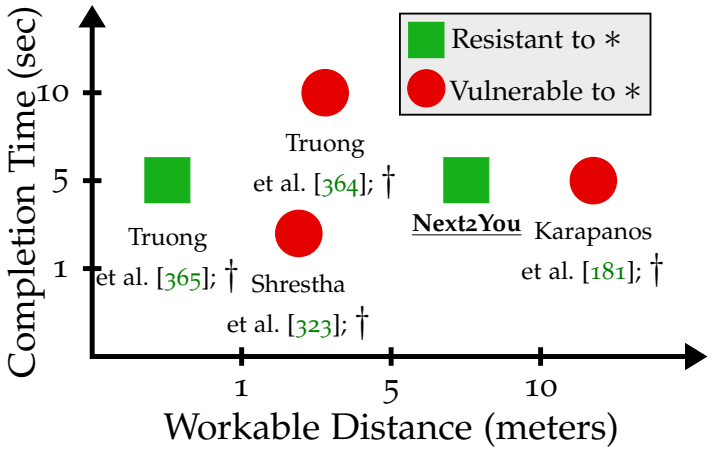


Figure 29: Design space of *Next2You*: comparison with state-of-the-art schemes. *Next2You* prevents attacks caused by low-entropy context and insufficiently separated environments (∗), and it *does not* require devices to have common sensors (†). Similar to the state of the art, *Next2You* provides reliable copresence detection at distances of several meters, requiring a few seconds to complete the procedure.

Copresence detection relies on fewer assumptions compared to existing localization techniques.

devices, geometry and materials of surroundings (e.g., walls), as well as wireless spectrum busyness and mobility [222, 389]. Prior work utilizes this property for secure localization [82, 83, 391], exploiting small changes in the environment (e.g., motion) affecting CSI to detect attacks. However, these works make unrealistic assumptions for the IoT, namely they use bulky software-defined radios with multiple antennas and significant processing power, relying on tight time synchronization between devices and hundreds of CSI measurements per second. Our goal is different, as we aim to detect copresence (e.g., if devices are in the same room) and not precise location. This is challenging because we need to obtain common CSI features for copresent devices within the same environment, ensuring that such features are distinctive among non-copresent devices in different environments. In addition, we evaluate *Next2You* on smartphones that have hardware similar to IoT devices (e.g., portable, single antenna), assuming realistic conditions such as loose time synchronization between devices and a few CSI measurements per second.

We conduct over 90 hours of CSI measurements in realistic scenarios using off-the-shelf smartphones.

To achieve our goal, we propose using CSI magnitude and phase values in a neural network to capture robust wireless context commonly observed by copresent devices. We design the network such that it not only automatically learns relevant copresence features, which we corroborate with prior research, but also enables generalizability and high performance of *Next2You*. To demonstrate the effectiveness of *Next2You*, we collect CSI data in five real-world scenarios, including a busy office, an urban apartment, a rural house as well as parked and moving cars, resulting in over 90 hours of CSI measurements. We show that *Next2You* provides reliable copresence detection with

error rates below 4% in both 2.4 GHz and 5 GHz frequency bands, and it is capable of running on off-the-shelf smartphones in real-time. *Next2You* maintains accurate copresence detection in challenging cases of low-entropy context and insufficiently separated environments. Through our real-world experiments, we demonstrate the robustness of *Next2You* copresence detection, its ability to generalize to new application scenarios, and its resilience to attacks.

Next2You demonstrates lower error rates under challenging conditions than state-of-the-art schemes.

In summary, we make the following contributions:

- We design *Next2You*, a novel copresence detection scheme that combines CSI and neural networks, justifying their mutual suitability for copresence detection.
- We collect a real-world dataset of CSI in five scenarios using off-the-shelf smartphones.
- We implement *Next2You* and evaluate it based on these data, demonstrating its efficacy in distinguishing copresent and non-copresent devices, considering different frequency bands, heterogeneous devices, and attack scenarios. We also show the capability of *Next2You* to work reliably in real-time.
- We publicly release the collected dataset as well as the source code of our CSI data collection app, evaluation stack, and *Next2You* prototype.

6.1 BACKGROUND AND RELATED WORK

In this section, we explain how context-based copresence detection works and review existing schemes, demonstrating the advantages of *Next2You*.

Background. Context-based copresence detection involves two devices: a *prover* and a *verifier*, where the former tries to proof its physical proximity to the latter, and it works as follows. First, the prover sends a copresence verification request to the verifier over a wireless channel such as Bluetooth. Second, both devices capture their context using available sensors for a predefined timeframe (e.g., 10 seconds). Third, the prover transmits its context readings to the verifier over the wireless channel. This channel is secured by means of a shared key, thus the context readings are encrypted and authenticated, protecting them from an adversary. Such a shared key is assumed to be priorly established between the prover and verfier (e.g., via secure pairing [93]). Fourth, the verifier compares its context readings with the ones sent by the prover and decides if they are copresent. To compare context readings, the verifier can either use similarity metrics (e.g., cross-correlation) and check them against the set thresholds [136, 181, 317] or compute features from context readings (e.g., median) and input them to a trained machine learning classifier [229, 321, 323, 364, 365].

A prover and verifier are the two actors in a copresence detection process.

Related Work. To date, a number of context-based copresence detection schemes relying on various sensor modalities, including audio, signal strength, GPS as well as inertial measurement unit (IMU) (e.g., accelerometer, gyroscope) and environmental sensors (e.g., thermometer, barometer) have been proposed [136, 181, 229, 317, 321, 323,

364, 365]. Their details can be found in the survey by Conti and Lal [58]. The existing schemes require devices to have common sensors such as microphones, limiting their applicability, whereas *Next2You* only needs a Wi-Fi chipset, which is ubiquitous in IoT devices. Recent works demonstrate that existing schemes have reduced copresence detection accuracy, hence lower security, in low-entropy context (e.g., empty room with few events occurring) and insufficiently separated environments (e.g., adjacent rooms) [98, 324, 365]. Specifically, Fomichev et al. [98] reproduce three state-of-the-art schemes, showing their vulnerability to the above threats using real-world data. Similarly, Shrestha et al. [324] present successful context injection by a nearby adversary utilizing off-the-shelf home appliances. Truong et al. [365] perform an efficient context manipulation attack, rendering their previous copresence detection scheme [364] insecure. In Section 6.5, we demonstrate that *Next2You* achieves accurate copresence detection in low-entropy context and insufficiently separated environments, mitigating the corresponding attacks.

Next2You does not require common sensors such as microphones for copresence detection.

The location-sensitive properties of CSI have been used for various purposes such as localization, activity recognition as well as user identification and authentication, as detailed by Ma et al. [222]. CSI describes how a wireless signal—spread over multiple carrier frequencies (i.e., *subcarriers*)—propagates between a transmitter and receiver. CSI captures the combined effect of path-loss, fading, reflections, and scattering, which impacts its magnitude and phase [318]. Generally, CSI between a transmitter and receiver at the k-th subcarrier of a Wi-Fi system is defined as:

$$H_k = M_k e^{j\phi_k}. \tag{1}$$

Here, M_k and ϕ_k denote magnitude and phase, which are affected by the placement of the transmitter and receiver, obstacles, geometry of the surroundings, and motion [222]. In Section 6.3.2, we provide a more formal justification for the suitability of CSI for copresence detection.

To leverage location-sensitive properties of CSI, we utilize neural networks, which achieve top performance in CSI-based localization, activity recognition, and user authentication [72, 111, 318]. Differently from these works, we do not focus on feature engineering allowing the neural network to automatically learn the representation of CSI data. Thus, we can avoid a time-consuming and error-prone feature engineering process, enable generalizability of *Next2You* via transfer learning (cf. Section 6.5.3), and reduce its run-time overhead. Despite such advantages of neural networks, we need to ensure that they learn relevant copresence features, as neural networks often overfit the data, providing high classification accuracy, yet learning the wrong hypotheses [197]. For this, we apply a recently introduced method [289] to *interpret* hypotheses learned by our network. Thus, we can quantify which parts of CSI contribute to copresence decision-making and verify our findings against prior work that uses CSI for localization and identification purposes (cf. Section 6.5.4).

In Next2You, we use neural networks, interpreting their learning process to ensure soundness of Next2You.

Compared to the body of work on CSI-based localization, activity recognition, and user authentication [222], we evaluate *Next2You* in a more realistic setup. First, we collect CSI with smartphones, which have hardware closer to IoT devices (e.g., portable, single antenna) than laptops, routers, or software-defined radios used by prior works. Second, we do not assume precise time synchronization between devices, which is required for secure localization [82, 83, 391]. Third, we capture CSI at realistic rates of a few packets (i.e., measurements) per second, which are typical for IoT devices [411], while existing works rely on CSI granularity of hundreds and thousands packets per second [72, 221, 313, 387]. Such high rates are impractical on battery-powered IoT devices, and existing approaches decrease in performance (e.g., classification accuracy) with lower packet rates [111, 313].

Compared to prior work utilizing CSI for various purposes, we evaluate Next2You assuming realistic IoT conditions.

6.2 SYSTEM AND THREAT MODELS

In this section, we present our system model, describing the goal, requirements, and assumptions of *Next2You* as well as our threat model, detailing adversary's goal and capabilities.

System Model. The main *goal* of *Next2You* is for one device (*prover*) to prove its copresence within a trusted boundary (e.g., inside a room) to another device (*verifier*) using their context. We design *Next2You* to fulfill the following *requirements*: (1) be free of user interaction (*usability*), (2) provide reliable copresence detection in low-entropy contexts and insufficiently separated environments (*robustness*), and (3) work on off-the-shelf devices such as smartphones equipped only with a Wi-Fi chipset (*deployability*). To achieve the set goal while satisfying the requirements, we make the following *assumptions*: (1) the prover can send Wi-Fi frames to the verifier, which extracts CSI upon frame reception, (2) the prover and verifier share a secret key, allowing the verifier to ensure frame origin, mitigating replay attacks (e.g., by using a random nonce encrypted with a shared key).

Next2You only needs two devices to be equipped with Wi-Fi chipsets.

Threat Model. The *goal* of the adversary is to convince the verifier that they are copresent, while not being located within the trusted boundary. Specifically, the adversary aims to either *impersonate* a legitimate prover or launch a *relay attack*, where a pair of colluding adversaries forward messages between the prover and verifier. We assume that the adversary can neither compromise the verifier or legitimate prover nor break the encryption between them. However, the adversary fully controls the wireless channel (i.e., can drop, modify, or replay frames) and uses the following capabilities to achieve their goal. In the first case (*passive attack*), the adversary located outside the trusted boundary sends Wi-Fi frames to the verifier, triggering CSI extraction. The adversary is equipped with similar off-the-shelf hardware (e.g., smartphones) to the prover's, thus they can stealthily deploy their devices right outside the trusted boundaries (e.g., in the adjacent office), increasing the attack surface [411]. In addition, the adversary can move their devices along the perimeter of the trusted boundary and stay there for

We consider active and passive attacks against Next2You.

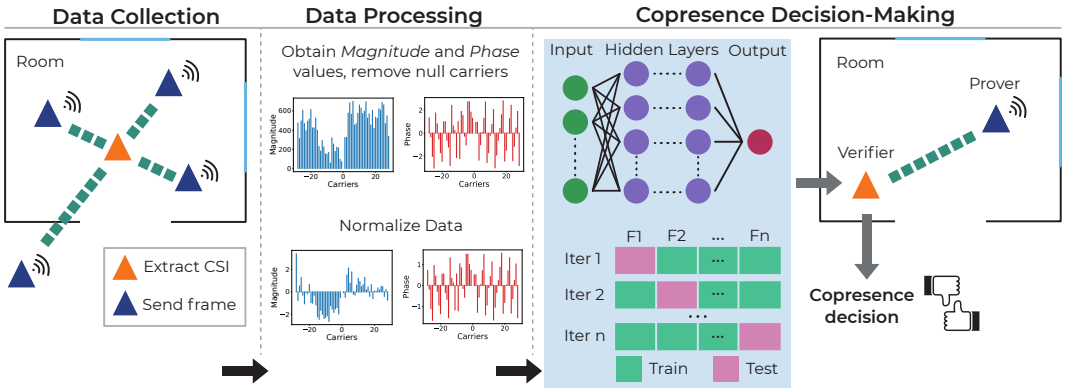


Figure 30: System overview. *Next2You* uses a representative dataset of processed CSI data to train a neural network model that is deployed on a verifier device to predict copresence in real-time.

a prolonged time, including periods of low-entropy context (e.g., adjacent offices at night). In the second case (*active attack*), the adversary has all the capabilities as in the passive attack and can additionally manipulate transmission of Wi-Fi frames such as sending frames of different types or increasing transmission power, aiming to match their CSI with that of legitimate devices inside the trusted boundaries.

We consider adversaries copresent with legitimate devices (e.g., within the same room) to be outside the scope of this work. Prior research shows that for audio such attacks can be mitigated, however, it limits copresence detection to distances below half a meter apart [365]. In the case of CSI, a pairing scheme exhibits a similar trade-off: pairing devices must be no further than a few centimeters apart if the adversary is copresent [389].

6.3 SYSTEM DESIGN

In this section, we first explain the rationale for using CSI together with neural networks for copresence detection and then present the architecture of *Next2You*, describing its modules: *data collection*, *data processing*, and *copresence decision-making*.

6.3.1 *System Overview*

The main goal of *Next2You* is to allow the prover device to confirm its copresence to the verifier device. We consider the following use cases of *Next2You*: it determines copresence of devices located inside the same room or car. Such use cases are typical in the IoT, for example, in a smart home, devices are often moved between different rooms (e.g., smart lamp). Once the relocated device is deployed, the room's smart hub can automatically provide access to the local subnetwork based on copresence. Similarly,

in a connected car, passengers' smartphones can seamlessly share content with the infotainment system due to their copresence.

Next2You achieves copresence detection in three steps (cf. Figure 30). First, the *data collection* module allows the verifier to obtain representative CSI data collected by multiple devices at different spots inside a room or car. This is feasible because the number of IoT devices already reaches roughly a dozen per household [13], increasing to hundreds in few years [144], thus each room in a smart home will be densely covered by distributed IoT devices; the same trend is observed for connected cars [326]. Second, the *data processing* module converts the collected CSI data into magnitude and phase values of Wi-Fi subcarriers, removes irrelevant values, and performs data normalization. Third, the processed CSI data is input to the *copresence decision-making* module to train a neural network model that is deployed on the verifier. Afterwards, the prover wishing to confirm its copresence sends a number of Wi-Fi frames to the verifier, which extracts CSI data upon frame reception, processes it as described above, and inputs the processed CSI to the trained neural network model that outputs a copresence decision.

Next2You has three building blocks: data collection, processing, and decision-making.

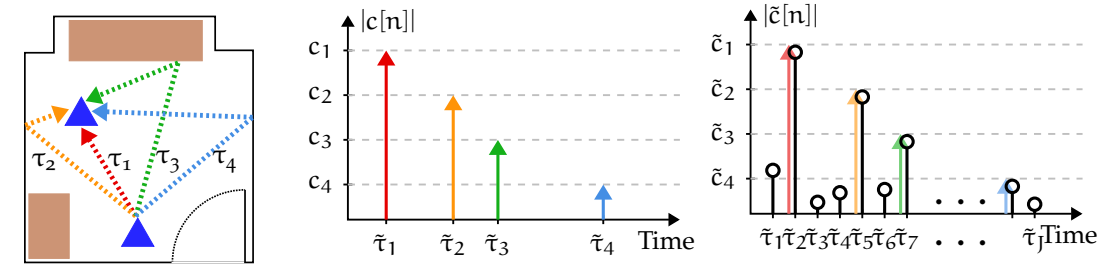
6.3.2 *Rationale for Using CSI and Neural Networks for Copresence Detection*

In the following, we (1) justify the suitability of CSI for copresence detection, (2) rationalize why neural networks can best leverage location-sensitive properties of CSI to detect copresence, and (3) motivate the choice of the Wi-Fi standards that we use for *Next2You* experimentation.

CSI as a Copresence Feature. We demonstrate that CSI is useful for copresence detection because of two reasons. First, it provides a discretized frequency-domain representation of the channel impulse response (CIR), which captures a wireless fingerprint of an environment (e.g., a room) in terms of path-loss, fading, reflections, and scattering of the wireless channel [222]. Second, CSI is the default information generated when two Wi-Fi devices communicate, and it becomes increasingly available in off-the-shelf devices such as routers, laptops, and smartphones (cf. Section 6.6). In the following, we provide an expression relating CSI and CIR via a linear transform, demonstrating that the former indirectly measures the latter, and thus captures a wireless fingerprint of an environment.

CSI is a discretized frequency-domain representation of channel impulse response, capturing unique characteristics of the surroundings.

A signal propagating through an environment such as a room experiences changes that are distinctive to physical characteristics of the surroundings. Such changes—observed as magnitude and phase variations—characterize the communication channel, capturing geometry of the environment, distribution of objects within, and nature of the materials. Mathematically, the communication channel between a transmitter and receiver is represented by the CIR, which *models the overall effect of reflectors, absorbers, path-loss, and complexity of the environment between them*. Figure 31a shows an example of two devices communicating inside the same room that we use for our explanation.



(a) A room with L = 4 paths (b) Ideal CIR (c) Measured CIR

Figure 31: Channel impulse response (CIR) of a room. Figure 31a shows propagation of the transmitted signal between two devices via L = 4 paths. Figure 31b depicts the ideal continuous-time CIR of the scenario in 31a, while Figure 31c shows its discrete-time version, occurring due to limitations in sampling and bandwidth.

We denote the continuous-time CIR of an L-path baseband wireless communication channel as:

$$c(t) = \sum_{i=1}^{L} c_i \delta(t - \tau_i). \tag{2}$$

In Equation 2, $\delta(t - \tau_i)$ is the Dirac delta function representing a delayed multi-path replica of the transmitted signal arriving at time τ_i with power $|c_i|^2$. In particular, $c_i = a_i e^{j\theta_i}$, where a_i and θ_i denote the amplitude and phase of the i-th replica, as shown in Figure 31a and Figure 31b. We note that $c(t)$ fully describes the communication channel between the transmitter and receiver. Nevertheless, there exist technical challenges in the wireless communication chain that hinder its accurate acquisition. Specifically, limitations in the sampling frequency and bandwidth incur in information loss, thereby preventing accurate knowledge of $c(t)$. As a result, only a surrogate version of the CIR can be obtained, which is expressed as:

$$c[n] = \sum_{i=1}^{L} c_i \frac{\sin(\pi(n \cdot \Delta\tau - \tau_i))}{\pi(n \cdot \Delta\tau - \tau_i)}. \tag{3}$$

Note that the discrete-time CIR in Equation 3 is a sampled version of the ideal continuous-time CIR in Equation 2. In particular, $\Delta\tau = \frac{1}{B}$ represents the time resolution (i.e., spacing in seconds between samples), which is inversely proportional to the channel bandwidth B. If all τ_i are multiples of $\Delta\tau$ then $c[n]$ and $c(t)$ become equivalent. Otherwise, if some τ_i is not a multiple of $\Delta\tau$, the energy of that element spreads across all the elements $c[n]$ (i.e., energy leakage) owing to oscillations of the sampling function $\frac{\sin(\pi(n\cdot\Delta\tau-\tau_i))}{\pi(n\cdot\Delta\tau-\tau_i)}$, which produce artificial small-valued samples, as depicted in Figure 31c. *In general, due to sampling and bandwidth limitations, energy leakage inevitably*

occurs among the samples of $c[n]$. Thus, a more realistic representation of the measured CIR is given by:

$$\tilde{c}[n] = \sum_{i=1}^{J} \tilde{c}_i \delta(n \cdot \Delta\tau - \tilde{\tau}_i). \tag{4}$$

Here, all $\tilde{\tau}_i$ are multiples of $\Delta\tau$, and $\tilde{c}[n]$ represents a discrete-time distorted version of $c(t)$, which may not only include perturbations due to sampling and bandwidth limitations but also due to amplitude quantization. We see that $\tilde{c}[n]$ in Figure 31c has $L = 4$ prominent paths similarly to $c(t)$ except for the additional spurious small-valued samples. Although $c(t)$ cannot be completely captured, due to the reasons stated above, there is still valuable information in $\tilde{c}[n]$, which approximately describes the propagation environment. Thus, using CIR (or more precisely $\tilde{c}[n]$) for copresence detection is a sound strategy. However, in the case of Wi-Fi that uses OFDM, the CIR is not readily available. Instead, every Wi-Fi device measures CSI, as it is required for channel estimation and equalization. Fortunately, in OFDM systems, CSI and CIR are related by a bijective mapping through the discrete Fourier transform (DFT) matrix (cf. Equation 5). It means that the matrix $\mathbf{F}$ is invertible, ensuring one-to-one correspondence between CIR and CSI. Thus, for a given CSI measurement of a Wi-Fi frame with K subcarriers, the resulting CIR is unique, and vice versa.

CSI is a ubiquitous measurement at every OFDM-based Wi-Fi device.

$$\underbrace{\begin{pmatrix} H_1 \\ H_2 \\ \vdots \\ H_K \end{pmatrix}}_{\text{CSI}} = \underbrace{\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \exp\left(-j\frac{2\pi}{K}\right) & \cdots & \exp\left(-j\frac{2\pi(K-1)}{K}\right) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \exp\left(-j\frac{2\pi(K-1)}{K}\right) & \cdots & \exp\left(-j\frac{2\pi(K-1)(K-1)}{K}\right) \end{pmatrix}}_{\mathbf{F}\text{: discrete Fourier transform matrix}} \underbrace{\begin{pmatrix} \tilde{c}_1 \\ \tilde{c}_2 \\ \vdots \\ \tilde{c}_K \end{pmatrix}}_{\text{CIR}} \tag{5}$$

The CSI at the k-th subcarrier is defined as $H_k = \sum_{n=1}^{K} \tilde{c}_n \exp\left(-j\frac{2\pi(k-1)(n-1)}{K}\right)$, showing that the CSI at every subcarrier is a linear combination of all CIR elements (cf. Section C.1). While computing the CIR is possible, it requires additional processing (i.e., inverse discrete Fourier transform), which will be computationally expensive on commodity devices. In contrast, CSI is inherently computed by every Wi-Fi device, capturing the same characteristics of the environment as the CIR.

Neural Networks: Leveraging CSI for Copresence Detection. Despite capturing distinct characteristics of an environment (e.g., a room), CSI is sensitive to changes within such as motion and relocation of obstacles [222, 318, 411]. To enable copresence detection, we need to obtain unique features of the environment embedded in CSI, ensuring their robustness to insignificant changes. Hence, we require a technique that can accurately approximate data, capturing its distinct features but tolerating certain noise. To the best of our knowledge, neural networks best fulfill this purpose, in addition to their other advantages (cf. Section 6.3.5).

We find that two points need to be considered to leverage CSI by neural networks for copresence detection. First, CSI captures characteristics of the environment from the viewpoint of a transmitter-receiver pair, thus we need to provide a neural network with CSI observations from different spots inside the environment to obtain its general picture. Collecting CSI in multiple spots is feasible due to the growing number of IoT devices equipped with Wi-Fi (cf. Section 6.3.1). Second, for copresence detection, manually computed features from CSI frequently used by prior work (e.g., mean, power) [222] perform worse in the neural network than raw magnitude and phase values. This happens because prior works engineer features capturing subtle CSI variations to detect a specific location, human, or activity. Such features inevitably reduce the amount of useful information in CSI, hindering the generalization capability of neural networks. The feature computation requires extra processing and more CSI data, and it prevents the representation and transfer learning provided by neural networks (cf. Section 6.3.5). We design *Next2You* guided by the above two points, demonstrating the capability of our neural network to utilize the rich environment information embedded in CSI to automatically learn robust copresence features (cf. Section 6.5.4).

We leverage the representation learning of neural networks in Next2You.

Wi-Fi Standards Used in *Next2You*. To demonstrate the practicality of *Next2You*, we utilize IEEE 802.11n (at 2.4 GHz) and IEEE 802.11ac (at 5 GHz) Wi-Fi standards. We choose them because of their favorable characteristics (described next) and ubiquity in various devices, ranging from simple sensors to powerful routers. First, the lower carrier frequency of 2.4 GHz in IEEE 802.11n enables communication between distant devices due to its robustness to path-loss and blockage. In contrast, the higher carrier frequency of 5 GHz in IEEE 802.11ac makes it more vulnerable to such phenomena but allows capturing subtle details of the environment due to its shorter wavelength. Second, the narrower channel bandwidth of IEEE 802.11n (i.e., 20 MHz and 40 MHz) reduces the circuity requirements (e.g., 64-point and 128-point FFT chipsets), making it suitable for low-power IoT devices. In contrast, the broader channel bandwidth of IEEE 802.11ac (i.e., up to 160 MHz) requires more expensive chipsets (e.g., 512-point FFT chipsets) but provides higher data rates for end-user devices such as laptops.

We evaluate Next2You using different Wi-Fi standards to show its practicality.

6.3.3 *Data Collection*

To obtain a realistic CSI dataset in an environment such as a room, we collect CSI from several copresent devices located inside it at different spots (e.g., on a desk, window sill), varying in terms of nearby obstacles and height above the floor. Similarly, we deploy a number of non-copresent devices outside the environment such as in an adjacent room, collecting CSI data from them as well. Thus, we obtain positive and negative copresence samples in comparable environments that are nearby. This allows a neural network to learn features that are common for copresent and distinct for non-copresent devices, considering the proximate environments. Both copresent and non-copresent devices are fixed, however, we introduce dynamics to their environments by having

We collect CSI using several devices distributed inside a room or a car.

$$\mathbf{X} = \underbrace{\begin{pmatrix} M_{1,1} & M_{1,2} & \cdots & M_{1,K'} \\ M_{2,1} & M_{2,2} & \cdots & M_{2,K'} \\ \vdots & \vdots & \ddots & \vdots \\ M_{N,1} & M_{N,2} & \cdots & M_{N,K'} \end{pmatrix}}_{\text{CSI magnitude}} \underbrace{\begin{pmatrix} \phi_{1,1} & \phi_{1,2} & \cdots & \phi_{1,K'} \\ \phi_{2,1} & \phi_{2,2} & \cdots & \phi_{2,K'} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{N,1} & \phi_{N,2} & \cdots & \phi_{N,K'} \end{pmatrix}}_{\text{CSI phase}} \begin{matrix} \text{CSI measurement 1} \\ \text{CSI measurement 2} \\ \vdots \\ \text{CSI measurement N} \end{matrix}$$

Figure 32: Structure of processed CSI data. K' is the number of subcarriers of a Wi-Fi channel excluding null carriers, N is the number of CSI measurements; $M_{n,k}$ and $\phi_{n,k}$ are normalized to be in the same range.

people within, who frequently move, change position of obstacles (e.g., relocate a chair, open a door), and use other Wi-Fi devices such as laptops and access points. The data collection is finished when a copresent device in each spot have recorded CSI data from other copresent devices and all non-copresent devices for several minutes.

6.3.4 *Data Processing*

We convert the raw CSI of a Wi-Fi frame to magnitude and phase shift values. Specifically, the magnitude M_k and phase ϕ_k of the CSI at the k subcarrier (denoted by H_k), are given by modulus and argument, as shown in Equation 6.

$$M_k = \sqrt{\Re\{H_k\}^2 + \Im\{H_k\}^2}, \quad \phi_k = \operatorname{atan}\left(\frac{\Im\{H_k\}}{\Re\{H_k\}}\right). \tag{6}$$

Here, $\Re\{H_k\}$ and $\Im\{H_k\}$ are the real and imaginary parts of H_k. The number of subcarriers depends on the Wi-Fi channel bandwidth. For example, the 20 MHz channel in 802.11n consists of 64 subcarriers, resulting in 128 magnitude and phase values. In *Next2You*, we consider all subcarriers of a Wi-Fi channel in order to obtain a fine-grained wireless fingerprint of an environment (e.g., a room) captured by CSI. For example, similar to Shi et al. [318], we find that some subcarriers are more susceptible to noise than others with no discernible pattern observed. Such noise susceptibility might be distinctive to the environment, indicating a specific interference behavior. However, not all subcarriers provide meaningful CSI, namely *null subcarriers*, which do not carry any information [114]. Thus, we remove magnitude and phase values of such subcarriers from our data. We normalize the computed magnitude and phase values to have the same range, making them suitable to train a machine learning classifier on. The structure of the processed CSI dataset is arranged in a matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$, as shown in Figure 32, where $D = 2K'$ is the dimension of the feature vector for every measurement $n = \{1, \cdots, N\}$, K' is the number of useful subcarriers, while $M_{n,k}$ and $\phi_{n,k}$ denote the magnitude and phase of the k-th subcarrier in the n-th measurement.

We obtain CSI magnitude and phase values from all subcarries of a Wi-Fi channel except the null subcarries.

6.3.5 *Copresence Decision-making*

Next2You relies on neural networks because of their four advantages.

To capture a wireless fingerprint commonly observed by copresent devices inside the same environment (e.g., a room), we input the processed CSI data to a machine learning classifier. Differently from existing copresence detection schemes utilizing machine learning [136, 229, 317, 321, 323, 364, 365], we choose to use neural networks for the following reasons. First, neural networks under mild assumptions are universal function approximators, thus they have the potential for representing the classification function we are interested in learning [154, 217, 331]. Second, neural networks allow representation learning [122], which replaces the manual feature engineering process, simplifying the modeling assumptions, saving time, and increasing accuracy. The prior work on wireless signal classification demonstrates that neural networks are capable of learning the right representation for this domain, producing high predictive accuracy results [150, 233, 259, 320]. Third, the representation learning of neural networks enables transfer learning [350], where we can reuse the representation learned in one problem and embed it into the solution of another. This has become a standard practice in the deep learning community, where big networks are trained on massive amounts of data. Such pretrained networks are publicly shared with other researchers and practitioners who adapt them to new tasks without training from scratch, significantly lowering the computational costs. In Section 6.5.3, we leverage the capability for representation and transfer learning to demonstrate that our neural network can be adapted to new environments while reducing computational costs, making model training in *Next2You* feasible on battery-powered devices. Fourth, numerous deep learning frameworks, support for different devices, and constant improvements in neural networks facilitate the deployment of *Next2You*.

6.4 IMPLEMENTATION

In this section, we provide the implementation details of our CSI collector, copresence decision-making module, and *Next2You* prototype.

6.4.1 *CSI Collector*

Our CSI collector works on Nexus 5 and Nexus 6P smartphones.

For data collection, we develop an Android app, utilizing the *Nexmon framework* [308] to extract CSI. Nexmon allows modifying the firmware of Broadcom Wi-Fi chipsets of Nexus 5 and Nexus 6P smartphones. We customize the original CSI-extractor [307] to our needs, allowing us to conduct experiments with both Nexus 5 and Nexus 6P devices.

Our app works in two modes: the *prover* and *verifier*. The former broadcasts Wi-Fi frames at a predefined rate, while the latter listens for these frames, extracting CSI upon frame reception and storing it on a smartphone. The app allows CSI collection

in both 2.4 GHz and 5 GHz Wi-Fi bands. As input, the number and bandwidth of a Wi-Fi channel on which the prover and verifier communicate needs to be provided. We experiment with a number of Wi-Fi channels in both frequency bands using two criteria: (1) the stability of CSI collection and (2) maximum transmission range. We find that many channels in the 2.4 GHz band vary significantly in terms of CSI collection stability, while the 5 GHz channels show a wide spread of transmission ranges. Our results obtained in various environments demonstrate that the channel 1 (20 MHz bandwidth, 2.4 GHz band) and channel 157 (80 MHz bandwidth, 5 GHz band) best satisfy the above criteria, thus we use them to collect CSI data in our experiments. With these channels, we obtain 128 and 510 magnitude and phase values from a single CSI measurement for 2.4 GHz and 5 GHz bands, respectively, which corresponds to using all subscarries of a Wi-Fi channel, including null ones. Furthermore, we implement different types of frames (i.e., quality of service (QoS) and beacon) used by the prover and verifier as well as varying transmission power of the prover, allowing us to evaluate the robustness of *Next2You* and its resilience to attacks. Using frames of around 100 bytes in size, we find that Nexmon allows extracting CSI at the rate of up to 20 frames per second. In our experiments, we use several provers and a single verifier (cf. Section 6.5.1), thus we set the transmission rate of a prover to three frames per second, ensuring reliable CSI data collection on the verifier.

We experiment with different Wi-Fi channels to find most relevant for Next2You.

6.4.2 *Copresence Decision-making*

We treat the problem of copresence detection as a binary classification task. Thus, we use the collected CSI data to train a machine learning classifier, predicting whether two devices are copresent (i.e., in the same room or car).

To collect CSI, we change in turns the position of the verifier (e.g., on a window sill, on a desk), resulting in a number of CSI datasets from different verifier-provers combinations (cf. Section 6.5.1). To estimate the performance of our classification approach, we use a 5-fold cross-validation (CV). This is a reasonable compromise between the computational costs of training the model and a good estimation of the predictive performance. Due to a larger number of non-copresent devices in our experiments, our dataset has an imbalanced distribution of the classes. Hence, we employ a stratified CV and classification metrics that are not affected by the imbalanced data (cf. Section 6.5.1). We construct the folds by (1) loading the CSI data from all verifier-provers combinations, (2) shuffling it with a set seed (i.e., 123), and (3) evenly sampling the data into each fold. Thus, we obtain five folds, each containing the same amount of data and ratio of positive and negative samples as the original CSI datasets from all verifier-provers combinations. Prior to training, we delete the irrelevant *null subcarriers* from the CSI data, resulting in 112 and 484 features (i.e., magnitude and phase values) per CSI measurement for 2.4 GHz and 5 GHz bands, respectively (cf. Figure 32). We normalize features in training and test sets using variance scaling [408], ensuring that

We employ a 5-fold cross-valuation for copresence classification.

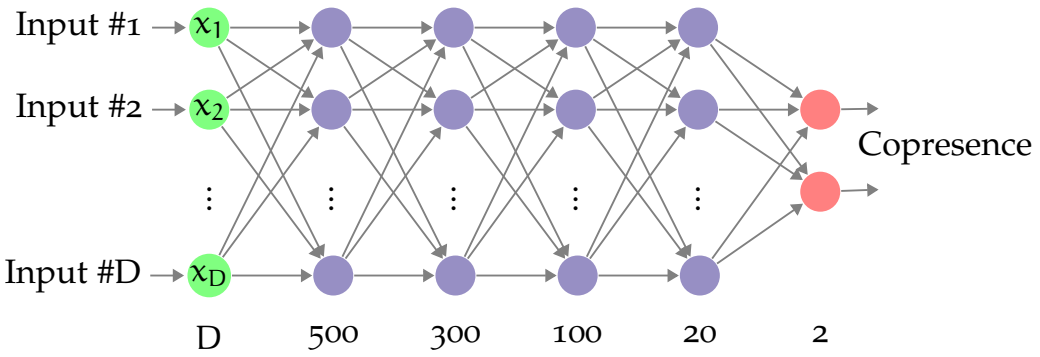


Figure 33: Structure of a neural network with four hidden layers (number of neurons is shown below) used in *Next2You*; the network takes as input the processed CSI data and outputs if two devices are copresent or not.

they have the same range, and thus are equally weighted by a classifier. The normalized features are obtained as:

$$x_{i,\text{norm}} = \frac{x_i - \mu_i}{\sigma_i} \tag{7}$$

Here, μ_i and σ_i are the mean and standard deviation of the i-th feature, where $i = \{1, \cdots, D\}$.

As discussed in Section 6.3.5, we choose neural networks as a machine learning classifier, allowing us to avoid manual feature engineering. With the 5-fold CV, we need to train a neural network five times, thus we design it with performance and reasonable training time in mind. Specifically, we implement a neural network in *Keras* [353] using four hidden layers with a decreasing number of neurons in each subsequent layer (i.e., 500, 300, 100, 20) and a softmax output layer (cf. Figure 33). We denote each input vector as $\mathbf{x} = [x_1, \cdots, x_D] \in \mathbb{R}^{D \times 1}$, containing K' magnitude and K' phase values for every CSI measurement. Our network architecture has many neurons in the layers close to the input, reducing the number of neurons closer to the output. This gives the network enough capacity to learn a feature representation and gradually reduces the classifier's complexity in the last layers. We do not use convolutions, as they introduce interdependencies among the features given the mask size and stride hyper-parameters. To avoid this hyper-parameter search, we use a dense neural network with enough neurons and layers to model CSI data collected in different environments (e.g., office rooms, cars). However, we leave as future work the test of pruning methods to reduce the network size as well as the behavior of convolutions and attention mechanisms [106, 198, 373]. We use the Leaky-ReLU [392] activation function for the hidden layers' neurons, as it avoids the vanishing gradient problem and also back-propagates gradients for negative values. For the loss function, we utilize the standard cross-entropy, and to avoid overfitting, we apply the dropout [335], which is a standard technique used for regularization. We train the network until convergence, setting the number of epochs to 35 and 25 for 2.4 GHz and 5 GHz bands, respectively.

We implement our Next2You neural network in Keras.

6.4.3 *Next2You Prototype*

We implement a *Next2You* prototype working in real-time by combining the functionality of our CSI collector with the *TensorFlow Lite* framework [354], allowing us to use pretrained neural network models directly on smartphones. Specifically, the prototype extracts CSI from incoming frames, processes it to the required input format (i.e., removing null subcarriers, performing feature normalization), and feeds the processed CSI to the model, which, in turn, outputs a copresence prediction. For the prototype, we reuse the same neural network architecture described in Section 6.4.2. To leverage CSI temporality, we accumulate successive CSI measurements within a time window, generating a copresence prediction for each measurement and using a majority vote to obtain the final decision. We consider time windows of 5 and 10 seconds, which provide a fair trade-off between the speed and reliability of copresence detection, respectively [364].

Our Next2You prototype is capable of working in real-time.

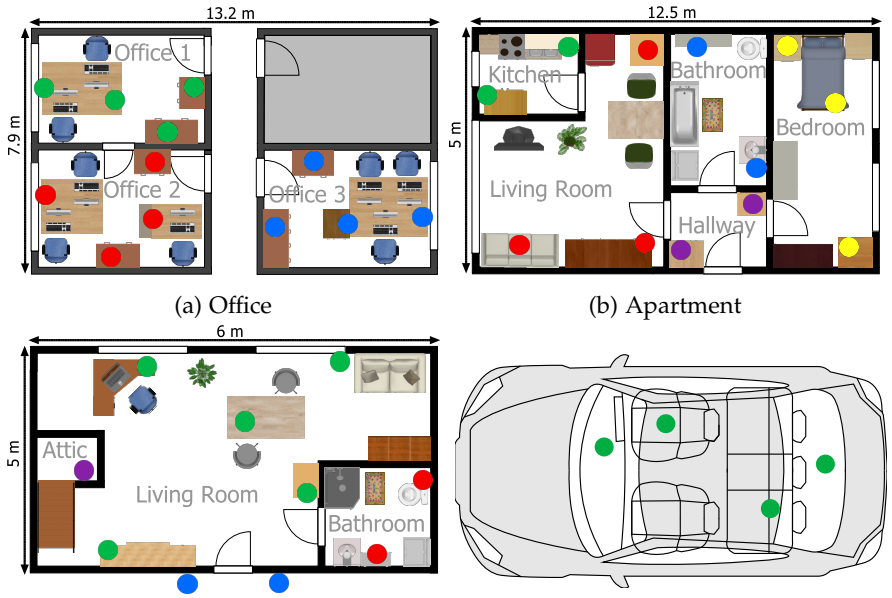
6.5 EVALUATION

In this section, we present a comprehensive evaluation of *Next2You* based on the real-world CSI data that we collect.

6.5.1 *Experiment Setup*

To evaluate the capability of *Next2You* to detect copresence inside the same office, room, or car, we collect CSI data in five different scenarios: *office*, *apartment*, *house* as well as *parked* and *moving cars*. These scenarios vary in terms of size and geometry, wall and obstacle materials, and the number of occupants. Specifically, the *office* consists of three office rooms: two adjacent offices and one across a hallway, occupied by one to three persons, the *apartment* is a two-room flat inhabited by two people, while the *house* is a single-person household, and the *cars* are either parked side by side without any occupants in them, or being driven one after another with a single person inside each car for a total of 120 km. To evaluate the impact of (1) heterogeneous smartphones, (2) different frame types, and (3) varying transmission power on copresence detection performance of *Next2You*, we collect CSI data in these three configurations reusing the office setup, resulting in *heterogeneous*, *frame*, and *power* scenarios. In the *office*, we collect the data over the course of several days, allowing us to assess copresence detection performance of *Next2You* at different times of the day (e.g., morning vs. night), while in other scenarios we record CSI for up to eight hours, as shown in Table 17. To capture CSI data, we use our app installed on Nexus 5 and Nexus 6P smartphones (cf. Section 6.4.1), which are deployed in spots where IoT devices are typically found such as on a desk in the *office*, on a dashboard in the *car*, and near a TV set in the *apartment* (cf. Figure 34).

To evaluate Next2You, we collect CSI data in five different scenarios.



(a) Office (b) Apartment

(c) House (d) Parked and moving cars

Figure 34: Placement of smartphones collecting CSI data in different scenarios. The smartphones are marked with circles, which are of the same color for copresent devices. We depict the exterior and main obstacles in each scenario. The goal of *Next2You* is to distinguish between copresent and non-copresent devices located in different offices, rooms, or cars (e.g., devices in Office 1 from devices in Offices 2 and 3, cf. Figure 34a).

Table 17: Overview of CSI data collection settings in different scenarios for 2.4 GHz and 5 GHz bands.

Scenario	Number of Devices	Accumulation round, minutes		Overall Time, hours
		2.4 GHz	5 GHz	
Office†	12	35 (20)	25 (20)	44
– Heterogeneous*	12	20	20	8
– Frame*	12	20	20	8
– Power*	12	20	20	8
Apartment	12	20	20	8
House	10	20	20	4.7
Parked Cars	8	10	10	2.7
Moving Cars	8	20	10	4

†Collected over several days; *Uses setup in Figure 34a; () - Accumulation round at nighttime.

To obtain representative CSI data, we collect it at each spot in the scenario. Specifically, during a data accumulation round, we have one device (*verifier*) capturing CSI data from all other devices (*provers*). For example, if a device on a window sill in Office 1 is a verifier, then it collects CSI data from three copresent provers in Office 1 and eight non-copresent provers located in Offices 2 and 3 (cf. Figure 34a). In the next data accumulation round, we change the position of the verifier (e.g., to a desk in Office 1), capturing CSI from all provers again, repeating this procedure until we obtain CSI data from each spot in the scenario (e.g., all circles in Figure 34a). We set the length of the data accumulation round based on the amount of CSI data available from 2.4 GHz and 5 GHz frequency bands (i.e., 112 and 484 magnitude and phase values, respectively) and complexity of the scenario (i.e., the amount of motion and number of obstacles), as presented in Table 17.

Reproducibility and Reusability. In total, we collect over 90 hours of CSI data. We release the collected dataset and trained neural network models as well as the source code of our data collection app, evaluation stack, and *Next2You* prototype [91].

Ethical Considerations. Due to the sensitivity of the CSI data [411], we obtained the approval for this study from our institutional ethical review board, the participants residing in experimental locations gave informed consent for the collection, use, and release of the CSI data.

Performance Metrics. We evaluate whether *Next2You* can correctly classify copresent and non-copresent devices (e.g., distinguish devices in Office 1 from devices in Offices 2 and 3, cf. Figure 34a). Specifically, we train a neural network model on the CSI data and compare its predictions with the ground truth, computing two performance metrics: *Area Under the Curve (AUC)* and *Equal Error Rate (EER)*. The former shows how well the model can distinguish between copresent and non-copresent classes, with a higher AUC indicating a more accurately discriminative model; AUC is invariant to class imbalance compared to other metrics (e.g., accuracy). The latter is the intersection point of *False Acceptance Rate (FAR)* and *False Rejection Rate (FRR)*, balancing the number of misclassified non-copresent and copresent devices, respectively. The FAR represents the *security* of the system (i.e., non-copresent devices classified as copresent), while the FRR shows its *usability* (i.e., copresent devices classified as non-copresent), thus a low EER is desirable to achieve both these properties.

We use EER and AUC metrics to evaluate the copresence detection performance of Next2You.

6.5.2 *Copresence Detection Performance*

The results of *Next2You* copresence detection performance from our experiments are provided in Table 18, where the two scenarios (i.e., *frame* and *power*) correspond to the active adversary, while the rest—represent the passive adversary (cf. Section 6.2). In these experiments, we train a neural network on both copresent and non-copresent samples, investigating how well *Next2You* performs on unseen CSI data in advanced attack scenarios (cf. Section 6.5.5). From Table 18, we observe that *Next2You* provides

Table 18: AUC and EER of *Next2You* in different scenarios for 2.4 GHz and 5 GHz bands in the presence of active and passive adversaries; the 5 GHz band shows slightly better results.

Scenario	Adversary	2.4 GHz		5 GHz	
		AUC	EER	AUC	EER
Office	Passive	0.958	0.040	0.995	0.005
– Heterogeneous	Passive	0.982	0.014	0.996	0.002
– Frame	Active	0.988	0.010	0.993	0.005
– Power	Active	0.995	0.002	0.999	0.000
Apartment	Passive	0.961	0.022	0.984	0.012
House	Passive	0.993	0.007	0.984	0.015
Parked Cars	Passive	0.998	0.001	0.999	0.000
Moving Cars	Passive	0.996	0.002	0.997	0.001

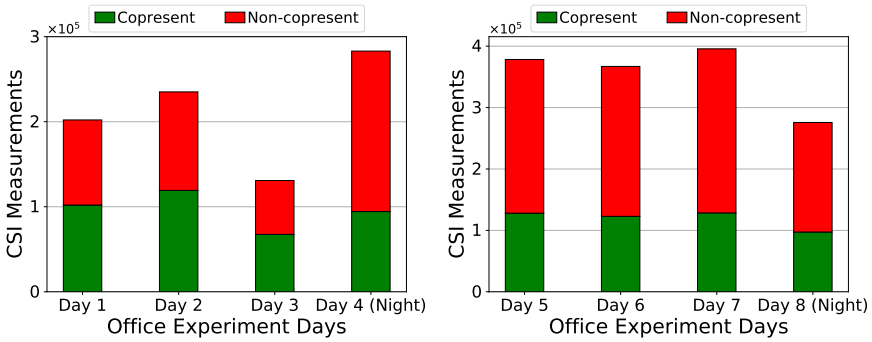
- shows best achievable EER.

reliable copresence detection in different scenarios, achieving EERs between 0 and 0.04. *Next2You* shows more accurate copresence detection using the 5 GHz frequency band, and its performance decreases in larger environments with many obstacles as well as human and object motion (e.g., closing a door) compared to smaller stationary scenarios (e.g., *office* vs. *parked cars*).

Impact of Frequency Band and Bandwidth. Table 18 reports that CSI data from the 5 GHz band allows distinguishing copresent and non-copresent devices more accurately. We see three reasons for this result. First, the higher sensitivity of 5 GHz to path-loss causes more severe power attenuation from non-copresent devices, limiting their communication range. Second, the shorter wavelength at 5 GHz (i.e., 6 cm), which is more easily perturbed by small-sized objects compared to 2.4 GHz (wavelength = 12.5 cm), allows discerning characteristics of the environment such as obstacle placement with higher granularity. Third, the broader channel bandwidth at 5 GHz (i.e., 80 MHz) improves the time resolution of the CIR. Thus, more CIR paths can be distinguished from one another (cf. Figure 31a), resulting in more detailed CSI measurement, including information on distance ranges between devices inside the same environment.

Using 5 GHz frequency band, Next2You achieves more accurate copresence detection in small- and medium-sized environments.

From a security perspective, the shorter range of 5 GHz is beneficial, as it forces the adversary to stay closer to legitimate devices. For example, in the *moving cars* scenario, we had to drive two vehicles very slowly to maintain a distance of a few meters to be able to capture any CSI data from non-copresent devices. In reality, following another car in such a way will immediately raise suspicion, imposing a physical barrier on the adversary's capability. However, the shorter communication range of 5 GHz can hinder



(a) 2.4 GHz frequency band (b) 5 GHz frequency band

Figure 35: Impact of frequency band on the number and ratio of CSI measurements collected by copresent and non-copresent devices in the office scenario on different days and at night; the number of CSI measurements collected at night is expected to be smaller due to a shorter data accumulation round (cf. Table 17).

usability—we see that the AUC and EER for the *apartment* and *house* in 5 GHz are the lowest among the scenarios (cf. Table 18), indicating that copresent devices in larger rooms with many obstacles inside experience a rapid decrease in performance.

We study the structure of our CSI data, finding that for the 2.4 GHz band in the *office*, *frame*, and *heterogeneous* scenarios, the number of copresent and non-copresent CSI measurements does not conform to the expected ratio of roughly 30% to 70% (i.e., three copresent vs. eight non-copresent devices, cf. Figure 34a). Instead, the number of copresent and non-copresent measurements is almost equal, which occurs only during working hours and not at night (cf. Figure 35a). In contrast, all 5 GHz measurements (cf. Figure 35b for the *office*) as well as 2.4 GHz in other scenarios, follow the expected ratio. We attribute this behavior to the crowded spectrum in the 2.4 GHz band during working office hours (e.g., heavy Wi-Fi and Bluetooth traffic), preventing many frames from non-copresent *provers* reaching the *verifier* in their wireless range due to interference. The spectrum busyness generally reduces the overall number of collected CSI measurements (e.g., Day 3 vs. Day 4 (Night) in Figure 35a). This effect is beneficial for security, providing better separation between copresent and non-copresent devices. However, it may hinder usability or availability of *Next2You* in large environments (e.g., lecture hall) with many wireless devices operating in the same spectrum.

Interference on a Wi-Fi channel hinders the attack on Next2You.

Impact of Time of Day. Figure 36 depicts the AUC performance of *Next2You* at different times of day (i.e., morning, afternoon, evening, and night) in the *office* scenario. We see that during a day (i.e., morning till evening), the 5 GHz band shows both higher and more consistent AUCs compared to 2.4 GHz, confirming its suitability for small- and medium-sized environments. In addition, AUCs of different days retain similar trends in both frequency bands, suggesting that such factors as spectrum busyness

Next2You shows accurate copresence detection with people present and absent in the environment.

(e.g., possibly congested 2.4 GHz spectrum at Day 3, cf. Figure 35a) and motion do not significantly affect the performance of *Next2You*. This finding is confirmed by the AUC at night, which remains comparable to the day AUCs in both frequency bands, despite the absence of motion and minimum spectrum busyness. The high AUC at night shows that *Next2You* can cope with low-entropy context even for adjacent Offices 1 and 2 separated by a thin wall (cf. Figure 34a).

Impact of Heterogeneous Devices. To evaluate the capability of *Next2You* to work on heterogeneous devices, we make a customized port of the Nexmon CSI-extractor [307] to the Nexus 6P smartphone, which has a *different* Wi-Fi chipset than Nexus 5, allowing us to send frames and extract CSI data with Nexus 6P. We find that due to two transmitting antennas in Nexus 6P, the CSI resulting from a frame sent by Nexus 6P differs[18] from CSI extracted by Nexus 5, which contains a single antenna. Hence, we use a Nexus 6P smartphone to capture CSI data from frames sent by Nexus 5 devices. Table 18 shows that the AUC and EER in the *heterogeneous* scenario are moderately better compared to the *office* (two scenarios share a setup, cf. Figure 34a) in both frequency bands, indicating the practicality of *Next2You* on heterogeneous devices.

We demonstrate the capability of Next2You to work on devices with heterogeneous Wi-Fi chipsets.

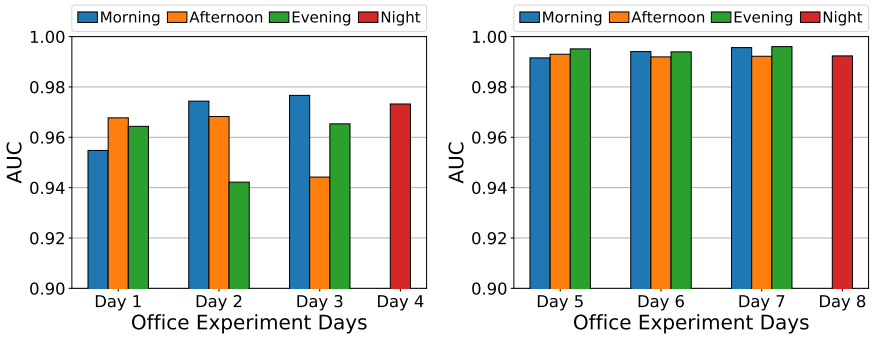
Impact of Frame Type. To assess the robustness of *Next2You*, we change the transmitted data format extracting CSI from *beacon* instead of QoS frames. We do not observe significant changes in the structure of collected CSI data (e.g., similar trends for CSI measurements, cf. Figure 35). The AUC and EER in the *frame* scenario are noticeably better compared to the *office* in 2.4 GHz but are similar in the 5 GHz band (cf. Table 18). We attribute the improved performance in 2.4 GHz to a shorter data accumulation round in the *frame* scenario (i.e., 20 vs. 35 minutes in the *office*, cf. Table 17), capturing fewer complex events such as relocation of obstacles, leading to more accurate copresence detection. Thus, we demonstrate that *Next2You* is agnostic to the frame type, facilitating its deployability (e.g., one verifier can execute *Next2You* with several provers, each using a different frame type).

Impact of Transmission Power. We study if an adversary can attack *Next2You* by increasing the transmission power of non-copresent devices to overcome such effects as path-loss and shadowing [318]. To find by how much the power needs to be increased, we increment it until we obtain the expected ratio of CSI measurements (i.e., roughly 30% to 70%) for copresent and non-copresent devices at 2.4 GHz in the office setup during working hours. Thus, all frames sent by non-copresent provers reach the verifier despite the interference and blockages as if they are "copresent". We empirically find that the power needs to be increased by ten times to achieve this.

Next2You can discern devices communicating with different power levels.

Table 18 shows that the AUC and EER of the *power* scenario are significantly better compared to the *office* in both frequency bands. We verify this result by finding no

18 To the best of our knowledge, the official port of the Nexmon CSI-extractor on Nexus 6P now also supports single or multiple spacial streams, which should address this limitation (cf. https://github.com/seemoo-lab/nexmon_csi).



(a) 2.4 GHz frequency band (b) 5 GHz frequency band

Figure 36: Impact of time of day on AUC of *Next2You* in the office scenario.

noticeable difference between CSI magnitude[19] and its variance in both scenarios as well as normalizing the CSI magnitude by a unit power and obtaining unchanged AUC and EER. Based on the interpretation of hypotheses learned by our neural network from the CSI data (cf. Section 6.5.4), we conclude that increased power affects the statistical properties of CSI, making it easier to classify devices transmitting with different power. Specifically, higher power produces additional CIR paths (cf. Section 6.3.2) originated from the adversary's environment, distinguishing the CSI of non-copresent devices from copresent.

Real-time Performance and Impact of Mobility. We evaluate the real-time performance of *Next2You* in the *office* scenario using our prototype (cf. Section 6.4.3) in two cases. First, we adjust the position of smartphones within 50 cm from their initial spots (cf. circles in Figure 34a), putting some devices on top of books or boxes. Then, a device at each spot in turn acts as the verifier predicting copresence in real-time, while all other devices are provers. In this case, *Next2You* correctly detects copresence around 95% of the time in both frequency bands, regardless of the CSI measurement window (i.e., 5 or 10 seconds). Second, we challenge *Next2You* by introducing mobility. Specifically, the provers are deployed as shown in Figure 34a, whereas the verifier is carried by a user. The user continuously moves inside one of three offices or a hallway, approaching office doors from the hallway or walking close to a thin wall separating adjacent offices. We note that the neural network models for 2.4 GHz and 5 GHz bands that are deployed on the mobile verifier have been trained on the CSI data collected by stationary devices, as described in Section 6.5.1.

We challenge Next2You by evaluating the mobile use case based on CSI training data from only stationary devices.

Table 19 shows the FAR and FRR performance of *Next2You* in the case of mobility for each location, frequency band, and time window. We see that FAR when the verifier moves in the hallway varies from 0.111 to 0.009, decreasing with the higher frequency

19 Nexus devices have automatic gain control (AGC) enabled, thus power change is not reflected in the CSI, cf. https://github.com/seemoo-lab/mobisys2018_nexmon_channel_state_information_extractor/issues/2#issuecomment-384088517.

Table 19: FAR and FRR of *Next2You* for 2.4 GHz and 5 GHz bands in the case of mobility.

Location	2.4 GHz				5 GHz			
	5 sec		10 sec		5 sec		10 sec	
	FAR	FRR	FAR	FRR	FAR	FRR	FAR	FRR
Hallway	0.111	n/a	0.083	n/a	0.016	n/a	0.009	n/a
Office 3	0.075	0.203	0.054	0.085	0.000	0.329	0.000	0.298
Office 2	0.196	0.262	0.131	0.220	0.128	0.274	0.065	0.326
Office 1	0.148	0.382	0.190	0.308	0.027	0.238	0.000	0.200

band and longer time window. Exploring the misclassified provers, we find that FAR is caused by devices that either have a line of sight with open office doors or located close to the hallway wall; closing offices' doors reduces FARs to almost zero for both frequency bands and time windows. In Office 3, separated by the hallway, both FAR and FRR steadily decrease with a longer time window for 2.4 GHz. Inspecting FAR and FRR, we discover that the former is caused by devices from opposite Office 2 located near the door, while the latter comes from either heavily obstructed (i.e., window sill) or highest above the floor (i.e., office cabinet) devices. For 5 GHz, FAR remains zero for both time windows, while FRR is relatively high and does not change significantly. We find that a device on top of the office cabinet accounts for over 70% of the FRR, which we attribute to heavier attenuation and reflections due to the moving verifier being blocked by the user as well as the least amount of training data available for devices high above the floor. In adjacent Offices 1 and 2, FAR and FRR follow the above trends but are higher, especially for 2.4 GHz, showing the impact of insufficiently separated environments; FAR is overwhelmingly caused by devices from the adjacent office, while FRR, as before, comes from heavily obstructed and high above the floor devices.

The longer CSI observation window leads to reduced FAR and FRR, especially in 2.4 GHz band.

The above results demonstrate that *Next2You* not only works reliably in real-time but also shows high potential for classifying mobile CSI data despite being trained on the stationary data.

Comparison with Prior Work. We compare the EER performance of *Next2You* with two state-of-the-art audio-based copresence detection schemes from Karapanos et al. [181] and Truong et al. [364]. These schemes are evaluated by Fomichev et al. [98] in very similar to our *office* and *car* scenarios (i.e., adjacent offices, parked cars), allowing for a direct comparison. *Next2You* shows at least 16-times lower EERs compared to the state-of-the-art audio schemes in the case of insufficiently separated environments such as adjacent offices, while in low-entropy context of parked cars *Next2You* preforms several orders of magnitude better (cf. Table 20).

We compare EERs of Next2You with two state-of-the-art schemes.

Table 20: EER comparison between *Next2You* and state-of-the-art copresence detection schemes based on audio in the office and car scenarios (we present best achievable EERs for each scheme).

Scheme	EER (Office)		EER (Car)	
	Full*	Night	Moving	Parked
Next2You (this work)	0.005	0.005	0.001	0.000
Karapanos et al. [181]	0.098	0.090	0.006	0.037
Truong et al. [364]	0.084	0.080	0.111	0.271

*Aggregated performance over several days, including all times of day (i.e., morning till night).

6.5.3 *Generalizability*

We evaluate the capability of *Next2You* to generalize to new application scenarios (e.g., different apartment) by investigating how much effort is required for transfer learning, namely to reuse a pretrained neural network and retrain it with the data of the new environment. As stated in Section 6.3.5, we choose neural networks because of their inherent ability to automatically learn the feature representation. We leverage this property as follows: based on our network structure (cf. Figure 33), we observe that the first two layers (i.e., 500 and 300 neurons, respectively) are mostly dedicated to learning the CSI feature representation, while the last two layers (i.e., 100 and 20 neurons, respectively) are focusing on the classification task. Thus, we can learn the CSI feature representation once on the most comprehensive *office* dataset that contains the largest amount of CSI data, capturing complex geometry and motion, and then reuse this network and retrain it on other scenarios, significantly reducing the computations required for the learning step.

We demonstrate how Next2You can generalize to new use cases by leveraging the transfer learning capability of neural networks.

Specifically, we take the *office* model, which can be efficiently trained on a powerful machine or in the cloud, and set the first two layers to be non-trainable. We then retrain the last two layers of such *office* model on the train set of another scenario (e.g., *house*) and use the obtained model to classify copresence on the test set of this scenario, performing this procedure for each of our scenarios except the *office*. The resulting AUCs are within one percentage point from the AUCs of scenario models trained from scratch (cf. Table 18), confirming that our network architecture allows for feature representation and transfer learning successfully. Since the retrained model is much simpler, we can train for fewer epochs (e.g., we use 10) and reduce the number of floating point operations per second (FLOPS) in the forward training loop of our neural network by a factor of 7 and 13 for 2.4 GHz and 5 GHz bands, respectively. Given that modern end-user devices already perform in the gigaFLOPS [130] range, we consider it feasible to deploy and retrain *Next2You* in new scenarios using the described approach.

As all context-based schemes utilizing machine learning, *Next2You* requires initial data collection when being deployed in a new environment [388].

6.5.4 *Interpretability of Next2You Copresence Detection*

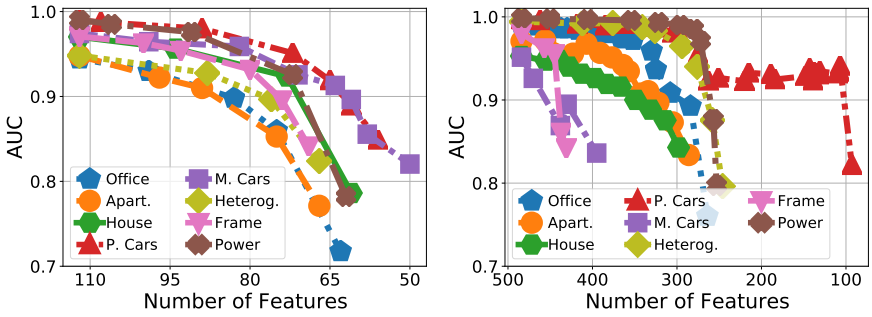
Our results demonstrate that *Next2You* performs well in classifying copresent and non-copresent devices in a variety of scenarios (cf. Table 18), and it generalizes to new environments (cf. Section 6.5.3). However, neural networks might not learn the right hypothesis, yet achieve high classification results for the wrong reasons by overfitting the data [197]. To avoid this, we need to understand which factors play a role in a copresence decision produced by *Next2You*. Since CSI captures the combined scattering, path-loss, shadowing, and multi-path effects [318], which cannot be easily discerned, we identify features in the CSI data contributing to copresence detection and validate our findings against prior work on CSI-based localization and identification.

To interpret the copresence decision-making of *Next2You*, we apply a recently introduced *Right for the Right Reasons (RRR)* method [289], which identifies relevant features used by a neural network in a classification process. Thus, we can not only quantify parts of the CSI determining copresence but also harden *Next2You* against attacks by combining multiple hypotheses learned from CSI data (cf. Section 6.5.5). We utilize the RRR method as follows (details in Section C.2): (1) we train a neural network on our CSI data, obtaining a gradient mask, containing the features considered important by the network, (2) using this mask we find relative importance of these features in the whole training data, (3) we select and penalize features with relative importance above 10%, retraining the network on the same data to make it learn another hypothesis, (4) we repeat the above steps (1)–(3), increasing the number of penalized features until the AUC on the test data drops below 0.85; we choose the 10% feature importance and 0.85 stopping thresholds empirically.

To interpret hypotheses learned by the Next2You neural network, we adopt Right for the Right Reasons approach.

Figure 37 shows the AUC performance of different neural network models produced by the RRR method in 2.4 GHz and 5 GHz bands. We see that multiple models relying on different features can be trained from the CSI data, retaining high AUC, confirming the suitability of CSI for copresence detection. In larger and more complex scenarios (i.e., *office*, *apartment*, and *house*) more features are required to classify copresent and non-copresent devices, while in the smaller and simpler *parked cars*, we may need as few as four features. We observe distinct feature patterns in *heterogeneous* and *power* scenarios. In the former scenario, more features are required to distinguish copresent and non-copresent devices compared to the *office* (two scenarios share a setup, cf. Figure 34a) in both frequency bands. Thus, a neural network likely captures properties of CSI-extracting hardware, which may by used to fingerprint specific types of devices, providing an additional layer of protection. In the latter scenario, the models often rely on 6–8 important features, originating from the CSI of subcarriers uniformly

Applying the RRR method, we see that our neural network bases its copresence decision on various features.



(a) 2.4 GHz frequency band (b) 5 GHz frequency band

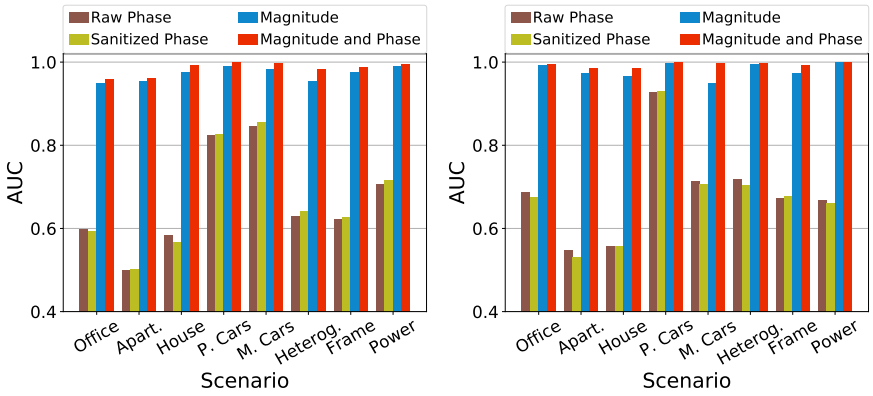
Figure 37: Right for the Right Reasons method applied to the CSI data of different scenarios. A marker denotes a neural network model trained on a different set of features, showing its AUC on the test data.

spread over a Wi-Fi channel, indicating that increased power affects statistical properties of the CSI.

We find that the neural network models in Figure 37 rely on CSI magnitude for all scenarios except the *parked* and *moving cars*, which start to use phases once magnitude features are penalized. In comparable scenarios (e.g., *office* and *house*), the models without penalization use similar important features resulting from the CSI of subcarriers located at the beginning, in the middle, or at the end of a Wi-Fi channel. These subcarriers are relevant because of their stability and low susceptibility to noise [318]. Thus, they can accurately capture a particular effect in the environment such as shadowing due to large objects or scattering caused by window grids. Our findings about the important CSI features for copresence detection agree with prior work. For example, the CSI magnitude is known to be relevant in complex scenarios with many obstacles and human motion [120], while the phases contain too much noise, and thus are not useful [212]. In the environments with shorter distances and fewer obstacles between devices, the phases can be successfully utilized [313], as we see in the *parked* and *moving cars*. We find that the magnitude and phases of adjacent Wi-Fi subcarriers are jointly identified as important features because similar frequencies are likely to be affected by the same phenomenon (e.g., scattering) [318]. These results demonstrate the soundness of the RRR interpretations, confirming that *Next2You* is capable of learning a robust wireless fingerprint of the environment embedded in the CSI.

Our RRR findings about CSI features relevant for copresence detection are in line with prior work.

To quantify the contributions of CSI features to *Next2You* classification performance, we train a neural network on either magnitudes or phases (cf. Figure 38). We apply a phase sanitization method by Sen et al. [313] to overcome the problem of random phase behavior caused by unsynchronized clocks between devices and random noise, training the network on both raw and sanitized phases to allow their comparison. We see that in complex scenarios (e.g., *apartment*) copresence detection based on CSI phases is infeasible, reaching AUCs of around 0.5, indicating that the neural network



(a) 2.4 GHz frequency band

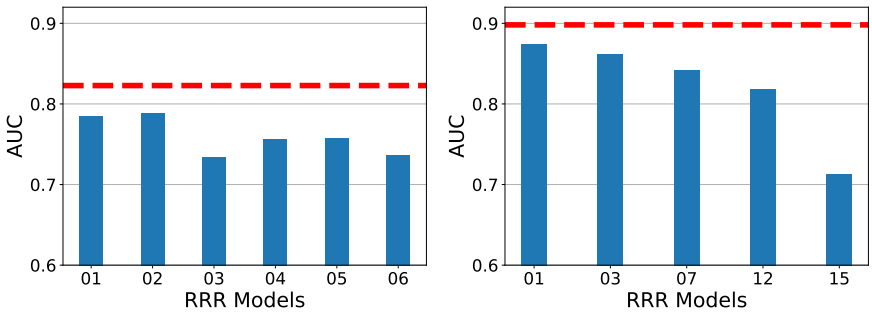
(b) 5 GHz frequency band

Figure 38: AUC of *Next2You* when a neural network is trained on either phase (raw or sanitized) or magnitude features in different scenarios; we also show AUC when the network uses both magnitude and phase.

The CSI magnitude is more relevant for copresence detection, especially in large environments.

classification is no better than a random guess. However, in the simpler environments of *parked* and *moving cars*, the phases become feasible for copresence detection, showing AUCs above 0.8 and 0.9 for 2.4 GHz and 5 GHz bands, respectively. The sanitized phases perform marginally better than the raw phases, which is in contrast with findings of prior work [273, 313]. Such a discrepancy is due to simpler hardware in our experiments (i.e., smartphones), containing a single antenna, which has lower sensitivity and higher susceptibility to noise, compared to multiple antennas in routers used by the prior work. Our phase classification results agree with previous research, showing the higher relevance of 5 GHz phases for localization [379]. The AUC performance based on magnitudes is stable across our scenarios, however using both magnitude and phase features results in consistently higher AUCs (cf. Figure 38). Thus, CSI phases indeed contain relevant copresence information, which can be utilized by a neural network to improve the overall classification performance.

The above results demonstrate the soundness of *Next2You* utilizing CSI and neural networks for copresence detection. We see a direct impact of the environment complexity on the capability of a neural network to make copresence predictions from the CSI data. The fact that the neural network captures distinct properties of diverse CSI data (i.e., heterogeneity, transmission power) confirms its inherent ability for autonomous representation learning and suitability for *Next2You*. We also find that using off-the-shelf devices in real-world scenarios may render existing methods for leveraging CSI inefficient (e.g., phase sanitization), urging the need to conduct experiments in realistic setups with heterogeneous hardware.



(a) Moving car model–parked car data (2.4 GHz)

(b) Office model–parked car data (5 GHz)

Figure 39: AUC of different RRR models that are used to resist the following attacks: (a) an adversary precollects CSI data in a similar car, and (b) they train a neural network model on the rich office CSI and use it to classify copresence in a simpler environment of parked cars. The number of penalized features in the models increases from left to right; the dashed red line is the best achievable AUC without feature penalization; Figure 39b shows a subset of models.

6.5.5 *Advanced Attack Scenarios*

We evaluate how to strengthen Next2You against advanced attacks.

We investigate the robustness of *Next2You* to advanced attacks and propose mitigation strategies. In the first attack, the adversary either precollects CSI data in a similar environment (e.g., car), where legitimate devices execute *Next2You*, or they train a neural network model on a comprehensive dataset (e.g., *office*) and use this model to predict copresence on the CSI data of simpler scenarios (e.g., *parked cars*). We find that such threats are feasible allowing the adversary to predict copresence with relatively high AUCs between 0.8 and 0.9. To mitigate this attack, we utilize multiple neural network models, each learning a different hypothesis, generated by the RRR method (cf. Section 6.5.4).

Combining several neural network models obtained with the RRR method, mitigates the attacker who precollects CSI data in a similar environment.

Figure 39a and Figure 39b depict the AUC performance of different RRR models from Figure 37. The former, shows the models trained on the *moving cars* CSI and tested on the *parked car* data for 2.4 GHz, while the latter—the models trained on the *office* CSI and tested on the *parked cars* data for 5 GHz. We see that the RRR models with penalized features perform 12–25 AUC percentage points better when applied to the legitimate data compared to the adversarial data (cf. Figure 37 vs. Figure 39). In the adversarial case, the AUC drops from the model top-performing (cf. dotted red line in Figure 39) more rapidly than in the legitimate case. Thus, the above attack can be mitigated by training a number of the RRR models and combining their majority vote with thresholding for the final copresence decision (e.g., at least three models must have AUC above 0.9).

Table 21: AUC performance of *Next2You* under the increased power attack for 2.4 GHz and 5 GHz bands.

CSI data used for training (included samples)	Area Under the Curve (AUC)	
	2.4 GHz	5 GHz
Office + Power (all samples)	0.9781	0.9967
Office + Power (no high-power samples)	0.8165	0.8825
Office + Power (10% of high-power samples)	0.9313	0.9948

In the second attack, the adversary who knows that the neural network is trained on a single power level, which is typical for legitimate devices, increases the transmission power of their devices. We implement this attack by training our neural network on the CSI data combined from the *office* and *power* scenarios; in the latter scenario, we exclude high-power samples corresponding to non-copresent devices. Thus, we evaluate the capability of our neural network to classify the unseen high-power CSI data.

Table 21 shows that under the power attack, the performance of *Next2You* drops by 16 and 11 AUC percentage points for 2.4 GHz and 5 GHz, respectively. We find that devices in adjacent Offices 1 and 2 and near open doors (cf. Figure 34a) are mostly affected by such an attack, accounting for the AUC drop. The fact that only neighboring devices in such a challenging scenario (i.e., insufficient separation between Offices 1 and 2) are vulnerable to this attack suggests its limited scope. To mitigate the attack, we tried applying different RRR models with penalized features, however, we observe a similar sharp drop in AUC from the top-performing model, as in Figure 39. This indicates that the neural network simply lacks enough cues to distinguish copresent and non-copresent devices under such an attack. We discuss which other data, in addition to CSI, can improve the robustness of *Next2You* in Section 6.6. To alleviate the effect of the increased power attack, we find that including only 10% of high-power samples brings classification performance very close to the baseline (cf. first row in Table 21). This clearly helps improve the performance of the classifiers, as it provides more data close to the boundaries of the classes. We highlight the utmost importance of training neural networks considering adversarial samples to harden *Next2You*.

We see that classifying increased power CSI data from nearby non-copresent devices is challenging.

6.6 DISCUSSION

In this section, we provide relevant discussion points for *Next2You*.

Robustness. Our evaluation shows that *Next2You* provides reliable copresence detection with error rates below 4% and 1.5% for 2.4 GHz and 5 GHz bands, respectively (cf. Table 18). We demonstrate the soundness of using CSI and neural networks for copresence detection (cf. Section 6.5.4), and find that *Next2You* can be hardened by training a number of neural network models, each relying on a different set of features, and

combining their predictions (cf. Section 6.5.5). However, *Next2You* might be vulnerable to the increased power attack if the neural network is trained without considering the advanced capabilities of the adversary.

To improve the robustness of *Next2You*, a number of physical layer metrics can be used in addition to CSI. For example, the signal-to-noise ratio (SNR) estimated directly in a wireless chipset should capture a unique noise pattern in the environment of copresent devices. Also, as shown by Won et al. [387], CSI power levels are useful for traffic classification. In *Next2You*, these levels can be used to detect the increased power attack (cf. Section 6.5.5). In addition, power delay profiles (PDPs), namely the squared magnitudes of the CIR, can be used as an additional input to a neural network, improving copresence detection accuracy by combining features from time (i.e., PDP or CIR) and frequency (i.e., CSI) domains. Furthermore, devices running *Next2You* can measure the energy on different Wi-Fi channels to make copresence detection more reliable. Specifically, the energy sensed by copresent devices on a given Wi-Fi channel depends on its busyness in their environment (i.e., amount of traffic, other wireless devices using the channel in this environment). This energy will differ from measurements made by non-copresent devices, providing additional cues for differentiation. With several antennas, a direction from which a signal arrives can be determined, as we detail in the following discussion point. We also see that the temporal properties of CSI (i.e., behavior over time) increase the reliability of copresence detection (cf. Section 6.5.2).

We can further strengthen Next2You by supplementing CSI with SNR and PDP metrics.

Leveraging CSI Phase. Our results demonstrate that the CSI phase is less relevant for copresence detection than the magnitude, especially in complex environments with many obstacles and a large amount of motion. Prior research uses multiple antennas to remove a phase offset by finding the difference between phases received on different antennas, or obtain the relative phase by subtracting phases of two successively received frames [222]. The former approach only works if devices have multiple antennas, however as we discuss in the next point, it is unlikely that commodity IoT devices will receive them. The latter approach requires high packet rates (e.g., hundreds of packets per second) to provide sufficient CSI granularity while degrading fast in performance with reduced rates [313, 318]. However, such high packet rates are impractical on IoT devices running on batteries. The accurate phase estimation allows obtaining Angle-of-Arrival (AoA) and Time-of-Flight (ToF) [222], which have the potential for increasing the robustness of *Next2You* to advanced attacks.

To leverage CSI phase, the devices require multiple antennas.

Deployment Considerations. To use *Next2You*, devices performing copresence detection should be capable of extracting CSI. In recent years, several CSI extracting tools have emerged [129, 142, 308, 390], enabling various devices such as routers, laptops, and smartphones with this capability. Utilizing these tools, security researchers discover severe vulnerabilities in proprietary firmwares of popular Wi-Fi chipsets [54, 228], highlighting the necessity for open-source wireless stacks, which will provide better security and functionality (e.g., finer-grained control over a wireless chipset). Thus,

we are positive that an increasing number of devices will receive the CSI-extraction capability, facilitating the deployment of *Next2You*.

We expect more devices to receive CSI-extracting capability in near future.

Another deployment consideration is using *Next2You* on devices that have a different number of antennas, namely single or multiple. As we discover in Section 6.5.2, the CSI of one and two antenna devices varies, hindering the deployment of *Next2You*, if we cannot disable the second antenna and its spatial stream. This again urges the importance of fine-grained control over wireless chipsets. For conventional 2.4 GHz and 5 GHz Wi-Fi, we do not expect low-power IoT devices (e.g., thermostat, smart lock) to have multiple antennas because of the size and energy constraints (i.e., two antennas require more processing), however end-user devices such as smartphones, laptops, and routers will likely feature two or more antennas [395]. For higher frequency Wi-Fi based on mm-waves, even simple devices should be equipped with multiple antennas [305].

Our results demonstrate that Next2You applies to typical flat, office, and car environments.

Copresence Distance. We consider devices to be copresent if they are located inside the same (office) room or car, which is commonly assumed by context-based copresence detection schemes [98, 181, 365]. Our empirical results demonstrate that *Next2You* reliably detects copresence in rooms of up to 5×6 meter's size and in typical passenger cars (e.g., hatchback). However, we see that environments larger than 5×6 meter in size might become problematic for the 5 GHz band. Also, the presence of prominent obstacles in the environment (e.g., sizeable wardrobe set, moving walls) can reduce the copresence detection accuracy of *Next2You*. On the other hand, increasing the transmission power will extend the applicability range of *Next2You*. Overall, *Next2You* suits typical environments where copresence detection is applied, however its parameters (e.g., used frequency band, transmission power) need to be adjusted for specific use cases.

6.7 SUMMARY

We recap contributions of this chapter.

In the age of the Internet of Things (IoT), the demand for secure and usable authentication systems is on the rise. Context-based copresence detection enables such systems, allowing one device to verify the proximity of another device based on their physical context (e.g., audio), eliminating user interaction. In this chapter, we propose *Next2You*, a robust context-based copresence detection scheme utilizing channel state information (CSI). As its main contribution, *Next2You* provides reliable copresence detection, including the challenging cases of low-entropy context (e.g., empty room with few events occurring) and insufficiently separated environments (e.g., adjacent rooms), and it does not require devices to have common sensors such as microphones. We implement and evaluate *Next2You* in five real-world scenarios, demonstrating its high classification accuracy in distinguishing copresent and non-copresent devices, the capability of working in real-time, and resilience to various attacks. The obtained error rates below 4% show that *Next2You* outperforms the state-of-the-art copresence detection schemes, and it is deployable on off-the-shelf devices such as smartphones.

Part III

CONCLUSIONS

7

CONCLUSIONS

In this thesis, we have investigated how to improve security, usability, and deployability of zero-interaction pairing (ZIP) and zero-interaction authentication (ZIA) schemes. We have achieved this goal in two steps: (1) we analyzed theoretically and empirically state-of-the-art ZIP and ZIA schemes, identifying their major shortcomings in Chapter 3 and Chapter 4 and (2) we addressed these shortcomings developing new ZIP and ZIA schemes in Chapter 5 and Chapter 6, respectively.

We conducted theoretical and empirical analysis of existing ZIP and ZIA schemes, and developed two novel schemes.

While conducting theoretical analysis of existing pairing schemes, including ZIP, we found that the current design flow of pairing schemes starting from hardware interfaces leads to incomparability of the schemes. Furthermore, we identified that unrealistic security assumptions about out-of-band (OoB) channels (e.g., confidentiality of data transmission), which are the cornerstone of pairing schemes, resulted in practical attacks against many schemes. Towards this end, we proposed a consistent terminology and new system model for designing pairing schemes based on physical channels, human-computer interaction (HCI) channels, and application classes. Using this system model, we surveyed existing pairing schemes, including ZIP, revealing their security and usability weaknesses. We also suggested ways to address these weaknesses, enabling the development of more robust pairing schemes.

Our theoretical analysis of pairing schemes, including ZIP, revealed avenues for future research.

As a next step, we decided to compare the security and usability performance of state-of-the-art ZIP and ZIA schemes under realistic conditions, finding that it is infeasible due to the lack of context data on which these schemes were validated and the unavailability of their implementations. Hence, we reproduced five state-of-the-art ZIP and ZIA schemes [181, 243, 310, 323, 364] from the ground up and collected various context data in three real-world scenarios, using a representative number of heterogeneous sensing devices, distributed inside our scenarios to reflect a realistic Internet of Things (IoT) setting. We found that four out of the five reproduced schemes struggled under realistic conditions, showing significantly lower security and usability than originally reported. In addition, we discovered that many ZIP and ZIA schemes were sensitive to heterogeneous hardware and user mobility, limiting their deployability in real settings. To ensure that future ZIP and ZIA schemes can be compared against our findings, we released the collected context data and the codebase that we used for data collection along with schemes' implementations. Furthermore, we shared our experiences from this study such as building a reliable data collection platform in realistic scenarios, processing a large amount of context data, and subsequently releasing it as well as challenges in reproducing existing ZIP and ZIA schemes (cf. Appendix B).

We demonstrated that real-world evaluation of ZIP and ZIA schemes is crucial to pave the way for their practical deployment.

Releasing context data on which ZIP/ZIA schemes are validated and their source code is crucial for further development in the domain.

In the second part of the thesis, we addressed the above shortcomings, proposing a new architecture for ZIP and a novel copresence detection method for ZIA. To improve on ZIP, we rethought the design followed by the state-of-the-art schemes, namely replacing the fuzzy commitments cryptographic primitive [175] commonly used by these schemes with a better primitive called Fuzzy Password-Authenticated Key Exchange (fPAKE) [75] that provides stronger security against brute-force offline attacks on a shared secret key and allows reducing pairing time, which improves the usability of ZIP. We demonstrated how to leverage fPAKE for ZIP and implemented this protocol, benchmarking it on off-the-shelf IoT devices. In addition to fPAKE, we introduced sensor fusion as a mechanism to strengthen context against predictable context attacks (e.g., context replay) and further assist fPAKE in shortening the pairing time. We evaluated our new ZIP scheme called *FastZIP*, combining the fPAKE protocol and sensor fusion for in-car device pairing, by collecting real-world driving data using off-the-shelf devices, demonstrating the stronger security of *FastZIP* and its up to three times faster pairing time compared to state-of-the-art ZIP schemes. To facilitate future research, we make the context data used to evaluate *FastZIP*, its codebase, and implementation of the fPAKE protocol publicly available.

Our developed ZIP and ZIA schemes outperform state-of-the-art counterparts.

To improve on ZIA, we devised a novel copresence detection scheme named *Next2You* that is built upon a robust sensor modality: channel state information (CSI) and neural networks. Through our real-world experiments using off-the-shelf smartphones, we demonstrated that *Next2You* achieves more accurate copresence detection (thus higher security) than state-of-the-art solutions, especially in challenging cases of low-entropy context (e.g., empty room with few events occurring) and insufficiently separated environments (e.g., adjacent rooms). Furthermore, *Next2You* provides improved deployability, as it only requires devices to be equipped with Wi-Fi, which is ubiquitous in IoT devices, as compared to dedicated sensors (e.g., microphones) needed by the state-of-the-art copresence detection schemes. We make the collected CSI dataset along with the source code of *Next2You* and data collection tools available for future research.

We envision three major directions for further research in the ZIP/ZIA domain.

Based on the insights gained in this thesis, we have identified three avenues for the future development of ZIP and ZIA schemes. First, our *FastZIP* findings and the results of prior ZIA schemes [323, 364] demonstrate that sensor fusion can prevent advanced attacks and reduce pairing/authentication time. In Chapter 5, we discussed that the proliferation of integrated sensors facilitates the use of sensor fusion on various smart devices. In addition to an inertial measurement unit (IMU) (accelerometer, gyroscope, magnetometer), camera (light and RGB sensors), and wireless chipset (Wi-Fi, Bluetooth), we witness a rapid increase of integrated environmental sensors measuring ambient characteristics of the surroundings (i.e., temperature, barometric pressure, and humidity) [34] or air quality (i.e., CO_2 and total volatile organic compound (TVOC)) [314]. Moreover, the integrated sensors become more affordable [201], facilitating their wide spread. With *FastZIP*, we have shown how the multi-sensor context can be captured inside a moving car, while other research demonstrates that IMU sensors (i.e., ac-

celerometer, gyroscope, magnetometer) can be leveraged to jointly record events such as door/window opening and closing in smart homes [146, 226, 407] as well as various user activities (e.g., walking, typing) in the case of wearables [229, 309, 388]. Second, we foresee that off-the-shelf IoT actuators (e.g., voice assistant, robot vacuum cleaner) can be utilized to inject context stimuli (e.g., sound) into an environment such as a room in a random fashion, thwarting active attacks against ZIP and ZIA schemes as well as increasing context entropy, which can be leveraged to shorten paring/authentication time and render predictable context attacks (e.g., context replay) infeasible. Third, machine learning is already the backbone of many ZIA schemes [229, 317, 323, 364, 365], and it has recently been utilized to speed up a key generation process in ZIP [388]. We envision that one step further would be the use of artificial intelligence (AI) that has been advancing to many fields [295]. Specifically, utilizing AI can improve the adaptiveness of ZIP and ZIA schemes to varying use cases (e.g., in-car vs. in-bus pairing) and challenging environmental conditions (e.g., profound noise) by automatically adapting schemes' parameters (e.g., quantization, threshold values) accordingly.

Part IV

APPENDIX

A

REPRODUCED ZERO-INTERACTION SECURITY SCHEME

In this appendix, we provide more details about the reproduced zero-interaction security (ZIS) schemes. We give a brief overview of each scheme's functionality and use case and describe the implementation of context features utilized by the scheme.

AUDIO PREPROCESSING: Before computing audio features, we aligned audio recordings from different sensing devices as follows. At the beginning of data collection all devices were synchronized using the Network Time Protocol (NTP). First, we performed the coarse-grained alignment using devices' timestamps to synchronize the start of the audio recordings. Second, during the feature computation, we performed a fine-grained alignment between two input audio recordings using the cross-correlation function in Matlab [359]. Specifically, we considered the first hour of audio recordings to find a lag between them, using the *xcorr* function ($maxlag = 3$ seconds), then we used this lag to align two audio recordings, and cut them to the length of the shortest recording. These aligned recordings are then split into intervals and used to compute audio features.

In the mobile scenario, we increased the $maxlag$ to 15 seconds to more precisely find the lag between audio recordings of heterogeneous devices. In addition, we found that heterogeneous devices have an inherent audio drift, causing desynchronization of audio recordings. We removed this drift by applying a time-stretching effect to audio recordings in the *Audacity* tool (change Tempo).

A.1 KARAPANOS ET AL.

The scheme by Karapanos et al. [181] calculates a similarity score between snippets of ambient audio from two devices to decide if these devices are colocated. The similarity score is the average of the maximum cross-correlations between two audio snippets computed on a set of one-third octave bands. To prevent erroneous authentication when audio activity is low, a power threshold is applied to discard similarity scores from audio snippets with insufficient average power. The similarity score is then checked against a fixed similarity threshold to decide if two devices are colocated. The scheme is designed to provide colocation evidence between a user's smartphone and a computer with a running browser. This evidence is utilized as a second authentication factor when a user wants to log-in to an online service such as a bank account. In this work, we focus on computing and comparing similarity scores and do not target the specific use case of the second authentication factor.

Table 22: Notations used by Karapanos et al.

Notation	Explanation
x, y	input audio snippets
L	length of input audio snippets in seconds
l_{max}	max cross-correlation lag in seconds
r	sampling rate of input audio snippets in kHz
τ_{dB}	average power threshold in dB
B	set of considered one-third octave bands
n	number of considered one-third octave bands
$S_{x,y}$	similarity score

Table 23: Parameters of the sound similarity algorithm used in [181] (highlighted) and our implementations.

L, sec	l_{max}, sec	r, kHz	τ_{dB}, dB	B (n)
3	0.15	44.1	40	50Hz – 4kHz (20)
5	1	16	40/38/35	50Hz – 4kHz (20)
10	1	16	40/38/35	50Hz – 4kHz (20)
15	1	16	40/38/35	50Hz – 4kHz (20)
30	1	16	40/38/35	50Hz – 4kHz (20)
60	1	16	40/38/35	50Hz – 4kHz (20)
120	1	16	40/38/35	50Hz – 4kHz (20)

We first provide notations adopted from the original paper in Table 22. Second, we present parameters of the sound similarity algorithm used in the original and our implementations in Table 23. Our goal was to follow the original implementation as close as possible, however, we introduced a few changes, as we did not have tight synchronization between audio snippets. Third, we present our implementation of the sound similarity algorithm in Section A.1.1.

As shown in Table 23, our implementation differs with respect to these parameters from the implementation by Karapanos et al. First, we increase both the length of input audio snippets L from 3 to 5 seconds and the length of the maximum cross-correlation lag l_{max} from 0.15 to 1 second to achieve a comparable level of authorization to the authors. We observed that even after the alignment procedure (cf. *Audio preprocessing*), there might be an offset within long audio recordings (24 hours), which can affect synchronization between audio snippets. That is why, we set $l_{max} = 1$ to maintain a balance between security (l_{max} thwarts attackers trying to guess the audio environment)

and non-tight synchronization, which can happen in a realistic Internet of Things (IoT) scenario. The increase of l_{max} leads to the increase of the audio snippet length L to 5 seconds. Second, we use a lower sampling rate for the input audio snippets r: 16 vs. 44.1 kHz, which does not affect the sound similarity algorithm itself, but can be used to speed up the computations, as a smaller number of samples needs to be processed. Despite the lower sampling rate, and thus narrower audio spectrum (8 kHz), we cover the same set of octave bands as the original implementation.

As stated in Section 4.3, we evaluate the performance of the scheme on a number of intervals from 5 to 120 seconds.

A.1.1 *Implementation of the Sound Similarity Algorithm*

0. As input, we have two aligned audio snippets x and y of equal length L with a sampling rate r.

1. Both x and y are split into n one-third octave bands using a bank of band-pass filters:
$$\begin{aligned}(x_{B_1}, \ldots, x_{B_n}) &= \text{BP\_filter\_bank}(x)\\ (y_{B_1}, \ldots, y_{B_n}) &= \text{BP\_filter\_bank}(y)\end{aligned} \tag{1}$$
Table 24 shows the used one-third octave bands B from 50 Hz to 4 kHz, and each band-pass filter is constructed as a 20th-order Butterworth filter [358] with cut-off frequencies $[F_l, F_h]$.

2. For each x_{B_i} and $y_{B_i} \forall i \in [1, n]$, the normalized maximum cross-correlation $\hat{C}_{x,y}(l)$ is computed as the function of the lag $l \in [0, l_{max}]$ (we omit B_i indexes for simplicity):
$$\hat{C}_{x,y}(l) = \max_l(|C'_{x,y}(l)|) = \max_l\left(\left|\frac{C_{x,y}(l)}{\sqrt{C_{x,x}(0) \cdot C_{y,y}(0)}}\right|\right) \tag{2}$$
In Equation 2, the term $C_{x,y}(l)$ is a cross-correlation function between two discrete signals x and y:
$$C_{x,y}(l) = \sum_{i=0}^{N-1} x(i) \cdot y(i-l) \tag{3}$$
N is the number of samples in the signals[1], and the lag is bounded within a range $l \in [0, N-1]$. The normalization term $\sqrt{C_{x,x}(0) \cdot C_{y,y}(0)}$ accounts for different amplitudes of signals x and y, with $C_{x,x}(0)$ and $C_{y,y}(0)$ being the auto-correlation functions. The resulting maximum cross-correlation is bounded within a range $\hat{C}_{x,y}(l) \in [0, 1]$, because we take the absolute value of the normalized cross-correlation $|C'_{x,y}(l)|$.

1 We assume signals x and y have the same length

Table 24: Used one-third octave bands.

Band Number	F_l, Hz	F_c, Hz	F_h, Hz
6	44.194	49.606 (50)	55.681
7	55.681	62.500 (63)	70.154
8	70.154	78.745 (80)	88.388
9	88.388	99.213 (100)	111.362
10	111.362	125.000 (125)	140.308
11	140.308	157.490 (160)	176.777
12	176.777	198.425 (200)	222.725
13	222.725	250.000 (250)	280.616
14	280.616	314.980 (315)	353.553
15	353.553	396.850 (400)	445.449
16	445.449	500.000 (500)	561.231
17	561.231	629.961 (630)	707.107
18	707.107	793.701 (800)	890.899
19	890.899	1000.000 (1000)	1122.462
20	1122.462	1259.921 (1250)	1414.214
21	1414.214	1587.401 (1600)	1781.797
22	1781.797	2000.000 (2000)	2244.924
23	2244.924	2519.842 (2500)	2828.427
24	2828.427	3174.802 (3150)	3563.595
25	3563.595	4000.000 (4000)	4489.848

F_l - lower band frequency, Fc - calculated center frequency (nominal frequency), F_h - upper band frequency.

3. The resulting similarity score between two audio snippets x and y is obtained by taking the average of the normalized maximum cross-correlations computed in each one-third octave band:

$$S_{x,y} = \frac{1}{n} \sum_{i=1}^{n} \hat{C}_{x_{B_i}, y_{B_i}}(l) \tag{4}$$

The similarity score is only used if the input audio snippets have sufficient average power: $\overline{P}_x, \overline{P}_y > \tau_{dB}$. Otherwise, it is discarded, and no authentication is attempted.

Table 25: Notations used by Schürmann and Sigg.

Notation	Explanation
S	input audio snippet
l	length of the input audio snippet in seconds
r	sampling rate of the input audio snippet in kHz
n	number of frames to split the input audio snippet
m	number of frequency bands to split each frame
d	length of each frame in seconds (duration)
b	width of each frequency band in Hz
f	binary fingerprint of length $(n-1)\cdot(m-1)$ in bits

A.2 SCHÜRMANN AND SIGG

The scheme by Schürmann and Sigg [310] computes a binary fingerprint from a snippet of ambient audio based on energy differences in successive frequency bands. Two devices wishing to establish pairing, compute such fingerprints from their ambient environments. These fingerprints are used in a fuzzy commitment scheme to obtain a shared secret. One device uses its fingerprint to hide a randomly chosen secret and sends this commitment to the other device, which can only retrieve the random secret from the commitment if it has a sufficiently similar fingerprint. In this work, we focus on deriving and comparing binary fingerprints, and we do not target a specific use case of establishing a shared secret key.

We first provide notations adopted from the original paper in Table 25. Second, we present parameters of the audio fingerprinting algorithm used in the original and our implementations in Table 26, where we introduce a few changes, as our audio snippets have a lower sampling rate. Third, we present our implementation of the audio fingerprinting algorithm in Section A.2.1.

As shown in Table 26, our implementation differs with respect to some parameters from the implementation by Schürmann and Sigg. First, we use a lower sampling rate r of 16 kHz instead of the original 44.1 kHz, which affects the number of frequency bands m we can split our frames n into. With a 16 kHz sampling rate, our audio spectrum is only 8 kHz, thus we can only obtain 32 non-overlapping frequency bands, each of width 250 Hz b. Having 32 frequency bands instead of 33, as in the original implementation, results in shorter binary fingerprints f of 496 instead of 512 bits. Second, we vary the lengths l from 5 to 120 seconds, which also affects the length of a single frame d, which varies between 0.29 and 7.06 seconds. We note that shorter audio frames (e.g., $d = 0.29$) are more susceptible to synchronization issues between input audio snippets, thus reducing the similarity of binary fingerprints generated from these snippets. However,

Table 26: Parameters of the audio fingerprinting algorithm used in [310] (highlighted) and our implementations.

f, bits	l, sec	r, kHz	n, frames	m, bands	d, sec	b, Hz
512	6.375	44.1	17	33	0.375	250
496	5	16	17	32	~0.29	250
496	10	16	17	32	~0.59	250
496	15	16	17	32	~0.88	250
496	30	16	17	32	~1.76	250
496	60	16	17	32	~3.53	250
496	120	16	17	32	~7.06	250

starting from $l = 10$, our frame length d is bigger than in the original implementation, which makes our results comparable and allows us to access the performance of the scheme (i.e., distinguishing between colocated and non-colocated devices) on longer audio snippets.

A.2.1 *Implementation of the Audio Fingerprinting Algorithm*

0. As input, we have an audio snippet S of length l with a sampling rate r (audio snippets from different devices are aligned). The number of frames n and the number of frequency bands m are selected to obtain the binary fingerprint of the desired length:

$$L_f = (n-1) \cdot (m-1) \tag{1}$$

   The width of a frequency band depends not only on the number of bands but also on the available audio spectrum, which is limited by the Nyquist frequency ($f_N = \frac{r}{2}$):

$$b = \frac{\text{maxfreq}(S) - \text{minfreq}(S)}{m} \tag{2}$$

1. The audio snippet S is split into n successive frames $F_1, \ldots, F_n$ of equal length $d = r \cdot \frac{l}{n}$ in samples (F_i is a $d \times 1$ vector).

2. Each frame $F_1, \ldots, F_n$ is split into m non-overlapping frequency bands of width b using a bank of band-pass filters:

$$(F_{B_1}, \ldots, F_{B_m})_i = \text{BP\_filter\_bank}(F_i), \quad \forall i \in [1, n] \tag{3}$$

   In our implementation, the available audio spectrum is 8 kHz, thus we split it into the following 32 bands of width 250 Hz: $B_1 = [1, 250], B_2 = [251, 500], \ldots, B_m = [7751, 7999]$, using a 20th-order Butterworth filter [358] for each band.

3. For each frame $F_1, \ldots, F_n$, the energy of each frequency band $B_1, \ldots, B_m$ is computed as (superscript T denotes transpose):

$$(E_{B_j})_i = (F_{B_j}^T \cdot F_{B_j})_i, \quad \forall i \in [1, n]; \forall j \in [1, m] \tag{4}$$

4. The results of energy computation are stored in the energy matrix ($\forall i \in [1, n]; \forall j \in [1, m]$):

$$E_{i,j} = \begin{pmatrix} E_{F_1,B_1} & E_{F_1,B_2} & \cdots & E_{F_1,B_m} \\ E_{F_2,B_1} & E_{F_2,B_2} & \cdots & E_{F_2,B_m} \\ \vdots & \vdots & \ddots & \vdots \\ E_{F_n,B_1} & E_{F_n,B_2} & \cdots & E_{F_n,B_m} \end{pmatrix} \tag{5}$$

5. The binary fingerprint f is obtained by iterating over consecutive frames $\forall i \in [1, n-1]$ and frequency bands $\forall j \in [1, m-1]$. Each bit of the fingerprint is generated by checking the energy difference between successive frequency bands of two consecutive frames ($\forall k \in [1, L_f]$):

$$f_k = \begin{cases} 1, & (E_{i+1,j} - E_{i+1,j+1}) - (E_{i,j} - E_{i,j+1}) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

A.3 MIETTINEN ET AL.

The scheme by Miettinen et al. [243] is inspired by the audio fingerprinting scheme proposed by Schürmann and Sigg (cf. Section A.2) but works on longer timescales. It uses noise level and luminosity measurements to derive long-term binary fingerprints, which can defend against adversaries that are colocated for short timeframes. The scheme utilizes such fingerprints in a fuzzy commitment scheme (as described in Section A.2) to gradually evolve a shared secret key to achieve pairing between two devices that are colocated for a sustained period of time. In this work, we focus on deriving and comparing long-term binary fingerprints, and we do not target a specific use case of establishing a shared secret key.

We first provide notations adopted from the original paper in Table 27. Second, we present parameters of the context fingerprinting algorithm used in the original and our implementations in Table 28. Our goal was to follow the original implementation as close as possible, however, we introduced a few changes, as we use audio with a higher sampling rate to generate noise levels. We discuss the effect of those changes on the parameters of the context fingerprinting algorithm. Third, we present our implementation of the context fingerprinting algorithm in Section A.3.1.

As shown in Table 28, our implementation differs with respect to these parameters from the implementation by Miettinen et al. [243]. First, we use audio with a higher

Table 27: Notations used by Miettinen et al.

Notation	Explanation
w	length of the context snapshot in seconds
f	a new snapshot is recorded every f seconds
r	sampling rate of recorded audio in kHz
m_w	measurement window in seconds
Δ_{rel}	relative threshold for fingerprint generation
Δ_{abs}	absolute threshold for fingerprint generation

Table 28: Parameters of the context fingerprinting algorithm used in [243] (highlighted) and our implementations.

w, sec	f, sec	r, kHz	m_w, sec	Δ_{rel}	Δ_{abs}
120	120	8	0.1/1	0.1	10
5	5	16	1	0.1	10
10	10	16	1	0.1	10
15	15	16	1	0.1	10
30	30	16	1	0.1	10
60	60	16	1	0.1	10
120	120	16	1	0.1	10

sampling rate r: 16 vs. 8 kHz to generate noise levels. The noise levels are generated by averaging absolute amplitudes of audio samples over m_w seconds, given by the measurement window. Thus, for $m_w = 1$, we obtain one noise level from 16000 audio samples, whereas the original implementation computes one noise level from only 8000 audio samples, which makes our noise levels more fine-grained. The original implementation uses two different measurement windows m_w: 0.1 and 1 sec. The shorter measurement window speeds up the fingerprint generation but may be susceptible to synchronization issues, thus we opt for a longer measurement window. For luminosity measurements, we do not use the measurement window. We collect luminosity readings at 10 samples per second and use all samples generated during context snapshot length w to obtain the fingerprint. Second, we evaluate the context fingerprinting algorithm on the context snapshots of different lengths w from 5 to 120 seconds. Thus, we can assess the performance of the scheme (i.e., distinguishing between colocated and non-colocted devices) on shorter context snapshots.

A.3.1 *Implementation of the Context Fingerprinting Algorithm*

0. As input, we have sets of noise level S_{nl} and luminosity S_{lux} measurements generated from context information collected in our scenarios (i.e., car and office) as stated above. The number of bits b in the resulting context fingerprints is given by $\frac{|S_{nl}|}{f}$ and $\frac{|S_{lux}|}{f}$, where $|\cdot|$ denotes the set cardinality.

1. The *context snapshot* c_w for a timeslot t consists of all measurements m taken in the timeslot of w seconds, $c_w(t) = (m_i, m_{i+1}, \ldots, m_{i+n})$. For each context fingerprint, the average value $\overline{c}(t)$ is computed as:

$$\overline{c}(t) = \frac{\sum_{m_i \in c(t)} m_i}{|\{m_i \in c(t)\}|} \tag{1}$$

2. Each set of measurements (S_{nl} or S_{lux}) can be represented as a sequence of context snapshots $C(t, t+nf) = (c(t), c(t+f), \ldots, c(t+nf))$. Then, the fingerprint bit $b(t_i)$, which corresponds to each snapshot $c(t_i)$, is generated as:

$$b(t_i) = \begin{cases} 1, & \left|\frac{\overline{c}(t_i)}{\overline{c}(t_i - f)} - 1\right| > \Delta_{rel} \wedge \left|\overline{c}(t_i) - \overline{c}(t_i - f)\right| > \Delta_{abs} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

   We note that the values for Δ_{rel} and Δ_{abs} (cf. Table 28) are not given in the original paper but were provided by the authors in private communication.

3. The resulting fingerprint for the set of measurements (S_{nl} or S_{lux}) is obtained as:

$$\phi(C(t, t+nf)) = (b(t), b(t+f), \ldots, b(t+nf)) \tag{3}$$

To avoid using fingerprints that are exclusively zero in times of low ambient noise and light, Miettinen et al. proposed an extension to their system: they propose computing the *surprisal* of a fingerprint before using it. The surprisal of a single bit b of the fingerprint is defined as its self-information I, measured in bits:

$$\sigma(b) = I(b) = -\log_2(P(B = b)) \tag{4}$$

The surprisal of the whole fingerprint F is the sum of the surprisal of its individual bits:

$$\sigma(F) = \sum_{b \in F} \sigma(b) \tag{5}$$

Calculating this surprisal requires knowledge about how often bits occur at specific positions of the fingerprint during specific times of the day, indicated as $P(B = b)$

in the formula. Miettinen et al. do not state the time resolution, but it is implied that the probabilities are tracked on a per-hour basis. For the office scenario, which covers multiple days, we track the probabilities independently for the individual days, i.e., fingerprints generated on weekdays do not influence the probabilities, and thus surprisals for the weekend.

Miettinen et al. propose setting a surprisal threshold σ_{thr} that the surprisal of a fingerprint has to exceed in order to be considered valid for pairing. This avoids the problem of attacks by adversaries guessing the low-entropy fingerprints generated at night. The threshold is defined as:

$$\sigma_{thr} = t + \sigma_{marg} \tag{6}$$

t denotes the number of incorrect bits the fuzzy commitment will tolerate, and σ_{marg} denotes an extra security margin. However, the authors do not state how this margin should be chosen. Our margin choice is described in Section 4.3.3

A.4 TRUONG ET AL.

The scheme by Truong et al. [364] uses Wi-Fi, Bluetooth, GPS, and ambient audio collected by two devices to compute a number of context features, which are then fed into a machine learning classifier that outputs a prediction if these devices are colocated. This scheme is designed to provide colocation evidence to thwart relay attacks on wireless channels between a user's device and a terminal, which employs zero-interaction authentication (ZIA) (e.g., unlock a computer if a user's smartphone is nearby). In this work, we focus on computing context features and obtaining classification results from the machine learning algorithms, and we do not target the specific use case of thwarting relay attacks.

We first provide notations adopted from the original paper in Table 29. Second, we describe how different context features are computed. Third, we provide details of our machine learning methodology, where we discuss our datasets, the parameters of machine learning algorithms that we use, and the evaluation procedure.

Due to a lack of GPS support in the used hardware, we were unable to collect GPS information. However, since our office scenario is static, and the car scenario mostly considers geographically close cars, the information value of the GPS features would have been low. In addition, the original authors report that the GPS feature contains the least amount of discriminative power in their dataset.

A.4.1 *Non-audio Features*

The features for Wi-Fi, Bluetooth, and GPS are defined over a number of sets. Individual samples for each context information are defined as a tuple (m, s), where m denotes the identifier of the observed beacon (i.e., BLE MAC address, Wi-Fi BSSID), and s

Table 29: Notations used by Truong et al.

Notation	Explanation
$m_i^{(a)}$	identifier of the ith beacon observed by device a
$s_i^{(a)}$	signal strength of ith beacon observed by device b
θ	value substituted for missing signal strengths
S_a	set of records sensed by device a
n_a	number of different beacons observed by device a
$S_\cap$	beacons seen by a and b
$S_\cup$	beacons seen by a or b, θ substituted for missing s
x, y	input audio snippets
L	length of input audio snippets in seconds
r	sampling rate of input audio snippets in kHz

denotes the received signal strength. The set of records observed by devices a and b is denoted as S_a and S_b, respectively, while n_a and n_b denote the number of unique beacons observed by the devices. The notation is also given in Table 29. Given these preconditions, the following sets are defined:

$$
\begin{aligned}
S_a &= \{(m_i^{(a)}, s_i^{(a)}) \mid i \in \mathbb{Z}_{n_a - 1}\} \\
S_b &= \{(m_i^{(b)}, s_i^{(b)}) \mid i \in \mathbb{Z}_{n_b - 1}\} \\
S_a^{(m)} &= \{m \; \forall (m, s) \in S_a\} \\
S_b^{(m)} &= \{m \; \forall (m, s) \in S_b\} \\
S_\cap &= \{(m, s^{(a)}, s^{(b)}) \; \forall m \mid (m, s^{(a)}) \in S_a, \; (m, s^{(b)}) \in S_b\} \\
S_\cup &= S_\cap \cup \{(m, s^{(a)}, \theta) \; \forall m \mid (m, s^{(a)}) \in S_a, \; m \notin S_b^{(m)}\} \\
&\quad \cup \{(m, \theta, s^{(b)}) \; \forall m \mid (m, s^{(b)}) \in S_b, \; m \notin S_a^{(m)}\} \\
S_\cap^{(m)} &= \{m \; \forall m \mid (m, s^{(a)}, s^{(b)}) \in S_\cap\} \\
S_\cup^{(m)} &= \{m \; \forall m \mid (m, s^{(a)}, s^{(b)}) \in S_\cup\} \\
L_a^{(s)} &= \{s^a \mid (m, s^{(a)}, s^{(b)}) \in S_\cap\} \\
L_b^{(s)} &= \{s^b \mid (m, s^{(a)}, s^{(b)}) \in S_\cap\}
\end{aligned}
$$

Truong et al. uses these sets to define a total of six features, five of which we implement: (1) the Jaccard Distance $J_{a,b}$, (2) mean Hamming distance $\overline{H}_{a,b}$, (3) Euclidean distance $E_{a,b}$, (4) mean exponential of difference $\overline{Exp}_{a,b}$, (5) the sum of squared ranks

$S^{(sr)}_{a,b}$. The sixth feature, subset count, is only used for the GPS data, and thus omitted. The features are given by the following formulas, where θ is specific to certain context information.

$$J_{a,b} = 1 - \frac{|S^{(m)}_{\cap}|}{|S^{(m)}_{\cup}|} \tag{1}$$

$$\overline{H}_{a,b} = \frac{\sum_{k=1}^{|S_{\cup}|} \left| s^{(a)}_k - s^{(b)}_k \right|}{|S_{\cup}|} \tag{2}$$

$$E_{a,b} = \sqrt{\sum_{k=1}^{|S_{\cup}|} (s^{(a)}_k - s^{(b)}_k)^2} \tag{3}$$

$$\overline{Exp}_{a,b} = \frac{\sum_{k=1}^{|S_{\cup}|} \exp \left| s^{(a)}_k - s^{(b)}_k \right|}{|S_{\cup}|} \tag{4}$$

$$S^{(sr)}_{a,b} = \sum_{k=1}^{|S_{\cap}|} (r^{(a)}_k - r^{(b)}_k)^2 \tag{5}$$

$|\cdot|$ denotes the set cardinality; $r^{(a)}_k$ ($r^{(b)}_k$) is the rank of $s^{(a)}_k$ ($s^{(b)}_k$) in the set L_a (L_b) sorted in ascending order.

For Wi-Fi, all features are used. The signal strength s for each observed identifier is set to the average observed signal strength for that identifier over all included scans. θ, which is substituted as signal strength for devices that have been observed by one but not the other device, is set to -100. For Bluetooth Low Energy (BLE), features 1 and 3 are used, once again, using the average observed signal strength for each identifier as s and $\theta = -100$.

In case both sensors observe no beacons, the distances are not defined, and the original paper does not specify a behavior for this case. In private communication, the authors recommended choosing either zero (if the system should be biased towards accepting) or a very high number (if it should be biased towards rejecting). In our case, we chose to replace undefined values with the distance 10 000 to bias the system towards rejecting when in doubt.

A.4.2 *Audio Features*

Truong et al. use two audio features: (1) the maximum cross-correlation and (2) time-frequency distance computed on snippets of ambient audio of length $L = 10$ seconds.

The authors do not provide the sampling rate r of their audio snippets; in our implementation $r = 16$ kHz. We compute these context features on audio snippets of different lengths L from 5 to 120 seconds. In the end, we create and evaluate two different datasets for machine learning, one using $L = 10$, the other $L = 30$.

In the following, we explain how the maximum cross-correlation and time-frequency distance are computed. We note that this information is not available in the original paper and was obtained via private communication with the authors.

0. As input, we have two aligned audio snippets x and y of equal length L with a sampling rate r.

1. x and y are normalized as (superscript T denotes transpose):

$$x' = \frac{x}{\sqrt{x^T \cdot x}} \qquad y' = \frac{y}{\sqrt{y^T \cdot y}} \tag{6}$$

   Here, the denominator represents a square root of the signal's energy.

2. The *maximum cross-correlation* between the normalized audio snippets x' and y' is computed as (we omit prime superscripts in $\hat{C}_{x,y}(l)$ for simplicity):

$$\hat{C}_{x,y}(l) = \max(|C_{x,y}(l)|) = \max\left(\left|\sum_{i=0}^{N-1} x'(i) \cdot y'(i-l)\right|\right) \tag{7}$$

   $|\cdot|$ denotes the absolute value, N is the number of samples in audio snippets, and the lag l is set to the default value $2N - 1$ [359]. The resulting maximum cross-correlation is bounded within a range $\hat{C}_{x,y}(l) \in [0, 1]$, because we take the absolute value $|C_{x,y}(l)|$.

3. To compute the frequency distance between audio snippets x and y, a fast Fourier transform (FFT) weighted by a Hamming window is applied:

$$X = \mathrm{FFT}(HW(x)) \qquad Y = \mathrm{FFT}(HW(y)) \tag{8}$$

4. Since the FFT is symmetric, only a half of the FFT values is taken to construct frequency vectors for x and y:

$$X_h = \left|X[1, \frac{L_X}{2}]\right| \qquad Y_h = \left|Y[1, \frac{L_Y}{2}]\right| \tag{9}$$

   $|\cdot|$ denotes the absolute value, L_X and L_Y are lengths of FFT vectors X and Y.

5. Frequency vectors X_h and Y_h are normalized similarly to step (1):

$$X'_h = \frac{X_h}{\sqrt{X_h^T \cdot X_h}} \qquad Y'_h = \frac{Y_h}{\sqrt{Y_h^T \cdot Y_h}} \tag{10}$$

6. The frequency distance between audio snippets x and y is given by:

$$D_{f,xy} = \sqrt{\sum \left((X'_h - Y'_h) * (X'_h - Y'_h)\right)} \tag{11}$$

   $*$ denotes element-wise multiplication.

7. The time distance between audio snippets x and y is given by:

$$D_{t,xy} = 1 - \hat{C}_{x,y}(l) \tag{12}$$

8. The *time-frequency distance* between audio snippets x and y is given by:

$$D_{tf,xy} = \sqrt{D^2_{t,xy} + D^2_{f,xy}} \tag{13}$$

A.4.3 *Machine Learning*

After calculating these features over their dataset, Truong et al. used the machine learning suite *Weka* [139] using Multiboost [382] with grafted C4.5 decision trees [381] as weak learners in their evaluation. As Weka does not support large datasets, we chose to use the H2O framework [357] instead. For the training of the classifiers, we set the seed to 1619 and the early stopping to 5 rounds. This means that the training is repeatable when using the same seed and dataset, and the system will consider learning complete once no improvements have been made for five iterations. We let H2O train a set of independent models and perform a hyperparameter search to optimize the parameters (e.g., number of trees in the random forest) for the dataset, maximizing the cross-validated Area Under the Curve (AUC). Afterwards, we select the top performing model and determine its Equal Error Rate (EER) as described in Section 4.3.4.

A.5 SHRESTHA ET AL.

The scheme by Shrestha et al. [323] utilizes ambient temperature, humidity, pressure, and precision gas collected by two devices to compute a number of context features, which are then fed into a machine learning classifier that outputs a prediction if these devices are colocated. Similarly to Truong et al., this scheme addresses relay attacks by providing colocation evidence between two devices involved in ZIA. In this work, we focus on computing context features and obtaining classification results from the machine learning algorithms, and we do not target a specific use case of thwarting relay attacks.

We first provide notations adopted from the original paper in Table 30. Second, we describe how different context features are computed. Third, we provide details of our machine learning methodology, where we discuss our datasets, the parameters of machine learning algorithms that we use, and the evaluation procedure.

Table 30: Notations used by Shrestha et al.

Notation	Explanation
$s_a^{(k)}$	sample of context information k by device a
$D_{a,b}^{(k)}$	distance between samples of devices a and b

Due to a lack of hardware support, we were unable to collect precision gas, and thus omit this context feature.

A.5.1 *Context Features*

The authors convert ambient pressure P in millibars to altitude in meters using the following formula before computing context features:

$$h_{altitude} = \left(1 - \left(\frac{P_{station}}{1013.25}\right)^{0.190284}\right) * 145366.45 * 0.3048 \quad (1)$$

For each of the considered context information (ambient temperature, humidity, and altitude), the context feature is given by the absolute difference between two samples of context information collected devices a and b at time t:

$$D_{a,b}^{(k)} = |s_a^{(k)} - s_b^{(k)}| \quad (2)$$

A.5.1.1 *Machine Learning*

The resulting distances are passed to a Multiboost classifier [382], with random forests [35] as a weak learner, using Weka [139]. The process for machine learning is identical to that described in the previous section.

A.6 STUDY DESIGN

Table 31 presents hardware used to collect context information in car, office, and mob/het scenarios.

Table 32 contains a description of the device deployment in the car scenario, while Table 33 contains the mapping for the office scenario, and Table 34 shows device locations in the mob/het scenario.

Figure 40 shows the route the cars took during the car scenario (cf. Section 4.2.2). The route covers city traffic, country roads, and highways between the cities of Darmstadt and Frankfurt in the state of Hesse in Germany (the actual GPS traces can be found in [97]).

Table 31: Sensing devices used for data collection.

Sensor type	Sensing device (sampling rate)			
	TI SensorTag CC2650 + Raspberry Pi 3 + Samson Go	Samsung Galaxy S6	Samsung Gear S3	RuuviTag+
Audio	16 kHz	16 kHz	16 kHz	–
Barometric pressure	10 Hz	5 Hz	10 Hz	10 Hz
Humidity	10 Hz	–	–	10 Hz
Luminosity	10 Hz	5 Hz	10 Hz	–
Temperature	10 Hz	–	–	10 Hz
BLE beacons	0.1 Hz	0.1 Hz	0.1 Hz	–
Wi-Fi beacons	0.1 Hz	0.1 Hz	0.1 Hz	–
Accelerometer	10 Hz	50 Hz	50 Hz	–
Gyroscope	10 Hz	50 Hz	50 Hz	–
Magnetometer	10 Hz	50 Hz	50 Hz	–

– = sensor not available.

Table 32: Device deployment in the car scenario.

Car 1 Device	Device location	Car 2 Device
01	Dashboard	07
02	Glove compartment	08
03	Between front seats	09
04	Right back handhold	10
05	Left back handhold	11
06	Trunk	12

Table 33: Device location mapping in the office scenario.

Office 1		Office 2		Office 3	
Dev.	Location	Dev.	Location	Dev.	Location
01	Near Wi-Fi access point (*h*)	09	Screen of User 2 (*m*)	17	Wall behind Users 2 and 3 (*h*)
02	Window sill (*m*)	10	Window sill (*m*)	18	Window sill (*m*)
03	Above door to Office 2 (*h*)	11	Above door to Office 1 (*h*)	19	Lamp above User 1 (*h*)
04	Lamp above User 1 (*h*)	12	Lamp above User 1 (*h*)	20	Screen of User 1 (*m*)
05	Screen of User 1 (*m*)	13	Right screen of User 1 (*m*)	21	Screen of User 2 (*m*)
06	Screen of User 2 (*m*)	14	Left screen of User 1 (*m*)	22	Screen of User 3 (*m*)
07	In the cupboard (*h*)	15	In the cupboard (*l*)	23	Shelf next to the door (*m*)
08	Wall next to the door (*h*)	16	Shelf left of the door (*m*)	24	In the cupboard (*h*)

h = high position; *m* = medium position; *l* = low position.

Table 34: Device location mapping in the mob/het scenario, initial configuration.

Office 1		Office 2		Office 3	
Dev.	Location	Dev.	Location	Dev.	Location
01	Screen of User 1 (*m*)	11	Screen left from User 3 (*m*)	18	Screen in front of User 4 (*m*)
02	Screen of User 2 (*m*)	12	Screen of User 3 (*m*)	19	Screen of User 4 (*m*)
03	Near a power plug (*l*)	13	Near a power plug (*l*)	20	Near a power plug (*l*)
04	On top of robot station (*l*)	14	Near a fan (*l*)	21	On top of coffee machine (*m*)
05	Smartphone of User 1†	15	Smartphone of User 3†	22	Smartphone of User 4†
06	Smartwatch of User 1†	16	Smartwatch of User 3†	23	Smartwatch of User 4†
07	Laptop of User 1†	17	Laptop of User 3†	24	Laptop of User 4†
08	Smartphone of User 2†				
09	Smartwatch of User 2†				
10	Laptop of User 2†				
25	Smartphone on top of robot†				

h = high position; *m* = medium position; *l* = low position; † = mobile device.

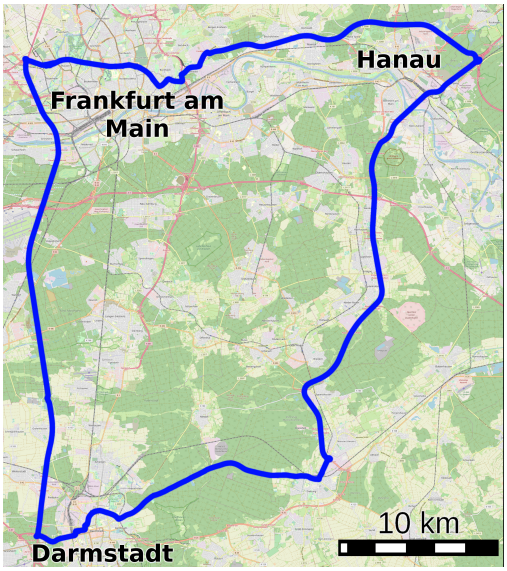


Figure 40: Route driven in the car scenario.

B

PRACTICAL CHALLENGES IN EVALUATING ZERO-INTERACTION SECURITY SCHEMES

In this appendix, we illustrate the difficulties that we faced in our study by first describing the challenges encountered when *reproducing* five zero-interaction security (ZIS) schemes such as the absence of the source code and documentation, ambiguities, and unspecified parameters. Second, we elaborate on challenges faced in the *data collection* starting from building a realistic setup to issues hindering reliable data collection such as power, connectivity, and fault tolerance. Third, we describe our experience in *processing collected data* such as ensuring its quality (e.g., identifying erroneous data) and working with large datasets. Finally, we outline best practices we follow for *data release*.

B.1 REPRODUCING PUBLISHED ALGORITHMS

Researchers commonly differentiate between *reproducibility* (being able to rerun the same code on the same data and obtain the same results) and *replicability* (being able to write a new implementation of the proposed algorithm from scratch, running it on the same data, and obtaining the same results) [23]. In our study [98], we find both to be impossible, as none of the papers published their source code or data, which is common practice in computer science research [57]. After requesting access to code and data via e-mail, one team of authors provided us with their code (but no data), and another team provided us with their data. In the latter case, we were still unable to reproduce their results using the machine learning tool *Weka* that the authors had employed, likely due to different default parameters of machine learning algorithms in different versions of *Weka*, a problem anticipated by Benureau and Rougier [23]. The other authors did not respond to our requests or denied us access to the code and data due to intellectual property and privacy concerns.

We could neither reproduce nor replicate the results of existing ZIS schemes due to the lack of source code and data.

This left us with the task of reimplementing all five schemes from scratch, based on the information given in the publication. This effort was hampered by ambiguous descriptions of parts of the algorithms, underspecified behavior for edge cases and, in some cases, missing values for system parameters (e.g., threshold values, sampling rates). We resolved these issues and validated our interpretations of the algorithms in communication with the original authors. However, due to the lack of original datasets, we were unable to replicate the results from the original papers.

To allow for a fair comparison between the five schemes, we decided to collect our own dataset, which will be described in the next section.

B.2 DATA COLLECTION, PROCESSING, AND RELEASE

We describe issues in data collection, processing, and release.

In this section, we first describe our experiences in building a reliable data collection platform, highlighting major issues we faced when deploying it in realistic environments. We then elaborate on challenges when processing our large dataset, and finally, we summarize the main points to consider when releasing data and code. It goes without saying that for large data collection studies involving human subjects (e.g., audio data collection) an institutional review board (IRB) approval needs to be sought, which we recommend doing well in advance, as the process may take several months in complex cases.

B.2.1 *Data Collection*

Sensing devices need to be deployed in realistic spots.

In our study [98], we collect data from three scenarios: connected car, smart office, and mobile. In each scenario, we deploy multiple sensing devices to represent realistic Internet of Things (IoT) environments—each device is placed in a spot reflecting potential IoT functionality such as under the ceiling (e.g., smart light) or inside a trunk (e.g., smart sensor). Each scenario differs in terms of types of sensing devices used, their mobility, and duration of data collection, posing different challenges. In the car scenario, we equip two cars with six homogenous static devices each, collecting data during a four-hour trip. In the office scenario, we equip three offices with eight homogenous static devices each and collect data for one week. In the mobile scenario, we use heterogeneous sensing devices, both statically deployed and carried by users, and collect data for eight hours. Overall, we use four types of sensing devices: a Raspberry Pi 3 with attached TI SensorTag and a Samson Go USB microphone (Pi+Tag+Mic), a Samsung Galaxy S6 smartphone, a Samsung Gear S3 smartwatch, and a RuuviTag. Figure 41 shows examples of sensing devices deployed in our scenarios. In the following, we elaborate on power, connectivity, fault tolerance, and testing issues experienced during data collection, and solutions to these issues found.

B.2.2 *Power*

Powering stationary and mobile sensing devices poses different challenges.

While smartphones, smartwatches, and RuuviTags have built-in batteries, customized sensing devices such as Pi+Tag+Mic, need an external power supply, accommodating the power consumption of both the attached peripherals and processes running on the main board. We empirically find that a power supply rated below 2.1A current (at 5V) leads to unpredictable behavior of Pi+Tag+Mic devices, hindering reliable data collection, and emphasizing the need to carefully choose the power supply for customized sensing devices.

In the office scenario, we use Pi+Tag+Mic devices, which needed to run for one week non-stop, making the use of mains supply an obvious choice. However, we could not use

(a) Pi+Tag+Mic (b) Samsung S6 (c) RuuviTag

Figure 41: Sensing devices deployed: (a) Pi+Tag+Mic on a wall powered over PoE, (b) Samsung S6 on top of a robot vacuum cleaner, (c) RuuviTag on the upper arm.

AC power adapters due to their insufficient cable length, as many devices were placed in spots (e.g., under the ceiling) without power sockets in proximity. Thus, we consider two other alternatives to deliver power over 5–10 meter distances: (1) USB cables and (2) Power over Ethernet (PoE) supplied via external adapters. While the former is much easier to deploy, we find that USB cables experience substantial voltage drops, resulting in insufficient power delivered to Pi+Tag+Mic devices, hindering reliable data collection. In contrast, PoE does not suffer from voltage drops and additionally brings connectivity to the devices (cf. Section B.2.3), making us favor this option.

Long USB cables are subject to substantial voltage drop.

In the car scenario, we also use Pi+Tag+Mic devices; however, the shorter duration of data collection, absence of mains supply, and difficulties using bulky cables to deploy sensing devices inside a car motivated the use of portable power banks (10000 mAh). We utilize the same power banks in the mobile scenario to power a user-carried Raspberry Pi 3, capturing data from RuuviTags, showing that portable power supplies are suitable for several-hour data collections conducted with mobile devices, or when delivering power to static devices over wires is impractical.

B.2.3 *Connectivity*

In data collections, devices often need different types of connectivity (e.g., to a core network or between each other) for purposes such as time synchronization, monitoring, and data delivery. In our scenarios, all sensing devices required access to the Internet to perform the Network Time Protocol (NTP) update, facilitating the synchronous start of data collection and correct timestamping of sensor readings. We find that zone-specific NTP servers rate limit requests from the same network, so if the number of devices is above 20, we recommend either setting up a dedicated NTP server or spreading NTP requests over time.

Interference from neighboring devices hinders transmission of sensor readings via Bluetooth.

Compared to a one-time NTP update, data delivery and data collection monitoring require permanent connectivity between the devices. In realistic environments, connectivity is affected by interferences caused by sensing devices themselves and neighboring

devices communicating in the same frequency band. In our scenarios, we observe connectivity drops between Wi-Fi and Bluetooth devices communicating in the 2.4 GHz band, which we attribute to the overloaded spectrum. This caused occasional drops of the Bluetooth link between a SensorTag and Raspberry Pi, permanently terminating sensor data delivery from the SensorTag. We see similar connectivity drops between sensing devices and a Wi-Fi access point (AP), hindering the use of Wi-Fi to remotely access devices for monitoring. These examples show that wireless connections should not be assumed reliable in realistic environments, and wires should be used instead if reliable connectivity is critical.

B.2.4 *Fault Tolerance*

Fault tolerance is indispensable to ensure reliable data collection. In realistic environments with distributed sensing devices, it is important to monitor liveness of data collection. The easiest way is to remotely access the devices, however, this can either be infeasible (cf. Section B.2.3) or undesirable (e.g., security/privacy concerns), making visual inspection a viable alternative. In smartphones and smartwatches, visual inspection is easy to implement due to available user interfaces, however, customized sensing devices such as Pi+Tag+Mic often lack user interfaces, making the use of LEDs imperative to visually monitor liveness of data collection. We leverage this observation by shutting down Pi+Tag+Mic devices (LEDs go off—easy to notice) in cases of critical data collection errors.

Fault tolerance is crucial to enable seamless data collection for a prolonged time.

Liveness detection can be coupled with recovery procedures, increasing the reliability of data collection. For example, a connectivity drop between a SensorTag and Raspberry Pi (cf. Section B.2.3) terminated the process, fetching data from the SensorTag. Thus, we introduced a watchdog process, continuously monitoring the data-fetching process, and restarting it if the process is terminated.

In a distributed setup with many devices, it makes sense to implement a scheduled start-up of data collection. If the data collection is interrupted (e.g., Pi+Tag+Mic powers off), a manual intervention becomes unavoidable. To facilitate a seamless restart of data collection, two points need to be considered: first, the restart must be fast, requiring minimum user interaction such as unplugging a device, and plugging it back in or relaunching the data collection app; second, the data collected before the interrupt must be saved separately and should not be overwritten by newly collected data.

B.2.5 *Testing*

Testing a data collection platform carefully allows identifying many error cases and ensuring reliability during the real experiment. Here, we outline three crucial points for testing derived from our experience, providing concrete examples of the encountered problems and pitfalls to avoid.

First, a data collection platform must be tested for realistic deployment time, corresponding to the duration of actual data collection. This allows identifying memory leaks, durability of power supplies, and file size limitations—we empirically find a 4 GB WAV size limit imposed by the standard, making us adopt the FLAC format instead. Second, a data collection platform must be thoroughly tested in the exact environment it will be deployed in. For example, we undertested our platform in running cars before collecting data in them, resulting in incorrectly chosen microphone settings, ruining audio recordings, as they become saturated by the engine hum, and necessitating a repeat of the experiment.

Testing should be performed (1) for realistic time, (2) in a comparable environment, and (3) with the same number of devices as the final experiment.

Third, a data collection platform must be tested with a realistic number of devices, running under realistic loads. Using this principle, we identify a number of problems related to interference and overloaded 2.4 GHz spectrum. For example, before opting for Samson Go microphones, we tried several more affordable alternatives, all of which suffered from interference caused by communicating SensorTags, making the quality of audio recordings unacceptable. In the case of the overloaded spectrum, we find that Wi-Fi captures (i.e., scanning visible APs) crash on Raspberry Pi and Samsung S6 devices, freezing the whole Wi-Fi interface, indicating that there might be a serious flaw in the Wi-Fi stack of Linux-based devices.

B.2.6 *Data Processing*

Regarding data processing, we elaborate on two main points: first, ensuring the quality of collected data, and second, dealing with large datasets.

Having collected the data, one must ensure its quality, which can be affected by devices stopping recording (and manually restarted), faulty sensor readings, and sampling drift. If the restart of data collection is properly implemented, the sensor data only needs to be stitched, which is straightforward for most sensor modalities except audio. If an audio recording is not terminated correctly, the resulting audio file may become corrupted due to missing file headers. We experienced such cases using binary hex editors to manually craft audio file headers, completely restoring the audio recording; for stitching audio recordings we utilize *Audacity*.

We use binary hex editors to restore headers of corrupted audio files.

Sometimes sensors deliver erroneous readings, which need to be identified and excluded. To do so, we plotted the collected sensor data and visually inspected it. This turns out to be a very powerful tool to spot outliers (cf. Figure 42) and missing data. This type of sanity checks suffices for most sensor modalities except audio. In audio recordings made by heterogeneous devices, we observe non-negligible sampling drift (cf. Figure 43) caused by internal clock offsets of different devices, despite the synchronous start of audio recordings. To remedy this, we find the lag between heterogeneous audio recordings and apply the time-stretching effect in *Audacity*.

Plotting sensor data helps spot outliers and missing data.

Similar to prior research [241], we observe *sensor bias* among heterogeneous devices, most notably in barometric pressure readings. We also find *sampling rate instability*

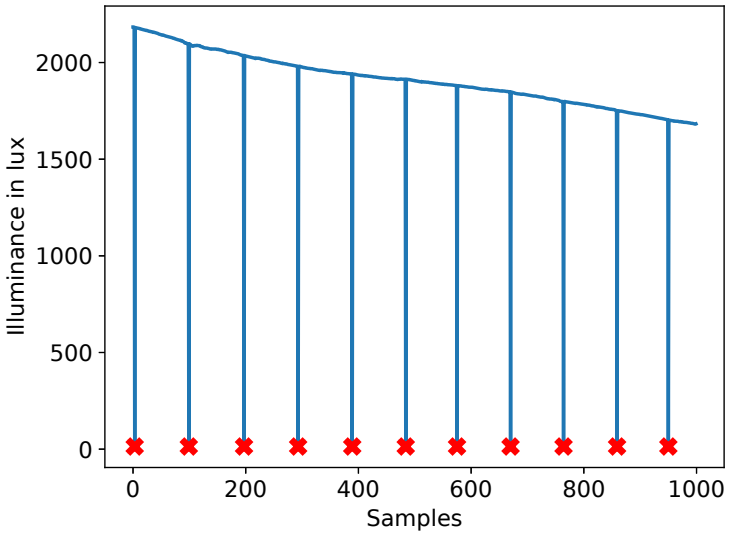


Figure 42: Example of faulty illuminance values being periodically delivered by SensorTag.

Sensor bias and sampling rate instability hinder reliable data collection.

among smartphones of the same model, especially for IMU sensors (tested with Galaxy S6, Nexus 5X, and Nexus 6P), with the actual sampling rate deviating by up to 10% from the set value. The sampling rate instability can hinder data collection, for example, light sensor readings on Nexus 5X and Nexus 6P miss the advertised sampling rate (i.e., 5 Hz) by a wide margin, going as low as 0.2 Hz and, thus preventing us from using these smartphones in our study. The sampling rate instability is less prominent on smartwatches and insignificant on SensorTags. This demonstrates that although modern smartphones contain powerful sensors, they can be unsuitable for scientific measurements [342].

After sanity checking the collected data, we ended up with the dataset of 239 GB to compute on. To deal with such large data in a reasonable time, we follow known best practices, which are often overlooked.

To process large data, we (1) ensure computational resources, (2) use caching, and (3) apply compression.

First, we ensure sufficient resources with access to our institution's high-performance cluster. To leverage full potential of the cluster, the code running on it needs to be customized to work in a highly parallel environment. When running highly parallel computations, one must consider licensing issues if commercial software is used. For example, to work around shortage of Parallel Computing Toolbox licenses in MATLAB, we use the MATLAB compiler, allowing us to create a standalone application, which can be launched royalty-free on an arbitrary number of machines.

Second, we try avoiding redundant operations on large datasets. This can be done using stateful systems such as *Jupyter Notebook*, loading data once and keeping it in

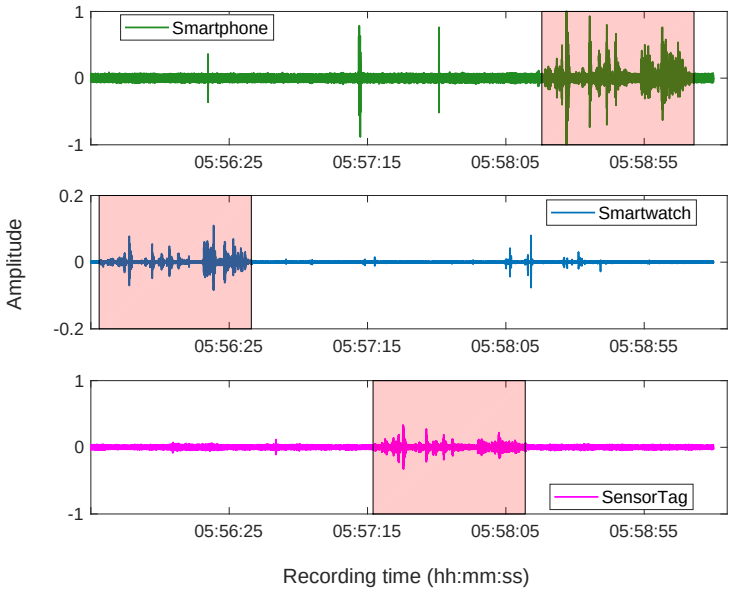


Figure 43: Sampling drift between heterogeneous devices: insignificant at the beginning of audio recording, it becomes prominent over the course of several hours (highlighted regions indicate the same part of audio recording).

RAM afterwards. In the case of time-consuming computations, we recommend liberally caching intermediate results to files, which can significantly speed up evaluation reruns.

Third, we apply compression to reduce storage and computing requirements. Sensor data is often redundant (e.g., slowly changing sensor modalities such as temperature), showing great potential for compression. We leverage this observation by counting and deduplicating identical instances and assigning them weights in the dataset used to train the machine learning classifiers, reducing its size from 81 GB to 600 MB, and decreasing training time by orders of magnitude.

B.2.7 *Data and Code Release*

We anonymize data prior to release.

When releasing a dataset, several issues need to be considered. First and most importantly, the data needs to be sufficiently anonymized to protect the study subjects and comply with legal requirements. If the dataset will be released, this should be communicated clearly to the study participants as part of the informed consent process. We decided to keep the audio recordings from two of our three scenarios private and only make the audio from the third scenario available to others upon request. We take precautions to limit the privacy impact of this release before and during the recording and obtain consent from all involved parties, including our IRB. Wi-Fi and Bluetooth device identifiers are replaced with pseudonyms. The other parts of the dataset are deemed to be non-critical, as they do not contain any sensitive information.

For hosting our dataset, we choose a non-commercial general-purpose repository Zenodo.

Once the data has been cleaned and anonymized, an appropriate data repository needs to be chosen. Prior work discusses this issue in more detail [281]. As no fitting specialized repository for zero-interaction security data exists, we choose the non-commercial general-purpose repository *Zenodo*. To ensure that data can be selectively downloaded, we split our dataset into several parts (i.e., raw data, processed data, and results for each of the scenarios). Where our data exceedes the size limits of Zenodo, we host it on Google Drive and create a *stub* dataset on Zenodo containing a link to the Google Drive folder and a list of file hashes to verify the download. We also create an index dataset that includes links to all individual datasets [97].

The source code is released under open-source license on GitHub.

To release the code, we create a public GitHub repository, choose an Open Source license, and link it to Zenodo to obtain a DOI for it. This allows others to use our reference implementations of the algorithms under test in their own research. Using a version control system like Git also allows us to annotate our result files with the exact version of the code that generated them, as defined by the Git commit identifier, and the hashes of the input files. Together with a list of the exact versions of all libraries, this makes it possible for others to exactly reproduce our results, as recommended by Benureau and Rougier [23].

C

DETAILS ON NEXT2YOU COPRESENCE DETECTION SCHEME

C.1 CSI ACQUISITION IN WI-FI

Let $\mathbf{F} \in \mathbb{C}^{K\times K}$ denote the discrete Fourier transform (DFT) matrix, and $\mathbf{F}^H \in \mathbb{C}^{K\times K}$ its inverse (where $(\cdot)^H$ represents the Hermitian transpose). Further, since $\mathbf{F}$ and $\mathbf{F}^H$ are unitary matrices, then $\mathbf{F}\mathbf{F}^H = \mathbf{I}$. The OFDM symbol is obtained upon modulating the K subcarriers by the BPSK symbols $\mathbf{s}$. The discrete-time complex baseband representation of such OFDM symbol is

$$\begin{aligned} \mathbf{z} &= \mathbf{F}^H\mathbf{s}, \\ &= [z_1, \cdots, z_K]^T \in \mathbb{C}^{K\times 1}. \end{aligned} \tag{3}$$

In Wi-Fi, a cyclic prefix (CP) is appended to the OFDM symbol $\mathbf{z}$, with two objectives. First, the CP helps in combating the inter-symbol interference among adjacent OFDM symbols, caused by multi-path propagation. Second, it simplifies channel equalization at the receiver. Essentially, the CP consists of the last N_{CP} samples of $\mathbf{z}$, which are attached at the beginning of the OFDM symbol $\mathbf{z}$, thus making the transmit signal cyclic. Such extended signal is denoted by

$$\tilde{\mathbf{z}} = \left[z_{K-N_{CP}+1}, \cdots, z_K, z_1, \cdots, z_K\right]^T \in \mathbb{C}^{(K+N_{CP})\times 1}. \tag{4}$$

Let $\tilde{\mathbf{c}} = [\tilde{c}_1, \cdots, \tilde{c}_J]^T \in \mathbb{C}^{J\times 1}$ be the estimated channel impulse response (CIR) as depicted in Figure 31c, such that $J < N_{CP}$. Thus, the received signal is expressed as

$$\tilde{\mathbf{r}} = \tilde{\mathbf{z}} \circledast \mathbf{c}. \tag{5}$$

The operator $\circledast$ denotes circular convolution as a result of $\tilde{\mathbf{z}}$ being cyclic due the CP insertion. The receiver removes the first N_{CP} samples of $\tilde{\mathbf{r}}$ so as to neglect the effect of inter-symbol interference caused by multi-path propagation. Thus, the resulting signal is $\mathbf{r} = [r_1, \cdots, r_K]^T \in \mathbb{C}^{K\times 1}$, which can be equivalently expressed as:

$$\underbrace{\begin{pmatrix} r_1 \\ \vdots \\ r_K \end{pmatrix}}_{\mathbf{r}} = \underbrace{\begin{pmatrix} \tilde{c}_1 & 0 & \cdots & 0 & \tilde{c}_J & \cdots & \tilde{c}_2 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \tilde{c}_J & \tilde{c}_{J-1} & & 0 & 0 & & 0 \\ 0 & c_J & & 0 & 0 & & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \tilde{c}_J & \tilde{c}_{J-1} & \cdots & \tilde{c}_1 \end{pmatrix}}_{\mathbf{C}} \underbrace{\begin{pmatrix} z_1 \\ \vdots \\ z_K \end{pmatrix}}_{\mathbf{z}} + \underbrace{\begin{pmatrix} w_1 \\ \vdots \\ w_K \end{pmatrix}}_{\mathbf{w}}. \tag{6}$$

Here, $\mathbf{w} \sim \mathcal{CN}\left(0, \sigma_w^2 \mathbf{I}\right)$ denotes circularly-symmetric complex Gaussian noise. The receiver demodulates the received signal $\mathbf{r}$ using the DFT matrix $\mathbf{F}$, thus obtaining

$$\begin{aligned} \mathbf{y} &= \mathbf{F}\mathbf{r}, \\ &= [y_1, \cdots, y_K]^T. \end{aligned} \tag{7}$$

Using Equation 6, the demodulated signal can be further expressed as

$$\begin{aligned} \mathbf{y} &= \mathbf{F}\left(\mathbf{C}\mathbf{z} + \mathbf{w}\right), \\ &= \mathbf{F}\mathbf{C}\mathbf{z} + \mathbf{F}\mathbf{w}. \end{aligned} \tag{8}$$

The matrix $\mathbf{C} \in \mathbb{C}^{K \times K}$ is circulant as a consequence of making the transmit signal periodic by means of inserting the CP. In general, circulant matrices can be factorized using eigen-decomposition. Thus,

$$\mathbf{C} = \mathbf{F}^H \mathbf{H} \mathbf{F}. \tag{9}$$

Here, $\mathbf{H} = \text{diag}\left([H_1, \cdots, H_K]\right)$ is a diagonal matrix, where H_k are the eigenvalues of matrix $\mathbf{C}$ [127]. Replacing Equation 9 on Equation 8, the demodulated signal $\mathbf{y}$ collapses to

$$\begin{aligned} \mathbf{y} &= \mathbf{F}\left(\mathbf{F}^H \mathbf{H} \mathbf{F}\right)\left(\mathbf{F}^H \mathbf{s}\right) + \mathbf{F}\mathbf{w}, \\ &= \mathbf{F}\mathbf{F}^H \mathbf{H} \mathbf{F}\mathbf{F}^H \mathbf{s} + \mathbf{F}\mathbf{w}, \\ &= \mathbf{I}\mathbf{H}\mathbf{I}\mathbf{s} + \mathbf{F}\mathbf{w}, \\ &= \mathbf{H}\mathbf{s} + \mathbf{w}. \end{aligned} \tag{10}$$

More specifically,

$$\underbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_K \end{pmatrix}}_{\mathbf{y}} = \underbrace{\begin{pmatrix} H_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & H_K \end{pmatrix}}_{\mathbf{H}} \underbrace{\begin{pmatrix} s_1 \\ \vdots \\ s_K \end{pmatrix}}_{\mathbf{s}} + \underbrace{\begin{pmatrix} w_1 \\ \vdots \\ w_K \end{pmatrix}}_{\mathbf{w}}. \tag{11}$$

Thus, for the k-th subcarrier, we have:

$$\begin{aligned} y_k &= H_k s_k + w_k, \\ &= M_k e^{j\phi_k} s_k + w_k. \end{aligned} \tag{12}$$

This shows that each symbol s_k is affected only by a complex-valued factor $H_k = M_k e^{j\phi_k}$ and additive noise w_k. Since the BPSK symbols $\mathbf{s}$ are known by the receiver, all H_k (i.e. CSI across all the subcarriers of a Wi-Fi frame) can be estimated upon

multiplying each y_k by the factor $\frac{s_k^*}{|s_k|^2}$. Thus, the collection of estimated CSI values is obtained by:

$$\begin{aligned}\widetilde{H}_k &= y_k \frac{s_k^*}{|s_k|^2}, \\ &= M_k e^{j\phi_k} s_k \frac{s_k^*}{|s_k|^2} + w_k \frac{s_k^*}{|s_k|^2}, \\ &= M_k e^{j\phi_k} + w_k.\end{aligned} \tag{13}$$

C.2 BACKGROUND ON THE RIGHT FOR THE RIGHT REASONS METHOD

Before describing the Right for the Right Reasons (Right for the Right Reasons (RRR)[2]) method [289], let us first introduce some notation and discuss the neural network approach to classification. We consider neural networks as parametric $\theta = [\theta_1, \cdots, \theta_W]^T \in \mathbb{R}^{W\times 1}$ functions whose inputs $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]^T \in \mathbb{R}^{N\times D}$ are the features and the outputs $\hat{\mathbf{Y}} = \left[\hat{\mathbf{y}}_1, \cdots, \hat{\mathbf{y}}_N\right]^T \in \mathbb{R}^{N\times Z}$ are classification probabilities of the one-hot-encoded labels $\mathbf{Y} \in \mathbb{R}^{N\times Z} = [\mathbf{y}_1, \cdots, \mathbf{y}_N]^T$, namely $\hat{\mathbf{Y}} = f(\mathbf{X}|\theta)$. More specifically, $\mathbf{x}_n = [x_{n,1}, \cdots, x_{n,D}]$, $\hat{\mathbf{y}}_n = \left[\hat{y}_{n,1}, \cdots, \hat{y}_{n,Z}\right]$ and $\mathbf{y}_n = [y_{n,1}, \cdots, y_{n,Z}]$ (for $n = \{1, \cdots, N\}$), where W is the number of parameters of the neural network, N is the number of training samples, D is the number of features at the input, and Z is the number of outputs.

Training such a network is the process of obtaining the optimal parameters θ, such that the predictions are as accurate as possible. To achieve this, we define the cross-entropy loss function $\mathcal{L}(\theta, \mathbf{X}, \mathbf{Y}) = -\frac{1}{N}\sum_{n=1}^{N} \mathbf{y}_n \bullet \log\left(\hat{\mathbf{y}}_n\right)$ that, given the features, labels, and parameters, estimates the classification error of the network. This loss function might include a regularization term $\mathcal{R}(\theta)$ such as the ℓ_2-norm weighted by a scalar ϕ, namely $\mathcal{R}(\theta) = \phi \sum_i \theta_i^2$. In general, the use of regularization helps to improve the behavior of the network when tested on unseen data, i.e., it facilitates generalization. Luckily, both the loss and network functions are differentiable, and therefore learning is simply the process of optimizing the parameters by minimizing the loss function using a gradient descent approach.

Now that we have a basic description of the standard training approach to neural networks, let us consider two more topics: interpretability and hypotheses space. Baehrens et al. [16] points to an interesting fact about these network functions, specifically, the gradient of the network's output with respect to the input features $\nabla_{\mathbf{x}_n} \hat{\mathbf{y}}_n$ is a vector normal to the decision boundary, and thus serves as a description of the model behavior near $\mathbf{x}_n$. In the RRR method [289], the authors propose to use these gradient vectors as explanations, and they further penalize those input gradients, making the network focus on relevant features and discarding irrelevant ones. With this approach, we can

2 The original source code is available at https://github.com/dtak/rrr.

obtain explanations by finding which features are relevant for the prediction of an instance.

For the penalization term, the authors introduce the annotation matrix $\mathbf{A} \in \{0,1\}^{N \times D}$, which is a binary mask that indicates whether a feature should be relevant or not for a given instance. They then proceed to extend the standard loss functions by introducing a penalty $\mathcal{P}(\mathbf{A}, \nabla_{\mathbf{X}} \hat{\mathbf{y}}) = \sum_{n=1}^{N} \sum_{d=1}^{D} \left(A_{n,d} \cdot \frac{\partial}{\partial x_{n,d}} \sum_{z=1}^{Z} \log\left(\hat{y}_{n,z}\right) \right)^2$ function on the input gradients controlled by a parameter λ, namely $\widetilde{\mathcal{L}}(\theta, \mathbf{X}, \mathbf{y}) = \mathcal{L}(\theta, \mathbf{X}, \mathbf{y}) + \lambda \mathcal{P}(\mathbf{A}, \nabla_{\mathbf{X}} \hat{\mathbf{y}})$. This penalty function $\mathcal{P}(\mathbf{A}, \nabla_{\mathbf{X}} \hat{\mathbf{y}})$ and its influence value λ guide the optimization algorithm to find optimal parameters given the restrictions imposed by $\mathbf{A}$ on the features, while minimizing the prediction error. To understand the parameter λ, let us consider the two extremes: if λ is low, the optimizer focuses only on the predictions, but if λ is high, it will focus on the importance of the features and ignore the quality of the predictions. Here, we use the recommended $\lambda = 1000$ because it keeps the values from the standard loss and the penalty on the same order of magnitude, as suggested by Ross et al. [289]. The abovementioned approach gives us interpretability by quantifying how much each feature contributes to the prediction of the network. However, it also gives us a way to obtain different classification hypotheses.

Each classifier encodes one classification hypothesis, but there might be many different alternative explanations for the classification of a dataset. We can obtain different hypothesis by computing the input gradients to get a magnitude ratio, specifically, we divide the input gradients by the component with the maximum magnitude. We then compute the features per instance above a c threshold, setting it to 0.67 according to the original work [289]. After that, we aggregate the values for all the instances and remove the top most important features. This allows us to obtain different parameters for our neural network architecture that classify the data according to other alternative explanations, as they will not have access to the same input features.

BIBLIOGRAPHY

[1] Michael Adeyeye and Paul Gardner-Stephen. "The Village Telco project: A Reliable and Practical Wireless Mesh Telephony Infrastructure." In: *EURASIP Journal on Wireless Communications and Networking* 2011.1 (2011), pp. 1–11.

[3] Imtiaj Ahmed, Yina Ye, Sourav Bhattacharya, Nadarajah Asokan, Giulio Jacucci, Petteri Nurmi, and Sasu Tarkoma. "Checksum Gestures: Continuous Gestures as an Out-of-band Channel for Secure Pairing." In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2015, pp. 391–401.

[4] Marco Domenico Aime, Giorgio Calandriello, and Antonio Lioy. "Dependability in Wireless Networks: Can We Rely on WiFi?" In: *IEEE Security & Privacy* 5.1 (2007), pp. 23–29.

[5] George T Amariucai, Clifford Bergman, and Yong Guan. "An Automatic, Time-based, Secure Pairing Protocol for Passive RFID." In: *International Workshop on Radio Frequency Identification: Security and Privacy Issues*. Springer. 2011, pp. 108–126.

[6] S Abhishek Anand and Nitesh Saxena. "A Sound for a Sound: Mitigating Acoustic Side Channel Attacks on Password Keystrokes with Active Sounds." In: *International Conference on Financial Cryptography and Data Security*. Springer. 2016, pp. 346–364.

[7] S Abhishek Anand and Nitesh Saxena. "Vibreaker: Securing Vibrational Pairing with Deliberate Acoustic Noise." In: *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. 2016, pp. 103–108.

[8] S Abhishek Anand and Nitesh Saxena. "Noisy Vibrational Pairing of IoT Devices." In: *IEEE Transactions on Dependable and Secure Computing* 16.3 (2018), pp. 530–545.

[9] S Abhishek Anand and Nitesh Saxena. "Speechless: Analyzing the Threat to Speech Privacy from Smartphone Motion Sensors." In: *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2018, pp. 1000–1017.

[10] S Abhishek Anand, Prakash Shrestha, and Nitesh Saxena. "Bad Sounds Good Sounds: Attacking and Defending Tap-based Rhythmic Passwords Using Acoustic Signals." In: *International Conference on Cryptology and Network Security*. Springer. 2015, pp. 95–110.

[11] Shlomi Arnon. *Visible Light Communication*. Cambridge University Press, 2015.

[12] Marcelo Atenas, Sandra Sendra, Miguel Garcia, and Jaime Lloret. "IPTV Performance in IEEE 802.11 n WLANs." In: *2010 IEEE Globecom Workshops*. IEEE. 2010, pp. 929–933.

[13] Aviva. *Tech Nation: Number of Internet-connected Devices Grows to 10 per Home*. https://www.aviva.com/newsroom/news-releases/2020/01/tech-nation-number-of-internet-connected-devices-grows-to-10-per-home/. 2020.

[14] TV-B-Gone. *How Does TV-B-Gone Work?* http://www.tvbgone.com/using-your-tv-b-gone/how-does-tv-b-gone-work/. 2016.

[15] Michael Backes, Tongbo Chen, Markus Dürmuth, Hendrik PA Lensch, and Martin Welk. "Tempest in a Teapot: Compromising Reflections Revisited." In: *2009 30th IEEE Symposium on Security and Privacy*. IEEE. 2009, pp. 315–327.

[16] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. "How to Explain Individual Classification Decisions." In: *The Journal of Machine Learning Research* 11 (2010), pp. 1803–1831.

[17] Dirk Balfanz, Diana K Smetters, Paul Stewart, and H Chi Wong. "Talking to Strangers: Authentication in Ad-hoc Wireless Networks." In: *NDSS*. 2002.

[18] Sourangsu Banerji and Rahul Singha Chowdhury. "On IEEE 802.11: Wireless LAN Technology." In: *arXiv preprint arXiv:1307.2661* (2013).

[19] Álex Barredo. *A Comprehensive Look at Smartphone Screen Size Statistics and Trends*. https://medium.com/@somospostpc/a-comprehensive-look-at-smartphone-screen-size-statistics-and-trends-e61d77001ebe. 2014.

[20] Lawrence Bassham, Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Stefan Leigh, M Levenson, M Vangel, Nathanael Heckert, and D Banks. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. 2010.

[21] Emrah Bayraktaroglu, Christopher King, Xin Liu, Guevara Noubir, Rajmohan Rajaraman, and Bishal Thapa. "Performance of IEEE 802.11 under Jamming." In: *Mobile Networks and Applications* 18.5 (2013), pp. 678–696.

[22] Steven Michael Bellovin and Michael Merritt. "Encrypted Key Exchange: Password-based Protocols Secure against Dictionary Attacks." In: *Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy*. 1992, pp. 72–84.

[23] Fabien CY Benureau and Nicolas P Rougier. "Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions." In: *Frontiers in Neuroinformatics* 11 (2018), p. 69.

[24] Denis Besnard and Budi Arief. "Computer Security Impaired by Legitimate Users." In: *Computers & Security* 23.3 (2004), pp. 253–264.

[25] BestMobile.pk. *Evolution of Samsung Galaxy S series Sensors*. https://www.bestmobile.pk/post/evolution-of-samsung-galaxy-s-series-sensors. 2016.

[26] BlueKrypt. *Cryptographic Key Length Recommendation*. https://www.keylength.com/en/4/. 2020.

[27] Bluetooth SIG, Inc. *The Global Standard for Connection*. https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics. 2021.

[28] Bluetooth, SIG. *Simple Pairing Whitepaper*. https://silo.tips/download/simple-pairing-whitepaper. 2006.

[29] Bluetooth, SIG. *Security, Bluetooth Low Energy*. https://www.bluetooth.com/bluetooth-resources/le-security-study-guide/. 2020.

[30] Stanley J Bolanowski Jr, George A Gescheider, Ronald T Verrillo, and Christin M Checkosky. "Four Channels Mediate the Mechanical Aspects of Touch." In: *The Journal of the Acoustical society of America* 84.5 (1988), pp. 1680–1694.

[31] Connor Bolton, Kevin Fu, Josiah Hester, and Jun Han. "How to Curtail Oversensing in the Home." In: *Communications of the ACM* 63.6 (2020), pp. 20–24.

[32] Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, and Frank Stajano. "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes." In: *2012 IEEE Symposium on Security and Privacy*. IEEE. 2012, pp. 553–567.

[33] Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, and Frank Stajano. "Passwords and the Evolution of Imperfect Authentication." In: *Communications of the ACM* 58.7 (2015), pp. 78–87.

[34] Bosch Sensortec GmbH. *IoT-ready Environmental Sensor BME680 from Bosch*. https://www.bosch-sensortec.com/news/iot-ready-environmental-sensor-bme680.html. 2021.

[35] Leo Breiman. "Random Forests." In: *Machine Learning* 45.1 (2001), pp. 5–32.

[36] Arne Brüsch, Ngu Nguyen, Dominik Schürmann, Stephan Sigg, and Lars Wolf. "Security Properties of Gait for Mobile Device Pairing." In: *IEEE Transactions on Mobile Computing* 19.3 (2019), pp. 697–710.

[37] Bump Technologies. *Bump*. http://bu.mp/. 2014.

[38] Chris Burns. *Which Phones Let Me Control any TV?* http://www.slashgear.com/which-phones-let-me-control-any-tv-24338249/. 2014.

[39] Jerrold T Bushberg and John M Boone. *The Essential Physics of Medical Imaging*. Lippincott Williams & Wilkins, 2011.

[40] Liang Cai, Kai Zeng, Hao Chen, and Prasant Mohapatra. "Good Neighbor: Ad hoc Pairing of Nearby Wireless Devices by Multiple Antennas." In: *NDSS*. 2011.

[41] Carmen Camara, Honorio Martín, Pedro Peris-Lopez, and Muawya Aldalaien. "Design and Analysis of a True Random Number Generator Based on GSR Signals for Body Sensor Networks." In: *Sensors* 19.9 (2019), p. 2033.

[42] Srdjan Čapkun, Mario Čagalj, Ramkumar Rengaswamy, Ilias Tsigkogiannis, Jean-Pierre Hubaux, and Mani Srivastava. "Integrity Codes: Message Integrity Protection and Authentication over Insecure Channels." In: *IEEE Transactions on Dependable and Secure Computing* 5.4 (2008), pp. 208–223.

[43] Maurantonio Caprolu, Savio Sciancalepore, and Roberto Di Pietro. "Short-range Audio Channels Security: Survey of Mechanisms, Applications, and Research Challenges." In: *IEEE Communications Surveys & Tutorials* (2020).

[44] Brent Carrara and Carlisle Adams. "On Acoustic Covert Channels Between Air-gapped Systems." In: *International Symposium on Foundations and Practice of Security*. Springer. 2014, pp. 3–16.

[45] Brent Carrara and Carlisle Adams. "Out-of-band Covert channels—A survey." In: *ACM Computing Surveys (CSUR)* 49.2 (2016), pp. 1–36.

[46] Claude Castelluccia and Gildas Avoine. "Noisy Tags: A Pretty Good Key Exchange Protocol for RFID Tags." In: *International Conference on Smart Card Research and Advanced Applications*. Springer. 2006, pp. 289–299.

[47] Claude Castelluccia and Pars Mutaf. "Shake Them Up! A Movement-based Pairing Protocol for Cpu-constrained Devices." In: *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*. 2005, pp. 51–64.

[215] Check Point Software Technologies Ltd. *Cyber Security Report 2020*. https://www.ntsc.org/assets/pdfs/cyber-security-report-2020.pdf. 2020.

[48] Ke-Yu Chen, Rahul C Shah, Jonathan Huang, and Lama Nachman. "Mago: Mode of Transport Inference Using the Hall-effect Magnetic Sensor and Accelerometer." In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.2 (2017), p. 8.

[49] Kyong-Tak Cho, Yuseung Kim, and Kang G Shin. "Who Killed My Parked Car?" In: *arXiv preprint arXiv:1801.07741* (2018).

[50] Ming Ki Chong, Rene Mayrhofer, and Hans Gellersen. "A Survey of User Interaction for Spontaneous Device Association." In: *ACM Computing Surveys (CSUR)* 47.1 (2014), pp. 1–40.

[51] Delphine Christin, Andreas Reinhardt, Salil S Kanhere, and Matthias Hollick. "A Survey on Privacy in Mobile Participatory Sensing Applications." In: *Journal of Systems and Software* 84.11 (2011), pp. 1928–1946.

[52] Thomas Claburn. *Newsflash: Car Cyber-security Still Sucks*. https://www.theregister.co.uk/2018/01/26/car_hacking_wireless/. 2018.

[53] Jiska Classen, Joe Chen, Daniel Steinmetzer, Matthias Hollick, and Edward Knightly. "The Spy Next Door: Eavesdropping on High Throughput Visible Light Communications." In: *Proceedings of the 2nd International Workshop on Visible Light Communications Systems*. 2015, pp. 9–14.

[54] Jiska Classen and Matthias Hollick. "Inside Job: Diagnosing Bluetooth Lower Layers Using Off-the-shelf Devices." In: *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*. 2019, pp. 186–191.

[55] Jiska Classen, Daniel Wegemer, Paul Patras, Tom Spink, and Matthias Hollick. "Anatomy of a Vulnerable Fitness Tracking System: Dissecting the Fitbit Cloud, App, and Firmware." In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.1 (2018), pp. 1–24.

[56] Jolyon Clulow, Gerhard P Hancke, Markus G Kuhn, and Tyler Moore. "So Near and Yet so Far: Distance-bounding Attacks in Wireless Networks." In: *European Workshop on Security in Ad-hoc and Sensor Networks*. Springer. 2006, pp. 83–97.

[57] Christian Collberg and Todd A Proebsting. "Repeatability in Computer Systems Research." In: *Communications of the ACM* 59.3 (2016), pp. 62–69.

[58] Mauro Conti and Chhagan Lal. "Context-based Co-presence Detection Techniques: A Survey." In: *Computers & Security* (2019), p. 101652.

[59] Tim Cooper and Ryan LaSalle. "Guarding and Growing Personal Data Value." In: *White Paper* (2016).

[60] Mark D Corner and Brian D Noble. "Zero-interaction Authentication." In: *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*. 2002, pp. 1–11.

[61] "Cyber Security Breaches Survey 2020." In: *Computer Fraud & Security* 2020.4 (2020), p. 4.

[62] Pulse Secure Cybersecurity Insiders. *2020 Endpoint and IoT Zero Trust Security Report*. https://www.pulsesecure.net/resource/endpoint-iot-securityreport-infographic/. 2020.

[63] Alexei Czeskis, Karl Koscher, Joshua R Smith, and Tadayoshi Kohno. "RFIDs and Secret Handshakes: Defending against Ghost-and-leech Attacks and Unauthorized Reads with Context-aware Communications." In: *Proceedings of the 15th ACM Conference on Computer and Communications Security*. 2008, pp. 479–490.

[64] Sauvik Das, Eiji Hayashi, and Jason I Hong. "Exploring Capturable Everyday Memory for Autobiographical Authentication." In: *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2013, pp. 211–220.

[65] Sauvik Das, Adam DI Kramer, Laura A Dabbish, and Jason I Hong. "Increasing Security Sensitivity with Social Proof: A Large-scale Experimental Confirmation." In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 2014, pp. 739–749.

[66] Sauvik Das, Gierad Laput, Chris Harrison, and Jason I Hong. "Thumprint: Socially-inclusive Local Group Authentication through Shared Secret Knocks." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017, pp. 3764–3774.

[67] Debashis Dash, Hassan El Madi, Guru Gopalakrishnan, et al. "WiGig and IEEE 802.11 ad-For Multi-gigabyte-per-second WPAN and WLAN." In: *arXiv preprint arXiv:1211.7356* (2012).

[68] Abe Davis, Michael Rubinstein, Neal Wadhwa, Gautham J Mysore, Fredo Durand, and William T Freeman. "The Visual Microphone: Passive Recovery of Sound from Video." In: *ACM Trans. Graph.* 33.4 (2014), 79:1–79:10.

[69] Alan Dix, Janet Finlay, Gregory D Abowd, and Russell Beale. *Human-computer Interaction*. Pearson Education, 2003.

[70] Danny Dolev and Andrew Yao. "On the Security of Public Key Protocols." In: *IEEE Transactions on Information Theory* 29.2 (1983), pp. 198–208.

[71] Saar Drimer, Steven J Murdoch, et al. "Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks." In: *USENIX Security Symposium*. Vol. 312. 2007.

[72] Shihong Duan, Tianqing Yu, and Jie He. "Widriver: Driver Activity Recognition System Based on WiFi CSI." In: *International Journal of Wireless Information Networks* 25.2 (2018), pp. 146–156.

[73] Francis Duck and Timothy Leighton. "Frequency Bands for Ultrasound, Suitable for the Consideration of Its Health Effects." In: *The Journal of the Acoustical Society of America* 144.4 (2018), pp. 2490–2500.

[74] John Dunning. "Taming the Blue Beast: A Survey of Bluetooth Based Threats." In: *IEEE Security & Privacy* 8.2 (2010), pp. 20–27.

[75] Pierre-Alain Dupont, Julia Hesse, David Pointcheval, Leonid Reyzin, and Sophia Yakoubov. "Fuzzy Password-authenticated Key Exchange." In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2018, pp. 393–424.

[76] Morris J Dworkin. *SP 800-38d. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*. National Institute of Standards & Technology, 2007.

[77] Andrzej Dziech, Jaroslaw Bialas, Andrzej Glowacz, Pawel Korus, Mikolaj Leszczuk, Andrzej Matiolalski, and Remigiusz Baran. "Overview of Recent Advances in CCTV Processing Chain in the INDECT and INSIGMA Projects." In: *2013 International Conference on Availability, Reliability and Security*. IEEE. 2013, pp. 836–843.

[78] EBV Elektronik. *RFID Selection Guide*. https://cdn-shop.adafruit.com/datasheets/rfid+guide.pdf. 2010.

[79] Mahmoud Elkhodr, Seyed Shahrestani, and Hon Cheung. "The Internet of Things: New Interoperability, Management and Security Challenges." In: *International Journal of Network Security and its Applications* 8.2 (2016), pp. 85–102.

[80] everything RF. *Worlds First Smartphone Integrates 60 GHz Transmitter from SiBEAM*. https://www.everythingrf.com/News/details/1575-worlds-first-smartphone-integrates-60-ghz-transmitter-from-sibeam. 2015.

[81] Adam Fabio. *Hacklet 118 – Infrared and Universal Remote Controls*. https://hackaday.com/2016/07/30/hacklet-118-infrared-and-universal-remote-controls/. 2016.

[82] Song Fang, Yao Liu, Wenbo Shen, and Haojin Zhu. "Where Are You From? Confusing Location Distinction Using Virtual Multipath Camouflage." In: *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*. 2014, pp. 225–236.

[83] Song Fang, Yao Liu, Wenbo Shen, Haojin Zhu, and Tao Wang. "Virtual Multipath Attack and Defense for Location Distinction in Wireless Networks." In: *IEEE Transactions on Mobile Computing* 16.2 (2016), pp. 566–580.

[84] Michael Farb, Yue-Hsun Lin, Tiffany Hyun-Jin Kim, Jonathan McCune, and Adrian Perrig. "Safeslinger: Easy-to-use and Secure Public-key Exchange." In: *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*. 2013, pp. 417–428.

[85] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. "Do We Need Hundreds of Classifiers to Solve Real World Classification Problems?" In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 3133–3181.

[86] Wi-Fi Alliance. *Wi-Fi Protected Setup Specification, Version 1.0h*. 2006.

[87] Wi-Fi Alliance. *Wi-Fi Simple Configuration Technical Specification Version 2.0.5*. https://www.wi-fi.org/download.php?file=/sites/default/files/private/Wi-Fi_Simple_Configuration_Technical_Specification_v2.0.5.pdf. 2014.

[88] Wi-Fi Alliance. *Wi-Fi Certified WiGig: Wi-Fi Expands to 60 GHz (2016)*. https://www.wi-fi.org/downloads-registered-guest/wp_Wi-Fi_CERTIFIED_WiGig_20161024.pdf/29706. 2016.

[89] Wi-Fi Alliance. *Wi-Fi Direct*. http://www.wi-fi.org/discover-wi-fi/wi-fi-direct. 2016.

[90] Rainhard Dieter Findling, Muhammad Muaaz, Daniel Hintze, and René Mayrhofer. "Shakeunlock: Securely Transfer Authentication States between Mobile Devices." In: *IEEE Transactions on Mobile Computing* 16.4 (2016), pp. 1163–1175.

[91] Mikhail Fomichev, Luis F Abanto-Leon, Max Stiegler, Alejandro Molina, Jakob Link, and Matthias Hollick. *Index of Supplementary Files from "Next2You: Robust Copresence Detection Based on Channel State Information"*. https://doi.org/10.5281/zenodo.5105815. 2021.

[92] Mikhail Fomichev, Luis F. Abanto-Leon, Max Stiegler, Alejandro Molina, Jakob Link, and Matthias Hollick. "Next2You: Robust Copresence Detection Based on Channel State Information." In: *ACM Transactions on Internet of Things* 1.1 (2021), pp. 1–30.

[93] Mikhail Fomichev, Flor Álvarez, Daniel Steinmetzer, Paul Gardner-Stephen, and Matthias Hollick. "Survey and Systematization of Secure Device Pairing." In: *IEEE Communications Surveys & Tutorials* 20.1 (2018), pp. 517–550.

[94] Mikhail Fomichev, Julia Hesse, Lars Almon, Timm Lippert, Jun Han, and Matthias Hollick. "FastZIP: Faster and More Secure Zero-interaction Pairing." In: *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*. 2021, pp. 440–452.

[95] Mikhail Fomichev, Julia Hesse, Lars Almon, Timm Lippert, Jun Han, and Matthias Hollick. *Index of Supplementary Files from "FastZIP: Faster and More Secure Zero-Interaction Pairing"*. https://doi.org/10.5281/zenodo.4777836. 2021.

[96] Mikhail Fomichev, Max Maass, Lars Almon, Alejandro Molina, and Matthias Hollick. *Audio Data from Mobile Scenario from "Perils of Zero-Interaction Security in the Internet of Things"*. https://doi.org/10.5281/zenodo.2537984. 2019.

[97] Mikhail Fomichev, Max Maass, Lars Almon, Alejandro Molina, and Matthias Hollick. *Index of Supplementary Files from "Perils of Zero-Interaction Security in the Internet of Things"*. https://doi.org/10.5281/zenodo.2537721. 2019.

[98] Mikhail Fomichev, Max Maass, Lars Almon, Alejandro Molina, and Matthias Hollick. "Perils of Zero-interaction Security in the Internet of Things." In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3.1 (2019), pp. 1–38.

[99] Mikhail Fomichev, Max Maass, and Matthias Hollick. "Zero-interaction Security-Towards Sound Experimental Validation." In: *GetMobile: Mobile Computing and Communications* 23.2 (2019), pp. 16–21.

[100] Python Software Foundation. *Pickle — Python Object Serialization*. https://docs.python.org/3/library/pickle.html?highlight=pickle. 2020.

[101] Python Software Foundation. *Socket — Low-level Networking Interface*. https://docs.python.org/3/library/socket.html?highlight=socket. 2020.

[102] Python Software Foundation. *Time — Time Access and Conversions*. https://docs.python.org/3/library/time.html#module-time. 2020.

[103] Aurélien Francillon, Boris Danev, and Srdjan Capkun. "Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars." In: *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. 2011.

[104] Lishoy Francis, Gerhard Hancke, Keith Mayes, and Konstantinos Markantonakis. "Practical NFC Peer-to-peer Relay Attack Using Mobile Phones." In: *International Workshop on Radio Frequency Identification: Security and Privacy Issues*. Springer. 2010, pp. 35–49.

[105] Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. "Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication." In: *IEEE Transactions on Information Forensics and Security* 8.1 (2012), pp. 136–148.

[106] Jonathan Frankle and Michael Carbin. "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks." In: *arXiv preprint arXiv:1803.03635* (2018).

[107] Jerome H Friedman. "Greedy Function Approximation: A Gradient Boosting Machine." In: *Annals of Statistics* (2001), pp. 1189–1232.

[108] Fraida Fund. *Run a Man-in-the-middle Attack on a WiFi Hotspot*. https://witestlab.poly.edu/blog/conduct-a-simple-man-in-the-middle-attack-on-a-wifi-hotspot/. 2016.

[109] Futurae Technologies AG. *Futurae Authentication Suite*. https://www.futurae.com/product/strongauth/. 2019.

[110] Alexander Gallego, Nitesh Saxena, and Jonathan Voris. "Playful Security: A Computer Game for Secure Wireless Device Pairing." In: *2011 16th International Conference on Computer Games (CGAMES)*. IEEE. 2011, pp. 177–184.

[111] Qinhua Gao, Jie Wang, Xiaorui Ma, Xueyan Feng, and Hongyu Wang. "CSI-based Device-free Wireless Localization and Activity Recognition Using Radio Image Features." In: *IEEE Transactions on Vehicular Technology* 66.11 (2017), pp. 10346–10356.

[112] Paul Gardner-Stephen. "The Serval Project: Practical Wireless Ad-hoc Mobile Telecommunications." In: *Flinders University, Adelaide, South Australia, Tech. Rep* (2011).

[113] Matthias Gauger, Olga Saukh, and Pedro José Marrón. "Enlighten Me! Secure Key Assignment in Wireless Sensor Networks." In: *2009 IEEE 6th International Conference on Mobile Adhoc and Sensor Systems*. IEEE. 2009, pp. 246–255.

[114] Piotr Gawlowicz, Anatolij Zubow, Suzan Bayhan, and Adam Wolisz. "Punched Cards over the Air: Cross-Technology Communication Between LTE-U/LAA and WiFi." In: *2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE. 2020, pp. 297–306.

[115] Christian Gehrmann, Chris J Mitchell, and Kaisa Nyberg. "Manual Authentication for Wireless Devices." In: *RSA Cryptobytes* 7.1 (2004), pp. 29–37.

[116] Nirnimesh Ghose, Loukas Lazos, and Ming Li. "SFIRE: Secret-free In-band Trust Establishment for COTS Wireless Devices." In: *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE. 2018, pp. 1529–1537.

[117] Ilias Giechaskiel and Kasper Rasmussen. "Taxonomy and Challenges of Out-of-band Signal Injection Attacks and Defenses." In: *IEEE Communications Surveys & Tutorials* 22.1 (2019), pp. 645–670.

[118] Alexander S. Gillis. *Internet of Things (IoT)*. https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT. 2020.

[119] Shyamnath Gollakota, Nabeel Ahmed, Nickolai Zeldovich, and Dina Katabi. "Secure In-Band Wireless Pairing." In: *USENIX Security Symposium*. 2011, pp. 1–16.

[120] Liangyi Gong, Wu Yang, Zimu Zhou, Dapeng Man, Haibin Cai, Xiancun Zhou, and Zheng Yang. "An Adaptive Wireless Passive Human Detection via Fine-grained Physical Layer Information." In: *Ad Hoc Networks* 38 (2016), pp. 38–50.

[121] Michelle X Gong, Brian Hart, and Shiwen Mao. "Advanced Wireless LAN Technologies: IEEE 802.11 ac and Beyond." In: *GetMobile: Mobile Computing and Communications* 18.4 (2015), pp. 48–52.

[122] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[123] Michael T Goodrich, Michael Sirivianos, John Solis, Gene Tsudik, and Ersin Uzun. "Loud and Clear: Human-verifiable Authentication Based on Audio." In: *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*. IEEE. 2006, pp. 10–10.

[124] Google, Inc. *Nexus Tech Specs*. https://support.google.com/nexus/answer/6102470?hl=en. 2016.

[125] Google, Inc. *Sensors Overview*. https://developer.android.com/guide/topics/sensors/sensors_overview.html. 2016.

[126] Google, Inc. *Wi-Fi Peer-to-peer*. https://developer.android.com/guide/topics/connectivity/wifip2p.html. 2017.

[127] Robert M. Gray. "Toeplitz and Circulant Matrices: A Review." In: *Found. Trends Commun. Inf. Theory* 2.3 (2005).

[128] Samuel Greengard. *The Internet of Things*. MIT press, 2015.

[129] Francesco Gringoli, Matthias Schulz, Jakob Link, and Matthias Hollick. "Free Your CSI: A Channel State Information Extraction Platform For Modern Wi-Fi Chipsets." In: *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*. 2019, pp. 21–28.

[130] The Weaver Computer Engineering Research Group. *The Top 50 Fastest Computers in the Weaver Research Group*. http://web.eece.maine.edu/~vweaver/group/machines.html. 2020.

[131] Bogdan Groza, Adriana Berdich, Camil Jichici, and Rene Mayrhofer. "Secure Accelerometer-based Pairing of Mobile Devices in Multi-modal Transport." In: *IEEE Access* 8 (2020), pp. 9246–9259.

[132] Bogdan Groza and Rene Mayrhofer. "SAPHE: Simple Accelerometer Based Wireless Pairing with Heuristic Trees." In: *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*. 2012, pp. 161–168.

[133] Jens Grubert, Matthias Kranz, and Aaron Quigley. "Challenges in Mobile Multi-device Ecosystems." In: *mUX: The Journal of Mobile User Experience* 5.1 (2016), pp. 1–22.

[134] Iakovos Gurulian, Carlton Shepherd, Konstantinos Markantonakis, Raja Naeem Akram, and Keith Mayes. "When Theory and Reality Collide: Demystifying the Effectiveness of Ambient Sensing for NFC-based Proximity Detection by Applying Relay Attack Data." In: *arXiv preprint arXiv:1605.00425* (2016).

[135] Keijo Haataja and Pekka Toivanen. "Two Practical Man-in-the-middle Attacks on Bluetooth Secure Simple Pairing and Countermeasures." In: *IEEE Transactions on Wireless Communications* 9.1 (2010), pp. 384–392.

[136] Tzipora Halevi, Di Ma, Nitesh Saxena, and Tuo Xiang. "Secure Proximity Detection for NFC Devices Based on Ambient Sensor Data." In: *European Symposium on Research in Computer Security*. Springer. 2012, pp. 379–396.

[137] Tzipora Halevi and Nitesh Saxena. "Acoustic Eavesdropping Attacks on Constrained Wireless Device Pairing." In: *IEEE Transactions on Information Forensics and Security* 8.3 (2013), pp. 563–577.

[138] Tzipora Halevi and Nitesh Saxena. "Keyboard Acoustic Side Channel Attacks: Exploring Realistic and Security-sensitive Scenarios." In: *International Journal of Information Security* 14.5 (2015), pp. 443–456.

[139] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. "The WEKA Data Mining Software: An Update." In: *ACM SIGKDD Explorations Newsletter* 11.1 (2009), pp. 10–18.

[140] Josef Hallberg and Marcus Nilsson. *Positioning with Bluetooth, IrDA and RFID*. 2002.

[141] Daniel Halperin, Thomas S Heydt-Benjamin, Benjamin Ransford, Shane S Clark, Benessa Defend, Will Morgan, Kevin Fu, Tadayoshi Kohno, and William H Maisel. "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses." In: *IEEE Symposium on Security and Privacy (S&P)*. IEEE. 2008, pp. 129–142.

[142] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. "Tool Release: Gathering 802.11n Traces with Channel State Information." In: *ACM SIGCOMM CCR* 41.1 (2011), p. 53.

[143] Dianqi Han, Yimin Chen, Tao Li, Rui Zhang, Yaochao Zhang, and Terri Hedgpeth. "Proximity-Proof: Secure and Usable Mobile Two-factor Authentication." In: *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. 2018, pp. 401–415.

[144] Jun Han, Albert Jin Chung, Manal Kumar Sinha, Madhumitha Harishankar, Shijia Pan, Hae Young Noh, Pei Zhang, and Patrick Tague. "Do You Feel What I Hear? Enabling Autonomous IoT Device Pairing Using Different Sensor Types." In: *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2018, pp. 836–852.

[145] Jun Han, Madhumitha Harishankar, Xiao Wang, Albert Jin Chung, and Patrick Tague. "Convoy: Physical Context Verification for Vehicle Platoon Admission." In: *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications*. 2017, pp. 73–78.

[146] Jun Han, Shijia Pan, Manal Kumar Sinha, Hae Young Noh, Pei Zhang, and Patrick Tague. "Smart Home Occupant Identification via Sensor Fusion across On-object Devices." In: *ACM Transactions on Sensor Networks (TOSN)* 14.3-4 (2018), pp. 1–22.

[147] Open Data Handbook. "What Is Open Data." In: *Retrieved 17th February* (2015).

[148] Michael Hanspach and Michael Goetz. "On Covert Acoustical Mesh Networks in Air." In: *arXiv preprint arXiv:1406.1213* (2014).

[149] Tian Hao, Ruogu Zhou, and Guoliang Xing. "COBRA: Color Barcode Streaming for Smartphone Systems." In: *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*. 2012, pp. 85–98.

[150] Steven C Hauser, William C Headley, and Alan J Michaels. "Signal Detection Effects on Deep Neural Networks Utilizing Raw IQ for Modulation Classification." In: *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*. IEEE. 2017, pp. 121–127.

[151] Hertz Blog. *Connected Cars: Improve Your Driving Experience with the Internet of Things (IoT)*. https://www.hertz.com/blog/automotive/improve-your-driving-experience-with-internet-of-things. 2020.

[152] Grant Ho, Derek Leung, Pratyush Mishra, Ashkan Hosseini, Dawn Song, and David Wagner. "Smart Locks: Lessons for Securing Commodity Internet of Things Devices." In: *Proceedings of the 11th ACM on Asia conference on computer and communications security*. 2016, pp. 461–472.

[153] Joel Höglund, Samuel Lindemer, Martin Furuhed, and Shahid Raza. "PKI4IoT: Towards Public Key Infrastructure for the Internet of Things." In: *Computers & Security* 89 (2020), p. 101658.

[154] Kurt Hornik, Maxwell Stinchcombe, Halbert White, et al. "Multilayer Feedforward Networks Are Universal Approximators." In: *Neural Networks* 2.5 (1989), pp. 359–366.

[155] HP Development Company, L.P. *HP Elite x2 1011 G2 - Connecting to the Wireless Dock*. https://support.hp.com/hr-en/document/c04587366. 2015.

[156] Hsu-Chun Hsiao, Yue-Hsun Lin, Ahren Studer, Cassandra Studer, King-Hang Wang, Hiroaki Kikuchi, Adrian Perrig, Hung-Min Sun, and Bo-Yin Yang. "A Study of User-friendly Hash Comparison Schemes." In: *2009 Annual Computer Security Applications Conference*. IEEE. 2009, pp. 105–114.

[157] Kao-Cheng Huang and Zhaocheng Wang. *Millimeter Wave Communication Systems*. Vol. 29. John Wiley & Sons, 2011.

[158] Otto Huhta, Swapnil Udar, Mika Juuti, Prakash Shrestha, Nitesh Saxena, and N Asokan. "Pitfalls in Designing Zero-Effort Deauthentication: Opportunistic Human Observation Attacks." In: *23rd Annual Network and Distributed System Security Symposium (NDSS 2016)*. 2016.

[159] Ahmed Taha Hussein, Mohammed T Alresheedi, and Jaafar MH Elmirghani. "20 Gb/s Mobile Indoor Visible Light Communication System Employing Beam Steering and Computer Generated Holograms." In: *Journal of Lightwave Technology* 33.24 (2015), pp. 5242–5260.

[160] HyperPhysics. *Sound Propogation*. http://hyperphysics.phy-astr.gsu.edu/hbase/Sound/sprop.html. 2016.

[161] HyperPhysics. *Ultrasonic Sound*. http://hyperphysics.phy-astr.gsu.edu/hbase/Sound/usound.html. 2016.

[162] IEEE Standards Association. *802.15.7-2011 Part 15.7: Short-range Wireless Optical Communication Using Visible Light*. https://standards.ieee.org/findstds/standard/802.15.7-2011.html. 2011.

[163] IEEE Standards Association. *802.11ad-2012 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. https://standards.ieee.org/findstds/standard/802.11ad-2012.html. 2012.

[164] Infrared Data Association. *What Is Infrared?* http://www.irda.org/. 2011.

[165] American National Standards Institute. *Specification for Octave-band and Fractional-octave-band Analog and Digital Filters*. 1986.

[166] Iulia Ion, Marc Langheinrich, Ponnurangam Kumaraguru, and Srdjan Čapkun. "Influence of User Perception, Security Needs, and Social Factors on Device Pairing Method Choices." In: *Proceedings of the Sixth Symposium on Usable Privacy and Security*. 2010, pp. 1–13.

[167] Ironpaper. *Smart Car Statistics – The Increasingly Digital Experience of the Connected Vehicle*. https://www.ironpaper.com/webintel/articles/smart-car-statistics-the-increasingly-digital-experience-of-the-connected-vehicle/. 2018.

[168] Alejandro Jaimes and Nicu Sebe. "Multimodal Human–computer Interaction: A Survey." In: *Computer Vision and Image Understanding* 108.1-2 (2007), pp. 116–134.

[169] Stanisław Jarecki and Xiaomin Liu. "Fast Secure Computation of Set Iintersection." In: *International Conference on Security and Cryptography for Networks*. Springer. 2010, pp. 418–435.

[170] Sławomir Jasek. "Gattacking Bluetooth Smart Devices." In: *Black Hat USA Conference*. 2016.

[171] Yunhan Jack Jia, Qi Alfred Chen, Shiqi Wang, Amir Rahmati, Earlence Fernandes, Zhuoqing Morley Mao, Atul Prakash, and SJ Unviersity. "ContexIoT: Towards Providing Contextual Integrity to Appified IoT Platforms." In: *NDSS*. 2017.

[172] Rong Jin, Liu Shi, Kai Zeng, Amit Pande, and Prasant Mohapatra. "MagPairing: Exploiting Magnetometers for Pairing Smartphones in Close Proximity." In: *2014 IEEE Conference on Communications and Network Security*. IEEE. 2014, pp. 445–453.

[173] Wenqiang Jin, Ming Li, Srinivasan Murali, and Linke Guo. "Harnessing the Ambient Radio Frequency Noise for Wearable Device Pairing." In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020, pp. 1135–1148.

[174] Ari Juels and Madhu Sudan. "A Fuzzy Vault Scheme." In: *Designs, Codes and Cryptography* 38.2 (2006), pp. 237–257.

[175] Ari Juels and Martin Wattenberg. "A Fuzzy Commitment Scheme." In: *Proceedings of the 6th ACM Conference on Computer and Communications Security*. 1999, pp. 28–36.

[176] Mika Juuti, Christian Vaas, Ivo Sluganovic, Hans Liljestrand, N Asokan, and Ivan Martinovic. "STASH: Securing Transparent Authentication Schemes Using Prover-side Proximity Verification." In: *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE. 2017, pp. 1–9.

[177] Ronald Kainda, Ivan Flechais, and Andrew William Roscoe. "Usability and Security of Out-of-band Channels in Secure Device Pairing Protocols." In: *Proceedings of the 5th Symposium on Usable Privacy and Security*. 2009, pp. 1–12.

[178] Ronald Kainda, Ivan Flechais, and AW Roscoe. "Two Heads Are Better Than One: Security and Usability of Device Associations in Group Scenarios." In: *Proceedings of the Sixth Symposium on Usable Privacy and Security*. 2010, pp. 1–13.

[179] Andre Kalamandeen, Adin Scannell, Eyal de Lara, Anmol Sheth, and Anthony LaMarca. "Ensemble: Cooperative Proximity-based Authentication." In: *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*. 2010, pp. 331–344.

[180] Elliott D Kaplan and Christopher Hegarty. *Understanding GPS/GNSS: Principles and Applications*. Artech House, 2017.

[181] Nikolaos Karapanos, Claudio Marforio, Claudio Soriente, and Srdjan Capkun. "Sound-Proof: Usable Two-factor Authentication Based on Ambient Sound." In: *24th {USENIX} Security Symposium ({USENIX} Security 15)*. 2015, pp. 483–498.

[182] Chucri A Kardous and Peter B Shaw. "Evaluation of Smartphone Sound Measurement Applications (Apps) Using External Microphones—A Follow-up Study." In: *The Journal of the Acoustical Society of America* 140.4 (2016), EL327–EL333.

[183] Timo Kasper, David Oswald, and Christof Paar. "Wireless Security Threats: Eavesdropping and Detecting of Active RFIDs and Remote Controls in the Wild." In: *SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks*. IEEE. 2011, pp. 1–6.

[184] Wooseong Kim, Yejin Shin, and Soonuk Seol. "Smart Phone Assisted Personal IoT Service." In: *Advanced Science and Technology Letters* 110.13 (2015), pp. 61–66.

[185] Tim Kindberg and Kan Zhang. "Validating and Securing Spontaneous Associations between Wireless Devices." In: *International Conference on Information Security*. Springer. 2003, pp. 44–53.

[186] Paris Kitsos. *Security in RFID and Sensor Networks*. CRC Press, 2016.

[187] Alfred Kobsa, Rahim Sonawalla, Gene Tsudik, Ersin Uzun, and Yang Wang. "Serial Hook-ups: A Comparative Usability Study of Secure Device Pairing Methods." In: *Proceedings of the 5th Symposium on Usable Privacy and Security*. 2009, pp. 1–12.

[188] Will Koehrsen. *Feature Engineering: What Powers Machine Learning*. https://towardsdatascience.com/feature-engineering-what-powers-machine-learning-93ab191bcc2d. 2018.

[189] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. "DDoS in the IoT: Mirai and Other Botnets." In: *Computer* 50.7 (2017), pp. 80–84.

[190] Tonko Kovačević, Toni Perković, and Mario Čagalj. "Flashing Displays: User-friendly Solution for Bootstrapping Secure Associations between Multiple Constrained Wireless Devices." In: *Security and Communication Networks* 9.10 (2016), pp. 1050–1071.

[191] Sara Kraemer and Pascale Carayon. "Human Errors and Violations in Computer and Information Security: The Viewpoint of Network Administrators and Security Specialists." In: *Applied Ergonomics* 38.2 (2007), pp. 143–154.

[192] Klaudia Krawiecka, Arseny Kurnikov, Andrew Paverd, Mohammad Mannan, and N Asokan. "Safekeeper: Protecting Web Passwords Using Trusted Execution Environments." In: *Proceedings of the 2018 World Wide Web Conference*. 2018, pp. 349–358.

[193] Christian Kray, Daniel Nesbitt, John Dawson, and Michael Rohs. "User-defined Gestures for Connecting Mobile Phones, Public displays, and Tabletops." In: *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*. 2010, pp. 239–248.

[194] Dan Kreiser, Zoya Dyka, Stephan Kornemann, Christian Wittke, Ievgen Kabin, Oliver Stecklina, and Peter Langendörfer. "On Wireless Channel Parameters for Key Generation in Industrial Environments." In: *IEEE Access* 6 (2018), pp. 79010–79025.

[195] Jacob Kröger. "Unexpected Inferences from Sensor Data: A Hidden Privacy Threat in the Internet of Things." In: *IFIP International Internet of Things Conference*. Springer. 2018, pp. 147–159.

[196] Arun Kumar, Nitesh Saxena, Gene Tsudik, and Ersin Uzun. "A Comparative Study of Secure Device Pairing Methods." In: *Pervasive and Mobile Computing* 5.6 (2009), pp. 734–749.

[197] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. "Unmasking Clever Hans Predictors and Assessing What Machines Really Learn." In: *Nature Communications* 10.1 (2019), pp. 1–8.

[198] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. "Backpropagation Applied to Handwritten ZIP Code Recognition." In: *Neural Computation* 1.4 (1989), pp. 541–551.

[199] Eunchong Lee, Hyunsoo Kim, and Ji Won Yoon. "Various Threat Models to Circumvent Air-gapped Systems for Preventing Network Attack." In: *International Workshop on Information Security Applications*. Springer. 2015, pp. 187–199.

[200] Kyuin Lee, Neil Klingensmith, Suman Banerjee, and Younghyun Kim. "VoltKey: Continuous Secret Key Generation Based on Power Line Noise for Zero-Involvement Pairing and Authentication." In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3.3 (2019), p. 93.

[201] Matt Leonard. *Declining Price of IoT Sensors Means Greater Use in Manufacturing*. https://www.supplychaindive.com/news/declining-price-iot-sensors-manufacturing/564980/. 2019.

[202] Lingjun Li, Xinxin Zhao, and Guoliang Xue. "A Proximity Authentication System for Smartphones." In: *IEEE Transactions on Dependable and Secure Computing* 13.6 (2015), pp. 605–616.

[203] Ming Li, Shucheng Yu, Wenjing Lou, and Kui Ren. "Group Device Pairing Based Secure Sensor Association and Key Management for Body Area Networks." In: *2010 Proceedings IEEE INFOCOM*. IEEE. 2010, pp. 1–9.

[204] Sugang Li, Ashwin Ashok, Yanyong Zhang, Chenren Xu, Janne Lindqvist, and Macro Gruteser. "Whose Move Is It Anyway? Authenticating Smart Wearable Devices Using Unique Head Movement Patterns." In: *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE. 2016, pp. 1–9.

[205] Xiaopeng Li, Fengyao Yan, Fei Zuo, Qiang Zeng, and Lannan Luo. "Touch Well Before Use: Intuitive and Secure Authentication for IoT Devices." In: *The 25th Annual International Conference on Mobile Computing and Networking*. 2019, pp. 1–17.

[206] Xiaopeng Li, Qiang Zeng, Lannan Luo, and Tongbo Luo. "T2Pair: Secure and Usable Pairing for Heterogeneous IoT Devices." In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020, pp. 309–323.

[207] Xiaohui Liang, Tianlong Yun, Ronald Peterson, and David Kotz. "LightTouch: Securely Connecting Wearables to Ambient Displays With User Intent." In: *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE. 2017, pp. 1–9.

[208] Divakaran Liginlal, Inkook Sim, and Lara Khansa. "How Significant Is Human Error as a Cause of Privacy Breaches? An Empirical Study and a Framework for Error Management." In: *Computers & Security* 28.3-4 (2009), pp. 215–228.

[209] Felix Xiaozhu Lin, Daniel Ashbrook, and Sean White. "RhythmLink: Securely Pairing I/O-constrained Devices by Tapping." In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. 2011, pp. 263–272.

[210] Qi Lin, Weitao Xu, Jun Liu, Abdelwahed Khamis, Wen Hu, Mahbub Hassan, and Aruna Seneviratne. "H2B: Heartbeat-based Secret Key Generation Using Piezo Vibration Sensors." In: *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*. 2019, pp. 265–276.

[211] Andrew Y Lindell. "Attacks on the Pairing Protocol of Bluetooth v2.1." In: *Black Hat USA, Las Vegas, Nevada* (2008).

[212] Jialin Liu, Lei Wang, Linlin Guo, Jian Fang, Bingxian Lu, and Wei Zhou. "A Research on CSI-based Human Motion Detection in Complex Scenarios." In: *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*. IEEE. 2017, pp. 1–6.

[213] Yanpei Liu, Stark C Draper, and Akbar M Sayeed. "Exploiting Channel Diversity in Secret Key Generation from Multipath Fading Randomness." In: *IEEE Transactions on Information Forensics and Security* 7.5 (2012), pp. 1484–1497.

[216] Hong Lu, Wei Pan, Nicholas D Lane, Tanzeem Choudhury, and Andrew T Campbell. "SoundSense: Scalable Sound Sensing for People-centric Applications on Mobile Phones." In: *Proceedings of the 7th International Conference on Mobile systems, Applications, and Services*. 2009, pp. 165–178.

[217] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. "The Expressive Power of Neural Networks: A View from the Width." In: *Advances in Neural Information Processing Systems*. 2017, pp. 6231–6239.

[218] Zhuo Lu, Xiang Lu, Wenye Wang, and Cliff Wang. "Review and Evaluation of Security Threats on the Communication Networks in the Smart Grid." In: *2010-Milcom 2010 Military Communications Conference*. IEEE. 2010, pp. 1830–1835.

[219] Di Ma and Nitesh Saxena. "A Context-aware Approach to Defend against Unauthorized Reading and Relay Attacks in RFID Systems." In: *Security and Communication Networks* 7.12 (2014), pp. 2684–2695.

[220] Di Ma, Nitesh Saxena, Tuo Xiang, and Yan Zhu. "Location-aware and Safer Cards: Enhancing RFID Security and Privacy via Location Sensing." In: *IEEE Transactions on Dependable and Secure Computing* 10.2 (2013), pp. 57–69.

[221] Yongsen Ma, Sheheryar Arshad, Swetha Muniraju, Eric Torkildson, Enrico Rantala, Klaus Doppler, and Gang Zhou. "Location- and Person-independent Activity Recognition with WiFi, Deep Neural Networks, and Reinforcement Learning." In: *ACM Transactions on Internet of Things* 2.1 (2021), pp. 1–25.

[222] Yongsen Ma, Gang Zhou, and Shuangquan Wang. "WiFi Sensing with Channel State Information: A Survey." In: *ACM Computing Surveys (CSUR)* 52.3 (2019), pp. 1–36.

[223] Max Maass, Uwe Müller, Tom Schons, Daniel Wegemer, and Matthias Schulz. "NFCGate: An NFC Relay Application for Android." In: *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. 2015, pp. 1–2.

[224] Gilad David Maayan. *The IoT Rundown For 2020: Stats, Risks, and Solutions*. https://securitytoday.com/Articles/2020/01/13/The-IoT-Rundown-for-2020.aspx?Page=2. 2020.

[225] Anil Madhavapeddy, Richard Sharp, David Scott, and Alastair Tse. "Audio Networking: The Forgotten Wireless Technology." In: *IEEE Pervasive Computing* 4.3 (2005), pp. 55–60.

[226] Michael Mahler. "A Home Security System Based on Smartphone Sensors." In: (2018).

[227] Nicolas Maisonneuve, Matthias Stevens, Maria E Niessen, and Luc Steels. "NoiseTube: Measuring and Mapping Noise Pollution with Mobile Phones." In: *Information Technologies in Environmental Engineering*. Springer, 2009, pp. 215–228.

[228] Dennis Mantz, Jiska Classen, Matthias Schulz, and Matthias Hollick. "Internalblue-Bluetooth Binary Patching and Experimentation Framework." In: *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. 2019, pp. 79–90.

[229] Shrirang Mare, Andrés Molina Markham, Cory Cornelius, Ronald Peterson, and David Kotz. "Zebra: Zero-effort Bilateral Recurring Authentication." In: *2014 IEEE Symposium on Security and Privacy*. IEEE. 2014, pp. 705–720.

[230] Claudio Marforio, Nikolaos Karapanos, Claudio Soriente, Kari Kostiainen, and Srdjan Capkun. "Smartphones as Practical and Secure Location Verification Tokens for Payments." In: *NDSS*. Vol. 14. 2014, pp. 23–26.

[231] Konstantinos Markantonakis, Lishoy Francis, Gerhard Hancke, and Keith Mayes. "Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones." In: *Radio Frequency Identification System Security: RFIDsec* 12 (2012), p. 21.

[232] Suhas Mathur, Robert Miller, Alexander Varshavsky, Wade Trappe, and Narayan Mandayam. "Proximate: Proximity-based Secure Pairing Using Ambient Wireless Signals." In: *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*. 2011, pp. 211–224.

[233] Jan Matuszewski and Katarzyna Sikorska-Łukasiewicz. "Neural Network Application for Emitter Identification." In: *2017 18th International Radar Symposium (IRS)*. IEEE. 2017, pp. 1–8.

[234] Rene Mayrhofer, Jürgen Fuß, and Iulia Ion. "UACAP: A Unified Auxiliary Channel Authentication Protocol." In: *IEEE Transactions on Mobile Computing* 12.4 (2013), pp. 710–721.

[235] Rene Mayrhofer and Hans Gellersen. "On the Security of Ultrasound as Out-of-band Channel." In: *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE. 2007, pp. 1–6.

[236] Rene Mayrhofer and Hans Gellersen. "Shake Well before Use: Intuitive and Secure Pairing of Mobile Devices." In: *IEEE Transactions on Mobile Computing* 8.6 (2009), pp. 792–806.

[237] Rene Mayrhofer, Mike Hazas, and Hans Gellersen. "An Authentication Protocol Using Ultrasonic Ranging." In: (2006).

[238] Rene Mayrhofer and Martyn Welch. "A Human-verifiable Authentication Protocol Using Visible Laser Light." In: *The Second International Conference on Availability, Reliability and Security (ARES'07)*. IEEE. 2007, pp. 1143–1148.

[239] René Mayrhofer and Stephan Sigg. "Adversary Models for Mobile Device Authentication." In: *ACM Computing Surveys (CSUR)* 54.9 (2021), pp. 1–35.

[240] Jonathan M McCune, Adrian Perrig, and Michael K Reiter. "Seeing-is-believing: Using Camera Phones for Human-verifiable Authentication." In: *2005 IEEE Symposium on Security and Privacy (S&P'05)*. IEEE. 2005, pp. 110–124.

[241] Abu Zaher Md Faridee, Sreenivasan Ramasamy Ramamurthy, and Nirmalya Roy. "HappyFeet: Challenges in Building an Automated Dance Recognition and Assessment Tool." In: *GetMobile: Mobile Computing and Communications* 22.3 (2019), pp. 10–16.

[242] Maryam Mehrnezhad, Feng Hao, and Siamak F Shahandashti. "Tap-Tap and Pay (TTP): Preventing the Mafia Attack in NFC Payment." In: *International Conference on Research in Security Standardisation*. Springer. 2015, pp. 21–39.

[243] Markus Miettinen, N Asokan, Thien Duc Nguyen, Ahmad-Reza Sadeghi, and Majid Sobhani. "Context-based Zero-interaction Pairing and Key Evolution for Advanced Personal Devices." In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 2014, pp. 880–891.

[244] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. "Iot Sentinel: Automated Device-type Identification for Security Enforcement in IoT." In: *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2017, pp. 2177–2184.

[245] Markus Miettinen, Thien Duc Nguyen, Ahmad-Reza Sadeghi, and N Asokan. "Revisiting Context-based Authentication in IoT." In: *Proceedings of the 55th Annual Design Automation Conference*. 2018, pp. 1–6.

[246] Emiliano Miluzzo, Michela Papandrea, Nicholas D Lane, Hong Lu, and Andrew T Campbell. "Pocket, Bag, Hand, etc.-Automatically Detecting Phone Context through Discovery." In: *Proc. PhoneSense 2010* (2010), pp. 21–25.

[247] Shahab Mirzadeh, Haitham Cruickshank, and Rahim Tafazolli. "Secure Device Pairing: A Survey." In: *IEEE Communications Surveys & Tutorials* 16.1 (2014), pp. 17–40.

[248] Motorola, Inc. *5GHz IEEE 802.11a for Interference Avoidance*. https://www.motorolasolutions.com/content/dam/msi/docs/business/_documents/static_files/interference_tb_0809.pdf. 2009.

[249] Dibya Mukhopadhyay, Maliheh Shirvanian, and Nitesh Saxena. "All Your Voices Are Belong to Us: Stealing Voices to Fool Humans and Machines." In: *European Symposium on Research in Computer Security*. Springer. 2015, pp. 599–621.

[250] ABM Musa and Jakob Eriksson. "Tracking Unmodified Smartphones Using Wi-Fi Monitors." In: *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. 2012, pp. 281–294.

[251] Moni Naor, Lior Rotem, and Gil Segev. "Out-of-band Authenticated Group Key Exchange: From Strong Authentication to Immediate Key Delivery." In: *1st Conference on Information-Theoretic Cryptography (ITC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2020.

[2] National Security Agency. *Limiting Location Data Exposure*. https://media.defense.gov/2020/Aug/04/2002469874/-1/-1/0/CSI_LIMITING_LOCATION_DATA_EXPOSURE_FINAL.PDF. 2020.

[252] NFC Forum and Bluetooth, SIG. *Bluetooth Secure Simple Pairing Using NFC*. http://members.nfc-forum.org/apps/group_public/download.php/18688/NFCForum-AD-BTSSP_1_1.pdf. 2014.

[253] Thien Duc Nguyen, Samuel Marchal, Markus Miettinen, Hossein Fereidooni, N Asokan, and Ahmad-Reza Sadeghi. "DÏoT: A Federated Self-learning Anomaly Detection System for IoT." In: *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2019, pp. 756–767.

[254] Trung Nguyen and Jean Leneutre. "Formal Analysis of Secure Device Pairing Protocols." In: *2014 IEEE 13th International Symposium on Network Computing and Applications*. IEEE. 2014, pp. 291–295.

[255] Jianwei Niu, Fei Gu, Ruogu Zhou, Guoliang Xing, and Wei Xiang. "VINCE: Exploiting Visible Light Sensing for Smartphone-based NFC Systems." In: *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE. 2015, pp. 2722–2730.

[256] Yong Niu, Yong Li, Depeng Jin, Li Su, and Athanasios V Vasilakos. "A Survey of Millimeter Wave Communications (mmWave) for 5G: Opportunities and Challenges." In: *Wireless Networks* 21.8 (2015), pp. 2657–2676.

[257] Dominic O'Brien, Hoa Le Minh, Lubin Zeng, Grahame Faulkner, Kyungwoo Lee, Daekwang Jung, YunJe Oh, and Eun Tae Won. "Indoor Visible Light Communications: Challenges and Prospects." In: *Free-Space Laser Communications VIII*. Vol. 7091. International Society for Optics and Photonics. 2008, p. 709106.

[258] Samuel O'Malley and Kim-Kwang Raymond Choo. "Bridging the Air Gap: Inaudible Data Exfiltration by Insiders." In: *20th Americas Conference on Information Systems (AMCIS 2014)*. 2014, pp. 7–10.

[259] Timothy James O'Shea, Tamoghna Roy, and T Charles Clancy. "Over-the-air Deep Learning Based Radio Signal Classification." In: *IEEE Journal of Selected Topics in Signal Processing* 12.1 (2018), pp. 168–179.

[260] Katarzyna Olejnik, Italo Dacosta, Joana Soares Machado, Kévin Huguenin, Mohammad Emtiyaz Khan, and Jean-Pierre Hubaux. "Smarper: Context-aware and Automatic Runtime-permissions for Mobile Devices." In: *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2017, pp. 1058–1076.

[261] John Padgette, Karen Scarfone, and Lily Chen. "Guide to Bluetooth Security." In: *NIST Special Publication* 800.121 (2012), p. 25.

[262] Shijia Pan, Carlos Ruiz, Jun Han, Adeola Bannis, Patrick Tague, Hae Young Noh, and Pei Zhang. "Universense: IoT Device Pairing through Heterogeneous Sensing Signals." In: *Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications*. 2018, pp. 55–60.

[263] Parth H Pathak, Xiaotao Feng, Pengfei Hu, and Prasant Mohapatra. "Visible Light Communication, Networking, and Sensing: A Survey, Potential and Challenges." In: *IEEE Communications Surveys & Tutorials* 17.4 (2015), pp. 2047–2077.

[264] Al-Sakib Khan Pathan. *Security of Self-organizing Networks: MANET, WSN, WMN, VANET*. CRC press, 2016.

[265] Neal Patwari and Sneha K Kasera. "Robust Location Distinction Using Temporal Link Signatures." In: *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*. 2007, pp. 111–122.

[266] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. "Context Aware Computing for the Internet of Things: A Survey." In: *IEEE communications Surveys & Tutorials* 16.1 (2014), pp. 414–454.

[267] Toni Perkovic, Mario Cagalj, Toni Mastelic, Nitesh Saxena, and Dinko Begusic. "Secure Initialization of Multiple Constrained Wireless Devices for an Unaided User." In: *IEEE Transactions on Mobile Computing* 11.2 (2011), pp. 337–351.

[268] Samuel David Perli, Nabeel Ahmed, and Dina Katabi. "PixNet: Interference-free Wireless Links Using LCD-camera Pairs." In: *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking*. 2010, pp. 137–148.

[269] Adrian Perrig and Dawn Song. "Hash Visualization: A New Technique to Improve Real-world Security." In: *International Workshop on Cryptographic Techniques and E-Commerce*. 1999, pp. 131–138.

[270] Giuseppe Petracca, Ahmad-Atamli Reineh, Yuqiong Sun, Jens Grossklags, and Trent Jaeger. "Aware: Preventing Abuse of Privacy-sensitive Sensors via Operation Bindings." In: *26th {USENIX} Security Symposium ({USENIX} Security 17)*. 2017, pp. 379–396.

[271] Timothy J Pierson, Xiaohui Liang, Ronald Peterson, and David Kotz. "Wanda: Securely Introducing Mobile Devices." In: *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE. 2016, pp. 1–9.

[272] Ramnath Prasad and Nitesh Saxena. "Efficient Device Pairing Using "Human-comparable" Synchronized Audiovisual Patterns." In: *International Conference on Applied Cryptography and Network Security*. Springer. 2008, pp. 328–345.

[214] PurpleSec LLC. *2020 Cyber Security Statistics The Ultimate List Of Stats, Data & Trends*. https://purplesec.us/resources/cyber-security-statistics/. 2020.

[273] Kun Qian, Chenshu Wu, Zheng Yang, Yunhao Liu, and Zimu Zhou. "PADS: Passive Detection of Moving Targets with Dynamic Speed Using PHY Layer Information." In: *2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE. 2014, pp. 1–8.

[274] Kiran K Rachuri, Theus Hossmann, Cecilia Mascolo, and Sean Holden. "Beyond Location Check-ins: Exploring Physical and Soft Sensing to Augment Social Check-in Apps." In: *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE. 2015, pp. 123–130.

[275] Mahmudur Rahman, Umut Topkara, and Bogdan Carbunar. "Seeing Is not Believing: Visual Verifications through Liveness Analysis Using Mobile Devices." In: *Proceedings of the 29th Annual Computer Security Applications Conference*. 2013, pp. 239–248.

[276] Sridhar Rajagopal, Richard D Roberts, and Sang-Kyu Lim. "IEEE 802.15. 7 Visible Light Communication: Modulation Schemes and Dimming Support." In: *IEEE Communications Magazine* 50.3 (2012), pp. 72–82.

[277] Soundarya Ramesh, Harini Ramprasad, and Jun Han. "Listen to Your Key: Towards Acoustics-based Physical Key Inference." In: *Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications*. 2020, pp. 3–8.

[278] Aanjhan Ranganathan and Srdjan Capkun. "Are We Really Close? Verifying Proximity in Wireless Systems." In: *IEEE Security & Privacy* 15.3 (2017), pp. 52–58.

[279] Theodore S Rappaport, Shu Sun, Rimma Mayzus, Hang Zhao, Yaniv Azar, Kevin Wang, George N Wong, Jocelyn K Schulz, Mathew Samimi, and Felix Gutierrez. "Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!" In: *IEEE Access* 1 (2013), pp. 335–349.

[280] Kasper Bonne Rasmussen and Srdjan Capkun. "Implications of Radio Fingerprinting on the Security of Sensor Networks." In: *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm 2007*. IEEE. 2007, pp. 331–340.

[281] Reza Rawassizadeh and David Kotz. "Datasets for Mobile, Wearable and IoT Research." In: *GetMobile: Mobile Computing and Communications* 20.4 (2017), pp. 5–7.

[282] Irving S Reed and Gustave Solomon. "Polynomial Codes over Certain Finite Fields." In: *Journal of the Society for Industrial and Applied Mathematics* 8.2 (1960), pp. 300–304.

[283] Jamie Rigg. *Smartphone Concept Incorporates LiFi Sensor for Receiving Light-based Data*. https://www.engadget.com/2014/01/11/oledcomm-lifi-smartphone-concept/. 2014.

[284] Ronald L Rivest and Adi Shamir. "How to Expose an Eavesdropper." In: *Communications of the ACM* 27.4 (1984), pp. 393–394.

[285] Marc Roeschlin, Ivan Martinovic, and Kasper Bonne Rasmussen. "Device Pairing at the Touch of an Electrode." In: *NDSS*. Vol. 18. 2018, pp. 18–21.

[286] Michael Roland, Josef Langer, and Josef Scharinger. "Applying Relay Attacks to Google Wallet." In: *2013 5th International Workshop on Near Field Communication (NFC)*. IEEE. 2013, pp. 1–6.

[287] Rodrigo Roman and Javier Lopez. "KeyLED-transmitting Sensitive Data over Out-of-band Channels in Wireless Sensor Networks." In: *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE. 2008, pp. 796–801.

[288] Stuart Rosen and Peter Howell. *Signals and Systems for Speech and Hearing*. Vol. 29. Brill, 2011.

[289] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. "Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations." In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI-17)*. 2017, pp. 2662–2670.

[290] Masoud Rostami, Ari Juels, and Farinaz Koushanfar. "Heart-to-heart (H2H) Authentication for Implanted Medical Devices." In: *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*. 2013, pp. 1099–1112.

[291] Volker Roth, Wolfgang Polak, Eleanor Rieffel, and Thea Turner. "Simple and Effective Defense against Evil Twin Access Points." In: *Proceedings of the 1st ACM Conference on Wireless Network Security*. 2008, pp. 220–235.

[292] Nirupam Roy, Mahanth Gowda, and Romit Roy Choudhury. "Ripple: Communicating through Physical Vibration." In: *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*. 2015, pp. 265–278.

[293] Nirupam Roy and Romit Roy Choudhury. "Listening through a Vibration Motor." In: *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. 2016, pp. 57–69.

[294] Jan Ruge, Jiska Classen, Francesco Gringoli, and Matthias Hollick. "Frankenstein: Advanced Wireless Fuzzing to Exploit New Bluetooth Escalation Targets." In: *29th {USENIX} Security Symposium ({USENIX} Security 20)*. 2020, pp. 19–36.

[295] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, 2009.

[296] Ahmad-Reza Sadeghi, Christian Wachsmann, and Michael Waidner. "Security and Privacy Challenges in Industrial Internet of Things." In: *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE. 2015, pp. 1–6.

[297] Kartik Sankaran, Minhui Zhu, Xiang Fa Guo, Akkihebbal L Ananda, Mun Choon Chan, and Li-Shiuan Peh. "Using Mobile Phone Barometer for Low-power Transportation Context Detection." In: *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*. 2014, pp. 191–205.

[298] Angelica Sanz. *IoT Connected Car 3: Seat and Volvo Customer Experience*. http://blogs.icemd.com/blog-iot-and-digital-marketing/iot-connected-car-seat-volvo-customer-experience/. 2016.

[299] Martina Angela Sasse, Sacha Brostoff, and Dirk Weirich. "Transforming the "Weakest Link"—a Human/computer Interaction Approach to Usable and effective Security." In: *BT Technology Journal* 19.3 (2001), pp. 122–131.

[300] Nitesh Saxena, Jan-Erik Ekberg, Kari Kostiainen, and N Asokan. "Secure Device Pairing Based on a Visual Channel: Design and Usability Study." In: *IEEE Transactions on Information Forensics and Security* 6.1 (2010), pp. 28–38.

[301] Nitesh Saxena and Md Borhan Uddin. "Secure Pairing of "Interface-constrained" Devices Resistant against Rushing User Behavior." In: *International Conference on Applied Cryptography and Network Security*. Springer. 2009, pp. 34–52.

[302] Nitesh Saxena, Md Borhan Uddin, Jonathan Voris, and N Asokan. "Vibrate-to-unlock: Mobile Phone Assisted User Authentication to Multiple Personal RFID Tags." In: *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE. 2011, pp. 181–188.

[303] Bruce Schneier. *Secrets and Lies: Digital Security in a Networked World*. John Wiley & Sons, 2015.

[304] Tobias Schultes, Markus Grau, Daniel Steinmetzer, and Matthias Hollick. "Far Away and Yet Nearby-A Framework for Practical Distance Fraud on Proximity Services for Mobile Devices." In: *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. 2016, pp. 205–207.

[305] B Schultz. "802.11 ad–WLAN at 60 GHz–A Technology Introduction." In: *Rohde & Schwarz* (2013).

[306] Matthias Schulz, Francesco Gringoli, Daniel Steinmetzer, Michael Koch, and Matthias Hollick. "Massive Reactive Smartphone-based Jamming Using Arbitrary Waveforms and Adaptive Power Control." In: *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 2017, pp. 111–121.

[307] Matthias Schulz, Jakob Link, Francesco Gringoli, and Matthias Hollick. "Shadow Wi-Fi: Teaching Smartphones to Transmit Raw Signals and to Extract Channel State Information to Implement Practical Covert Channels over Wi-Fi." In: *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. 2018, pp. 256–268.

[308] Matthias Schulz, Daniel Wegemer, and Matthias Hollick. *Nexmon: The C-based Firmware Patching Framework*. https://nexmon.org. 2017.

[309] Dominik Schürmann, Arne Brüsch, Stephan Sigg, and Lars Wolf. "BANDANA—Body Area Network Device-to-device Authentication Using Natural Gait." In: *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE. 2017, pp. 190–196.

[310] Dominik Schürmann and Stephan Sigg. "Secure Communication Based on Ambient Audio." In: *IEEE Transactions on Mobile Computing* 12 (2013), pp. 358–370.

[311] Dipankar Sen, Prosenjit Sen, and Anand M Das. *RFID for Energy & Utility Industries*. Pennwell Books, 2009.

[312] Sougata Sen and David Kotz. "VibeRing: Using Vibrations from a Smart Ring as an Out-of-band Channel for Sharing Secret Keys." In: *Proceedings of the 10th International Conference on the Internet of Things*. 2020, pp. 1–8.

[313] Souvik Sen, Božidar Radunovic, Romit Roy Choudhury, and Tom Minka. "You Are Facing the Mona Lisa: Spot Localization Using PHY Layer Information." In: *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*. 2012, pp. 183–196.

[314] Sensirion AG. *VOC Sensor SGP30 / SGPC3 (NRND)*. https://www.sensirion.com/en/environmental-sensors/gas-sensors/sgp30/. 2021.

[315] Mohit Sethi, Markku Antikainen, and Tuomas Aura. "Commitment-based Device Pairing with Synchronized Drawing." In: *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE. 2014, pp. 181–189.

[316] Wenlong Shen, Bo Yin, Xianghui Cao, Lin X Cai, and Yu Cheng. "Secure Device-to-device Communications over WiFi Direct." In: *IEEE Network* 30.5 (2016), pp. 4–9.

[317] Carlton Shepherd, Iakovos Gurulian, Eibe Frank, Konstantinos Markantonakis, Raja Naeem Akram, Emmanouil Panaousis, and Keith Mayes. "The Applicability of Ambient Sensors as Proximity Evidence for NFC Transactions." In: *2017 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2017, pp. 179–188.

[318] Cong Shi, Jian Liu, Hongbo Liu, and Yingying Chen. "Smart User Authentication through Actuation of Daily Activities Leveraging WiFi-enabled IoT." In: *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 2017, pp. 1–10.

[319] Lu Shi, Ming Li, Shucheng Yu, and Jiawei Yuan. "BANA: Body Area Network Authentication Exploiting Channel Characteristics." In: *IEEE Journal on Selected Areas in Communications* 31.9 (2013), pp. 1803–1816.

[320] Yi Shi, Kemal Davaslioglu, Yalin E Sagduyu, William C Headley, Michael Fowler, and Gilbert Green. "Deep Learning for RF Signal Classification in Unknown and Dynamic Spectrum Environments." In: *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE. 2019, pp. 1–10.

[321] Babins Shrestha, Manar Mohamed, and Nitesh Saxena. "ZEMFA: Zero-effort Multi-factor Authentication Based on Multi-modal Gait Biometrics." In: *2019 17th International Conference on Privacy, Security and Trust (PST)*. IEEE. 2019, pp. 1–10.

[322] Babins Shrestha, Manar Mohamed, Sandeep Tamrakar, and Nitesh Saxena. "Theft-resilient Mobile Wallets: Transparently Authenticating NFC Users with Tapping Gesture Biometrics." In: *Proceedings of the 32nd Annual Conference on Computer Security Applications*. 2016, pp. 265–276.

[323] Babins Shrestha, Nitesh Saxena, Hien Thi Thu Truong, and N Asokan. "Drone to the Rescue: Relay-resilient Authentication Using Ambient Multi-sensing." In: *International Conference on Financial Cryptography and Data Security*. Springer. 2014, pp. 349–364.

[324] Babins Shrestha, Nitesh Saxena, Hien Thi Thu Truong, and N Asokan. "Sensor-based Proximity Detection in the Face of Active Adversaries." In: *IEEE Transactions on Mobile Computing* 18.2 (2018), pp. 444–457.

[325] Babins Shrestha, Maliheh Shirvanian, Prakash Shrestha, and Nitesh Saxena. "The Sounds of the Phones: Dangers of Zero-effort Second Factor Login based on Ambient Audio." In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 2016, pp. 908–919.

[326] Dave Shuman. *Looking under the Hood of the Connected Car*. https://www.techradar.com/news/looking-under-the-hood-of-the-connected-car. 2019.

[327] Shure, Inc. *Microphone Techniques*. https://www.shure.com/damfiles/default/global/documents/publications/en/performance-production/microphone_techniques_for_recording_english.pdf-bb0469316afdb6118691d2f3f5e3ff01.pdf. 2014.

[328] Sabrina Sicari, Alessandra Rizzardi, Luigi Alfredo Grieco, and Alberto Coen-Porisini. "Security, Privacy and Trust in Internet of Things: The Road Ahead." In: *Computer Networks* 76 (2015), pp. 146–164.

[329] Amit Kumar Sikder, Hidayet Aksu, and A Selcuk Uluagac. "6thsense: A Context-aware Sensor-based Attack Detector for Smart Devices." In: *26th {USENIX} Security Symposium ({USENIX} Security 17)*. 2017, pp. 397–414.

[330] Amit Kumar Sikder, Giuseppe Petracca, Hidayet Aksu, Trent Jaeger, and A Selcuk Uluagac. "A Survey on Sensor-based Threats to Internet-of-Things (IoT) Devices and Applications." In: *arXiv preprint arXiv:1802.02041* (2018).

[331] Sho Sonoda and Noboru Murata. "Neural Network with Unbounded Activation Functions Is Universal Approximator." In: *Applied and Computational Harmonic Analysis* 43.2 (2017), pp. 233–268.

[332] Claudio Soriente, Gene Tsudik, and Ersin Uzun. "BEDA: Button-Enabled Device Pairing." In: *IACR Cryptology ePrint Archive* 2007 (2007), p. 246.

[333] Claudio Soriente, Gene Tsudik, and Ersin Uzun. "HAPADEP: Human-assisted Pure Audio Device Pairing." In: *International Conference on Information Security*. Springer. 2008, pp. 385–400.

[334] Square, Inc. *How NFC Works*. http://nearfieldcommunication.org/how-it-works.html. 2017.

[335] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[336] Frank Stajano and Ross Anderson. "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks." In: *International Workshop on Security Protocols*. Springer. 1999, pp. 172–182.

[337] National Institute of Standards and Technology. *EntropyAssessment*. https://github.com/usnistgov/SP800-90B_EntropyAssessment. 2019.

[338] Statista. *Global Shipments of Smartphones with a Screen Size of 5 Inches or Larger from 2012 to 2016 (in Million Units)*. http://www.statista.com/statistics/253350/shipments-of-smartphones-with-screen-size-5-inches-orlarger/. 2013.

[339] Statista. *Internet of Things (IoT) Connected Devices Installed Base Worldwide from 2015 to 2025*. https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/. 2016.

[340] Statista. *Number of Network Connected Devices per Person around the World from 2003 to 2020*. https://www.statista.com/statistics/678739/forecast-on-connected-devices-per-person/. 2016.

[341] Daniel Steinmetzer, Joe Chen, Jiska Classen, Edward Knightly, and Matthias Hollick. "Eavesdropping with Periscopes: Experimental Security Analysis of Highly Directional Millimeter Waves." In: *2015 IEEE Conference on Communications and Network Security (CNS)*. IEEE. 2015, pp. 335–343.

[342] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. "Smart Devices Are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity Recognition." In: *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. 2015, pp. 127–140.

[343] R Dean Straw, LB Cebik, D Halliday, D Jansson, R Lewallen, R Severns, and F Witt. "The ARRL Antenna Book: The Ultimate Reference for Amateur Radio Antennas." In: *Amer Radio Relay League* (2003).

[344] Ahren Studer, Timothy Passaro, and Lujo Bauer. "Don't Bump, Shake on It: The Exploitation of a Popular Accelerometer-based Smart Phone Exchange and Its Secure Replacement." In: *Proceedings of the 27th Annual Computer Security Applications Conference*. 2011, pp. 333–342.

[345] Milan Stute. "Availability by Design: Practical Denial-of-service-Resilient Distributed Wireless Networks." In: (2020).

[346] Milan Stute, David Kreitschmann, and Matthias Hollick. "One Billion Apples' Secret Sauce: Recipe for the Apple Wireless Direct Link Ad hoc Protocol." In: *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. 2018, pp. 529–543.

[347] Milan Stute, Sashank Narain, Alex Mariotto, Alexander Heinrich, David Kreitschmann, Guevara Noubir, and Matthias Hollick. "A Billion Open Interfaces for Eve and Mallory: MitM, DoS, and Tracking Attacks on iOS and MacOS through Apple Wireless Direct Link." In: *28th {USENIX} Security Symposium ({USENIX} Security 19)*. 2019, pp. 37–54.

[348] Yingnan Sun, Charence Wong, Guang-Zhong Yang, and Benny Lo. "Secure Key Generation Using Gait Features for Body Sensor Networks." In: *2017 IEEE 14th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. IEEE. 2017, pp. 206–210.

[349] Jani Suomalainen. "Smartphone Assisted Security Pairings for the Internet of Things." In: *2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE)*. IEEE. 2014, pp. 1–5.

[350] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. "A Survey on Deep Transfer Learning." In: *International Conference on Artificial Neural Networks*. Springer. 2018, pp. 270–279.

[351] Fei Tang and Hemant Ishwaran. "Random Forest Missing Data Algorithms." In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 10.6 (2017), pp. 363–377.

[352] Twain Taylor. *6 Most Commonly Used IoT Communication Protocols*. http://techgenix.com/iot-communication-protocols/. 2019.

[353] Keras Development Team. *Keras: The Python Deep Learning library*. https://keras.io/. 2020.

[354] TensorFlow Development Team. *Deploy Machine Learning Models on Mobile and IoT Devices*. https://www.tensorflow.org/lite. 2020.

[355] The Pyca/cryptography Team. *pyca/cryptography*. https://github.com/pyca/cryptography. 2020.

[356] Jenny Terzic, Edin Terzic, Romesh Nagarajah, and Muhammad Alamgir. *Ultrasonic Fluid Quantity Measurement in Dynamic Vehicular Applications*. Springer, 2013.

[357] The H2O.ai Team. *H2O: Scalable Machine Learning*. http://www.h2o.ai. 2015.

[358] The MathWorks, Inc. *Bandpass IIR Filter*. https://mathworks.com/help/signal/ref/designfilt.html. 2018.

[359] The MathWorks, Inc. *Cross-correlation*. https://mathworks.com/help/signal/ref/xcorr.html#bual1fd-maxlag. 2018.

[360] James Thrasher. *How is RFID Used in the Real World*. http://blog.atlasrfidstore.com/what-is-rfid-used-for-in-applications. 2013.

[361] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. "On the Requirements for Successful GPS Spoofing Attacks." In: *Proceedings of the 18th ACM Conference on Computer and Communications Security*. 2011, pp. 75–86.

[362] Lam Tran, Thao Dang, and Deokjai Choi. "A Binarization Method for Extracting High Entropy String in Gait Biometric Cryptosystem." In: *Proceedings of the 9th International Symposium on Information and Communication Technology*. 2018, pp. 273–280.

[363] Sivaram Alukuru Trikutam. *Driving the Connected Car Revolution*. https://www.cypress.com/blog/corporate/driving-connected-car-revolution. 2019.

[364] Hien Thi Thu Truong, Xiang Gao, Babins Shrestha, Nitesh Saxena, N Asokan, and Petteri Nurmi. "Comparing and Fusing Different Sensor Modalities for Relay Attack Resistance in Zero-interaction Authentication." In: *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE. 2014, pp. 163–171.

[365] Hien Thi Thu Truong, Juhani Toivonen, Thien Duc Nguyen, Claudio Soriente, Sasu Tarkoma, and N Asokan. "DoubleEcho: Mitigating Context-manipulation Attacks in Copresence Verification." In: *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE. 2019, pp. 1–9.

[366] Maximilian von Tschirschnitz, Ludwig Peuckert, Fabian Franzen, and Jens Grossklags. "Method Confusion Attack on Bluetooth Pairing." In: *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*. 2021.

[367] Yazhou Tu, Zhiqiang Lin, Insup Lee, and Xiali Hei. "Injected and Delivered: Fabricating Implicit Control over Actuation Systems by Spoofing Inertial Sensors." In: *27th {USENIX} Security Symposium ({USENIX} Security 18)*. 2018, pp. 1545–1562.

[368] Meltem Sönmez Turan, Elaine Barker, John Kelsey, Kerry A McKay, Mary L Baish, and Mike Boyle. "Recommendation for the Entropy Sources Used for Random Bit Generation." In: *NIST Special Publication* 800.90B (2018).

[369] Ersin Uzun, Kristiina Karvonen, and Nadarajah Asokan. "Usability Analysis of Secure Pairing Methods." In: *International Conference on Financial Cryptography and Data Security*. Springer. 2007, pp. 307–324.

[370] Ersin Uzun, Nitesh Saxena, and Arun Kumar. "Pairing Devices for Social Interactions: A Ccomparative Usability Evaluation." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2011, pp. 2315–2324.

[371] Christian Vaas, Mika Juuti, N Asokan, and Ivan Martinovic. "Get in Line: Ongoing Co-presence Verification of a Vehicle Formation Based on Driving Trajectories." In: *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2018, pp. 199–213.

[372] Alex Varshavsky, Adin Scannell, Anthony LaMarca, and Eyal De Lara. "Amigo: Proximity-based Authentication of Mobile Devices." In: *International Conference on Ubiquitous Computing*. Springer. 2007, pp. 253–270.

[373] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention Is All You Need." In: *Advances in Neural Information Processing Systems*. 2017, pp. 5998–6008.

[374] Anran Wang, Shuai Ma, Chunming Hu, Jinpeng Huai, Chunyi Peng, and Guobin Shen. "Enhancing Reliability to Boost the Throughput over Screen-camera Links." In: *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*. 2014, pp. 41–52.

[375] Qian Wang, Man Zhou, Kui Ren, Tao Lei, Jikun Li, and Zhibo Wang. "Rain Bar: Robust Application-driven Visual Communication Using Color Barcodes." In: *2015 IEEE 35th International Conference on Distributed Computing Systems*. IEEE. 2015, pp. 537–546.

[376] Wei Wang, Jingqiang Lin, Zhan Wang, Ze Wang, and Luning Xia. "vBox: Proactively Establishing Secure Channels between Wireless Devices without Prior Knowledge." In: *European Symposium on Research in Computer Security*. Springer. 2015, pp. 332–351.

[377] Wei Wang, Zhan Wang, Wen Tao Zhu, and Lei Wang. "WAVE: Secure Wireless Pairing Exploiting Human Body Movements." In: *2015 IEEE Trustcom/BigDataSE/ISPA*. Vol. 1. IEEE. 2015, pp. 1243–1248.

[378] Wei Wang, Lin Yang, and Qian Zhang. "Touch-and-guard: Secure Pairing through Hand Resonance." In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2016, pp. 670–681.

[379] Xuyu Wang, Lingjun Gao, and Shiwen Mao. "BiLoc: Bi-modal Deep Learning for Indoor Localization with Commodity 5GHz WiFi." In: *IEEE Access* 5 (2017), pp. 4209–4220.

[380] Christopher Ware, John Judge, Joe Chicharo, and Eryk Dutkiewicz. "Unfairness and Capture Behaviour in 802.11 Adhoc Networks." In: *2000 IEEE International Conference on Communications. ICC 2000. Global Convergence Through Communications. Conference Record*. Vol. 1. IEEE. 2000, pp. 159–163.

[381] Geoffrey I Webb. "Decision Tree Grafting from the All-tests-but-one Partition." In: *Ijcai*. Vol. 2. 1999, pp. 702–707.

[382] Geoffrey I Webb. "Multiboosting: A Technique for Combining Boosting and Wagging." In: *Machine Learning* 40.2 (2000), pp. 159–196.

[383] Stephen A Weis. "RFID (Radio Frequency Identification): Principles and Applications." In: *System* 2.3 (2007), pp. 1–23.

[384] Gordon Wetzstein. *Inertial Measurement Units I*. https://stanford.edu/class/ee267/lectures/lecture9.pdf. 2019.

[385] Ryan Whitwam. *How Google Nearby Works, and How You Can Take Advantage of It*. https://www.greenbot.com/article/3078180/how-google-nearby-works-and-how-you-can-take-advantage-of-it.html. 2016.

[386] Primal Wijesekera, Arjun Baokar, Ashkan Hosseini, Serge Egelman, David Wagner, and Konstantin Beznosov. "Android Permissions Remystified: A Field Study on Contextual Integrity." In: *24th {USENIX} Security Symposium ({USENIX} Security 15)*. 2015, pp. 499–514.

[387] Myounggyu Won, Shaohu Zhang, and Sang H Son. "WiTraffic: Low-cost and Non-intrusive Traffic Monitoring System Using WiFi." In: *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE. 2017, pp. 1–9.

[388] Yuezhong Wu, Qi Lin, Hong Jia, Mahbub Hassan, and Wen Hu. "Auto-Key: Using Autoencoder to Speed Up Gait-based Key Generation in Body Area Networks." In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4.1 (2020), pp. 1–23.

[389] Wei Xi, Chen Qian, Jinsong Han, Kun Zhao, Sheng Zhong, Xiang-Yang Li, and Jizhong Zhao. "Instant and Robust Authentication and Key Agreement among Mobile Devices." In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 2016, pp. 616–627.

[390] Yaxiong Xie, Zhenjiang Li, and Mo Li. "Precise Power Delay Profiling with Commodity WiFi." In: *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. 2015, pp. 53–64.

[391] Jie Xiong and Kyle Jamieson. "Securearray: Improving WiFi Security with Fine-grained Physical-layer Information." In: *Proceedings of the 19th annual international conference on Mobile computing & networking*. 2013, pp. 441–452.

[392] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. "Empirical Evaluation of Rectified Activations in Convolutional Network." In: *arXiv preprint arXiv:1505.00853* (2015).

[393] Weitao Xu, Girish Revadigar, Chengwen Luo, Neil Bergmann, and Wen Hu. "Walkie-talkie: Motion-assisted Automatic Key Generation for Secure On-body Device Communication." In: *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE. 2016, pp. 1–12.

[394] Weitao Xu, Junqing Zhang, Shunqi Huang, Chengwen Luo, and Wei Li. "Key Generation for Internet of Things: A Contemporary Survey." In: *ACM Computing Surveys (CSUR)* 54.1 (2021), pp. 1–37.

[395] Jiang Xue, Sudip Biswas, Ali Cagatay Cirik, Huiqin Du, Yang Yang, Tharmalingam Ratnarajah, and Mathini Sellathurai. "Transceiver Design of Optimum Wirelessly Powered Full-duplex MIMO IoT Devices." In: *IEEE Transactions on Communications* 66.5 (2018), pp. 1955–1969.

[396] Zhenyu Yan, Qun Song, Rui Tan, Yang Li, and Adams Wai Kin Kong. "Towards Touch-to-access Device Authentication Using Induced Body Electric Potentials." In: *The 25th Annual International Conference on Mobile Computing and Networking*. 2019, pp. 1–16.

[397] Lin Yang, Wei Wang, and Qian Zhang. "Secret from Muscle: Enabling Secure Pairing with Electromyography." In: *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. 2016, pp. 28–41.

[398] Qinggang Yue, Zhen Ling, Xinwen Fu, Benyuan Liu, Kui Ren, and Wei Zhao. "Blind Recognition of Touched Keys on Mobile Devices." In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 2014, pp. 1403–1414.

[399] Murtaza Zafer, Dakshi Agrawal, and Mudhakar Srivatsa. "Limitations of Generating a Secret Key Using Wireless Fading under Active Adversary." In: *IEEE/ACM Transactions on Networking* 20.5 (2012), pp. 1440–1451.

[400] Andra Zaharia. *300+ Terrifying Cybercrime and Cybersecurity Statistics & Trends (2021 EDITION)*. https://www.comparitech.com/vpn/cybersecurity-cyber-crime-statistics-facts-trends/. 2020.

[401] Christian T Zenger, Mario Pietersz, Jan Zimmer, Jan-Felix Posielek, Thorben Lenze, and Christof Paar. "Authenticated Key Establishment for Low-resource Devices Exploiting Correlated Random Channels." In: *Computer Networks* 109 (2016), pp. 105–123.

[402] Bingsheng Zhang, Kui Ren, Guoliang Xing, Xinwen Fu, and Cong Wang. "SBVLC: Secure Barcode-based Visible Light Communication for Smartphones." In: *IEEE Transactions on Mobile Computing* 15.2 (2016), pp. 432–446.

[403] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. "Dolphinattack: Inaudible Voice Commands." In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017, pp. 103–117.

[404] Jiansong Zhang, Zeyu Wang, Zhice Yang, and Qian Zhang. "Proximity Based IoT Device Authentication." In: *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE. 2017, pp. 1–9.

[405] Tengxiang Zhang, Xin Yi, Ruolin Wang, Yuntao Wang, Chun Yu, Yiqin Lu, and Yuanchun Shi. "Tap-to-Pair: Associating Wireless Devices with Synchronous Tapping." In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.4 (2018), pp. 1–21.

[406] Cong Zhao, Shusen Yang, Xinyu Yang, and Julie A McCann. "Rapid, User-transparent, and Trustworthy Device Pairing for D2D-enabled Mobile Crowdsourcing." In: *IEEE Transactions on Mobile Computing* 16.7 (2016), pp. 2008–2022.

[407] Yiyang Zhao, Chen Qian, Liangyi Gong, Zhenhua Li, and Yunhao Liu. "LMDD: Light-weight Magnetic-based Door Detection with Your Smartphone." In: *2015 44th International Conference on Parallel Processing*. IEEE. 2015, pp. 919–928.

[408] Alice Zheng and Amanda Casari. *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. " O'Reilly Media, Inc.", 2018.

[409] Ruogu Zhou and Guoliang Xing. "nshield: A Noninvasive NFC Security System for Mobile Devices." In: *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*. 2014, pp. 95–108.

[410] Shuangyi Zhu, Yuan Ma, Tianyu Chen, Jingqiang Lin, and Jiwu Jing. "Analysis and Improvement of Entropy Estimators in NIST SP 800-90B for Non-IID Entropy Sources." In: *IACR Transactions on Symmetric Cryptology* (2017), pp. 151–168.

[411] Yanzi Zhu, Zhujun Xiao, Yuxin Chen, Zhijing Li, Max Liu, Ben Y Zhao, and Haitao Zheng. "Et Tu Alexa? When Commodity WiFi Devices Turn into Adversarial Motion Sensors." In: *27th Annual Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2020.

ERKLÄRUNG ZUR DISSERTATIONSSCHRIFT

gemäß § 9 der Allgemeinen Bestimmungen der Promotionsordnung der Technische Universität Darmstadt vom 12. Januar 1990 (ABI. 1990, S. 658) in der Fassung der 8. Novelle vom 1. März 2018

Hiermit versichere ich, Mikhail Fomichev, die vorliegende Dissertationsschrift ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Eigenzitate aus vorausgehenden wissenschaftlichen Veröffentlichungen werden in Anlehnung an die Hinweise des Promotionsausschusses Fachbereich Informatik zum Thema „Eigenzitate in wissenschaftlichen Arbeiten“ (EZ-2014/10) in Kapitel „*Collaborations and My Contribution*“ auf Seiten xxiii bis xxiv gelistet. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. In der abgegebenen Dissertationsschrift stimmen die schriftliche und die elektronische Fassung überein.

Darmstadt, 15. Juli 2021

Mikhail Fomichev